A Time-Dependent Method of Characteristics Formulation with Time Derivative Propagation

by

Adam J. Hoffman

**A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Nuclear Engineering and Radiological Sciences)
in the University of Michigan
2013**

Doctoral Committee:

**Professor John Lee, Chair
Assistant Research Scientist Benjamin Collins
Professor Thomas Downar
Professor Edward Larsen
Professor Divakar Viswanath**

Acknowledgements

This dissertation would not have been possible without the support of several people. I would first like thank my advisor and committee chair Prof. John Lee who has guided my academic development and provided a constant example of integrity and dedication to research. I would also like to thank my committee members for their valuable feedback and guidance.

I am thankful to Dr. Brendan Kochunas, Dr. Mathieu Hursin, and Prof. Han-Gyu Joo for their assistance as I learned the source code for DeCART. I also would like to express my gratitude to Prof. Tom Downar for providing the opportunity for me to work with DeCART, as well as to the original designers of DeCART: Prof. Han-Gyu Joo, Dr. Jin-Young Cho, and Dr. Kang-Seog Kim. DeCART was an excellent teaching tool because it provided an example of effective programming techniques and a platform to explore new methods.

I would also like to acknowledge the CASL research program. CASL not only funded this research but provided the inspiration and the opportunity to pursue it. I thank Dr. Ben Collins and Dr. Brendan Kochunas for their ongoing collaboration in applying this work to the computer code MPACT to support the CASL project.

I would also like to thank my parents, Alan and Wanda Hoffman, and Rose and William McLain, and my sister, Angela, for their continuous support and encouragement over my lengthy academic career. They have never doubted my abilities, and I will continue to strive to meet their expectations.

Finally, I want to express my deep gratitude to my wife, Michaela, who supported me in this endeavor in more ways than I can count. I will forever be grateful for her encouragement, patience, and love which made this dissertation possible.

Table of Contents

List of Tables

List of Figures

List of Appendices

List of Acronyms

2SDP	– 2nd Derivative SDP
ASTC	– Analytically-integrated Space-Time Characteristics
BDC	– Time-dependent MOC using BDF
BDF	– Backward Differentiation Formula
IBDC	– Time-dependent MOC using BDF with an isotropic flux time derivative
ISCDP	– Source and Cross-section Derivative Propagation with Isotropic Correction
ISDP	– Source Derivative Propagation with Isotropic Correction
MOC	– Method of Characteristics
NSDP	– Nth Derivative SDP
RBDC	– Reference Time-dependent MOC using BDF
SDP	– Source Derivative Propagation
TSDP	– Source Derivative Propagation with Truncated Second Derivative
TDP	– Time Derivative Propagation

Abstract

In this thesis, we developed a new time-dependent neutron transport method for nuclear reactor kinetics using method of characteristics (MOC) with angular flux time derivative propagation. In contrast to conventional time integration methods which use local finite difference approximations to treat the time derivative, the new method solves for the spatially-dependent angular flux time derivative by propagation along characteristics in the spatial domain. This results in the angular flux time derivative being recast in terms of the neutron source time derivatives, and thus the new method is called Source Derivative Propagation (SDP). We developed three SDP methods using different approximations.

When the angular flux is stored using conventional time integration techniques, the memory requirements for large reactor problems are prohibitively large. As a result, most time-dependent neutron transport codes for nuclear reactor kinetics either approximate the angular flux time derivative (e.g. an isotropic assumption) or limit the size and resolution of their models. SDP circumvents this obstacle because it only requires the storage of the neutron source, which requires substantially less memory.

In the SDP methods, we approximate the source derivatives to a user-selected order of accuracy using backward differences. This is analogous to the backward differentiation formula (BDF), and our results confirmed that the high-order source derivative approximations reproduced the high-order angular flux derivative approximation of equivalent order BDF.

We assessed the SDP methods by comparison to conventional time-dependent MOC methods. This included both a reference method (RBDC) which stored the angular flux and an efficient but approximate method that assumed that the time derivative was

isotropic (IBDC). RBDC was the benchmark for accuracy, while IBDC was the benchmark for computational efficiency.

We performed error analysis for the SDP methods as well as RBDC and IBDC. The error analysis informed the refinement of the SDP methods, and clarified the circumstances in which the SDP methods are expected to be accurate.

