An Asynchronous Ensemble-Averaging Approach to CMFD Source Acceleration: Rearchitecting Monte Carlo Reactor Simulation Paradigms for the Exasacle Computing Age

by

Shikhar Kumar
B.S., Columbia University (2015)
M.S., Massachusetts Institute of Technology (2018)

Submitted to the Department of Nuclear Science and Engineering
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Nuclear Science and Engineering
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 2021

Author .....
Department of Nuclear Science and Engineering
January 31, 2021

Certified by .....
Benoit Forget
Associate Department Head, Professor of Nuclear Science and Engineering
Thesis Supervisor

Certified by .....
Kord Smith
KEPCO Professor of the Practice of Nuclear Science and Engineering
Thesis Supervisor

Certified by .....
Paul Romano
Computational Scientist, Argonne National Laboratory
Thesis Reader

Accepted by .....
Ju Li
Battelle Energy Alliance Professor of Nuclear Science and Engineering
Chair, Committee on Graduate Students

An Asynchronous Ensemble-Averaging Approach to CMFD Source Acceleration: Rearchitecting Monte Carlo Reactor Simulation Paradigms for the Exascale Computing Age

by

Shikhar Kumar

Submitted to the Department of Nuclear Science and Engineering
on Feburary 23, 2021, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Nuclear Science and Engineering

Abstract

The safe and reliable operation of a nuclear power plant requires a meticulous understanding of the various physics phenomena at play within the reactor core. Monte Carlo neutron transport methods are commonly used for the purpose of high-fidelity full-core reactor analysis, stochastically sampling a large number of neutron histories throughout the reactor core with minimal approximations to the underlying phase space of the problem. Monte Carlo methods, however, face several issues, primary of which is the prohibitively large number of computational resources required for tallied quantities to reach adequate uncertainty levels for full-core analysis. Advances in High-Performance Computing (HPC) systems and significant code parallelization of Monte Carlo algorithms have helped to reduce overall time to solution. This thesis extends these efforts by proposing a novel *ensemble averaged Coarse Mesh Finite Difference (CMFD)* simulation strategy in order to improve parallel efficiency on HPC systems while also addressing other issues surrounding Monte Carlo methods including fission source stationarity, real variance calculations, and nonlinear coupling schemes.

This thesis begins by proposing a fission source stationarity diagnostic that monitors the behavior of Functional Expansion Tally (FET) coefficients. This method relies on the efficient computation of meshless quantities that signal whether various spatial modes of the fission source have stopped fluctuating within statistical limits, avoiding issues of false convergence that are prevalent with more commonly used diagnostics that rely on Shannon entropy calculations. This work provides the framework for establishing total runtimes associated with Monte Carlo inactive generations, as well as a heuristic for exact runtime savings from source acceleration methods such as CMFD.

Next, a rigorous sensitivity study of CMFD source acceleration is conducted in order to pinpoint the exact parameters that affect the evolution of Monte Carlo simulations in the inactive and active generations. Here, it is found that coarse CMFD meshes should be used with an expanding windowing scheme that employs a relatively large maximum window size in order to reduce overall runtime while ensuring that variance

levels are on par with un-accelerated simulations. Runtime analysis also indicates that a hybrid solution that combined a loosely-converged CMFD solution during initial inactive generations while reverting to un-accelerated Monte Carlo during the later inactive generations should be employed in order to reduce overall runtime to stationarity, continuing to turn off CMFD feedback into the active generations as well.

Finally, a comparison of various Monte Carlo run strategies motivates the development of the ensemble averaged CMFD simulation strategy, which is shown to be the optimal simulation paradigm on large-scale computing systems. From the perspective of parallel efficiency, ensemble averaged CMFD is shown to improve load balancing by 5-10% on representative 1-D, 2-D, and 3-D reactor problems, while the use of an aggregate seed stopping criteria ensures that initial error levels converge at ideal $1/\sqrt{N}$ rates, where N is the total number of Monte Carlo ensembles simulated in parallel. Ensemble averaging also provides a simple and accurate method for calculating real variance estimates across statistically independent ensembles of neutrons, something that is not typically possible with single seed Monte Carlo simulations. Moreover, the asynchronous ensemble averaged CMFD implementation provides better fault tolerance against possible node failures on large-scale computer facilities. Finally, arguments for how ensemble averaging can be applied to nonlinear multiphysics coupling schemes as well as Graphics Processing Unit (GPU)-based computer systems are also presented.

Thesis Supervisor: Benoit Forget
Title: Associate Department Head, Professor of Nuclear Science and Engineering

Thesis Supervisor: Kord Smith
Title: KEPCO Professor of the Practice of Nuclear Science and Engineering

Thesis Reader: Paul Romano
Title: Computational Scientist, Argonne National Laboratory

Acknowledgments

This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of two U.S. Department of Energy organizations (Office of Science and the National Nuclear Security Administration) responsible for the planning and preparation of a capable exascale ecosystem, including software, applications, hardware, advanced system engineering, and early testbed platforms, in support of the nation's exascale computing imperative. This research also made use of the resources of the High Performance Computing Center at Idaho National Laboratory, which is supported by the Office of Nuclear Energy of the U.S. Department of Energy and the Nuclear Science User Facilities under Contract No. DE-AC07-05ID14517. Additionally, this research gratefully acknowledges the use of the Bebop cluster in the Laboratory Computing Resource Center at Argonne National Laboratory.

I am incredibly indebted to my research advisors Professors Benoit Forget and Kord Smith who supported me tremendously during my Ph.D. and Master's research. I was lucky to take all of my reactor physics classes with Kord, who pushed his students hard to make sure they mastered the fundamentals of nuclear engineering. His guidance also helped steer my Ph.D. research in the right direction on countless number of occasions. Ben was generous with his time in mentoring me during the initial stages of my graduate school career, and was a frequent source of support during the latter stages, providing me with the vision and motivation to pursue my research goals, especially during times when I saw no clear way forward.

Next, I would like to acknowledge Professor Koroush Shirvan, who I interacted with to a large capacity during my initial years at MIT. He showed me the ropes with using core design and analysis tools, and was always willing to help me with my research struggles, no matter how basic they might have been.

I am also especially thankful to my current and former peers in the Computational Reactor Physics Group (CRPG) for their constant help and camaraderie throughout my Ph.D. journey, with a special thank you to Dr. Paul Romano and Dr. Sterling Harper for sitting down with me for countless number of hours to explain the OpenMC

code development process and provide me with invaluable advice on proper software development practices.

I also owe my gratitude to my roommates in graduate school - Dr. Guillaume Giudicelli, Dr. Stephen Lam, Lord Hobo, and Pawla "Kitty Perry". They carried me every step of the way through these past five and a half years, from helping me with the NSE module classes to passing the Ph.D. qualifying exams to providing me with countless advice about my research and career aspirations.

Finally, I express my deepest appreciation to all of my friends and family who helped me stay sane during these trying years. I am grateful to Lisa Magnano-Bleheen for always being a friendly and uplifting personality in the department, never hesitating to put her other commitments aside if I needed someone to talk to. I would also like to thank my parents Parag Kumar and Mukta Singh for their support, love, and encouragement. They provided me with immense autonomy from an early age and have always been supportive of my academic endeavors. Lastly, I am incredibly thankful to have my girlfriend Amanda Willis by my side during the final years of my Ph.D. She was a calming presence when the stresses of graduate school seemed insurmountable, and I relied on her companionship heavily during the extremely isolating last few months of my Ph.D. I am excited for the adventures that lie ahead of us.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Definitions and Acronyms

ACC	Autocorrelation Cofficient
AMD	Advanced Mico Devices
ANL	Argonne National Laboratory
AR	autoregressive
BEAVRS	Benchmark for Evaluation and Validation of Reactor Simulations
CMFD	Coarse Mesh Finite Difference
CFD	Computational Fluid Dynamics
CMM	Cumulative Migration Method
CPU	Central Processing Unit
FET	Functional Expansion Tally
GPGPU	General-Purpose Graphics Processing Unit
GPU	Graphics Processing Unit
HZP	Hot Zero Power
HPC	High-Performance Computing
INL	Idaho National Laboratory
LWR	Light Water Reactor
MC	Monte Carlo
MGXS	Multi-Group Cross Sections
MOC	Method of Characteristics
MPI	Message Passing Interface
NEA	Nuclear Energy Agency
ORNL	Oak Ridge National Laboratory
PWR	Pressurized Water Reactor
RMA	Remote Memory Access
RMS	Root Mean Square
SIMD	Single-Instruction Multiple-Data

SMR	Small Modular Reactor
SPMD	Single-Program Multiple-Data
TH	Thermal Hydraulic

Part I

Introduction

Chapter 1

Introduction and Background

1.1 Motivation

The safe and reliable operation of a nuclear power plant requires a meticulous understanding of the various physics phenomena at play within the reactor core. The frequency and vast diversity of particle interactions within a nuclear reactor dictate the isotopic evolution of materials that constitute the reactor core, the temperature distributions of fuel and coolant media, and any short-term transients that could potentially lead to accidents. Numerical simulations play an integral role in the field of nuclear reactor physics, whereby the neutron physics that govern reactor core conditions are modeled to accurately predict the power production, core reactivity, isotopic depletion, and time-dependent behavior of nuclear systems. Advances in such nuclear reactor simulations have contributed immensely to the safety and performance of the current fleet of Light Water Reactors (LWR's). Moreover, the predictive capabilities of reactor simulations are being used today to simulate novel reactor concepts, facilitating the design and development of next-generation nuclear reactors.

Current industrial standards in full-core Light Water Reactor (LWR) simulations employ nodal diffusion methods, which trade off modeling accuracy for computational speed. At the core-level, such methods assume a smoothly varying neutron flux distribution over a coarse spatial and energy grid and utilize a diffusion-based solution to solve for power distributions within the core. Macroscopic cross sections that serve as the

inputs to the diffusion equation are homogenized over assembly or quarter-assembly regions and a few energy groups, and these cross sections are interpolated over a wide range of Thermal Hydraulic (TH) conditions to solve the coupled neutron diffusion-TH equations. While a number of approximations are embedded into this simulation procedure, the biggest draw for nodal diffusion methods lies in its time to solution, where full-core analyses can be conducted on the order of core-seconds and yield $\sim$1% RMS errors in pin power reconstructions at beginning of cycle, and 3-4% at high depletion [3].

The various assumptions underlying these methods, however, are highly specialized to the modeling of LWR's and are not easily transferable to arbitrary reactor configurations. Furthermore, the homogenization techniques used in nodal diffusion simulators make it difficult to capture highly localized flux gradients that may arise in more complex core designs. Thus, the nuclear industry has looked to high-fidelity tools for reactor modeling in order to reduce the number of approximations embedded into the simulation tool-chain and allow for more accurate solutions on finer grids.

High-fidelity tools are typically divided into two categories: deterministic and probabilistic methods. Deterministic methods such as Discrete Ordinates (S_N), Simplified P_N, and Method of Characteristics (MOC) all aim to solve the Boltzmann transport equation by discretizing the spatial, angular, and energy phase space variables on a much finer grid than a diffusion-based solution. However, these discretizations, especially in the energy domain, can still yield significant errors in the simulations. For example, the process of generating multigroup cross sections for deterministic calculations requires a collapse of pointwise continuous-energy cross section values over a bin of discrete energy groups, and depending on the placement of these group boundaries and the number of groups used, important energy self-shielding effects within the core can be poorly predicted. As the grid refinement of the problem increases, so too do the memory requirements for the problem. The issues from cross section generation are further compounded once TH coupling and depletion are introduced, as the multigroup cross sections now need to be generated as a function of energy, material density, and material temperature [4, Chapter 9].

In order to avoid the generation of multigroup cross sections altogether, probabilistic

Monte Carlo methods utilize pointwise continuous-energy nuclear data to stochastically sample individual neutrons from life to death through a reactor, aggregating bulk neutron behavior with some stochastic uncertainty to characterize distributions of interest. While Monte Carlo methods make minimal approximations to the underlying geometry of the problem and the energy grid, a substantial number of particles must be simulated in order to reach adequate uncertainty levels for full-core analyses. This combined with poor convergence rates can make Monte Carlo methods quite computationally burdensome.

Many of the computational difficulties associated with high-fidelity nuclear reactor simulations have been alleviated with the advancements in HPC systems built for scientific computing. The nation's top supercomputers are expected to achieve peak calculation rates on the order of exaflops (10^{18} floating-point operations per second), and this increase in computational capacity has paved the way for the reactor physics community to develop highly scalable simulation techniques for modeling three-dimensional full-core LWR reactors. The peak performance of a supercomputer is only one small component of what dictates the overall runtime of such simulations, however, and this thesis expounds on this topic by pinpointing key bottlenecks in current probabilistic simulation strategies and proposing a novel algorithm for Monte Carlo simulations that is better suited for such exascale HPC architectures.

1.2 The Monte Carlo Method

Monte Carlo simulations track the particle interactions of neutrons within nuclear reactors by stochastically sampling the transport of neutrons from life to death. Probability distributions based on nuclear data dictate the distances these particles travel after each collision, the types of collisions that occur, as well as the positions and outgoing energies and angles of particles after each collision. As such, Monte Carlo methods treat the underlying geometry and energy of the particles almost exactly without the need for any discretizations to the solution phase space. The method of successive generations links fission sites of a previous generation of particles to the source sites for the following

generation, and the statistically-averaged behavior of these particles accumulated from generation to generation determines the distributions of tallied quantities such as reactor power and fission rates with some stochastic uncertainty. The independent nature of neutron histories lends Monte Carlo methods quite naturally to the parallelization of transport algorithms on multi-core computer architectures [5]. However, billions or even trillions of particle histories still need to be simulated for reactor-type problems before stochastic uncertainties reach acceptable limits in order to account for the number of unknowns in the problem when small tally regions are used. Moreover, these particles are not truly independent from a statistical point of view and this further degrades convergence in the tally uncertainty rates . Thus, Monte Carlo methods pose a significant computational burden and require large-scale computing systems to reduce time-to-solution to adequate levels.

1.3 Evolution of HPC Architectures

In order to understand the parallel performance of Monte Carlo on large-scale computer systems, a brief discussion about the trends in supercomputing is instructive. The first computer to be coined a supercomputer was the Control Data Corporation (CDC) 6600, built in 1964 and achieving 3 megaflops on a single Central Processing Unit (CPU) machine (by comparison, the first Raspberry Pi built in 2012 was able to achieve 42 megaflops) [6]. Cooled with Freon circulation, the CDC 6600 was a relatively small machine, built with four racks measuring 0.89m in width, 1.83m in depth, and 0.89m in height each. Since then, the computing capabilities of computers grew exponentially in accordance with Moore's law [7], where manufacturing advances in the number of transistors able to be packed onto a single circuit allowed for the doubling of processor speeds roughly every 18 months. By 2006, however, single thread performance on CPU 's began to flatten out as thermal limits began to restrict the number of transistors that could be effectively utilized on a chip. With this realization, the computer industry in general moved towards the adoption of multi-core architectures in order to keep up with Moore's law, where the number of logical cores in the system started increasing

to compensate for the lack of improvement in single thread performance for each core. With such systems, two forms of parallelization enable the efficient use of modern HPC systems.

In the first paradigm - Single-Program Multiple-Data (SPMD) - multiple processors execute a single program independently of each other. Each processor has its own set of compute cores and unique memory so that processor units can operate on independent chunks of a larger problem without ever having to communicate with each other. Typically, however, most algorithms require some level of broadcasting data to all processes or gathering data across processors onto a single processor. In such cases, the *Message Passing Interface* can be used to transfer data across processors through a network connection. High-speed interconnects allow for efficient data transfer across compute nodes on HPC systems, but can create bottlenecks in parallel efficiency as processors can spend significant time idling waiting for data to be transferred or received.

The second paradigm - Single-Instruction Multiple-Data (SIMD) - utilizes vector processing to efficiently perform the same instruction across a vector of data elements. The set of instructions supported for vector arithmetic can be quite limited, thus making it difficult to adapt a wide range of algorithms to be well-suited for such vectorization. Regardless, most CPU architectures in modern supercomputers support SIMD vectorization. For example, Argonne National Laboratory (ANL)'s supercomputer Theta, built in 2016, uses Intel's Xeon Phi Knights Landing processors to achieve a peak processor speed of 11.7 petaflops. Each processor contains 64 cores and each core supports 2 512-bit SIMD vector units [8].

SIMD vectorization is the basis of parallelization for GPU's as well, where such chipsets possess a large number of simple cores optimized for vectorized calculations to achieve higher peak computational performance than CPU 's. This has led to the adoption of General-Purpose Graphics Processing Unit (GPGPU)'s, where GPU's originally designed for efficient graphics processing are now used in general scientific and machine learning domains to speed up applications. GPU's will also become the building blocks for exascale systems expected to come online in 2021. Oak Ridge National Laboratory (ORNL)'s exascale supercomputer Frontier will contain 4 Advanced Mico Devices (AMD) GPU's and

1 AMD CPU per node and ANL's exascale supercomputer Aurora will contain 2 Intel Xeon CPU's and 6 Intel Xe GPU's per node to achieve exascale computing capabilities [9, 10].

1.4 State of the Art in Monte Carlo Simulations

With these evolving computer architectures in leading HPC systems, application developers looking to develop scalable algorithms on such machines have had to significantly adapt their methods and techniques to ensure that they leverage the supercomputer's parallelisms in the most optimal ways possible. This is especially true for Monte Carlo simulations, where significant strides have been made to advance high-fidelity modeling capabilities with incorporation of multiphysics feedback on leadership class computing facilities. Nevertheless, many challenges still remain. To fully understand the existing shortcomings with running full-core Monte Carlo simulations on these machines, one must examine the advances in the field of Monte Carlo methods over the past decade. In 2012, Kelly et al. reported on the computational requirements surrounding Monte Carlo simulations when trying to perform full-core simulations, and in the process illustrated the many challenges that surrounded calculations of this scale at the time [11]. 200 billion active neutron histories and roughly 56,000 CPU-hours were required to converge local tallies on the Nuclear Energy Agency (NEA) Monte Carlo benchmark [12] to uncertainty levels where 95% of the tally regions had relative error levels in power density of 1% or less at the 95% confidence interval. In order to ensure that statistical uncertainties in the tally regions were not biased due to correlation effects, real variance estimates had to also be calculated through a batching algorithm aimed to de-correlate neutron histories from successive batches. Considering the computational resources required just to simulate a single steady-state state point for a simplified performance benchmark, the possibility of high-fidelity modeling was deemed unrealistic at the time when considering any non-linear coupling scheme to incorporate thermal hydraulic feedback, isotopic depletion, and/or transient analysis.

Since the publishing of the paper by Kelly et. al, however, much progress has been made in terms of reducing the computational requirements of Monte Carlo while

maintaining accurate statistics in tallied quantities. The improvements to parallel efficiency, acceleration methods, real variance calculations, and non-linear coupling capabilities will be examined further as these subjects motivate the objectives of this thesis.

1.4.1 Parallelization Efforts

As previously mentioned, the perceived independence of neutron histories in Monte Carlo simulations makes such methods simple to implement with SPMD parallelism. However, the highly disjoint memory accesses and the stochastic nature of particle interactions associated with Monte Carlo methods significantly degrade single core performance. For example, every sampled particle collision requires a cross section lookup in memory across a very wide range of energies. Memory bandwidth and performance have not kept up at the same rate as Moore's Law and thus frequent memory accesses required by Monte Carlo simulations place significant performance penalties when run on HPC systems. Likewise, the stochastic nature of particle histories implies that history-based Monte Carlo simulations are unable to leverage the SIMD vectorization capabilities provided by modern supercomputers, which require simple shared arithmetic operations on large arrays of data.

Nevertheless, numerous efforts have been made to improve the parallel performance of Monte Carlo transport with the goal of reducing time-to-solution on leadership computing facilities. These improvements can come in different forms. Firstly, algorithmic improvements to the Monte Carlo method such as parallelized fission bank algorithms and tally reduction schemes are able to significantly reduce the total communication costs from transferring data across independent processors on multi-core computer platforms [13, 14]. Similarly, strategies such as data decomposition via tally servers and domain decomposition effectively reduce the memory limitations imposed from full-core simulations that track the calculation of multiple tallies across various depletable materials on fine spatial meshes [15–21]. Finally, many HPC system-specific optimizations have been explored in order to reduce the run-times associated with tally updates, focused

on reducing bottlenecks that arise when profiling tally routines for specific benchmark geometries and processor architectures [22].

With the shift in HPC system towards GPGPU computing, this has also pushed the reactor physics community towards overhauling the history-based Monte Carlo transport algorithm entirely in favor of a vectorized approach. Event-based Monte Carlo groups source particles based on the collision events in order to process like events on vector units [23, 24]. Given that history-based Monte Carlo is better suited for CPU-based systems and event-based Monte is better suited on the GPU, this can require two versions of the same Monte Carlo application to be maintained in order to support computation on both types of platforms [24]. Moreover, GPU's tend to be limited on available on-node memory, so this typically limits the maximum generation sizes that are achievable per GPU [24].

1.4.2 Acceleration Methods

With these efforts in improving parallel efficiency in mind, another way to reduce overall time to solution is to reduce the total number of neutron histories required to reach target uncertainty levels. The total number of particles needed in a Monte Carlo simulation is dependent on two factors. Firstly, a stationary fission source must be generated from an initial starting guess in order to remove tally bias in the initial generations of the simulation, since this fission source distribution is unknown a priori. This stage is known as the inactive generations, where neutron histories are discarded and do not contribute to accumulated tally results. A significant number of power iterations may be required to reach this stationary distribution for reactor geometries due to their high dominance ratios. While a variety of source acceleration schemes have been proposed to reduce the computational requirements during this part of the simulation, CMFD - first proposed in 1983 to accelerate the convergence of nodal diffusion methods - will be the focus of this thesis [25]. The specifics of the various source acceleration methods applied to Monte Carlo will be examined more closely in later chapters.

The second stage of Monte Carlo simulations is referred to as the active generations,

and this is where the tallies that are computed from a stationary fission source are accumulated over a large number of generations in order to reduce statistical uncertainties in each tally region. While ideal convergence rates in the active generations behave as $1/\sqrt{N}$ with respect to the total number of particle histories, the presence of high levels of correlation between successive generations of particles can significantly degrade convergence rates [11, 26, 27]. The correlation stems from daughter neutrons in one generation of particles being linked directly to the parent neutrons from a previous generation. Multiple independent simulations can be used to de-correlate particles across independent seeds, but will continue to be correlated over successive generations as the number of active generations simulated increases [28]. Here, this thesis will employ the term *ensembles* to describe independent simulations that differ solely by the pseudo-random seed used to begin a Monte Carlo simulation.

1.4.3 True Variance Calculations

The computational overhead associated with correlated neutron generations raises the subject of another important component to accurate Monte Carlo modeling - real variance estimates. In the presence of strongly correlated tally regions, convergence rates in tally uncertainties will reduce at a rate slower than $1/\sqrt{N}$. However, variance calculations that neglect these correlation effects yield estimates lower than the true values, resulting in biased confidence intervals for tally results. Kelly et al. observed this phenomenon when simulating the NEA benchmark and use the batching technique to obtain true variance estimates [11]. However, batching reduces the number of available samples to make variance estimates and can yield accurate but imprecise variance estimates [28].

Since then, the autoregressive (AR) model has been proposed to predict the underestimation of variance in Monte Carlo calculations but its scope is limited to 1-D problems where the eigenvalue and eigenfunctions of higher-order modes in the fission source distribution can be calculated analytically [29]. The simulation code MCCARD uses a real variance estimation method based on a stochastic propagation model that estimates the inter-cycle covariance of the fission source distribution [30, 31]. Miao designs

a rigorous method for quantifying the level of correlation present in a Monte Carlo simulation in order to accurately report variance estimates, and further goes on to develop a run strategy to reduce correlations between generations through the Delayed Neutron method [32–34]. However, the complexity associated with this method has limited its widespread use in Monte Carlo solvers.

Once again, independent ensembles of Monte Carlo runs can be used to report accurate variance levels, since these ensembles are uncorrelated from each other and thus no adjustment in variance calculations need to be made for such run strategies. However, the use of independent ensembles requires that each ensemble be evolved independently and therefore places a higher burden on the inactive generations. This ultimately increases the number of discarded particle histories and can reduce the active neutron histories available to calculate variance levels compared to single ensemble strategies. For example, if an ensemble of 10 million neutrons requires 10 generations to reach stationarity, then a simulation running ten independent ensembles with 10 million neutrons per ensemble will discard 10 ensembles $\times$ 10 inactives $\times$ 10 million histories $=$ 1 billion neutron histories before active generations can begin. By contrast, a single ensemble will require 100 million histories to reach stationarity and therefore the remaining 900 million histories can be dedicated towards simulating the active generations with correlated convergence rates. For typical simulations that require many more inactive generations to reach stationarity, the total number of discarded neutron histories in a multi-ensemble approach is prohibitively too high to be able to reduce total simulated neutron histories while providing accurate variance measurements. Thus, a single ensemble run with very large number of active generations and a batching technique continues to be the preferred approach to running Monte Carlo simulations with adequately low and accurate variance estimates.

1.4.4 Coupling with Nonlinear Solvers

With the advent of powerful supercomputers to dedicate towards high-fidelity simulations, recent trends in the nuclear community have focused on nonlinear coupling

schemes between the most accurate methods for multiphysics coupling. For example, significant efforts between ANL, ORNL, and MIT have been made on the front of coupled Monte Carlo for neutron transport with high-fidelity Computational Fluid Dynamics (CFD) for TH calculations on an Small Modular Reactor (SMR) geometry [35]. Eventually, depletion analysis for this problem will also be included that will significantly add to the nonlinearity of this tightly coupled problem [36]. Such high precision modeling was previously unimaginable due to the sheer computational cost associated with each physics solver involved in the coupled simulation workflow.

Recent investigations into coupling schemes show, however, that a direct coupling between fully converged high-fidelity physics solvers is not the most efficient way to generate accurate high-fidelity solutions to feedback problems. Instead, a single-batch approach to Monte Carlo-TH coupling is shown to increase stability in the steady-state solution while reducing time spent in costly Monte Carlo power iterations [37]. In such a scheme, temperature updates occur between unconverged (i.e., stationarity not being met) generations of Monte Carlo so that power levels change gradually between TH updates instead of potentially creating instabilities by updating temperature only once a fully converged Monte Carlo solution is generated.

Such a coupling scheme illustrates the importance of using low-order methods to inexpensively pass solution fields back and forth between coupled solvers to generate an approximate solution before ultimately refining the solution using higher-fidelity methods. Otherwise, a direct high-fidelity coupling scheme is more susceptible to instabilities that can be dampened through under-relaxation, but slows down convergence rates between solvers that already possess significant computational costs [38]. A recent extension of this idea comes from Harper in his Ph.D. dissertation where he tightly couples a Monte Carlo-based low-order CMFD solution with a subchannel code using tally derivatives to inexpensively generate an approximately converged TH coupling scheme that is later refined by high-fidelity Monte Carlo [39]. In such a simulation scheme, CMFD acts as the medium for efficient coupling between low-order transport and TH solvers, which is then refined by single Monte Carlo iterations to add higher fidelity to the solution. By doing so, Harper is able to generate stationary power distribu-

tions in roughly fifteen generations for a quarter-core Pressurized Water Reactor (PWR) benchmark simulation with temperature feedback. Johnson devises a similar approach to Monte Carlo simulations coupled with depletion solvers, where reduced order methods are used to evolve reaction rates on fine time steps efficiently and inform high-fidelity Monte Carlo updates on coarser time intervals [40].

1.5 Objectives of Thesis

In this thesis, the issues surrounding Monte Carlo parallelization, acceleration methods, real variance calculations, and nonlinear coupling will be explored in much detail. The development of acceleration methods and parallelization of Monte Carlo methods on HPC systems have occurred relatively independently of each other, and this work will serve to link these concepts together to propose a novel simulation strategy that can be used to run Monte Carlo efficiently on large-scale computing systems. This method, coined *ensemble averaged CMFD* begins with the CMFD acceleration method for Monte Carlo as its foundation and uses an ensemble averaged approach to generate reliable tally statistics while minimizing overall runtime. Here, *ensemble averaging* refers to the use of running multiple ensembles to generate mean and variance levels in the underlying tally data across independent ensembles.

Before discussing the specifics of the ensemble averaged CMFD approach, this thesis begins by investigating accurate stationarity diagnostics in Monte Carlo and also conducting a rigorous analysis of the exact performance benefits from using CMFD compared to a non-accelerated Monte Carlo simulation. In order to accurately distinguish performance benefits between inactive and active generations, a robust stationarity diagnostic is needed in order to pinpoint when a stationary fission source has been achieved and active generations can commence. Likewise, in order to make a fair comparison between accelerated and non-accelerated simulations, an in depth investigation of CMFD acceleration is carried out in order to identify the exact parameters that contribute to the performance benefits from acceleration methods. Figures of merit for such studies not only include total number of generations to stationarity, but will also look at variance

levels in the instantaneous fission source as well as the overall runtime on multi-core computing systems.

On the subject of nonlinear coupling, this thesis will make a broad case for how ensemble averaging can be applied to nonlinear multiphysics coupling schemes as well, but specific numerical experiments will not be carried out. Ultimately, ensemble averaged techniques can be applied to a variety of nonlinear solvers, and this thesis focuses on CMFD as a surrogate for such schemes since it is tightly coupled to the Monte Carlo routine and is relatively inexpensive to run. Similarly, the arguments for this work will focus on CPU-based computer architectures due to availability, but ensemble averaging has applications on GPU-based machines as well. The specifics of these applications towards general nonlinear coupling schemes and GPU systems will be presented in the concluding section of this thesis.

With this in mind, the following are regarded as major objectives of this thesis:

- Develop a heuristic and provide a framework to correctly diagnose fission source stationarity in order to distinguish between inactive and active generations.
- Characterize the key parameters affecting the performance of CMFD acceleration, based on generations to stationarity, variance levels in the instantaneous fission source, and overall runtime.
- Develop and evaluate the efficacy of an ensemble averaged CMFD approach on HPC systems and compare its performance to a variety of Monte Carlo run strategies.

This thesis will heavily utilize the open source Monte Carlo particle transport code OpenMC to carry out numerical experiments. A detailed description of the code's implementation can be found in Romano's PhD dissertation and accompanying journal articles [41–43]. Furthermore, the following definitions are included below to disambiguate terms that relate to Monte Carlo methods that will commonly occur in this thesis:

- *Particle history*: The simulation of an individual neutron from life to death through a reactor, as well as any secondary particles created. Each collision in a particle's

histories can contribute to the accumulation of tallied quantities.

- *Generation*: A collection of independent neutron histories grouped together to solve as a single power iteration
- *Generation size*: Number of particles in a generation
- *Batch*: A grouping of multiple generations into a single tally realization in order to reduce correlation effects between successive batches
- *Batch size*: Number of generations in a batch. A large generation size implies a larger generation lag between batches and can be used to measure real variance according to [11].
- *Ensemble*: An independent run of Monte Carlo with some fixed generation size but a different pseudorandom starting seed. Multiple ensembles are uncorrelated from each other and thus can accurately measure real variance estimates.
- *Inactive generation*: Initial generations in a Monte Carlo simulation where bias in the initial fission source distribution guess is removed before eventually reaching stationary levels
- *Active generation*: Latter generations in a Monte Carlo simulation after which fission initial source distribution bias and tallied quantities can begin to be accumulated
- *Stationarity*: The point at which the instantaneous fission source has reached a stable level of fluctuation and active generations can commence. Since the evolution of the fission source from inactive to active generations is a stochastic process, the exact point at which stationarity occurs is not precisely defined.
- *Fission bank*: The spatial locations of fission sites that occur after the simulation of a single generation of neutron histories
- *Source bank*: Initial location of source sites at the beginning of a generation. Typically in an eigenvalue simulation, the source bank is sampled from the fission

bank of a previous generation in order to retain the same number of neutron histories per generation.

- *Instantaneous fission source*: Nu-fission reaction rates at each mesh and energy group for a single generation

- *Accumulated fission source*: The average of instantaneous fission source from the first active generation to a given active generation.

1.6 Thesis Outline

This thesis is segmented into three parts. Part I consists of this introductory chapter.

Part II discusses the development of reliable stationarity diagnostics for reactor-type Monte Carlo eigenvalue simulations and explores a myriad of Monte Carlo run strategies to evaluate the holistic benefits of CMFD on the grounds of overall runtime as well as fission source variance levels. Chapter 2 introduces the framework for using Functional Expansion Tallies (FET's) to assess stationarity in the fission source. Chapter 3 discusses how this stationarity diagnostic can be applied to simulations that use source acceleration schemes such as CMFD and delineates the key parameters that contribute to the efficacy of CMFD to determine optimal run strategies for CMFD-based Monte Carlo simulations.

Part III discusses the novel ensemble averaged CMFD approach and how it can be optimized to run efficiently on HPC architectures. Chapter 4 motivates the need for ensemble averaging in order to obtain accurate real variance estimates, and outlines the different run strategies that will be evaluated on much larger computing machines. One of the biggest draws of the ensemble-averaged CMFD approach is that it lends itself to a client-server like communication model that improves load balancing through asynchronous data communication. The implementation of this asynchronous ensemble averaged CMFD approach will also be examined in Chapter 4. Finally in Chapter 5, the asynchronous ensemble averaged CMFD approach will be applied to the final test case - a three-dimensional full-core benchmark geometry.

Part IV summarizes the progress made in this thesis for ensemble averaging techniques

in Chapter 6.

Part II

Optimal Run Strategies Using Traditional Monte Carlo Methods

Chapter 2

Stationarity Diagnostics

As discussed in Chapter 1, Monte Carlo simulations are divided into two stages - the inactive and active generations. This is needed since the converged source distribution is not known a priori so during the inactive generations, an initial source distribution is iterated on until it reaches stochastic convergence. At this point, the source distribution is said to have reached stationarity and the tallying process can begin, where quantities of interest such as power distribution and reaction rates are computed. The difficulty with this approach, however, is that an adequate transition point between inactive and active generations is difficult to determine, since the underlying source distribution will always include statistical fluctuations. What may seem like a stationary source distribution could be a slowly converging process with small statistical variations, while a source distribution that looks unconverged due to large variations could actually be stationary relative to the inherent noise in the system. In any case, determining a stationarity point is an essential part of the Monte Carlo algorithm for eigenvalue problems. Start the active generations too quickly and tallied results will bear significant bias from an unconverged source distribution. On the other hand, if too many inactive generations are simulated, computation time is wasted since the tallying process is delayed and each additional generation no longer contributes to making the final tallied source distribution any more accurate. Once source acceleration methods are employed to reduce runtime in the inactive generations, a consistent metric for determining this cutoff point becomes necessary as well.

In order to address these issues, a new stationarity diagnostic is introduced in this chapter that is meshless, determines stationarity on-the-fly, and characterizes how converged the fission source is based on the number of neutrons in the system. This chapter begins with an overview of commonly used metrics to diagnose stationarity, presented in Section 2.1. The newly proposed stationarity diagnostic is based on FET's, which are explained in Section 2.2. Section 2.3 describes the methodology that will be used to determine stationarity, and the results will be tested on 1-D and 2-D reactor problems in Sections 2.4 and 2.5 respectively, where this newly-developed stationarity diagnostic will be shown to outperform more traditional methods for assessing fission source stationarity. Finally, Section 2.6 discusses the specific implementation details of the FET-based stationarity diagnostic in the OpenMC Monte Carlo particle transport code in order to illustrate how this methodology carries negligible overhead to the overall runtime. The material in this chapter largely follows from a paper published in Annals of Nuclear Energy [44].

2.1 Background on Stationarity Metrics

Much work has been carried out in the domain of stationarity diagnostics to characterize when exactly to transition from inactive to active generations [45–49]. Shannon entropy is perhaps the most commonly used metric, and gained wide acceptance once it was understood broadly that simply monitoring the value of k_{eff} was insufficient in trying to determine a stationary source distribution [45–49]. Shannon entropy relies on the use of a spatial mesh for binning fission source sites, and is defined as,

$$S(n) = -\sum_{m} p_m(n) \log_2(p_m(n)) \tag{2.1}$$

where n is the generation number, m is the mesh index, and $p_m(n)$ is the fraction of neutron source sites found in the m-th mesh cell in generation n. Once this scalar value has stopped changing within statistical bounds, the fission source is said to have reached stationarity, and active generations can commence.

While Shannon entropy is a straightforward value to compute within a Monte Carlo solver, it has been shown to lead to instances of false convergence, especially for loosely-coupled problems with high dominance ratios [50, 51]. Furthermore, the value of Shannon entropy is dependent on the kind of spatial mesh used for the problem, making it difficult to generate reference values to compare to when trying to assess stationarity. Reference Shannon entropy values change based on how refined the spatial mesh being used is, and as a result different prescriptions can be made as to when to begin active generations. These drawbacks will be explored more closely in Section 2.3, where stationarity behavior based on Shannon entropy will be compared to the behavior of a newly-devised diagnostic.

The limitations with Shannon entropy stem largely from the fact that a piecewise constant function is used to characterize the fission source distribution. The fractional neutron population in each mesh cell is summed over the entire domain, and while the entropy value can represent fluctuations in the fission source distribution over the entire domain of interest, it is likely to neglect any local changes that occur over a finer region within the core. Moreover, Shannon entropy values are dependent on the transport of neutrons from cell to cell, and the slow migration of fission source sites from one region of the reactor to another can yield similar results for Shannon entropy despite an unconverged fission source. Thus, more recent methods have started to look at variations in higher order shape functions of the underlying fission source distribution. Nowak et al. propose two additional metrics to supplement the use of the entropy function to detect local fluctuations due to neutron clustering [52]. The first is based on examining higher order moments of Shannon entropy to ensure that these moments of the underlying fission source distribution have also reached stationarity, while the second looks at the center of mass of the neutron population to make sure that it does not fluctuate in the presence of strong neutron clustering. While these metrics are clearly linked in systems with considerable undersampling, this work does not identify a clear criterion to establish how much fluctuations in these metrics can be tolerated in the context of systems that appear to have converged fission sources. This is especially true for more complicated, full-core geometries, where Monte Carlo simulations with a

seemingly sufficient number of particles per generation exhibit first moment entropy values that don't ever seem to fully converge [1].

Wenner expands on the center-of-mass-based diagnostic by devising a rigorous set of statistical tests to definitively pinpoint when the fission source has reached stationarity [53]. Since a neutron population's center of mass will always be fluctuating around the problem center for symmetric problems with symmetric starting sources, Ueki introduces a stationarity indicator based on analyzing geometric centers of nine different locations via the Wilcoxon rank sum to better characterize fluctuations occurring within the fission source [54]. Shim and Kim also devise two stopping criteria for determining stationarity based on the "intergeneration correlation length", the minimum number of generations between two uncorrelated fission source distributions [55]. The stopping criteria rely on statistical analysis of the covariance matrix of stochastic error components between two stationary cycles, and assume that a generation length between two completely uncorrelated source distributions exists and can be estimated. However, for high dominance ratio problems, this correlation can be persistent for hundreds of generations, and it is unclear whether the estimation techniques for this generation length are applicable to a wide range of problems.

As a whole, the stationarity metrics presented thus far that have been applied to nuclear reactors have primarily focused on test problems with symmetric geometries and symmetric initial source guesses. However, more realistic problems are composed of systems with strong asymmetries, which these methods have not been tested on rigorously. Moreover, while these diagnostics provide prescriptions for how many generations should be simulated in the inactive generations to reach stationarity, they do not characterize the accuracy of the underlying fission source distribution itself. This accuracy is largely a function of the number neutrons simulated per generation, but most analysis is done with a fixed number of neutrons for each test problem, where it is assumed that the same number of inactive generations should be simulated irrespective of the number of neutrons per generation. Instead, stationarity metrics should account for the global system-level of variance inherent within the simulation in order to determine generations to stationarity.

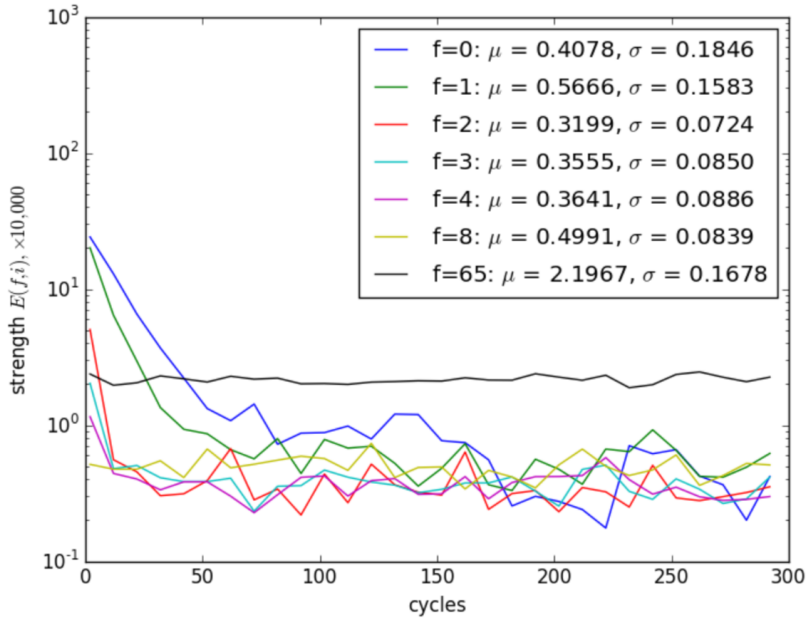


Figure 2-1: Monte Carlo errors decomposed by DFT frequency for 2D BEAVRS problem with 2.5 million neutrons per generation [1].

Spectral analysis of the fission source brings insight into this notion of system-level variance that is dictated by the number of particles in the problem [1]. Li decomposes the error in fission source into their spatial frequencies using the Discrete Fourier Transform, and the plots of these error frequency strengths are given in Figures 2-1 and 2-2 for the 2D BEAVRS problem with 2.5 million and 10 million neutrons per generation, respectively. These graphs illustrate how low-order frequency modes require source iterations to remove initial bias, while high-order frequencies (such as the order 65 mode) contribute to a nearly constant level of error due to statistical fluctuations. Additionally, the level of these statistical fluctuations decreases as the number of neutrons per generation increases from 2.5 million to 10 million. In this case, it can be argued that stationarity occurs when the low order modes reach an asymptotic value, while the high-order modes represent the system-wide statistical fluctuations. While this analysis provides insight into the evolution of fission source error, such calculations can only be conducted with a reference fission source distribution that can typically only be generated by running a number of computationally expensive simulations with large number of particles per generations and multiple independent seeds.

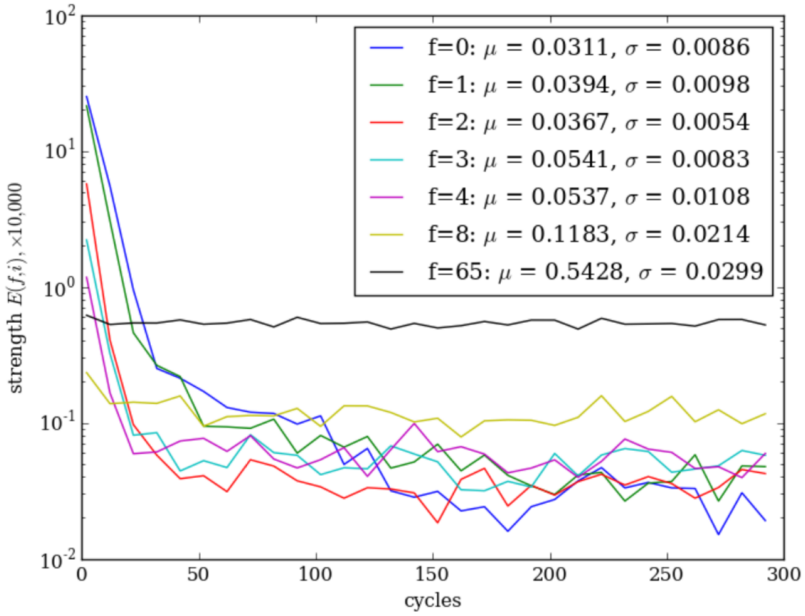


Figure 2-2: Monte Carlo errors decomposed by DFT frequency for 2D BEAVRS problem with 10 million neutrons per generation [1].

Shannon entropy is commonly used as a stationarity diagnostic but can sometimes lead to false stationarity. More recently, proposed metrics for reactor analysis have only been tested on symmetric problems with symmetric initial source guesses. Existing stationarity metrics disregard the relationship between generation size and generations to stationarity.

2.2 Background on Functional Expansion Tallies

FET's allow for the continuous representation of tally distributions via Monte Carlo simulations using a finite set of expansion coefficients, and have found applications in reactor analysis to reconstruct spatial distributions of reactor power, flux, and reaction rates [56–58]. Due to this efficient representation of continuous functions, FET's have also been implemented in multiphysics coupling algorithms where continuous feedback fields are passed between solvers that utilize dissimilar representations of the underlying geometry [59]. Additionally, the use of expansion coefficients has been explored as a

tool for predicting true variance levels of Monte Carlo tallies in the presence of inter-generational correlation [60].

As an example, FET's can be used to represent a spatially varying function, $f(\vec{x})$ as a linear combination of orthogonal basis functions over domain Γ:

$$f(\vec{x}) = \sum_{n=0}^{\infty} \overline{a_n} k_n \psi_n(\vec{x}) \tag{2.2}$$

$$\overline{a_n} = \langle f, \psi_n \rangle = \int_{\Gamma} f(\vec{x}) \psi_n(\vec{x}) d\overline{x} \tag{2.3}$$

$$k_n = \frac{1}{\|\psi_n\|^2} \tag{2.4}$$

$$\|\psi_n\|^2 = \int_{\Gamma} \psi_n^2(\vec{x}) d\overline{x} \tag{2.5}$$

where ψ_n is an orthonormal basis function of order n, k_n is a normalization constant, and $\overline{a_n}$ is the expansion coefficient. It can be shown that an unbiased collision-based estimator for coefficient $\overline{a_n}$ can be defined as:

$$\hat{a}_n = \frac{1}{N_g} \sum_{i=1}^{N} \sum_{k=1}^{K_i} w_{i,k} \frac{\Sigma_x(\vec{\xi}_{i,k})}{\Sigma_t(\vec{\xi}_{i,k})} \psi_n(\vec{\xi}_{i,k}) \tag{2.6}$$

where

$\vec{\xi} \equiv$ Neutron phase space in position, direction, and energy

$N_g \equiv$ Total neutrons per generation

$K_i \equiv$ Total collisions for particle i

$w_{i,k} \equiv$ Particle i weight at collision k

$\Sigma_x \equiv$ Macroscopic cross section for reaction x

$\Sigma_t \equiv$ Total macroscopic cross section

Such a formulation allows for these expansion coefficients to be computed natively

within Monte Carlo tallying routines. This functionality has already been implemented in the OpenMC particle transport simulation code for a variety of spatially varying basis functions [42] and will be used for test cases presented here onward. With expansion coefficients $\hat{a}_n$ computed, $f(\vec{x})$ can be approximated by:

$$f(\vec{x}) \approx \sum_{n=0}^{N} \hat{a}_n k_n \psi_n(\vec{x}), \tag{2.7}$$

where N is the degree of the basis set expansion. The equations presented in this section will be revisited in Sections 2.4 and 2.5 when discussing the specific test problems being simulated.

The FET formalism allows for the continuous representation of a tallied distribution as a set of coefficients over a predefined basis set.

2.3 Methodology for FET-based Stationarity Diagnostic

The high-level idea behind an FET-based stationarity diagnostic is to compute expansion coefficients to approximate the fission source at each generation, and observe when they have all stabilized. Once these expansion coefficients have all maintain a stable value, then stationarity of the underlying fission source is considered to have occurred. The magnitude of expansion coefficient of order n represents how significant the order n moment of the basis function is in terms of explaining the shape of the underlying distribution being represented by FET's. Expansion orders that do not contribute to the distribution under question will decay off to zero, while those that do will eventually reach a stable value. Unlike the entropy function, higher order moments of the fission source distribution are all being represented by the magnitude of each FET, so a stationarity diagnostic based on all computed FET's will ensure all shape functions specified by the basis set have converged instead of just a single spatial metric as predicted by Shannon entropy. The orthogonal set used to model the underlying fission source serves as a representation of the eigenmodes of the system, and the closer the basis set is to

the true eigenmodes the more accurately that FET's can characterize the fission source. Thus, such an approach to determining stationarity is applicable to reactors designed to achieve flat power profiles, where a finite set of expansion coefficients can accurately represent the spatial distribution of the fission source.

Consider a one-dimensional, 400cm homogeneous slab of uranyl sulfate (material composition provided in Appendix A) in the x-direction with vacuum boundary conditions and a starting point source at the center. The fission source at each generation can be represented by the fission production rate, and the one-dimensional Legendre polynomial basis set of order N is used to approximate this distribution:

$$\nu\Sigma_f(x)\psi(x) \approx \sum_{n=0}^{N} \hat{a}_n P_n\left(\frac{2(x-x_0)}{x_1-x_0}-1\right) \tag{2.8}$$

Here, the fission source is characterized by the neutron production rate from fission $\nu\Sigma_f(x)\psi(x)$, and it is decomposed as the sum of N well-defined Legendre polynomials, P_n. The argument to the Legendre polynomial P_n represents the normalized position between -1 and 1, the domain for which Legendre polynomials are defined, where x_0 and x_1 are the lower and upper bounds of the problem domain. The coefficients $\hat{a}_n$ include the normalization constants k_n defined in Equation 2.2 and determine how significant each Legendre mode is in explaining the overall fission source. These coefficients are computed as part of the Monte Carlo tallying routine as:

$$\hat{a}_n = \frac{2n+1}{x_1-x_0}\int_{-1}^{1} dx P_n\left(\frac{2(x-x_0)}{x_1-x_0}-1\right)\nu\Sigma_f(x)\phi(x) \tag{2.9}$$

The first term, $(2n+1)/(x_1-x_0)$, is the normalization constant that allows each coefficient to be compared on the same scale, while the integral quantity can be computed as a Legendre expansion filter that scores the nu-fission reaction rate. When calculated in OpenMC for each generation, the magnitudes of the coefficients for Legendre moments of up to order 7 in the 400cm problem with 100,000 neutrons are shown in Figure 2-3. The corresponding Legendre basis functions up to order 7 are also provided in Figure 2-4.

From Figure 2-3, it can be seen that even-parity FET coefficients require initial

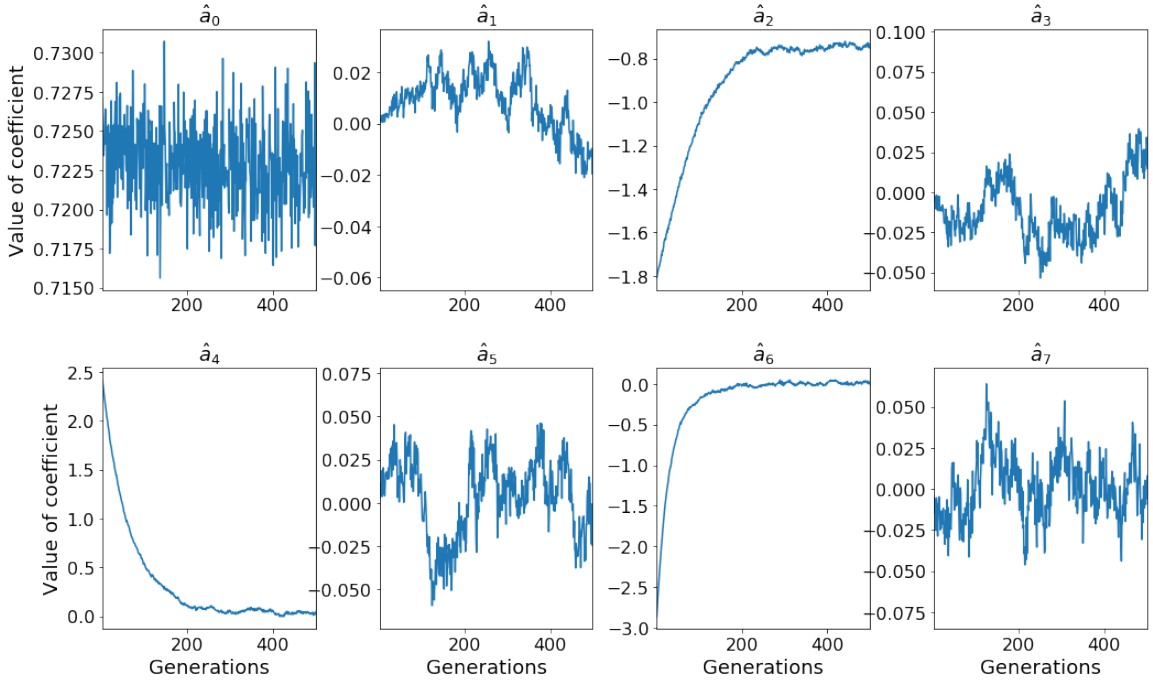


Figure 2-3: Value of FET coefficients $\hat{a}_0$ to $\hat{a}_7$ for 400cm homogeneous problem with 100,000 neutrons.

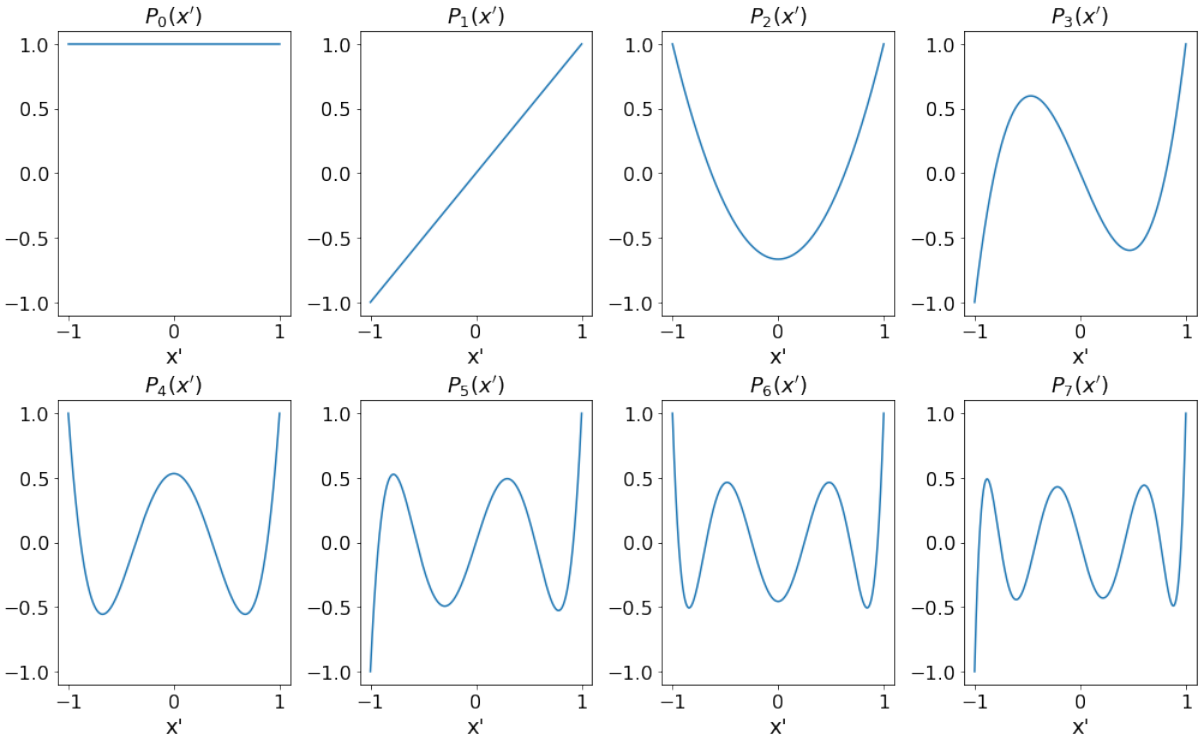


Figure 2-4: Legendre basis functions $P_n(x')$ for $n = 0$ to $n = 7$, where x' represents the normalized x position between -1 and 1.

generations to converge, beyond which point they fluctuate around a steady value. On the other hand, odd-parity coefficients immediately fluctuate considerably around zero. This is consistent with the idea that the magnitude of an FET coefficient represents the contribution of that particular basis function to the shape of the underlying fission source. Initially, high-order even parity modes will all contribute to the shape of a localized fission source that evolves from a point source, but as the fission source assumes its eventual cosine-shape, the contribution of these high-order even parity modes will all decay off, leaving only the zeroth-order and second-order modes as the dominant modes to explain the underlying shape of the fission source.

In this case, odd-parity moments reach a stable value very early on and do not need to be included as part of the convergence analysis to determine stationarity of the underlying fission source. To this end, a signal-to-noise ratio metric is used that is based on Griesheimer's work relating to fission source sampling methods from FET's [61]. This metric was used by Griesheimer to filter out modes that weren't contributing to the overall source when trying to approximate the source distribution from FET's. Similarly, this work uses the signal-to-noise ratio to exclude modes that are initially oscillating around a steady value from the convergence analysis. The ratio is defined as:

$$SNR = \frac{\mu_{\hat{a}_n}}{\sigma_{\hat{a}_n}} \tag{2.10}$$

where the mean and standard deviation are defined over an interval size of 10 generations. FET coefficients that exhibit $SNR << 1$ in the initial generations are deemed as "noisy" and are ignored when determining if all FET coefficients have reached stationarity. Thus, fission source stationarity is defined as the point when the slowest converging FET coefficient has reached stationarity. For the remainder of this paper, a cutoff of 1 is used to filter out highly oscillating FET coefficients.

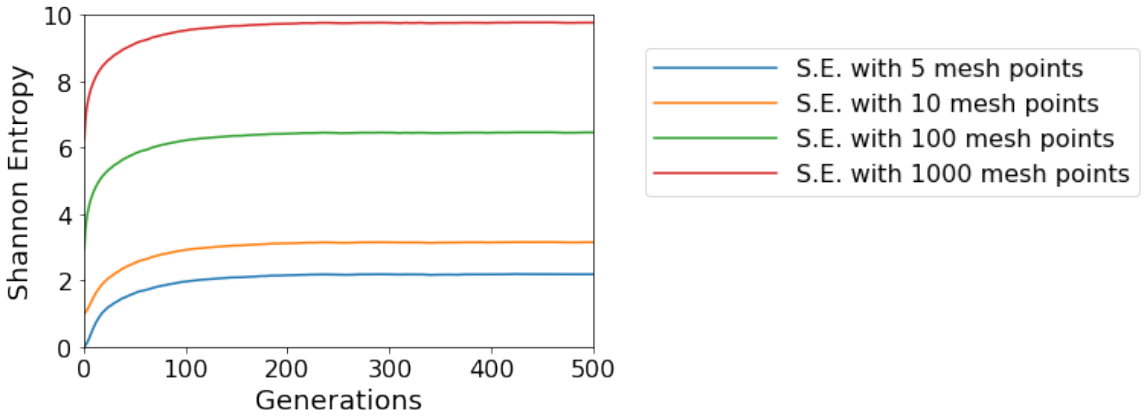


Figure 2-5: Value of Shannon entropy for 400cm homogeneous slab problem calculated using different number of mesh points.

2.3.1 Comparing FET-based Stationarity Diagnostic with Shannon Entropy

At this juncture, it is instructive to look at how Shannon entropy behaves for this problem to see how the FET-based stationarity diagnostic compares to traditional methods. As mentioned in Section 2.1, the value of Shannon entropy is dependent on the meshing scheme used to compute $p_m(n)$ in Equation 2.1. Figure 2-5 plots Shannon entropy for the one-dimensional 400cm slab problem with vacuum boundary conditions, computed using 5, 10, 100, and 1000 mesh points.

Based on the mesh discretization, it is clear that the value of Shannon entropy changes drastically. This can cause difficulty in determining the number of generations required to reach stationarity, since this can change based on the type of mesh used for the problem. MCNP prescribes a minimum of 20 source particles per mesh cell for accurate statistics, and can automatically resize the entropy grid to ensure that this condition is met [62].

Furthermore, the axis used to represent Shannon entropy can heavily influence how converged a problem looks. Figure 2-6 plots the Shannon entropy value using 100 mesh points for three different y-axis ranges, yielding very different outcomes for how stationary the entropy function looks in each case. The narrower the y-axis range used to represent Shannon entropy the more fluctuations within the entropy function begin to appear, resulting in less certainty that the fission source distribution is indeed converged.

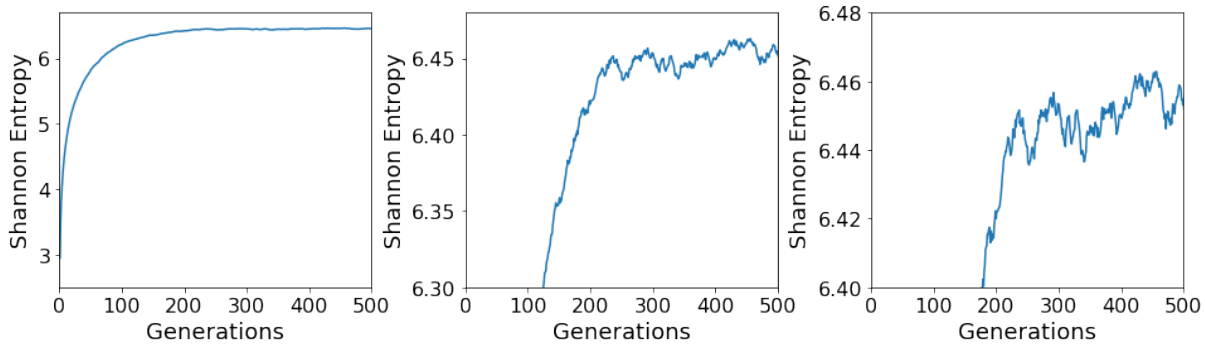


Figure 2-6: Value of Shannon entropy for 400cm homogeneous slab problem with 100 mesh points, represented with different y-axis ranges. As the range is narrowed, the noise within Shannon entropy becomes more apparent.

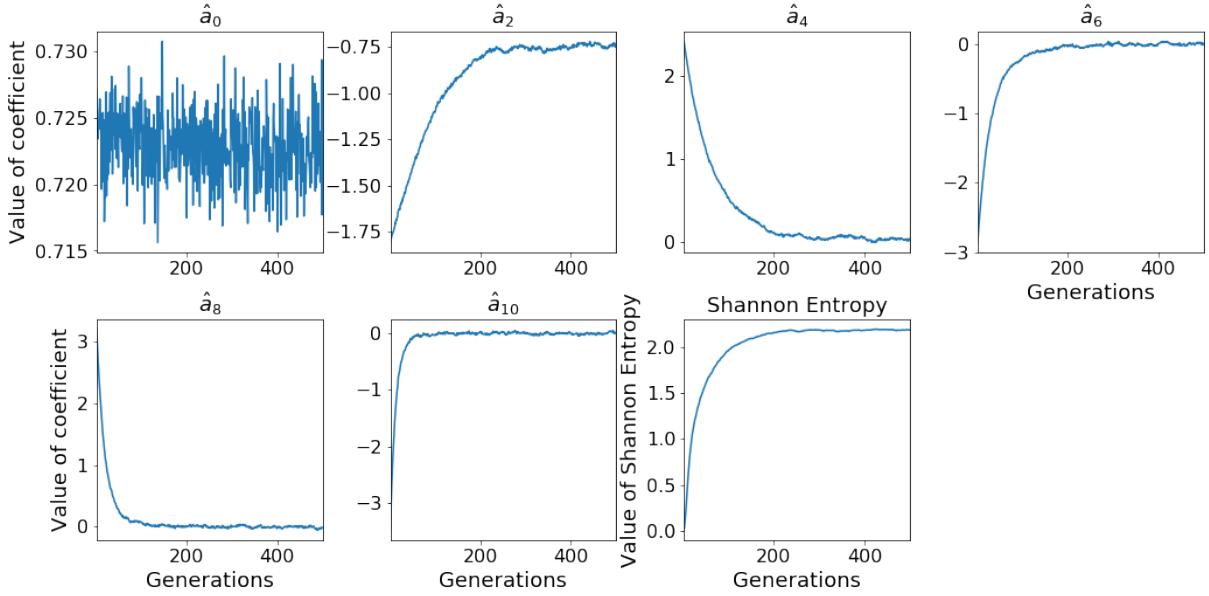


Figure 2-7: All even-parity FET coefficients up to order 10, as well as Shannon entropy as a function of number of generations.

The same is true for FET coefficients. Thus, a visual approach to determining stationarity should be avoided, and in Section 2.3.2, a statistical approach that can be used for both Shannon entropy and FET-based diagnostics is presented.

2.3.2 Defining a Stopping Criteria

Figure 2-7 shows the behavior of FET coefficients and Shannon entropy as a function of number of generations simulated. Only the even-parity FET coefficients are shown, since the odd ones do not have a significant signal-to-noise ratio, as defined by Equation 2.10.

Additionally, the value of Shannon entropy shown in Figure 2-7 is computed with 100 mesh points. While the a_0 coefficient seems to be highly oscillatory, it actually reaches stationarity very quickly and fluctuates around a stable mean. It can also be seen that the other FET coefficients resemble a similar shape as Shannon entropy, with different rates of reaching stationarity. In order to compare Shannon entropy and the FET-based diagnostic, a stopping criteria needs to be devised that indicates when the two methods claim stationarity has occurred.