We tested the SDP methods using the neutron transport computer code DeCART. DeCART was used to model three reactor transients based on the TWIGL and C5G7 benchmark problems. A fine time step reference solution was generated using RBDC. The SDP methods converged to the reference solution when the time step was refined and the order of the time derivative approximation increased. We also assessed the order of convergence for the SDP methods. For slow transients, the methods exhibited the expected theoretical order of convergence, but for transients driven by a step change the order of convergence was reduced.

In addition, we observed that the SDP methods accurately replicated the RBDC solution when the same time step and BDF order was used. This indicates that the propagated angular flux time derivative of SDP was accurately reproducing the local finite differenced time derivative of RBDC, regardless of the order of the method. The SDP methods were orders of magnitude more accurate than the IBDC methods.

We assessed the efficiency of the SDP methods by comparing the run-time and memory requirements of SDP with the RBDC and IBDC methods. The first-order SDP methods required about 50—100% more run-time than the first-order IBDC methods, but they required less run-time than the first-order RBDC methods. However, the high-order SDP methods did not increase the run-time relative to the first-order SDP; this is in contrast to the high-order RBDC methods.

The SDP methods required about 10% more memory than the IBDC methods of the same order. However, both the SDP and IBDC methods required about two orders of magnitude less memory than the RBDC methods. This is because storing the angular flux for RBDC completely dominates the memory requirements for time-dependent MOC. The difference was especially pronounced for the high-order RBDC methods.

Our results demonstrate that for the problems tested, the SDP methods can accurately solve the time-dependent transport equation for nuclear reactor kinetics while avoiding a prohibitive increase in memory requirements. In addition, SDP methods are capable of approximating the angular flux time derivative to high-order accuracy using backward differences without substantially increasing the memory or computational requirements. Finally, we observed that the SDP methods of various order were able to replicate the solution of the RBDC methods when the same time step size and order was used, indicating that the SDP approach to time derivative propagation accurately replicated the local finite difference approximation of RBDC.

Chapter 1

Introduction

Nuclear engineering is the application of nuclear and other sciences to design or study systems of employing ionizing radiation or radioactive material, such as nuclear reactors, radiation detectors, and nuclear medical devices. Nuclear reactor physics is a discipline within nuclear engineering which concerns the study of neutrons, fission, and the controlled nuclear chain reaction in a nuclear reactor. Nuclear reactor kinetics is a further subdivision of reactor physics which specifically concerns time-dependent phenomena over time scales of seconds or shorter. The accurate understanding and prediction of such phenomena is paramount to the safe and economic operation of nuclear reactors.

The neutron transport equation describes the motion of neutrons and their interaction with matter. The solution of the transport equation provides the neutron flux which is used to calculate nuclear reaction rates within a reactor. However, the transport equation generally does not permit analytical solutions, and it is computationally expensive to solve using numerical methods. The study and solution of approximations to the transport equation has been a predominant pursuit of reactor physicists, and the steady increase in computational power available over time has permitted steadily higher fidelity representations of the transport equation.

We are in the midst of a transition in the state-of-the art for reactor kinetics. Until recently, reactor transients have been principally modeled using the diffusion approximation to the transport equation [Sut96]. While diffusion methods are computationally efficient, there is growing interest in reactor designs and fuels for which diffusion may not be sufficiently accurate. Consequently, there is interest in the direct use of neutron transport methods for nuclear reactor kinetics [Gol01, Pau03, Cho05, Tay09, Seu11, Tal13, Tsu13], and this is increasingly practical due to the steady advances in computational power.

Method of Characteristics (MOC) [Ask72, Hal80] is a popular method for solving the steady-state neutron transport equation, especially for small, assembly-level models. Its popularity stems from its ability to resolve complex geometry without spatial homogenization. As with other transport methods, it has been extended to solve the time-dependent transport equation as well.

Most reactor kinetics methods in neutron transport and diffusion treat time dependence by discretizing the equations in time and applying a conventional time integration method to approximate the time derivative (e.g. Backward Euler, Theta Method, Runge-Kutta, etc.). In this case, the spatial and angular dependence are treated using the same approach that would be employed to solve the steady-state transport equation (e.g. P_N, S_N, MOC, etc.). The result is that the time-dependent transport problem is reduced to a series of pseudo-steady-state problems at discrete points in time coupled to previous time points through the update of state variables and the treatment of the time derivatives. This approach is used in many time-dependent neutron transport codes, including the three-dimensional whole core neutron transport code DeCART [Joo04, Cho05, Hur08] which was used in this work.