The stopping criteria is based on seeing whether the mean of the diagnostic has changed between some offset number of generations within statistical limits. Let S_A be the interval size over which the sample mean is computed, and S_B be the offset number of generations between the sample means. At each generation g, a check is made to see if the sample mean at $g + S_B$ falls within two standard deviations of the sample mean at generation g. In order to minimize the chance of a false positive, this behavior needs to persist for a continued number of generations. A third parameter S_C is introduced, which represents the minimum number of continued generations of unchanging means needed to define stationarity. Thus, stationarity is said to occur at generation G if all of the conditions are met:

$$\begin{aligned}
(\mu - 2\sigma)_{[G-S_A:G]} &< \mu_{[G-S_A+S_B:G+S_B]} < (\mu + 2\sigma)_{[G-S_A:G]} \\
(\mu - 2\sigma)_{[G-S_A+1:G+1]} &< \mu_{[G-S_A+S_B+1:G+S_B+1]} < (\mu + 2\sigma)_{[G-S_A+1:G+1]} \\
(\mu - 2\sigma)_{[G-S_A+2:G+2]} &< \mu_{[G-S_A+S_B+2:G+S_B+2]} < (\mu + 2\sigma)_{[G-S_A+2:G+2]} \\
&\vdots \\
(\mu - 2\sigma)_{[G-S_A+S_C:G+S_C]} &< \mu_{[G-S_A+S_B+S_C:G+S_B+S_C]} < (\mu + 2\sigma)_{[G-S_A+S_C:G+S_C]}
\end{aligned} \tag{2.11}$$

In this case, $\mu_{A:B}$ and $\sigma_{A:B}$ correspond to the sample mean and standard deviation computed over the range from generations A to B. S_A, S_B, and S_C are tunable parameters that control how conservatively stationarity should be defined. For the purposes of this work, values of $S_A = 10, S_B = 30, S_C = 10$ are used as they work well for a wide range of problems. This means that mean and standard deviations are computed over an interval

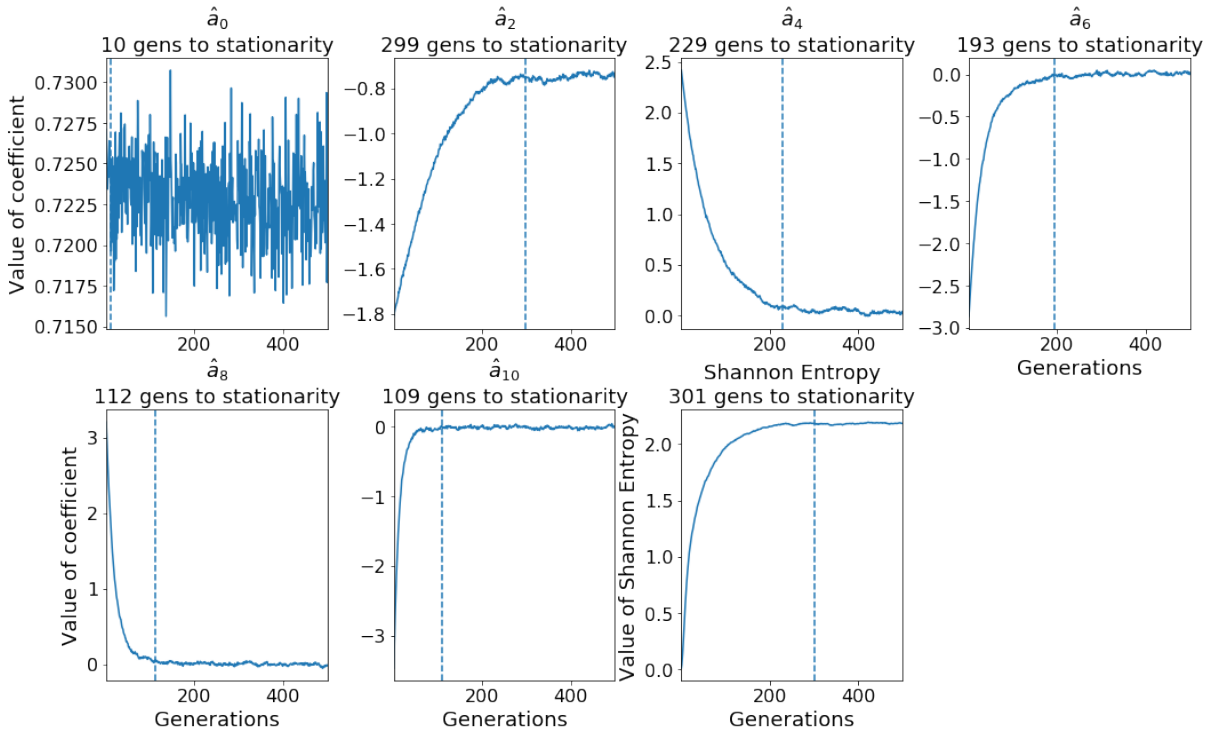


Figure 2-8: Even-parity FET modes and Shannon entropy, shown with corresponding stopping criteria.

of 10 generations, sample means are offset by 30 generations, and stationarity occurs when the offset mean falls within two standard deviations of the current sample mean for 10 consecutive generations.

Using this stopping criteria, the stationarity points for the FET coefficients and Shannon entropy values in Figure 2-7 are computed and shown in Figure 2-8 with a vertical line. This plot illustrates that Shannon entropy predicts stationarity to occur at generation 301 when 100 mesh points are used, while the FET-based diagnostic prescribes 299 generations to stationarity, showing close agreement between the two methods. Based on the stopping criteria presented in Equation 2.11, the stopping criteria varies between 290 generations to 311 generations when using 5, 10, 100, and 1000 mesh points to calculate Shannon entropy. Plotting the source distributions at these two points also shows that the source is cosine-shaped, as seen in Figure 2-9. Due to the relatively low neutron count for this case, there is a noticeable level of noise in the source distribution that will always persist regardless of the number of generations simulated.

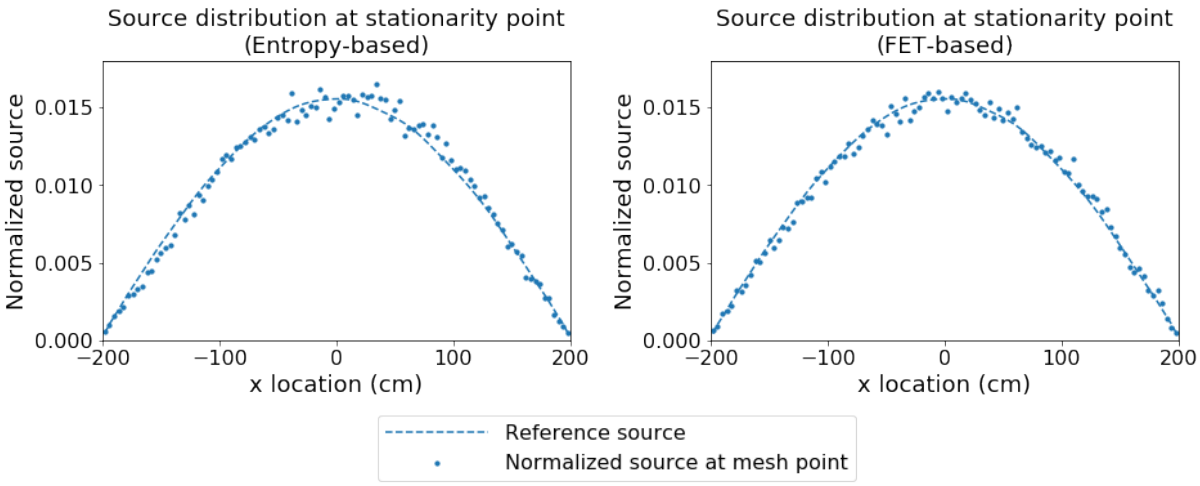


Figure 2-9: Source distributions at the stopping criteria generation, as predicted by the entropy-based metric (left) and the FET-based diagnostic (right). The reference distribution is also given by the dashed line.

2.3.3 Low-order vs. High-order Modes

Section 2.3.2 looks at the stationarity behavior of all coefficients up to order 10 to define stationarity. These are considered "low-order" modes for this problem that need to converge to some stable mode. While higher order coefficients could also be looked at, they typically converge much more rapidly than coefficients like a_2 or a_4, and so order 10 is chosen as the cutoff for low-order modes. The exact cutoff is not vital, as long as all of the slowest converging modes have been included in the stationarity analysis.

Looking at the behavior of higher order modes also yields some insight into the system-level fluctuations that are occurring. Figure 2-10 plots the behavior of a few higher-order modes. The basis functions corresponding to these coefficients are plotted in Figure 2-11. Similar to the odd-parity modes in Figure 2-3, these high-order modes are very noisy and all fluctuate around zero. Given how oscillatory the basis functions are at this level, it is unlikely that these polynomials would explain the underlying shape of the source distribution. Yet, these modes have a nonzero magnitude at certain generations, since they capture the deviations from the true cosine shape due to stochastic noise within the simulation. One important attribute of these high-order modes is that the average of a group of these modes will smoothen out and approach zero. Figure 2-12 plots the average value of FET coefficients from order 16 to order 30. After a few initial

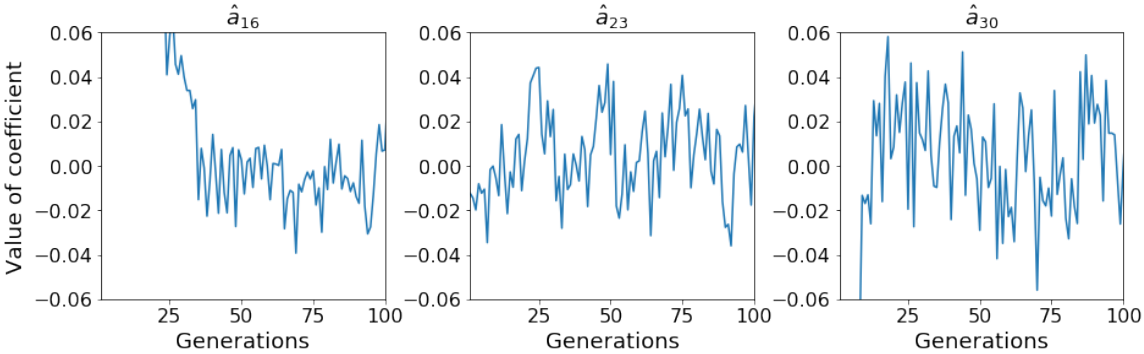


Figure 2-10: Value of FET coefficients for $\hat{a}_{16}(x')$, $\hat{a}_{23}(x')$, and $\hat{a}_{30}(x')$.

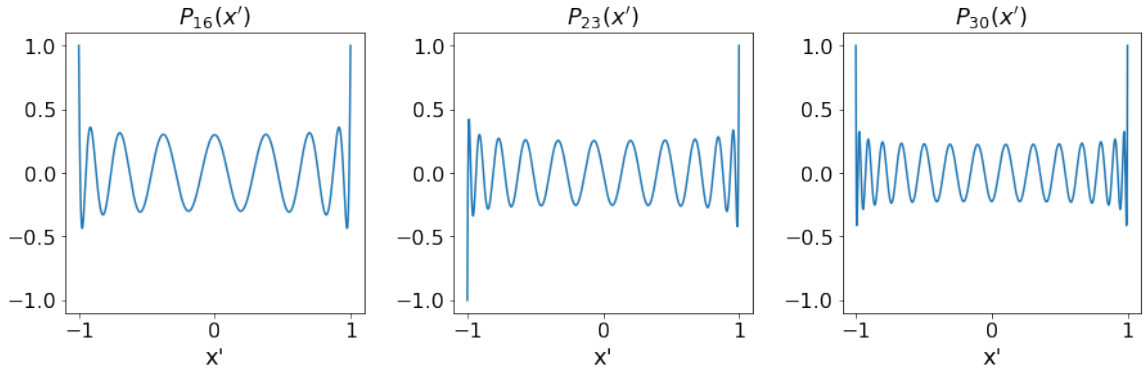


Figure 2-11: Legendre basis functions for $P_{16}(x')$, $P_{23}(x')$, and $P_{30}(x')$.

generations, a lot of the fluctuations dampen compared to the behavior of individual high-order coefficients. As will be explained in Section 2.4, the variance within this collection of high-order modes is largely a function of the number of neutrons in the system. A larger number of neutrons per generation will reduce the variance in the high-order modes indicating that a more precise solution is obtained.

The FET-based stationarity diagnostic monitors the stationarity of low-order FET coefficients and the slowest converging coefficient is the one that determines overall stationarity. A stopping criteria is devised based on the sample mean fluctuating within statistical limits for a persistent number of generations. This stopping criteria can be used to compare stationarity predictions between Shannon Entropy and FET-based methods. A signal-to-noise ratio is used to filter out FET modes that exhibit high noise levels.

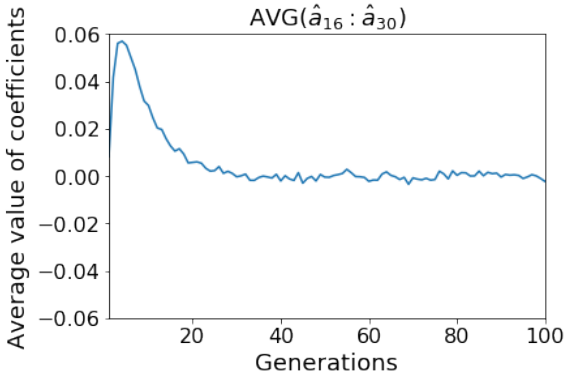


Figure 2-12: The averaged value of FET coefficients from $\hat{a}_{16}$ to $\hat{a}_{30}$, as a function of generations.

2.4 Results: One-Dimensional Test Problem

The 1-D results presented in Section 2.3 will be extended in this section to include how stationarity behavior changes as a function of problem size, position of starting source, and number of neutrons per generation. Section 2.3 looked at the results for a fixed 400cm problem, but this will now be varied between 200cm and 800cm in 200cm increments. Shannon entropy is calculated using a fixed number of 100 mesh points, since it was found that based on the stopping criteria in Equation 2.11 the results were relatively insensitive to the entropy mesh size. Additionally, the starting source was set as a point source at the center, and now the stationarity behavior will also be determined for problems where the starting source is 10cm away from the right boundary. An asymmetric starting source is analyzed to mimic the behavior of a problem with an asymmetric geometry, where now the odd-parity FET modes will become significant in explaining the underlying shape of the source distribution. Finally, the number of neutrons per generation will be varied from 10,000 to 10,000,000 to show how stationarity metrics are related to the number of particles in the system. The dominance ratio of the problem as a function of slab length is shown in Table 2.1. Dominance ratios were computed using CMFD [63].

Table 2.1: Dominance ratio of 1-D slab problem for different slab lengths.

Slab length (cm)	Dominance Ratio
200.0	0.9890
400.0	0.9971
600.0	0.9977
800.0	0.9987

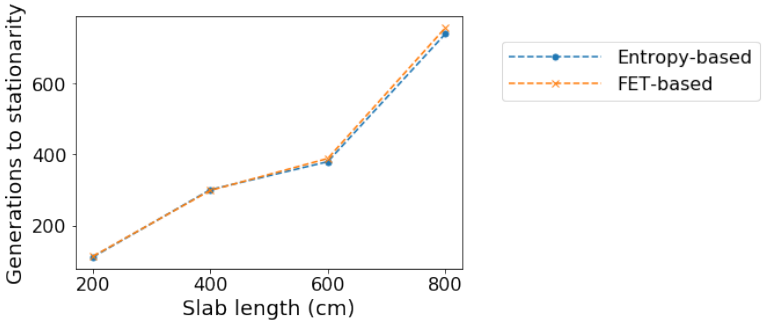


Figure 2-13: Generations to stationarity for FET-based and entropy-based approaches for different slab lengths at 100,000 neutrons per generation with initial point source at center of problem.

2.4.1 Stationarity Behavior as a Function of Problem Size

The number of generations to stationarity as a function of 1-D slab sizes is shown in Figure 2-13 for a generation size of 100,000 neutrons using both the FET-based and entropy-based approach. Both methods show close agreement, and as expected, a larger domain size requires more generations to stationarity due to the increased dominance ratio.

Figure 2-14 plots the Shannon entropy and slowest converging FET coefficient for each of the four domain sizes, along with a vertical line to identify when stationarity occurs for the stationarity metric. In all cases, the a_4 coefficient is the slowest converging FET coefficient, and the variance behavior of the a_4 coefficient, shown in Figure 2-15, also illustrates how more generations are required for stationarity as the slab length increases. Variance levels for each problem type reach a threshold level once enough generations are simulated, signaling the point at which the mean and variance of the underlying FET coefficient stop changing. The larger the problem, the slower the variance reduces, so more generations are required to reach these threshold values. In order to reduce the

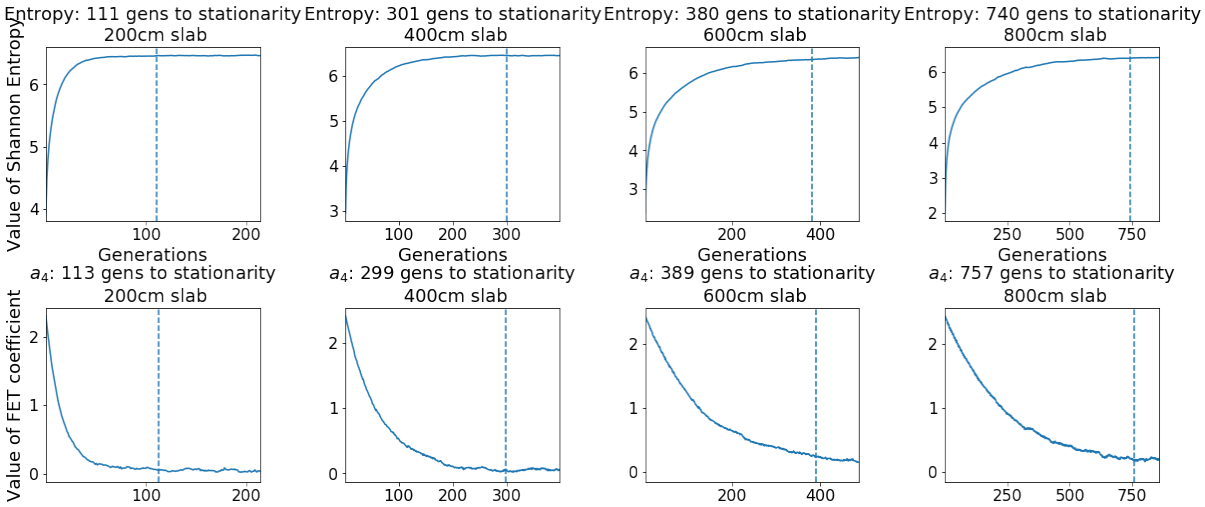


Figure 2-14: Value of Shannon entropy and slowest-converging FET coefficient as a function of slab length for a fixed generation size of 100,000 neutrons per generation. Values correspond to an initial point source located at the center.

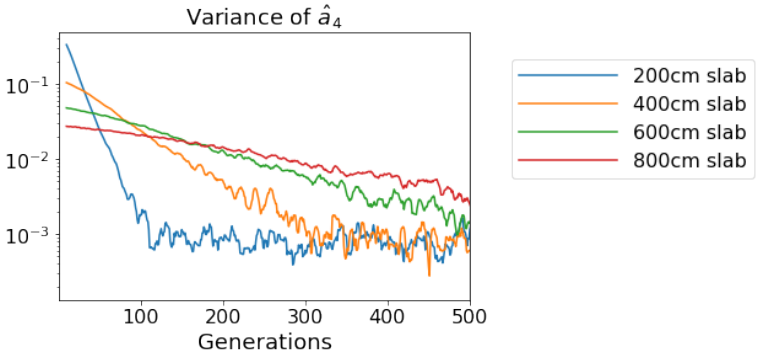


Figure 2-15: Variance of slowest converging FET coefficient, a_4, for different domain sizes, 10,000,000 neutrons per generation, and initial point source at center.

noisiness of the plots and clearly delineate the variance behavior between different slab lengths, Figure 2-15 uses a larger generation size of 10,000,000 neutrons per generation

While there is close agreement between the FET-based approach and entropy-based approach when the starting point source is at the center, large discrepancies start to arise when the initial source is offset from center. Best practices for criticality analysis mandate that starting fission sites be uniform across the entire geometry [49]. However, such an offset starting source distribution is presented to show the impact of a large asymmetric problem on source convergence. Instead of changing the underlying geometry, the starting source is modified as a highly offset point source to simulate the slow migration of fission source sites from one region of the reactor to another, illustrating another

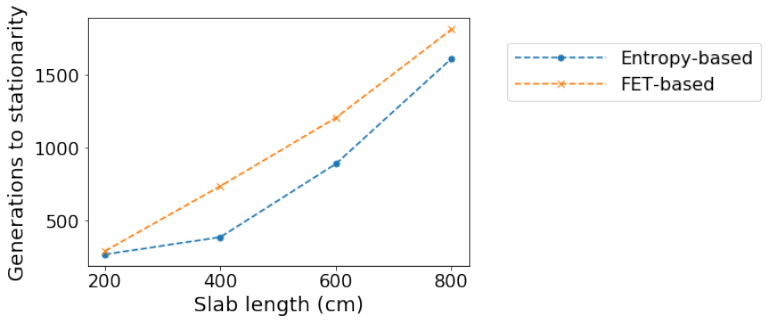


Figure 2-16: Generations to stationarity for FET-based and entropy-based approaches for different slab lengths at 100,000 neutrons per generation with initial point source offset 10cm away from right boundary of problem.

example problem where assessing stationarity is difficult. Figure 2-16 plots the number of generations to stationarity for a range of domain sizes, where now the initial point source is located 10cm away from the right boundary. These results indicate a larger number of generations is required for the FET-based method to reach stationarity.

In order to understand the behavior of these two methods and illustrate that the entropy-based method is indeed underpredicting the number of generations to stationarity, Figure 2-17 plots the Shannon entropy and slowest converging FET coefficient as a function of number of generations. Due to the asymmetry in the source distribution, the odd-parity FET modes become more significant, and now the a_1 coefficient takes the most number of generations to reach stationarity for all domain sizes.

Figure 2-17 shows that Shannon entropy flattens out much quicker than the a_1 coefficient for the same problem, and so the entropy-based approach predicts fewer generations to stationarity. This in turn leads to an incorrect source distribution that will lead to biased tallies in the active generations. Similar to Figure 2-9, Figure 2-18 shows the source distribution at the stationary generation as predicted by both the entropy-based and FET-based approaches for a 400cm problem with 100,000 neutrons and a starting point source 10cm away from the right boundary, indicating a clearly biased source distribution when using the entropy-based method.

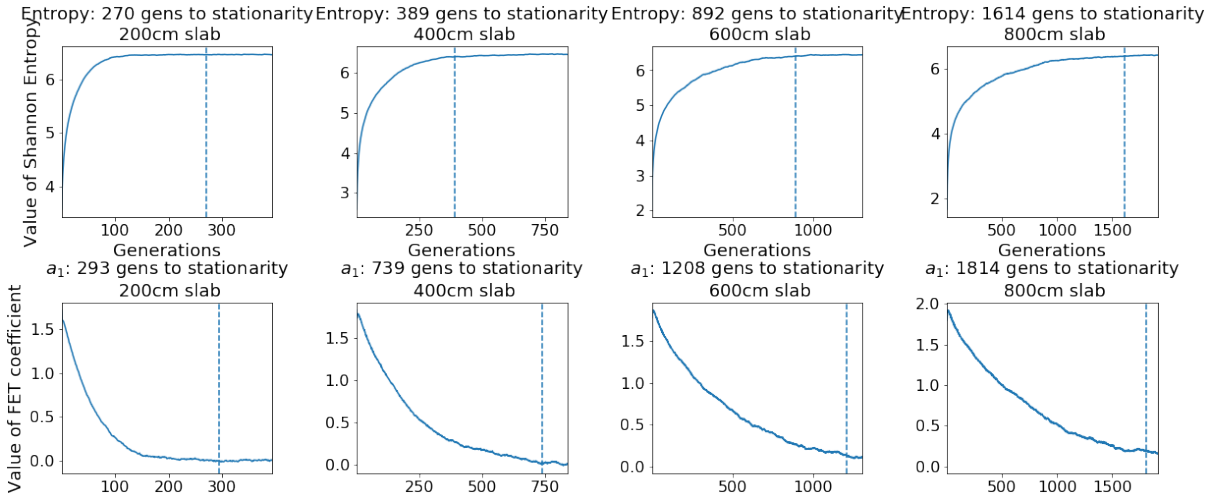


Figure 2-17: Value of Shannon entropy and slowest-converging FET coefficient as a function of slab length for a fixed generation size of 100,000 neutrons per generation. Values correspond to an initial point source located 10cm away from the right boundary of problem.

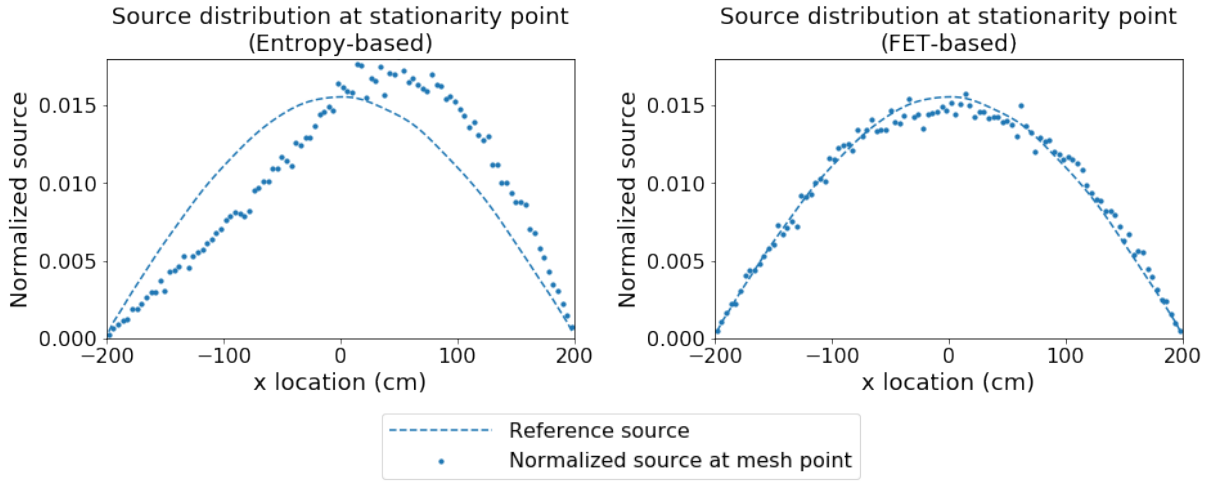


Figure 2-18: Source distributions at the stopping criteria generation, as predicted by the entropy-based metric (left) and the FET-based diagnostic (right) for a 400cm problem with 100,000 neutrons and a starting point source 10cm away from the right boundary. The reference distribution is also given by the dashed line.

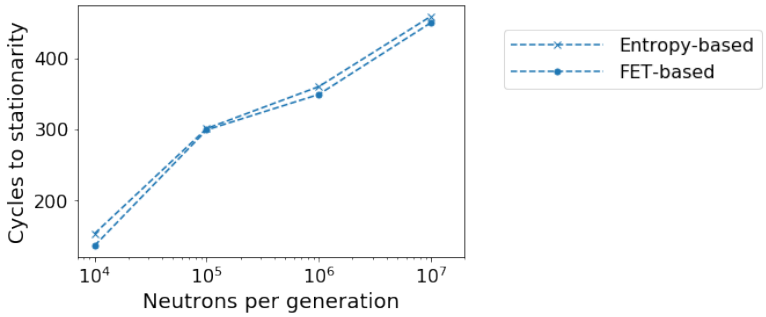


Figure 2-19: Generations to stationarity for FET-based and entropy-based approaches for varying generation sizes for 400cm slab with initial point source at center of problem.

2.4.2 Stationarity Behavior as a Function of Neutrons per Generation

Section 2.4.1 assumed a fixed neutron generation size and varied domain size, but this section will fix the domain size and vary the generation size to detail stationarity behavior as a function of neutrons per generation. Figure 2-19 plots the generations to stationarity for 10,000, 100,000, 1,000,000, and 10,000,000 neutrons per generation for a 400cm problem with an initial point source at the center, using the entropy-based and FET-based methods. For these four cases, the value of Shannon entropy and the slowest converging FET coefficient is plotted in Figure 2-20 as a function of generations, along with a vertical line to signify when stationarity occurs for each case. The a_4 FET coefficient was the slowest converging coefficient in all cases. These plots show close agreement between the two approaches, but also illustrate that a larger number of generations to stationarity is required when the number of neutrons per generation increases. Similar to Figure 2-15, the variance behavior of the low-order FET coefficients, shown in Figure 2-21 explains why this is so.

Figure 2-21 plots the variance behavior of a_2, a_4, a_6, and a_8 for a varying number of neutrons per generation. It is observed that the initial variance levels are comparable for each coefficient with similar convergence rates for each cases. As the number of neutrons per generation is increased, however, the threshold variance level also reduces. This asymptotic variance level represents the point at which the underlying FET coefficient has reached stationarity at a level comparable to the stochastic noise, beyond which

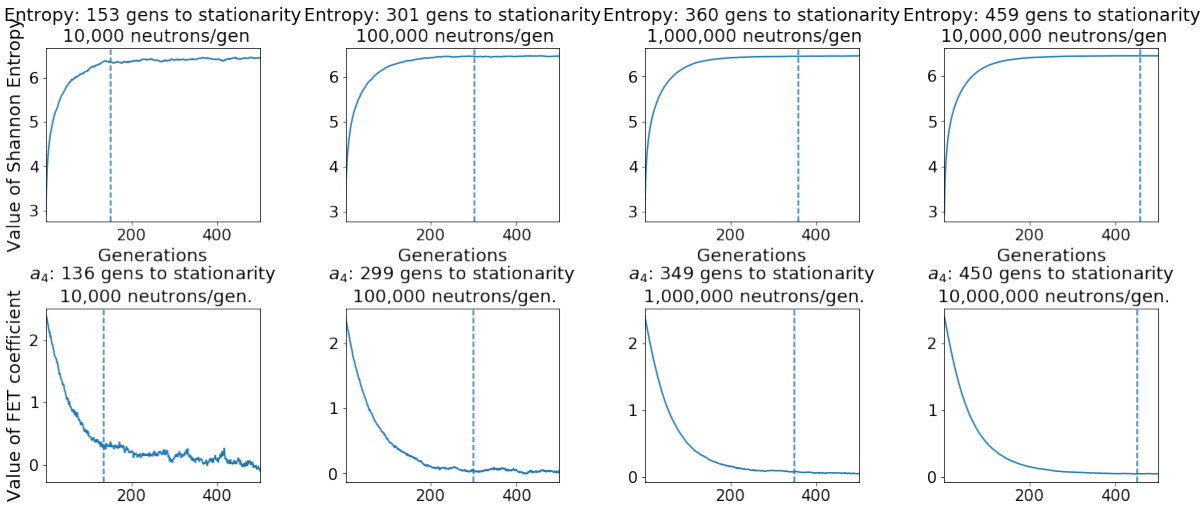


Figure 2-20: Value of Shannon entropy and slowest-converging FET coefficient as a function of generation size for a fixed slab length of 400cm. Values correspond to an initial point source located at center.

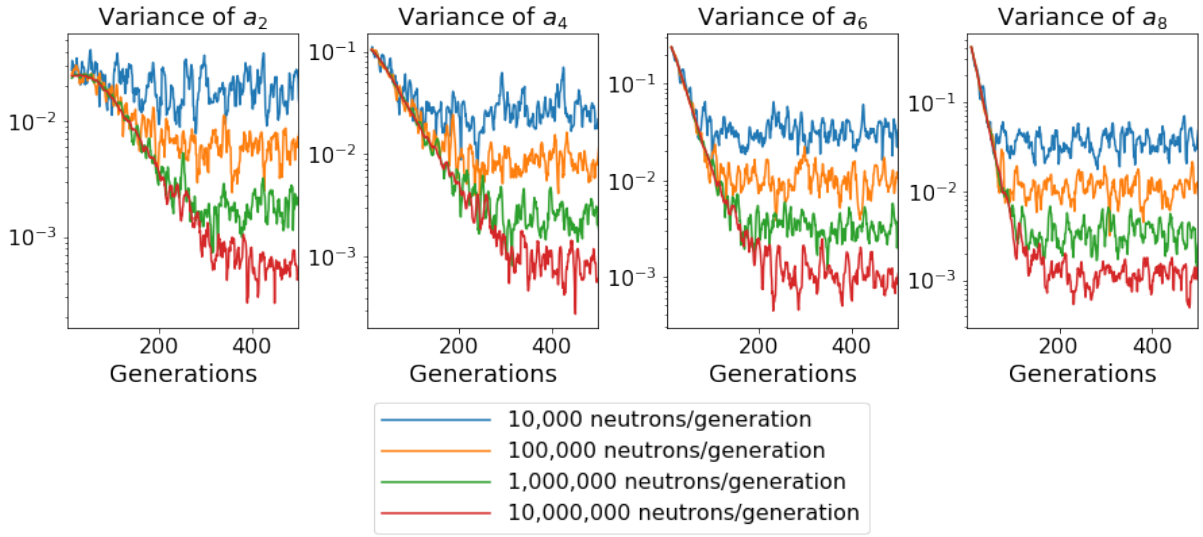


Figure 2-21: Variance of slowest converging FET coefficients a_2, a_4, a_6, and a_8, for different generation sizes and a fixed domain size of 400cm with an initial point source at center.

point the variance does not decrease with more generations simulated. Since a larger number of neutrons generation implies more precise FET coefficient values with less fluctuations, more generations are required to reach these lower asymptotic variance levels.

This also suggests that if the same number of generations to stationarity is used regardless of the number of neutrons per generations, then users are likely to carry the same fission source error level over to the active cycles, even though the simulation uses a larger number of particles that typically should yield more accurate solutions. This is analogous to a deterministic calculation where more source iterations are performed to converge the fission source error to lower values. In this context, the number of neutrons per generation dictates the minimum source residual that can be reached in a Monte Carlo simulation. In order to achieve lower variance levels at each generation, more neutrons must be simulated per generation for a longer number of generations.

The variance behavior of the high-order modes specified in Section 2.3.3 also points to similar observations. Figure 2-22 plots the average variance of FET coefficients from order 20 to 30, and similar to Figure 2-21, initial generations are required to reduce variance levels until a threshold value is reached that is related to the number of neutrons per generation. By aggregating the behavior of a collection of FET coefficients, much of the noise inherent within any single high-order coefficient is smoothened out. It can also be seen that the threshold variance level reduces by a factor of $\frac{1}{n}$, where n is the number of neutrons per generation. These high-order coefficients fluctuate around zero values because they do not explain the underlying cosine shape of the source distribution, but still have a non-zero magnitude since they are able to capture the stochastic noise intrinsic to the random walks of the simulated particles. Thus, the average variance of these high-order coefficients represents the system-level variance similar to the high-order Fourier modes depicted in Figures 2-1 and 2-2, and this variance level reduces in magnitude as the number of particles per generation is increased.

Similar to Section 2.4.1, when the starting source is offset from center, large differences start to arise between the entropy-based and FET-based approaches when the number of neutrons per generation is varied. Figure 2-23 plots the number of generations

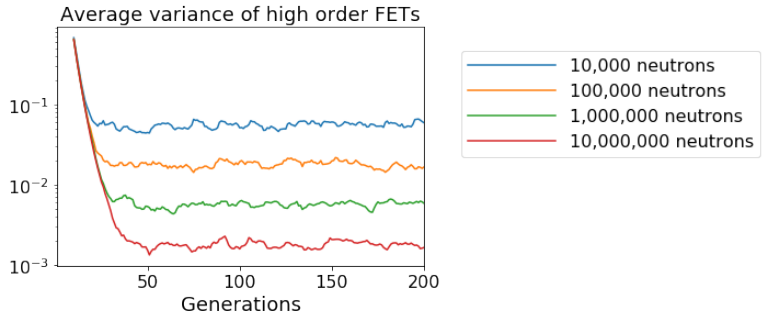


Figure 2-22: Average variance of high-order modes as a function of number of generations, for varying number of particles per generation and 1-D problem with starting point source at center.

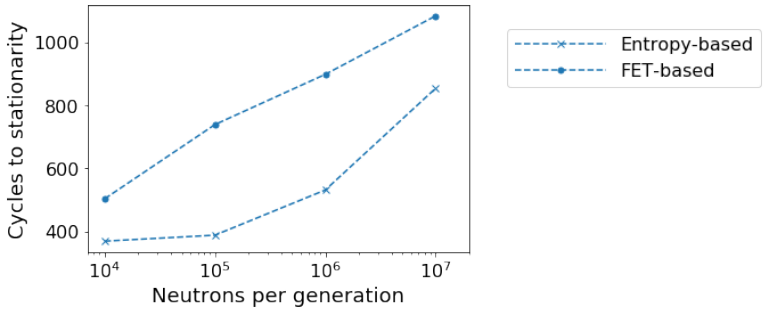


Figure 2-23: Generations to stationarity for FET-based and entropy-based approaches for generation sizes for 400cm slab with initial point source offset 10cm away from right boundary of problem.

to stationarity for a 400cm problem with varying number of neutrons per generation, but this time the starting point source is offset 10cm away from the right boundary. Once again, it can be seen that a larger number of neutrons is required to reach stationarity as the generation size increases.

Figure 2-24 shows the value of Shannon entropy and the slowest converging FET coefficient, where now the odd-parity FET coefficients become more significant due to the asymmetry in the source distribution. Once again it can be seen that the value of entropy flattens out much more rapidly than the a_1 coefficient, thus mispredicting the number of generations to stationarity. Figure 2-25 plots the relative difference between a reference cosine distribution and the stationary source distribution as determined by the entropy-based and FET-based methods. The entropy-based approach has a clear tilt in the stationary source distribution, and while the magnitude of the error diminishes as the number of neutrons in the system increases, the tilt does persist more clearly than

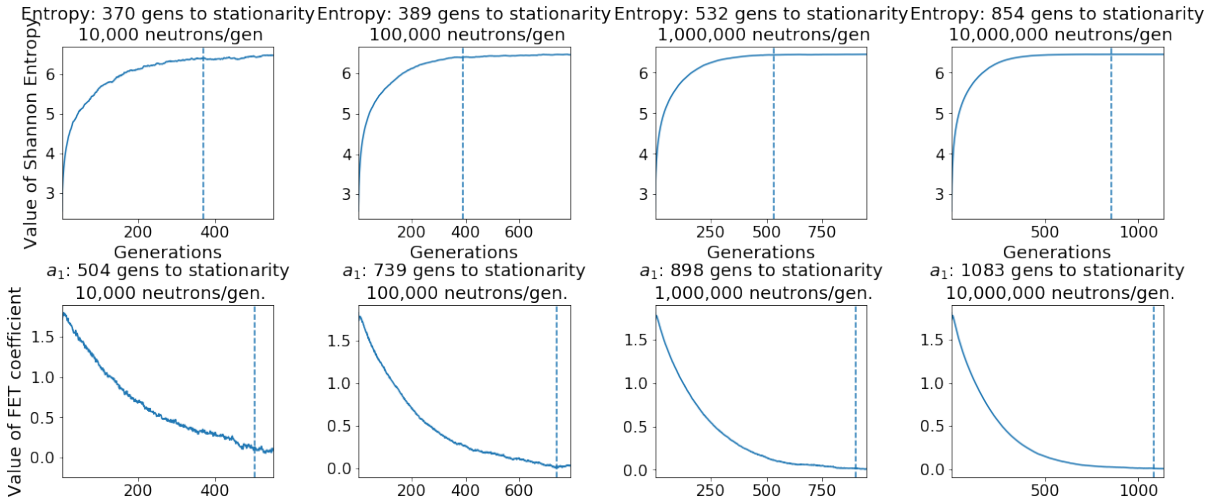


Figure 2-24: Value of Shannon entropy and slowest-converging FET coefficient as a function of generation size for a fixed slab length of 400cm. Values correspond to an initial point source offset 10cm away from right boundary of problem.

the FET-based approach. The large errors near the problem boundaries are due to the small magnitudes of the reference cosine shape in these regions.

For an offset starting source, the variance behavior of the low order and high order FET coefficients once again shows that a larger number of generations is required for simulations with larger number of neutrons per generation in order for variance levels to approach their lower asymptotic levels. Figure 2-26 plots the variance of the a_1, a_3, a_5, and a_7 coefficients, as well as the average variance of the high-order FET coefficients. The threshold variance levels of the high-order FET coefficients are on the same order as those in Figure 2-22, showing that the system-level variance levels represented by these high-order FET coefficients follow σ^2/n behavior and have consistent threshold values irrespective of the starting source guess.

2.4.3 Stationarity Behavior as a Function of Seed Number

In order to show that the results presented in Section 2.4.2 are not dependent on a particular random seed, the analysis is replicated for multiple independent seeds. Figure 2-27 plots the generations to stationarity when run with five different seeds using a starting point source at the center. Similarly, Figure 2-28 plots the generations to stationarity for five seeds when the starting point source is offset 10cm away from

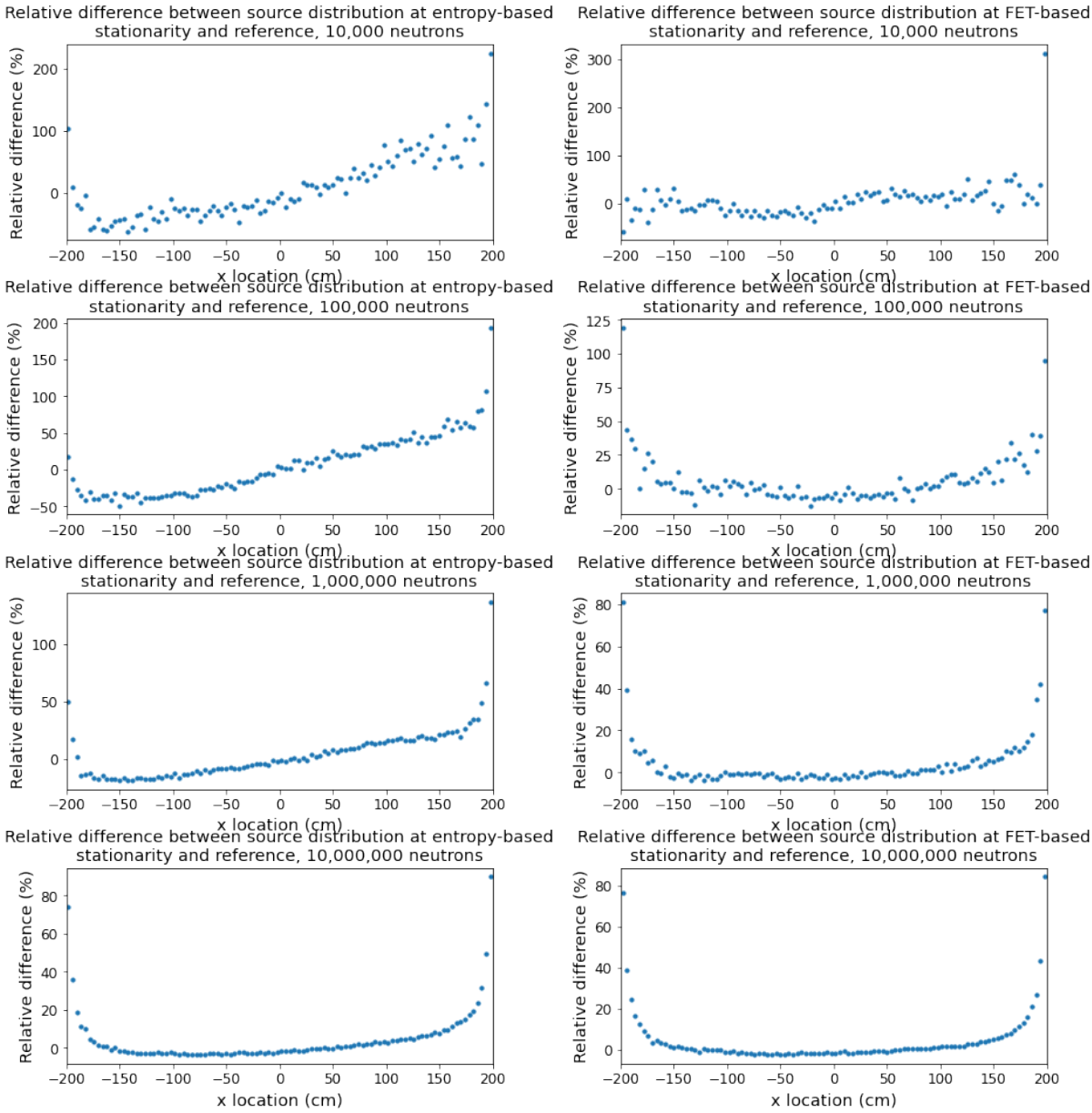


Figure 2-25: Relative error with reference cosine distribution for source distributions at the stopping criteria generation, as predicted by the entropy-based metric (left) and the FET-based diagnostic (right) for a 400cm problem with varying number of neutrons and a starting point source 10cm away from the right boundary.

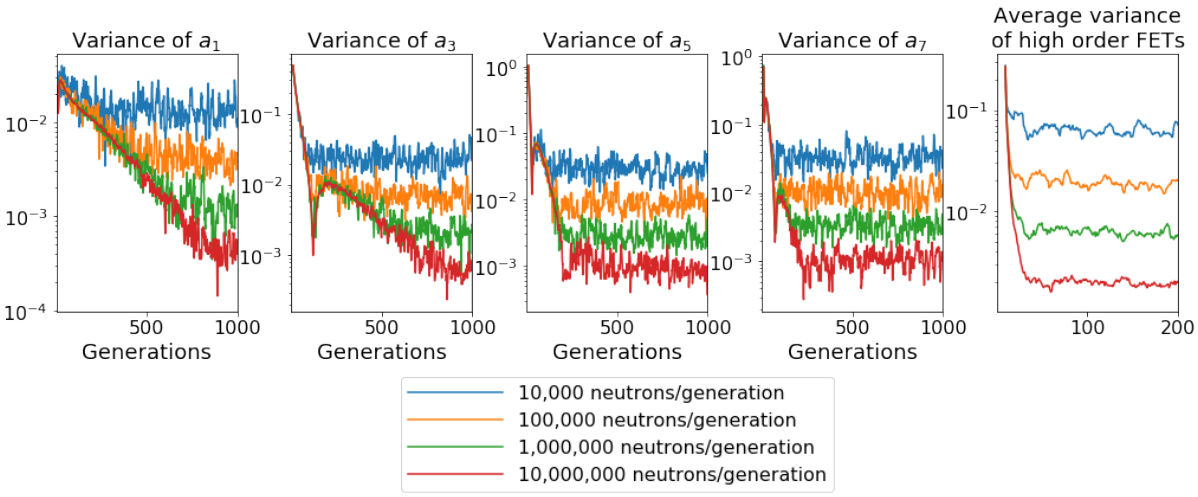


Figure 2-26: Variance of slowest converging FET coefficients a_1, a_3, a_5, and a_7, for different generation sizes and a fixed domain size of 400cm with an initial point source 10cm from right boundary of problem. The rightmost plot shows the average variance of high-order modes as a function of number of generations.

the right boundary. The average behavior of the five seeds shows that there is close agreement between the FET-based and entropy-based diagnostics with a central starting point source. On the other hand, the average of the five seeds yields a large discrepancy between the two methods when an offset starting point source is used, where Shannon entropy prescribes an insufficient number of generations to stationarity. Moreover, both Figure 2-27 and 2-28 suggest that a larger number of generations should be simulated to reach stationarity when the number of neutrons per generation is increased.

Stationarity analysis on a one-dimensional slab problem indicates that a larger number of inactive generations is required to reach stationarity for problems with larger dominance ratios (controlled by slab size) and for problems with larger generation sizes. FET-based diagnostic and Shannon entropy recommend similar generations to stationarity for problems with a symmetric source but Shannon entropy underpredicts the number of inactive generations required for problems with a highly asymmetric starting source.

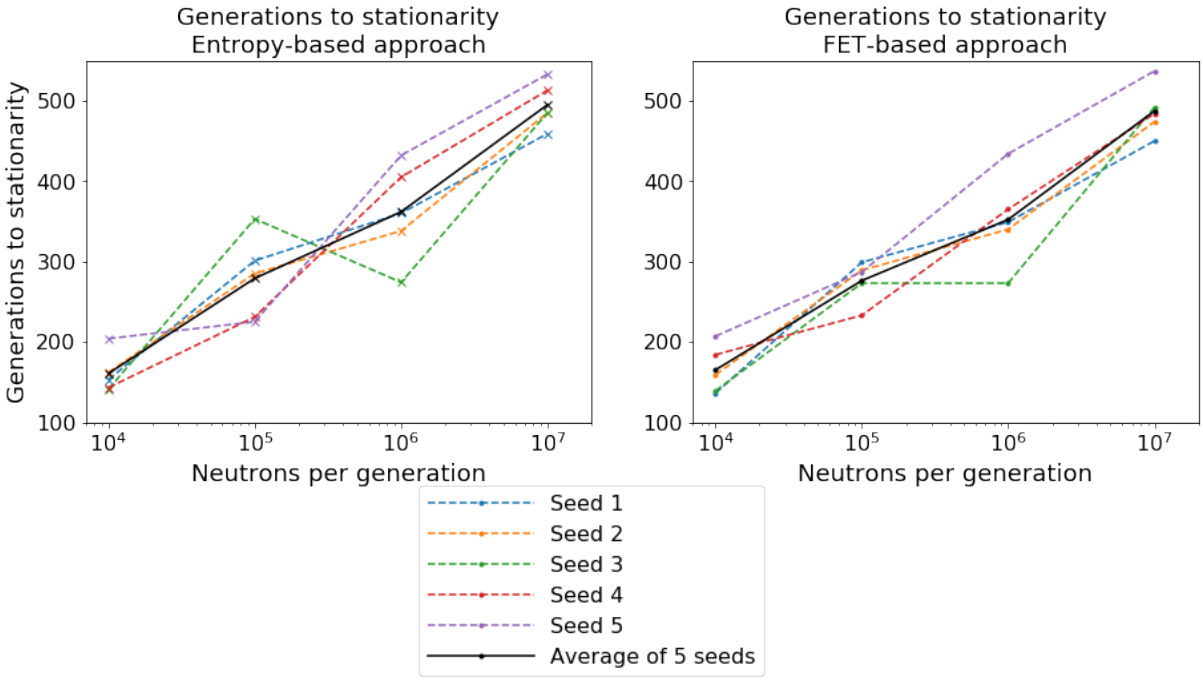


Figure 2-27: Generations to stationarity as a function of number of neutrons for five seeds of 400cm slab problem using entropy-based (right) and FET-based (left) approaches. The starting source is a point source at the center of the problem.

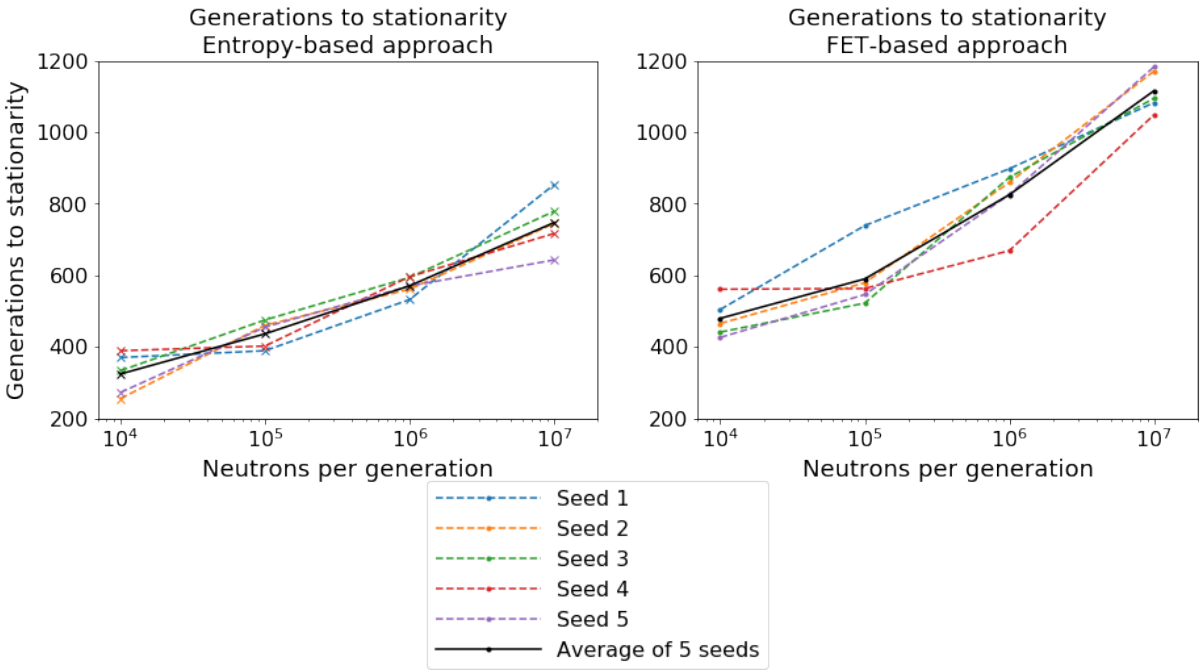


Figure 2-28: Generations to stationarity as a function of number of neutrons for five seeds of 400cm slab problem using entropy-based (right) and FET-based (left) approaches. The starting source is a point source offset 10cm away from right boundary of problem.

2.5 Results: Two-Dimensional PWR Benchmark Problem

The analysis in Section 2.4 can be extended to two dimensions by changing the basis set used to compute FET's. This section uses a Zernike expansion in the radial and azimuthal directions to capture the two-dimensional spatial detail of the fission source, and the 2-D BEAVRS benchmark is simulated in order to model the stationarity behavior of a typical PWR [64]. The dominance ratio of the problem, as measured using CMFD, is 0.9955. Now, the two-dimensional fission production rate is computed as:

$$\nu\Sigma_f(r,\theta)\psi(r,\theta) \approx \sum_{n}^{N}\sum_{m=-n}^{n} a_n^m Z_n^m\left(\frac{r}{R},\theta\right) \tag{2.12}$$

where N is the degree of radial order expansion of the Zernike basis set and Z_n^m is the Zernike polynomial of radial order n and azimuthal order m. a_n^m is the expansion coefficient corresponding to Zernike radial order n and azimuthal order m, and the radial domain is scaled over a unit disk and θ represents the azimuthal angle in cylindrical coordinates. In this case, the spatial domain is defined as the smallest circle that circumscribes the axial cross section of the core, where R is the radius of the circumscribed circle. The expansion coefficient a_n^m is computed as:

$$a_n^m = \begin{cases} \frac{2n+2}{\pi R^2}\int_0^R\int_0^\pi \nu\Sigma_f(r,\theta)\phi(r,\theta)Z_n^m\left(\frac{r}{R},\theta\right)rdrd\theta & \text{if } m \neq 0 \\ \\ \frac{n+1}{\pi R^2}\int_0^R\int_0^\pi \nu\Sigma_f(r,\theta)\phi(r,\theta)Z_n^m\left(\frac{r}{R},\theta\right)rdrd\theta & \text{if } m = 0 \end{cases} \tag{2.13}$$

For reference, the basis functions corresponding to the Zernike polynomials up to the fourth radial order are shown in Figure 2-29.

Likewise, a quarter-assembly mesh is used to compute Shannon entropy. As outlined in Section 2.3, the same stopping criteria is used to define stationarity for the FET-based and entropy-based approaches, and the same signal-to-noise ratio is utilized to filter out noisy Zernike FET coefficients. All Zernike coefficients up to radial order 10 are considered low-order coefficients that are analyzed for stationarity, and all Zernike

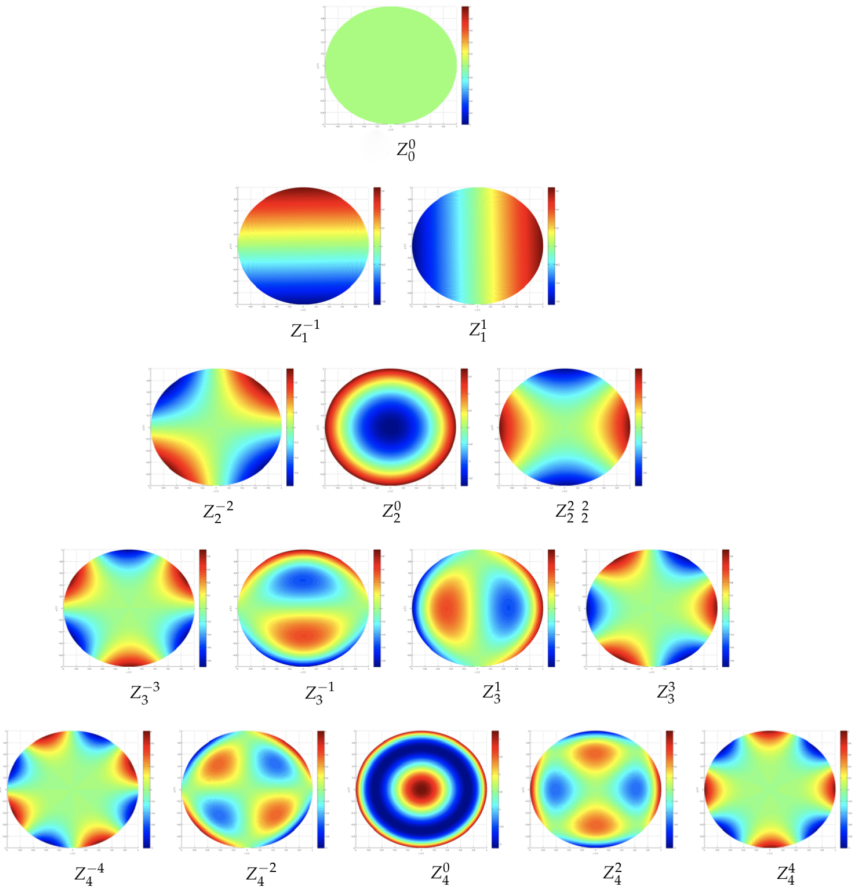


Figure 2-29: Zernike basis functions $Z_n^m(r,\theta)$ for $n = 0$ to $n = 4$.

coefficients of radial order 20 ($n = 20$ corresponds to 21 values for m) are aggregated to characterize high-order behavior.

Since the domain size of the 2-D BEAVRS problem is fixed, this section will focus solely on the effect of changing the number of neutrons per generation on the number of generations to stationarity. Similar to Section 2.4, the results will be presented for two different starting source distributions: one where the starting source is uniform over all fissionable areas and the other where all neutrons are born on the right half of the core. This offset source distribution will shed light into how the two stationarity diagnostics will fare in the presence of strong asymmetries in the fission source.

Figure 2-30 plots the generations to stationarity for 100,000, 1,000,000, 4,000,000, 10,000,000, and 40,000,000 neutrons per generation for the 2-D BEAVRS problem using the entropy-based and FET-based methods. The left plot uses a starting source that is uniform over all fissionable areas, while the starting source distribution for the right plot is uniform in all fissionable areas on the right half of the core. Thus, more generations to stationarity are required for the right plot as the neutrons need to migrate from the right half to the left half of the core. For the 2-D BEAVRS case, there is a small but noticeable discrepancy in the left plot between the two methods despite the symmetric starting source. This has to do with the fact that the fission source converges slowly for such problems so an entropy-based metric that aggregates fission source behavior into a single metric tends to falsely diagnose stationarity, when a closer analysis of specific spatial moments through an FET-based method would show that these modes have not fully converged yet. For both plots, it can also be seen again that a larger number of generations is required to reach stationarity as the number of neutrons per generation increases. As discussed in Section 2.4.2, this is due to the fact that for a larger number of neutrons per generation, more generations are required to converge the spatial modes to the reduced variance levels that result.

In order to illustrate the magnitude of the tilt that persists when starting with an offset starting source distribution, Figure 2-31 plots the relative difference in the stationary source distribution with the same distribution that is half-core symmetric for the different numbers of neutrons per generation. The plots on the left-hand side correspond to the

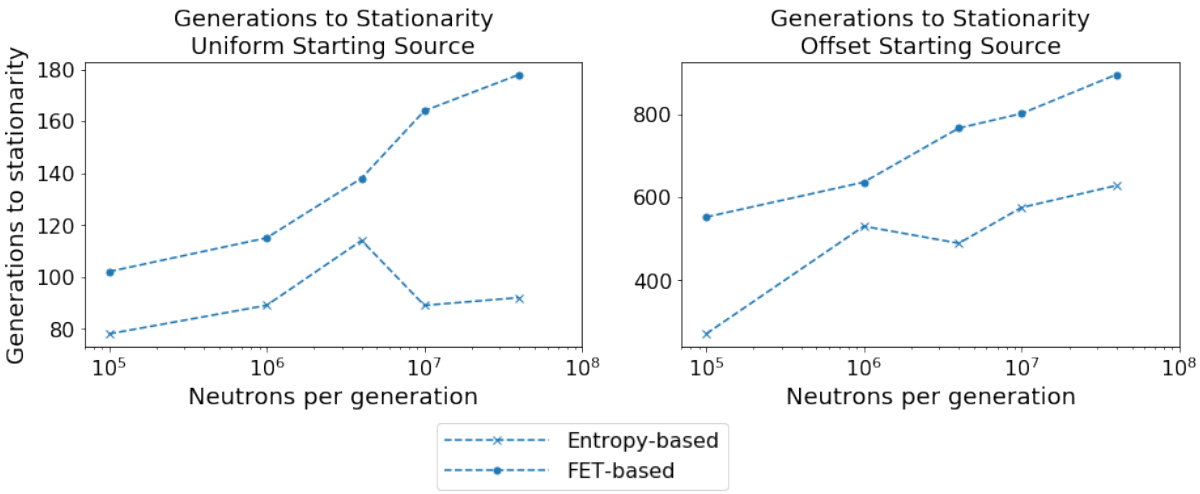


Figure 2-30: Generations to stationarity for FET-based and entropy-based approaches for different number of neutrons per generation when running on 2-D BEAVRS benchmark problem. Results based on starting source that is uniform in all fissionable areas (left) and starting source that is offset to the right half of the core (right).

relative error using the FET-based approach, while the plots on the right-hand side correspond to the entropy-based approach. Analogous to the one-dimensional case, there is a clear tilt in the source distribution as predicted by the entropy-based approach. As the number of neutrons per generation increases, the tilt becomes more prominent even though the maximum error decreases in magnitude. This same tilt is not observable for the FET-based approach, thus indicating that this diagnostic is able to accurately determine stationarity.

Finally, comparing the average variance of high-order modes for the two starting source distribution in Figure 2-32 shows that the threshold variance level is on the same order regardless of the initial source. Once again, the variance decreases as a function of $1/n$, where n is the number of neutrons per generation. Due to the similarity in the results for the 1-D and 2-D problems, detailed stationarity plots for the 2-D BEAVRS benchmark are included in Appendix B. These plots include the evolution of Shannon entropy and the slowest-converging FET's as a function of number of generations, the variance behavior of these stationarity diagnostics, as well as the aggregate stationarity results over multiple independent seeds.

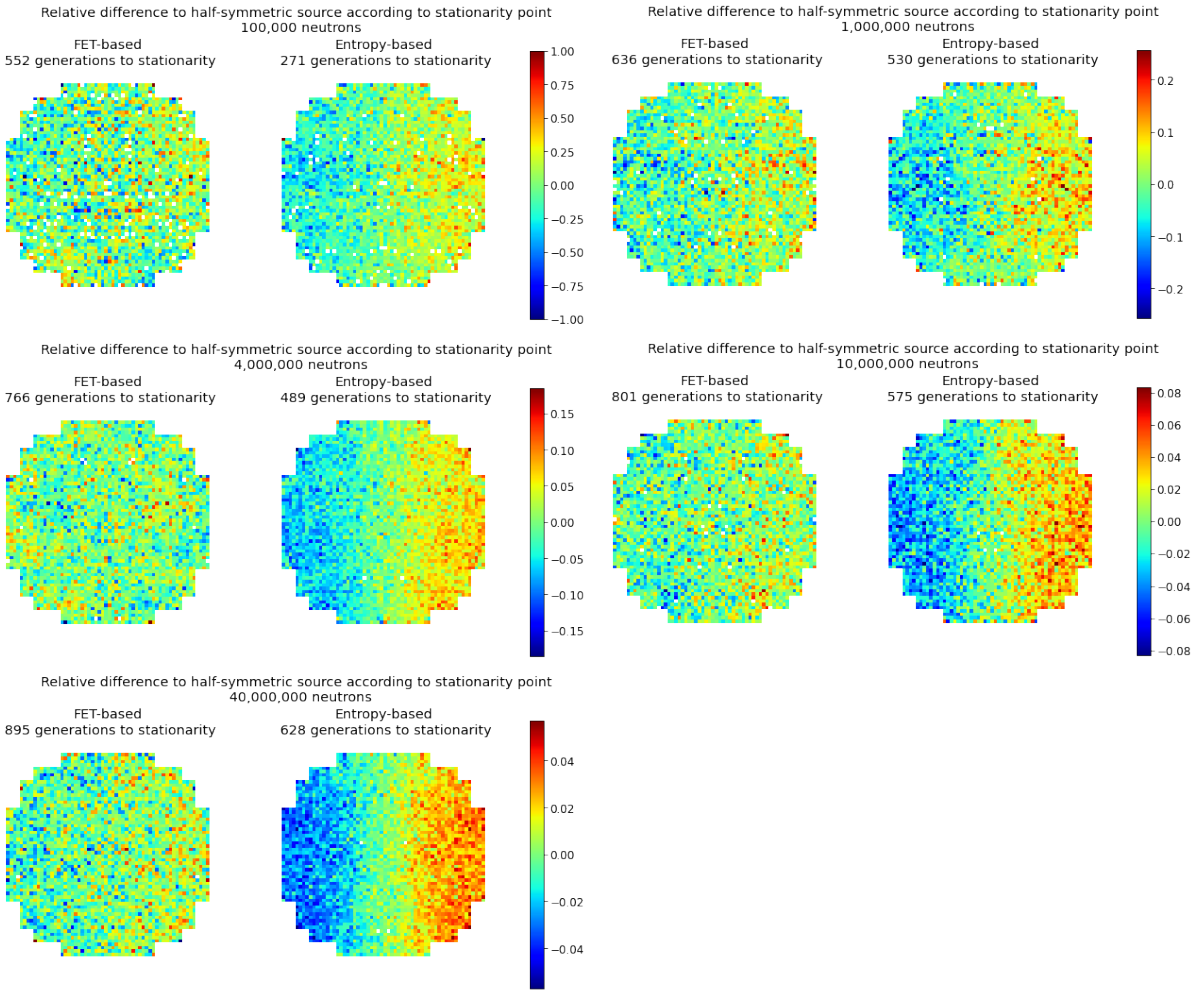


Figure 2-31: Relative difference in source distribution compared to source at stationary point, as predicted by the FET-based (left) and entropy-based (right) approaches for the 2-D BEAVRS benchmark problem with an offset starting source. Results are shown for various number of neutrons per generation.

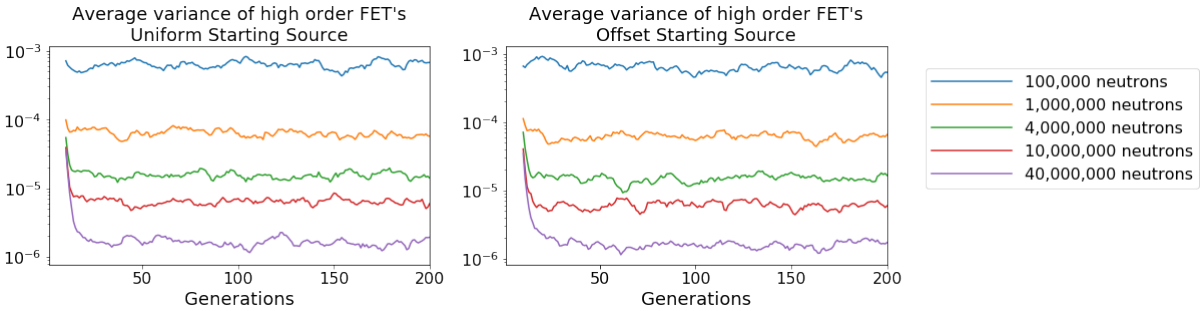


Figure 2-32: Average variance of high-order modes as a function of number of generations, for varying number of particles per generation when running on 2-D BEAVRS benchmark problem. Results based on starting source that is uniform in all fissionable areas (left) and starting source that is offset to the right half of the core (right).

Stationarity analysis on the 2-D BEAVRS problem shows that a larger number of inactive generations are required to reach stationarity for problems with larger generation sizes. The discrepancy between the FET diagnostic and Shannon entropy with respect to generations to stationarity grew larger even for problems with a symmetric source. Shannon entropy considerably underpredicts the number of inactive generations required with an offset initial source distribution.

2.6 FET Stationarity Implementation in OpenMC

Currently, OpenMC provides the functionality to compute FET coefficients by defining the FET as a tally, where the filter is the basis function of interest and the reaction rate is the nu-fission reaction rate. However, such an implementation for calculating FET coefficients is not efficient since it requires the computation of tallies in the inactive generations, which can severely reduce calculation rates. For example, the calculation rate drops from 3003 neutrons per second to 2091 neutrons per second when FET tallies are computed instead of Shannon entropy for the 2-D BEAVRS problem. This simulation was run on a single Intel Core i7 4790 processor with 1 Message Passing Interface (MPI) process and 8 OpenMP threads.

If the FET tally is represented as an analog estimator instead of a collision estimator, tally bins will be scored to only when a fission reaction occurs. The parallel efficiency of FET calculations can thus be greatly improved if FET coefficients are processed for groups of particles at a time through the stored fission bank at the end of each generation. Since the fission bank stores the spatial locations of all fission reactions that occur in a generation, FET coefficients can be computed by processing the expansion moments for each particle in the fission bank in parallel. In this light, these FET's can be viewed as the spatial moments of the fission bank for each generation, and thus the stationarity in the FET coefficients imply stationarity in the spatial moments of the fission bank. From an implementation standpoint, each MPI process is in charge of computing FET coefficients

on its local fission bank, while each OpenMP thread processes the contribution of each source site's location to each FET expansion coefficient in parallel. Once this is completed, these coefficient values are reduced onto the master process to obtain aggregate values over the entire global fission bank.

In order to assess the performance implications from FET computation, stationarity FET's are calculated for a variety of calculation methods and compared to the computation of traditional metrics such as Shannon entropy. Figure 2-33 plots the overall runtime as a function of number of simulated inactive generations for the 400cm 1-D homogeneous problem from this chapter with 10 million neutrons per generation. The problem is run once again on a single Intel Core i7 4790 processor with 1 MPI process and 8 OpenMP threads. The trendlines "Analog Estimator FET" and "Collision Estimator FET" refer to runs where FET's are specified directly as analog and collision estimators respectively, using a Legendre basis expansion up to order 10. The trendline "Post-process Fission Bank FET" signifies the computation of the same FET's by post-processing the fission bank in parallel after the particle transport routine has completed. Finally for comparison, the trendline "Shannon Entropy" indicates a run where Shannon entropy was calculated on a 1 cm mesh, and "No Shannon Entropy, No FET" specifies a run where neither Shannon entropy nor FET's were computed. These results indicate a significant computational overhead associated with specifying FET's directly as OpenMC tallies, whereas the post-processing of the fission bank yields similar runtimes compared to simulations with and without Shannon Entropy.

Similar conclusions can be drawn when looking at Figure 2-34, where the different stationarity metrics are computed on the 2-D BEAVRS problem with a generation size of 10 million. All FET calculations compute the coefficients of a radial and azimuthal Zernike expansion up to radial order 10. Thus, for the remainder of this thesis, the fission bank post processing method will be used to compute FET's for stationarity diagnostics. While an analog estimator possesses higher variance compared to a collision estimator, a relatively large generation size of 10 million will be used for most test cases and thus the conclusions drawn about stationarity should not vary greatly with the use of one estimator type over the other.

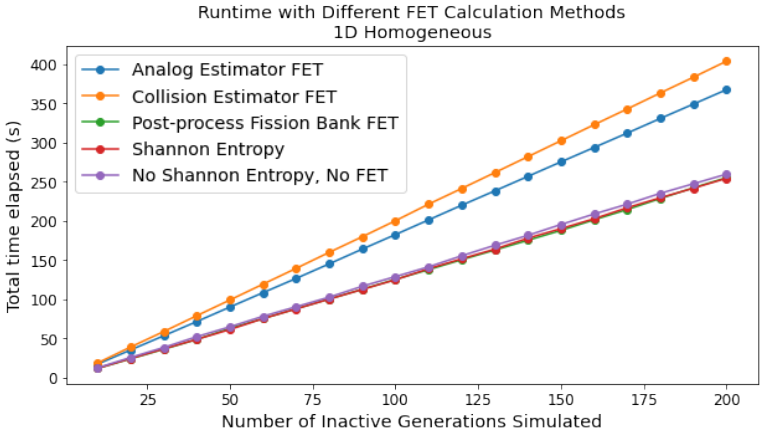


Figure 2-33: Runtime of different stationarity metrics as a function of number of inactive generations for a 400cm 1-D homogeneous problem with a generation size of 10 million. Green, purple, and red lines are on top of each other and indicate similar performance between these three cases.

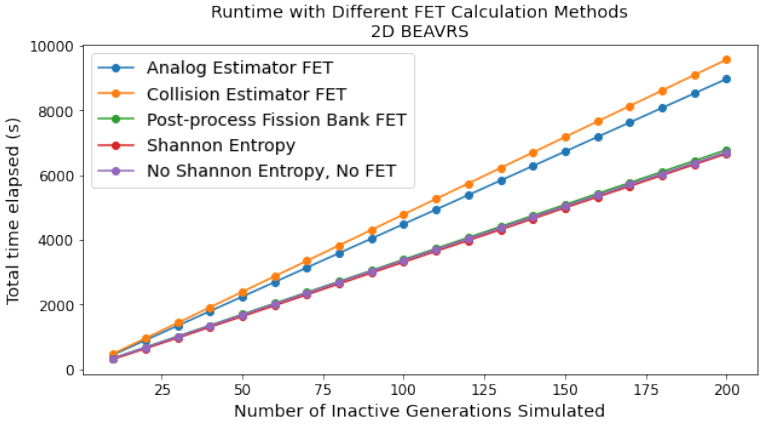


Figure 2-34: Runtime of different stationarity metrics as a function of number of inactive generations for the 2-D BEAVRS problem with a generation size of 10 million. Green, purple, and red lines are on top of each other and indicate similar performance between these three cases.

FET coefficients are calculated in OpenMC by post-processing the fission bank and using a Legendre basis set for 1-D problems and a Zernike basis set for 2-D problems. Runtime to compute these coefficients is on par with that for calculating Shannon entropy.

2.7 Conclusion

As more and more large-scale simulations are run to match the growing magnitude of available computing resources, it becomes even more imperative for users to be able to accurately pinpoint when stationarity occurs in Monte Carlo simulations in order to reduce the computational burden of simulating additional generations without introducing biases to the source distribution. This chapter investigates the issue of determining fission source stationarity by proposing a new methodology based on analyzing the stationarity of FET coefficients, which can represent the higher order spatial moments of the fission source. The stationarity diagnostics from FET expansion coefficients are compared to those from using Shannon entropy, which has been widely adopted by the Monte Carlo reactor physics community.

The stationarity behavior of the FET-based and entropy-based diagnostics are evaluated on one-dimensional and two-dimensional test cases with varying parameters. This analysis relies on defining a stopping criteria to determine whether the underlying stationarity metric has changed relative to its statistical fluctuations. For the one-dimensional homogeneous problem with vacuum boundary conditions, a Legendre basis set expansion was used to compute FET coefficients, while 100 mesh points were used to calculate Shannon entropy. It was found that:

- For a fixed number of neutrons per generation, there is a close agreement in the number of generations to stationarity between the entropy-based and FET-based approach when the domain size is varied and the starting source is a point source at the center of the problem. When the starting point source is offset 10cm away

from the right boundary, the two methods diverge and the number of generations to stationarity is underpredicted when using Shannon entropy.

- When the number of neutrons per generation is increased, a larger number of generations is required to reach stationarity. This is due to the fact that a larger number of particles per generation imposes a more stringent stopping criteria due to the lower threshold variance levels in the FET coefficients that can be achieved.

- The average variance of high-order modes represents the system-level noise of the system and follows the σ^2/n behavior of uncorrelated tallies.

This analysis is extended to the 2-D BEAVRS problem as well by using a Zernike expansion in the radial and azimuthal directions to represent the two-dimensional fission source and compute FET coefficients. Likewise, a quarter assembly mesh was utilized to compute Shannon entropy. For such problems, tracking all FET coefficients up to radial order 10 were more than sufficient in capturing low-order fluctuations for stationarity diagnosis. However, it is recommended that higher order FET coefficients should be computed if the converged source distribution is highly asymmetric and unlikely to be explained by Zernike expansion functions. Here, it was found that:

- There is a slight underprediction in the generations to stationarity when using Shannon entropy as opposed to an FET-based approach when the starting source is uniform over all fissionable areas. This discrepancy increases significantly when the starting source is restricted to the right half of the core. In some cases, Shannon entropy underpredicts the number of generations to stationarity by as many as 250 generations.

- Similar to the 1-D case, a larger number of generations is required to required stationarity when the number of neutrons per generation increases, and the average variance of the high-order modes follow σ^2/n behavior.

The 1-D and 2-D problems with offset starting distributions are extreme test cases but they help to illustrate the benefits of an FET-based diagnostic. Since FET's are meshless

and each source particle contributes to the basis functions anywhere in space, examining the stationarity in FET coefficients helps to ascertain whether each spatial mode has stopped fluctuating within statistical limits. On the other hand, Shannon entropy is a mesh-based approach that examines the migration of particles in each mesh cell from generation to generation in order to signal stationarity. If this migration occurs slowly, this frequently leads to situations of false convergence when using Shannon entropy.

Finally, it was shown that by post-processing the banked fission sites at the end of each generation, efficient computation of FET coefficients can be carried out with minimal overhead to the underlying particle transport algorithm. Thus, for problems with higher dimensionality and larger asymmetries, an FET-based diagnostic should be used in tandem with Shannon entropy in order to ensure that a properly converged fission source distribution is used in the active generations. Since the exact cutoff point between inactive and active generations is needed in order to understand the total runtime for Monte Carlo simulations, the FET-based stationarity diagnostic will be used throughout the rest of this thesis to assess the number of inactive generations needed to reach stationarity for a variety of test cases. The need for a reliable stationarity diagnostic will become especially important in Chapter 5, where a full-core 3-D PWR benchmark will be simulated, and it will be shown that Shannon Entropy does not adequately determine stationarity in the underlying fission source. For the 3-D case, the FET coefficients will be calculated by combining the 2-D Zernike basis set in the radial direction with the 1-D Legendre basis set in the axial direction.

Highlights

- An FET-based stationarity diagnostic is proposed that monitors the stationarity of a set of expansion coefficients to assess overall fission source stationarity.
- A stopping criteria is devised in order to compare generations to stationarity as prescribed by Shannon entropy and by the proposed FET-based method.
- The FET-based method is shown to outperform Shannon entropy for problems with a highly asymmetric starting source as well as higher-dimensionality problems.
- FET coefficients are computed efficiently in OpenMC by post-processing the fission bank in parallel after the particle transport routine once a generation has completed.

Chapter 3

Parameter Sensitivity Study of CMFD Source Acceleration

As alluded to in Chapter 1, a significant computational burden arises in Monte Carlo methods from having to simulate an adequate number of inactive generations from an initial starting guess in order to attain a stationary fission source. Chapter 2 discussed a novel heuristic to determine when this stationary point occurs using FET's, where it was shown that monitoring the stationarity in individual FET coefficients gave a more holistic diagnosis of fission source stationarity over more traditional metrics such as Shannon entropy. In such cases, however, nearly 1000 inactive generations could be required for certain problems before stationarity was achieved. In order to reduce the burden of computation in the inactive generations, this chapter will investigate the use of source acceleration methods to reduce the overall number of discarded generations before beginning the tallying process to compute quantities of interest and sample statistics.

This chapter begins with a review of source acceleration methods in Section 3.1, with a focus on CMFD source acceleration due to its extensive application to simulation of reactor problems. Section 3.2 discusses how FET stationarity diagnosis can be extended to Monte Carlo simulations that utilize CMFD source acceleration and defines the 1-D and 2-D test problems that will be analyzed for the remainder of this thesis. While Section 3.2 delineates the CMFD parameters used as part of the source acceleration scheme, a sensitivity study on select CMFD parameters will be conducted in Section 3.3

to illustrate how these parameters affect the evolution of Monte Carlo simulations in the inactive generations. A similar sensitivity study is conducted in Section 3.4 to show the effects of CMFD parameters in the active generations. Finally, Section 3.5 combines the findings from this chapter to discuss optimal run strategies when running Monte Carlo with CMFD source acceleration, with a focus on reducing overall runtime for the test problems defined in Section 3.2.

3.1 Review of CMFD Source Acceleration

A variety of methods for accelerating source convergence has been proposed for Monte Carlo methods in order to effectively reduce the number of generations required to reach fission source stationarity [65–68]. While a detailed review of these methods can be found in Li's thesis [1], very few of these methods have successfully been applied to full-core reactor problems without significantly increasing memory or computational requirements. CMFD source acceleration is one such method that has been shown by Kelly et al. and Herman to dramatically reduce the number of inactive generations required for the simulation of 3-D PWR problems using continuous energy Monte Carlo [26, 69], and will be the primary focus of this thesis.

3.1.1 CMFD Equations

CMFD can be understood as a diffusion-based multigrid acceleration method, where the feedback loop is carried out by taking tallied quantities during each Monte Carlo power iteration as input to solve an eigenvalue problem with a system of linear diffusion-based equations. The solution of these equations yields an approximate neutron source distribution, and the weights of each particle in the fission bank are adjusted accordingly after each CMFD update step to dictate the starting source distribution for subsequent Monte Carlo iterations. The effectiveness of CMFD comes from the fact that the diffusion-based CMFD fission source used to reweight fission bank particles is computed inexpensively and drives the system to an approximate solution much more quickly than a typical Monte Carlo power iteration process. CMFD-Monte Carlo coupling is typically divided

into four steps, with equations that govern each step provided in the following sections. A complete derivation of the CMFD equations can be found in Herman's thesis [26].

3.1.1.1 Computation of CMFD tallies

As input to the CMFD solver, volume-integrated reaction rates and surface-integrated currents are computed at each generation during the Monte Carlo transport step. A summary of these terms is provided in Table 3.1, where

$$
\begin{aligned}
g, g' &\equiv \text{Energy group index} \\
m &\equiv \text{Mesh index} \\
V_m &\equiv \text{Volume of } m\text{-th mesh cell} \\
A_m^u &\equiv \text{Area of surface of } m\text{-th mesh perpendicular to } u\text{-th direction} \\
J_{g,m}^{u\pm} &\equiv \text{Net current of } u \pm \text{surface at the } m\text{-th mesh}
\end{aligned}
$$

Here, $u \in (x, y, z)$ and refers to the current along a given axis, while $\pm$ refers to the left $(-x)$ or right $(+x)$ surface in the x axis, the back $(-y)$ or front $(+y)$ surface in the y axis, and bottom $(-z)$ or top $(+z)$ surface in the z direction.

Table 3.1: Tallied CMFD quantities during the Monte Carlo power iteration

Term	Description
$\langle \phi_{g,m} V_m \rangle$	Volume-integrated flux
$\langle \Sigma_{t,g,m} \phi_{g,m} V_m \rangle$	Volume-integrated total reaction rate
$\langle \nu_s \Sigma_{s,g' \to g,m} \phi_{g',m} V_m \rangle$	Volume-integrated scattering production rate
$\langle \nu_s \Sigma_{s1,g' \to g,m} \phi_{g',m} V_m \rangle$	Volume-integrated P_1 scattering production rate
$\langle (\chi \nu\Sigma)_{f,g' \to g,m} \phi_{g',m} V_m \rangle$	Volume-integrated fission production rate
$\langle J_{g,m}^{u\pm} A_m^u \rangle$	Surface-integrated net current

3.1.1.2 Computation of CMFD Cross Sections, Diffusion Coefficients, and Coupling Coefficients

During the restriction step, the CMFD tallies are used to define a set of diffusion equations over a coarse spatial mesh and energy grid. Here, the inputs to these equations are a set of MGXS, diffusion coefficients, and coupling coefficients that are computed from Monte Carlo tallies from the previous step. Table 3.2 outlines the different parameters that are calculated in this step in order to set up the CMFD equations that will be solved in the next step.

Table 3.2: CMFD MGXS and diffusion coefficient parameters.

Term	Description
$\Sigma_{t,g,m} = \frac{\langle \Sigma_{t,g,m} \phi_{g,m} V_m \rangle}{\langle \phi_{g,m} V_m \rangle}$	Total macroscopic cross section
$\nu_s \Sigma_{s,g' \to g,m} = \frac{\langle \nu_s \Sigma_{s,g' \to g,m} \phi_{g',m} V_m \rangle}{\langle \phi_{g,m} V_m \rangle}$	Scattering production macroscopic cross section
$(\chi \nu\Sigma)_{f,g' \to g,m} = \frac{\langle (\chi \nu\Sigma)_{f,g' \to g,m} \phi_{g',m} V_m \rangle}{\langle \phi_{g,m} V_m \rangle}$	Fission production macroscopic cross section
$D_{g,m} = \frac{\langle \phi_{g,m} V_m \rangle}{3(\langle \Sigma_{t,g,m} \phi_{g,m} V_m \rangle - \langle \nu_s \Sigma_{s1,g' \to g,m} \phi_{g',m} V_m \rangle)}$	Diffusion coefficient
$\tilde{D}^{u\pm}_{g,m} = -\frac{2 D_{g,m\pm u} D_{g,m}}{D_{g,m\pm u} A^u_m + D_{g,m} A^u_m}$	Linear coupling term between current and flux
$\hat{D}^{u\pm}_{g,m} = \frac{J^{u\pm}_{g,m} \pm \tilde{D}^{u\pm}_{g,m} (\phi_{g,m\pm u} - \phi_{g,m})}{\phi_{g,m\pm u} + \phi_{g,m}}$	Nonlinear coupling term between current and flux

Here, the term $\tilde{D}^{u\pm}_{g,m}$ represents the linear coupling term between current and flux in the diffusion equation. In order to maintain equivalence between the diffusion solution and the transport solution, the nonlinear term $\hat{D}^{u\pm}_{g,m}$ is introduced to the CMFD equations.

3.1.1.3 Solution of CMFD Equations

With the parameters from the previous section, the linear set of equations governing the CMFD solver can be constructed and solved for each energy group and spatial mesh:

$$\begin{aligned}\sum_{u\in(x,y,z)} A_m^u \Big[&-\tilde{D}_{g,m}^{u+}(\phi_{g,m+u}-\phi_{g,m}) + \hat{D}_{g,m}^{u+}(\phi_{g,m+u}+\phi_{g,m}) \\ &-\tilde{D}_{g,m}^{u-}(\phi_{g,m-u}-\phi_{g,m}) + \hat{D}_{g,m}^{u-}(\phi_{g,m-u}+\phi_{g,m})\Big] \\ &+\Sigma_{t,g,m}\phi_{g,m}V_m = \sum_{g'} \nu_s \Sigma_{s,g'\to g,m}\phi_{g',m}V_m + \frac{1}{k}\sum_{g'}(\chi\,\nu\Sigma)_{f,g'\to g,m}\phi_{g',m}V_m\end{aligned} \tag{3.1}$$

These equations can be written in matrix form as a balance equation:

$$\mathbb{M}\mathbf{\Phi} = \frac{1}{k}\mathbb{F}\mathbf{\Phi} \tag{3.2}$$

where

$\mathbb{M} \equiv$ Neutron loss matrix operator

$\mathbb{F} \equiv$ Neutron production matrix operator

$\mathbf{\Phi} \equiv$ Multigroup flux vector

$k \equiv$ CMFD eigenvalue

Given the sparsity of the matrices $\mathbb{M}$ and $\mathbb{F}$, these set of equations are solved in OpenMC using power iteration and a Gauss-Seidel linear solver for k and $\mathbf{\Phi}$. The computation of $\mathbf{\Phi}$ yields the scalar flux for every energy group g and spatial mesh m, and will be designated as $\phi_{g,m}^{\text{CMFD}}$ during the prolongation step in the next section.

3.1.1.4 Monte Carlo Fission Bank Source Reweighting

With the computation of $\phi_{g,m}^{\text{CMFD}}$, the Monte Carlo fission bank needs to be adjusted in order to incorporate the information from these newly computed scalar flux values. In

doing so, the source bank for the next Monte Carlo generation will follow a distribution that follows the fission source distribution as computed by the CMFD solver. The fractional CMFD fission source for each energy group g and spatial mesh m is computed as:

$$p_{g,m} = \frac{\sum_{g'} (\chi \nu\Sigma)_{f,g' \to g,m} \phi^{\text{CMFD}}_{g',m} V_m}{\sum_m \sum_{g'} (\chi \nu\Sigma)_{f,g' \to g,m} \phi^{\text{CMFD}}_{g',m} V_m} \tag{3.3}$$

and the weight adjustment factor for each energy group and spatial mesh is computed as:

$$f_{g,m} = \frac{p_{g,m}}{\sum_{i \in (g,m)} \omega_i} N \tag{3.4}$$

where $\sum_{i \in (g,m)} \omega_i$ represents the total weight of particles in the fission bank. Thus, in order to incorporate the CMFD fission source information back into the Monte Carlo solver, the weight of each particle in the fission bank is adjusted so as to match the CMFD fission source. For each particle i in the fission bank that belongs to energy group g and spatial mesh m, its weight ω_i is adjusted to weight ω_i' as:

$$\omega_i' = \omega_i f_{g,m} \tag{3.5}$$

Such a prolongation step assumes a flat fission source within each CMFD mesh cell, but a shape function can be imposed during the prolongation step as well to match the shape of the underlying fission source more closely. Section 3.2.3.5 discusses the use of a linear prolongation update step in more detail.

In this manner, the CMFD feedback loop represents a tight coupling between the Monte Carlo solver and the CMFD solver, where after each generation, tally information from Monte Carlo informs the parameters needed to construct the CMFD equations, and the resulting weight adjustment factors computed from the CMFD fission source are passed back to the Monte Carlo solver to adjust the weights of each particle in the Monte Carlo fission bank. This feedback loop is similar to that of a TH coupling mechanism where local temperature and density fields calculated by the TH solver are iteratively

passed to the Monte Carlo neutronics solver to update cross sections and recompute power distributions, which in turn informs the next update to TH temperatures and densities. In this light, CMFD can be seen as a surrogate for the non-linear coupling process and will be viewed as such in the ensemble averaging analysis of the following chapters.

3.1.2 Performance Considerations for CMFD Source Acceleration

Section 3.1.1 outlined the key equations used to describe the feedback loop between the Monte Carlo and CMFD solvers. In order to characterize the exact benefits from running CMFD, previous work has typically looked at generations to stationarity as a key indicator of runtime savings from running CMFD source acceleration [1, 26]. Two issues arise from such an analysis in order to measure the true impact of CMFD in the inactive generations. Firstly, a rigorous cutoff point between the inactive and active generations is not delineated in such works and instead a visual approach based on assessing when Shannon entropy has reached a stable level of fluctuations is used to mark this transition point. As was described in Chapter 2, Shannon entropy can lead to issues of false stationarity diagnosis and a more rigorous FET-based methodology should be opted for in order to characterize fission source stationarity. This is exactly the strategy that will be employed in this thesis to diagnose fission source stationarity for CMFD accelerated simulations and will be the subject of discussion in Section 3.2.

The second issue with focusing solely on number of inactive generations reduced in order to describe the benefits of running CMFD is that it tends to neglect other factors that come into play when trying to understand the efficacy of source acceleration methods with respect to overall simulation runtime. For example, while CMFD has been shown to reduce the number of inactive generations needed for Monte Carlo simulations, there is a computational overhead associated with running CMFD as well, namely in the form of additional tallied quantities that are required during the inactive Monte Carlo generations and the runtime associated with the CMFD solver. Moreover, naive CMFD implementations may yield a worse solution compared to an un-accelerated Monte Carlo

simulation, as will be shown in later sections. This will have implications in both the inactive and active generations, and thus Sections 3.3, 3.4, and 3.5 focus on analyzing the overall benefits of CMFD by analyzing its impact on the instantaneous fission source variance levels, RMS error levels in the active generations, and overall runtime.

CMFD source acceleration represents a tight coupling between Monte Carlo and a low-order diffusion-based solver, where CMFD tallies are passed from Monte Carlo to CMFD, and weight adjustment factors are passed from CMFD back to Monte Carlo.

3.2 CMFD Source Acceleration and FET Stationarity Diagnosis

As described in Chapter 2, the calculation of FET's is native to the Monte Carlo tallying routine, so any direct modification to the source of neutrons at each generations by an acceleration method will be reflected in the FET tally scoring for that generation. As was outlined in Section 3.1.1, this is exactly what occurs with CMFD acceleration where each particle in the fission bank is reweighted so as to drive source particles for subsequent generations closer to the CMFD-based fission source. The next generation of source particles is sampled from the updated fission bank where there is now a higher likelihood of sampling particles that belong to regions with a larger CMFD-based fission source. During the transport of particles in the next Monte Carlo generation, FET tallies are computed based on this updated source distribution, and will thus continue to reflect the spatial modes of the fission source that mirrors that of the CMFD-based solution. A simplified workflow of the CMFD acceleration routine coupled with FET tally computation is shown in Figure 3-1.

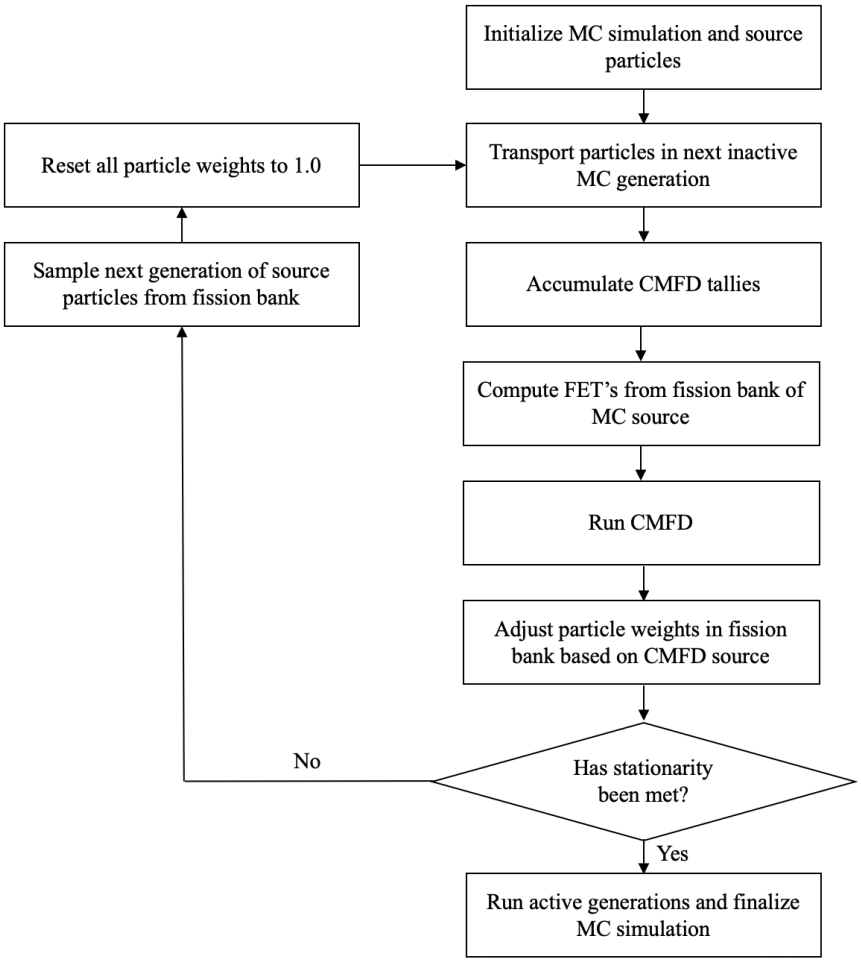


Figure 3-1: Flowchart for CMFD acceleration in the inactive generations with FET calculation.

3.2.1 Outline of Test Problems

Before illustrating the effects of CMFD on fission source stationarity, the one-dimensional and two-dimensional test problems that are the focus of this chapter and remaining chapters are outlined. The one-dimensional geometry will be the same homogeneous slab of uranyl sulfate that was introduced in Chapter 2. This time, however, the slab length will remain fixed at 400cm to match the dominance ratio of higher-dimensional reactor problems. In order to increase the difficulty of source stationarity, an offset starting source distribution will be used, where 10% of the initial source particles are distributed uniformly along the first 396cm of the slab, while the remaining 90% of

the particles are distributed uniformly along the remaining 4cm of the problem[1]. The two-dimensional problem will be the 2-D variant of the BEAVRS benchmark problem, also equivalent to the one presented in Chapter 2. The initial source distribution for this problem will be uniformly distributed throughout the core in fissionable areas. Thus, the 1-D problem will require many more inactive generations due to its highly tilted initial source distribution, but the 2-D problem possesses a more complicated geometry with many materials and will thus require more simulation time per generation. Chapter 4 will continue to investigate the same 1-D and 2-D problems but focuses on running simulations on larger computer systems. The ultimate test case will be the 3-D BEAVRS problem that models a tilted full-core BEAVRS benchmark problem in its complete three-dimensional geometry, which combines the difficulties of both the 1-D and 2-D problems. However, analysis of the 3-D problem occurs separately in Chapter 5 once the behavior of the 1-D and 2-D problems has been thoroughly investigated.

3.2.2 FET-based CMFD Stationarity on 1-D and 2-D Test Problems

In order to illustrate how the source distribution evolves as a function of the number of generations with the use of CMFD for the 1-D problem, Figure 3-2 plots the source distribution for the accelerated and un-accelerated cases at different generations. In both cases, the generation size is 10 million neutrons, and a fine CMFD mesh size of 0.4cm is used for the CMFD solver, where an expanding window scheme is used to reset tallies. A full list of CMFD parameters is listed in Table 3.3, and a more thorough discussion of how these parameters were chosen can be found in Section 3.2.3. For this problem, since CMFD feedback starts after the first generation, the source distributions are equivalent at generation 1. For subsequent generations, however, it can be seen that CMFD accelerates the source distribution to the true cosine shape much more rapidly.

Given that FET coefficients can continue to represent the strength of spatial modes of the fission bank for CMFD-accelerated schemes, the stopping criteria presented in

[1]Such a distribution is chosen instead of an offset point source since CMFD equations cannot be solved in regions with zero flux. While there are workarounds to this, modifying the initial source in this way was the simplest approach to this issue without having to significantly modify the CMFD solver in OpenMC

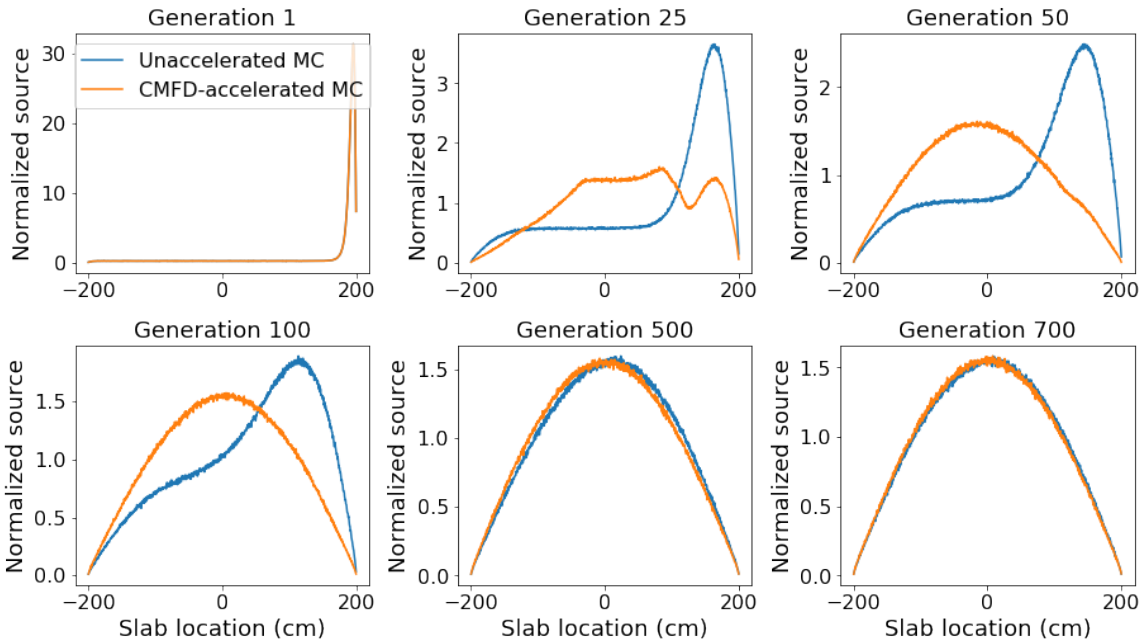


Figure 3-2: Source distribution at representative generations for the 1-D case with and without CMFD acceleration.

Chapter 2 can be applied to such schemes to diagnose stationarity as well, where the same parameters $S_A = 10, S_B = 30, S_C = 10$ are used for the remainder of problems in this thesis.

Figure 3-3 plots the FET coefficients a_1 to a_6 for the case with and without CMFD acceleration. It can be seen that the FET coefficients approach the same value for both cases, with the un-accelerated case requiring a larger number of generations to reach the stationary value. In addition, the vertical lines in each subplot indicate the point where the mean of the coefficients stops changing relative to the coefficient's noise levels. For a particular FET coefficient, the difference between the vertical lines represents the difference in number of generations required for that coefficient to reach stationarity from using CMFD acceleration. For the case with CMFD acceleration, the a_4 coefficient is the slowest converging, requiring 215 generations to reach stationarity, while the a_1 coefficient is the slowest converging for the un-accelerated case, requiring 874 generations. The difference between these two values represents the overall reduction in number of generations to stationarity by using CMFD acceleration.

When using Shannon entropy, however, the stationarity point is not correctly identi-

Table 3.3: CMFD parameters for 1-D homogeneous problem

Parameter	Value
CMFD tally estimator type	Analog
Number of energy groups	1
Effective downscatter	N/A
Acceleration map	None
Reference diffusion parameters	Not used
CMFD tally begin	1
CMFD solver begin	1
Weight clipping	0.2
Windowing scheme	Expanding window
Maximum window size	No limit
CMFD mesh size	0.4cm mesh
Generation size	10 million
Prolongation source shape	Fixed source

fied. Figure 3-4 plots Shannon entropy as a function of generation number for the same problem with and without CMFD acceleration. Based on the same stopping criteria for the FET coefficients, stationarity would appear to occur at the same vertical point due to the fact that the entropy function tends to flatten out relative to its noise before reaching a stable value. Once again, by using Shannon entropy the exact number of generations saved from using an acceleration method is not as apparent due to its inability to predict stationarity for the un-accelerated case. In such cases, the stopping criteria parameters S_A, S_B, and S_C would need to be adjusted to account for the slower migration of the fission source towards a stationary distribution. Such an approach, however, would make the success of Shannon entropy much more problem dependent.

Similar trends are observed for the 2-D BEAVRS problem. The behavior of representative FET coefficients based on the Zernike basis set is shown in Figure 3-5, and the list of CMFD parameters used for this problem are shown in Table 3.4. The CMFD-accelerated problem uses a quarter assembly acceleration mesh and takes 70 generations to reach stationarity, while the un-accelerated problem requires 160 generations. This difference in number of generations is not as large for the 2-D case since a symmetric starting

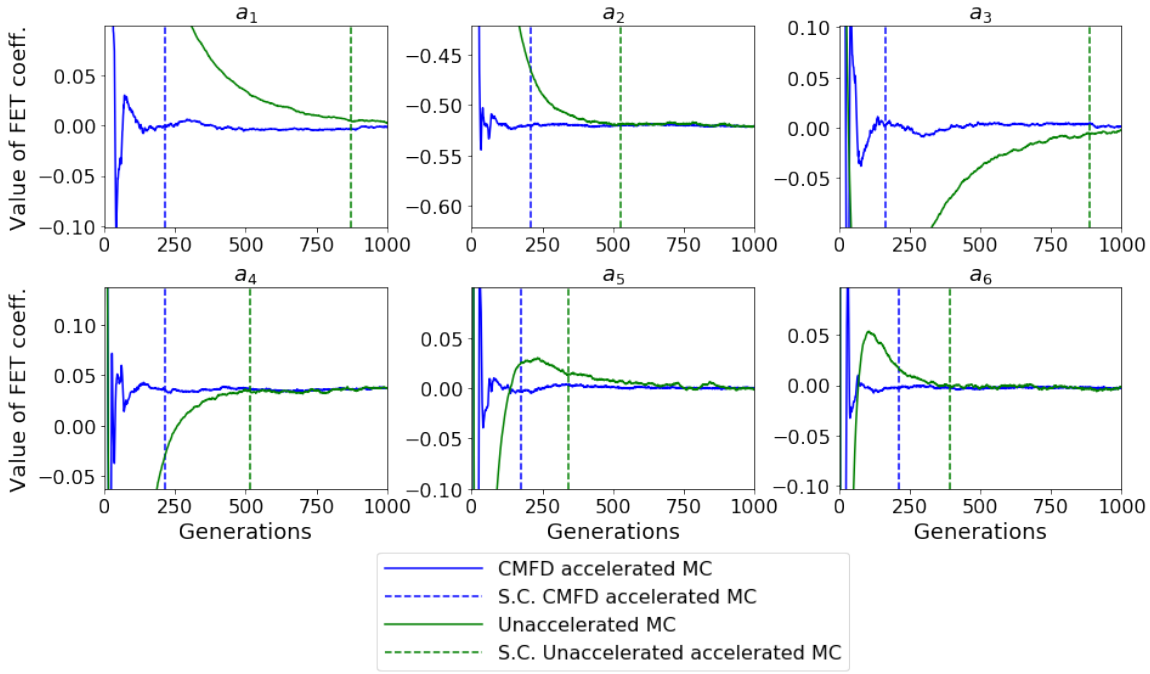


Figure 3-3: Value of FET coefficients a_1 to a_6 as a function of generation number for 1-D homogeneous test problem with CMFD acceleration (blue) and without CMFD acceleration (green). The dashed vertical lines represent the stopping criteria for each particular case.

source is used to simulate a symmetric problem, requiring fewer inactive generations to reach stationarity.

3.2.3 CMFD Parameter Choice

Many of the CMFD parameters selected for the 1-D and 2-D problems were chosen based on recommendations from the theses of Herman and Li, who focus significantly on determining optimal parameters to improve CMFD functionality [1, 26]. For the purposes of this chapter, the parameters that remain unchanged from these works are outlined in the next paragraph. Any parameter choices that deviate significantly from these works are outlined in Sections 3.2.3.1 to 3.2.3.5.

In this thesis, analog tallies are used to compute all CMFD tallies. Since many of the test cases will use an adequately large generation size of 10 million, the use of analog tallies trades faster computation time for a slight increase in tally variance levels. One energy group is used for the 1-D problem while the 2-D problem uses two energy

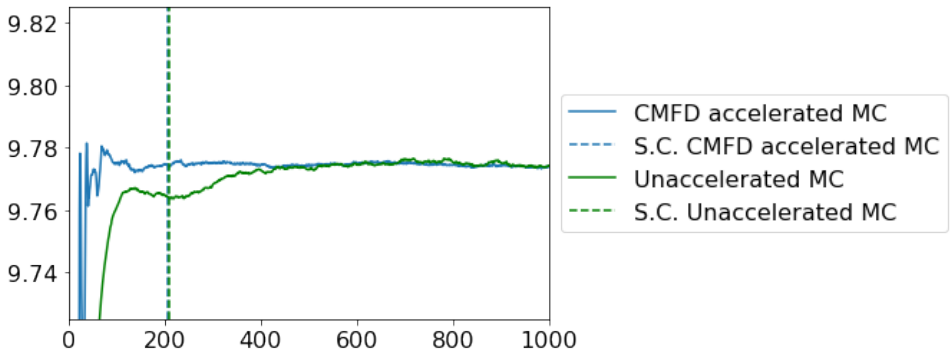


Figure 3-4: Value of Shannon entropy as a function of generation number for 1-D homogeneous test problem with CMFD acceleration (blue) and without CMFD acceleration (green). The dashed vertical lines represent the stopping criteria for each particular case.

groups as it was shown by Herman that a finer energy group structure did not increase the quality of the CMFD solution and also increases the runtime of the solver [26]. The effective down-scattering macroscopic cross section is used for the 2-D BEAVRS problem in order to remove any effects of infrequent neutron up-scattering events while preserving neutron balance. Moreover, for the 2-D case, an acceleration map proposed by Herman is used to overlay a Cartesian acceleration grid onto a non-Cartesian reactor geometry, where an albedo boundary condition is used to define partial currents at the core-reflector interfaces [26]. Finally, reference diffusion coefficients are used for the 2-D BEAVRS problem based on the Cumulative Migration Method (CMM) method developed by Liu et al. [70]. These diffusion coefficients were first used for Monte Carlo CMFD acceleration by Li in her thesis where she shows that fixing these values for each group reduces the number of CMFD tallies to be computed at each generation and drives the source distribution to a stationary value in fewer generations [1]. In this chapter, a Group 1 diffusion coefficient of 1.43 and Group 2 diffusion coefficient of 0.40 is used for the 2-D problem.

3.2.3.1 Tally Begin, Solver Begin, and Weight Clipping

Due to the high likelihood that the first generation of CMFD tallies can carry large bias and lead the initial generations of CMFD feedback in the wrong direction, both Herman and Li recommend waiting a few generations before starting the CMFD tallying process

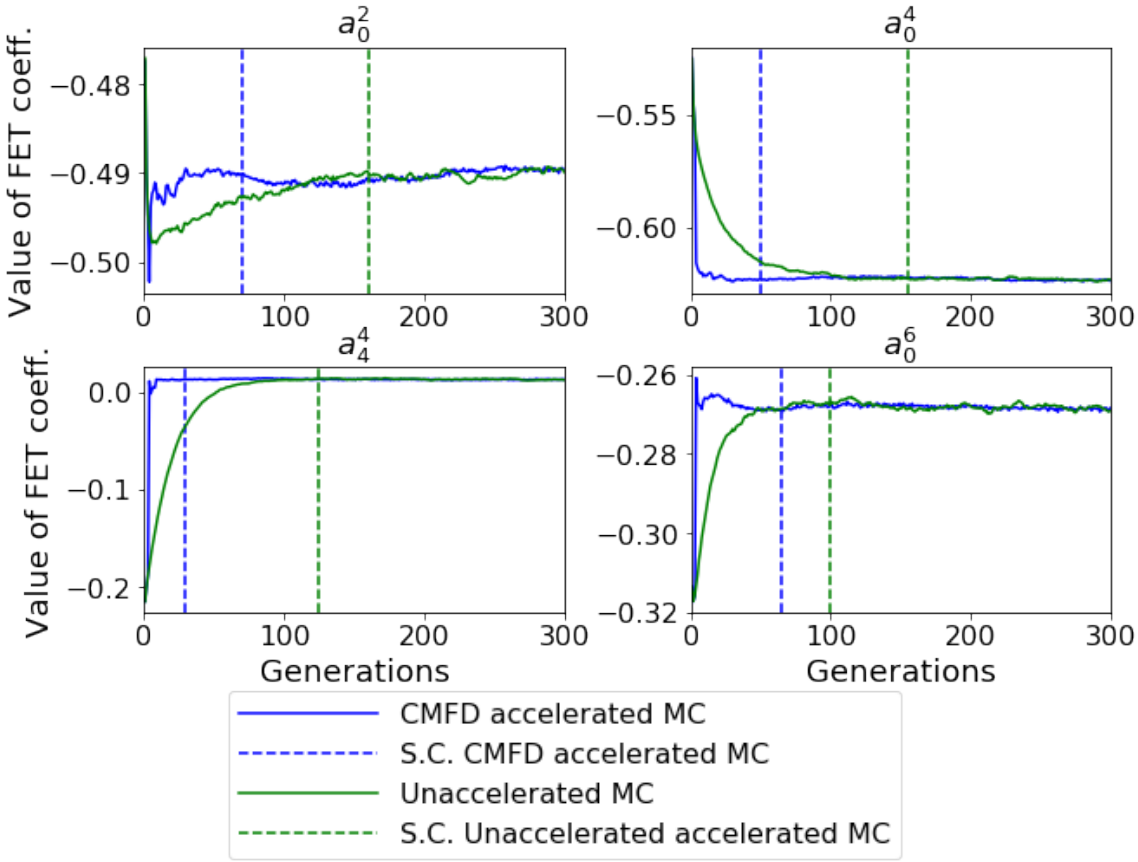


Figure 3-5: Value of representative FET coefficients as a function of generation number for the 2-D BEAVRS problem with CMFD acceleration (blue) and without CMFD acceleration (green). The dashed vertical lines represent the stopping criteria for each particular case.

and the CMFD feedback step. Here, *tally begin* denotes the first generation where CMFD tallying commences in Monte Carlo, while *solver begin* refers to the first generation when CMFD equations are solved and CMFD source information is fed back into the Monte Carlo fission bank weights. Instead, this thesis utilizes weight clipping in order to reduce the fluctuations observed with beginning the CMFD process too quickly. First proposed by Harper in his work coupling CMFD with TH feedback using Monte Carlo tally derivatives, weight clipping limits the amount a particle weight can change in a given CMFD mesh location [39]. A weight clipping amount of 20% is used in this thesis, meaning that at the prolongation step, a particle's weight can change by no more then this amount. All weight adjustment factors are therefore clipped between 1/1.2 and 1.2. This allows the CMFD tallying and feedback process to begin immediately at the first

Table 3.4: CMFD parameters for 2-D BEAVRS problem

Parameter	Value
CMFD tally estimator type	Analog
Number of energy groups	2
Effective downscatter	True
Acceleration map	Used
Reference diffusion parameters	Used
CMFD tally begin	1
CMFD solver begin	1
Weight clipping	0.2
Windowing scheme	Expanding window
Maximum window size	No limit
CMFD mesh size	Quarter assembly mesh
Generation size	10 million
Prolongation source shape	Fixed source

generation, and serves as a damping mechanism to stabilize fluctuations in the fission source at early generations. Figure 3-6 plots the a_1 FET coefficient for the 1-D problem with and without weight clipping, and it can be seen that stationarity in this coefficient is met in a much smoother fashion with the use of weight clipping.

3.2.3.2 Tally Windowing Scheme and Maximum Window Size

Li recommends the use of an expanding window scheme to reduce fluctuations that arise from running CMFD. In such a scheme, tallies from multiple previous generations are accumulated in order to pass into the CMFD solver at each generation, where the motivation for such tally aggregation is to increase the total neutron histories that contribute to the CMFD tallies, which in turn reduces fluctuations in the computed CMFD source distribution. Here, the term *window size* is used to denote the number of previous generations of tally data to compute CMFD cross sections, and in an expanding window scheme, this window size grows by a factor of 2 when the number of simulated generations increases by the same factor. Figure 3-7 illustrates which generations of tally data are used in the CMFD cross section calculation step at each generation, and

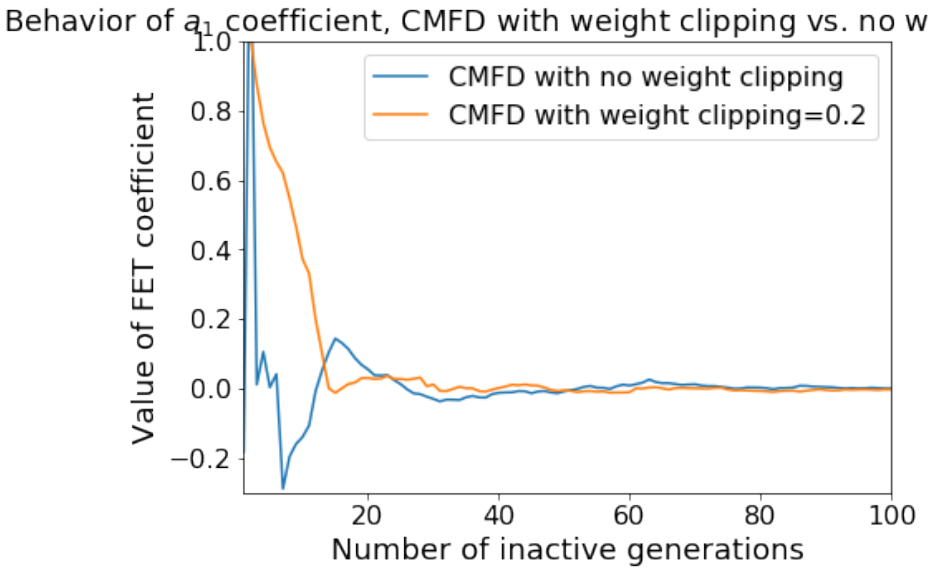


Figure 3-6: Behavior of a_1 FET coefficient for an expanding window scheme with and without weight clipping.

this thesis will be using an expanding window scheme as well.

One topic that was left unexplored in Li's thesis, however, was whether this window size should continue to expand indefinitely throughout the simulation. For example, it can be seen in Figure 3-7 that at generation 30, the tally information from the previous 16 generations are aggregated in order to compute CMFD cross sections. While CMFD cross section data is drawn from 16 times more generations of neutrons, it also introduces much more correlation into the simulation, since the source sites at a single generation are now directly linked to the fission sites from the previous 16 generations. Thus, the term *maximum window size* is introduced to describe the maximum window size that an expanding window scheme is allowed to grow to. Once this maximum is reached, the window size will remain fixed at this maximum level for the duration of the simulation. A sensitivity study on a variety of variable parameters is conducted in Sections 3.3 and 3.4, and the effect of the maximum window size on both the inactive and active generations will be explored in more detail.

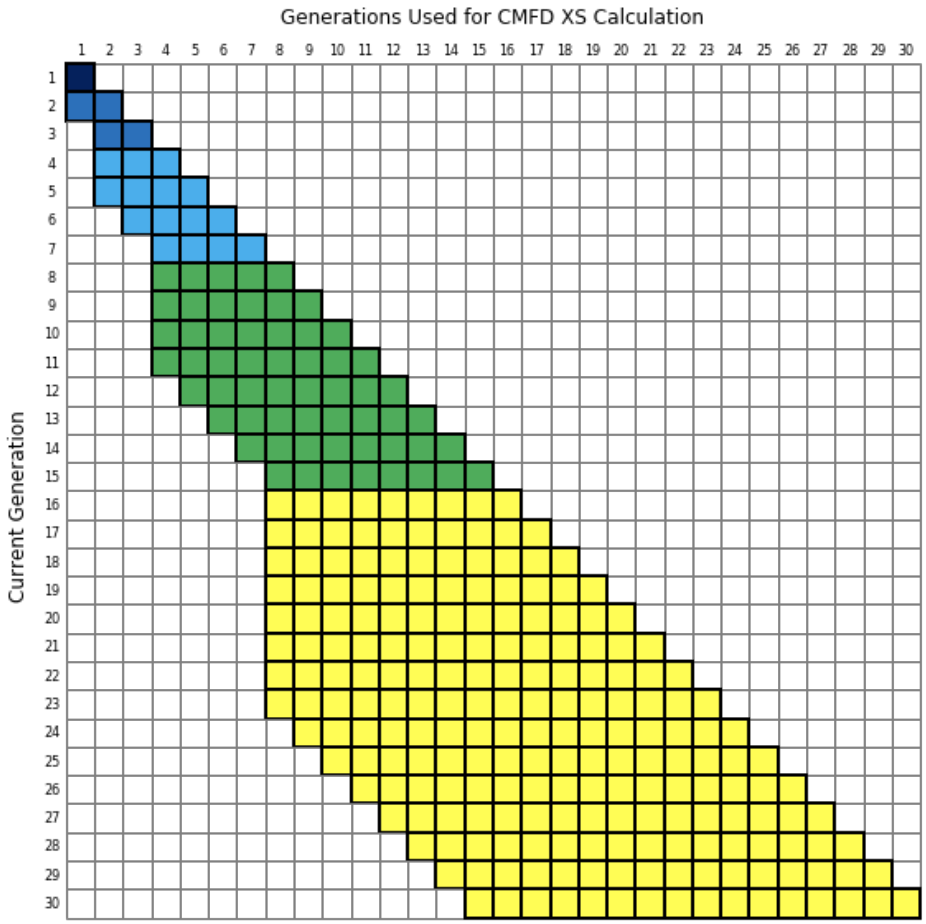


Figure 3-7: Illustration of expanding window scheme for first 30 generations.

3.2.3.3 CMFD Mesh Size

Intuitively it may seem that a finer CMFD mesh will result in a finer shape distribution of computed CMFD tallies and will lead to a finer representation of the underlying CMFD source distribution. However, when generation size is kept fixed, the number of tally events that contribute to each mesh cell will decrease, thus increasing overall variance levels in each mesh location. This interplay will also be explored in Sections 3.3 and 3.4.

One new feature added to OpenMC to improve CMFD runtime is the ability to solve CMFD equations with multiple OpenMP threads. By implementing the CMFD solver as a red-black Gauss-Seidel linear solver, shared memory parallelism can be utilized to reduce solver runtime [71, 72]. In such an algorithm, each gridpoint is ordered in a checkerboard pattern where in one-dimension a gridpoint i is colored red if i is even

and black if odd. Likewise, in two dimensions, each gridpoint (i, j) is colored red if $i + j$ is even and black if $i + j$ is odd. In this manner, all red grid points can be updated in parallel using multiple OpenMP threads since their results are decoupled from the black grid points. In the next iteration, all black grid points are then updated in parallel. Thus, a single Gauss-Seidel update step can be divided into parallel grid point updates of the red cells followed by the black cells.

While increasing the number of OpenMP threads can speed up the runtime of such a parallelized algorithm, the size of the problem must be adequately large enough in order for this to happen. If the size of the problem is not large enough, different threads will start to compete with one another to maintain cache coherency. For small problems, the likelihood of different threads trying to access the same cache line in order to update flux values increases. In such cases, threads will have to update entries in the same cache line in a sequential manner once competing threads have finished their updates. This situation is called *false sharing* and is a significant cause of performance degradation when using OpenMP multithreading [73].

This is the situation that arises when using multiple threads on a CMFD problem with a coarse mesh. Figure 3-8 plots the CMFD execution time per generation as a function of number of OpenMP threads for the 1-D homogeneous problem using a 0.4cm and 20cm CMFD mesh. Likewise, Figure 3-9 plots the CMFD execution per generation for the 2-D BEAVRS problem using an assembly, quarter assembly, and pincell CMFD mesh. It can be seen that CMFD runtime can actually increase if too many threads are used to solve a coarse mesh CMFD problem. Thus, throughout this chapter only a single OpenMP thread is used when running CMFD on a coarse mesh such as 1-D 20cm or 2-D assembly mesh, whereas a higher thread count is used for the 1-D 0.4cm and 2-D quarter assembly mesh. For the 2-D pincell mesh, while the size of the problem results in runtime savings even with 16 threads, the overall CMFD runtime at this level is still over 500 times slower than a single thread assembly mesh solver, so this mesh size is not pursued in the sensitivity studies conducted in Sections 3.3 and 3.4. All runtimes from this section were collected on simulations that were run on a single node of a computer cluster with dual-socket Intel Xeon E5-2650 v4 processors per node and 16 cores per

socket.

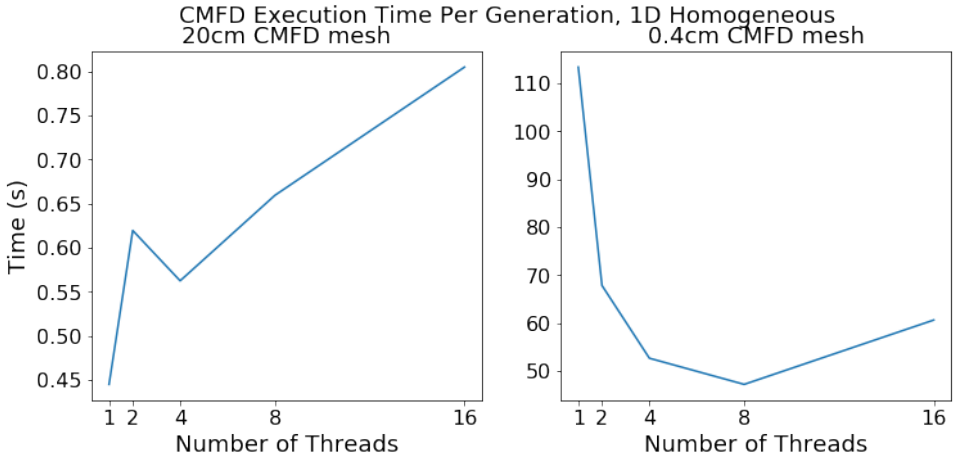


Figure 3-8: Effect of number of OpenMP threads on CMFD execution time per generation for 0.4cm and 20cm mesh size on 1-D test problem.

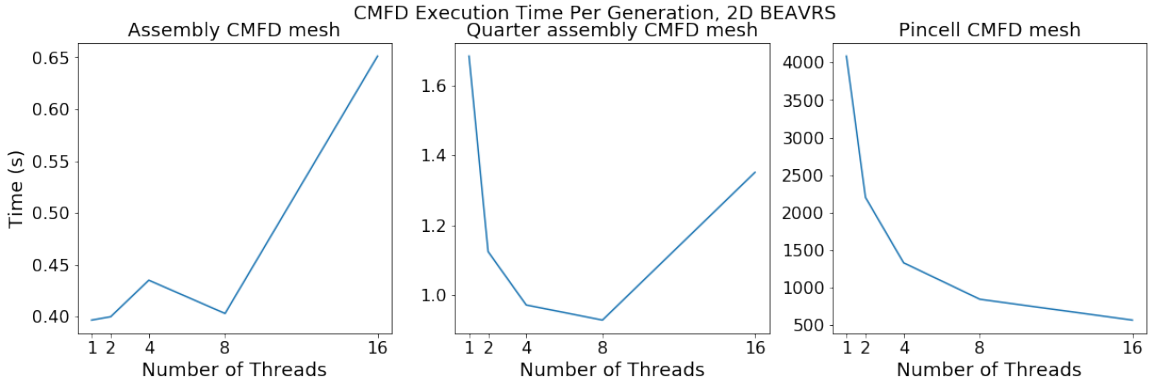


Figure 3-9: Effect of number of OpenMP threads on CMFD execution time per generation for assembly, quarter assembly, and pincell mesh size on 2-D test problem.

3.2.3.4 Generation Size

For the 1-D and 2-D results presented in Section 3.2.2, a fixed generation size of 10 million was used for both cases. While it is a well-known fact that standard deviation in tallied quantities reduces at a rate of $1/\sqrt{N}$ for uncorrelated particles that do not use source acceleration schemes where N is the generation size, Section 3.3 will also vary this parameter to see if this trend holds true for Monte Carlo simulations that utilize CMFD acceleration.

3.2.3.5 Prolongation Source Shape Function

Section 3.1.1.4 presented the equations to compute the weight adjustment factors used to update each particle weight in the fission bank. However, such a prolongation step assumes that the source in each mesh cell is spatially flat and does not assume any shape variation within the mesh cell. This can lead to large approximations especially when a flat shape is assumed over a coarsely defined mesh cell. For deterministic calculations, CMFD source acceleration applied to Method of Characteristics has benefited from assuming a shape profile within each mesh cell, where the coefficients that define this shape are calculated by preserving neutron flux within each cell [74]. In a similar vein, Section 3.4 will look at the impact of assuming a linear shape function within a coarsely defined CMFD mesh cell to see if it improves the quality of the Monte Carlo solution. A full derivation of the equations used to calculate the shape parameters for this linear prolongation step can be found in Appendix C. Linear prolongation is pursued only in a single dimension for the 1-D problem and is not conducted for the 2-D problem.

Given the multitude of CMFD parameters that have been introduced to this point, Table 3.5 summarizes these parameter choices for both 1-D and 2-D sensitivity studies that will be conducted in Sections 3.3 and 3.4. Parameters labeled "Variable" are allowed to vary in these sections, and are followed by the values that they can take on.

FET stationarity is extended to the use of CMFD source acceleration, where the exact savings in terms of generations to stationarity are identified for 1-D and 2-D test problems. CMFD parameters are defined and four of these parameters - generation size, CMFD mesh size, maximum window size, and prolongation source shape - will be varied in ensuing sensitivity studies.

Table 3.5: CMFD parameter choices for Sections 3.3 and 3.4.

Parameter	Value for 1-D Problem	Value for 2-D Problem
CMFD tally estimator type	Analog	Analog
Number of energy groups	1	2
Effective downscatter	N/A	True
Acceleration map	Not used	Used
Reference diffusion parameters	Not used	Used
CMFD tally begin	1	1
CMFD solver begin	1	1
Weight clipping	0.2	0.2
Windowing scheme	Expanding window	Expanding window
Maximum window size	Variable - 8, 16, 32, 64	Variable - 8, 16, 32, 64
CMFD mesh size	Variable - 0.4cm, 20cm	Variable - Assembly, quarter assembly
Generation size	Variable - 1 million, 10 million	Variable - 10 million, 100 million
Prolongation source shape	Variable - Flat shape, linear shape	Flat shape

3.3 CMFD Parameter Sensitivity Study on Monte Carlo Inactive Generations

During the inactive generations, the goal is to simulate enough generations so as to remove any bias in the initial source guess, and at this point no accumulation of the fission source occurs yet. The variance levels in the stationary instantaneous fission source describes the level of statistical noise present in the system, which in turn dictates the error level observed in the first active generation. An instantaneous fission source that varies greatly from generation to generation is more likely to exhibit higher starting error levels in the active generations, while lower starting error levels can be achieved if the variance in the instantaneous fission source is lowered. An example of this is when a larger generation size is used in Monte Carlo simulations. A larger generation size results in an instantaneous fission source that exhibits lower system-level noise levels,

and typically will yield a starting error level that is $1/\sqrt{N}$ times that of a simulation that has a generation size that is N times smaller. Similarly, simulations that utilize source acceleration schemes such as CMFD should also try to reduce variance levels in the underlying fission source, so as to lower starting error levels in the active generations.

For both 1-D and 2-D cases simulated in Section 3.2.2, the expanding window scheme allowed the tally windows to continue to expand exponentially without limit. In doing so, later generations received tally information from many more previous generations in order to compute CMFD cross sections compared to earlier generations, and over time this reduces fluctuations in the fission source during the CMFD feedback step. In order to illustrate this effect, Figure 3-10 plots the a_1 FET coefficient as a function of simulated generations for the 1-D case, where a maximum window size of 8 is imposed for the left plot and a maximum window size of 32 is used for the right plot. For both cases, a 0.4cm CMFD mesh is used. The equilibrium fluctuation levels in both of these plots represent the fluctuations that occur in the stationary instantaneous fission source. For the case with a maximum window size of 8, each of the later generations is only able to use the tallies from the previous 8 generations to pass into the CMFD cross section calculations, while the case with a maximum window size of 32 allows information from the previous 32 generations to be included as part of the cross section calculation process. From these plots, it can be seen that the FET coefficients and hence the instantaneous fission source at a given generation will have fewer fluctuations when a larger maximum window size is achieved.

In order to quantify the exact statistical fluctuations in the fission source, this section defines the standard deviation in the instantaneous fission source as:

$$s[X_n] = \frac{1}{M}\sum_{m=1}^{M}\left[\frac{\sqrt{\frac{1}{G-1}\sum_{g=n}^{n+G}\left(X_{g,m}-\overline{X}_{n:n+G,m}\right)^2}}{\overline{X}_{n:n+G,m}}\right] \tag{3.6}$$

which represents the average relative fluctuation per mesh cell computed over a moving window of $G = 100$ generations for a single Monte Carlo trial. Here, $X_{g,m}$ represents the fission source at generation g and mesh location m and $\overline{X}_{n:n+G,m}$ represents the average fission source value at mesh location m from generation n to $n+G$. $s[X_n]$ is

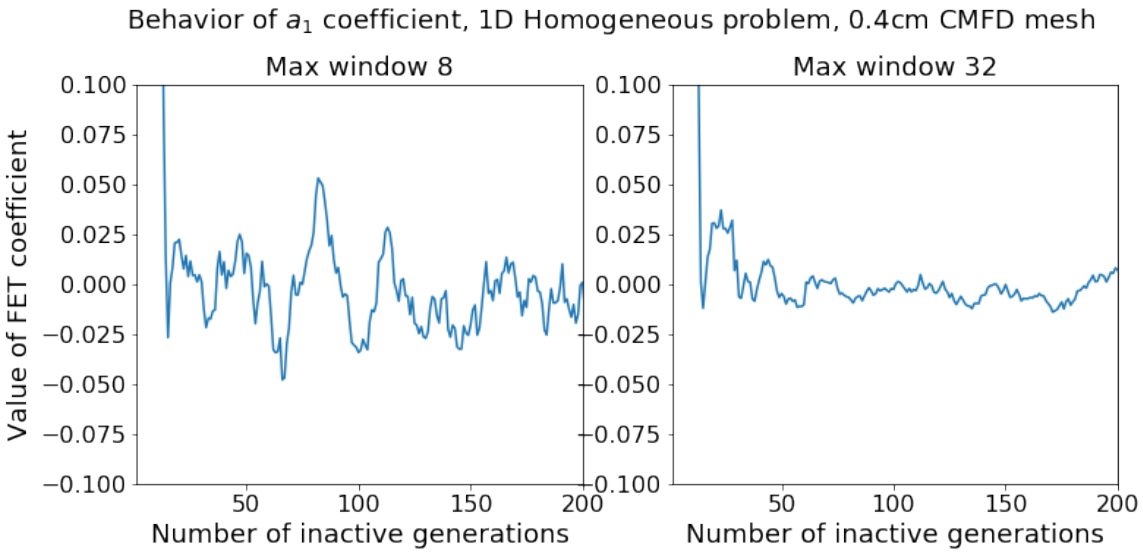


Figure 3-10: Behavior of a_1 FET coefficient for an expanding window scheme with a maximum window size of 8 (left) and 32 (right).

the mesh-averaged standard deviation in the instantaneous fission source at generation n and serves as a proxy to the system-level statistical noise in the instantaneous fission source. The mesh M over which the standard deviation is computed is a fine mesh of 0.4cm for the 1-D case and a pincell mesh for the 2-D case, and this is done in order to neglect any correlation effects that may arise when trying to compute source fluctuations over successive generations in a single seed simulation. Here, it is assumed that if a fine enough mesh is used to compute the fission source, then source values in a given mesh cell will behave independently from generation to generation [28].

3.3.1 Effect of Window Size on Instantaneous Fission Source Standard Deviation

Figure 3-11 plots the standard deviation in the instantaneous fission source as a function of window size for the CMFD accelerated problem with a 0.4cm CMFD mesh. Figure 3-11 also shows the standard deviation of a problem that doesn't use CMFD acceleration[2]. In line with the behavior of Figure 3-10, this plot shows that fluctuations in the fission source decrease as the maximum window size increases. However, it also shows that for

[2]Since we are interested in the equilibrium fission source standard deviation levels and not the generations to stationarity, a uniform starting source is used for the 1-D problem instead of an offset starting source.

standard deviation levels between CMFD-accelerated and un-accelerated simulations to match, a relatively large window size of 64 needs to be achieved. This is a result that is often overlooked when measuring the impact of CMFD on the inactive generations. While CMFD may reduce the number of generations to stationarity, unless a windowing scheme that uses a large window size is used, often times CMFD can introduce larger noise levels to the underlying fission source.

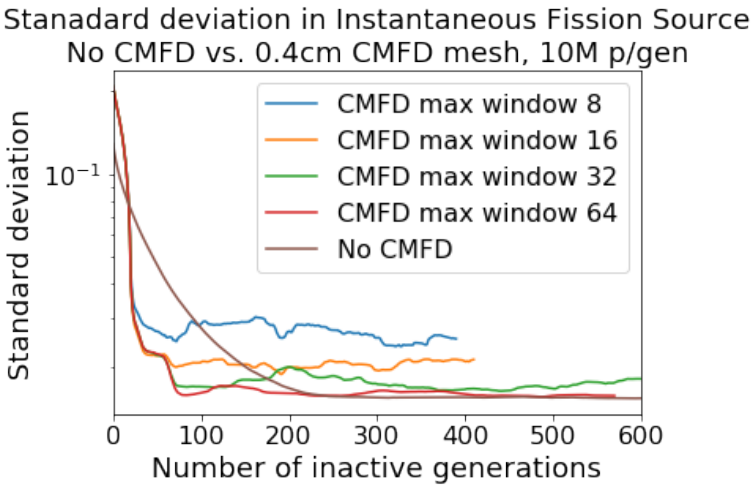


Figure 3-11: Standard deviation in instantaneous fission source - No CMFD vs. 0.4cm CMFD mesh with 10 million particles per generation and varying maximum window size.

As the maximum window size continues to increase indefinitely, the fission source variance levels for a CMFD-accelerated simulation actually overtake that of the un-accelerated run. Figure 3-12 plots the standard deviation in the instantaneous fission source for both the same CMFD-accelerated and un-accelerated cases as Figure 3-11 but this time the maximum window size is allowed to grow to 512. It can be seen that a lower standard deviation level can be achieved when the window size is this large. Typically, however, this is impractical in the inactive generations since at least 512 generations are required before such a window size can be achieved, and this is well beyond the cutoff point between the inactive and active generations when running CMFD for the 1-D case.