One obstacle with this approach is that it implies that the angular flux should be stored from one or more previous time steps in order to represent the angular flux time derivative. Although the angular flux is the fundamental solution of the transport equation, the scalar flux is often the desired solution in practice. The scalar flux is the integral of the angular flux over all angles in space, and it is required because it is used to calculate reaction rates. Thus in steady-state transport the angular flux is not generally stored; it is instead numerically integrated as it is generated to calculate the scalar flux. Storing the angular flux for large reactor models requires staggering amounts of memory that can exceed the capabilities of even leading-class supercomputers. As a result, time-dependent neutron transport codes are either very limited in the size or resolution of the reactor transients they can model, or they employ low-order angular or spatial approximations to treat the angular flux time derivative.

While the practice of discretizing the transport equation and diffusion equation in time and applying a time integration method has become so common that it is taken for granted, there has been some recent interest in the use of MOC to treat the time derivative as well as the spatial derivatives. This is possible because MOC is a general mathematical technique to rewrite multi-dimensional partial differential equations as ordinary differential equations in one dimension along characteristic curves in the domain. In steady-state neutron transport, these characteristics represent neutron flight paths in space which effectively couple the spatial regions and allow for the solution of the neutron flux distribution. When MOC is applied to the time derivative as well, the neutron flight paths are defined in time as well as space. Analogous space-time approaches have been applied using other transport methods as well.

While there are advantages to this space-time MOC (STC) approach, including enhanced accuracy due to the direct treatment of neutron time-of-flight and preservation of causality, it results in a drastic increase in the computational and memory requirements relative to the conventional approach of time-discretized MOC (TDC). This is because STC requires the evaluation of many discrete characteristics in space-time for every spatial characteristic that occurs in TDC. As a result, STC has not been developed into a practical reactor kinetics method for large, multi-dimensional problems.

However, STC can be made into a practical algorithm for reactor kinetics with limited approximations. In order to solve steady-state MOC (SSC) or TDC, the designer must make some assumptions about the spatial variation of the source term, cross sections, and angular flux time derivative terms (e.g. they are spatially flat or vary linearly within small regions). If similar assumptions are made in the time domain for the STC, the many discrete characteristic equations in space-time can be analytically integrated into a single function. If this function is employed in a similar way to the characteristic equations in TDC, the result is an analytically-integrated space-time MOC

(ASTC) [Hof13a] approach that can reduce the computational and memory burden of STC to a level comparable to SSC while circumventing the angular flux storage issue of TDC. ASTC is one of the new time-dependent MOC methods developed for this thesis.

In the course of analyzing ASTC, we devised an alternate but fully equivalent derivation based on the propagation of the angular flux time derivative along characteristics in space, avoiding the definition of characteristics in space-time required for ASTC. While the space-time characteristic derivation of ASTC provides some qualitative insights, it is cumbersome and circuitous. By contrast, the alternate derivation is straightforward, more general, and provides clearer identification of the leading error terms. These error terms provide both an improved understanding of the limitations of the new method as well as a basis for refinement. In light of the advantages of the alternate derivation, it will be the focus of this thesis.

The alternate derivation introduces a new class of time-dependent MOC methods based on angular flux time derivative propagation (TDP) along characteristics, avoiding the need to store the angular flux for the time derivative. While there may be alternate ways to define the equation for the angular flux time derivative along the characteristic, we focused on a method that effectively recasts the angular flux time derivative in terms of the propagated effects of neutron source derivatives; consequently, we call methods of this type Source Derivative Propagation (SDP). The SDP methods are the focus of this thesis, where we derive, analyze, and test several of these methods with favorable mathematical forms for neutron transport-based nuclear reactor kinetics.

We implemented several SDP methods in the neutron transport code DeCART for numerical testing along with conventional time-dependent MOC methods for reference. We modeled several reactor transients using these methods to assess the performance of SDP under different circumstances. The accuracy and performance of the methods were assessed in light of error analysis, and the numerical results demonstrated the advantages and limitations of SDP. Based on the results of this work, we close by recommending opportunities for further research in this area.

1.1 Outline

The remainder of this chapter is an outline of following chapters. This outline provides a brief summary of the content of each chapter.