At this point, the term *CMFD penalty factor* is introduced. This factor represents the minimum window size required in an expanding window scheme for a given generation

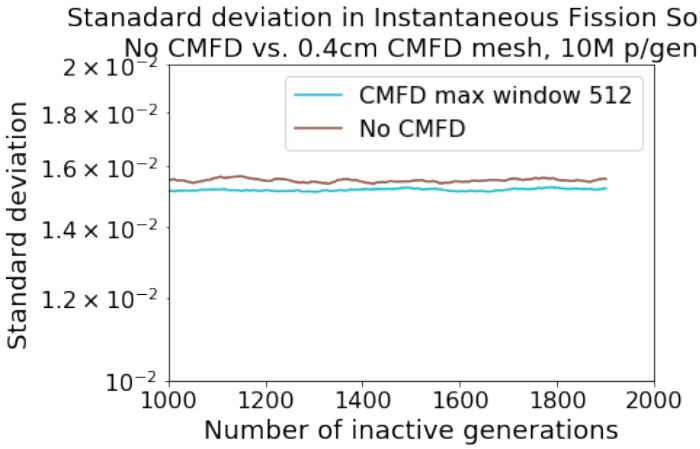


Figure 3-12: Standard deviation in instantaneous fission source - No CMFD vs. 0.4cm CMFD mesh with 10 million particles per generation and maximum window size 512.

size so that the standard deviation in the underlying fission source of a CMFD-accelerated simulation is comparable to that of an un-accelerated scheme. According to Figure 3-11, for the 1-D problem with a 0.4cm CMFD mesh and generation size 10 million, the CMFD penalty factor is 64. If this minimum window size is not met, then variance levels in the underlying instantaneous fission source will be higher when CMFD is used.

3.3.2 Effect of Generation Size on Instantaneous Fission Source Standard Deviation

Section 3.3.1 introduced the notion of the CMFD penalty factor and described how a maximum window size of at least 64 was needed in order for CMFD variance levels to match that of an un-accelerated run for the 1-D case with 0.4cm CMFD mesh and generation size of 64. This section keeps the window size fixed and varies the generation size to see if the CMFD penalty factor is independent of the generation size. Figure 3-13 plots the standard deviation in the instantaneous fission source where the generation size is varied between 1.25 million, 2.5 million, 5 million, and 10 million. For each of these cases, the CMFD-accelerated and un-accelerated simulations possess the same fluctuation levels when the generation size is the same and the maximum window size is fixed at 64, which seems to suggest that the CMFD penalty factor remains the same regardless of generation size.

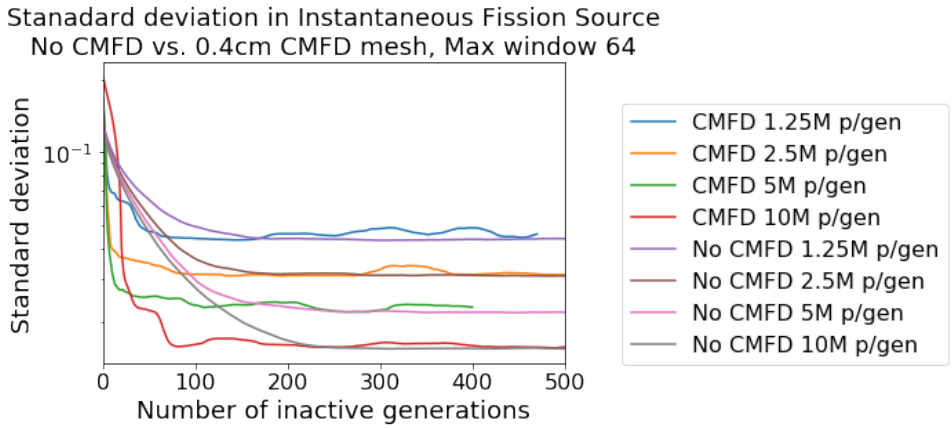


Figure 3-13: Standard deviation in instantaneous fission source - No CMFD vs. 0.4cm CMFD mesh with maximum window size 64 and varying generation size.

While it has been shown that increasing generation size and maximum window size both result in the reduction of instantaneous fission source variance, which parameter has a larger effect? Figure 3-14 plots the un-accelerated case and CMFD accelerated case where the CMFD penalty factor of 64 is met. The green line shows the case where the window size is reduced by half but the generation size is increased by a factor of two, and similarly the red line shows the case where window size is reduced by a factor of 4 but the generation size is increased by this same factor. Keeping the product of generation size and maximum window size constant between the orange, green, and red lines, these results indicate that increasing generation size has a larger effect on reducing standard deviation levels in the instantaneous fission source.This is likely due to the fact that increasing generation size decouples particles that belong to the same generation, while increasing the window size adds additional correlation to the fission source, which leads to a situation of diminishing returns with respect to reduced standard deviation levels.

3.3.3 Effect of Mesh Size on Instantaneous Fission Source Standard Deviation

All results in Section 3.3 so far have been shown for the 1-D problem with a 0.4cm CMFD mesh. This section analyzes the use of a coarser 20cm CMFD mesh and also

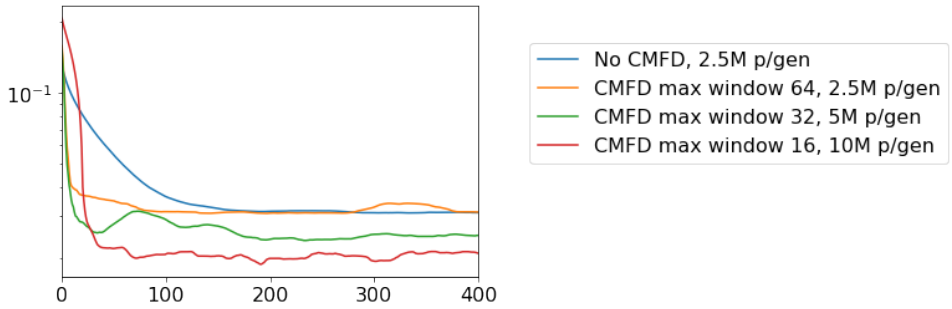


Figure 3-14: Standard deviation in instantaneous fission source - No CMFD vs. 0.4cm CMFD mesh with product of window size and generation size kept constant.

presents results for the 2-D BEAVRS case. Figure 3-15 plots the standard deviation in the instantaneous fission source for a variety of maximum window sizes and generation sizes using a 0.4cm CMFD mesh, while Figure 3-16 plots the same standard deviation levels using a 20cm CMFD mesh. It can be seen for the 20cm CMFD mesh, the CMFD penalty factor stays the same at 64 and is independent of generation size, and yields roughly the same standard deviation levels as the 0.4cm case. In this light, given the extra computational burden of computing a fine-mesh CMFD solution, it is prudent to use a coarse mesh CMFD solution in order to reduce runtime in the inactive generations. Moreover, fission source variance levels are not affected significantly with the use of a coarse mesh CMFD solution either. This is because a fine mesh solution has a more detailed shape profile, but the values within each mesh cell will exhibit higher noise levels as fewer neutron events contribute to each tally region.

Similar trends emerge for the 2-D BEAVRS case as well. Figures 3-17 and 3-18 show analogous plots to those in Figure 3-15, using assembly and quarter assembly CMFD meshes respectively for the 2-D case. The type of CMFD mesh has minimal impact on the instantaneous fission source, while the CMFD penalty factor now seems to be lower than the 1-D case, where a maximum window size between 32 and 64 yields the same standard deviation levels as the un-accelerated 2-D BEAVRS case. Once again, the CMFD penalty factor is also independent of the generation size.

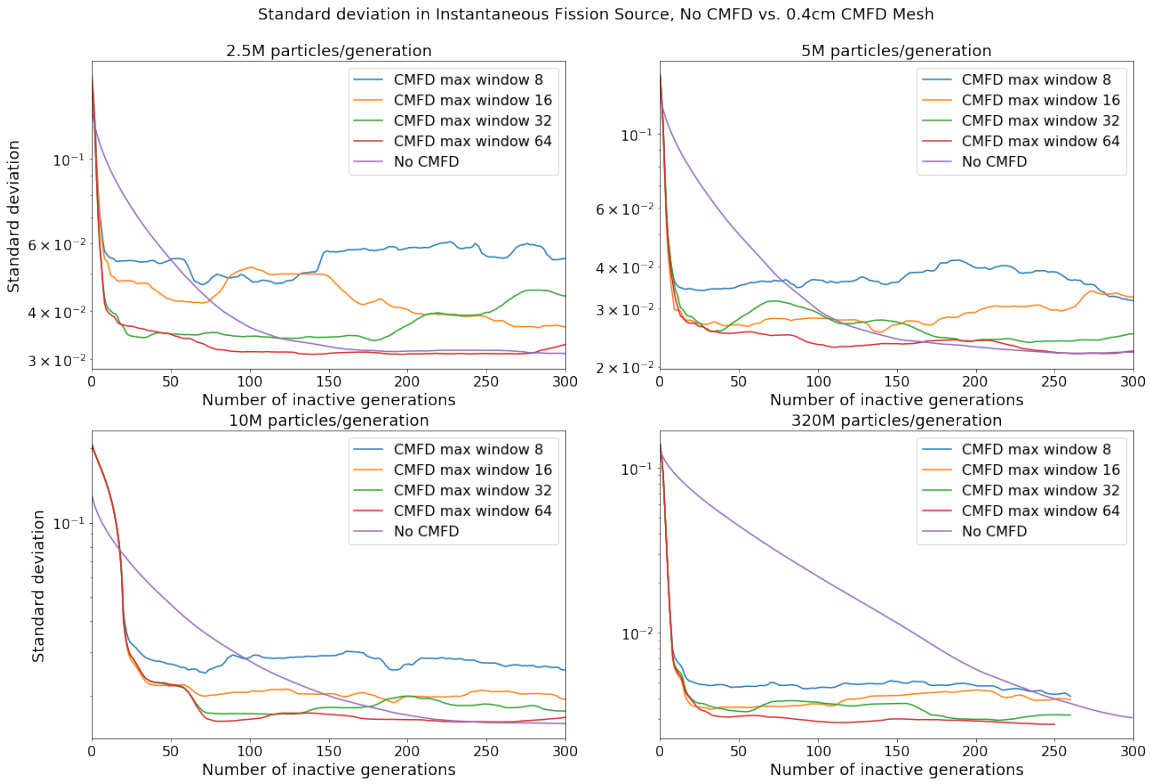


Figure 3-15: Standard deviation in instantaneous fission source - No CMFD vs. 0.4cm CMFD mesh with varying window size and generation size.

A fine-mesh CMFD solution and coarse-mesh CMFD solution exhibit similar levels of standard deviation in the instantaneous fission source when generation size is kept fixed. Furthermore, the maximum window size in a CMFD-based solution needs to be as large as the CMFD penalty factor in order for these standard deviation levels to be on par between CMFD and un-accelerated solutions. The CMFD penalty factor is independent of generation size.

3.4 CMFD Parameter Sensitivity Study on Monte Carlo Active Generations

While Section 3.3 looked at the inactive generations to determine the optimal CMFD parameters to minimize variance in instantaneous fission source variance levels, this section will look at the active generations to refine those parameter choices. The biggest

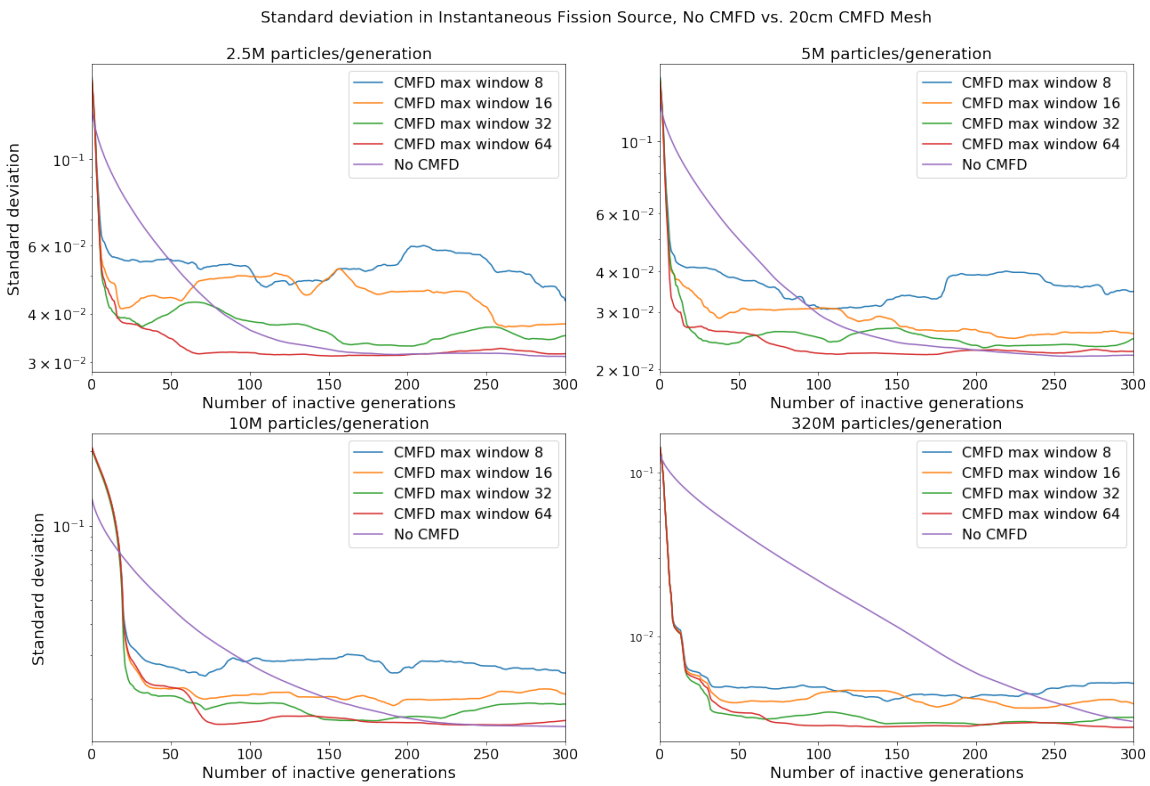


Figure 3-16: Standard deviation in instantaneous fission source - No CMFD vs. 20cm CMFD mesh with varying window size and generation size.

takeaway from Section 3.3 was that the maximum window size should be as large as possible in order to minimize the standard deviation in the fission source and in doing so, allows the fluctuations to match that of the un-accelerated case. Here, the impact of maximum window size, generation size, CMFD mesh size, and prolongation source shape on the active generations will be explored.

Once the active generations begin, the instantaneous fission source is accumulated so as to compute the fission source with greater confidence. The variable $Y_{n,m,p}$ denotes the accumulated fission source at the n-th active generation, m-th mesh cell, and p-th independent trial, and is defined as:

$$Y_{n,m,p} = \frac{1}{n}\sum_{i=1}^{n} X_{i,m,p} \tag{3.7}$$

where $X_{i,m,p}$ refers to the instantaneous fission source at active generation i, mesh m, and trial p. The cutoff point for the first active generation is no longer a fixed value like what is typically done in active generation analysis, but is determined using the

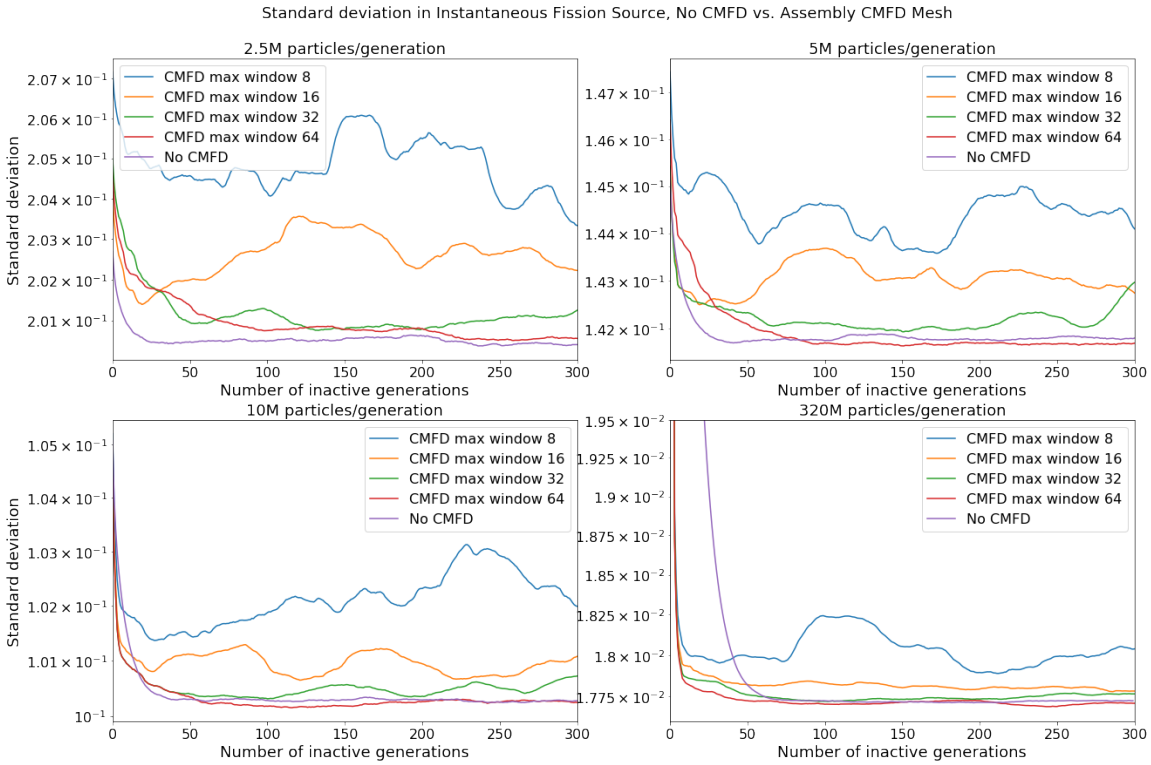


Figure 3-17: Standard deviation in instantaneous fission source - No CMFD vs. Assembly CMFD mesh with varying window size and generation size for the 2-D BEAVRS problem.

FET-based stopping criteria for each independent trial.

In this section, the accuracy of the accumulated fission source distribution is assessed by comparing it to a reference value, and the RMS error heuristic is used to describe the trial-averaged, mesh-averaged error in the source distribution, defined as:

$$\epsilon_n^{\text{RMS}} = \sqrt{\frac{1}{M}\sum_{m=1}^{M}\left(\frac{\sum_{p=1}^{P}\left(Y_{n,m,p}\right)/P - Y_m^{\text{ref}}}{Y_m^{\text{ref}}}\right)^2} * 100\% \tag{3.8}$$

where Y_m^{ref} represents the reference accumulated source distribution in mesh m[3]. Discussion of how these reference solutions are computed for each test case is explained in more detail in Section 3.4.1.

During the active generations, one of the main obstacles to ideal $1/\sqrt{N}$ convergence rates in the RMS error distribution is the emergence of correlation effects, since the source sites of each additional generation are now directly linked to the fission sites

[3]For the RMS error plots that will be shown in this section, the accumulated fission source $Y_{n,m,p}$ is computed with $P = 1$ trial.

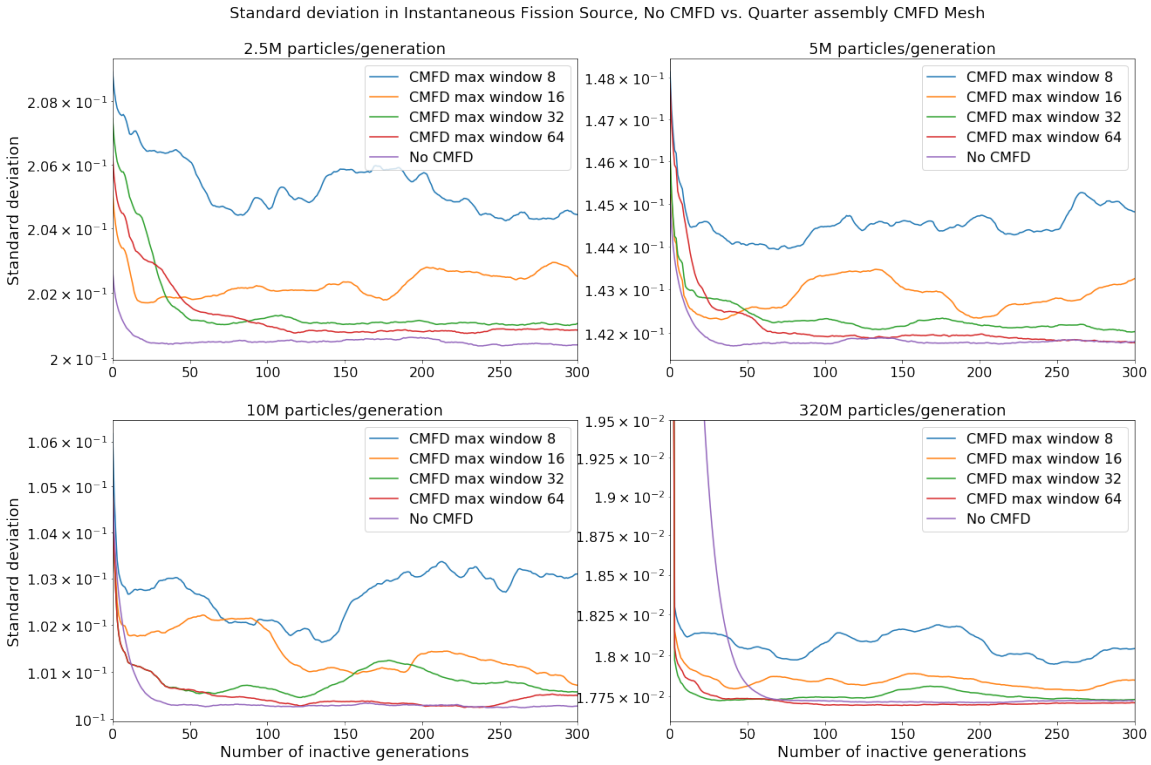


Figure 3-18: Standard deviation in instantaneous fission source - No CMFD vs. Quarter assembly CMFD mesh with varying window size and generation size for the 2-D BEAVRS problem.

of all previous active generations. Thus, the fission source sites that comprise the entire sample of tally data to compute sample statistics can no longer be assumed to be independent. Unless this correlation is accounted for, the *apparent standard deviation* computed from the sample standard deviation of a single trial will under-predict the *real standard deviation* [29, 30, 75]. Instead, the real standard deviation can be computed as the mesh-averaged sample standard deviation across independent trials of Monte Carlo simulations, and is defined as:

$$s[Y_n] = \frac{1}{M}\sum_{m=1}^{M}\sqrt{\frac{1}{P-1}\sum_{p=1}^{P}(Y_{n,m,p} - \overline{Y}_{n,m})^2} \tag{3.9}$$

Here, real standard deviation can be calculated without a reference source distribution, and it can be shown that the convergence rate of the real standard deviation serves as a proxy to assess the convergence rate in RMS error rates for cases where a reference source distribution is unavailable [1, 28]. Real standard deviation levels will

be calculated and examined in Section 3.4.4 for the 2-D BEAVRS problem.

3.4.1 Reference Solutions for Test Problems

In order to calculate RMS error to judge the quality of a simulation's accumulated fission source, reference solutions to the 1-D and 2-D test problems are required. For the 1-D problem, the reference solution is computed as the average over 100 independent trials and over 10,000 active generations per trial without CMFD source acceleration. In order to also assess the impact of the mesh size on the reference solution, the reference fission source is computed over different mesh sizes as well. The normalized fission source for the 1-D problem is collected over both a 0.4cm and 20cm mesh - corresponding to a similar mesh size to that of a PWR pincell and assembly respectively. In this section, the term *RMS error mesh* is used to define the mesh M for the calculation of RMS error in Equation 3.8, and the impact of RMS error mesh on active generation RMS will also be explored.

The normalized average fission source for the 1-D reference solution computed over these two mesh sizes is shown in Figure 3-19. The second column plots the real standard deviation computed over the 100 independent trials. The third column plots the relative error in the fission source for each mesh location, which is defined as the real standard deviation divided by the average. These results show that regions of low fission source values generally have low standard deviation levels as well, but if this standard deviation is normalized to the average value for that mesh cell, then regions of low fission source values have higher error levels. This is consistent with the idea that fewer neutron histories frequent these low-flux regions and thus will possess higher relative variance levels compared to high-flux regions.

For the 2-D problem, two reference solutions were generated - one with CMFD acceleration and one without. Both of these solutions are computed as the average over 100 independent trials and over 150 active generations, and the CMFD-accelerated solution utilizes a quarter assembly CMFD mesh. The reference solutions are collected over a pincell mesh and an assembly mesh, and these reference solutions are plotted in

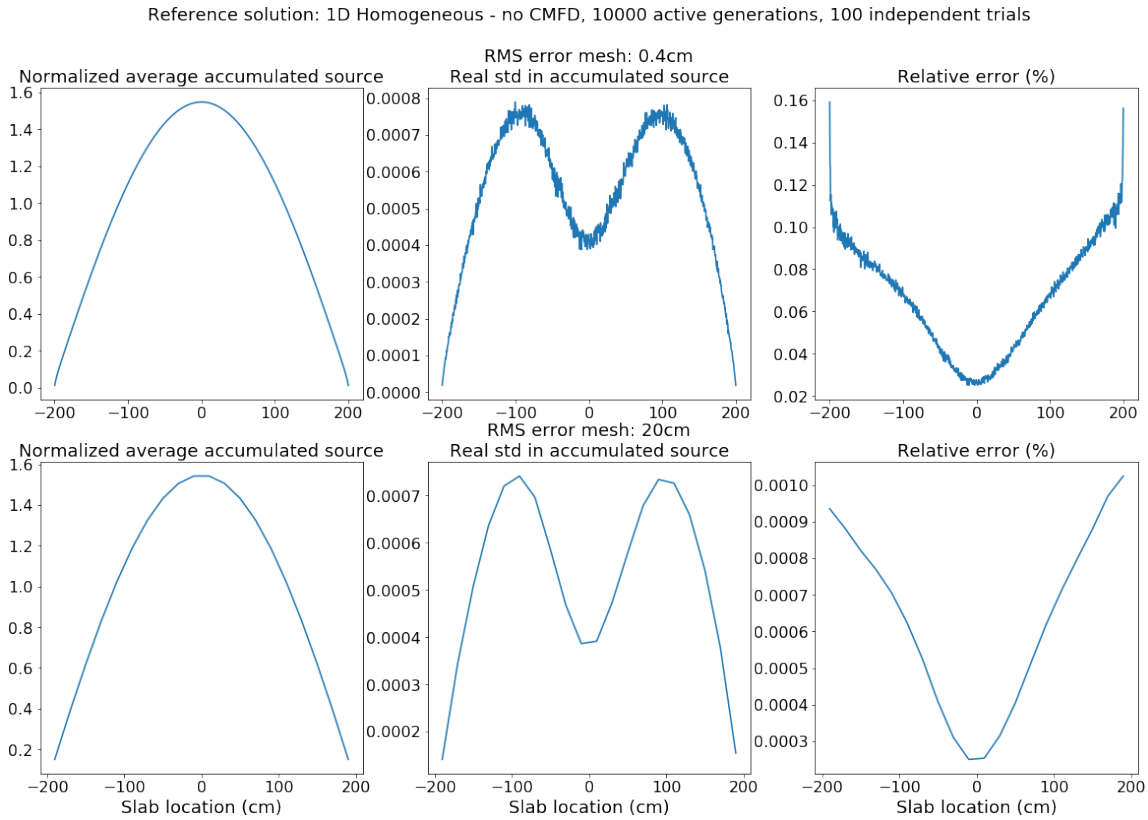


Figure 3-19: Reference solution for the 1-D homogeneous test problem. The top row plots the reference using a 0.4cm RMS error mesh while the bottom row uses a 20cm mesh. The first column plots the normalized average fission source, the second column plots the relative standard deviation for each mesh location, and the third column plots the relative error (standard deviation divided by the average, expressed as a percent).

Figures 3-20 and 3-21 for the un-accelerated and CMFD-accelerated cases respectively. Similar to the 1-D case, the relative error in the fission source shows that the regions with low source values have relatively higher standard deviation levels as well. Also, looking at the areas with higher relative error values, the distribution of errors is tilted for the un-accelerated case compared to the CMFD-accelerated case, so it seems as though a CMFD-based solution is able to remove variations in the low-order radial modes of the solution more effectively than without a source-acceleration scheme. Ultimately, the 2-D reference solution for this section was chosen as an eighth-core octant symmetric solution of the un-accelerated case, but it was found that the results in the active generations do not vary greatly based on whether an un-accelerated, CMFD-accelerated, or octant-symmetric variant of these cases is chosen as the reference solution.

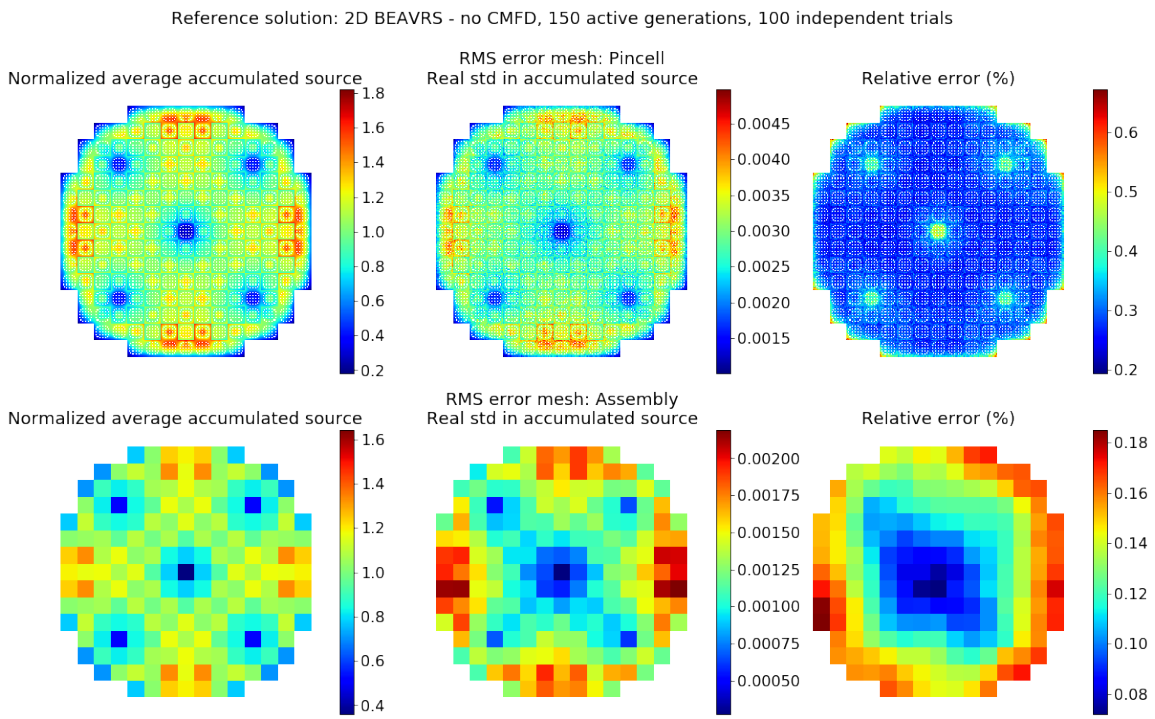


Figure 3-20: Reference solution for the 2-D BEAVRS test problem computed without CMFD acceleration. The top row plots the reference using a pincell RMS mesh while the bottom row uses an assembly mesh. The first column plots the normalized average fission source, the second column plots the relative standard deviation for each mesh location, and the third column plots the relative error (standard deviation divided by the average, expressed as a percent).

3.4.2 1-D Results - RMS Error in Accumulated Fission Source with Fine RMS Mesh

Figure 3-22 plots the RMS error in the accumulated fission source as a function of active generations for the 1-D un-accelerated case along with the CMFD-accelerated case with a 0.4cm CMFD mesh. A generation size of 1 million and a 0.4cm RMS error mesh is used, and the results are shown for a variety of maximum window sizes. The red line shows the ideal $1/\sqrt{N}$ convergence line, and the right plot zooms in on the first 5 active generations of the left plot. It can be seen that most of the trendlines behave similarly, where the convergence rates are in accordance with the $1/\sqrt{N}$ rate, and this is because the RMS error is computed over a fine mesh. At later active generations, there seems to be slight deviations from the idealized convergence rates. At initial active generations, there is also a slight decrease in RMS levels when a larger maximum window size is

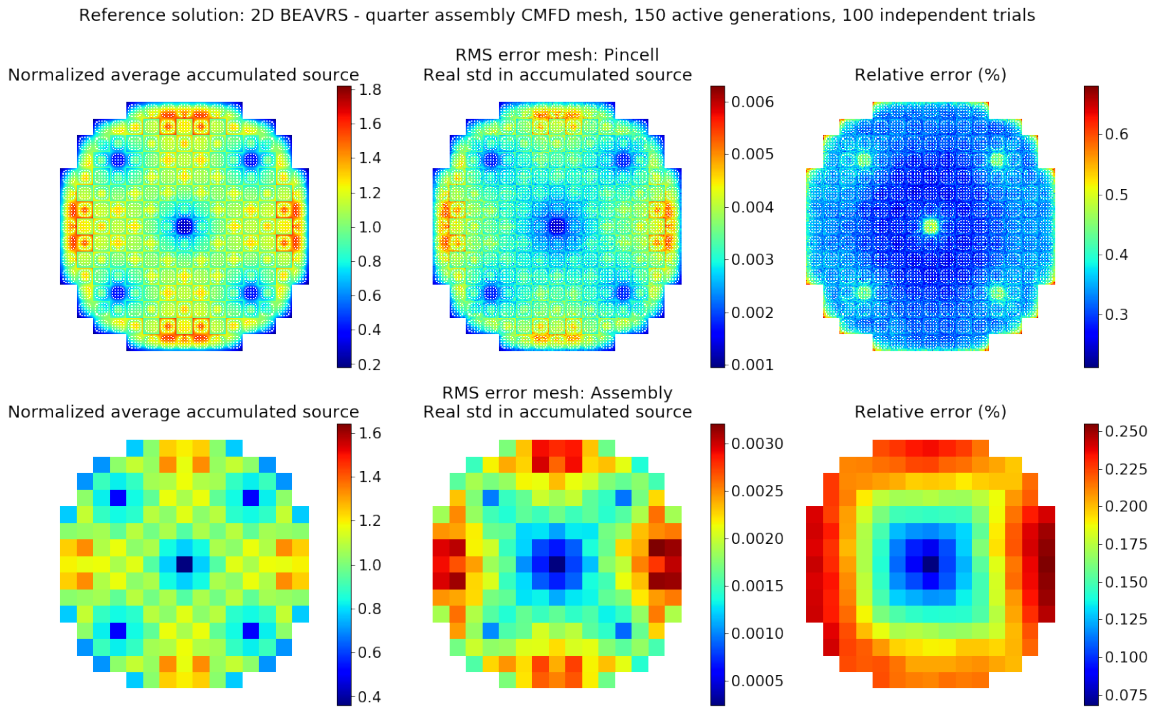


Figure 3-21: Reference solution for the 2-D BEAVRS test problem computed with CMFD acceleration using a quarter assembly CMFD mesh. The top row plots the reference using a pincell RMS mesh while the bottom row uses an assembly mesh. The first column plots the normalized average fission source, the second column plots the relative standard deviation for each mesh location, and the third column plots the relative error (standard deviation divided by the average, expressed as a percent).

used. This is consistent with the idea that initial RMS error levels are dictated by the standard deviation levels in the instantaneous fission source, which were shown to be lower when larger maximum window sizes were achieved.

Figure 3-23 plots similar RMS error plots as Figure 3-22 but uses a 20cm acceleration mesh instead. Once again, when a fine RMS error mesh is used, convergence rates conform to ideal $1/\sqrt{N}$ rates, and there is a slightly lower starting RMS error when a larger maximum window size is used. There does not seem to be a discernible difference between using a 0.4cm or 20cm CMFD mesh on RMS error convergence rates with a 0.4cm RMS error mesh.

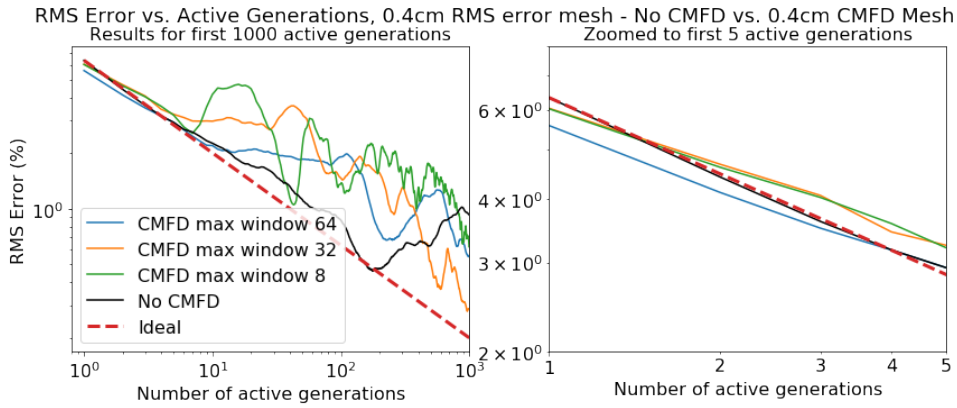


Figure 3-22: RMS error as a function of number of active generations - No CMFD vs. 0.4cm CMFD mesh with varying window size for 1-D problem. Accumulated fission source is computed with a generation size of 1 million and RMS error mesh of 0.4cm. RMS error is shown for the first 1000 active generations (left) and zoomed into the first 5 active generations (right).

3.4.3 1-D Results - RMS Error in Accumulated Fission Source with Coarse RMS Mesh

When a coarser 20cm RMS error mesh is used, convergence rate degradation from correlation effects begins to appear. Figure 3-24 plots the RMS error using a 20cm RMS error mesh for the un-accelerated and CMFD-accelerated case with a 0.4cm acceleration mesh. Here, the initial RMS levels can vary much more based on the maximum window size, where once again a larger window size results in the lowest RMS error. The discrepancy between the ideal red line and the other trendlines illustrates the deviation from $1/\sqrt{N}$ convergence rates in Monte Carlo simulations that occurs due to correlation effects. When a coarse RMS error mesh is used, fission source sites within a mesh cell become more closely linked from generation to generation, and thus the accumulation of source sites over larger spatial regions results in tallied quantities that are no longer independent of each other. In the limit of a large number of active generations, however, RMS errors seem to reach the same levels regardless of window size. This suggests that a larger maximum window size may be able to reduce initial error levels, but the larger window size could also be introducing additional correlation effects due to the fact that fission sites are now dependent on the CMFD feedback steps from a larger number of previous generations. This ultimately reduces convergence rates even further and

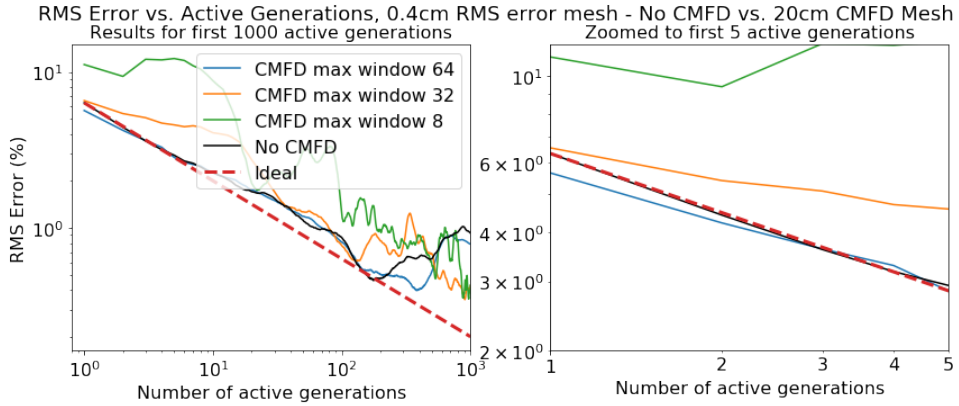


Figure 3-23: RMS error as a function of number of active generations - No CMFD vs. 20cm CMFD mesh with varying window size for 1-D problem. Accumulated fission source is computed with a generation size of 1 million and RMS error mesh of 0.4cm. RMS error is shown for the first 1000 active generations (left) and zoomed into the first 5 active generations (right).

therefore error levels can be overtaken by simulations that utilize a smaller maximum window size.

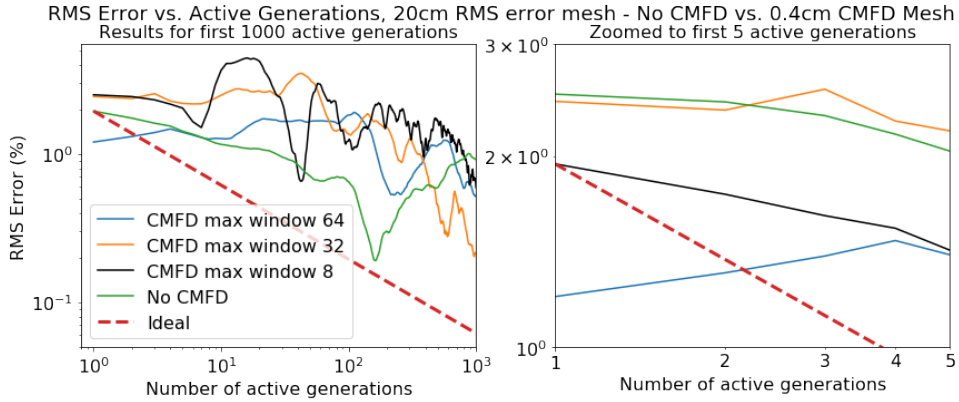


Figure 3-24: RMS error as a function of number of active generations - No CMFD vs. 0.4cm CMFD mesh with varying window size for 1-D problem. Accumulated fission source is computed with a generation size of 1 million and RMS error mesh of 20cm. RMS error is shown for the first 1000 active generations (left) and zoomed into the first 5 active generations (right).

It should be noted here that correlation effects exist even when a fine RMS error mesh is used. Simulations that utilize a fine RMS mesh achieve ideal $1/\sqrt{N}$ convergence rates but begin at a higher starting error level compared to a coarse RMS error mesh. In the limit of a very large number of active generations, fine RMS error mesh simulations slowly deviate from ideal $1\sqrt{N}$ convergence rates and begin to approach the same error

levels as the coarse RMS error mesh case. This can be seen when focusing on the region between 100 and 1000 active generations in Figures 3-22 and 3-24, where terminal RMS error levels reach similar values regardless of RMS error mesh size. A closer examination of this behavior can be found in Appendix D.

Finally, Figure 3-25 plots the RMS error with a 20cm RMS error mesh for the unaccelerated case and CMFD accelerated case with a 20cm CMFD mesh. A large discrepancy can be seen between using a maximum window size of 64 and using a lower maximum window size and could be due to the fact that a flat source approximation is used over a coarse mesh cell during the CMFD prolongation step. Thus, a CMFD prolongation step that utlizes a linear source approximation, denoted as *lpCMFD* for *linear prolongation CMFD*, is plotted in Figure 3-25 as well. It can be seen that linear prolongation CMFD with a smaller window size of 32 yields similar starting RMS error levels when compared to flat source prolongation CMFD with a larger window size. This suggests that assuming a linear shape within the CMFD source distribution for a coarse mesh cell helps to reduce error levels in the active generations.

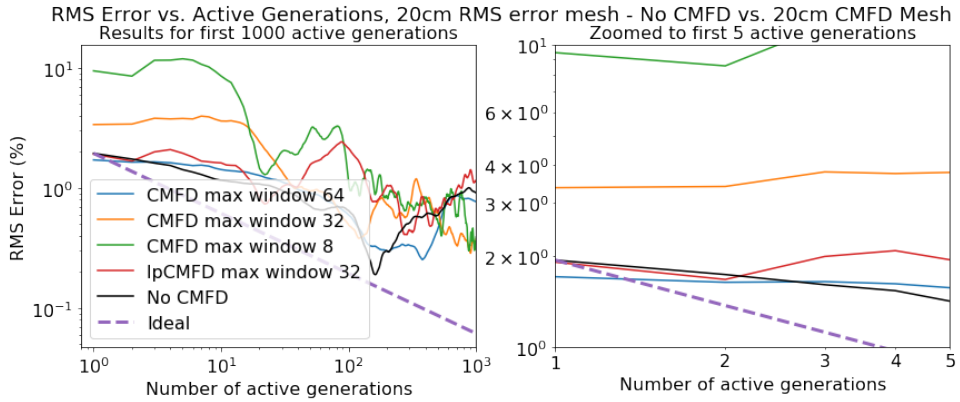


Figure 3-25: RMS error as a function of number of active generations - No CMFD vs. 20cm CMFD mesh with varying window size for 1-D problem. Accumulated fission source is computed with a generation size of 1 million and RMS error mesh of 20cm. RMS error is shown for the first 1000 active generations (left) and zoomed into the first 5 active generations (right).

When increasing the generation size from 1 million to 10 million, similar trends in active generation convergence rates also arise. Increasing the generation size merely shifts the starting RMS error level by a factor of $1/\sqrt{N}$. However, as the number of active generations increases, correlation effects continue to dominate when a coarse

RMS error mesh is used. Analogous plots for Figures 3-22, 3-23, 3-24, and 3-25 with a generation size of 10 million can be found in Appendix E.

3.4.4 2-D Results - RMS Error and Real Standard Deviation in Accumulated Fission Source

For the 2-D BEAVRS case, RMS error levels using a coarse assembly-sized RMS error mesh are used since typical Monte Carlo reactor simulations are concerned with convergence behavior in the regime where correlation effects dominate the aggregation of tallied quantities. Figure 3-26 plots the RMS error as a function of number of active generations for a single seed trial of un-accelerated Monte Carlo and a CMFD-accelerated Monte Carlo with an assembly-sized CMFD mesh. Both cases use a generation size of 10 million, and once again, the discrepancy between the RMS levels and the ideal $1/\sqrt{N}$ trendline illustrates the diminished convergence rates that result from inter-generational correlation. These results are also in line with the 1-D case, where a larger window size yields lower starting RMS error levels. However, as the number of active generations increases, the difference in error levels between the smaller window and larger window cases starts to decrease. Figure 3-27 shows a similar plot where a quarter assembly CMFD mesh is used instead of an assembly-size CMFD mesh, and once again there is no discernible advantage over using a finer CMFD mesh in the active generations for the 2-D case.

For the un-accelerated case and CMFD-accelerated case with a quarter assembly CMFD mesh, 100 independent trials of 100 active generations were also conducted. This allows for the calculation of real standard deviation presented in Equation 3.9 as a function of active generations, which is shown in Figure 3-28. The mesh over which real standard deviation is computed is assembly-sized, and each trial has a generation size of 10 million. These results are in accordance with Figure 3-27, where starting standard deviation levels are lower when a larger window size is used, but convergence rates slow down for simulations with larger window sizes as a larger number of active generations are simulated. The exact numerical characterization of correlation effects

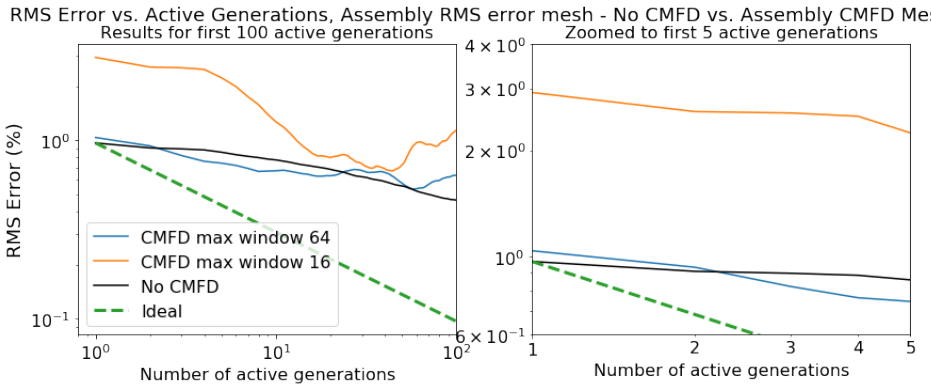


Figure 3-26: RMS error as a function of number of active generations - No CMFD vs. Assembly CMFD mesh with varying window size for 2-D problem. Accumulated fission source is computed with a generation size of 10 million and assembly-sized RMS error mesh. RMS error is shown for the first 100 active generations (left) and zoomed into the first 5 active generations (right).

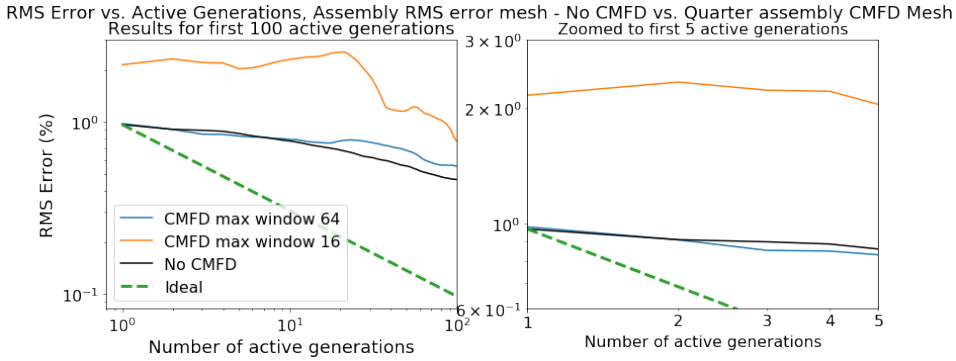


Figure 3-27: RMS error as a function of number of active generations - No CMFD vs. Quarter assembly CMFD mesh with varying window size for 2-D problem. Accumulated fission source is computed with a generation size of 10 million and assembly-sized RMS error mesh. RMS error is shown for the first 100 active generations (left) and zoomed into the first 5 active generations (right).

with CMFD acceleration is not explored in this thesis, but these trends are in line with the results shown by previous work in this domain [1].

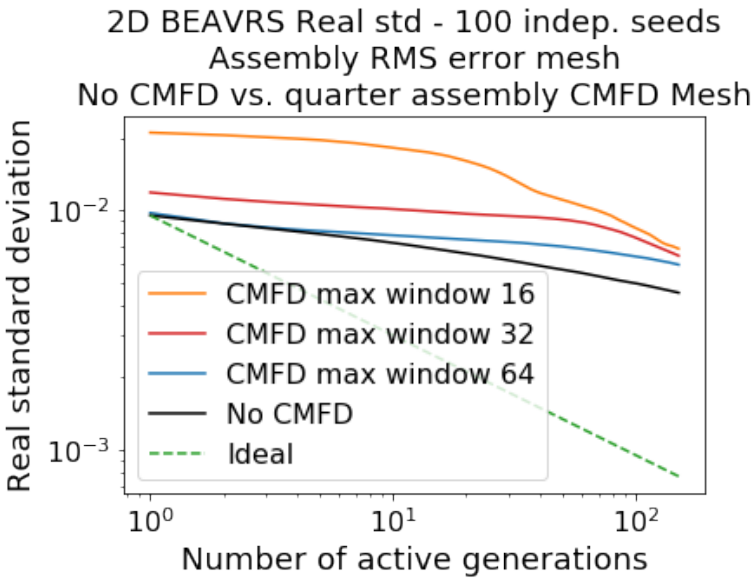


Figure 3-28: Real standard deviation as a function of number of active generations - No CMFD vs. Quarter assembly CMFD mesh with varying window size for 2-D problem. Real standard deivation is computed over 100 independent trial with a generation size per trial of 10 million and assembly-sized RMS error mesh. Real standard deviation is shown for the first 100 active generations (left) and zoomed into the first 5 active generations (right).

Convergence in RMS error for simulations that use a fine RMS error mesh behave in accordance with ideal $1/\sqrt{N}$ rates but correlation effects start to emerge when a coarse RMS error mesh is used instead. A fine-mesh CMFD solution and coarse-mesh CMFD solution exhibit similar RMS error levels when generation size is kept fixed. Furthermore, a larger maximum window size yields a lower starting RMS error level in the initial active generations, but tends to result in slower convergence rates as active generations progress when compared to the case when a smaller maximum window size is used. Increasing generation size shifts RMS error plots lower by a factor of $1/\sqrt{N}$ but does not eliminate correlation effects as the active generations progress. A linear prolongation CMFD update step helps to lower initial RMS levels for coarse CMFD acceleration meshes

3.5 Optimal Run Strategies for Monte Carlo Simulations Using CMFD Source Accelerations

The sensitivity studies conducted in Sections 3.3 and 3.4 investigated the parameters that affected the behavior of the standard deviation of the instantaneous fission source in the inactive generations and the RMS error in the active generations. If CMFD source acceleration is used in the inactive generations, it is recommended to use a maximum window size that matches the CMFD penalty factor of the problem to ensure that instantaneous fission source levels match that of the un-accelerated case. In the active generations, simulations that attained maximum window sizes equivalent to the CMFD penalty factor were able to reach initial RMS error levels in line with the un-accelerated case. However, as the number of active generations simulated increased, correlation effects tended to dominate convergence rates when a coarse RMS error mesh was used irrespective of whether CMFD was used or not. It was also observed that convergence rates in the active generations for CMFD source acceleration schemes with large window sizes were lower than those that used smaller window sizes. In this light,

it is recommended to turn off CMFD source acceleration in the active generations, as it adds computational overhead to the Monte Carlo transport process, and does not provide better convergence rates than the un-accelerated case. Finally, it was found that coarse CMFD acceleration meshes should be preferred over fine meshes since there was no discernible advantage in using a fine-mesh solution in the inactive and active generations, and the fine-mesh solver is much more computationally costly to run. For 1-D problems, the use of linear prolongation helped reduce RMS error levels in the active generations when a coarse CMFD acceleration mesh was used, but did not remove the underlying stochasticity inherent in the Monte Carlo process. This stochasticity is largely controlled by the generation size, where a larger generation size reduces initial error levels in the active generations by a factor of $1/\sqrt{N}$, but does not eliminate convergence degradation from correlation effects when a large number of active generations are simulated. From here onwards, all 1-D simulations will use a fixed source prolongation step, but a linear prolongation step will be used once again for the 3-D BEAVRS test cases in Chapter 5 in the axial dimension.

This section combines the findings from this chapter to assess overall runtime between CMFD source acceleration and the un-accelerated case. Given that in the limit of many active generations the behavior of RMS error does not change drastically based on whether CMFD acceleration is used or not, the focus on this section is to compare different run strategies and instead assess runtime in just the inactive generations to determine how to reach stationarity in the quickest possible way. Here, the figures of merit are not only total runtime and generations to stationarity, but also standard deviation in the instantaneous fission source. In order to reduce the number of variables for this study, generation size is kept fixed at 10 million and all test cases are run on a single node on the same computer cluster that was described in Section 3.2.3.3. Simulation strategies on larger computer systems will be explored in more detail in Chapter 4.

3.5.1 Summary of Run Strategies

Four simulation strategies will be explored in this section. The first run strategy, denoted Strategy 1, does not use any source acceleration schemes. Strategies 2 and 3 will use CMFD source acceleration with maximum window sizes of 16 and 64 respectively. Strategy 4 will be a hybrid run strategy, where CMFD is run with an expanding window scheme with no maximum window size until it reaches a loosely converged fission source distribution. At this point, CMFD feedback is turned off until a tighter convergence criteria is met. In this scenario, the convergence criteria is controlled by the stopping criteria parameters presented in Section 2.3.2, where $S_A = 5, S_B = 30, S_C = 5$ when CMFD feedback is running, and the typical stopping criteria parameters $S_A = 10, S_B = 30, S_C = 10$ are used when there is no CMFD feedback. This hybrid approach tries to combine the benefits of CMFD acceleration and un-accelerated Monte Carlo, where CMFD allows for the quickest reduction in fission source bias during initial inactive generations, while un-accelerated Monte Carlo offers faster runtime per batch and a reduction in instantaneous fission source variance without relying on a large CMFD window size.

3.5.2 Run Strategy Results

Table 3.6 summarizes the runtimes for a variety of run strategies and mesh sizes for the 1-D case, and these results are computed as the average over five independent trials. The case with no CMFD acceleration takes a considerable amount of time due to the number of generations required for stationarity. Similarly, a fine mesh CMFD solution requires a significant amount of runtime due to the computational constraints imposed from tallying on a fine mesh and from running a fine-mesh CMFD solver. On the other hand, a coarse mesh CMFD solution is able to effectively reduce the overall runtime to reach stationarity. When a window size of 16 is used, fewer generations to stationarity are required to reach the higher levels of system-level variance associated with using a smaller window size. A larger window size, however, requires a larger number of generations to reach lower variance levels. This behavior is in line with the observations

from Chapter 2, where a larger number of generations were required for simulations with larger generation sizes in order to reach lower target variance levels. In light of this, the hybrid approach employed in Strategy 4 is the most effective for reducing runtime while achieving variance levels on par with Strategy 1. Even though Strategy 4 takes more generations to stationarity than Strategy 3, many of these generations occur while CMFD is turned off, where runtime per generation is reduced by nearly 60%.

Similar results occur for the 2-D BEAVRS case, which are shown in Table 3.7, where Strategy 3 with a coarse assembly-sized CMFD mesh is able to reduce runtime most effectively while yielding variance levels on the same level as the un-accelerated case. The 2-D BEAVRS problem is a much more realistic reactor problem with heterogeneous geometry and many materials, so the runtime per generation for un-accelerated Monte Carlo is now only 14% lower than that of CMFD-accelerated Monte Carlo, which makes running Monte Carlo without CMFD relatively more costly. The symmetry of the initial starting source guess also results in lower overall time to stationarity compared to the 1-D case. Thus, if Strategy 4 is employed, it can be said that CMFD source acceleration results in a roughly 52% reduction in time to stationarity for the 1-D problem while maintaining the same fission source standard deviation levels as un-accelerated Monte Carlo. For the 2-D case, this reduction in runtime is roughly 60%, and the difference between the CMFD-only strategies and the hybrid strategy do not vary significantly.

Table 3.6: Summary of run strategies and figures of merit related to inactive generation stationarity for the 1-D Homogeneous case

CMFD Mesh Size	Run Strategy	Generations to Stationarity	Inst. Fission Source Std	Total runtime (CPU-hrs)
None	Strategy 1	879.0	1.554×10^{-2}	189.80
20cm	Strategy 2	190.0	2.132×10^{-2}	95.34
20cm	Strategy 3	213.0	1.557×10^{-2}	106.88
20cm	Strategy 4	241.6	1.555×10^{-2}	92.76
0.4cm	Strategy 2	193.4	2.120×10^{-2}	270.23
0.4cm	Strategy 3	216.8	1.568×10^{-2}	302.93
0.4cm	Strategy 4	243.0	1.556×10^{-2}	215.49

Table 3.7: Summary of run strategies and figures of merit related to inactive generation stationarity for the 2-D BEAVRS case

CMFD Mesh Size	Run Strategy	Generations to Stationarity	Inst. Fission Source Std	Total runtime (CPU-hrs)
None	Strategy 1	169.8	1.002×10^{-1}	116.85
Assembly	Strategy 2	65.4	1.012×10^{-1}	52.32
Assembly	Strategy 3	72.0	1.002×10^{-1}	57.60
Assembly	Strategy 4	75.0	1.002×10^{-1}	54.30
Quarter assembly	Strategy 2	68.4	1.021×10^{-1}	55.50
Quarter assembly	Strategy 3	71.0	1.004×10^{-1}	57.61
Quarter assembly	Strategy 4	79.2	1.003×10^{-1}	57.21

In order to reach stationarity in the inactive generations as quickly as possible for a single-node simulation, it is recommended to use a hybrid simulation strategy that uses CMFD source acceleration for initial inactive generations while prematurely turning off CMFD after a loosely converged solution has been achieved in order to reduce runtime per generation and achieve instantaneous fission source variance levels on par with the un-accelerated case. CMFD feedback should continue to be turned off during the active generations as well.

3.6 Conclusion

This chapter investigated the efficacy of CMFD as well as the different parameters associated with this source acceleration scheme. A significant portion of this work centered around comparing CMFD simulations to un-accelerated simulations on the same grounds to ensure both simulation strategies yielded the same quality of solution. To this end, the FET-based stationarity diagnostic was extended to apply to CMFD problems in order to make a fair comparison in terms of generations to stationarity with the un-accelerated case. Then, a sensitivity analysis was conducted in the inactive generations to determine the key parameters that affected the standard deviation in the

instantaneous fission source. Here, it was found that refining the mesh size had little impact on reducing global variance levels. On the other hand, increasing generation size and maximum CMFD window size yielded lower standard deviation levels, where a larger generation size had a significantly larger effect than window size on reducing fluctuations. Also, for a fixed generation size, a minimum window size - defined as the CMFD penalty factor - needed to be achieved in order for CMFD acceleration and un-accelerated schemes to achieve comparable variance levels in the instantaneous fission source. If this minimum window size was not met, a CMFD-based solution was likely to introduce larger noise levels into the simulation.

Next, a similar sensitivity study was conducted in the active generations to consider the effects of generation size, mesh size, maximum window size, and linear prolongation shape function on RMS error and real standard deviation levels. Here, it was found again that mesh size made little difference to error levels in the active generations, whereas a larger window size typically led to lower starting RMS errors. However, as the number of active generations increased, the convergence rate began to decline for CMFD simulations with larger window sizes, likely due to the added correlation effects introduced to each generation of source particles. For the 1-D case, assuming a linear source shape function in the prolongation step helped to reduce RMS error levels for coarsely-defined CMFD acceleration meshes but errors were still dominated by the stochasticity inherent within the Monte Carlo process. Likewise, increasing generation size helped to reduce error levels by a factor of $1/\sqrt{N}$ but did not remove correlation effects in the active generations. Furthermore, given the similarity in error levels when a large number of active generations was simulated, it is recommended to turn off CMFD in the active generations in order to reduce runtime and minimize time in the Monte Carlo tallying process.

Finally, a runtime analysis of various run strategies was administered to determine how to minimize total time to stationarity in the inactive generations when running Monte Carlo on a single-node machine. For this study, standard deviation in the fission source was also monitored as a figure of merit. Here, it was found that running CMFD with a small window size was the quickest way to reach stationarity but introduced higher

variance levels in the instantaneous fission source when compared to the un-accelerated case. Similarly, a CMFD acceleration scheme with a maximum window size as large as the CMFD penalty factor spent a considerably higher number of generations in the inactive generations just to reach a larger maximum window size. Thus a hybrid solution that combined a loosely-converged CMFD solution during initial inactive generations while reverting to un-accelerated Monte Carlo during the later inactive generations was shown to reduce overall runtime without introducing additional variance to the instantaneous fission source. Although this hybrid solution resulted in a larger number of total generations to stationarity compared to the CMFD-only cases, many of the later generations where CMFD is turned off run much faster since CMFD tallies no longer have to be calculated. Overall, a CMFD-based run strategy is able to reduce time to stationarity by 52% for the 1-D homogeneous test problem with an offset starting source, and by 60% for the 2-D BEAVRS problem with a symmetric starting source.

Highlights

- An FET-based stationarity diagnostic can be applied to simulations that utilize CMFD source acceleration.
- Standard deviation in the instantaneous fission source for CMFD-based simulations is dictated primarily by the maximum window size for a windowing scheme and by the generation size.
- Initial RMS errors levels in the active generations are dictated primarily by the maximum window size for a windowing scheme and by the generation size. As the number of active generations increases, most simulation strategies reach comparable error levels due to the presence of correlation effects. A linear prolongation step also helps reduce initial RMS error levels for coarsely defined CMFD acceleration meshes.
- In order to reduce overall runtime in the inactive generations, a hybrid run strategy can be employed that loosely converges a CMFD solution in the initial inactive generations and then turns off CMFD altogether so as not to introduce additional variance to the instantaneous fission source. However, the difference in runtime between a hybrid approach and CMFD-only approach is not substantial for realistic heterogeneous problems.

Part III

Optimal Run Strategies Using Ensemble Averaging

Chapter 4

Motivation for Ensemble Averaging

Chapter 3 examined the effects of various CMFD parameters in both the inactive and active generations, and one of the main takeaways from this work was that if a significant number of active generations is simulated, a considerable degradation in active convergence rate is incurred regardless of simulation strategy because of inter-generational correlation effects. When thinking about how to distribute a fixed "budget" of source particles between the inactive and active generations, a tradeoff emerges. If all of the effort in a Monte Carlo simulation is spent in a single trial with a small generation size and a large number of active generations, real standard deviation levels and thus RMS error levels will decrease at slower than ideal $1/\sqrt{N}$ rates. Miao characterizes the exact relationship between real standard deviation and number of active generations in terms of the autocorrelation coefficients (ACC's) and finds that asymptotic convergence rates do approach $1/\sqrt{N}$ [28]. However, a considerable number of active generations is required for high dominance ratio problems before this asymptotic regime is reached. Instead, if a larger generation size or if multiple independent trials of Monte Carlo are used, then a larger fraction of source particles will be in the discarded inactive generations since a larger number of initial starting source particles will have to be converged to a stationary fission source distribution. Once stationarity is met, however, the initial RMS error reduces at a rate of $1/\sqrt{N}$, where N is the generation size.

Romano et al. explore this relationship in more detail to determine optimal run strategies for Monte Carlo simulations [76]. They develop a relationship to relate the

real variance in a Monte Carlo simulation to the number of independent trials and the number of inactive and active generations simulated per trial, and the key equations are summarized below.

Consider the random variable Y with variance σ^2, which estimates a Monte Carlo tally score for a single source particle. If we let X_i represent the average tally score based on the aggregate source particles in active generation i, then the real variance of the sample mean $\overline{X}$ over N active generations and H particles per generation can be calculated as:

$$\mathrm{Var}\left(\overline{X}\right) = \frac{\sigma^2}{HN} r(N) \tag{4.1}$$

$r(N)$ represents the penalty factor incurred in the active convergence rate due to inter-generational correlation effects and is defined as a function of the ACC's ρ_k from generations $k = 1$ to $N-1$, where k is the generational lag and $r(N) > 1$ if $\rho_k > 0$. An exact derivation of the term $r(N)$ can be found in Appendix F. Now, assume S independent trials are conducted each with N_d discarded (inactive) generations, N active generations, and H source particles per generation[4]. If a fixed budget B of source particles is available to distribute among the inactive and active generations for each trial, then $B = SH(N_d + N)$ and the sample mean $\overline{\overline{X}}$ across all independent trials is computed as:

$$\overline{\overline{X}} = \frac{1}{S} \sum_{i=1}^{S} \overline{X}_i \tag{4.2}$$

where $\overline{X}_i$ is the realization $\overline{X}$ for the i-th independent trial. Since all trials are independent of each other, the variance of the sample mean across all independent trials is calculated as:

$$\mathrm{Var}\left(\overline{\overline{X}}\right) = \frac{\sigma^2}{B - SHN_d} r(N) \tag{4.3}$$

[4]Romano et al. use the term *simulation* to denote independent runs of Monte Carlo with a different starting pseudo-random seed, but this is synonymous with the term *trial* that has been used throughout this thesis.

This equation illustrates the tradeoff between inactive and active generations in an alternate way. If a significant number of active generations N are simulated, convergence rates in the variance will be worse than $1/N$ due to the strong correlation effects that are present during initial inactive generations, which is represented by the fact that $r(N) > 1$. On the other extreme, if the source particles are instead used to increase the number of independent trials S or particles per generation H, then fewer active particles - represented by the denominator $B - SHN_d$ - will be available to compute variance levels, but the term $r(N)$ goes to 1 and ACC's can be neglected assuming that only a single active generation is used to compute variance levels. In this scenario, all source particles have been independently converged to a stationary source distribution, and thus can be assumed un-correlated.

This chapter explores this idea of running multiple independent trials of Monte Carlo, defined as *ensembles*, and aggregating tallies over these independent ensembles to generate simple and accurate statistical quantities. Such a simulation strategy is coined *ensemble averaging* in this thesis. Ensemble averaging has been proposed in the domain of CFD simulations of turbulent flow in order to reduce time to solution [77]. However, this idea of ensemble averaging has typically been used in the field of Monte Carlo neutronics to verify real variance estimates, but has not been explored fully as a viable run strategy on large-scale computing systems.

As illustrated in Equation 4.3, ensemble averaging is best suited for running simulations with fewer active generations but larger generation sizes where each generation is split over multiple independent trials. While convention dictates that a large number of active generations is necessary to reach acceptable limits in tally variance levels for Monte Carlo simulations, it was shown recently that there is merit to running Monte Carlo simulations with larger generation sizes and fewer active generations in the context of multiphysics coupling as well. Harper shows that running fewer active generations between successive multiphysics Picard iterations helps limit the amount of change that occurs in the fission source distribution during the TH update step, thereby reducing instabilities in the multiphysics solver [39]. Running a larger generation size also helps to reduce fluctuations in the fission source from successive Monte Carlo iterations after

receiving TH updates from the multiphysics solver. Harper also designs a framework for coupling the Monte Carlo CMFD solver with a TH subchannel code through the use of Monte Carlo tally derivatives, and here too it was found that a relatively large generation size of 200 million was required on quarter-core problems in order to reduce fluctuations with the CMFD to appropriate levels [39]. Thus, the model of ensemble averaging lends itself organically to the simulation of Monte Carlo with multiphysics feedback, where fewer active generations but larger generations sizes should be run.

Several arguments can be made for ensemble averaging on the basis of increased computational performance and resilience on large-scale systems as well. Unlike Chapter 3 where all run strategy considerations were based on single-node runs of Monte Carlo, this chapter assumes a larger availability of computational resources. Each ensemble of particles are run on a single compute node in parallel, and a case will be made in this chapter as to why ensemble averaging is the optimal run strategy for such large-scale Monte Carlo simulations. All runtime analysis and performance considerations are based on simulations that run on multiple nodes of Idaho National Laboratory (INL)'s Falcon supercomputer, where each compute node is composed of 2 Intel Xeon E5-2695 v4 CPU's with 18 cores per CPU.

This chapter is therefore organized as follows: Section 4.1 introduces different strategies for ensemble averaging by running multiple independent trials in parallel. Given the larger job sizes that are considered in this chapter, these methods will be evaluated on the merits of both parallel efficiency as well as generations to stationarity. Section 4.2 motivates the ensemble averaged CMFD run strategy, where all CMFD calculations are offloaded onto a separate compute node while still running Monte Carlo in an ensemble averaged manner. This methodology allows for further parallelization based on asynchronous CMFD updates in a client/server manner, and the implications with respect to performance benefits and potential variance biases are explored. Section 4.3 summarizes all of the different ensemble averaging run strategies proposed in this chapter and compares them on the grounds of overall runtime. Finally, Section 4.4 summarizes the arguments for running ensemble averaging based on computational performance, ability to calculate real variance estimates, and applications to GPU computations and

coupling with nonlinear multiphysics solvers.

4.1 Performance Considerations with Ensemble Averaging

This section introduces the different run strategies that will be employed in this chapter. The same 1-D and 2-D problems are analyzed as Chapter 3. However, this time 32 compute nodes will be used to simulate these problems, and the number of total particles simulated across all of these compute nodes per generation will also be scaled up by a factor of 32. This can be done in two ways: by increasing the generation size for a single trial to 320 million or by increasing the number of independent trials to 32 but keeping generation size per trial fixed at 10 million. In the first case, since only a single trial is run, accurate variance estimates cannot be obtained without explicitly quantifying the correlation levels or by using a batching method. Moreover, as will be explored in Section 4.1.1, using a larger generation size means that each compute node will also have to wait for all other compute nodes to finish their respective particle transport routines in order to redistribute fission source sites across all processes for the following generation and also to gather global tallies. On the other hand, when multiple independent trials of Monte Carlo are run, they are implemented in such a way that the processes handling each trial don't need to communicate with each other during the entirety of the simulation. This allows for the calculation of real variance estimates across all independent trials, and also leads to higher parallel efficiency because now each ensemble of 10 million neutrons can continue to its next generation without having to wait for all other ensembles to finish their transport of neutrons as well. However, as will be shown in Section 4.1.2, without an explicit synchronizing step, a looser convergence of the fission source in the inactive generations will be attained since the inactive generation stopping criteria is based on the behavior of a single ensemble of 10 million neutrons instead of the aggregate 320 million neutrons available across all trials.

Using the term *single trial* to denote a Monte Carlo simulation that runs a single trial

and the term *multiple trial* to denote the ensemble averaging approach where multiple independent trials of Monte Carlo are run in parallel, the following four run strategies are therefore presented in this section:

- Run Strategy 1 - Multiple trial Monte Carlo: The first run strategy simulates 32 independent trials of Monte Carlo in parallel without CMFD, where each compute node simulates an independent trial with 10 million neutrons per generations per trial.

- Run Strategy 2 - Multiple trial CMFD: The second run strategy simulates 32 independent trials of Monte Carlo in parallel with CMFD, where each compute node simulates an independent trial with 10 million neutrons per generations per trial. The maximum window size is set to meet the CMFD penalty factor so as to ensure that equivalent variance levels are achieved between the CMFD-accelerated and un-accelerated cases. For both 1-D and 2-D problems, the maximum window size is set to 64.

- Run Strategy 3 - Single trial Monte Carlo with large generation size: The third run strategy simulates a single trial of Monte Carlo without CMFD, with a generation size of 320 million.

- Run Strategy 4 - Single trial CMFD with large generation size: The fourth run strategy simulates a single trial of Monte Carlo with CMFD and a generation size of 320 million. Once again, the maximum window size is set to be equal to the CMFD penalty factor

As will be discussed in Section 4.1.1, run strategies 1 and 2 have better parallel efficiency when compared to run strategies 3 and 4 due to the lack of a synchronizing step between all particles in a generation. However, run strategies 1 and 2 also represent variance levels that are higher than run strategies 3 and 4 due to the fact that only a single ensemble of neutrons is used to determine stationarity. This subject will be explored in more detail in Section 4.1.2.

Table 4.1: CMFD parameters for 1-D homogeneous and 2-D BEAVRS problems for test cases in Chapter 4

Parameter	Value for 1-D Problem	Value for 2-D Problem
CMFD tally estimator type	Analog	Analog
Number of energy groups	1	2
Effective downscatter	N/A	True
Acceleration map	None	Quarter assembly map
Reference diffusion parameters	Not used	Used
CMFD tally begin	1	1
CMFD solver begin	1	1
Weight clipping	0.2	0.2
Windowing scheme	Expanding window	Expanding window
Maximum window size	64	64
CMFD mesh size	20cm mesh	Quarter assembly mesh
Prolongation source shape	Fixed source	Fixed source

Table 4.1 summarizes the 1-D and 2-D CMFD parameters for the run strategies where CMFD is employed. Since it was found in Chapter 3 that CMFD mesh size did not greatly affect the results in both the inactive and active generations, this chapter keeps CMFD mesh size fixed at 20cm for the 1-D problem and uses a quarter assembly CMFD mesh for the 2-D problem. It should also be noted that for run strategies 1 and 2 where multiple trial ensemble averaging is employed, a single Python script is called using `mpi4py`, which provides the Python bindings to make MPI calls and run multiple trials in parallel [78]. Specifically, the global MPI communicator, `MPI_COMM_WORLD` is split into 32 equivalent communicators using `MPI_Comm_split` and each of these communicators is mapped to a separate compute node that runs a different trial of OpenMC. In addition, since each compute node is comprised of two sockets with 18 cores each, each split communicator is further divided into two MPI processes and 18 OpenMP threads each, so that each socket is bound to a separate MPI process and each OpenMP thread is bound to a separate process core on the socket. For run strategies 3 and 4 where a single trial is run, the global communicator is split into 64 MPI processes so that the total work is distributed among all available sockets in a similar fashion to run strategies 1 and 2. A

summary of the different run strategies introduced in this section as well as the compute resource parameters associated with these run strategies can be found in Table 4.2.

Table 4.2: Compute resource parameters for run strategies 1 to 4

Run Strategy	Total Compute Nodes	Total Independent Trials	Generation Size per Trial	MPI Processes per Trial	OpenMP Threads per MPI Process
1 - Multiple Trial Monte Carlo	32	32	10 Million	2	18
2 - Multiple Trial CMFD	32	32	10 Million	2	18
3 - Single Trial Monte Carlo	32	1	320 Million	64	18
4 - Single Trial CMFD	32	1	320 Million	64	18

4.1.1 Identifying Bottlenecks in Monte Carlo Simulations

In order to assess the parallel efficiency of these four run strategies, a performance profiling tool is created in order to analyze communication bottlenecks for large-scale Monte Carlo runs, which produces a graphical representation of when each MPI process is doing useful work. In the un-accelerated case, particle transport of all neutrons in the source bank constitutes the majority of useful work, while any time spent synchronizing the fission bank or accumulating global tallies after a generation would count towards process downtime. In the case of CMFD acceleration, all CMFD multigroup cross section and flux calculations occur on the master process (process 0), and the resulting CMFD fission source is broadcasted to all other processes for a particular trial in order to reweight all particles in the fission bank. Therefore, downtime in the CMFD algorithm constitutes time that processes have to wait for the CMFD source to synchronize across all processes.

Figures 4-1 through 4-4 illustrate the performance profiling plots for run strategies 1 to 4 respectively on the 1-D homogeneous problem for 10 generations. The horizontal axis represents time elapsed by the simulation, while the vertical axis plots the behavior

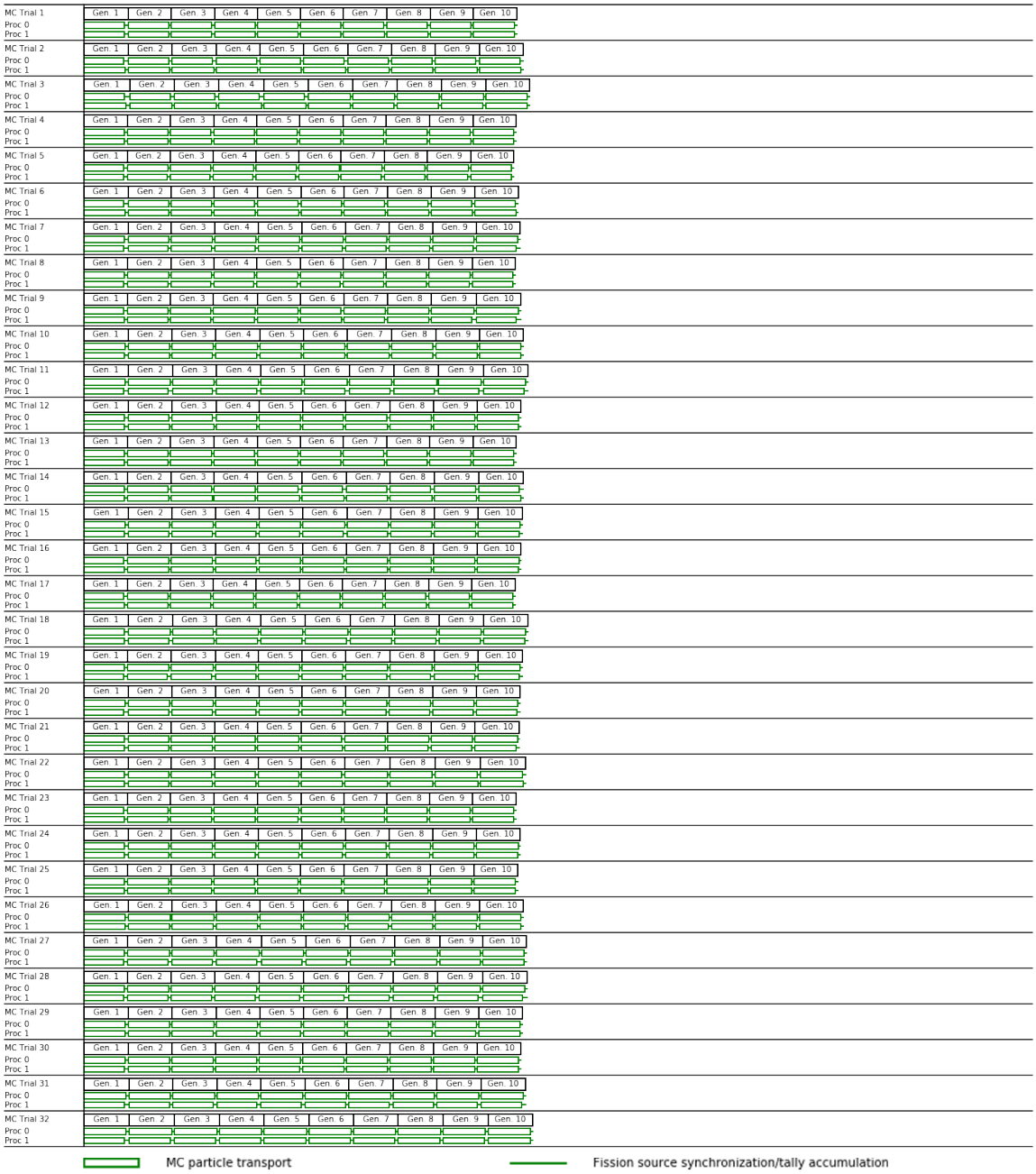


Figure 4-1: Performance profiling plot for run strategy 1 on 1-D homogeneous problem. The green boxes represent when processes are running particle transport at each generation for each trial, while horizontal green lines indicate when processes are stalling while synchronizing the fission source and accumulating global tallies.

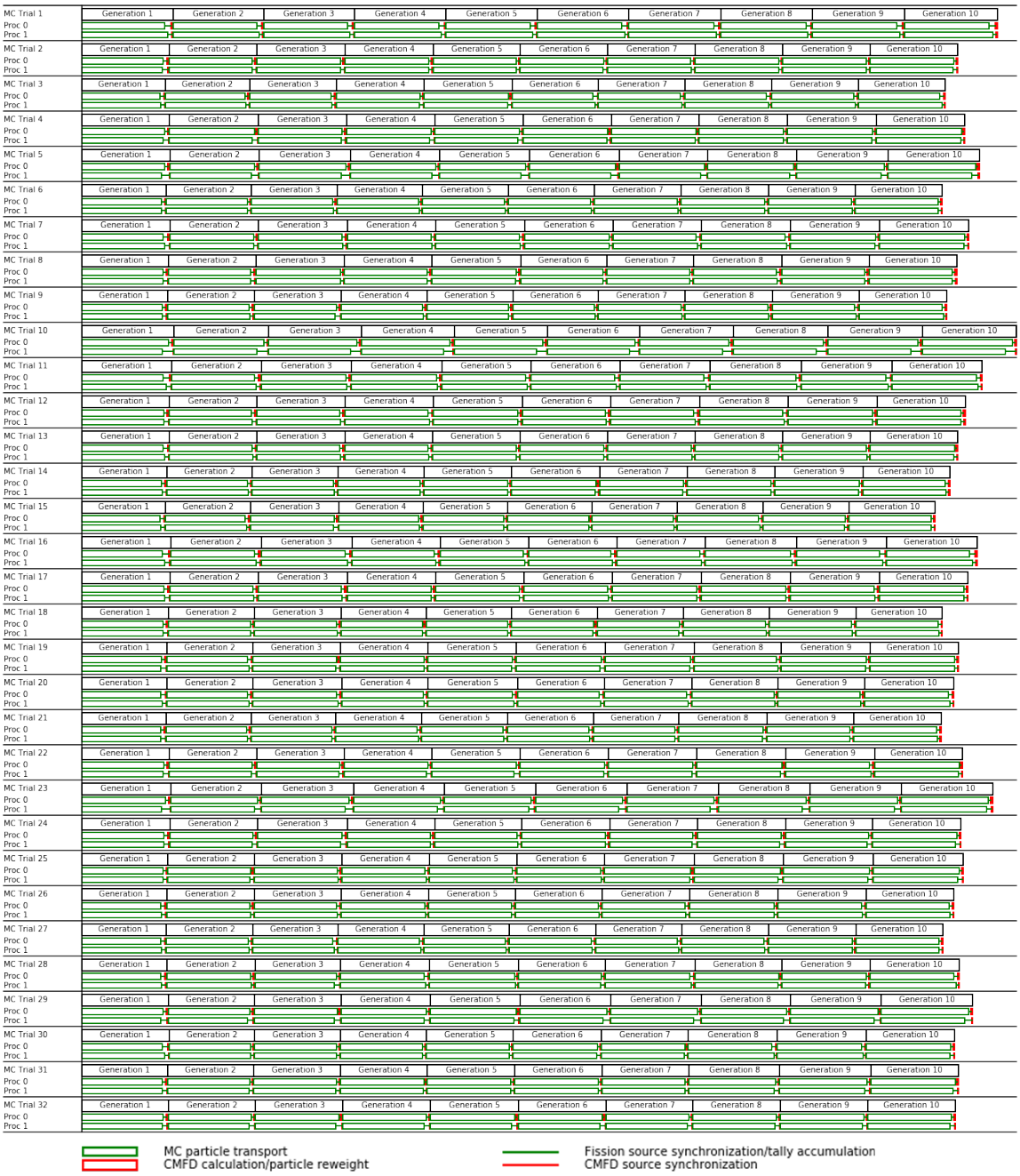


Figure 4-2: Performance profiling plot for run strategy 2 on 1-D homogeneous problem. The green portions indicate time spent in the Monte Carlo routine, while the red portion indicates time spent in the CMFD feedback step. Green boxes represent when processes are running particle transport at each generation for each trial, while horizontal green lines indicate when processes are stalling while synchronizing the fission source and accumulating global tallies. Likewise, red boxes indicate when CMFD calculation is occurring on the master process or when particle reweighting occurs on each process.

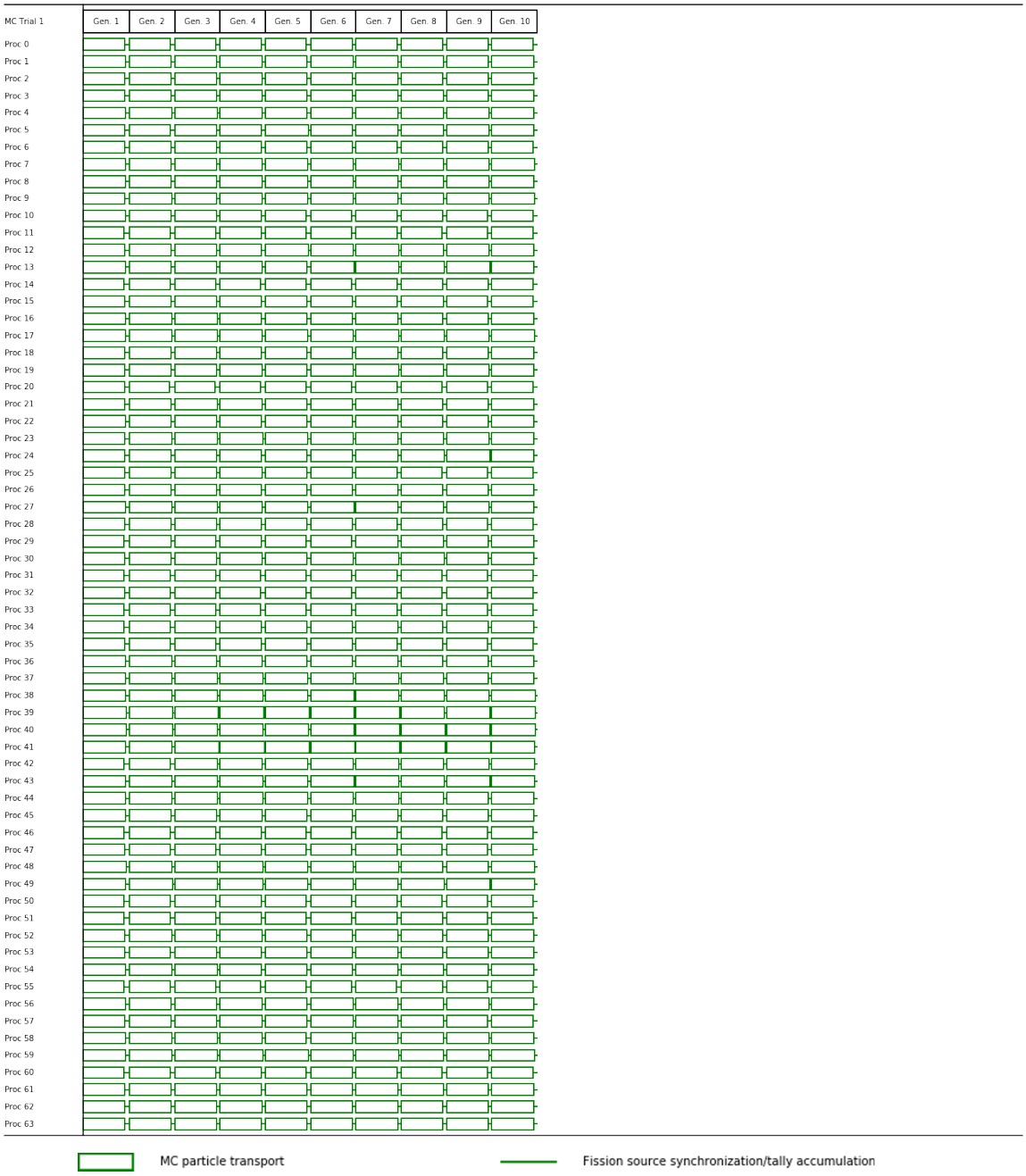


Figure 4-3: Performance profiling plot for run strategy 3 on 1-D homogeneous problem. The green boxes represent when processes are running particle transport at each generation for each trial, while horizontal green lines indicate when processes are stalling while synchronizing the fission source and accumulating global tallies.

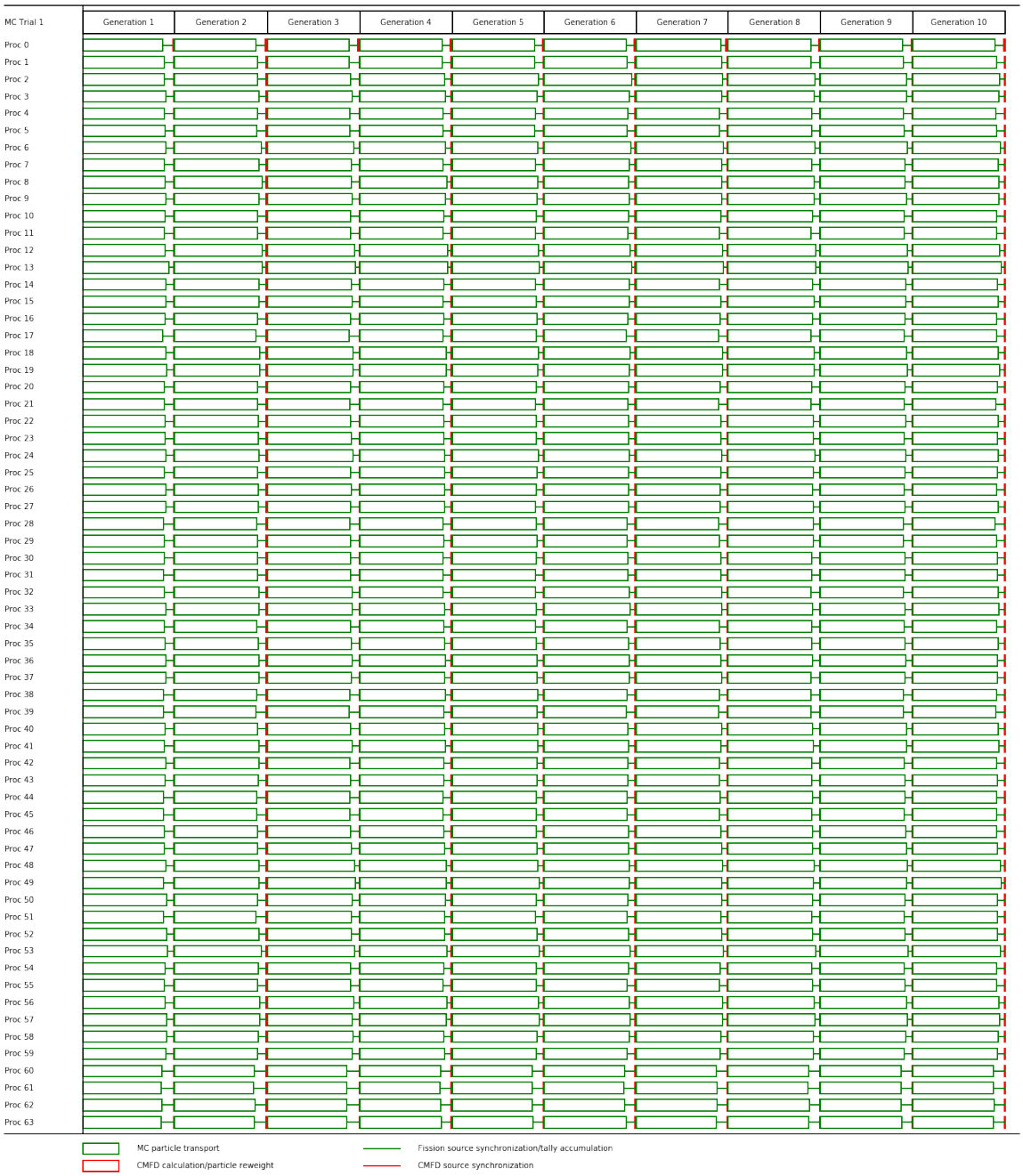


Figure 4-4: Performance profiling plot for run strategy 4 on 1-D homogeneous problem. The green portions indicate time spent in the Monte Carlo routine, while the red portion indicates time spent in the CMFD feedback step. Green boxes represent when processes are running particle transport at each generation, while horizontal green lines indicate when processes are stalling while synchronizing the fission source and accumulating global tallies. Likewise, red boxes indicate when CMFD calculation is occurring on the master process or when particle reweighting occurs on each process.

of each process running in parallel, and runtimes are normalized to the slowest-running trial across all different run strategies. Green portions indicate where the Monte Carlo solver routines are running, while red portions indicate areas where CMFD solver routines are being called. In addition, areas with green or red boxes represent parts of the simulation where the underlying process is busy with computation. On the other hand, horizontal lines represent sections of the program where the process is in an idle state, where the process is waiting to communicate or synchronize with other processes.

Run strategies 1 and 2 minimize time between generations, as each trial only has to wait for two processes to synchronize before moving onto simulating the next generation of neutrons, and each trial progresses completely independently of all other trials. On the other hand, run strategies 3 and 4 have to synchronize the fission bank and global tallies across all 64 available MPI processes, so each process has to wait for the slowest-running process to finish its particle transport routine before being able to progress to the next generation. The total integrated time over all trials and processes indicates the total CPU-time for a particular run strategy, and this amount is lower when an ensemble-averaged approach is chosen over a single trial approach.

Another observation from these plots is the additional runtime incurred by the use of the CMFD solver. While the time spent in the red CMFD feedback step is quite low compared to the overall time taken to simulate a generation of neutrons in the Monte Carlo solver, the runtime for the Monte Carlo transport algorithm is significantly larger for the CMFD-accelerated case as compared to the un-accelerated case. This is because CMFD tallies have to be computed during the particle transport routine, and this incurs a significant computational overhead in the inactive generations. While the un-accelerated case is able to run faster than the CMFD-accelerated case on a per-generation basis for a fixed number of generations, this analysis ignores the reduction in number of generations to stationarity that is provided through CMFD acceleration. Section 4.3 analyzes the overall runtime associated with these run strategies by accounting for both total time per generation as well as generations to stationarity.

Figures 4-5 through 4-8 plot the same profiling figures for run strategies 1 to 4 respectively on the 2-D BEAVRS problem. Similar to the 1-D homogeneous case, ensemble

averaging in run strategies 1 and 2 allows for a reduction in overall runtime due to better load balancing across all independent trials. For the 2-D BEAVRS case, however, the heterogeneous geometry of the problem as well as the presence of additional materials in the reactor core increases the time to simulate each additional Monte Carlo generation and thus reduces the relative cost of running CMFD.

From these figures, it can be seen that running multiple independent trials of Monte Carlo helps to reduce communication costs associated with fission source synchronization and tally accumulation. However, as will be explored in Section 4.1.2, the absence of a synchronization step across these independent trials of Monte Carlo will ultimately lead to a less stringent convergence criteria, since stationarity will be diagnosed based on the fission source behavior of a single ensemble of neutrons instead of looking at the entire source distribution across all ensembles. A tighter convergence criteria on par with run strategies 3 and 4 can only be attained if there is some level of synchronization across all trials.

4.1.2 Single Trial vs. Aggregate Trial Stopping Criteria

In Chapter 2, the idea of having the stopping criteria be a function of generation size was introduced. Figures 2-21, 2-22, and 2-23 illustrate how when a larger generation size is used, the variance in FET coefficients - which represents the variance in the underlying fission source - reduces to a lower asymptotic level, but this level is only attainable through the simulation of additional inactive generations. Similar to a deterministic calculation where more source iterations are required to reach a tighter fission source error convergence, a larger generation size in a Monte Carlo simulation allows for a lower fission source variance through the simulation of additional inactive generations. A run strategy that keeps the number of inactive generations constant regardless of generation size is likely to carry higher fission source error levels in the active generations.

This situation leads to an inherent tradeoff when analyzing inactive generation stopping criteria for the multiple trial ensemble averaging approaches introduced in run strategies 1 and 2. While these run strategies eliminate the need to synchronize data

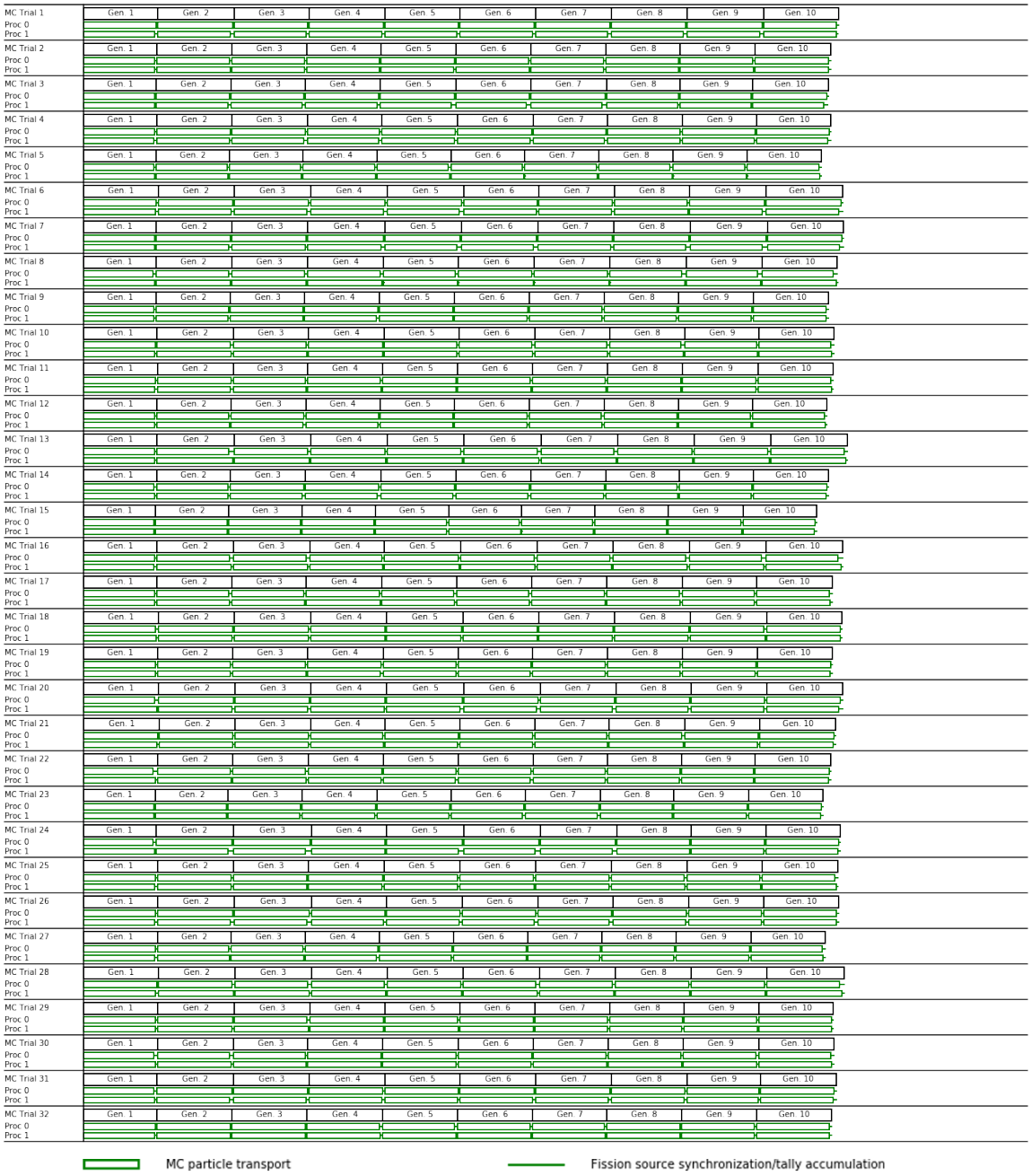


Figure 4-5: Performance profiling plot for run strategy 1 on the 2-D BEAVRS problem. The green boxes represent when processes are running particle transport at each generation for each trial, while horizontal green lines indicate when processes are stalling while synchronizing the fission source and accumulating global tallies.

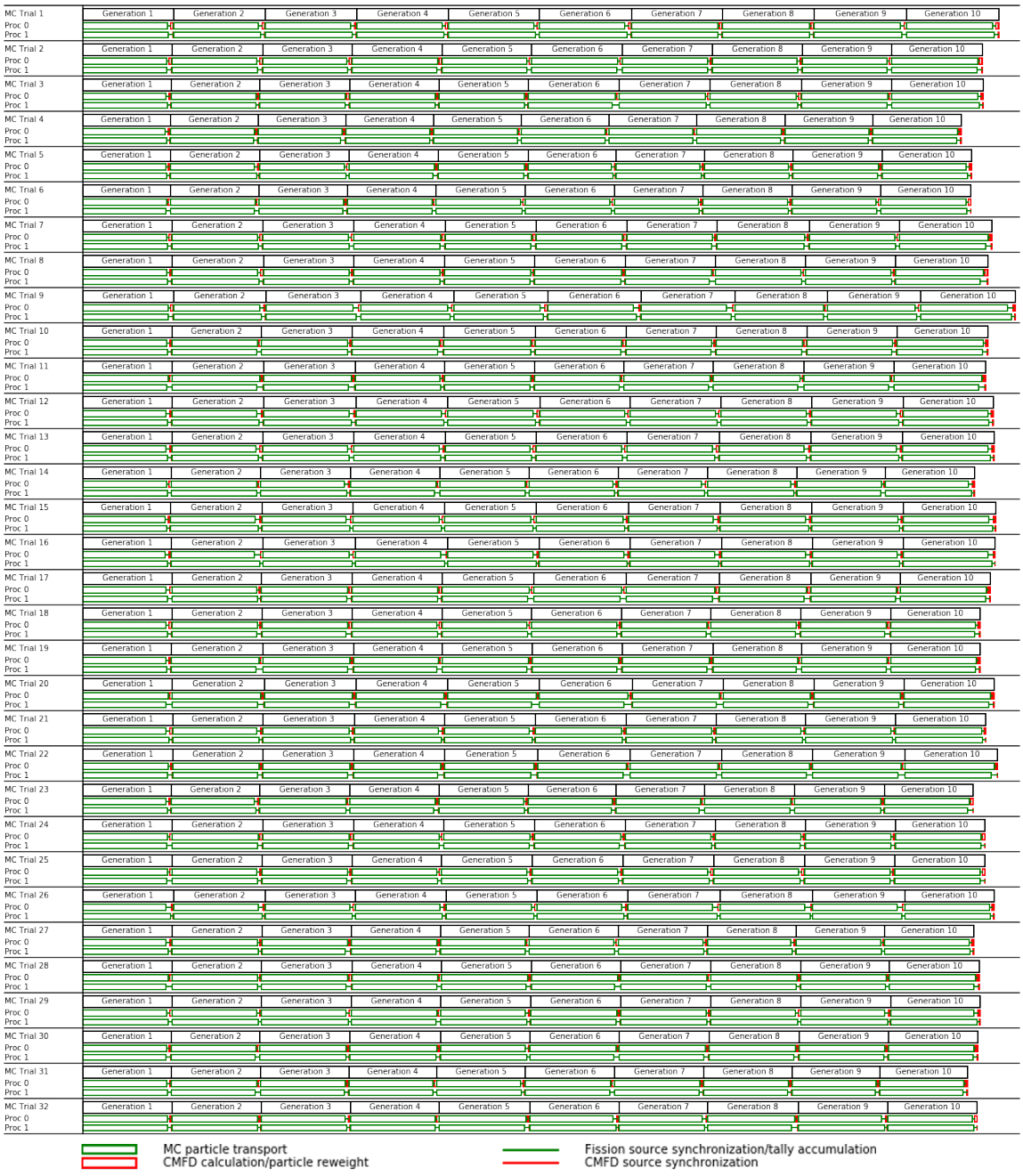


Figure 4-6: Performance profiling plot for run strategy 2 on the 2-D BEAVRS problem. The green portions indicate time spent in the Monte Carlo routine, while the red portion indicates time spent in the CMFD feedback step. Green boxes represent when processes are running particle transport at each generation for each trial, while horizontal green lines indicate when processes are stalling while synchronizing the fission source and accumulating global tallies. Likewise, red boxes indicate when CMFD calculation is occurring on the master process or when particle reweighting occurs on each process.

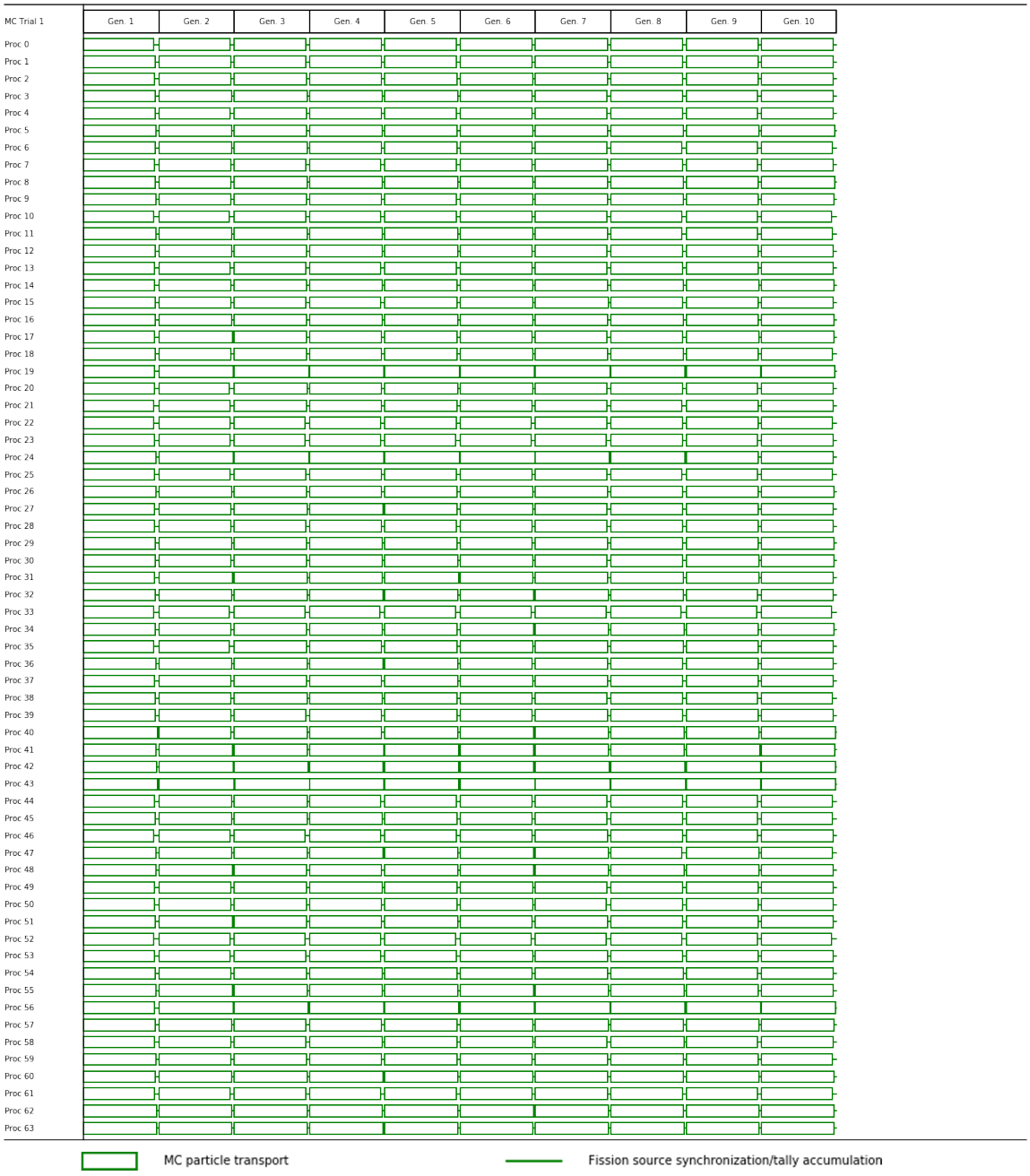


Figure 4-7: Performance profiling plot for run strategy 3 on the 2-D BEAVRS problem. The green boxes represent when processes are running particle transport at each generation for each trial, while horizontal green lines indicate when processes are stalling while synchronizing the fission source and accumulating global tallies.

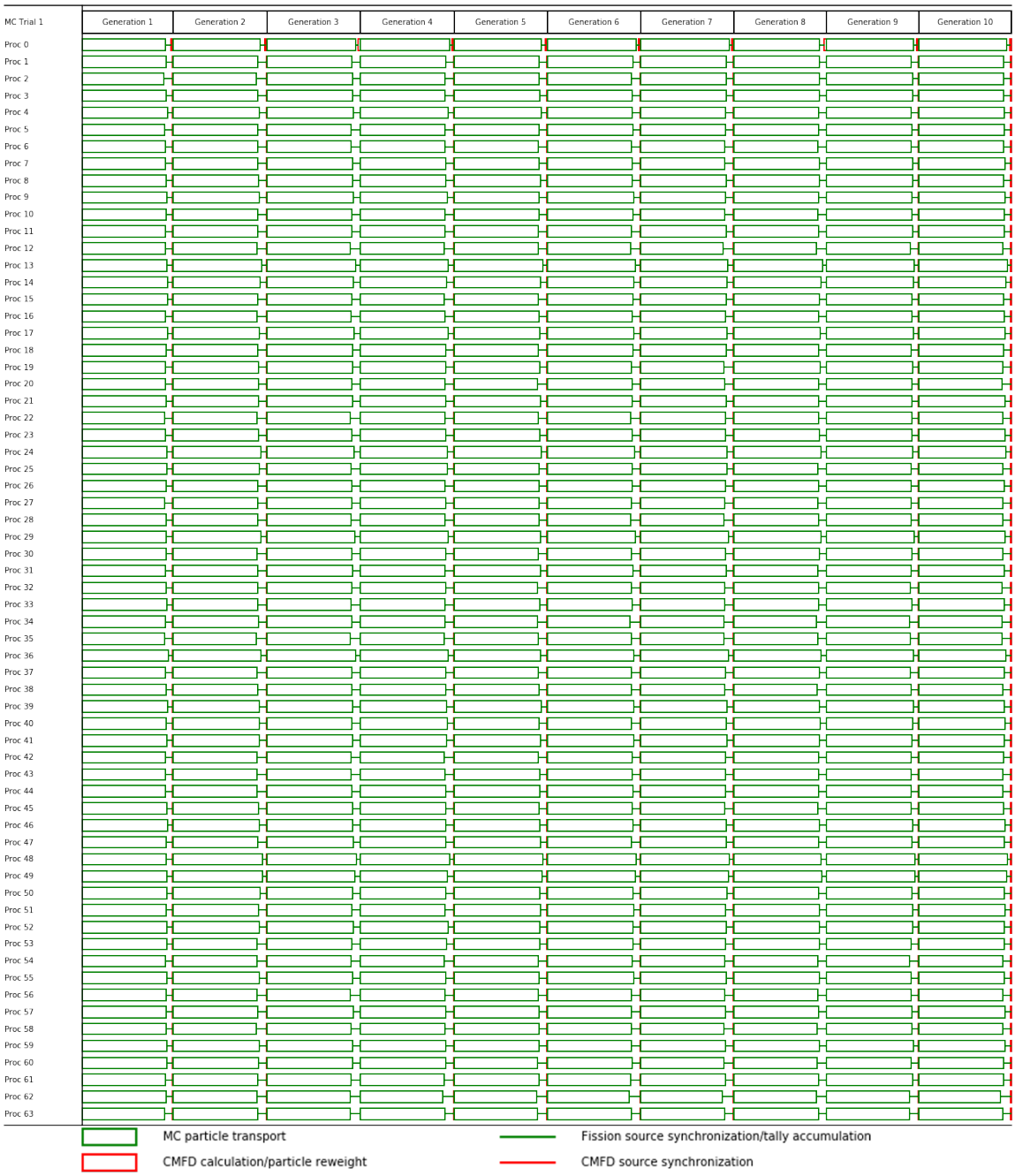


Figure 4-8: Performance profiling plot for run strategy 4 on the 2-D BEAVRS problem. The green portions indicate time spent in the Monte Carlo routine, while the red portion indicates time spent in the CMFD feedback step. Green boxes represent when processes are running particle transport at each generation, while horizontal green lines indicate when processes are stalling while synchronizing the fission source and accumulating global tallies. Likewise, red boxes indicate when CMFD calculation is occurring on the master process or when particle reweighting occurs on each process.

across multiple independent trials, the stopping criteria used to determine stationarity in the inactive generations will be based on the fission source of at most a single trial of fission source data. Specifically, since fission source data is not communicated between trials, each trial will base its stopping criteria on a fission bank of 10 million neutrons instead of the aggregate 320 million particles available across all multiple independent trials if a fission bank synchronizing mechanism is utilized like run strategies 3 and 4. Therefore, run strategies 1 and 2 base convergence on a less stringent stationarity criteria, and will be prone to higher starting error levels in the active generations.

To illustrate this point, consider the case where 100 independent trials of the 2-D BEAVRS problem is simulated without CMFD acceleration, where each trial has a generation size of 10 million particles. When diagnosing the stationarity point between the inactive and active generations, one of two approaches can be pursued. The first approach would be to compute the stopping criteria completely independently for each trial such that the stopping criteria is based on the 10 million source particles that comprise each trial's source distribution. For the remainder of this thesis, the term *single trial stopping criteria* will be used to designate such an approach, where a separate stopping criteria is established for each independent trial based on data from a single trial's source distribution. The second way to establish stationarity would be to aggregate the source particles for each generation over all trials and set a single stopping criteria based on this aggregated source distribution of $100 \times 10\text{million} = 1\text{billion}$ source particles per generation. Such an approach is coined *aggregate trial stopping criteria*, where the aggregated source over all multiple independent trials is used to determined a single stopping criteria for all trials.

Figure 4-9 plots the overall RMS error as a function of the number of independent trials that contribute to the observed source distribution. Each additional trial contributes a single active generation of 10 million source particles to the overall observed source distribution. The trendline marked as "single trial stopping criteria" designates its first active generation based on the single trial stopping criteria described in the previous paragraph, where stationarity for each Monte Carlo trial is based separately on each trial's fission source of 10 million particles. On the other hand, the "aggregate trial

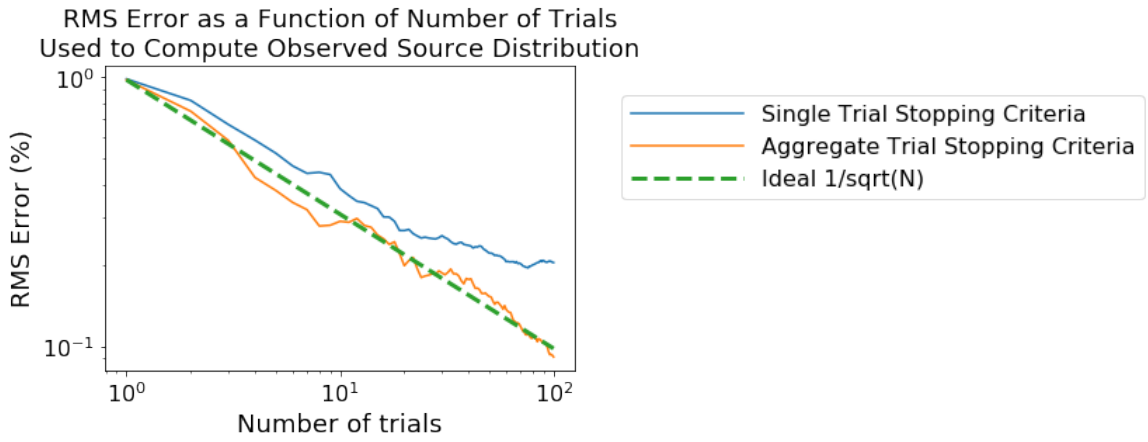


Figure 4-9: RMS error as a function of number of trials used to compute observed source distribution. Each additional trial contributes a single active generation of 10 million source particles to the overall source distribution. The blue curve corresponds to the case where the first active generation for each trial is determined based on a single trial stopping criteria, while the orange line corresponds to the case where the first active generation for each trial is determined based on an aggregate trial stopping criteria.

stopping criteria" trendline designates the first active generation based on the aggregate fission source across all 100 trials, and this stopping criteria is kept the same across all independent trials. RMS error is calculated on an assembly-sized mesh, and the RMS error is expected to decrease at a $1/\sqrt{N}$ rate, since the contribution of each additional trial yields an additional 10 million uncorrelated particles to the overall fission source. However, it can be seen that when a single trial stopping criteria is used, there is some residual error that remains from the fact that an insufficient number of inactive generations were simulated that did not account for the aggregate source distribution across all independent trials. Instead, when an aggregate trial stopping criteria is used, which sets the cutoff point between inactive and active generations as a constant value across all trials based on the aggregate fission source, the convergence in the RMS error follows ideal $1/\sqrt{N}$ rates. For comparison, the single trial stopping criteria diagnosed stationarity at 167.2 generations when averaged across all 100 trials, while the aggregate trial stopping criteria diagnosed stationarity at 198 generations, indicating a stricter convergence criteria when an aggregate trial stopping criteria is utilized.

Returning to the four run strategies presented in Section 4.1, a single trial stopping

criteria results in a roughly 29% higher starting RMS error value for the 1-D problem and a 33% higher starting RMS for the 2-D problem when compared to the case where an aggregate trial stopping criteria is used. Hence, a tradeoff arises between the multiple trial ensemble averaging approaches in run strategies 1 and 2 and the single trial approaches employed by run strategies 3 and 4. In the multiple trial case, each Monte Carlo trial is able to progress to subsequent generations without having to wait for all other trials to finish running their ensembles of neutrons, yielding higher parallel utilization on larger compute jobs. However, if these trials do not contain any synchronizing step, then stationarity will be determined based on the behavior of at most a single ensemble of neutrons, and this results in a looser stationarity criteria, which translates to a higher starting RMS error in the active generations. On the other hand, if a single trial simulates all available particles in a generation, then stationarity is determined by a stricter convergence criteria based on the aggregate source at each generation, ensuring asymptotic variance levels according to the larger generation sizes are met. However, such a simulation strategy requires that all particles in the generation are transported before being able to progress to the next generation. If the entire generation of particles is split evenly among all available processes, this results in a higher percentage of CPU idling as processes have to wait for all other processes to finish their work before being able to continue to the next generation of particle transport.

A multiple trial ensemble averaging approach improves parallel efficiency and load balancing on Monte Carlo runs with a larger availability of computational resources. However, fission source stationarity is based on a single trial of source data instead of considering the aggregate fission source, so this yields a less stringent stationarity criteria and hence a higher residual error in the active generations. On the other hand, running a single trial with a large generation size assesses stationarity on the aggregate fission source but increases communication costs in fission source synchronization and global tally accumulation.

4.2 Motivation for Asynchronous Ensemble Averaged CMFD

Section 4.1 illustrated the tradeoff between running a single trial simulation with a larger generation size and a multiple trial ensemble averaged approach where the total generation size was split into smaller ensembles that run in parallel. For the ensemble averaged case, running independent ensembles in parallel helped increase parallel efficiency, but the independence of these ensembles also meant that only a single ensemble of neutrons could be used to diagnose stationarity, which would yield a less stringent stationarity point when compared to the case where all source particles across all ensembles were aggregated. On the other hand, a single trial approach ensured stationarity was based on the aggregate source across all available particles in a generation, but yielded lower parallel efficiency due to explicit synchronizing calls across all available processes. Moreover, quantifying real variance estimates in a single trial simulation is not straightforward without computing correlation effects between active generations, which can become quite costly.

This section therefore presents an alternate avenue towards ensemble averaging, where all CMFD calculations are offloaded to a separate compute node. This run strategy is coined *ensemble averaged CMFD*. In this approach, all compute nodes running independent trials of Monte Carlo still continue to progress independently of each other. However, these compute nodes now communicate with a separate central compute node responsible for all CMFD calculations and FET stationarity diagnosis when entering the CMFD update step. The compute nodes running Monte Carlo will be referred to as the *Monte Carlo (MC) nodes*, while the centralized node responsible for CMFD and FET stationarity calculations will be called the *CMFD node*. Unlike the multiple trial CMFD approach described in Section 4.1 where all multiple independent trials of Monte Carlo calculate CMFD completely independently, now each trial will share the same CMFD operator while progressing with the Monte Carlo transport algorithm independently of other trials. The CMFD node aggregates FET coefficients across all ensembles, so stationarity can be diagnosed based on the aggregate source across all ensembles of neutrons per generation as well. In doing so, it will be argued that such an approach

allows for increased parallel efficiency and a stricter stationarity criteria, while continuing to enable real variance calculations across multiple independent trials.

Section 4.2.1 discusses the necessary background in process communication in order to explain how an ensemble averaged CMFD run strategy can be implemented. Section 4.2.2 introduces the bulk synchronous ensemble averaged CMFD run strategy, where all process communication with the CMFD node is carried out in a synchronous manner across all MC nodes. Section 4.2.3 then goes on to propose an asynchronous ensemble averaged CMFD run strategy, where each MC node communicates with the CMFD node without having to synchronize with other MC nodes, in an attempt to significantly reduce communication overhead. Finally, Section 4.3 evaluates the ensemble averaged CMFD run strategies on the merits of overall runtime to stationarity and any potential variance biases introduced.

4.2.1 Background on Remote Memory Access

In order to understand how ensemble averaged CMFD run strategies can be implemented, some discussion about processor-to-processor communication is pertinent, as this is the core mechanism through which CMFD calculations and FET stationarity diagnosis are offloaded to a separate compute node in a parallelized manner. Specifically, CMFD tally data and FET coefficients from each MC node need to be communicated to the CMFD node, while the updated CMFD source and the signal for whether active generations should begin need to be communicated from the CMFD node back to the MC node. Since this communication model is based on distributed memory, any updates that occur to a variable on one process will not be observed by another process and instead this updated variable needs to be communicated to other processes through MPI calls. The MPI implementation allows for such communication to occur in multiple ways, and will be summarized in the following paragraphs.

Firstly, blocking communication allows for point-to-point data communication between two processes, where one process sends data and the other process receives this data. Calls such as `MPI_Send` and `MPI_Recv` achieve this sort of communication pattern,

but an important point to note here is that these function calls do not return until the communication has been finished between both processes. What this means is that if process *A* is not ready to send or receive data while process *B* is, then process *B* will be in an idle state and will be unable to progress with any further function calls until process *A* has reached the part of the code where it is ready to communicate data. While blocking communication allows for the transfer of data in a predictable and replicable manner, it also creates communication bottlenecks in parallel programs, especially when processes have to wait a considerable time for communication to be established.

To reduce this overhead, non-blocking communication provides an alternate pathway for point-to-point communication, where processes are now allowed to return from function calls instead of having to wait in an idle state until the other process is ready to communicate. MPI calls such as `MPI_Isend` and `MPI_Irecv` initialize send/receive requests but do not complete them, and this allows for processes to initiate other computational tasks while they wait for processes to become available to communicate. For example, if process *B* is ready to communicate but process *A* is not, process *B* can periodically check on process *A* to see if it is ready to communicate at a later time through calls such as `MPI_Status`, and process *B* can go and do other work while it waits for process *A* to reach a point where it is ready to communicate.

Ultimately, however, non-blocking communication requires both sender and receiver to be free to communicate before data can actually be transferred. For Monte Carlo simulations where the completion time of particle transport routines is stochastic in nature, this sort of two-way communication can lead to undesirable behavior where processes have to continually check if other processes are ready to communicate. Furthermore, a process that is waiting to communicate with another process may not have any meaningful work to occupy itself with while it waits for the other process to participate in data communication and will therefore be left in an idle state, much like the model for blocking communication.

Instead, one-way communication allows for the transfer of data from a target process to a remote process without explicit synchronization with the target process. In such a communication model, a portion of a process memory is made accessible to all other

processes by the target process at the beginning of the program, and remote processes are allowed to access and modify this data without the approval of the target process. The function call `MPI_Get` allows for the one-sided retrieval of remote memory, while `MPI_Fetch_and_op` performs a one-sided read-modify-write on remote memory[5]. In order to avoid data races where one process tries to read or modify shared data while another process also modifies this data, data locks need to be appropriately acquired and released so that only one process handles this shared memory at a time.

Therefore, blocking communication will be the primary mode of data transfer for the bulk synchronous ensemble averaged CMFD case, while one-sided communication through RMA will be used for the asynchronous case. For both ensemble averaged CMFD cases, the global MPI commmunicator `MPI_COMM_WORLD` is split into $S + 1$ equivalent communicators corresponding to S MC nodes and an additional communicator for the CMFD node. It should be noted here that while each compute node could be comprised of multiple MPI processes, for simplicity, it is assumed that all MPI calls that occur between MC nodes and the CMFD node happen exclusively on the master processes of these node communicators. For scenarios where compute nodes comprise of multiple MPI processes, extra broadcasting steps are required to communicate remote memory data such as the CMFD source to all helper MPI processes on an MC node. Additionally, all other MPI calls or OpenMP multithreading routines that would typically occur in a Monte Carlo transport algorithm remain exactly the same on MC nodes.

4.2.2 Implementation of Bulk Synchronous Ensemble Averaged CMFD

Algorithms 4-1 and 4-2 summarize the workflow for the MC nodes and CMFD node respectively in the bulk synchronous ensemble averaged CMFD implementation. After each OpenMC generation completes on the MC node, CMFD tallies and FET coefficients from each MC node are transferred to the CMFD node through blocking MPI send operations, and the CMFD node initiates the corresponding receive calls such that it expects to receive this data from MC Node process 1 first, MC node process 2 next, and

[5]Here, op refers to the operation to perform on the remote memory to modify it. Only basic operations such as `MAX`, `MIN`, `SUM`, `PROD`, `REPLACE` etc. are allowed

so on until finally it receives data from MC node process S. Only until all data has been transferred from all MC node processes to the CMFD node can the CMFD equations be solved. The flag `MPI_ANY_SOURCE` can be used by the CMFD node so that it can handle MPI messages in the order they are sent by the MC nodes instead of processing them sequentially by process ID. However, each MC node process will ultimately have to wait for all other MC node process to finish their next generations in order to proceed with the CMFD update step, since each MC nodes also invokes a synchronous receive call to receive the updated CMFD fission source data from the CMFD node. This source data is computed only once the CMFD node receives tally data from all MC nodes. On the CMFD node, a similar windowing scheme is implemented as described in Section 3.2.3.2, where the window size is updated based on the number of generations of banked CMFD tallies across all MC nodes. Once the CMFD node solves the CMFD equations, it also processes the FET coefficients it receives from all MC nodes to determine whether stationarity has been met, and it then sends both the CMFD source and the signal to begin active generations to all MC nodes through blocking communication. The transferred CMFD source is used to reweight the local fission bank on each MC node. If stationarity has not been met, then each MC node will proceed to running another inactive generation of OpenMC. Otherwise, each MC node will begin its active generations.

In this manner, a bulk synchronous ensemble averaged CMFD approach allows for each MC node to progress independently of each other, while sharing the same CMFD operator during the CMFD feedback step that aggregates CMFD tally data across all MC nodes. Moreover, the CMFD node aggregates FET coefficient data across all Monte Carlo trials and therefore stationarity diagnosis will be based on the aggregate source across all available source particles in a generation. Such a run strategy thus combines the ability to compute variance estimates across parallel ensembles while determining stationarity on a stricter convergence criteria. However, the synchronous send and receive calls between the MC nodes and CMFD node pose significant communication bottlenecks, and the asynchronous ensemble averaged CMFD run strategy proposed in Section 4.2.3 serves to reduce this overhead.

Algorithm 4-1: MC Node Algorithm for Bulk Synchronous Ensemble Averaged CMFD

Let `proc_id` = process id's of master process for each MC node, mapped from `proc_id` = 1...S
Let `proc_id` = 0 refer to process id of master process for CMFD node
Let F = number of FET coefficients computed at each generation by MC node
Let C = length of CMFD tally data communicated from MC node to CMFD node
Let M = total number of mesh cells (spatial × energy) in CMFD problem

if `proc_id` $\neq 0$ **then** ▷ MC Node
    `begin_active` ← **false**
    Initialize `fet_coeffs` as vector of length F
    Initialize `cmfd_tallies` as vector of length C
    loop
        Run next generation of OpenMC
        Update `cmfd_tallies` with tally data from most recent generation
        Update `fet_coeffs` with FET data from most recent generation
        Send `cmfd_tallies` to process 0 through blocking communication
        Send `fet_coeffs` to process 0 through blocking communication
        Receive `cmfd_source` from process 0 through blocking communication
        Reweight fission bank based on `cmfd_source` ▷ Eqn. 3.3 to 3.5
        Receive `begin_active` from process 0 through blocking communication
        if `begin_active` **then** ▷ Begin active generations
            exit loop
        end if
    end loop
end if

4.2.3 Implementation of Asynchronous Ensemble Averaged CMFD

In the asynchronous ensemble averaged CMFD approach, one-sided communication through RMA is used to facilitate all communication between each MC node and the CMFD node. Algorithms 4-3 and 4-4 describe the workflow for the MC node and CMFD node respectively for this run strategy. All variables that are underlined are defined as shared variables that can be accessed and modified through RMA. As described in Section 4.2.1, data locks need to be acquired and subsequently released when these variables are accessed or modified in order to prevent race conditions. On the MC node, the FET coefficient variables and CMFD tally variables for each node are shared with the CMFD node. An additional shared variable `begin_active` is also introduced, which the CMFD node can directly modify to signal to each MC node to begin the active generations. On the CMFD node, the variable `cmfd_source` represents the CMFD fission source and

Algorithm 4-2: CMFD Node Algorithm for Bulk Synchronous Ensemble Averaged CMFD

Let `proc_id` = process id's of master process for each MC node, mapped from `proc_id` = 1...S
Let `proc_id` = 0 refer to process id of master process for CMFD node
Let F = number of FET coefficients computed at each generation by MC node
Let C = length of CMFD tally data communicated from MC node to CMFD node
Let M = total number of mesh cells (spatial × energy) in CMFD problem

if `proc_id` = 0 **then** ▷ CMFD Node
    Initialize `cmfd_source` as vector of length M
    Initialize `received_cmfd_tallies` as matrix of size $S \times C$
    Initialize `windowed_cmfd_tallies` as matrix of size $W \times S \times C$, where the size of W is determined by the windowing scheme
    Initialize `global_fet` as matrix of size $S \times F$
    `begin_active` ← **false** ▷ Has FET stationarity been met?
    while not `begin_active` **do**
        for $s \leftarrow 1$ **to** S **do** ▷ Get data from MC nodes
            Receive `cmfd_tallies` from process s through blocking communication, bank to `received_cmfd_tallies`[s]
            Receive `fet_coeffs` from process s through blocking communication, bank to `global_fet`[s]
        end for
        Update `windowed_cmfd_tallies` with `received_cmfd_tallies` according to windowing scheme
        Recompute CMFD MGXS from `windowed_cmfd_tallies` ▷ Table 3.2
        Recompute `cmfd_source` ▷ Eqn. 3.1, 3.3
        Update `begin_active` from `global_fet` ▷ Eqn. 2.11
        if `begin_active` **then**
            `begin_active` ← **true**
        else
            `begin_active` ← **false**
        end if
        for $s \leftarrow 1$ **to** S **do**
            Send `cmfd_source` to process s through blocking communication
            Send `begin_active` to process s through blocking communication
        end for
    end while
end if

is shared with all MC nodes, and the shared variable `finished_next_gen` allows each MC node to signal to the CMFD node that an MC node has finished simulating its next generation.

The algorithm for asynchronous ensemble averaged CMFD is loosely related to that of the bulk synchronous case, where all CMFD and FET stationarity calculations are offloaded to the CMFD node and each MC node only directly communicates with the CMFD node. At the end of each OpenMC generation, however, instead of facilitating two-way communication with the CMFD node to exchange data fields, the MC node with process ID p accesses the most current version of the CMFD source available on the CMFD node through RMA to reweight its fission source. It also signals to the CMFD node that new CMFD tally data and FET coefficients are available by setting `finished_next_gen`$[p]$ to true on the CMFD node.

On the CMFD node, once all source update calculations have been performed, the node keeps checking to see if any elements of `finished_next_gen` have been set to true. If so, the elements where `finished_next_gen` are set to true indicate which processes the CMFD node has to obtain CMFD tally and FET coefficient data fields from. Once these values have been retrieved through RMA and banked to global arrays `global_fet` and `windowed_cmfd_tallies`, all elements of `finished_next_gen` are reset to false. Then, the CMFD node goes on to recompute the CMFD source as well as whether FET stationarity has been met. While these calculations occur, other MC nodes could finish their next generations and thus the CMFD node repeats the loop once again by checking to see if any new elements of `finished_next_gen` have been set to true.

This asynchronous ensemble averaged CMFD is therefore implemented in such a way where time between CMFD update steps on the MC nodes is minimal. During each CMFD update step on the MC node, the master process obtains the most current version of the CMFD fission source available on the CMFD node and signals to the CMFD node to obtain the newly available data on the MC node. The CMFD node, on the other hand, is responsible for recomputing its shared values whenever new data is available, and does not have to concern itself with synchronizing with specific processes. It also signals to all MC node processes to begin active generations when stationarity has been met

by remotely modifying the variable `begin_active` on all MC nodes. While the newly computed CMFD tally information will not be incorporated into the CMFD source data that is retrieved from the CMFD node, the assumption here is that this lag does not affect the quality of the CMFD update step since the CMFD operator still incorporates tally data from previous generations across all other MC nodes as well. The relatively low cost of running CMFD compared to Monte Carlo also allows for quick CMFD updates in time for subsequent MC nodes that finish their next generation and require most up-to-date CMFD source data.

Once again, an expanding window update scheme is used for asynchronous ensemble averaged CMFD, where the window size grows exponentially to discard earlier generations of biased CMFD data while banking tally data that comes in at later generations. In this case, however, the sequence with which CMFD tallies comes to the CMFD node is not known with certainty anymore, so the window size grows in multiples of S once the CMFD node has undergone S new source update calculations, where S is the total number MC nodes running in parallel.