Chapter 2. Overview of Neutron Transport and Time Integration Methods

In Chapter 2 we provide an overview of deterministic neutron transport methods that may be applied to the reactor kinetics problem, including MOC. We discuss time-dependent neutron transport computer codes that have used each method and consider the methods' advantages and disadvantages.

In Chapter 2 we also discuss various time integration methods that have been applied to nuclear reactor kinetics. This discussion includes multi-step methods, multi-stage methods, and space-time methods; the latter being the class of methods which employ the same technique to treat the temporal derivative as is employed for the spatial derivative. We also discuss options for approximating the angular dependence of the angular flux time derivative to limit the memory expense of storing angular fluxes.

Finally, we will briefly discuss the treatment of the delayed neutron precursor equations. This includes the direct solution of the equation using a conventional time integration technique and the use of analytical precursor integration.

Chapter 3. Derivation of Steady-State MOC (SSC)

In Chapter 3 we will present a derivation of SSC including a careful examination of each of the approximations to the neutron transport equation. This derivation is typical of MOC applications for reactor physics. Since MOC has been widely used for steady-state transport, the primary purpose of this chapter is to provide background in MOC conventions and establish the nomenclature that will be built upon for time-dependent methods in subsequent chapters.

This derivation also provides an important baseline for reactor transient modeling. Reactor transients generally begin from a steady-state critical configuration, so this method is used to initialize the time-dependent MOC methods. Further, the time-dependent MOC methods described in later chapters are ultimately transient fixed source problems that are solved using a similar algorithm to SSC. The chapter closes with an overview of the SSC algorithm.

Chapter 4. Derivation of Time-Dependent MOC with BDF (BDC)

In Chapter 4 we present a derivation for two conventional time-dependent MOC methods: one which stores the angular flux for the angular flux time derivative and will serve as a reference solution (RBDC), and another which assumes that the angular flux time derivative is isotropic (IBDC), which is a popular and efficient approximation. The BDC methods are similar to previous and recent developments in time-dependent transport for reactor kinetics and represent the state-of-the-art in time-dependent MOC. Both have been implemented in the computer code DeCART and are used for benchmarking the accuracy and efficiency of SDP methods described in Chapter 5.

For both derivations, the backward difference formula (BDF) [But08] is used for time integration. We selected BDF because the first-order version is equivalent to the popular Backward Euler method, and the high-order BDF methods correspond closely with the high-order SDP methods derived in the Chapter 5.

Chapter 5. Derivation of Time-Dependent MOC with SDP

In Chapter 5 we derive several new time-dependent MOC methods that use Source Derivative Propagation (SDP). These methods are the focus of this thesis. The chapter begins with a review of the motivation for this work and briefly explains the relationship between the SDP methods and the ASTC method. The derivation of ASTC is provided in Appendix B.

The first two SDP methods differ in their treatment of the leading error term that arises in the derivation. The first method (TSDP) truncates the error term, while the second method (ISDP) approximates it as isotropic. TSDP is of interest because it is equivalent to ASTC, while ISDP is expected to be more accurate without substantially increasing computational expense.

TSDP and ISDP are derived using the assumption that the macroscopic cross sections are changing slowly. While this assumption is reasonable throughout most of the time and space domain of reactor kinetics, there may be important situations where it is not applicable. The third method we derived (ISCDP) avoids this approximation, which results in the angular flux time derivative being a function of both neutron source derivatives and cross section derivatives. ISCDP is more computationally expensive than TSDP and ISDP, but it is expected to be more accurate when cross sections are changing quickly.

For the SDP methods, we approximate the source and cross section time derivatives to a user-selected order of accuracy using backward differences. Although this is not formally BDF, it is closely related; in Chapter 8 we observe that the solutions for SDP methods with a particular order source approximation correspond closely to RBDC solutions with BDF of the same order and time step.

The SDP methods (TSDP, ISDP, and ISCDP) propagate the angular flux time derivative along characteristics, and they require the approximation of an angular flux second time derivative term (e.g. by truncation or as isotropic). An alternate approach to treat this term is to propagate the second time derivative of the angular flux as well, resulting in a second-derivative SDP (2SDP) method. This concept can be extended to an arbitrarily-high derivative propagation method (NSDP). In light of this, the first-derivative SDP methods explored in this thesis can be understood as the first members of a family of arbitrarily-high derivative methods. While the higher derivative methods are not practical, they are useful for understanding the SDP methods, and thus we derive 2SDP and NSDP in this chapter.