4.2.4 Performance Profiling for Ensemble Averaged CMFD

Run strategies 5 and 6 are now included to the list of potential ways to simulate large-scale Monte Carlo simulations, corresponding to the bulk synchronous and asynchronous ensemble averaged CMFD methodologies respectively. Both run strategy 5 and 6 simulate $S = 32$ trials of Monte Carlo in parallel with $S + 1 = 33$ total compute nodes to account for the additional CMFD node. These run strategies also use the same CMFD parameters as described in Table 4.2, and the updated table of compute resource parameters with run strategies 5 and 6 included are shown in Table 4.3.

Similar to Section 4.1.1, performance profiling is conducted on run strategies 5 and 6, and the profiling plots for the 1-D case are shown in Figures 4-10 and 4-11, while the same plots for the 2-D case are shown in Figures 4-12 and 4-13. The top row of these plots marked as "CMFD" shows the CPU utilization for the CMFD node. For the bulk synchronous case, it can be seen that CPU utilization is very low since this node

Algorithm 4-3: MC Node Algorithm for Asynchronous Ensemble Averaged CMFD
(All underlined variables are shared across processes through RMA and locks for these variables need to be acquired and released before and after any data reads/writes.)

Let `proc_id` = process id's of master process for each MC node, mapped from `proc_id` = 1...S
Let `proc_id` = 0 refer to process id of master process for CMFD node
Let F = number of FET coefficients computed at each generation by MC node
Let C = length of CMFD tally data communicated from MC node to CMFD node
Let M = total number of mesh cells (spatial × energy) in CMFD problem

if `proc_id` $\neq 0$ **then** ▷ MC Node
    `begin_active` ← **false**
    Initialize `fet_coeffs` as vector of length F
    Initialize `cmfd_tallies` as vector of length C
    loop
        Run next generation of OpenMC
        Update `cmfd_tallies` with tally data from most recent generation
        Update `fet_coeffs` with FET data from most recent generation
        `finished_next_gen`[`proc_id`] ← **true** for process 0 through RMA ▷ Signal to CMFD node that MC node has finished next generation
        Get `cmfd_source` from process 0 through RMA
        Reweight fission bank based on `cmfd_source` ▷ Eqn. 3.3 to 3.5
        if `begin_active` **then** ▷ Set to **true** by CMFD node
            exit loop
        end if
    end loop
end if

Algorithm 4-4: CMFD Node Algorithm for Asynchronous Ensemble Averaged CMFD (All underlined variables are shared across processes through RMA and locks for these variables need to be acquired and released before and after any data reads/writes.)

Let `proc_id` = process id's of master process for each MC node, mapped from `proc_id` = 1...S
Let `proc_id` = 0 refer to process id of master process for CMFD node
Let F = number of FET coefficients computed at each generation by MC node
Let C = length of CMFD tally data communicated from MC node to CMFD node
Let M = total number of mesh cells (spatial × energy) in CMFD problem

if `proc_id` = 0 **then** ▷ CMFD Node
    Initialize `finished_next_gen` to boolean vector of length S with all elements equal **false**
    Initialize `cmfd_source` as vector of length M
    Initialize `windowed_cmfd_tallies` as matrix of size $W \times C$, where the size of W is determined by the windowing scheme
    Initialize `global_fet` as matrix of size $S \times F$
    `begin_active` ← **false** ▷ Has FET stationarity been met?
    while not `begin_active` **do**
        if any element of `finished_next_gen` = **true then**
            for $s \leftarrow 1$ **to** S **do** ▷ Get data from MC nodes that finished next generation
                if `finished_next_gen` [s] **then**
                    Get `cmfd_tallies` from process s through RMA, bank to `windowed_cmfd_tallies`
                    Get `fet_coeffs` for process s through RMA, bank to `global_fet`[s]
                    `finished_next_gen`[s] ← **false** ▷ Reset value
                end if
            end for
            Recompute CMFD MGXS from `windowed_cmfd_tallies` ▷ Table 3.2
            Recompute `cmfd_source` ▷ Eqn. 3.1
            Update `begin_active` from `global_fet` ▷ Eqn. 2.11
            if `begin_active` **then** ▷ Send signal to all MC nodes to start active generations
                for $s \leftarrow 1$ **to** S **do**
                    `begin_active` ← **true** for process s through RMA
                end for
            end if
        end if
    end while
end if

Table 4.3: Compute resource parameters for run strategies 1 to 6

Run Strategy	Total Compute Nodes	Total Independent Trials	Generation Size per Trial	MPI Processes per Trial	OpenMP Threads per MPI Process
1 - Multiple Trial Monte Carlo	32	32	10 Million	2	18
2 - Multiple Trial CMFD	32	32	10 Million	2	18
3 - Single Trial Monte Carlo	32	1	320 Million	64	18
4 - Single Trial CMFD	32	1	320 Million	64	18
5 - Bulk Synchronous Ensemble Averaged CMFD	33	32	10 Million	2	18
6 - Asynchronous Ensemble Averaged CMFD	33	32	10 Million	2	18

is responsible for a single CMFD update only once all Monte Carlo (MC) nodes have communicated their data to the CMFD node. The rest of the time, the CMFD node is left in an idle state simply waiting for incoming tally data. Furthermore, the communication bottlenecks from blocking communication can be clearly seen on the MC nodes as well. Each trial, no matter when it finishes its generation, is stuck waiting for CMFD fission source data from the CMFD node, and this data is obtained only once the last trial to finish the next generation has transferred its tally data to the CMFD node. Only then is the CMFD source recomputed. This CPU downtime is indicated by the horizontal red lines that appear after each trial finishes a generation.

On the other hand, the asynchronous case has a higher CPU utilization on the CMFD node, since now this node recomputes the fission source whenever new data is available from any MC node that finishes its next generation. Moreover, each trial is now able to proceed to its next generation without having to synchronize with other trials, as is indicated by the visible reduction in red horizontal lines in the profiling plots. This indicates that the CMFD update step is conducted with very little latency, as each node simply has to invoke an RMA call to retrieve source data in order to reweight its fission source weights. Even in the asynchronous case, the CMFD node still has a considerable amount of downtime waiting for newly available tally data, which indicates that scaling up the number of MC nodes in the problem could lead to a higher CPU utilization by the CMFD node.

The profiling plots shown here and in Section 4.1 were presented in order to provide a graphical illustration of parallel communication bottlenecks and per-batch runtimes among the six different run strategies introduced so far. However, Section 4.3 will take a closer look at overall runtime to stationarity as well as any potential variance biases in order to assess these run strategies on more holistic grounds.

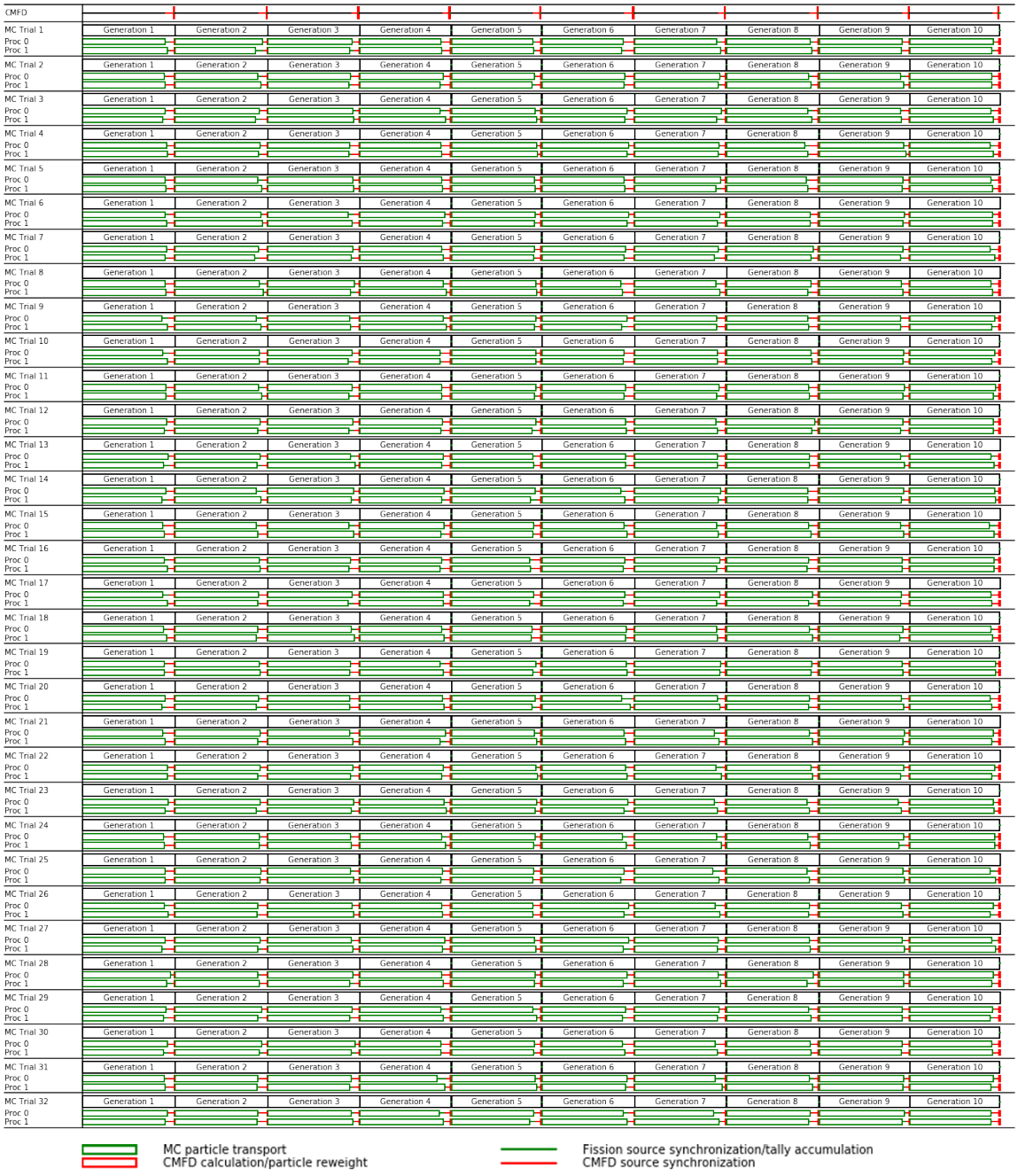


Figure 4-10: Performance profiling plot for run strategy 5 on the 1-D homogeneous problem. The green portions indicate time spent in the Monte Carlo routine, while the red portion indicates time spent in the CMFD feedback step. Green boxes represent when processes are running particle transport at each generation, while horizontal green lines indicate when processes are stalling while synchronizing the fission source and accumulating global tallies. Likewise, red boxes indicate when CMFD calculation is occurring on the master process or when particle reweighting occurs on each process, while red horizontal lines signify when processes are stalled due to synchronization of CMFD parameters.

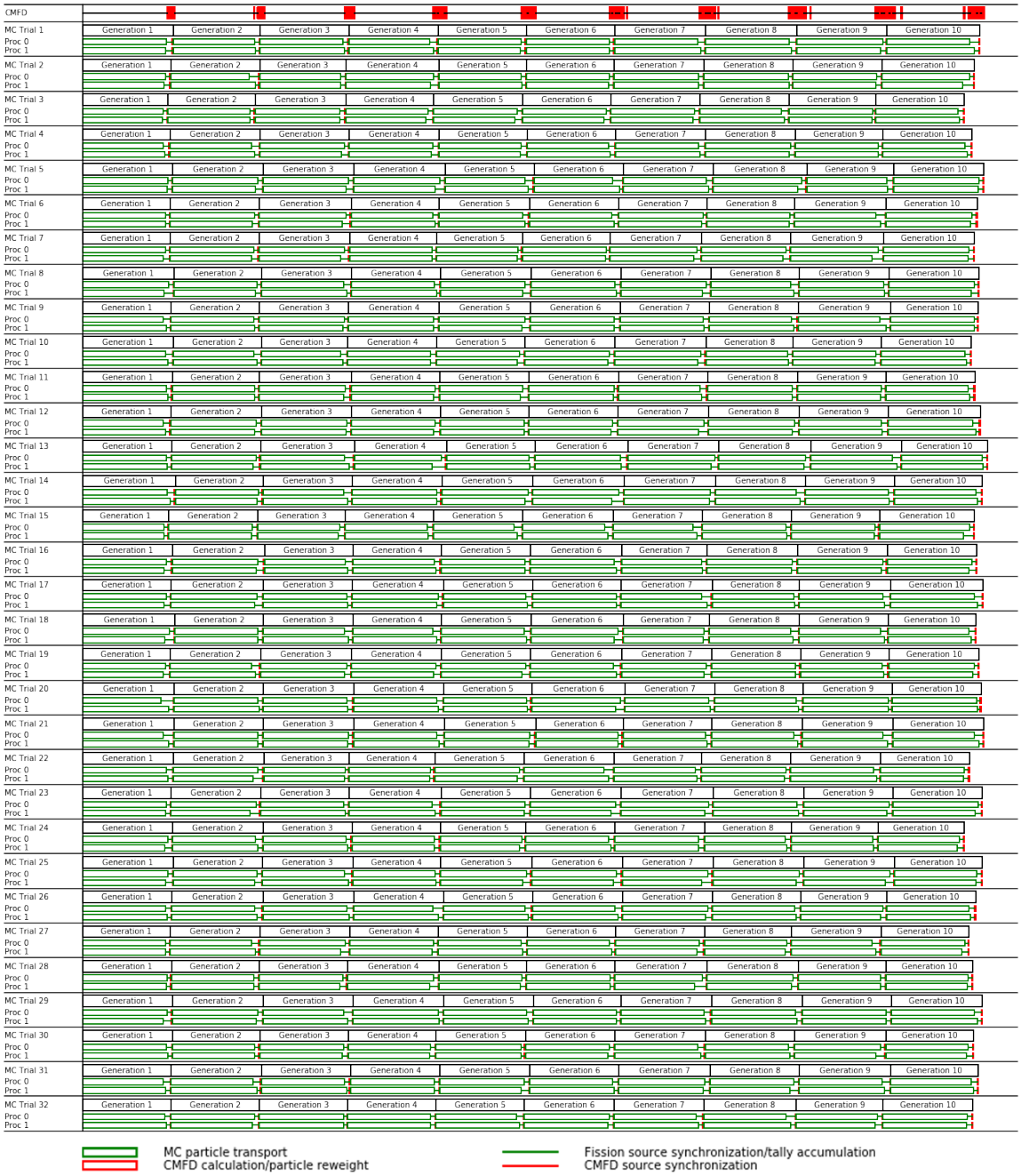


Figure 4-11: Performance profiling plot for run strategy 6 on the 1-D homogeneous problem. The green portions indicate time spent in the Monte Carlo routine, while the red portion indicates time spent in the CMFD feedback step. Green boxes represent when processes are running particle transport at each generation, while horizontal green lines indicate when processes are stalling while synchronizing the fission source and accumulating global tallies. Likewise, red boxes indicate when CMFD calculation is occurring on the master process or when particle reweighting occurs on each process, while red horizontal lines signify when processes are stalled due to synchronization of CMFD parameters.

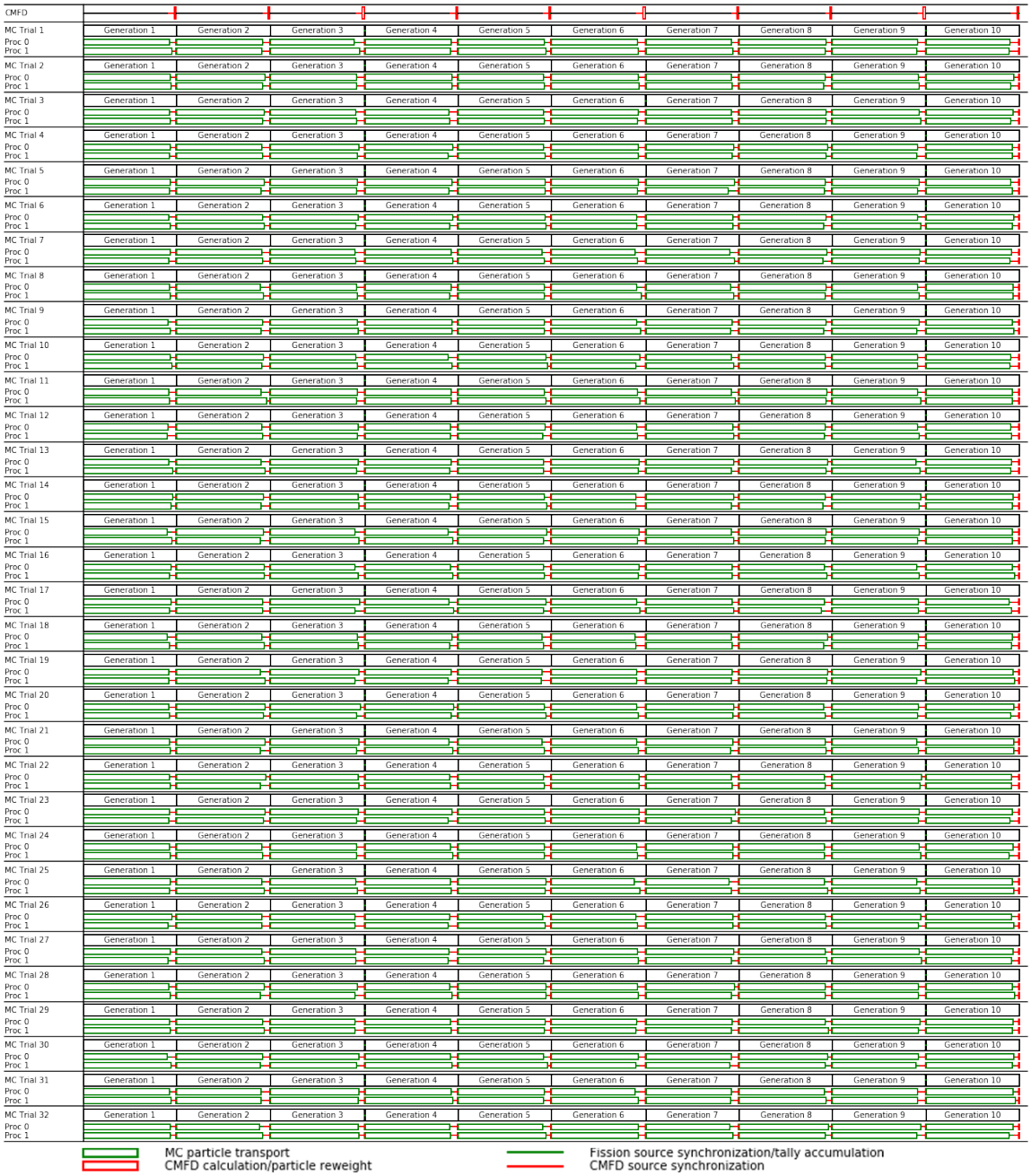


Figure 4-12: Performance profiling plot for run strategy 5 on the 2-D BEAVRS problem.

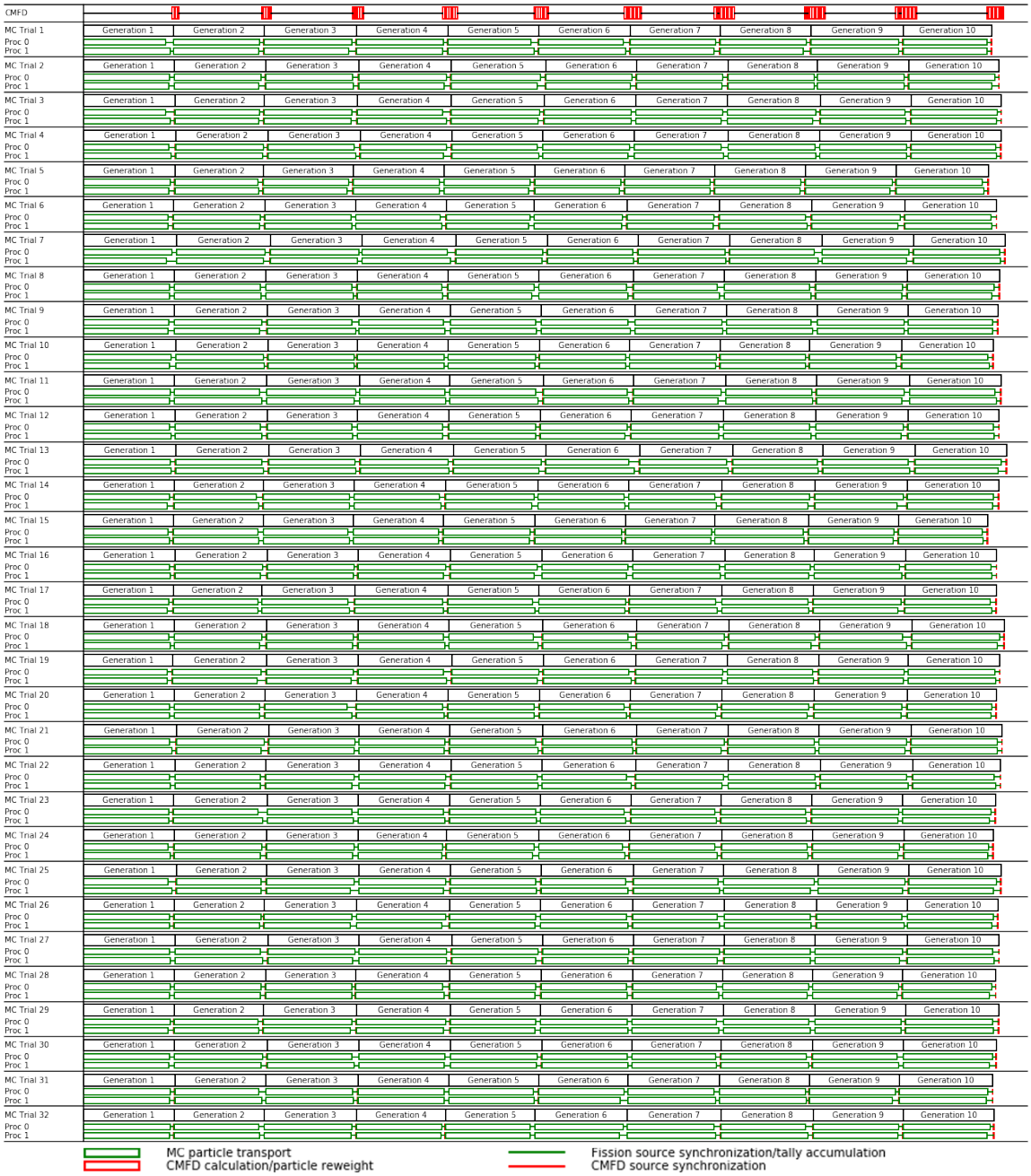


Figure 4-13: Performance profiling plot for run strategy 6 on the 2-D BEAVRS problem.

An ensemble averaged CMFD run strategy offloads all CMFD and FET stationarity calculations to a separate compute node while still running ensembles of Monte Carlo in parallel. The bulk synchronous implementation communicates CMFD and FET data back and forth between MC nodes and the CMFD node using blocking two-sided communication, while the asynchronous implementation uses one-sided RMA to communicate this data with smaller communication overhead.

4.3 Runtime Analysis and Variance Bias for Ensemble Averaging

Section 4.2 introduced the bulk-synchronous and asynchronous implementations of the ensemble averaged CMFD run strategies and compared the performance profiling plots of these run strategies to those presented in Section 4.1. These performance plots illustrate parallel performance and runtime for a fixed number of generations. However, this section looks at overall runtime for run strategies 1 to 6 by accounting for total generations to stationarity. Additionally, this section examines whether ensemble averaged CMFD causes any potential variance biases that stem from the fact that the same CMFD operator is shared across all MC nodes. For these ensembles of Monte Carlo trials to be characterized as statistically independent, it must be shown that the use of a shared CMFD operator during the CMFD feedback step does not introduce biases to the reported mean and variance levels of these ensembles.

As described in Section 4.1.2, the use of a single trial stopping criteria results in a less stringent stationarity criteria compared to the use of an aggregate trial stopping criteria. Therefore, Section 4.3.1 describes a methodology for applying a single trial stopping criteria onto the ensemble averaged CMFD strategy that uses an aggregate trial stopping criteria. Once this procedure is established, runtimes between different ensemble averaging approaches that use different stopping criteria can be compared on the same footing. Section 4.2.3 then summarizes the overall runtime and real variance

calculations across run strategies 1 to 6, and finally in Section 4.3.3, the ensemble averaged CMFD approach is compared to the case where a single trial is run with a small generation size and a large number of active generations.

4.3.1 Comparing the Variance Between Multiple Trial Ensemble Averaging and Bulk Synchronous Ensemble Averaged CMFD Simulations

Run strategies 1 and 2 use a single trial stopping criteria to determine stationarity and this results in fewer generations to stationarity but also a higher variance level in the stationary fission source. On the other hand, the ensemble averaged CMFD run strategies 5 and 6 bank FET coefficients across all MC nodes and thus determine stationarity based on the aggregate fission source. For ensemble averaged run strategies to be compared to the multiple trial ensemble averaged approaches on the same grounds, a procedure for providing an apples-to-apples comparison between these run strategies must be devised. Specifically, since the bulk synchronous implementation synchronizes the FET coefficients after each generation across all MC nodes, the fluctuations in the aggregated coefficients can be post-processed and compared to the fluctuations in the FET coefficients for a single ensemble of neutrons in the multiple trial ensemble averaging approach.

For instance, Figure 4-14 plots the standard deviation in the instantaneous FET coefficient for the a_3 coefficient for a single ensemble of neutrons in run strategy 1 (Multiple trial Monte Carlo with no CMFD) along with the standard deviation in the average a_3 coefficient across all Monte Carlo trials in run strategy 5 (bulk-synchronous ensemble averaged CMFD). Here, the a_3 FET coefficient represents the slowest converging FET coefficient for the 1-D homogeneous problem, and the standard deviation in the instantaneous FET coefficient is defined as

$$s[a_{n,m}] = \sqrt{\frac{1}{G-1}\sum_{g=m}^{m+G}\left(a_{n,g} - \overline{a}_{n,m:m+G}\right)^2} \tag{4.4}$$

for the single ensemble case, where $a_{n,g}$ represents the value of the a_n FET coefficient

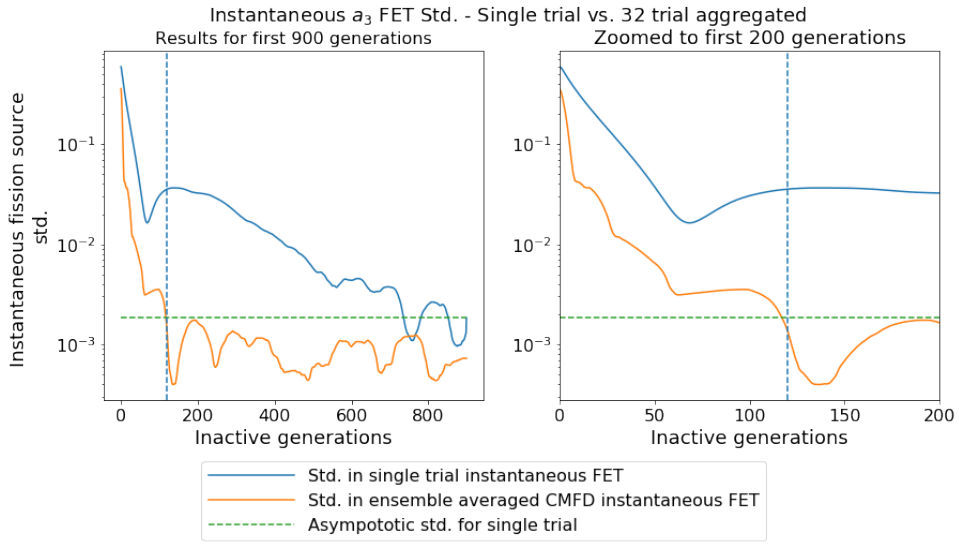


Figure 4-14: Standard deviation in instantaneous a_3 FET coefficient the 1-D homogeneous problem. The blue line plots the single trial standard deviation for run strategy 1, the orange line plots the 32-trial standard deviation for run strategy 5, the horizontal dashed green plots the asymptotic standard deviation, and the vertical blue line plots the generation where the orange line crosses the asymptotic dashed green line. The right plot is the zoomed version of the left plot.

at generation g, and $\overline{a}_{n,m:m+G}$ represents the average FET coefficient computed over a moving window of $G = 100$ generations starting at generation m. For the ensemble averaged CMFD case where FET coefficients over $S = 32$ ensembles are aggregated, the standard deviation in the instantaneous FET is defined as

$$s[a_{n,m}] = \sqrt{\frac{1}{G-1}\sum_{g=m}^{m+G}\left(\frac{1}{S}\left[\sum_{s=1}^{S}(a_{n,g,s}) - \sum_{s=1}^{S}(\overline{a}_{n,m:m+G,s})\right]\right)^2} \tag{4.5}$$

where now the term $a_{n,g,s}$ represents the value of the a_n FET coefficient at generation g and trial s, and $\sum_{s=1}^{S}\overline{a}_{n,m:m+G,s}/S$ represents the trial-averaged a_n FET coefficient over a moving window of $G = 100$ generations starting at generation m. In Figure 4-14, the horizontal green dashed line represents the asymptotic variance in the FET coefficient when a single ensemble is used to compute the standard deviation in the instantaneous FET coefficient. For the ensemble averaged CMFD case where the average of 32 ensembles is used to compute the standard deviation in the instantaneous FET coefficient, the asymptotic standard deviation level is $1/\sqrt{32}$ times lower, and the

generation at which this orange line intersects the green line represents the point at which the bulk synchronous ensemble averaged CMFD possesses the same standard deviation level as that dictated by a single trial stopping criteria. This generation number is indicated by the blue vertical line, and this occurs at generation 120 for the 1-D homogeneous problem. This is the stopping criteria used for run strategy 5 when comparing runtime results for run strategies 1 and 2, where a looser stopping criteria based on the stationarity behavior of a single ensemble of 10 million neutrons is applied to strategies 1, 2, and 5. The analogous plot to Figure 4-14 for the 2-D BEAVRS case is shown in Figure 4-15, and the single trial stopping criteria for run strategy 5 now occurs at generation 19. These plots show that the presence of $S = 32$ ensembles towards the computation of instantaneous FET coefficient standard deviation yields a much quicker convergence to a single trial stopping criteria compared to the case when only $S = 1$ ensemble of data is available.

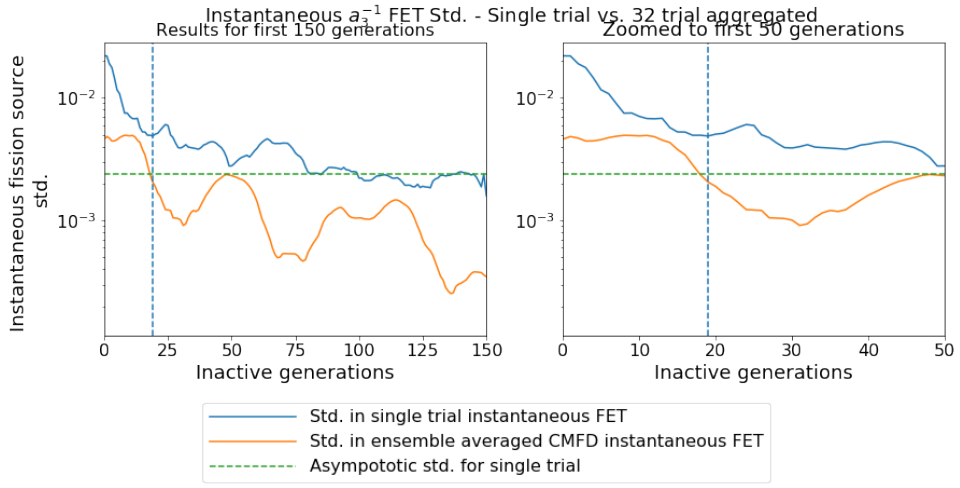


Figure 4-15: Standard deviation in instantaneous a_3^{-1} FET coefficient the 2-D BEAVRS problem. The blue line plots the single trial standard deviation for run strategy 1, the orange line plots the 32-trial standard deviation for run strategy 5, the horizontal dashed green plots the asymptotic standard deviation, and the vertical blue line plots the generation where the orange line crosses the asymptotic dashed green line. The right plot is the zoomed version of the left plot.

4.3.2 Runtime Results and Variance Bias

This section presents the overall runtime results based on generations to stationarity for the six run strategies presented so far. The RMS error and relative uncertainty levels associated with the ensemble averaged CMFD cases will also be compared to the cases where ensemble averaging is not utilized in order to show any potential mean or standard deviation biases that may arise from the presence of a shared CMFD operator. Here, RMS error is calculated using Equation 3.8 and relative uncertainty is defined as the mesh-averaged ratio of the sample standard deviation in the fission source across P independent trials to the average fission source across P independent trials:

$$\begin{aligned} \epsilon_n^{\text{rel.}} &= \frac{1}{M}\sum_{m=1}^{M}\frac{\sqrt{\frac{1}{P-1}\sum_{p=1}^{P}(Y_{m,n,p}-\overline{Y}_{m,n})^2}}{\overline{Y}_{m,n}} * 100\% \\ &= \frac{1}{M}\sum_{m=1}^{M}\sqrt{\frac{1}{P-1}\sum_{p=1}^{P}\left(\frac{Y_{m,n,p}-\overline{Y}_{m,n}}{\overline{Y}_{m,n}}\right)^2} * 100\% \end{aligned} \tag{4.6}$$

where $Y_{m,n,p}$ is the accumulated fission source at active generation n, mesh cell m, and trial p, and $\overline{Y}_{m,n}$ is the trial-averaged accumulated fission source at active generation n and mesh cell m. All accumulated fission source values in this section are computed over a single active generation.

As described in Section 4.3.1, run strategies that utilize a single trial stopping criteria cannot be compared directly to the run strategies that employ an aggregate trial stopping criteria, so runtime results will be separated by stopping criteria type. Table 4.4 and Table 4.5 report the runtime results for the 1-D and 2-D problems respectively with run strategies that utilize a single trial stopping criteria, and the results for the bulk synchronous ensemble averaged CMFD from applying a single trial stopping criteria as described in Section 4.3.1 are shown as well. Similarly, Table 4.6 and Table 4.7 summarize the runtime results for the run strategies that utilize an aggregate trial stopping criteria for the 1-D and 2-D problems respectively. For these problems, the mesh-averaged RMS error levels and relative uncertainty levels as established by the

aggregate trial stopping criteria are calculated as well. The mesh over which these quantities are computed are 20cm for the 1-D case and an assembly-sized mesh for the 2-D case. Also, for the single trial simulations, multiple trials were conducted in order to compute relative uncertainties since these run strategies do not have a straightforward way to compute accurate standard deviation levels otherwise.

Table 4.4: 1-D runtime results for run strategies that utilize a single trial stopping criteria

Run Strategy	Trial-Averaged Generations to Stationarity	Total Computational Time (CPU-hrs)
1 - Multiple Trial Monte Carlo	874.4	5903.06
2 - Multiple Trial CMFD	215.3	2975.30
5 - Bulk Synchronous Ensemble Averaged CMFD	120	1772.02

Table 4.5: 2-D BEAVRS runtime results for run strategies that utilize a single trial stopping criteria

Run Strategy	Trial-Averaged Generations to Stationarity	Total Computational Time (CPU-hrs)
1 - Multiple Trial Monte Carlo	161.2	4071.91
2 - Multiple Trial CMFD	70.5	2256.8
5 - Bulk Synchronous Ensemble Averaged CMFD	19	658.27

For the results in this section, the total computational time for a run strategy, in CPU-hours, is computed as

$$\text{Total computational time} = \sum_{n=1}^{N_{\text{node}}} T_{\text{stationarity},n} N_{\text{CPU},n} \tag{4.7}$$

where N_{node} represents the total compute nodes in the simulation, $T_{\text{stationarity},n}$ represents the total time to stationarity for the OpenMC trial running on compute node n, and $N_{\text{CPU},n}$ represents the total number of CPU's running on node n. For multiple trial ensemble averaging, total time to stationarity is trial-dependent, while time to stationarity is the same across all compute nodes for both the single trial and ensemble averaged CMFD cases since stationarity is determined based on the global instantaneous fission source across all trials. For the ensemble averaged CMFD case, an additional compute node is included as part of N_{node} to account for the CMFD node.

Table 4.6: 1-D runtime, relative uncertainty, and RMS error results for run strategies that utilize an aggregate single trial stopping criteria.
*32 Multiple independent trials were conducted to generate statistics on single trial simulations.

Run Strategy	Trial-Avg. Gens. to Stationarity	Total Comp. Time (CPU-hrs)	Rel. Uncertainty (%)	RMS Error (%)
3 - Single Trial Monte Carlo w/ Large Gen.	1119	8008.8	0.113*	0.115
4 - Single Trial CMFD w/ Large Gen.	235	3400.44	0.115*	0.114
5 - Bulk Synchronous Ensemble Averaged CMFD	234	3450.56	0.114	0.112
6 - Asynchronous Ensemble Averaged CMFD	232.6	3210.70	0.114	0.113

These results illustrate the benefits and drawbacks of different Monte Carlo run strategies on overall Monte Carlo runtime. Firstly, the use of CMFD results in a significantly larger reduction in runtime compared to the un-accelerated case, due to the fact that source acceleration reduces generations to stationarity by a larger percentage than the computational overhead per generation associated with running CMFD. These results are consistent with the conclusions from Chapter 3.

Secondly, a single trial stopping criteria results in a lower total runtime compared to the aggregate trial stopping criteria, but this is due to the fact that fewer generations

Table 4.7: 2-D runtime, relative uncertainty, and RMS error results for run strategies that utilize an aggregate trial stopping criteria.
*32 Multiple independent trials were conducted to generate statistics on single trial simulations

Run Strategy	Trial-Avg. Gens. to Stationarity	Total Comp. Time (CPU-hrs)	Rel. Uncertainty (%)	RMS Error (%)
3 - Single Trial Monte Carlo w/ Large Gen.	192	5105.78	0.170*	0.171
4 - Single Trial CMFD w/ Large Gen.	90	2979.76	0.170*	0.171
5 - Bulk Synchronous Ensemble Averaged CMFD	89	3081.60	0.171	0.172
6 - Asynchronous Ensemble Averaged CMFD	89.1	2725.76	0.171	0.170

to stationarity are run, and this leads to a looser stationarity criteria. When applying the single trial stopping criteria onto the bulk synchronous ensemble averaged CMFD case, the total runtime is lower than the multiple trial ensemble averaging approach due to the fact that a significantly lower number of generations are required to reach this equivalent stopping criteria. Here, the presence of $S = 32$ times more ensembles reduces fission source variance at a much faster rate than if only a single ensemble of data is used to determine stationarity. As shown by Figure 4-9, this single trial stopping criteria also represents a higher error level than the aggregate trial stopping criteria, as an insufficient number of generations are simulated in the inactive generations commensurate to the terminal variance level attainable by the aggregate source across all ensembles.

Finally, the asynchronous ensemble averaged CMFD case most effectively reduces overall computational time to stationarity when comparing the various strategies that utilize an aggregate trial stopping criteria. This is because the asynchronous implementation minimizes communication bottlenecks between Monte Carlo generations as there are no longer any blocking MPI calls between processes. This results in a roughly 5% reduction in overall runtime compared to the single trial case with a large generation

size. Moreover, there is no significant bias in reported statistics across the different run strategies, and this is likely due to the fact that the same overall generation size and maximum CMFD window size was used across all run strategies, which were the two factors shown in Chapter 3 to affect instantaneous fission source variance levels. It should be noted once again that single trial run strategies 3 and 4 have no direct way of computing accurate relative uncertainty and standard deviation levels, and any true measures of these statistical quantities need to be obtained by running multiple independent simulations separately.

4.3.3 Comparing Asynchronous Ensemble Averaged CMFD to Single Trial Simulations with Small Generation Sizes

As a final simulation scheme for comparison to the ensemble averaged CMFD approach, this section looks at the efficacy of running a single trial simulation with a small generation size and a large number of active generations. Keeping the total runtime from run strategy 5 with a single trial stopping criteria fixed, the same test cases are now simulated with the same computational resources but with a single trial of 10 million neutrons. Once the single trial stopping criteria is met, the remaining runtime is devoted towards simulating as many active generations as possible. Since this approach only simulates a single trial, a batching method is used to generate relative uncertainty estimates, and these levels are compared with the relative uncertainty levels as computed by the ensemble averaged CMFD approach with multiple independent trials and a single trial stopping criteria. Thus, run strategies 7 and 8 are introduced as single trial Monte Carlo and single trial CMFD respectively with a small generation size, and Table 4.8 summarizes the compute resource parameters for these run strategies along with the previous 6 run strategies introduced so far.

Figure 4-16 plots the relative uncertainty as predicted by run strategies 5, 7, and 8 where a single trial stopping criteria is used to delineate inactive and active generations. Total runtime is kept fixed among all three run strategies, and multiple independent trials are conducted for each run strategies to get a sense of how much these standard

Table 4.8: Compute resource parameters for run strategies 1 to 8

Run Strategy	Total Compute Nodes	Total Independent Trials	Generation Size per Trial	MPI Processes per Trial	OpenMP Threads per MPI Process
1 - Multiple Trial Monte Carlo	32	32	10 Million	2	18
2 - Multiple Trial CMFD	32	32	10 Million	2	18
3 - Single Trial Monte Carlo	32	1	320 Million	64	18
4 - Single Trial CMFD	32	1	320 Million	64	18
5 - Bulk Synchronous Ensemble Averaged CMFD	33	32	10 Million	2	18
6 - Asynchronous Ensemble Averaged CMFD	33	32	10 Million	2	18
7 - Single Trial Monte Carlo	32	1	10 Million	64	18
8 - Single Trial CMFD	32	1	10 Million	64	18

deviation levels vary through repeated trials. Firstly, run strategy 7 (single trial Monte Carlo) resulted in 3400 active generations, which were split into 17 batches with 200 generations per batch, while run strategy 8 (single trial CMFD) yielded 3000 active generations that were split into 15 batches with 200 generations per batch. The value for generations per batch was chosen based on recommendations by Kelly et. al to ensure independence between batches. The first observation from this plot is the higher relative uncertainty levels when running CMFD compared to the un-accelerated case. This has to do with the overhead associated with CMFD source acceleration in the active generations, which results in fewer available active generations to compute batch statistics.

Secondly, the ensemble averaged CMFD case results in the lowest overall relative uncertainty levels, which illustrates the point that an equivalent run strategy that simulates

a single trial with a large number of active generations still results in a higher relative uncertainty due to correlation effects that degrade convergence rates in the initial active generations.

Finally, when comparing the fluctuations in relative uncertainty by trial, it can be seen that the use of multiple trial statistics through ensemble averaging results in a much narrower confidence interval than through the use of batching. This is because all particle histories in a single generation across multiple independent trials are used to generate statistical quantities. On the other hand, generational quantities are averaged over a large number of generations when splitting active generations into batches, so the number of tallied quantities associated with a batching method is based on the total number of batches available (17 batches for the un-accelerated case and 15 batches for the case with CMFD acceleration), which is markedly smaller than the case where multiple independent trials are conducted. This is consistent with the findings by Miao, who shows that the uncertainty in a generation-based variance estimator is lower than that of a batch-based variance estimator [28].

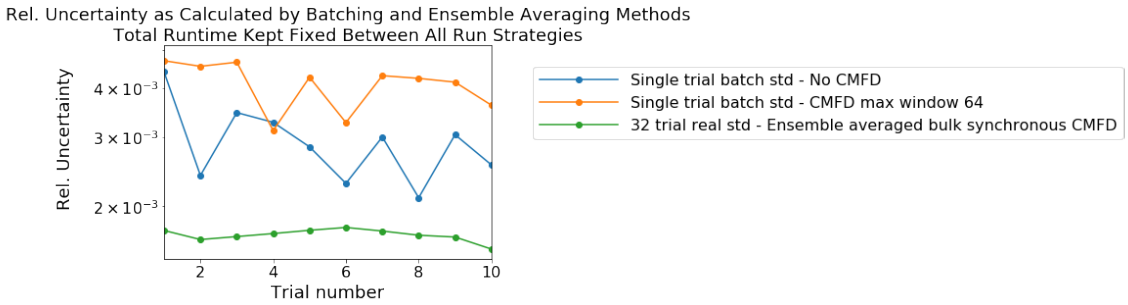


Figure 4-16: Relative uncertainty as predicted by multiple independent trials of bulk synchronous ensemble averaged CMFD and by the batching method using a single trial with a large number of active generations. Runtime is kept fixed across all run strategies and multiple independent trials are conducted for each of the three run strategies shown, and results are shown for the 1-D homogeneous problem.

For the 2-D BEAVRS case, the use of batch statistics is complicated by the fact that now only 750 active generations are available for run strategy 7 with no CMFD, and only 375 active generations are available for run strategy 8 with CMFD. Thus, using a value of 200 for the number of generations per batch results in an insufficient number of samples to generate batch statistics, and a smaller number of generations per batch

must be used. In this specific case, the active generations are split into 10 batches with 75 generations per batch for the un-accelerated case, while the CMFD accelerated case is split into 5 batches of 75 generations per batch. Now, however, the batches cannot be assumed to be independent anymore as there is a non-negligible level of correlation associated with an inter-generational lag of 75. Thus, relative uncertainty levels are calculated assuming correlated batches, and these uncertainty levels using single trial batching and multiple trial ensemble averaging are shown in Figure 4-17. The details of how standard deviation levels are corrected to account for inter-batch correlation is outlined in Appendix F. Once again, a simulation strategy that utilizes a single trial with a small generation size yields uncertainty levels that are higher and more dispersed than the case where ensemble averaged CMFD is used. Stated another way, for the batching method to yield the same uncertainty as the ensemble averaged CMFD case, a larger number of active generations are required, thus increasing overall runtime. Assuming independent batches, ∼50 more batches of 200 generations per batch are needed for the 1-D homogeneous case and ∼17 more batches of 75 generations per batch are required for the 2-D BEAVRS case in order for relative uncertainties between the un-accelerated single trial batching method and ensemble averaged CMFD approach to reach parity.

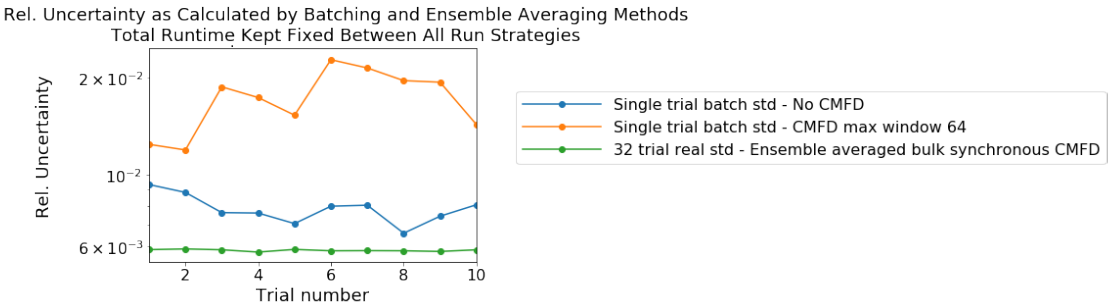


Figure 4-17: Relative uncertainty as predicted by multiple independent trials of bulk synchronous ensemble averaged CMFD and by the batching method using a single trial with a large number of active generations. Runtime is kept fixed across all run strategies and multiple independent trials are conducted for each of the three run strategies shown, and results are shown for the 2-D BEAVRS problem. 75 generations per batch are used so the standard deviation levels shown in the blue and orange lines correct for the correlation levels that are present between batches.

The various run strategies presented so far are compared in terms of overall computational time to stationarity. Multiple trial ensemble averaging results in higher runtime compared to the bulk synchronous ensemble averaged CMFD when an apples-to-apples single trial stopping criteria is applied. For single trial simulations with small generations, a larger variance level resulted when compared to the bulk synchronous ensemble averaged CMFD case where total runtime was kept the same across all strategies. Finally, for single trial simulations with large generation sizes, runtimes were roughly 5% higher when compared to runtimes for the asynchronous ensemble averaging case. It was also shown that there was no significant evidence of variance bias when using a shared CMFD operator for the ensemble averaged CMFD run strategy.

4.4 Summary of Arguments for Ensemble Averaging

This chapter proposed a multitude of run strategies for running Monte Carlo simulations on multi-node computer systems, with a focus on increasing parallel efficiency while still being able to calculate real variance estimates. All numerical experiments were conducted on INL's Falcon computer cluster, where each compute node comprises of 36 CPU cores. The following bullets summarize the merits and drawbacks of each run strategy

- **CMFD source acceleration vs. Un-accelerated Monte Carlo**: For every run strategy proposed in this chapter, two variants were investigated - one that utilized CMFD source acceleration and one that used vanilla Monte Carlo without any source acceleration. For reactor simulations, the reduction in generations to stationarity far outweighs the computational overhead associated with running CMFD source acceleration, thus CMFD should be turned on in the inactive generations. In the active generations, however, there is no direct benefit in terms of improving the convergence rate and so CMFD should be turned off in the active generations

to minimize computational costs. These results for large-scale simulations are consistent with those described in Chapter 3.

- **Multiple trial ensemble averaging**: The naive approach to ensemble averaging runs multiple independent trials in parallel. This improves parallel efficiency as each ensemble of neutrons no longer has to wait for other ensembles of neutrons to finish running in order to be able to continue to subsequent generations. However, as was described in Section 4.1.2, without synchronization each independent trial diagnoses stationarity based on a single trial stopping criteria, which yields a less stringent stationarity criteria and a higher fission source error residual when starting the active generations. It should also be noted that in this chapter where multiple independent trials with CMFD were run, each ensemble had a generation size of 10 million, which was a sufficiently large enough generation size for CMFD fluxes to be computed without any stability issues. However, if the number of computational resources available were to increase while keeping the total number of particles per generation fixed, the number of particles per ensemble would decrease to a level where the CMFD solver would be unable to run due to highly biased CMFD tallied quantities. Herman describes this phenomenon when simulating the 3-D BEAVRS problem with 1 million particles per generation [26]. However, CMFD is susceptible to these solver failures regardless of problem type if an insufficient number of particles per generation are simulated to generate CMFD parameters, so the multiple trial CMFD strategy fails to produce solutions in the strong scaling limit.

- **Single trial simulations**: This chapter separated the effectiveness of a single trial simulation based on whether a large generation size or small generation size was used. When a large generation size is used, stationarity is diagnosed based on the aggregate source. However, parallel efficiency is degraded as all particles in the generation need to be transported before the Monte Carlo solver can proceed to its next generation. Moreover, the fact that a single trial is run means that accurate variance estimates cannot be computed. When a small generation size is used,

it was shown that the use of batch statistics to compute real standard deviation leads to higher variance levels and higher uncertainty levels in the variance when compared to an ensemble averaged CMFD approach that is run for an equivalent number of CPU-hours. Specifically for the 2-D BEAVRS case, it was also shown that an insufficient number of active generations was simulated in order to be able to compute batch statistics with completely uncorrelated batches. In this case, batch statistics were computed assuming correlated batches. In most cases, however, these correlation levels are a priori unknown and unless they are accounted for, reported variance levels can be significantly biased. It should also be noted here that as outlined by Romano et. al, as the number of processors used to simulate a single trial simulation increases while keeping generation size fixed, situations of processor *starvation* will occur, where a limited amount of work divided over a large number of processes leads to scenarios where each process no longer has enough work to run efficiently [76].

- **Bulk synchronous ensemble averaged CMFD**: This was the first and simpler of the two implementations of an ensemble averaged CMFD run strategy that makes use of a shared CMFD operator across ensembles of Monte Carlo that all have different starting pseudo-random seeds. It uses an aggregate trial stopping criteria and enables the computation of unbiased real variance estimates across multiple Monte Carlo ensembles. However, parallel efficiency is sacrificed due to the synchronous communication pattern between MC nodes and CMFD node during the CMFD feedback step.

- **Asynchronous ensemble averaged CMFD**: The second of two implementations for the ensemble averaged CMFD run strategy tries to combine the benefits of a bulk synchronous approach while trying to improve parallel efficiency. In the asynchronous case, each ensemble communicates only with the CMFD node through one-sided MPI RMA calls, and does not have to synchronize with other MC nodes. This run strategy was shown to reduce overall runtime to stationarity without leading to biased real variance estimates.

Thus, the asynchronous ensemble averaged CMFD combines the following advantages into a single run strategy

1. **A CMFD-based run strategy**: Asynchronous ensemble averaged CMFD uses CMFD as a basis for source acceleration, which results in reduced overall runtime and generations to stationarity.

2. **Utilizes an aggregate trial stopping criteria**: The CMFD node determines stationarity based on an aggregate trial stopping criteria so a sufficient number of generations to stationarity is prescribed to minimize variance levels in the aggregate fission source across all ensembles of neutrons in a generation.

3. **Better load balancing**: As shown through the performance profiling plots in Figures 4-11 and 4-13 and from the runtime results in Tables 4.6 and 4.7, an asynchronous ensemble averaged CMFD approach reduces communication overhead between generations of ensembles, which leads to lower overall runtime to stationarity and better load balancing on large-scale Monte Carlo simulations.

4. **Better resilience to node failures**: Through ensemble averaging, if any compute node were to go offline due to a node failure, results could continue to be collected across the remaining compute nodes still in operation. Therefore, ensemble averaging provides additional fault tolerance on large-scale compute jobs.

5. **Ability to calculate real variance levels**: A recurring theme throughout this chapter has been the idea that single trial simulations do not provide the ability to compute accurate variance levels without simulating a substantially large number of active generations. An asynchronous ensemble averaged CMFD solves this problem by running multiple ensembles of neutrons in parallel, and it has been shown that even with the use of a shared CMFD operator, these ensembles are statistically independent and can be used to calculate unbiased real variance levels in a simple manner.

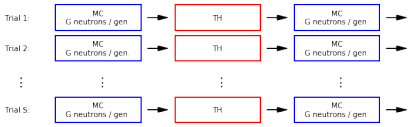
Additionally, the ensemble averaging method has applications to TH coupling and GPU computation, which are detailed below:

6. **Extension to TH Coupling**: As was alluded to in Chapter 3, CMFD can be seen as a surrogate for the non-linear coupling process, and once again ensemble averaging can be leveraged to improve Monte Carlo coupling capabilities with deterministic non-linear solvers. As described at the beginning of this chapter, running Monte Carlo with larger generations and fewer active generations should be the preferred method for TH-coupling, and such a schematic is depicted in Figure 4-18a. The issue with such an approach is that since only a single trial is being run, it is difficult to generate Monte Carlo variance estimates before and after the deterministic coupling step to get a sense of how much fluctuations tallied quantities possess. On the other hand, if completely independent trials of Monte Carlo are run with smaller generation sizes per trial - as illustrated in Figure 4-18b - then the Monte Carlo parameters passed to the TH solver are subject to larger noise levels since fewer neutron histories contribute to the calculation of these parameters. This may push the deterministic solver into an erroneous solution. Therefore, an ensemble averaging approach allows for variance levels to be calculated across multiple Monte Carlo trials while aggregating coupling parameters across all trials in order to reduce statistical fluctuations during the coupling step. The schematic for ensemble averaging in shown in Figure 4-18c.

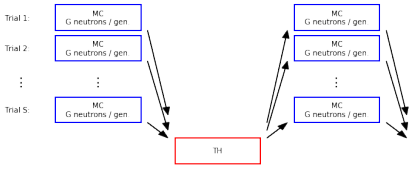
7. **Extension to GPU Computing**: As mentioned in Chapter 1, both exascale machines being currently built at ANL and ORNL are GPU-based machines, so the reactor physics community has been working towards porting existing Monte Carlo algorithms to a vectorized, event-based algorithm more suited for GPU computation. However, GPU's have limited availability of on-device memory, and this restricts the maximum generation sizes that are achievable per GPU [24]. For example, Hamilton et al. showed that on the NVIDIA Tesla K20X GPU that has 6 gigabytes of memory, only 3.2 million particles per generation could be run before the device ran out of memory when running a fresh fuel SMR problem. Similarly, on the NVIDIA Tesla V100, which has 16GB of on-GPU memory, the maximum generation size was 6.4 million on the same SMR problem. The computation of tallies

in the active generations would further increase the memory constraints on the problem and reduce the number of particles per generation that can be run on the GPU. ORNL's exascale supercomputer Frontier features 4 AMD GPU's and 1 AMD CPU per node, where each GPU will now have 32 gigabytes of memory, so memory will continue to be limited on these machines as well [10]. Given the smaller generation sizes that are attainable on each GPU, an ensemble averaging approach can be leveraged to run on multiple nodes of such supercomputers. Since CMFD requires less computational time to run than Monte Carlo, each CPU on a compute node can be designated as the CMFD server responsible for communicating CMFD parameters, while an ensemble of neutrons running the maximum possible number of particles per generation can be run with Monte Carlo on each GPU of the compute node. In this manner, the total target generation size is distributed among the available GPU's in a compute job, while CMFD source updates and FET stationarity diagnosis continue to occur on the CPU.

(a) MC-TH coupling with single MC trial and large generation size



(b) MC-TH coupling with completely independent MC trials



(c) MC-TH coupling with multiple independent MC trials and a shared TH operator

Figure 4-18: Schematics for running MC-TH coupling

Highlights

- The asynchronous ensemble averaged CMFD approach is a CMFD-based run strategy that utilizes an aggregate trial stopping criteria and is shown to reduce communication overhead and increase parallel efficiency for Monte Carlo on large-scale computing systems. The use of ensemble averaging also allows for the calculation of unbiased real variance estimates.
- When compared to single trial simulations with small generation sizes and larger number of active generations with a fixed runtime, an ensemble averaged CMFD approach is able to achieve lower variance levels with lower uncertainty in the variance as well.
- CMFD is a surrogate for the non-linear coupling process, and ensemble averaging can be leveraged with other types of deterministic non-linear solvers to generate Monte Carlo parameters with smaller statistical noise levels while still providing a sense of Monte Carlo uncertainty levels before and after the deterministic coupling step.
- Ensemble averaging can be extended to the GPU-based architectures of next-generation exascale machines by distributing target generation sizes over the available GPU's in a compute job while centralizing all CMFD calculations and FET diagnosis on the CPU, thus overcoming memory limitations faced by GPU's.

Chapter 5

Comparison of Simulation Strategies on the 3-D BEAVRS Benchmark

This chapter combines the findings from this thesis so far and applies them to the final test case - the 3-D BEAVRS benchmark. This benchmark represents the full three-dimensional PWR core geometry with all axial heterogeneity at HZP conditions [64]. The FET stationarity diagnostic introduced in Chapter 2 will be applied to this test problem to determine an adequate number of generations to stationarity in the inactive generations. CMFD will be used to speed up fission source stationarity, and the CMFD parameters for this problem will be based on recommendations from Chapter 3. Finally, a comparison of the run strategies proposed in Chapter 4 will be applied to the 3-D BEAVRS case using similar-sized computational resources.

When using Monte Carlo methods, the 3-D BEAVRS benchmark represents a similar computational complexity as the 2-D BEAVRS benchmark problem in terms of per-generation runtime. For the 3-D case, however, more fission source generations are required in the inactive generations to attain stationarity [26]. This difference in generations in stationarity is largely due to the fact that a uniform starting source closely represents the true radial solution, while in the axial dimension, a larger number of generations is required to converge a uniform starting source to the true cosine-shaped solution. In the active generations, the 2-D and 3-D BEAVRS problems have similar ACC's, so convergence rates remain on par between these two variants. As such, most

Monte Carlo analysis for PWR reactor simulations conclude by analyzing the 2-D BEAVRS problem in detail with a recommendation to increase computational resources in order to simulate the 3-D problem in full. In order to bridge this gap between the 2-D and 3-D models using Monte Carlo, Herman successfully applies CMFD source acceleration to the full 3-D BEAVRS model in order to gauge the efficacy of CMFD in the inactive generations [26], while Jo et al. expands on this work by applying the unconditionally stable p-CMFD method to the 3-D BEAVRS problem in order to shed light on how p-CMFD feedback affects real variance estimates and inter-generational correlation effects [79].

This chapter expands on work by Herman and Jo et al. by investigating how ensemble averaging run strategies fare on the 3-D BEAVRS problem, with a focus on overall runtime and real variance levels associated with each run strategy. While previous studies focus on the symmetric 3-D BEAVRS core, which is easier to converge in the inactive generations, flux measurements from actual operating data indicate a substantial radial tilt in the axially integrated fission source despite an eighth-core symmetric geometry [2, 80]. This tilt is hypothesized to be caused by varying inter-assembly gaps from the core loading process [2]. In order to truly capture the impact of a non-symmetric core on the overall Monte Carlo time to solution, this work will alter the 3-D BEAVRS model by artificially inducing a radial tilt in the core to match the tilt observed from BEAVRS operational data. This is achieved by modifying the water density in each assembly. Modifying the core geometry in this way adds a level of computational complexity to the Monte Carlo process that is indicative of modeling realistic full-core problems that are not always guaranteed to be symmetric in nature.

This chapter is therefore laid out as follows: The exact problem parameters and geometry specifications of the benchmark problem will be discussed in Section 5.1, with a description of how the radial tilt is induced in Section 5.1.1. The details of stationarity diagnosis will be included in Section 5.2. Finally, all performance profiling and runtime analysis for the 3-D BEAVRS problem are covered in Section 5.3.

Table 5.1: CMFD parameters for the 3-D BEAVRS problem for test cases in Chapter 5

Parameter	Value for 3-D BEAVRS Problem
CMFD tally estimator type	Analog
Number of energy groups	2
Effective downscatter	True
Acceleration map	Used
Reference diffusion parameters	Used
CMFD tally begin	1
CMFD solver begin	1
Weight clipping	0.2
Windowing scheme	Expanding window
Maximum window size	32
CMFD mesh size	Assembly mesh radial, ∼20cm mesh axial
Prolongation source shape	Linear source axially, fixed source radially

5.1 Problem Parameters

Table 5.1 summarizes the CMFD parameters for this chapter. At the CMFD prolongation step, a linear source prolongation described in Section 3.2.3.5 and Appendix C is used in the axial direction, while a fixed source prolongation is used in the radial direction. It was also found that the CMFD penalty factor - defined in Section 3.3.1 - for the 3-D BEAVRS problem is now 32, so a maximum window size of 32 is used in all CMFD-accelerated problems to ensure that instantaneous fission source variance levels match those of the un-accelerated case. The acceleration map used for the 3-D BEAVRS problem is similar to the 2-D case radially, mapping directly to an assembly-sized mesh with an albedo boundary condition used to define partial currents at the core-reflector interfaces. Axially, the acceleration map uses the same discretization as the CMFD mesh with an additional axial layer above the top of the active fuel and below the bottom of the active fuel to define partial currents.

The CMFD mesh is defined in a similar manner as the quarter-core CMFD mesh defined by Harper in his dissertation [39]. where a rectilinear mesh is imposed on the 3-D BEAVRS active fuel region to account for the non-uniform spacing of spacer grids

in the axial direction. In the radial direction, an assembly sized mesh is used. In the axial direction, mesh boundaries are places at the bottom and top of the grid spacers in order to encompass each grid spacer region exactly. Between the grid spacer mesh cells, each segment is filled with the smallest number of mesh cells that yields a spacing of less than 20cm.

While this mesh overlay represents a coarsely defined CMFD mesh, the added dimensionality of the problem results in a larger number of total mesh cells to solve for in the CMFD solver, with a total of 193 radial mesh locations $\times$ 22 axial mesh locations $\times$ 2 energy groups $= 8,492$ total mesh cells in the problem. Thus, OpenMP multi-threading as outlined in Section 3.2.3.3 is now used with 18 threads to reduce computational overhead in the CMFD solver. A finer CMFD mesh is not considered for this chapter since it was found that refining the mesh size added a significant performance penalty without increasing the quality of the solution. For reference, when the CMFD mesh is defined as a quarter assembly mesh radially and a 10cm mesh axially, the overall runtime per generation increases by nearly 30% when a generation size of 10 million is used. Moreover, the use of a finer mesh CMFD solution does not lead to lower variance levels in the inactive or active generations.

5.1.1 Inducing a Radial Tilt in the 3-D BEAVRS Benchmark

As described in previous uncertainty quantification work for the BEAVRS benchmark model, the BEAVRS core possesses a radial tilt in operating data despite an eighth-core symmetric core loading pattern. This tilt is largest at HZP and dissipates with burnup and power ramp-up [2, 80]. The exact source of this tilt is unknown, but it is hypothesized that uneven water gaps during core loading causes this irregularity [2]. This radial tilt represents an un-modeled phenomenon that cannot be replicated by simulation tools, which produce symmetric reaction rate maps for symmetric core loading patterns. Figure 5-1 plots the radial distribution of axially integrated fission rates at HZP as computed from measured in-core detector signals for the BEAVRS benchmark, along with the fractional difference with simulated data when modeling the BEAVRS core with the

CASMO/SIMULATE code package.

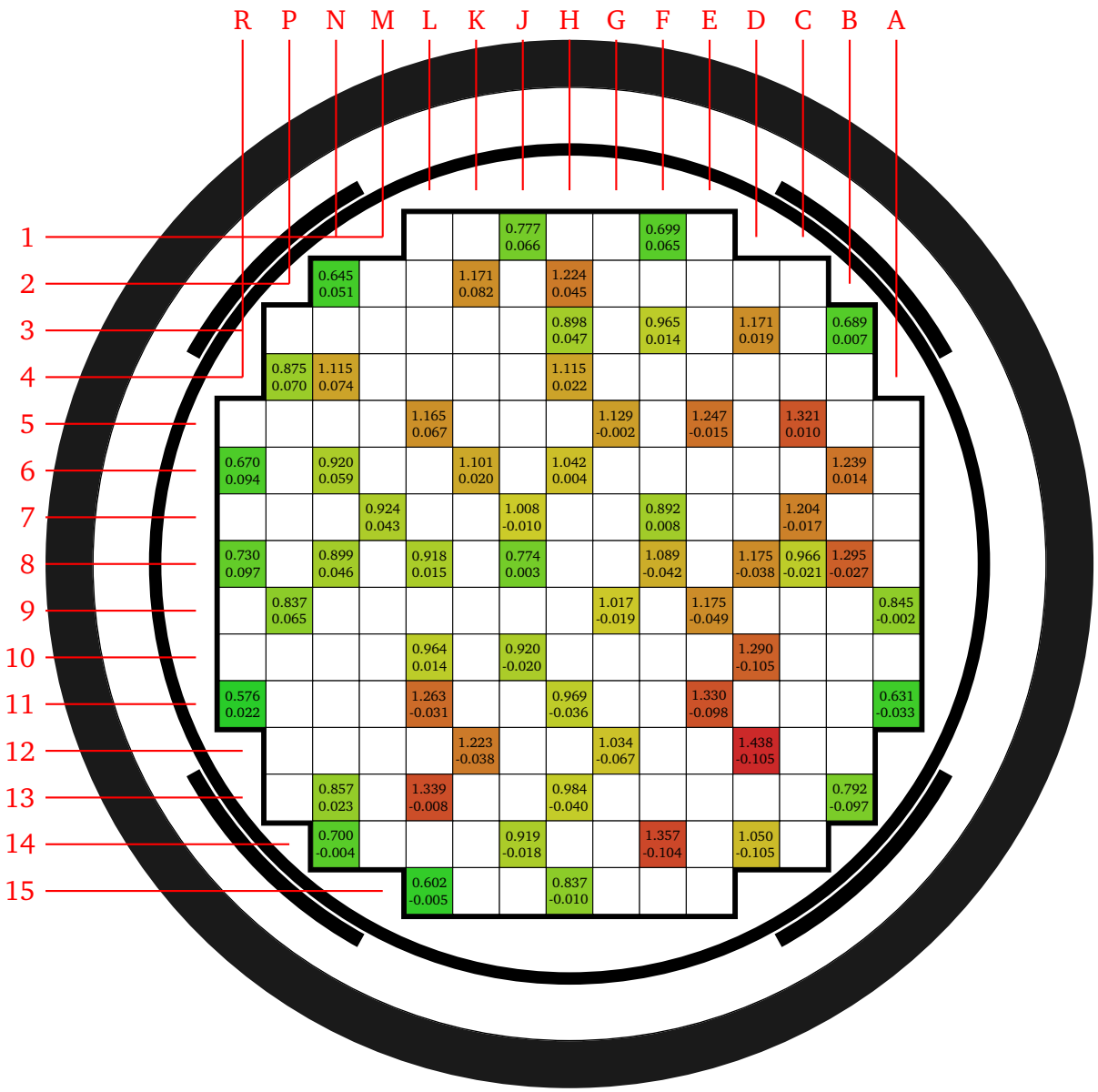


Figure 5-1: Radial assembly measurements of fission rates calculated at HZP for BEAVRS Cycle 1 operating data. The top number is the axially integrated normalized assembly signal while the bottom number is the fractional difference with data simulated from the CASMO/SIMULATE code [2].