We close Chapter 5 with a summary of the algorithm for the SDP methods, comparing and contrasting these methods with the BDC methods in Chapter 4. Finally, we discuss the implications of the SDP methods for time-dependent transport.

Chapter 6. Error Analysis of the Angular Flux Time Derivative Approximations

In Chapter 6 we provide error analysis for the time-dependent MOC methods used in this thesis (i.e. RBDC, IBDC, TSDP, ISDP, and ISCDP). The purpose of this chapter is to identify the leading error terms for each method and understand the scaling of these terms. Because the BDC and SDP methods differ primarily in their treatment of the angular flux time derivative, this chapter focuses on the error in the approximations for the angular flux time derivative rather than the error in the angular flux itself. These results provide insight into the limitations of the SDP methods.

Chapter 7. Overview of DeCART and Implementation of MOC Methods

The time-dependent MOC methods were implemented in the computer code DeCART. Chapter 7 provides an overview of the neutron transport code DeCART, including an explanation of the important features used in this research. The most important feature is the acceleration of the MOC source solution using coarse mesh finite difference (CMFD) neutron diffusion, which is explained in detail. After the relationship between CMFD and MOC is explained, the chapter provides a detailed description of the MOC algorithm in DeCART as it relates to the steady-state and time-dependent MOC methods derived in the preceding chapters.

Chapter 8. Test Problems and Numerical Results

In Chapter 8 we present three numerical test problems that were used to empirically evaluate the accuracy and efficiency of the SDP methods derived in Chapter 5. The SDP methods were tested by comparison to the BDC methods derived in Chapter 4. RBDC was used to generate the reference solutions, while IBDC was used as a benchmark for efficiency.

The first two transients are based on the TWIGL reactor [Yas65] and are widely used to test new reactor kinetics methods. The first transient is driven by a linear reduction in the thermal absorption cross section, which results in a slow exponential power increase. The second transient is driven by a step change in that cross section, which results in a faster transient.

The third transient is based on the C5G7 benchmark problem [Lew01]. The C5G7 problem includes a more realistic representation of the heterogeneous fuel assemblies, and includes a mix of uranium and mixed oxide fuel. The C5G7 transient is driven by ejecting a control rod drive which results in a fast exponential power increase. The C5G7 transient is more challenging than the TWIGL transients because of the larger spatial gradients and faster power increase.

The accuracy of the SDP methods was evaluated for each transient by comparing the SDP results to a reference solution generated using RBDC with a fine time step. As the time step was reduced and the order of the source derivative approximation increased, the SDP methods converged to the reference solution. We also assessed the order of the convergence by evaluating the error in the solution as a function of time step size.

In addition, the SDP methods were compared to RBDC methods of the same order and time step size to assess whether the propagated angular flux time derivative accurately represented the finite-difference angular flux time derivative used by RBDC. For all three transients, the SDP methods accurately replicated the RBDC solution with the same time step. The IBDC method was also assessed this way, and was found less accurate than the SDP methods for all cases.

Finally, we assessed the efficiency of the SDP methods in comparison to RBDC and IBDC by comparing the run-time and total memory required for the test problems. The SDP methods required more run-time than the IBDC methods but less than the RBDC methods. The memory requirements for SDP were slightly greater than for IBDC, but two orders of magnitude less than for RBDC.

Chapter 9. Summary, Conclusions, and Recommendations for Future Work

In Chapter 9 we provide a summary of the work performed for this thesis and summarize the conclusions identified in the preceding chapters. We also review the limitations and disadvantages of the SDP methods. In light of the results, we suggest opportunities for future research in this area.

Chapter 2

Overview of Neutron Transport and Time Integration Methods

The development of the new time-dependent Method of Characteristics (MOC) formulations in this thesis is best understood in the context of previous and recent experience in time-dependent neutron transport methods, particularly those applied to nuclear reactor kinetics. This chapter will begin with a discussion of the Boltzmann transport equation which is used to understand the behavior of neutrons in nuclear reactors. After this discussion, the chapter will provide an overview of deterministic neutron transport methods applied to solve time-dependent problems.