In order to mimic a similar tilt with Monte Carlo simulation tools, the input files to the 3-D model are altered in OpenMC so as to artificially induce a radial tilt in the fission source distribution. This is done by adjusting the density of each water region in each assembly. Figure 5-2 plots the value for water density for each assembly, where 0.741 g/cc is the base water density level in the core without any modification, and

each crescent-shaped layer in the northwest and southeast directions observes a change in density of ±0.004 g/cc. This is done as a simple way to mimic variations in inter-assembly spacing. This change in density was tuned in order to produce a roughly ∼10% radial tilt in the axially-integrated fission source distribution when compared to the symmetric 3-D BEAVRS model, on par with the level of tilt observed in Figure 5-1. Figure 5-3 plots the relative difference in the axially-integrated stationary fission source distribution between the symmetric and titled 3-D BEAVRS cases. In this manner, a radially tilted distribution is more difficult to converge as it activates the odd modes of the radial distribution, while a symmetric core takes fewer generations to stationarity since it activates fewer modes in the radial solution. In most cases, tilts in the reactor core may not be this large, but minor asymmetries can still exist.

The induced tilt in the 3-D BEAVRS problem therefore represents the added computational demands required of realistic reactor-type problems, where more generations are needed in the inactive regime to resolve the asymmetries in the stationary fission source. As such, this 3-D problem combines the difficulties of the 1-D and 2-D problems presented in this thesis. Specifically, the simulation of an asymmetric fission source distribution from a symmetric starting source distribution for the 3-D case is similar to the 1-D case where an asymmetric starting source is used to simulate a symmetric stationary distribution. As will be explored in Section 5.2, this adds computational difficulty in the form of added generations to stationarity. Similar to the 2-D problem, the heterogeneity in the 3-D core geometry and materials also results in computational overhead to the per-generation runtime. It should be noted that the radially tilted 3-D BEAVRS test problem does not contain significant axial asymmetries. However, such asymmetries in the form of control rod insertions, for example, would necessitate an even larger number of generations to stationarity.

A radial tilt is induced to the 3-D BEAVRS test case in order to create an asymmetric stationary fission source distribution, which is on par with BEAVRS operating data. This is done in OpenMC by modifying the water density of each water region in the core.

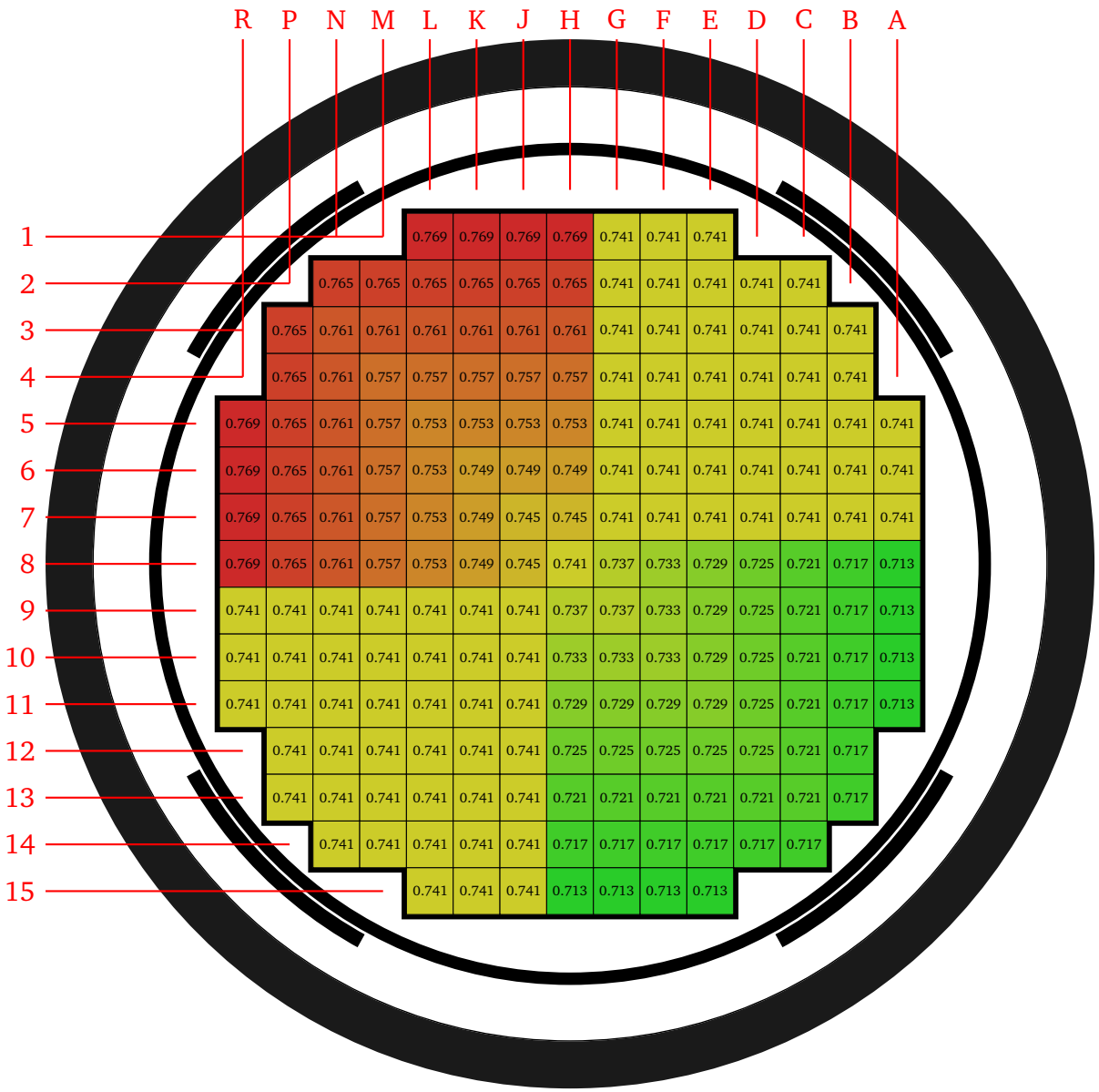


Figure 5-2: Modified water density values for each assembly location

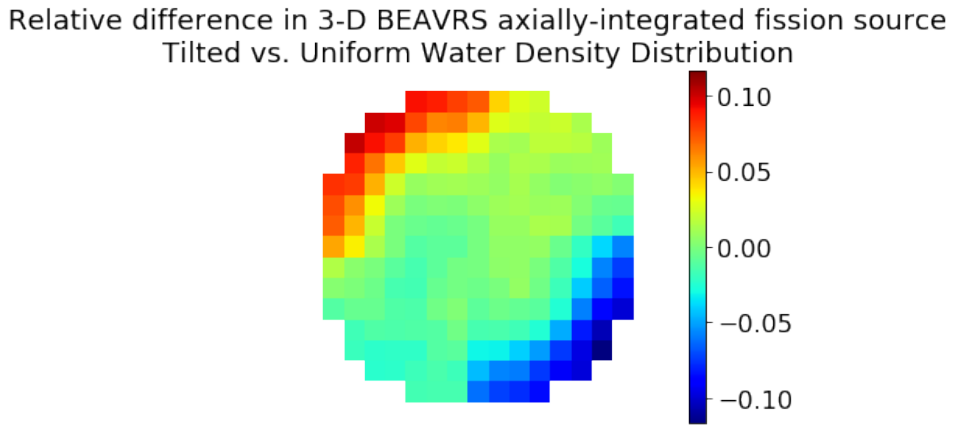


Figure 5-3: Relative difference in axially integrated stationary fission source distribution between the symmetric 3-D BEAVRS and the tilted 3-D BEAVRS case. The tilted case uses the water density distribution shown in Figure 5-2.

5.2 Stationarity Diagnosis

This chapter will once again use an FET-based diagnostic to assess fission source stationarity. In line with the findings from Chapter 2, Shannon entropy does not properly diagnose stationarity due to the asymmetry of the 3-D problem. Figure 5-4 plots the entropy as a function of inactive generations and it can be seen that the entropy value flattens out relatively quickly for both the un-accelerated and CMFD-accelerated cases. Here, A generation size of 320 million is used to match the total generation size of the run strategies that will be considered in Section 5.3, and Shannon entropy is computed over a quarter assembly radial and roughly 10cm axial mesh. According to the entropy-based diagnostic presented in Section 2.3.2, stationarity occurs at generation 236 when CMFD is used and at generation 392 when CMFD is not used, as represented by the vertical lines in Figure 5-4.

When an FET-based diagnostic is used instead, however, a more conservative value for generations to stationarity is recommended. Here, a generation size of 320 million is used again, and the neutron fission rate is represented over the 3-D spatial domain as a separable Zernike expansion in the radial and azimuthal directions, and Legendre expansion in the axial direction:

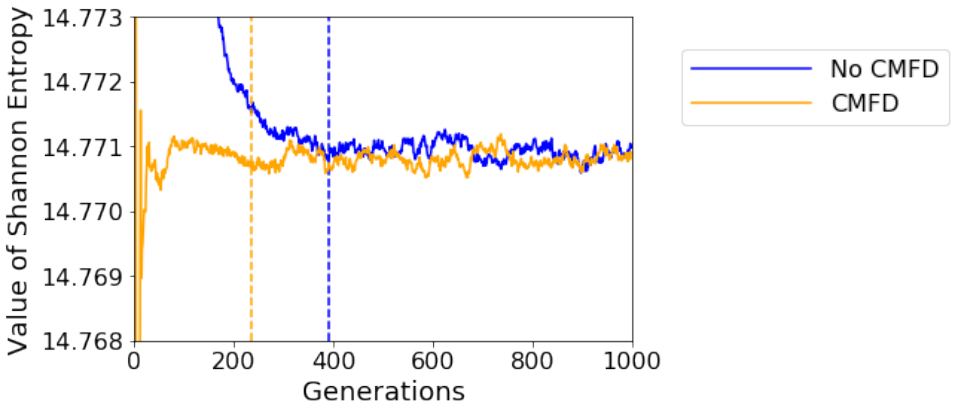


Figure 5-4: Value of Shannon entropy as a function of generation number for the tilted 3-D BEAVRS problem with CMFD acceleration (orange) and without CMFD acceleration (blue). The dashed vertical lines represent the stopping criteria according to Shannon entropy for each case.

$$\nu\Sigma_f(r,\theta,z)\psi(r,\theta,z) \approx \sum_{l}^{L}\sum_{n}^{N}\sum_{m=-n}^{n} a_{n,l}^{m} P_l\left(2.0\frac{z-z_0}{z_1-z_0}-1.0\right)Z_n^m\left(\frac{r}{R},\theta\right) \tag{5.1}$$

where L is the degree of the Legendre basis expansion, N is the degree of the radial order expansion of the Zernike basis set, P_l is the Legendre polynomial of order l, and Z_n^m is the Zernike polynomial of radial order n and azimuthal order n. $a_{n,l}^m$ is the expansion coefficient corresponding to Legendre order l, Zernike radial order n, and azimuthal order m. The axial domain is normalized between -1 and 1 and the Zernike radial domain is scaled to be defined over a unit disk. Effectively, the spatial domain encompasses the smallest cylinder that circumscribes the entire core. The expansion coefficient $a_{n,l}^m$ is tallied as part of the Monte Carlo algorithm as:

$$a_{n,l}^m = \begin{cases} \frac{(2l+1)(2n+2)}{(z_1-z_0)\pi R^2}\int_{z_0}^{z_1}\int_0^R\int_0^{2\pi} \nu\Sigma_f(r,\theta,z)\psi(r,\theta,z)P_l\left(2.0\frac{z-z_0}{z_1-z_0}-1.0\right)Z_n^m\left(\frac{r}{R},\theta\right)rdrd\theta dz & \text{if } m\neq 0 \\ \frac{(2l+1)(n+1)}{(z_1-z_0)\pi R^2}\int_{z_0}^{z_1}\int_0^R\int_0^{2\pi} \nu\Sigma_f(r,\theta,z)\psi(r,\theta,z)P_l\left(2.0\frac{z-z_0}{z_1-z_0}-1.0\right)Z_n^m\left(\frac{r}{R},\theta\right)rdrd\theta dz & \text{if } m=0 \end{cases} \tag{5.2}$$

All Zernike-Legendre FET pairs up to Legendre order 10 and Zernike radial order 10 are considered “low-order" and analyzed to assess stationarity. Figure 5-5 plots the value of representative FET coefficients for the tilted 3-D BEAVRS problem with and

without CMFD, and the vertical lines represent the point when each of these modes has reach stationarity based on the FET-based stopping criteria. Here, it can be seen that a significantly larger number of generations to stationarity are required when using an FET-based diagnostic, where 830 inactive generations are recommended without CMFD and 344 generations are recommended with CMFD. Like the 1-D and 2-D cases, the entropy-based stationarity diagnostic underpredicts generations to stationarity due to the slow migration of source particles through the core. On the other hand, the FET-based approach more suitably diagnoses stationarity as each source particle contributes to the evaluation of each spatial moment of the fission source. For reference, Figure 5-6 plots the relative error between the axially integrated fission source as predicted by an FET-based and entropy-based stationarity diagnostic, and shows a noticeable error that results from using an entropy-based stopping criteria that underpredicts the number of inactive generations to stationarity. The mesh-averaged relative error between the axially integrated fission source on a quarter assembly mesh between these two stationarity diagnostics equates to ∼0.4%.

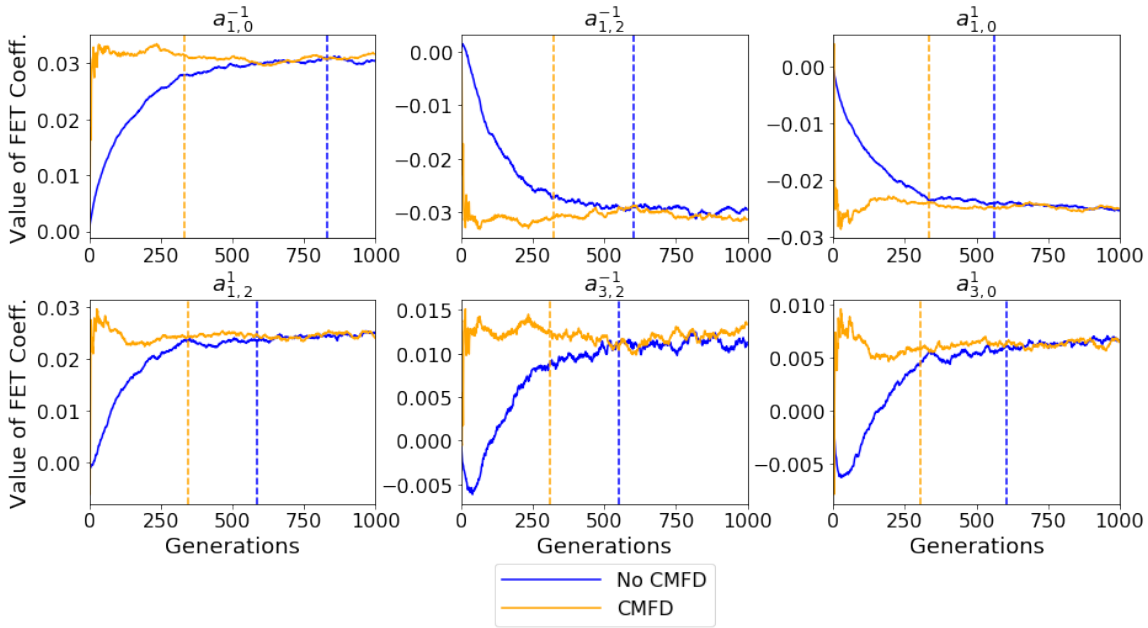


Figure 5-5: Value of representative FET coefficients as a function of generation number for the tilted 3-D BEAVRS problem with CMFD acceleration (orange) and without CMFD acceleration (blue). The dashed vertical lines represent the stopping criteria for each particular mode.

An FET-based diagnostic that utilizes separable radial Zernike and axial Legendre basis functions is used to diagnose fission source stationarity for the 3-D BEAVRS test case, and recommends a larger number of generations to stationarity when compared to an entropy-based diagnostic.

5.3 Performance Profiling and Runtime Analysis

Table 5.2: Compute resource parameters for run strategies 1 to 6 applied to 3-D BEAVRS problem

Run Strategy	Total Compute Nodes	Total Independent Trials	Generation Size per Trial	MPI Processes per Trial	OpenMP Threads per MPI Process
1 - Multiple Trial Monte Carlo	32	32	10 Million	2	18
2 - Multiple Trial CMFD	32	32	10 Million	2	18
3 - Single Trial Monte Carlo	32	1	320 Million	64	18
4 - Single Trial CMFD	32	1	320 Million	64	18
5 - Bulk Synchronous Ensemble Averaged CMFD	33	32	10 Million	2	18
6 - Asynchronous Ensemble Averaged CMFD	33	32	10 Million	2	18

This section demonstrates the performance profiling studies and runtime analysis for the tilted 3-D BEAVRS problem. The run strategies pursued are summarized in Table 5.2 and mirror the same computational parameters as run strategies 1 to 6 explained in Chapter 4. Due to the computational constraints of producing a reference solution for this problem, RMS errors are not calculated for these run strategies, and instead real standard deviation levels calculated across multiple independent Monte Carlo trials will

be used to determine any potential variance biases among the run strategies. Moreover, the single trial run strategies with small generation sizes are not considered either (run strategies 7 and 8 in Chapter 4), as it was shown in Section 4.3.3 that such run strategies possess higher variance levels and higher uncertainties in the variance.

Performance plots similar to the ones in Sections 4.1.1 and 4.2.4 are shown for the 3-D case for run strategies 1 to 6 in Figures 5-7 to 5-12. Here, it can be seen that the CMFD solver runs more slowly than the 1-D and 2-D cases due to the added number of CMFD mesh cells in the axial dimension. However, the use of multithreading on 18 threads is able to effectively reduce this runtime, and the cost of running the solver is still significantly lower than that of running a single Monte Carlo power iteration.

Table 5.3: 3-D runtime results for run strategies that utilize a single trial stopping criteria

Run Strategy	Trial-Averaged Generations to Stationarity	Total Computational Time (CPU-hrs)
1 - Multiple Trial Monte Carlo	612.7	15,217
2 - Multiple Trial CMFD	269.5	10,778
5 - Bulk Synchronous Ensemble Averaged CMFD	47	1,401

Table 5.3 summarizes the total runtime associated with run strategies that utilize a single trial stopping criteria. Similar to the methodology in Section 4.3.1, the multiple trial test cases are compared to the bulk synchronous ensemble averaged CMFD case, and it is found that only 47 generations to stationarity are required when imposing a single trial stopping criteria onto the aggregate trial source computed by the bulk synchronous approach. Figure 5-13 plots the standard deviation in the instantaneous $a^1_{1,2}$ FET coefficient for a single ensemble of neutrons for the multiple trial Monte Carlo case (run strategy 1) along with the standard deviation in the average $a^1_{1,2}$ coefficient across all Monte Carlo trials in the bulk-synchronous case (run strategy 5). Here, the $a^1_{1,2}$ FET coefficient represents the slowest FET coefficient to reach the same terminal

variance levels as a single ensemble of neutrons for the 3-D BEAVRS problem.

Once again, the results from Table 5.3 signify that the use of an ensemble averaged CMFD approach results in a significantly lower number of generations to stationarity when compared to the multiple trial run strategies. As explained in Chapter 4, this is due to the fact that the presence of a larger number of ensembles towards the calculation of FET coefficient standard deviation yields a much faster convergence to the single trial stopping criteria when compared to the case where only a single ensemble of data is available.

Table 5.4: 3-D runtime and real standard deviation results for run strategies that utilize an aggregate trial stopping criteria.
*Calculated over a 2-D axially integrated assembly mesh.
**Calculated over a 3-D quarter assembly radial, 10cm axial mesh.
***32 Multiple independent trials were conducted to generate statistics on single trial simulations

Run Strategy	Trial-Avg. Gens. to Stationarity	Total Comp. Time (CPU-hrs)	2-D Mesh Rel. Uncertainty* (%)	3-D Mesh Rel Uncertainty** (%)
3 - Single Trial MC	830	19,334	0.953***	9.59***
4 - Single Trial CMFD	344	9,916	0.955***	9.60***
5 - Bulk Synch. Ens. Avg. CMFD	345	10,259	0.953	9.59
6 - Asynch. Ens. Avg. CMFD	341.3	9,083	0.954	9.59

Table 5.4 summarizes the runtimes and standard deviation levels associated with run strategies that utilize an aggregate trial stopping criteria. Here, a few things should be noted. Firstly, it can be seen that the asynchronous ensemble averaged CMFD case has the lowest overall computational time to stationarity, leading to a nearly $\sim$10% decrease compared to the single trial CMFD case. This is primarily due to the fact that the CMFD solver is relatively more expensive now compared to the 2-D BEAVRS case, so in a single trial CMFD update step, each process has to wait for a comparatively higher time in

order to receive CMFD source data from the master process. On the other hand, the asynchronous approach continues to have low levels of latency during the CMFD update step since each OpenMC node only has to probe the CMFD node for the latest available source data in an asynchronous manner without having to wait for the CMFD node to recalculate any source values.

Next, the results also show a consistent level of relative uncertainty across the different run strategies presented. Relative uncertainty is computed using Equation 4.6 with two different types of meshes. The fourth column of Table 5.4 uses a 2-D mesh, where all fission rates are axially-integrated over an assembly-sized mesh. The fifth column, on the other hand, uses a finer 3-D mesh, where each mesh cell is quarter assembly sized radially and roughly 10cm in size axially. These results show that axially integrated quantities possess a roughly $\sim$1% uncertainty on the assembly mesh, while a high level of uncertainty on the order of $\sim$10% still remains when a 3-D mesh is used.

Extrapolating these values to target uncertainties on the 3-D mesh to $\sim$1% levels would require on the order of 800,000 - 900,000 CPU-hours. This is a significant computational burden that stems largely from the fact that a uniform starting source distribution is a poor starting point for each ensemble to begin the inactive generations. As a result, the total runtime is compounded from the fact that each additional ensemble spends an extremely large percentage of runtime in the inactive generations trying to converge an incorrect starting source distribution to a very strict aggregate trial stopping criteria. By comparison, the comparable runtimes for Table 5.4 for the symmetric 3-D BEAVRS problem with an asynchronous ensemble averaging run strategy requires 94.8 generations to stationarity and a total runtime of 2,481 CPU-hours, indicating that the total runtime is directly proportional to the total number of inactive generations to stationarity per ensemble. Thus, reducing the total number of inactive generations simulated by each ensemble is imperative in order to alleviate the amount of work required in the inactive generations. This could be done by relaxing the convergence criteria by using a single trial stopping criteria. Alternatively, another way to achieve this is to pass a loosely converged probability distribution function of a single ensemble run in the inactive generations as the starting guess for all ensuing ensembles, where a

better starting source guess translates to fewer work required to reach stationarity in the inactive generations. The concluding chapter proposes such a methodology as future work, in Section 6.3.1.

Finally, it should be noted that this work illustrates the need for accurate variance estimates, which are provided by ensemble averaging techniques. Without an proper treatment of true variance estimates, results from Monte Carlo methods can be incorrectly reported and made to look a lot more accurate than they truly are. As shown in Section 4.3.3, even if traditional batching methods are used to generate real variance estimates for single trial simulations, a prohibitively large number of generations per batch are required before batches can be assumed independent of each other, and the uncertainty in the reported variance estimates will also be larger. Therefore, ensemble averaging techniques provide a simple and reliable approach to measuring true variance levels in Monte Carlo simulations.

In summary, this chapter focused on the simulation of the 3-D BEAVRS problem, and an artificial radial tilt was induced to the fission source distribution by modifying the density of each water region in the core. This was done so as to mimic the actual tilt observed in operating data for the BEAVRS core and make the problem more difficult to converge. Thus, this problem combined the difficulties of the 1-D and 2-D test cases presented in this thesis, where the asymmetry of the core fission source distribution requires more generations to stationarity, while the heterogeneity in the core geometry and materials results in a larger time to simulate each additional generation. An FET-based stationarity diagnostic was shown to determine stationarity more accurately when compared to an entropy-based approach. Finally, runtime analysis shows a nearly 10% reduction in computational time to stationarity for the asynchronous ensemble averaged CMFD case when compared to a single trial CMFD approach. Thus, the asynchronous ensemble averaged CMFD approach is deemed as the optimal run strategy for full-core 3-D Monte Carlo simulations on the grounds of better load balancing and ability to calculate simple and accurate variance levels.

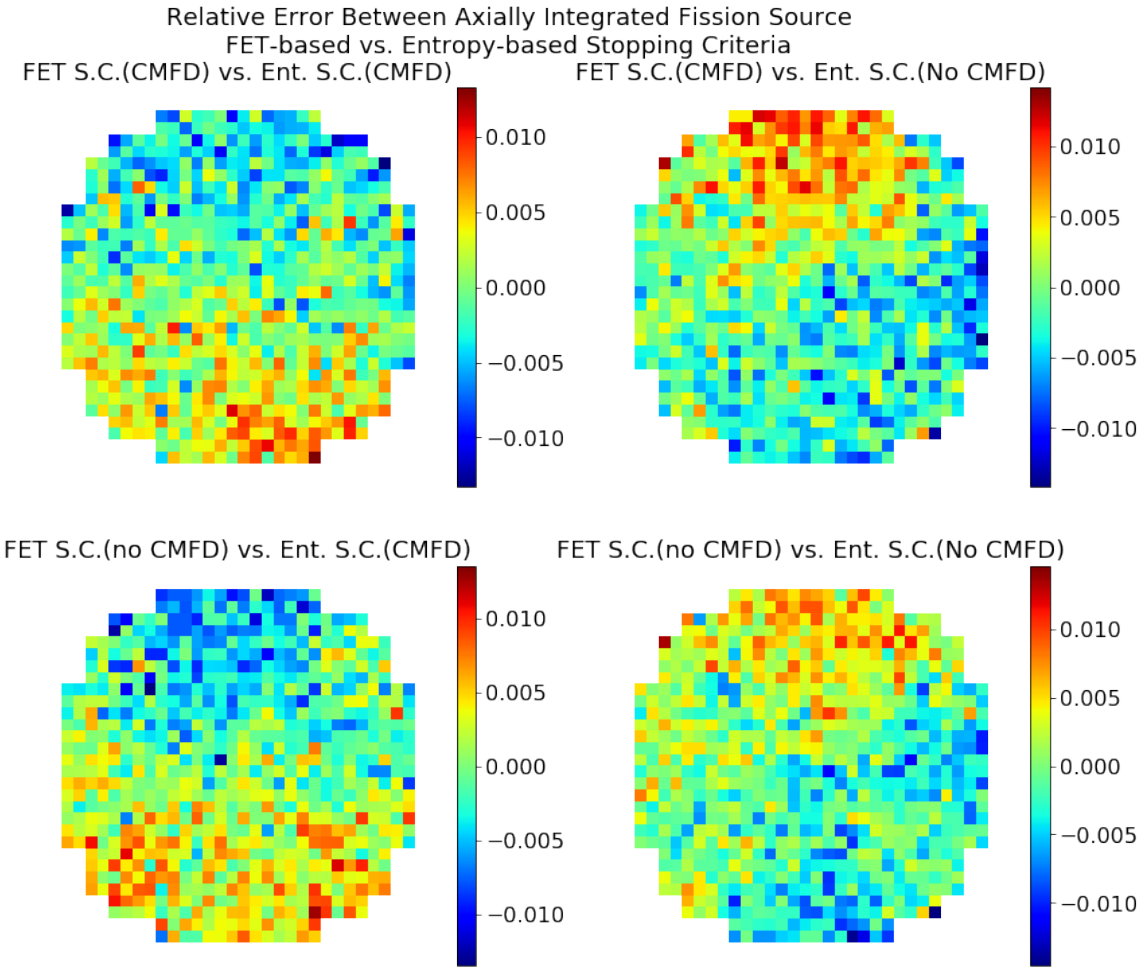


Figure 5-6: Relative error in axially integrated stationary fission source distribution as predicted by an FET-based and entropy-based stopping criteria for the tilted 3-D BEAVRS problem. The first row corresponds to the fission source as predicted by an FET-based stopping criteria when CMFD is used (344 generations to stationarity), while the second row corresponds to the fission source as predicted by an FET-based stopping criteria when CMFD is not used (830 generations to stationarity). Likewise, the first column uses an entropy-based stopping criteria with CMFD (236 generations to stationarity), while the second column uses an entropy-based stopping criteria with no CMFD feedback (392 generations to stationarity).

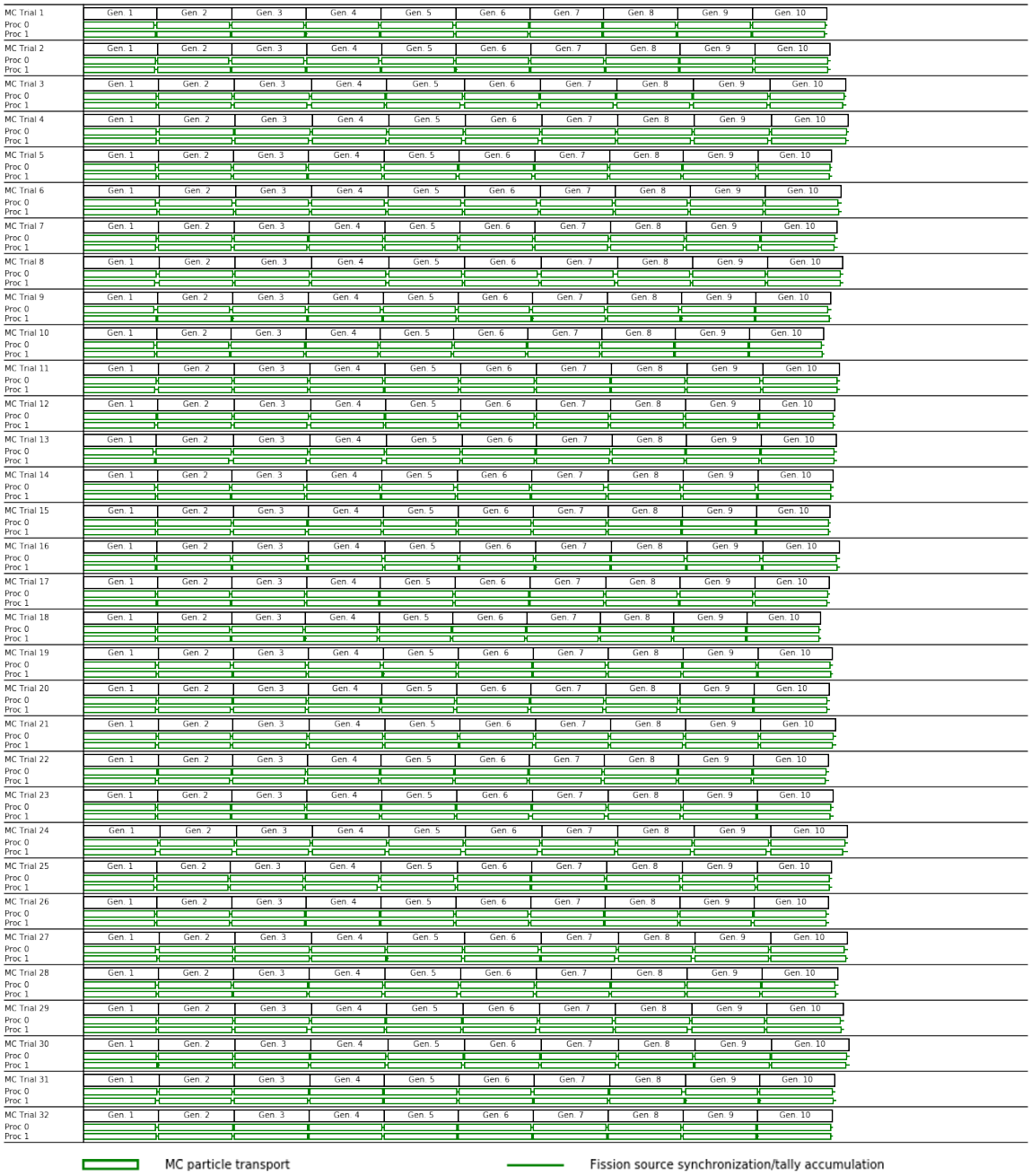


Figure 5-7: Performance profiling plot for run strategy 1 on the 3-D BEAVRS problem.

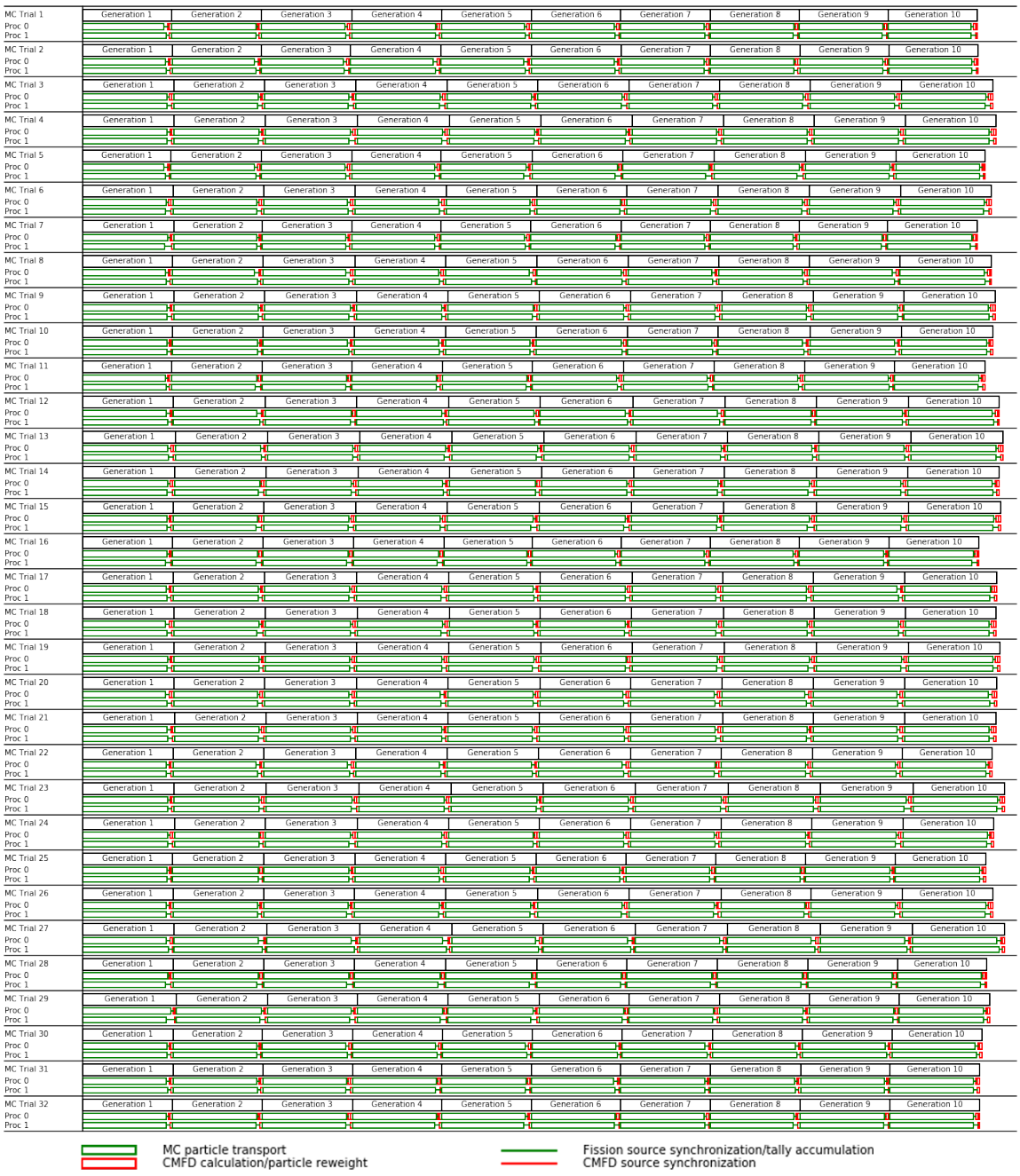


Figure 5-8: Performance profiling plot for run strategy 2 on the 3-D BEAVRS problem.

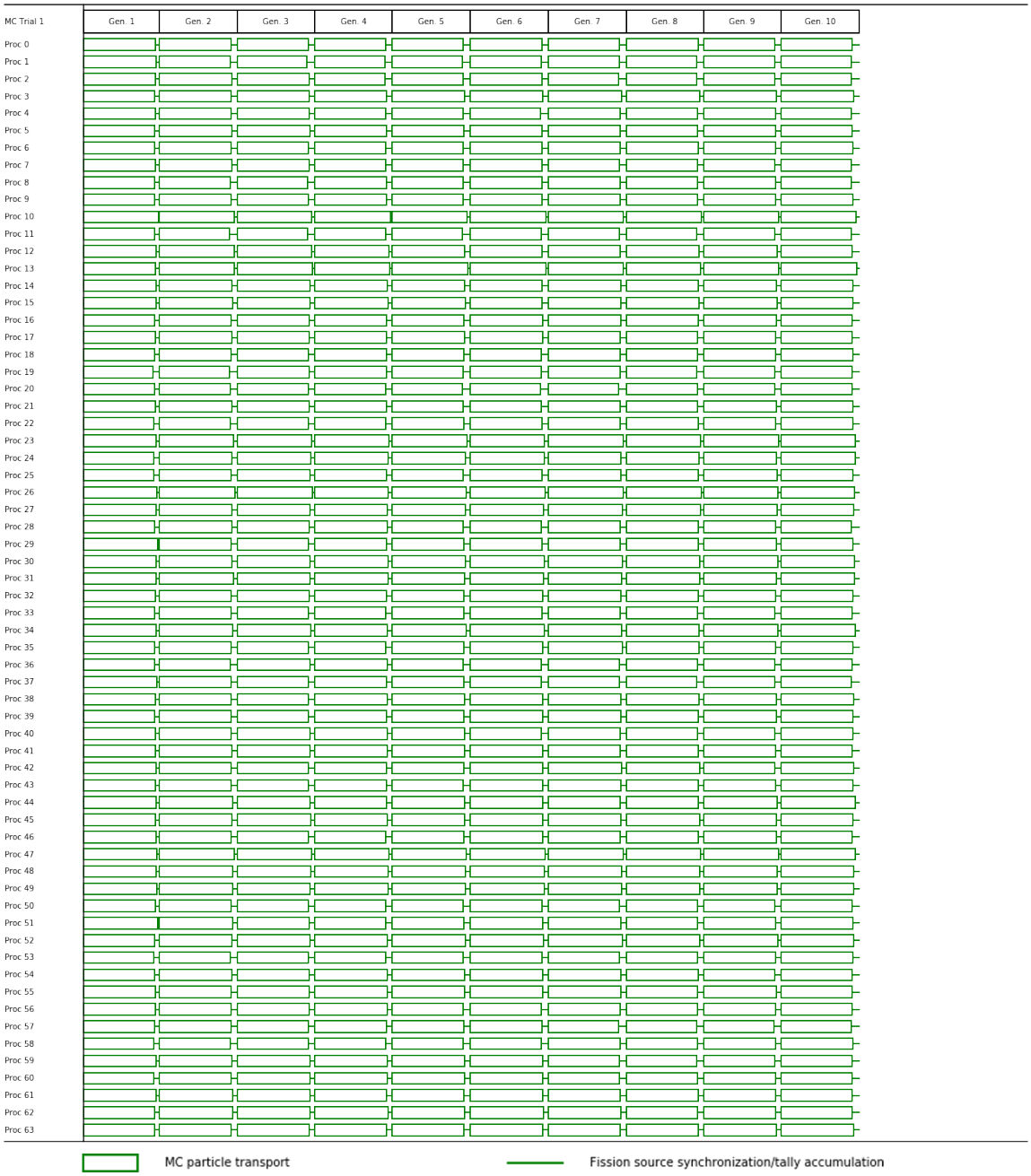


Figure 5-9: Performance profiling plot for run strategy 3 on the 3-D BEAVRS problem.

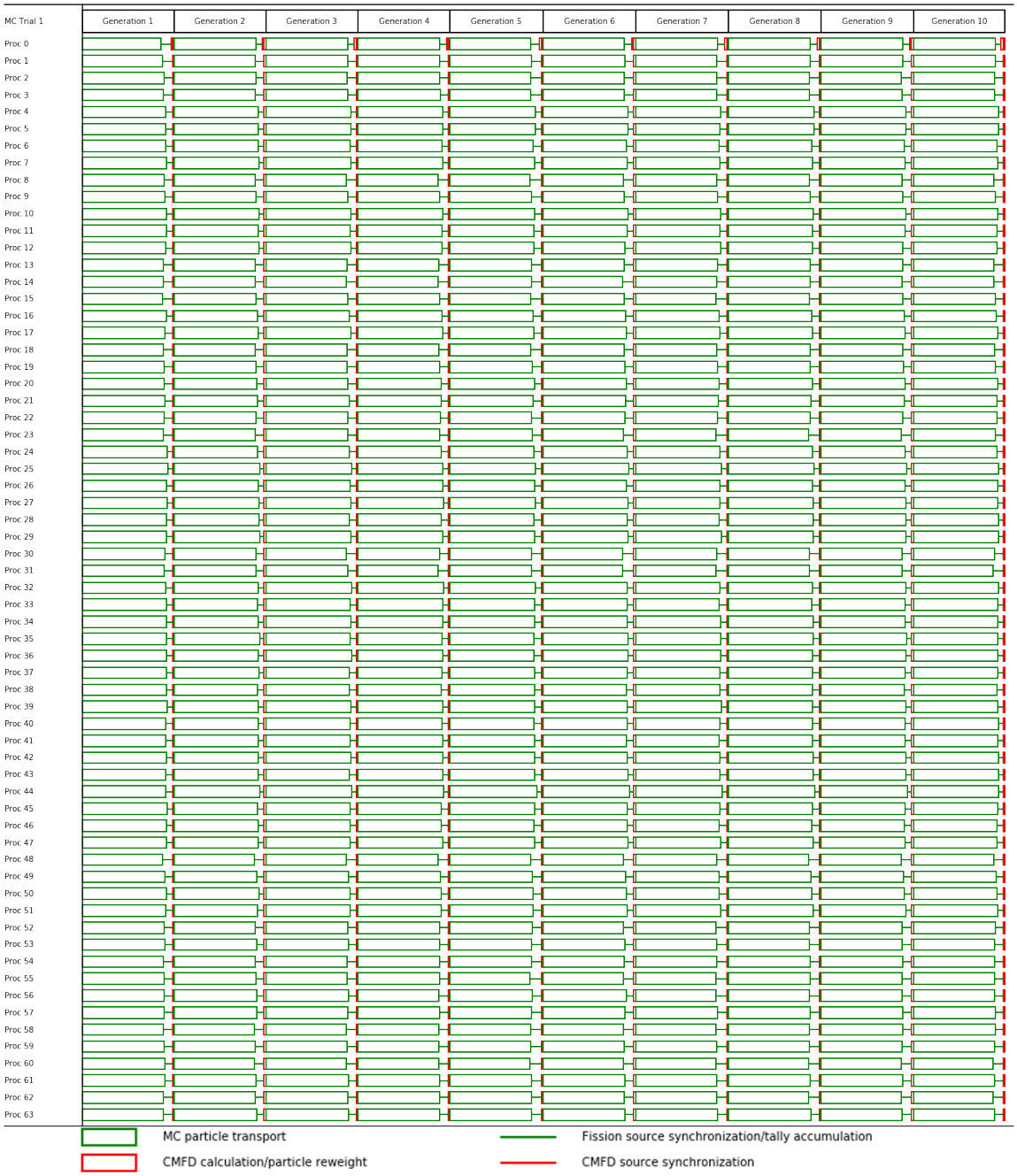


Figure 5-10: Performance profiling plot for run strategy 4 on the 3-D BEAVRS problem.

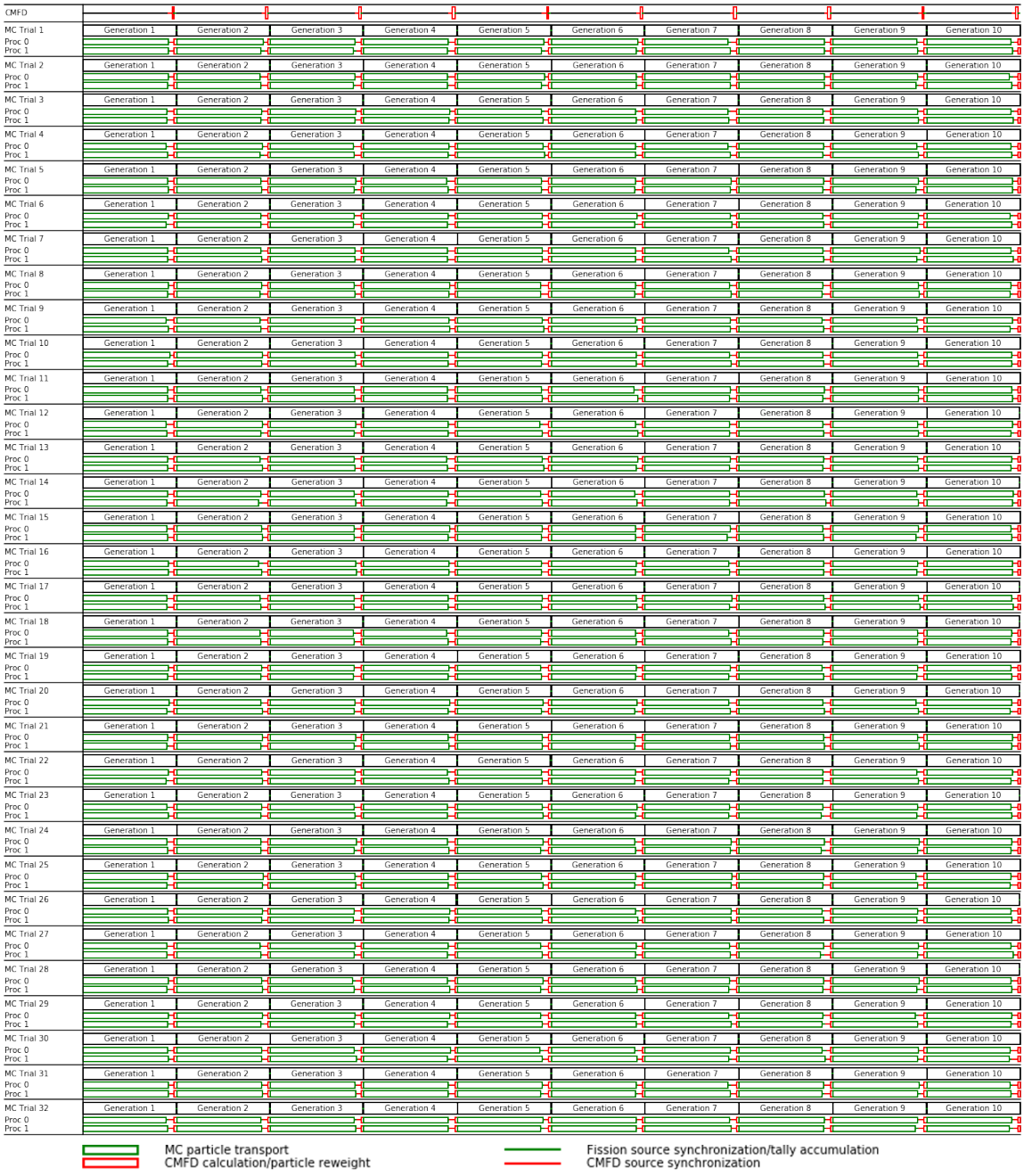


Figure 5-11: Performance profiling plot for run strategy 5 on the 3-D BEAVRS problem.

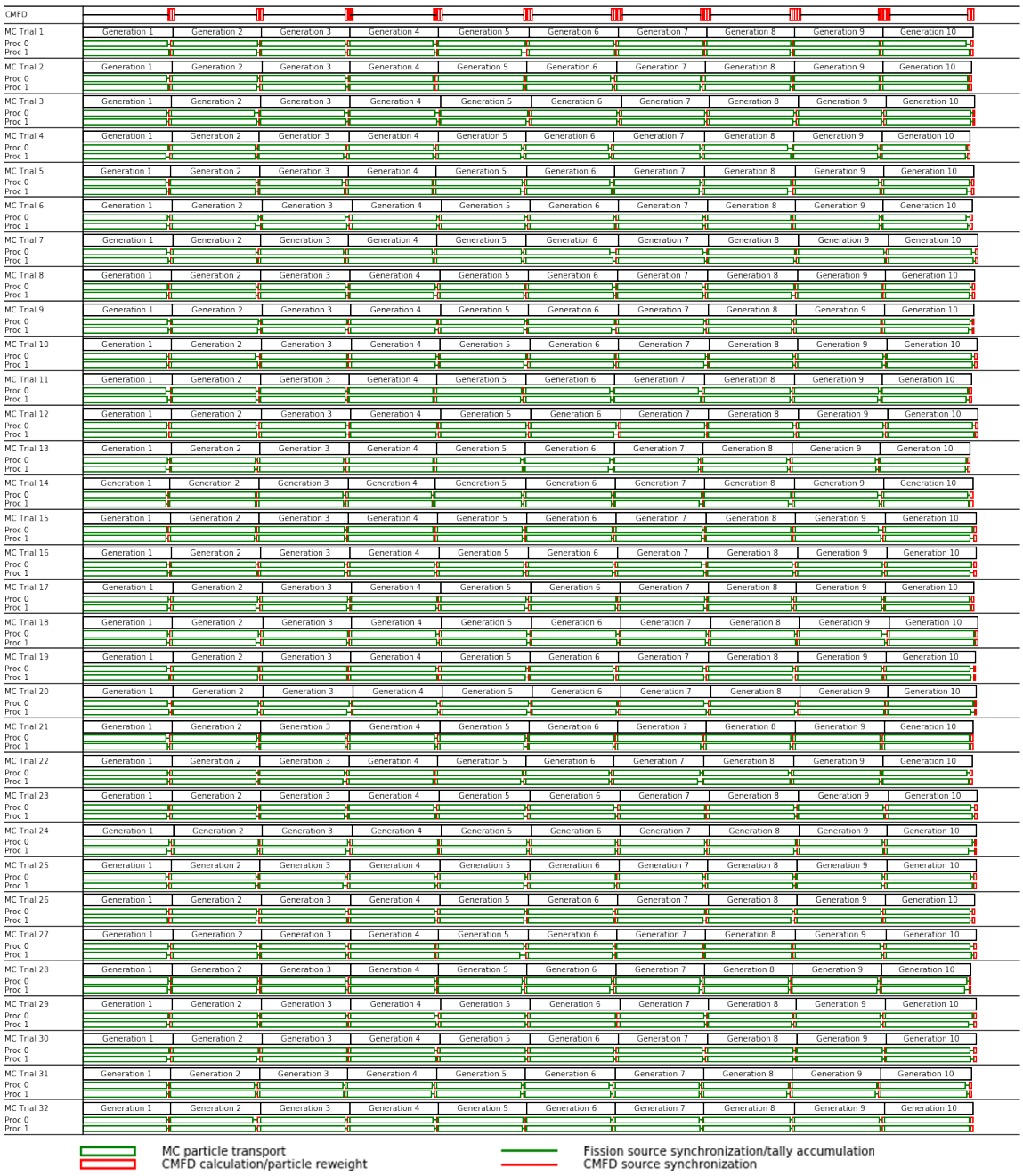


Figure 5-12: Performance profiling plot for run strategy 6 on the 3-D BEAVRS problem.

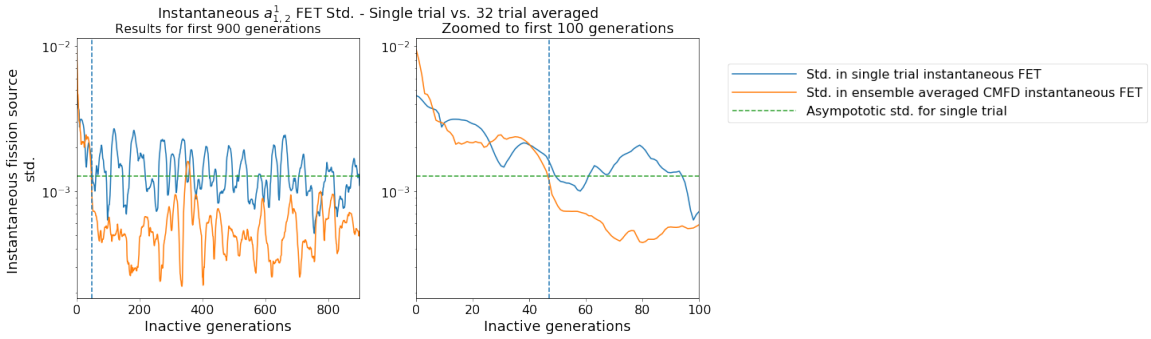


Figure 5-13: Standard deviation in instantaneous $a^1_{1,2}$ FET coefficient the 3-D BEAVRS problem. The blue line plots the single trial standard deviation for run strategy 1, the orange line plots the standard deviation computed across 32 trials for run strategy 5, the horizontal dashed green plots the asymptotic standard deviation, and the vertical blue line plots the generation where the orange line crosses the asymptotic dashed green line. The right plot is the zoomed version of the left plot.

Highlights

- The findings from previous chapters are applied to the 3-D BEAVRS model. This 3-D test case is altered by modifying water densities in each water region so that a radial tilt is induced to the stationary fission source, which mimics the radial tilt observed in the actual BEAVRS core.
- FET coefficients are calculated and used to assess stationarity in the 3-D BEAVRS core by representing the spatial domain as a separable Zernike expansion in the radial and azimuthal directions, and Legendre expansion in the axial direction.
- The asynchronous ensemble averaged CMFD run strategy is shown to be the optimal run strategy for the 3-D BEAVRS problem, improving load balancing by $\sim$10% when compared to a single trial approach while providing accurate and simple variance estimates.
- The full core solution requires 9,083 CPU-hours on INL's Falcon super computer, and yields a relative uncertainty level of 0.954% on 2-D axially integrated assembly-sized meshes and a relative uncertainty level of 9.59% on 3-D quarter assembly-sized meshes.

Part IV

Conclusions

Chapter 6

Conclusions and Perspectives

The main objective of this thesis was to develop a novel run strategy for Monte Carlo simulations of typical reactor problems. This objective was motivated by the need to efficiently run such simulations on large-scale computer systems while addressing many of the outstanding issues with Monte Carlo methods. Before analyzing overall runtime associated with Monte Carlo, two initial steps were taken to carry out this objective. Firstly, an FET-based diagnostic was devised in order to delineate between the inactive and active generations, and this stopping criteria was shown to accurately determine when stationarity occurred in the fission source. This work was the basis for all ensuing runtime analysis, providing a systematic approach to assessing the number of inactive generations and hence the overall runtime in the inactive generations associated with various Monte Carlo run strategies. Secondly, a sensitivity study of CMFD parameters was conducted in order to assess how these parameters affected the evolution of Monte Carlo simulations in both the inactive and active generations. This sensitivity study identified the key parameters that affected variance levels in both the inactive and active generations and provided a framework to ensure similar quality of solutions between the CMFD-accelerated and un-accelerated cases. The findings from this work were ultimately used to establish the optimal quantities for parameters such as CMFD mesh size, maximum window size, and generation size for all subsequent Monte Carlo runtime analysis.

The newly developed run strategy in this thesis, coined *asynchronous ensemble aver-*

aged CMFD, was then shown to increase parallel efficiency and improve load balancing by 5-10% on representative 1-D, 2-D, and 3-D reactor problems, while providing a simple and reliable approach to measuring real variance estimates. Arguments for how ensemble averaging can be applied to nonlinear multiphysics coupling schemes as well as GPU-based computer systems were also presented. In light of these advantages, this chapter summarizes how the ensemble averaged CMFD run strategy addresses the overarching objective of this thesis. Section 6.1 highlights the key results of this thesis, Section 6.2 underscores the key contributions of this thesis to the field of Monte Carlo reactor physics, and Section 6.3 discusses avenues for future work in the field of ensemble averaging.

6.1 Summary of Work

The following sections outline the key results of this thesis. Section 6.1.1 highlights the merits of an FET-based stationarity diagnostic, Section 6.1.2 lays out the recommendations for the various parameter choices associated with CMFD source acceleration. Section 6.1.3 lists the pros and cons associated with various Monte Carlo run strategies, while Section 6.1.4 presents the arguments for an ensemble averaging approach. Finally, the runtime results for the 3-D BEAVRS benchmark are discussed in Section 6.1.5.

6.1.1 FET-based Stationarity Diagnostics

Monte Carlo simulations are divided into two stages - the inactive and active generations. During the inactive generations, an initial source distribution is iterated on until it reaches stochastic convergence. At this point, the active generations commence and the tallying process can begin, where quantities of interest such as power distribution and reaction rates are computed. Typically, Shannon entropy is calculated in order to determine stationarity in the fission source, but as is shown in this thesis, an entropy-based approach can lead to scenarios of false stationarity, especially for higher-dimensional problems as well as for problems with large degrees of asymmetry relative to the stationary fission source.

Instead, an FET-based stationarity diagnostic is devised, where the stationarity in a set of FET coefficients computed over a predefined basis set is assessed in order to determine overall stationarity in the fission source. The fission source is represented in one-dimension as a Legendre polynomial expansion, as a Zernike radial and azimuthal expansion in two dimensions, and a separable Legendre and Zernike expansion in three dimensions. In doing so, it is shown that analyzing the stationarity behavior of low-order FET coefficients provides an accurate methodology for determining stationarity in the fission source, while the average behavior of high-order FET coefficients represents the degree of system-level noise in the Monte Carlo simulation.

Since FET's are meshless and each source particle contributes to the basis functions anywhere in space, examining the stationarity in FET coefficients helps to ascertain whether each spatial mode has stopped fluctuating within statistical limits. On the other hand, Shannon entropy is a mesh-based approach that examines the migration of particles in each mesh cell from generation to generation in order to signal stationarity. If this migration occurs slowly, this frequently leads to situations of false convergence when using Shannon entropy.

Moreover, stationarity analysis indicates that a larger number of inactive generations is required to reach stationarity for problems with larger dominance ratios and for problems with larger generation sizes. Finally, it is shown that post-processing the fission bank allows for the efficient calculation of FET coefficients with runtimes on par with Shannon entropy calculations, and the FET-based diagnostic can also be applied to problems that utilize CMFD source acceleration. Such a formulation allows for the exact savings in terms of generations to stationarity when utilizing CMFD when compared to an un-accelerated Monte Carlo simulation.

6.1.2 Best Practices for CMFD Source Acceleration

While investigating the parameters associated with CMFD source acceleration in Chapter 3, it is found that while CMFD may reduce the number of generations to stationarity, it can often times introduce larger noise levels to the underlying fission source when

compared to the case without CMFD acceleration. When using the expanding window scheme to reduce these fluctuations, it is shown that increasing the maximum CMFD window size helps to reduce variance levels in the instantaneous fission source, leading to the notion of the *CMFD penalty factor*. This factor represents the minimum window size required in an expanding window scheme for a given generation size so that the standard deviation in the underlying fission source of a CMFD-accelerated simulation is comparable to that of an un-accelerated scheme, and this penalty factor is independent of generation size. Here, it is found that the CMFD penalty factor for the 1-D homogeneous case is roughly 64, the penalty factor the 2-D BEAVRS case is between 32 and 64, and the penalty factor is 32 for the 3-D BEAVRS case. If the maximum window size does not meet this penalty factor, a larger variance in the instantaneous fission source arises in the CMFD-based solution when compared to the un-accelerated case with the same generation size.

The CMFD sensitivity study also shows that a fine-mesh CMFD solution and coarse-mesh CMFD solution exhibit similar variance levels when generation size is kept fixed. However, a fine-mesh solver is much more computationally costly to run, and thus it is recommended to use a coarse CMFD mesh in order to reduce overall runtime.

In the active generations, simulations that attain maximum window sizes equivalent to the CMFD penalty factor are able to reach initial RMS error levels in line with the un-accelerated case. However, as the number of active generations simulated increases, correlation effects tend to dominate convergence rates when a coarse RMS error mesh is used, irrespective of whether CMFD is used or not. It is also observed that correlation effects in the active generations tend to be stronger when a larger maximum window size is used, as evidenced by the slower convergence rates in the accumulated fission source. Therefore, it is recommended not to utilize CMFD source acceleration in the active generations, as it adds computational overhead to the Monte Carlo transport process without providing better convergence rates.

Finally, a runtime analysis of CMFD source acceleration is conducted in order to assess how to reach stationarity in the inactive generations as quickly as possible. Here, a hybrid simulation strategy is recommended that uses CMFD source acceleration for

initial inactive generations while prematurely turning off CMFD after a loosely converged solution has been achieved in order to reduce runtime per generation and achieve instantaneous fission source variance levels on par with the un-accelerated Monte Carlo case. On a single compute node with a generation size of 10 million, a CMFD-based run strategy is able to reduce time to stationarity by 52% for the 1-D homogeneous test problem with an offset starting source, and by 60% for the 2-D BEAVRS problem with a symmetric starting source.

6.1.3 Pros and Cons of Various Monte Carlo Run Strategies

Given the significant degradation in active generation convergence rates associated with Monte Carlo simulations, Chapter 4 examines the validity and efficacy of an ensemble averaging approach, where multiple independent trials of Monte Carlo with varying starting seeds are simulated in parallel. Such a simulation strategy allows for the computation of real variance across independent ensembles of neutrons, and the initial RMS error reduces at a $1/\sqrt{M}$ rate, where M is the number of independent ensembles. However, a larger number of independent trials (equivalently, a larger generation size per trial) equates to a larger number of source particles being discarded in the inactive generations. Thus CMFD is employed in the inactive generations to reduce time to stationarity.

Through a preliminary comparison of run strategies on 32 compute nodes of INL's Falcon supercomputer cluster, it was found that a multiple trial ensemble averaging approach improves parallel efficiency and load balancing on Monte Carlo runs with a larger availability of computational resources. However, fission source stationarity is based on a single trial of source data instead of considering the aggregate fission source, so this yields a less stringent stationarity criteria and hence a higher residual error in the active generations. On the other hand, running a single trial with a large generation size assesses stationarity on the aggregate fission source but comes at the cost of reduced parallelization due to the synchronizing step at the end of each generation in order to perform fission source synchronization and global tally accumulation across all

running processes. Similar to the findings from Chapter 3, overall time to stationarity is minimized for run strategies that employ CMFD source acceleration when compared to those that do not. Specific to the multiple trial CMFD case, however, a minimum generation size is required in order for the CMFD solver to run without failing, and such a strategy is susceptible to solver failures if the generation size per ensemble is too small.

In order to address these issues, the ensemble averaged CMFD run strategy is proposed, where all CMFD and FET stationarity calculations are offloaded to a separate compute node while still running ensembles of Monte Carlo in parallel. The bulk synchronous implementation communicates CMFD and FET data back and forth between MC nodes and the CMFD node using blocking two-sided communication, and such an approach suffers from poor parallelization due to the synchronizing step across all processes to accumulate CMFD and FET data onto the CMFD node. In order to reduce communication overhead with ensemble averaged CMFD, an asynchronous implementation that uses one-sided RMA to communicate FET and CMFD data is shown to minimize latency in the CMFD update step on each MC node.

These various run strategies are compared in terms of overall computational time to stationarity, and the asynchronous ensemble averaged CMFD approach is shown to reduce overall runtime without introducing additional variance bias. Bulk synchronous ensemble averaged CMFD (and hence asynchronous ensemble averaged CMFD) results in lower runtime compared to the multiple trial ensemble averaging case when an apples-to-apples single trial stopping criteria is applied. For single trial simulations with small generations, batch statistics are used to compute real variance levels. However, this results in higher variance levels as well as higher uncertainties in the variance when compared to the variance levels as computed across the multiple trials of the bulk synchronous ensemble averaged CMFD case, keeping total runtime constant. Finally, the asynchronous ensemble averaging case results in higher parallel efficiency when compared to single trial simulations with large generation sizes, reducing overall runtimes by roughly 5-10% on representative 1-D, 2-D, and 3-D reactor problems.

Table 6.1 summarizes the disadvantages associated with the preliminary Monte Carlo run strategies proposed in Chapter 4, where the asynchronous ensemble averaged CMFD

Table 6.1: Disadvantages associated with various Monte Carlo run strategies.

Multiple Trial MC	• Stationarity based on single trial stopping criteria • Doesn't utilize CMFD
Multiple Trial CMFD	• Stationarity based on single trial stopping criteria • CMFD solver is likely to fail if generation size per ensemble is too small
Single Trial MC w/ Large Gen.	• Poor parallel efficiency • No method for real variance calculations • Doesn't utilize CMFD
Single Trial CMFD w/ Large Gen.	• Poor parallel efficiency • No method for real variance calculations
Bulk Synchronous Ens. Avg. CMFD	• Poor parallel efficiency
Single Trial MC w/ Small Gen.	• Requires a large number of active generations • Batch statistics result in higher variance levels and higher uncertainty in the variance • Doesn't utilize CMFD
Single Trial CMFD w/ Small Gen.	• Requires a large number of active generations • Batch statistics result in higher variance levels and higher uncertainty in the variance

is recommended as the optimal run strategy to address all of these issues. The specific advantages of the asynchronous ensemble averaged CMFD approach are presented in the following section.

6.1.4 Arguments for Ensemble Averaging and Ensemble Averaged CMFD

This thesis develops the ensemble averaged CMFD run strategy with the aim of addressing many of the outstanding issues associated with Monte Carlo methods. On the

issue of source acceleration, this run strategy utilizes CMFD as a means to reduce the computational burden in the inactive generations. From a parallelization perspective, ensemble averaged CMFD uses one-sided RMA to communicate data, which is shown to improve load balancing on large-scale computer systems. Such an implementation also helps to provide better fault tolerance on such large-scale computer facilities, as results can continue to be collected across operational compute nodes if any compute node were to go offline during a node failure. Moreover, ensemble averaging provides a simple and accurate method for calculating real variance estimates across statistically independent ensembles of neutrons, and the use of an aggregate trial stopping criteria ensures that a sufficient number of inactive generations are simulated so that initial error levels converge at ideal $1/\sqrt{N}$ rates, where N is the total number of Monte Carlo ensembles simulated in parallel.

The CMFD feedback loop can also be viewed as a surrogate for the non-linear coupling process between Monte Carlo and deterministic solvers for thermal hydraulic feedback, isotopic depletion, and transient analysis. As such, the findings from Chapters 3, 4, and 5 can be applied to workflows that involve non-linear coupling schemes with Monte Carlo neutronics as well. A CMFD windowing scheme and a larger generation size can be used in order to reduce variance in the underlying Monte Carlo fission source before the coupling step, while ensemble averaging provides an avenue for calculating real variance across multiple Monte Carlo trials while aggregating coupling parameters across all trials in order to reduce statistical fluctuations during the coupling step.

Additionally, the ensemble averaging method has applications to GPU-based computer systems, where each CPU on a compute node can be designated as the CMFD server responsible for communicating CMFD parameters, while an ensemble of neutrons running the maximum possible number of particles per generation can be run with event-based Monte Carlo on each GPU of the compute node. In this manner, the total target generation size is distributed among the available GPU's in a compute job, while CMFD fission source updates and FET stationarity diagnosis continue to occur on the CPU. Such an approach helps to alleviate the memory limitations inherent to GPU processors, and can be applied to the GPU-based exascale machines that are currently

being designed.

6.1.5 Extension of Work to 3-D Full-Core Reactor Problems

As a final test case for the ensemble averaging approach, the run strategies introduced in Chapter 4 are applied to the 3-D BEAVRS full-core model. A radial tilt is induced to this test case by modifying the water density of each water region in the core, and this was done so as to mimic the actual tilt observed in operating data for the BEAVRS core and make the problem more difficult to converge.

A uniform starting source guess now results in a larger number of inactive generations to reach fission source stationarity, as the asymmetries in the stationary fission source distribution activate a larger number of spatial modes of the radial distribution. An FET-based diagnostic is used to determine stationarity by representing the 3-D spatial domain as a separable Zernike expansion in the radial and azimuthal directions, and a Legendre expansion in the axial direction. Once again, the entropy-based stationarity diagnostic underpredicts generations to stationarity, which results in a noticeable error due to a premature stationarity diagnosis.

Finally, runtime analysis shows a nearly 10% reduction in computational time to stationarity for the asynchronous ensemble averaged CMFD case when compared to a single trial CMFD approach, while yielding consistent levels of relative uncertainties among different run strategies. The full core solution requires 9,083 CPU-hours on INL's Falcon supercomputer, and yields an average relative uncertainty level (standard deviation divided by mean) of 0.954% on 2-D axially integrated assembly-sized meshes and a relative uncertainty level of 9.59% on 3-D quarter assembly-sized meshes. Thus, the asynchronous ensemble averaged CMFD approach is once again deemed as the optimal run strategy for full-core 3-D Monte Carlo simulations on the grounds of better load balancing and ability to calculate simple and accurate variance levels.

6.2 Contributions

- Devised a meshless, FET-based diagnostic that analyzes the behavior of specific spatial moments of the fission source in order to characterize stationarity and prevent situations of false convergence.
- Conducted a rigorous parameter sensitivity to pinpoint the key parameters that affect the performance of CMFD source acceleration in both the inactive and active generations. More importantly, introduced the notion of the *CMFD penalty factor* to describe how a minimum window size needs to be attained in order for variance levels between CMFD-accelerated and un-accelerated simulations to be comparable.
- Investigated the merits of various Monte Carlo run strategies on large-scale computing systems, with an emphasis not only on total runtime but also ability to calculate accurate real variance estimates.
- Proposed the novel ensemble averaged CMFD run strategy and demonstrated that this is the optimal run strategy for large-scale Monte Carlo simulation on the basis of parallel efficiency, ability to calculate real variance estimates, and better fault tolerance.
- Described the framework for how ensemble averaging can be applied to GPU-based computer systems and Monte Carlo workflows with non-linear multiphysics coupling for thermal-hydraulics, depletion, and/or transient analysis.
- Applied the ensemble averaged CMFD methodology to the radially tilted 3-D BEAVRS full-core problem with figures of merit for overall runtime and relative uncertainty levels.

6.3 Future Work

This thesis introduced ensemble averaging as a viable run strategy in the field of Monte Carlo neutronics and outlined its merits beyond simply a verification tool for real variance estimates. However, future work is needed in order to realize further benefits of ensemble averaging outside of its current scope of steady-state Monte Carlo simulations with CMFD source acceleration.

6.3.1 Particle Ramp-Up Approach to Ensemble Averaging

The results from Section 5.3 indicate a significant runtime is required for a 3-D full-core problem to converge to low statistical uncertainties on fine, pellet-level meshes. A single trial stopping criteria is shown to reduce number of inactive generations and overall runtime by requiring a looser convergence criteria in the inactive generations, but this is also likely to induce larger biases in the fission source that would therefore require a larger number of independent ensembles in order to reach target uncertainty levels.

The work in Chapter 5 also illustrates that a significant source of computational overhead for the asymmetric 3-D BEAVRS case stems from the fact that each ensemble has to converge independently to a stationary fission source from a poor starting guess, thus magnifying the number of inactive generations across all independent trials. Instead, a particle ramp up approach could be used, where only a single ensemble is simulated initially to get a loosely converged fission source distribution as a probability distribution over a fine mesh. Then, the number of ensembles being run in parallel can be ramped up using this approximate probability distribution as the starting guess for the inactive generations among these ensembles. The inactive generations during this stage of ensemble averaging would therefore be responsible for reducing the fission source error residual according to the aggregate trial stopping criteria. Such an approach is similar to the particle ramp-up technique pursued by Lund et al. [81], but instead of modifying the source distribution directly, a probability distribution function defined over the fission source mesh cells is passed during the particle ramp-up stage, where a single ensemble is responsible for removing the large bias from a poor starting source guess, while the

number of independent ensembles being run is ramped up from a coarsely converged starting source in order to refine the error residual in the inactive generations.

Assuming that on the order of 20-30 inactive generations would be required for ensembles to converge to an aggregate trial stopping criteria from a coarsely converged starting source distribution, the projected total runtime to reach 1% uncertainties will be on the order of 80,000 - 100,000 CPU-hours.

6.3.2 Further Parallelizations to the CMFD Solver

The CMFD solver used in this thesis relies on OpenMP multithreading in order to reduce the computational overhead associated with running the CMFD solver on a problem with a large number of mesh cells. While CMFD multithreading can lead to degradation in parallel efficiency due to false sharing if the problem is too small, it was shown in Chapter 5 to effectively reduce runtime in the CMFD solver for the 3-D BEAVRS problem, which used 18 threads to solve the CMFD equations over 8,492 total CMFD mesh cells.

An additional avenue for parallelization is to use domain decomposed CMFD in order to introduce an additional level of parallelism to the CMFD solver. Such an implementation splits the problem into disjoint subdomains over which each process would solve for, transferring only any boundary data that need to be accessed between processes through MPI communication. Domain decomposition has been shown to successfully increase parallel efficiency on CMFD acceleration coupled with deterministic solvers [74, 82], and would help to increase CPU utilization as well. This is especially important if ensemble averaged CMFD was to be implemented on GPU-based machines, where each compute node has a designated number of CPU's. With domain decomposition, each of these CPU's would be responsible for solving CMFD equations on a subdomain of the problem, instead of relying on a central CPU to carry out all calculations solely through OpenMP multithreading. However, communication costs between CPU's need to be assessed in order to determine the overall effectiveness of domain decomposed CMFD.

6.3.3 Testing on Larger Compute Jobs

The runtime analysis conducted in Chapter 4 and 5 used roughly 30 compute nodes totaling job sizes on the order of thousands of CPU cores in order to assess overall time to solution. Compared to the total size of modern supercomputers, this represents a small fraction of how large compute jobs can be. Thus, in order to illustrate the complete picture of how effective ensemble averaged CMFD can be, a more thorough analysis of runtime behavior for a larger number of computational resources should be pursued. As was shown in the performance profiling plots for ensemble averaged CMFD in Figures 4-11, 4-13, and 5-12, the CPU utilization of the CMFD node is relatively low, with a visibly large portion of CPU idling. This indicates that the presence of additional MC nodes that communicate with the CMFD node would increase CPU utilization on the CMFD node without sacrificing on data contention among MC nodes. However, as the size of the compute job gets larger and larger, latency issues and bandwidth limits associated with node-to-node data transfer may start to arise, especially if these compute nodes are separated by a large physical space without a direct communication pathway.

6.3.4 Testing on GPU-based Computer Systems

As indicated in Section 4.4, an ensemble averaging approach to Monte Carlo neutronics can be extended to GPU-based computer systems. By extending this work to GPU-based systems, the runtime savings from performing Monte Carlo calculations on the GPU can be realized, while avoiding memory limitations from having to run large generation sizes by distributing the total number of particles per generations across all available GPU's. All CMFD calculations and FET stationarity diagnosis can occur on the CPU's of each compute node, but further work is required in order to ascertain how data should be transferred between the CPU and GPU. Similar to the CPU-only ensemble averaging approach taken in this thesis, all data on each GPU can be transferred directly to a central CPU in order to conduct CMFD updates and FET stationarity diagnosis. On the other hand, a multi-level approach can also be pursued, where GPU's communicate only directly with the CPU's that are available on the compute node, while CPU's across

different compute nodes are responsible for periodically synchronizing CMFD and FET data with each other.

6.3.5 CMFD as a Surrogate for Nonlinear Multiphysics Coupling

As a final avenue for future work, this thesis discusses how CMFD can be viewed as a surrogate for the non-linear coupling process, where ensemble averaging can be leveraged to improve Monte Carlo coupling capabilities with deterministic solvers. Typically, single trial Monte Carlo simulations with large generation sizes are used to reduce fluctuations in the tallying process, but accurate variance estimates are difficult to compute for such simulation strategies. Instead, ensemble averaging allows for variance levels to be calculated across multiple independent Monte Carlo trials while aggregating coupling parameters across all trials in order to reduce statistical fluctuations during the coupling step. This is the issue faced in Harper's thesis, for example, where his work tightly couples a Monte Carlo-based low-order CMFD solution with a subchannel code using tally derivatives to inexpensively generate an approximately converged TH coupling scheme, which is later refined by high-fidelity Monte Carlo [39]. Harper is forced to run multiple independent simulations in order to gauge whether fluctuations during the Monte Carlo surrogate iterations were within statistical bounds of these independent trials. Alternatively, ensemble averaging could also be used to verify whether similar variance levels across ensembles can be calculated during the Monte Carlo surrogate iterations.

Appendices

Appendix A

Uranyl Sulfate Material Composition

The material composition of uranyl sulfate is taken from the water boiler experiment outlined in the International Handbook of Evaluated Criticality Safety Benchmark Experiments handbook, identification number IEU-SOL-THERM-004 [83]. The material density and nuclide densities are included in Table A.1

Table A.1: Material composition for 14.7% enriched uranyl sulfate

Uranyl sulfate 14.7% enriched, density (g/cc)	1.33175
Isotopes	Number density (atom/b-cm)
U-235	9.6795E-05
U-234	7.4257E-07
U-238	5.5518E-04
S-32	6.5272E-04
O-16	3.5185E-02
H-1	6.2538E-02

Appendix B

Stationarity Behavior of FET's and Shannon Entropy for 2-D BEAVRS Benchmark Problem

Figure B-1 plots the value of Shannon entropy and the slowest converging FET coefficient as a function of generations, along with a vertical line to signify when stationarity occurs for each case where a varied number of neutrons per generation is used. Likewise, Figure B-2 plots the variance of the slowest converging FET coefficients as a function of generations for different neutrons per generation. Finally, Figure B-3 plots the generations to stationarity using the FET-based and entropy-based diagnostics for five independent trials. Figures B-1, B-2, and B-3 use a starting source that is uniform over the fissionable areas. The analogous plots using a starting source that is offset to the right half of the core is shown in Figures B-4, B-5, and B-6.

These results show that as generation size increases, a larger number of generations in the inactive generations are required to reach stationarity. This is consistent with the idea that when a larger generation size is used, a lower threshold variance level is attainable by the FET coefficients, but this threshold requires a larger number of inactive generations to reach.

These results also illustrate how when an asymmetric starting source is used, large differences occur between the entropy-based and FET-based approaches. In such scenar-

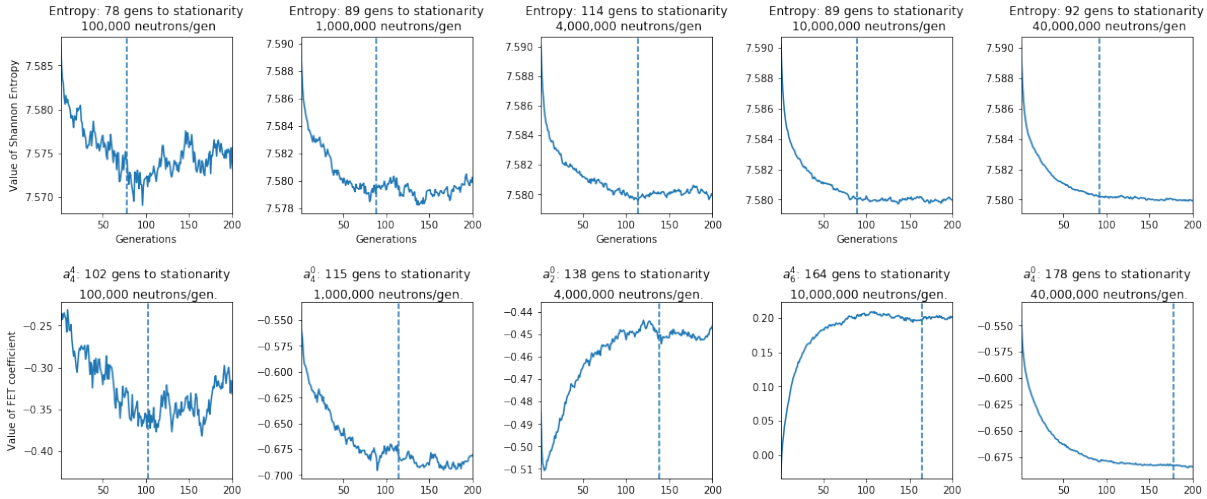


Figure B-1: Value of Shannon entropy and slowest-converging FET coefficient as a function of neutrons per generations for 2-D BEAVRS benchmark problem. Values correspond to a starting source that is uniform in the fissionable areas.

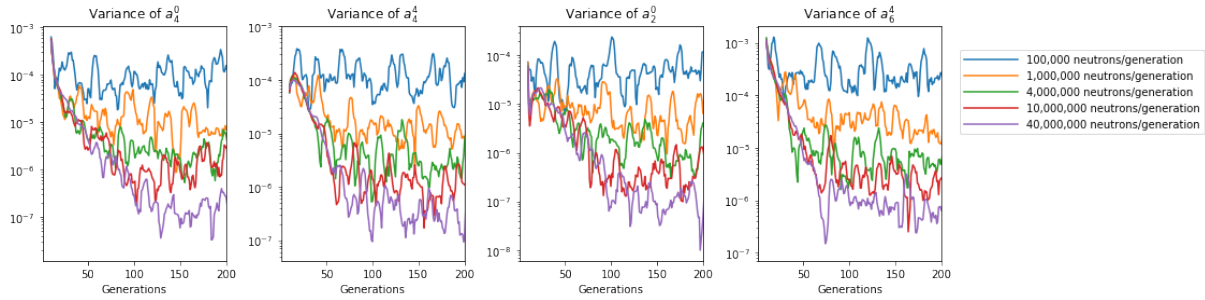


Figure B-2: variance of slowest converging FET coefficients for 2-D BEAVRS benchmark problem using different generation sizes. Values correspond to a starting source that is uniform in the fissionable areas.

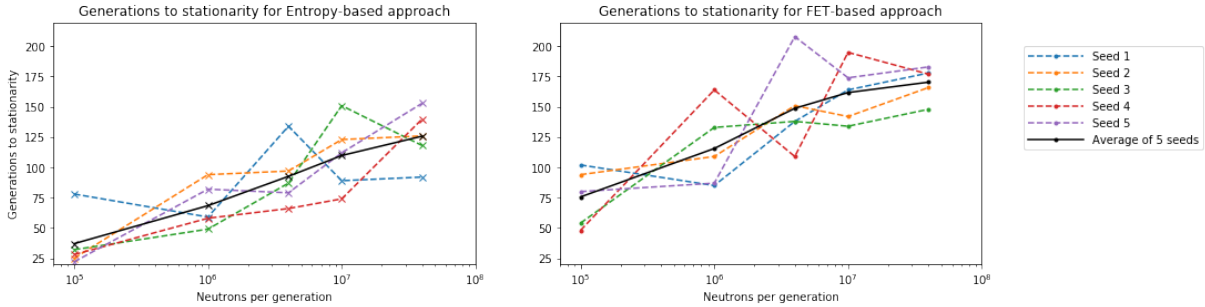


Figure B-3: Generations to stationarity as a function of number of neutrons for five independent trials of 2-D BEAVRS benchmark problem using entropy-based (right) and FET-based (left) approaches. The starting source is uniform in the fissionable areas.

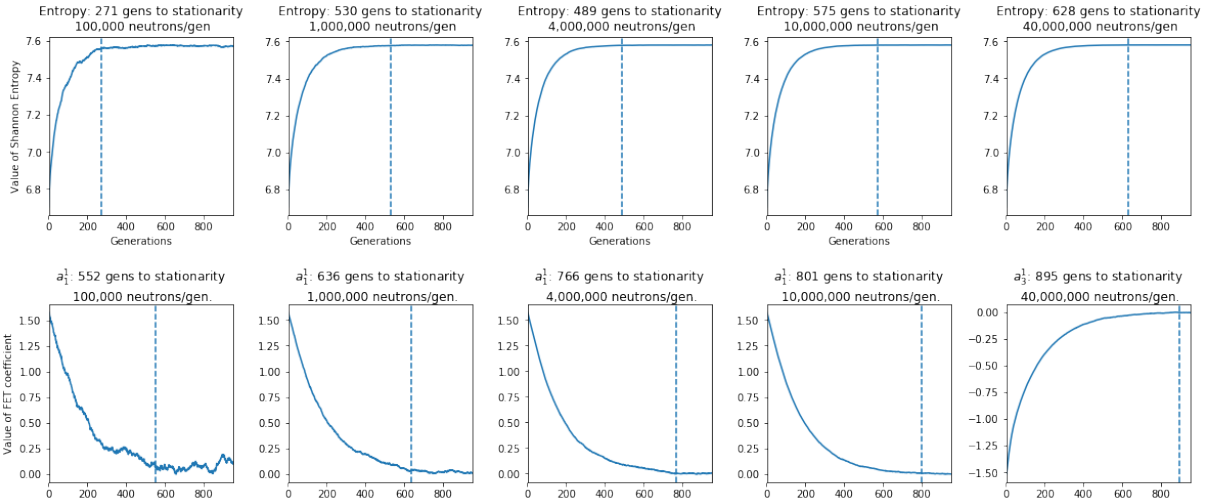


Figure B-4: Value of Shannon entropy and slowest-converging FET coefficient as a function of neutrons per generations for 2-D BEAVRS benchmark problem. Values correspond to a starting source that is offset to the right half of the core.

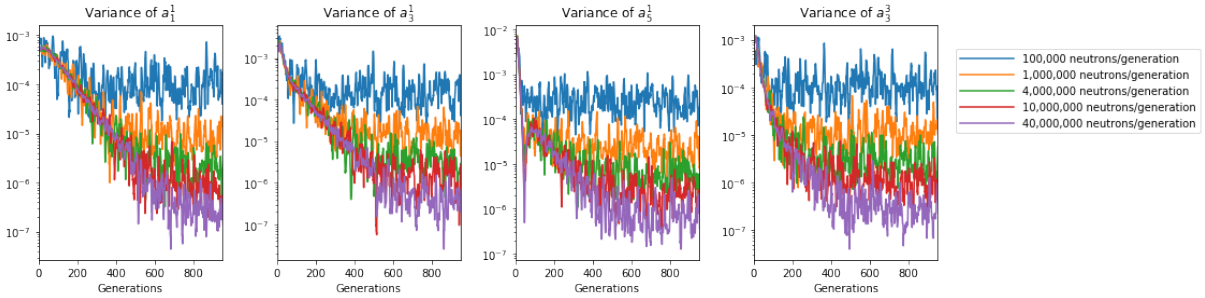


Figure B-5: variance of slowest converging FET coefficients for 2-D BEAVRS benchmark problem using different generation sizes. Values correspond to a starting source that is offset to the right half of the core.

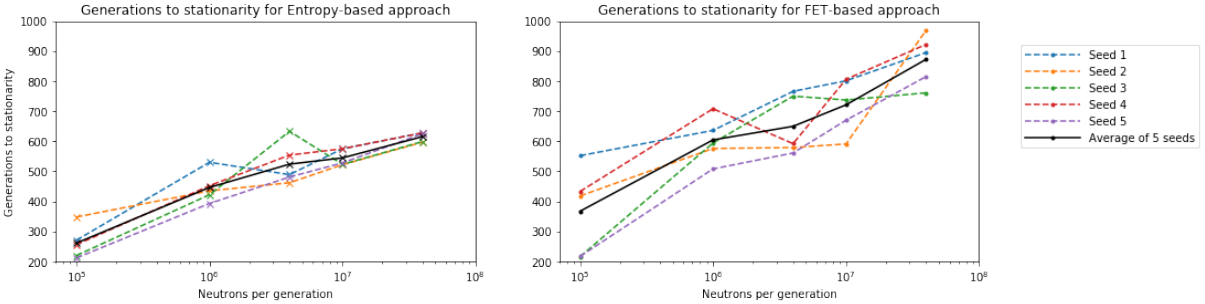


Figure B-6: Generations to stationarity as a function of number of neutrons for five independent trials of 2-D BEAVRS benchmark problem using entropy-based (right) and FET-based (left) approaches. The starting source is offset to the right half of the core.

ios, the value of Shannon entropy flattens out much more quickly than the values for the FET coefficients, and therefore Shannon entropy mispredicts when stationarity occurs. Finally, these results are independent of seed number, where the average behavior of the five trials shows that there is close agreement between the FET-based and entropy-based diagnostics with a symmetric starting point source. On the other hand, the average of the five trials yields a larger discrepancy between the two methods when an asymmetric starting point source is used, where Shannon entropy prescribes an insufficient number of generations to stationarity.

Appendix C

Equations for Linear Prolongation CMFD

This section defines the equations used to apply a CMFD prolongation step that assumes a linearly varying fission source in a single dimension. In this case, the weight adjustment factor from Equation 3.4 will now be a function of the spatial dimension of the axis with which linear prolongation is applied, and is represented as:

$$f_{g,m}^{\mathrm{LP}}(z) = a_{g,m}(z - z_{m-1/2}) + b_{g,m} \tag{C.1}$$

Here, $a_{g,m}$ and $b_{g,m}$ represent the slope and intercept of the linear fit of the weight adjustment factor through mesh cell with spatial index m and group index g. $z_{m-1/2}$ and $z_{m+1/2}$ will be used to represent the lower and upper bounds of spatial mesh m. The slope $a_{g,m}$ is defined as a function of the weight adjustment factors of the neighboring mesh cells:

$$a_{g,m} = \frac{f_{g,m+1} - f_{g,m-1}}{z_{m+1/2} - z_{m-1/2}} \tag{C.2}$$

and the intercept shifts $f_{g,m}^{\mathrm{LP}}(z)$ vertically such that $f_{g,m}^{\mathrm{LP}}(z_m) = f_{g,m}$, where $f_{g,m}$ is the weight adjustment factor when a fixed source approximation is used. This implies that

$$b_{g,m} = f_{g,m} - \frac{f_{g,m+1} - f_{g,m-1}}{2} \tag{C.3}$$

and $b_{g,m}$ is chosen in this manner so that the fission source is conserved within the mesh cell for the linear source prolongation case and the fixed source prolongation case. In other words, with this formulation it can be shown that:

$$\frac{1}{z_{m+1/2} - z_{m-1/2}} \int_{z_{m-1/2}}^{z_{m+1/2}} f_{g,m}^{\mathrm{LP}}(z) dz = f_{g,m} \tag{C.4}$$

At boundary locations where neighbor source regions are not defined, the boundary weight values will be set as the neighbor weight value. For example, if the CMFD problem has M spatial indices indexed from 1 to M, then:

$$a_{g,1} = \frac{f_{g,2} - f_{g,1}}{z_{3/2} - z_{1/2}} \tag{C.5}$$

$$b_{g,1} = f_{g,1} - \frac{f_{g,2} - f_{g,1}}{2} \tag{C.6}$$

$$a_{g,M} = \frac{f_{g,M} - f_{g,M-1}}{z_{M+1/2} - z_{M-1/2}} \tag{C.7}$$

$$b_{g,M} = f_{g,M} - \frac{f_{g,M} - f_{g,M-1}}{2} \tag{C.8}$$

Figure C-1 plots the weight adjustment factor as predicted by a fine mesh CMFD solution with a fixed source prolongation (orange), a coarse mesh CMFD solution with a fixed source prolongation (blue), and a coarse mesh CMFD solution with a linear source prolongation (green). One issue with such such a forumlation is that the slope $a_{g,m}$ is not a function of $f_{g,m}$, so in order to improve this prolongation model, a bi-linear fit could also be used where $a_{g,m}$ for the left half of cell m is dependent on $f_{g,m}$ and $f_{g,m-1}$, and $a_{g,m}$ the right half of cell m is dependent on $f_{g,m}$ and $f_{g,m+1}$.

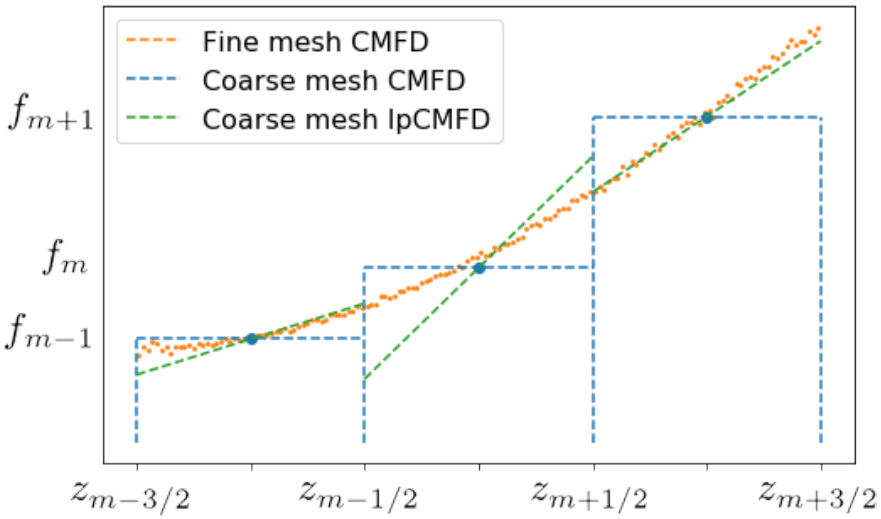


Figure C-1: Illustration of weight adjustment factor as prescribed by a fine mesh CMFD solution with a fixed source prolongation (orange), a coarse mesh solution with a fixed source prolongation (blue), and a coarse mesh CMFD solution with a linear source prolongation (green).

Appendix D

Effect of RMS Error Mesh Size in the Limit of Large Number of Active Generations

This section plots RMS error as a function of active generations for the un-accelerated CMFD accelerated cases, in order to show the effect of RMS error mesh size on the active generations. Figure D-1 plots RMS error as a function of number of active generations for the 1-D problem, where the first subplot uses no CMFD acceleration while the remaining four subplots vary the CMFD mesh size and maximum window size. Each subplot has two trendlines - one where RMS error is calculated with a coarse 20cm error mesh and one where a fine 0.4cm error mesh is used. Here it can be seen that when a coarse error mesh is used, initial error levels start lower than the case where a fine error mesh is used. However, due to correlation effects, the RMS error level for the fine error mesh case catches up to the coarse mesh error. Eventually, in the limit of a large number of active generations, the error levels converge to similar values, and RMS error no longer depends on the error mesh size used. The fluctuations observed at higher number of active generations is likely due to the statistical uncertainties in the reference solution at this level of error convergence.

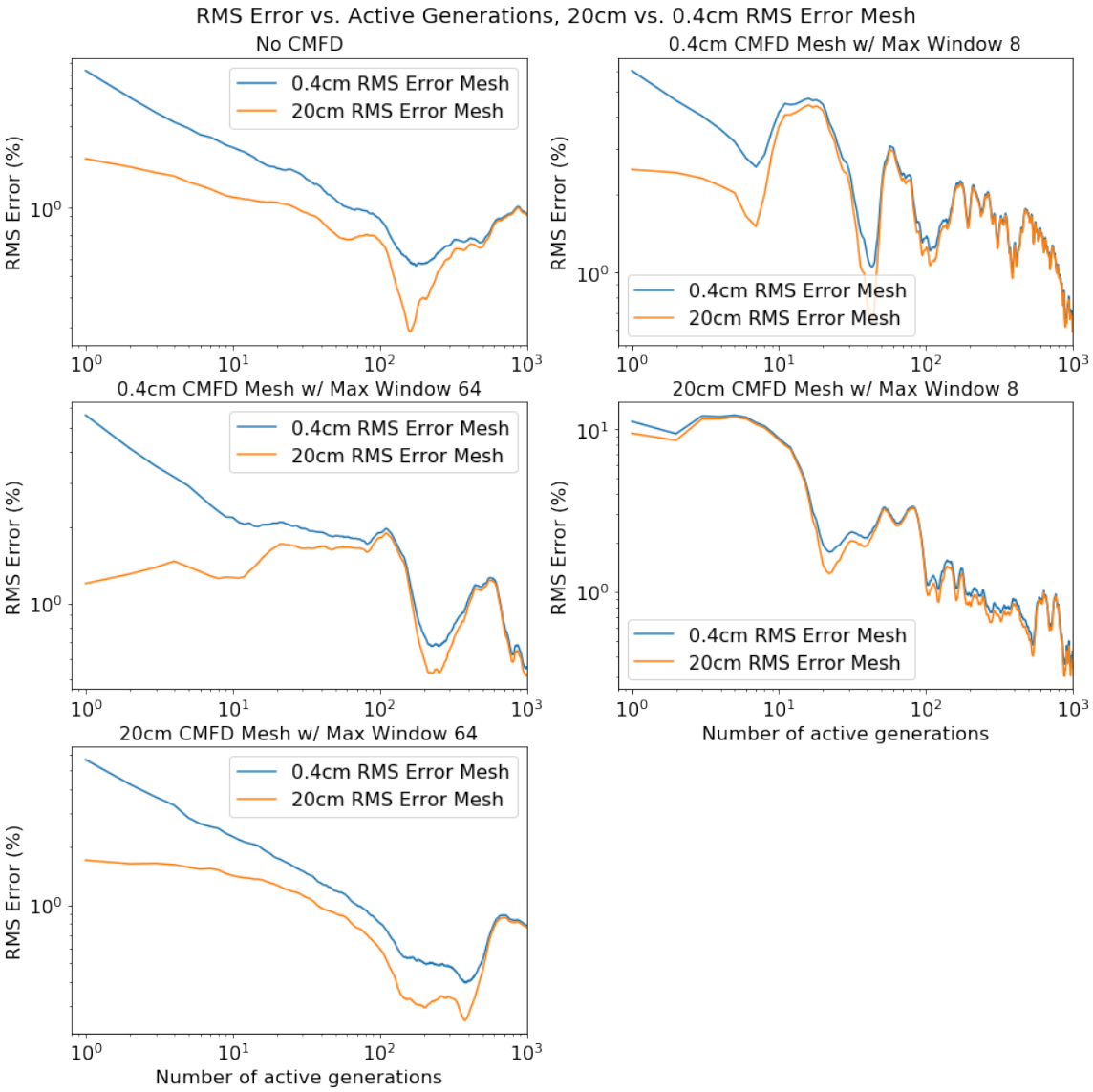


Figure D-1: RMS error as a function of number of active generations - 20cm RMS error mesh vs. 0.4cm RMS error mesh for a variety of CMFD / No CMFD configurations.

Appendix E

Effect of Generation Size on Active Generations

Figures E-1 to E-4 show the analogous plots of Figures 3-22 to 3-25, but with a generation size of 10 million instead of 1 million. For these cases, the increase in the generation size shifts the overall RMS error by a factor of $1/\sqrt{10}$, but correlation effects are still present when a coarse RMS error mesh is used. When a fine RMS error mesh is used, convergence in the active generations follows $1/\sqrt{N}$ behavior, but deviates slightly from this as the number of active generations increases. Finally, as shown in Figure E-4, the use of linear prolongation helps to reduce initial RMS errors when a coarse CMFD mesh is used.

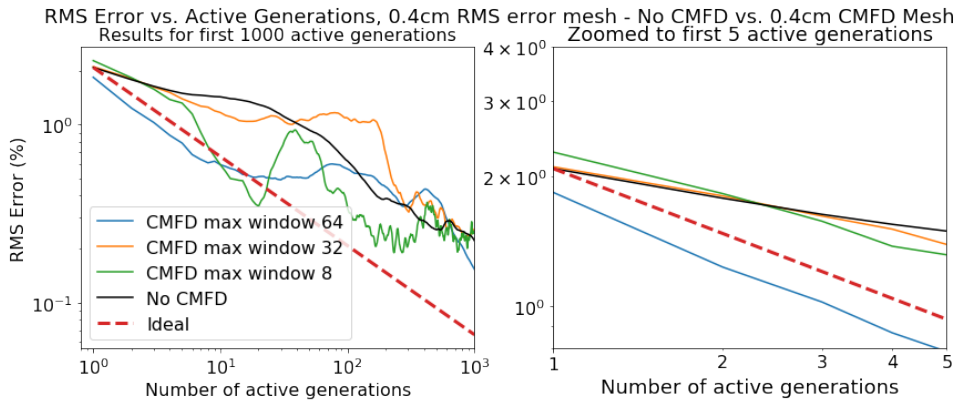


Figure E-1: RMS error as a function of number of active generations - No CMFD vs. 0.4cm CMFD mesh with varying window size for 1-D problem. Accumulated fission source is computed with a generation size of 10 million and RMS error mesh of 0.4cm. RMS error is shown for the first 1000 active generations (left) and zoomed into the first 5 active generations (right).

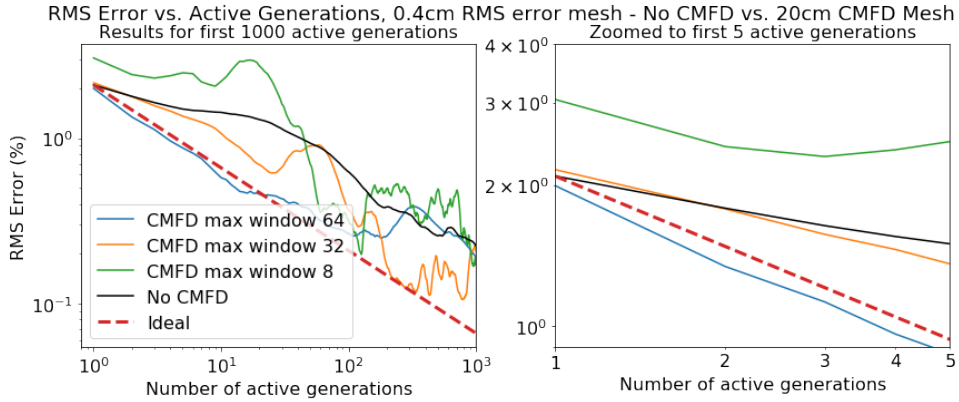


Figure E-2: RMS error as a function of number of active generations - No CMFD vs. 20cm CMFD mesh with varying window size for 1-D problem. Accumulated fission source is computed with a generation size of 10 million and RMS error mesh of 0.4cm. RMS error is shown for the first 1000 active generations (left) and zoomed into the first 5 active generations (right).

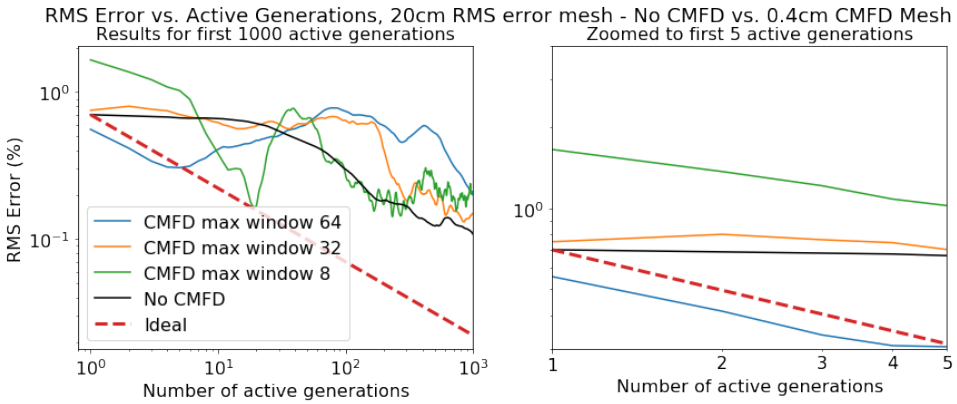


Figure E-3: RMS error as a function of number of active generations - No CMFD vs. 0.4cm CMFD mesh with varying window size for 1-D problem. Accumulated fission source is computed with a generation size of 10 million and RMS error mesh of 20cm. RMS error is shown for the first 1000 active generations (left) and zoomed into the first 5 active generations (right).

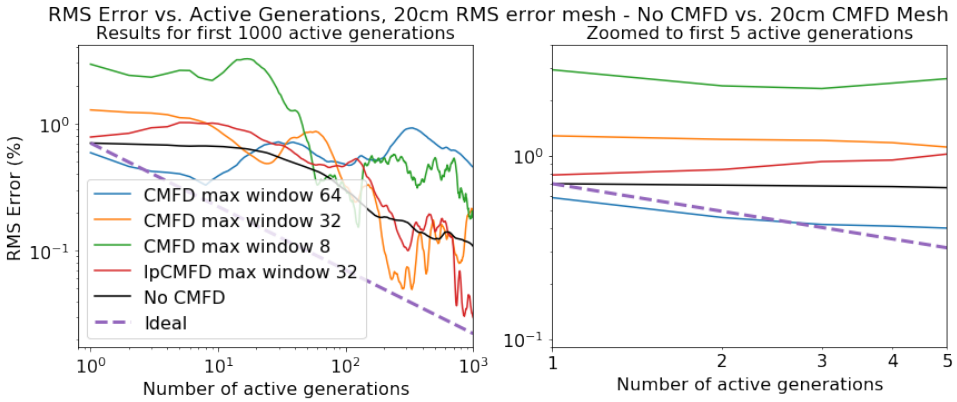


Figure E-4: RMS error as a function of number of active generations - No CMFD vs. 20cm CMFD mesh with varying window size for 1-D problem. Accumulated fission source is computed with a generation size of 10 million and RMS error mesh of 20cm. RMS error is shown for the first 1000 active generations (left) and zoomed into the first 5 active generations (right).

Appendix F

Variance Underestimation Ratio for Correlated Tally Samples Using Generation Statistics and Batch Statistics

This section computes real variance levels when correlated samples are used, both for the case of generation statistics as well as for batch statistics. The equations that result will then be applied to the 2-D BEAVRS problem to compute the real standard deviation levels shown in Figure 4.3.3. These derivations are largely taken from work by Romano et. al and Miao [28, 76].

F.1 Variance of the Sample Mean in the Presence of Correlated Generations

We begin once again with the problem formulation at the beginning of Chapter 4, where we let random variable Y with mean μ and variance σ^2, which represents the Monte Carlo tally score for a single source particle. If $\{Y_j | 1 \leq j \leq B\}$ represents the set of all realizations of Y and $\{Y_j | (i-1)H \leq j \leq iH\}$ denotes the realizations for generation i

with generation size H, then the random variable X will be used to define the generation average of the tally scores, i.e.:

$$X_i = \frac{1}{H} \sum_{j=(i-1)H+1}^{iH} Y_j \tag{F.1}$$

Here, it is assumed that all Y_j are independent and identically distributed and X converges to a normal distribution with mean μ and variance σ^2/H. The variance of $\overline{X}(N)$, which is the average of X over N active generations, can be written as:

$$\begin{aligned} \mathrm{Var}(\overline{X}(N)) &= \mathrm{Var}\left(\frac{1}{N}\sum_{i=1}^{N} X_i\right) = \frac{1}{N^2}\mathrm{Var}\left(\sum_{i=1}^{N} X_i\right) \\ &= \frac{1}{N^2}\left(\sum_{i=1}^{N}\mathrm{Var}(X_i) + 2\sum_{k=1}^{N-1}\sum_{i=1}^{N-k}\mathrm{Cov}(X_i, X_{i+k})\right) \\ &= \frac{1}{N^2}\left(\frac{N\sigma^2}{H} + 2\sum_{k=1}^{N-1}(N-k)\mathrm{Cov}(X_i, X_{i+k})\right) \end{aligned} \tag{F.2}$$

where the last expression results from modeling $X(n)$ as a stationary time series with covariance that depends on the batch lag k. Next, we define the ACC p_k with generation lag k as

$$\rho_k = \frac{\mathrm{Cov}(X_i, X_{i+k})}{\sqrt{\mathrm{Var}(X_i)\mathrm{Var}(X_{i+k})}} \tag{F.3}$$

and now the covariance can be expressed as:

$$\mathrm{Cov}(X_i, X_{i+k}) = \frac{\sigma^2\rho_k}{H} \tag{F.4}$$

Substituting Equation F.4 into Equation F.2, we obtain:

$$\begin{aligned} \mathrm{Var}(\overline{X}(N)) &= \frac{1}{N^2}\left(\frac{N\sigma^2}{H} + 2\sum_{k=1}^{N-1}\frac{(N-k)\sigma^2\rho_k}{H}\right) \\ &= \frac{\sigma^2}{HN}\left(1 + 2\sum_{k=1}^{N-1}\left(1 - \frac{k}{N}\right)\rho_k\right) = \frac{\sigma^2}{HN}r(N) \end{aligned} \tag{F.5}$$

In the absence of inter-generational correlation effects, the variance in the sample mean converges at a $1/N$ rate, however, the term $r(N)$, defined as:

$$r(N) = 1 + 2\sum_{k=1}^{N-1}\left(1 - \frac{k}{N}\right)\rho_k \tag{F.6}$$

represents the penalty factor incurred in the active convergence rate due to inter-generational correlation effects and is a function of the ACC's ρ_k, where $r(N) > 1$ if $\rho_k > 0$. The term $r(N)$ is the same term discussed at the beginning of Chapter 4. This analysis shows that unless correlation coefficients are calculated for problems where generation statistics are used in the active generations, the variance levels will be underestimated by a factor of $r(N)$.

F.2 Variance of the Sample Mean in the Presence of Correlated Batches

A batching algorithm helps to calculate tally variance levels without the need for calculating $r(N)$, but this is only true if the number of generations in a batch is sufficiently large. Typically, however, there is no established way to evaluate how large the number of generations in a batch needs to be in order for the assumption of uncorrelated batches to hold. As will be shown, if this value is too small, correlation effects can still persist between batches and need to be properly accounted for in order to report accurate variance estimates.

Suppose the total number of active generations N is divided into N_B=N/B batches, where B represents the number of consecutive generations per batch. Similar to Equation F.1, the batch k tally value is calculated as:

$$\chi_k = \frac{1}{B}\sum_{i=(k-1)B+1}^{kB} X_i \tag{F.7}$$

and the variance in the sample mean over $\{\chi_1, \ldots, \chi_{N_B}\}$ can be represented as

$$\begin{aligned}\mathrm{Var}(\overline{\chi}(N_B)) &= \mathrm{Var}\left(\frac{1}{N_B}\sum_{i=1}^{N_B}\chi_i\right) = \frac{1}{N_B^2}\mathrm{Var}\left(\sum_{i=1}^{N_B}\chi_i\right) \\ &= \frac{1}{N_B^2}\left(\frac{N_B\sigma^2}{HB} + 2\sum_{k=1}^{N_B-1}\sum_{i=1}^{N_B-k}\mathrm{Cov}(\chi_i,\chi_{i+k})\right)\end{aligned} \tag{F.8}$$

which takes the same form as Equation F.2. Miao relates the covariance between batch J and K in terms of the batch lag and the generational correlation coefficients as [28]:

$$\mathrm{Cov}(\chi_J,\chi_K) = \frac{\sigma^2}{HB^2}\sum_{l=(\Delta-1)B+1}^{(\Delta+1)B-1}(\Delta B - |\Delta B - l|)\rho_l \tag{F.9}$$

where $\Delta = |J-K| \geq 1$ is the batch lag and ρ_l is the correlation coefficient between generations with lag l, equivalent to the definition in Equation F.4. Substituting Equation F.9 into Equation F.8,

$$\begin{aligned}\mathrm{Var}(\overline{\chi}(N_B)) &= \frac{1}{N_B^2}\left(\frac{N_B\sigma^2}{HB} + 2\sum_{\Delta=1}^{N_B-1}(N_B-\Delta)\mathrm{Cov}(\chi_i,\chi_{i+\Delta})\right) \\ &= \frac{1}{N_B^2}\left(\frac{N_B\sigma^2}{HB} + 2\frac{\sigma^2}{HB^2}\sum_{\Delta=1}^{N_B-1}(N_B-\Delta)\sum_{l=(\Delta-1)B+1}^{(\Delta+1)B-1}(\Delta B - |\Delta B - l|)\rho_l\right) \\ &= \frac{\sigma^2}{N_B HB}\left(1 + \frac{2}{B}\sum_{\Delta=1}^{N_B-1}\left(1-\frac{\Delta}{N_B}\right)\sum_{l=(\Delta-1)B+1}^{(\Delta+1)B-1}(\Delta B - |\Delta B - l|)\rho_l\right) \\ &= \frac{\sigma^2}{N_B HB}r(N_B)\end{aligned} \tag{F.10}$$

Once again, we relate the variance in the sample mean over N_B batches, and define the variance underestimation ratio $r(N_B)$ for the case of batch statistics as:

$$r(N_B) = 1 + \frac{2}{B}\sum_{\Delta=1}^{N_B-1}\left(1-\frac{\Delta}{N_B}\right)\sum_{l=(\Delta-1)B+1}^{(\Delta+1)B-1}(\Delta B - |\Delta B - l|)\rho_l \tag{F.11}$$

The extra $1/B$ term in the second term in this equation comes from the fact that the expression for covariance was substituted directly into Equation F.8 instead of expressing

the covariance in terms of the ACC's between batches. However, if a large enough B is used, $r(N_B)$ should approach 1 and thus each batch can be assumed to be uncorrelated with each other. The following section uses Equation F.11 to determine what the variance underestimation ratio is for the 2-D BEAVRS problem with $B = 75$, as was used in Section 4.3.3.

F.3 Variance Underestimation Ratio for the 2-D BEAVRS Problem with B=75

Equation F.11 relates the variance underestimation ratio to the case where a batching method is used to determine tally variance levels. This equation is used to compute the variance underestimation ratio and hence the correlation-corrected standard deviation levels for the 2-D BEAVRS case presented in Figure 4-17. Generational ACC's ρ_k are computed using a single-term exponential fit of observed autocorrelation coefficients for the 2-D BEAVRS problem in the form $\rho_k = \rho_0 q^k$. The exact parameters for this exponential fit are taken from work by Li, who studies the convergence behavior of ACC's in detail for both the un-accelerated and the CMFD-accelerated cases of the 2-D BEAVRS problem [1]. In order to estimate the variance underestimation ratio for $B = 75$, we compare the ratio of $r(N_B = 75)$ to $r(N_B = 200)$, where it is assumed that $r(N_B) \approx 1$ when $N_B = 200$. For the un-accelerated case, $\rho_0 = 0.351$, and $q = 0.969$, which yields a value of $r(N_B = 75)/r(N_B = 200) = 2.87$. Likewise, for the CMFD-accelerated case, $\rho_0 = 0.251$, and $q = 0.984$, and this results in a value of $r(N_B = 75)/r(N_B = 200) = 3.92$. Therefore, there is a considerable amount of variance underestimation when a value of $B = 75$ is used for both un-accelerated and CMFD-accelerated cases, and correlation effects cannot be neglected when using a batching method with such a small number of generations per batch. This is evident by the fact that a considerable amount of correlation still persists even at a generation lag of 75. For reference, the fitted ACC's as a function of generation lag are shown in Figure F-1 for both the un-accelerated and CMFD-accelerated case.

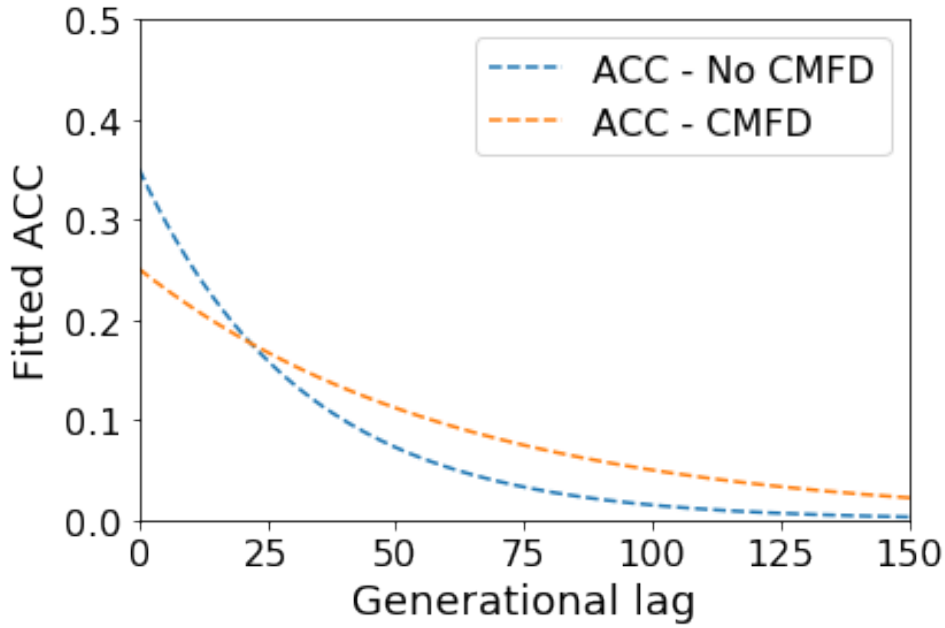


Figure F-1: Fitted ACC's as a function of generation lag for the 2-D BEAVRS problem using un-accelerated Monte Carlo (blue) and CMFD acceleration (orange) [1].

References

[1] L. LI et al., *Acceleration methods for Monte Carlo particle transport simulations*, PhD thesis, Massachusetts Institute of Technology, 2017.

[2] S. KUMAR, J. LIANG, B. FORGET, and K. SMITH, "BEAVRS: An integral full core multi-physics PWR benchmark with measurements and uncertainties," *Progress in Nuclear Energy*, **129**, 103488 (2020).

[3] T. BAHADIR and S.-Ö. LINDAHL, "Studsvik's next generation nodal code SIMULATE-5," *Proc. Proceedings of the ANFM-2009 conference, Advances in Nuclear Fuel Management IV (Hilton Head Island, South Carolina, USA, 2009)*, 2009.

[4] D. G. CACUCI, *Handbook of Nuclear Engineering: Vol. 1: Nuclear Engineering Fundamentals; Vol. 2: Reactor Design; Vol. 3: Reactor Analysis; Vol. 4: Reactors of Generations III and IV; Vol. 5: Fuel Cycles, Decommissioning, Waste Disposal and Safeguards*, Springer Science & Business Media (2010).

[5] J. E. HOOGENBOOM, "Is Monte Carlo Embarassingly Parallel?," *Proc. PHYSOR–Advances in Reactor Physics–Linking Research, Industry, and Education* (2012).

[6] J. E. THORNTON, "The cdc 6600 project," *Annals of the History of Computing*, **2**, *4*, 338 (1980).

[7] G. E. MOORE et al., "Cramming more components onto integrated circuits," 1965.

[8] M. R. FAHEY et al., "Theta and Mira at Argonne National Laboratory," in *Contemporary High Performance Computing*, p. 31–61, CRC Press, 2019.

[9] R. STEVENS, J. RAMPRAKASH, P. MESSINA, M. PAPKA, and K. RILEY, "Aurora: Argonne's Next-Generation Exascale Supercomputer," ANL (Argonne National Laboratory (ANL), Argonne, IL (United States)) (2019).

[10] Oak Ridge National Laboratory, "Direction of Discovery," `https://www.olcf.ornl.gov/frontier/`, 2019, [Online; accessed 12/2/2020].

[11] D. J. KELLY, T. M. SUTTON, and S. C. WILSON, "MC21 analysis of the nuclear energy agency Monte Carlo performance benchmark problem," *Proc. PHYSOR–Advances in Reactor Physics–Linking Research, Industry, and Education* (2012).

[12] J. E. HOOGENBOOM, W. R. MARTIN, and B. PETROVIC, "Monte Carlo performance benchmark for detailed power density calculation in a full size reactor core," *Benchmark specifications revision*, **1**, *1* (2011).

[13] P. K. ROMANO and B. FORGET, "Parallel fission bank algorithms in Monte Carlo criticality calculations," *Nuclear Science and Engineering*, **170**, *2*, 125 (2012).

[14] P. K. ROMANO and B. FORGET, "Reducing parallel communication in Monte Carlo simulations via batch statistics," *Transactions*, **107**, *1*, 519 (2012).

[15] P. K. ROMANO, A. R. SIEGEL, B. FORGET, and K. SMITH, "Data decomposition of Monte Carlo particle transport simulations via tally servers," *Journal of Computational Physics*, **252**, 20 (2013).

[16] N. E. HORELIK, *Domain decomposition for Monte Carlo particle transport simulations of nuclear reactors*, PhD thesis, Massachusetts Institute of Technology, 2015.

[17] H. J. ALME, G. H. RODRIGUE, and G. B. ZIMMERMAN, "Domain decomposition models for parallel monte carlo transport," *The Journal of Supercomputing*, **18**, *1*, 5 (2001).

[18] R. PROCASSINI, M. O'BRIEN, and J. TAYLOR, "Load balancing of parallel Monte Carlo transport calculations," Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States) (2005).

[19] T. A. BRUNNER and P. S. BRANTLEY, "An efficient, robust, domain-decomposition algorithm for particle Monte Carlo," *Journal of Computational Physics*, **228**, *10*, 3882 (2009).

[20] J. LIANG, K. WANG, Y. QIU, X. CHAI, and S. QIANG, "Domain decomposition strategy for pin-wise full-core Monte Carlo depletion calculation with the reactor Monte Carlo code," *Nuclear Engineering and Technology*, **48**, *3*, 635 (2016).

[21] N. C. SLY, B. T. MERVIN, S. W. MOSHER, T. M. EVANS, J. C. WAGNER, and G. I. MALDONADO, "Verification of the shift Monte Carlo code with the C5G7 reactor benchmark," *Proc. Advances in Reactor Physics (PHYSOR 2012)* (2012).

[22] J. L. SALCEDO-PÉREZ, "On-node performance optimization of a Monte-Carlo transport code for leadership architectures," Master's thesis, Massachusetts Institute of Technology, 2019.

[23] R. M. BERGMANN and J. L. VUJIĆ, "Algorithmic choices in WARP–A framework for continuous energy Monte Carlo neutron transport in general 3D geometries on GPUs," *Annals of Nuclear Energy*, **77**, 176 (2015).

[24] S. P. HAMILTON and T. M. EVANS, "Continuous-energy Monte Carlo neutron transport on GPUs in the Shift code," *Annals of Nuclear Energy*, **128**, 236 (2019).

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

[26] B. R. HERMAN, *Monte Carlo and thermal hydraulic coupling using low-order nonlinear diffusion acceleration*, PhD thesis, Massachusetts Institute of Technology, 2014.

[27] E. DUMONTEIL and T. COURAU, "Dominance ratio assessment and Monte Carlo criticality simulations: dealing with high dominance ratio systems," *Nuclear Technology*, **172**, *2*, 120 (2010).

[28] J. MIAO, *Correlations in Monte Carlo eigenvalue simulations: uncertainty quantification, prediction and reduction*, PhD thesis, Massachusetts Institute of Technology, 2018.

[29] A. YAMAMOTO, K. SAKATA, and T. ENDO, "Prediction on underestimation of variance for fission rate distribution in Monte-Carlo calculation," *Transactions*, **110**, *1*, 515 (2014).

[30] H. J. SHIM and C. H. KIM, "Real variance estimation using an intercycle fission source correlation for Monte Carlo eigenvalue calculations," *Nuclear Science and Engineering*, **162**, *1*, 98 (2009).

[31] H. J. SHIM et al., "McCARD: Monte Carlo code for advanced reactor design and analysis," *Nucl. Eng. Technol*, **44**, *2*, 161 (2012).

[32] J. MIAO, B. FORGET, and K. SMITH, "Analysis of correlations and their impact on convergence rates in Monte Carlo eigenvalue simulations," *Annals of Nuclear Energy*, **92**, 81 (2016).

[33] J. MIAO, B. FORGET, and K. SMITH, "Predicting correlation coefficients for Monte Carlo eigenvalue simulations with multitype branching process," *Annals of Nuclear Energy*, **112**, 307 (2018).

[34] J. MIAO, B. FORGET, and K. SMITH, "Improving variance convergence rate in monte carlo eigenvalue simulations via delayed neutrons," *Annals of Nuclear Energy*, **142**, 107376 (2020).

[35] P. K. ROMANO, S. P. HAMILTON, R. O. RAHAMAN, A. NOVAK, E. MERZARI, S. M. HARPER, P. C. SHRIWISE, and T. M. EVANS, "A Code-Agnostic Driver Application for Coupled Neutronics and Thermal-Hydraulic Simulations," *Nuclear Science and Engineering* (2020).

[36] E. MERZARI, R. RAHAMAN, M. MIN, and P. FISCHER, "Performance analysis of nek5000 for single-assembly calculations," *Proc. Fluids Engineering Division Summer Meeting*, volume 51562, p. V002T09A031, American Society of Mechanical Engineers, 2018.

[37] M. A. KREHER, B. FORGET, and K. SMITH, "Single-Batch Monte Carlo Multiphysics Coupling," *Proc. Proc. Int. Conf. Mathematics and Computational Methods Applied to Nuc. Sci. & Eng*, 2019.

[38] D. F. GILL, D. P. GRIESHEIMER, and D. L. AUMILLER, "Numerical methods in coupled Monte Carlo and thermal-hydraulic calculations," *Nuclear Science and Engineering*, **185**, *1*, 194 (2017).

[39] S. M. HARPER, *Tally Derivative Based Surrogate Models for Faster Monte Carlo Multiphysics*, PhD thesis, Massachusetts Institute of Technology, 2020.

[40] A. JOHNSON, *Efficient Coupled Transport-Depletion Sequence Using Hybrid Monte Carlo-Reduced Order Scheme*, PhD thesis, Georgia Institute of Technology, 2020.

[41] P. K. ROMANO, *Parallel algorithms for Monte Carlo particle transport simulation on exascale computing architectures*, PhD thesis, Massachusetts Institute of Technology, 2013.

[42] P. K. ROMANO and B. FORGET, "The OpenMC Monte Carlo particle transport code," *Annals of Nuclear Energy*, **51**, 274 (2013).

[43] P. K. ROMANO, N. E. HORELIK, B. R. HERMAN, A. G. NELSON, B. FORGET, and K. SMITH, "OpenMC: A state-of-the-art Monte Carlo code for research and development," *Proc. SNA+ MC 2013-Joint International Conference on Supercomputing in Nuclear Applications+ Monte Carlo*, p. 06016, EDP Sciences, 2014.

[44] S. KUMAR, B. FORGET, and K. SMITH, "Stationarity Diagnostics using Functional Expansion Tallies," *Annals of Nuclear Energy*, **143**, 107388 (2020).

[45] T. UEKI and F. B. BROWN, "Stationarity diagnostics using Shannon entropy in Monte Carlo criticality calculation I: F test," *Trans. Am. Nucl. Soc*, **87**, 156 (2002).

[46] T. UEKI and F. B. BROWN, "Informatics approach to stationarity diagnostics of the Monte Carlo fission source distribution," *TRANSACTIONS-AMERICAN NUCLEAR SOCIETY* (2003).

[47] T. UEKI and F. B. BROWN, "Stationarity modeling and informatics-based diagnostics in Monte Carlo criticality calculations," *Nuclear science and engineering*, **149**, *1*, 38 (2005).

[48] T. UEKI, "Information theory and undersampling diagnostics for Monte Carlo simulation of nuclear criticality," *Nuclear science and engineering*, **151**, *3*, 283 (2005).

[49] F. B. BROWN, "On the use of Shannon entropy of the fission distribution for assessing convergence of Monte Carlo criticality calculations," *Proc. ANS topical meeting on reactor physics (PHYSOR 2006). Canadian Nuclear Society, Canada*, 2006.

[50] B. SHI and B. PETROVIC, "Investigation of the entropy-based convergence diagnostics in MCNP5 using the OECD/NEA fuel storage pool benchmark problem," *International Conference on the Physics of Reactors 2010, PHYSOR 2010*, **1**, 605 (2010).

[51] A. HAGHIGHAT, *Monte Carlo Methods for Particle Transport*, Crc Press (2016).

[52] M. NOWAK, J. MIAO, E. DUMONTEIL, B. FORGET, A. ONILLON, K. S. SMITH, and A. ZOIA, "Monte Carlo power iteration: Entropy and spatial correlations," *Annals of Nuclear Energy*, **94**, 856 (2016).

[53] M. T. WENNER, *Development of new source diagnostic methods and variance reduction techniques for Monte Carlo eigenvalue problems with a focus on high dominance ratio problems*, PhD thesis, University of Florida, 2010.

[54] T. UEKI and B. S. CHAPMAN, "A Mesh-Input-Free On-The-Fly Source Convergence Indicator in Monte Carlo Power Method," *Journal of Nuclear Science and Technology*, **48**, *5*, 805 (2011).

[55] H. J. SHIM and C. H. KIM, "Stopping criteria of inactive cycle Monte Carlo calculations," *Nuclear Science and Engineering*, **157**, *2*, 132 (2007).

[56] D. P. GRIESHEIMER, W. R. MARTIN, and J. P. HOLLOWAY, "Convergence properties of Monte Carlo functional expansion tallies," *Journal of Computational Physics*, **211**, *1*, 129 (2006).

[57] D. P. GRIESHEIMER, W. R. MARTIN, and J. P. HOLLOWAY, "Estimation of flux distributions with Monte Carlo functional expansion tallies," *Radiation Protection Dosimetry*, **115**, *1-4*, 428 (2005).

[58] D. P. GRIESHEIMER, *Functional expansion tallies for Monte Carlo simulations*, PhD thesis, University of Michigan, 2005.

[59] M. S. ELLIS, *Methods for including multiphysics feedback in Monte Carlo reactor physics calculations*, PhD thesis, Massachusetts Institute of Technology, 2017.

[60] K. HAYASHI, T. ENDO, and A. YAMAMOTO, "Underestimation of statistical uncertainty of local tallies in Monte Carlo eigenvalue calculation for simple and LWR lattice geometries," *Journal of Nuclear Science and Technology*, **55**, *12*, 1434 (2018).

[61] D. P. GRIESHEIMER, W. R. MARTIN, and J. P. HOLLOWAY, "A functional expansion method for Monte Carlo eigenvalue calculations," *Proc. of Monte Carlo Method: Versatility Unbounded in a Dynamic Computing World (MC2005)* (2005).

[62] L. S. WATERS, G. W. MCKINNEY, J. W. DURKEE, M. L. FENSIN, J. S. HENDRICKS, M. R. JAMES, R. C. JOHNS, and D. B. PELOWITZ, "The MCNPX Monte Carlo radiation transport code," *Proc. AIP conference Proceedings*, volume 896, p. 81–90, AIP, 2007.

[63] B. HERMAN, B. FORGET, and K. SMITH, "Utilizing CMFD in OpenMC to estimate dominance ratio and adjoint," *Transactions of the American Nuclear Society*, **109**, 1389 (2013).

[64] N. HORELIK, B. HERMAN, B. FORGET, and K. SMITH, "Benchmark for evaluation and validation of reactor simulations (BEAVRS), v1. 0.1," *Proc. Proc. Int. Conf. Mathematics and Computational Methods Applied to Nuc. Sci. & Eng*, 2013.

[65] E. W. LARSEN and J. YANG, "A functional Monte Carlo method for k-eigenvalue problems," *Nuclear Science and Engineering*, **159**, *2*, 107 (2008).

[66] R. BLOMQUIST and E. GELBARD, "Alternative implementations of the Monte Carlo power method," *Nuclear science and engineering*, **141**, *2*, 85 (2002).

[67] E. R. WOLTERS, E. W. LARSEN, and W. R. MARTIN, "Hybrid monte carlo-cmfd methods for accelerating fission source convergence," *Nuclear Science and Engineering*, **174**, *3*, 286 (2013).

[68] T. YAMAMOTO and Y. MIYOSHI, "Reliable method for fission source convergence of Monte Carlo criticality calculation with Wielandt's method," *Journal of Nuclear Science and Technology*, **41**, *2*, 99 (2004).

[69] D. J. KELLY III, B. N. AVILES, and B. R. HERMAN, "MC21 analysis of the MIT PWR benchmark: Hot zero power results," *Proc. Proceedings of the 2013 International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering-M and C 2013*, 2013.

[70] Z. LIU, K. SMITH, B. FORGET, and J. ORTENSI, "A cumulative migration method for computing rigorous transport cross sections and diffusion coefficients for LWR lattices with Monte Carlo," Idaho National Lab.(INL), Idaho Falls, ID (United States) (2016).

[71] G. A. GUNOW, *Full core 3D neutron transport simulation using the method of characteristics with linear sources*, PhD thesis, Massachusetts Institute of Technology, 2018.

[72] M. CLARK and K. F. HANSEN, *Numerical methods of reactor analysis*, volume 3, Academic Press (1964).

[73] Sun Microsystems, "OpenMP API User's Guide. Chapter 6," `https://docs.oracle.com/cd/E19205-01/819-5270/6n7c71vec/index.html`, 2010.

[74] W. WU, G. GIUDICELLI, K. SMITH, B. FORGET, D. YAO, Y. YU, and Q. LUO, "Improvements of the CMFD acceleration capability of OpenMOC," *Nuclear Engineering and Technology* (2020).

[75] T. UEKI, T. MORI, and M. NAKAGAWA, "Error estimations and their biases in Monte Carlo eigenvalue calculations," *Nuclear science and engineering*, **125**, *1*, 1 (1997).

[76] P. K. ROMANO, A. L. LUND, and A. R. SIEGEL, "Optimal Run Strategies in Monte Carlo Iterated Fission Source Simulations," *Nuclear Science and Engineering*, **188**, *1*, 43 (2017).

[77] V. MAKARASHVILI, E. MERZARI, A. OBABKO, A. SIEGEL, and P. FISCHER, "A performance analysis of ensemble averaging for high fidelity turbulence simulations at the strong scaling limit," *Computer Physics Communications*, **219**, 236 (2017).

[78] L. DALCÍN, R. PAZ, and M. STORTI, "MPI for Python," *Journal of Parallel and Distributed Computing*, **65**, *9*, 1108 (2005).

[79] Y. JO and N. Z. CHO, "Acceleration and Real Variance Reduction in Continuous-Energy Monte Carlo Whole-Core Calculation via p-CMFD Feedback," *Nuclear Science and Engineering*, **189**, *1*, 26 (2018).

[80] S. KUMAR et al., *Quantifying time-dependent uncertainty in the BEAVRS benchmark using time series analysis*, PhD thesis, Massachusetts Institute of Technology, 2018.

[81] A. L. LUND, P. K. ROMANO, and A. R. SIEGEL, "Accelerating Source Convergence in Monte Carlo Criticality Calculations Using a Particle Ramp-Up Technique," *Proc. Proc. Int. Conf. Mathematics and Computational Methods Applied to Nuc. Sci. & Eng*, 2017.

[82] G. GUNOW, B. FORGET, and K. SMITH, "Full core 3D simulation of the BEAVRS benchmark with OpenMOC," *Annals of Nuclear Energy*, **134**, 299 (2019).

[83] J. B. BRIGGS et al., "International Handbook of Evaluated Criticality Safety Benchmark Experiments," *Nuclear Energy Agency, NEA/NSC/DOC (95)* (2010).