Neutron transport methods are classified based on how they treat the angular and spatial dependence of the neutron flux. For time-dependent neutron transport, there is some flexibility regarding the treatment of the neutron flux time derivative. In light of this, we provide an overview of time integration methods that may be applied to neutron transport. This discussion will be limited to implicit time integration methods because the time-dependent neutron transport equation is “stiff” and explicit methods require prohibitively small time steps. This section will focus on two classes of time integration methods: linear multi-step methods and multi-stage methods. This section will also provide a brief overview of work in space-time transport methods, which have not yet been used to solve large, 3D reactor kinetics problems. In space-time transport methods, the time derivative is treated using the same technique as the spatial derivatives.

One of the unique challenges introduced by time-dependent neutron transport is the need to store one or more previous angular fluxes to represent the angular flux time derivative. For large reactor problems, storing the angular flux requires excessive memory which can exceed the capabilities of even leading class supercomputers. To avoid this issue, steady-state neutron transport codes for reactor physics will avoid storing the angular flux by numerically integrating the scalar flux as the angular flux is generated. In light of this we provide a brief overview of techniques for addressing this issue.

Finally, the chapter ends with a discussion on the treatment of the delayed neutron precursor equation. This includes both the direct solution of the delayed neutron precursor equation with a time integration method and the solution of the precursor equations using analytic precursor integration.

Before beginning the chapter, we note that while stochastic neutron transport methods can be applied to reactor kinetics, they are generally much more computationally expensive and memory-intensive than deterministic methods. Considering this and the fact that their implementation is substantially different from deterministic methods, a review of time-dependent stochastic transport methods will not be provided in this thesis.

2.1 Neutron Transport Methods

2.1.1 The Boltzmann Transport Equation

Neutron transport methods are used to solve the steady-state or time-dependent Boltzmann transport equation. The time-dependent form of the Boltzmann transport equation can be written as:

$$\begin{aligned}\frac{1}{v(E)}\frac{d\varphi(\mathbf{x},\boldsymbol{\Omega},E,t)}{dt} &\\ &= -\boldsymbol{\Omega}\cdot\boldsymbol{\nabla}\varphi(\mathbf{x},\boldsymbol{\Omega},E,t) - \Sigma_t(\mathbf{x},E,t)\varphi(\mathbf{x},\boldsymbol{\Omega},E,t) \\ &\quad + q(\mathbf{x},\boldsymbol{\Omega},E,t),\end{aligned} \tag{2.1}$$

where the total neutron source is:

$$q(\mathbf{x},\mathbf{\Omega},E,t) = (1-\beta)\frac{\chi_p(\mathbf{x},E)}{4\pi}\int_0^{\infty}\int_{4\pi} \nu\Sigma_f(\mathbf{x},E',t)\varphi(\mathbf{x},\mathbf{\Omega}',E',t)d\mathbf{\Omega}'dE'$$
$$+\int_0^{\infty}\int_{4\pi} \Sigma_s(\mathbf{x},\mathbf{\Omega}'\rightarrow\mathbf{\Omega},E'\rightarrow E,t)\varphi(\mathbf{x},\mathbf{\Omega}',E',t)d\mathbf{\Omega}'dE' \qquad 2.2$$
$$+\frac{S_d(\mathbf{x},E,t)}{4\pi},$$

and where S_d is the delayed neutron source which is discussed in detail in a later section.

The solution of the neutron transport equation is the neutron angular flux (or fluence rate) $\varphi(\mathbf{x},\mathbf{\Omega},E,t)$, which is the neutron path-length-rate (or total path length traveled in unit time) in a unit volume around position $\mathbf{x}$ in unit solid angle around $\mathbf{\Omega}$ in unit energy interval around E at time t. Although the angular flux is the fundamental solution of the transport equation, we are often more interested in the neutron scalar flux $\phi(\mathbf{x},E,t)$, which is the angular flux integrated over all angles:

$$\phi(\mathbf{x},E,t) = \int_{4\pi} \varphi(\mathbf{x},\mathbf{\Omega},E,t)d\mathbf{\Omega}. \qquad 2.3$$

Predicting the neutron flux distribution within a reactor is one of the major concerns of nuclear reactor physics because the neutron flux is used with the macroscopic cross sections to determine nuclear reaction rates of interest. These reaction rates provide crucial information about the reactor such as where heat is generated, what actinides are transmuted, and how structural materials are damaged.

The macroscopic neutron cross sections (e.g. Σ_t, Σ_f, or Σ_s) are material properties which represent the probability per unit path length that a neutron within the material will interact with a nucleus with a specific nuclear reaction. For example, the fission cross section corresponds to fission events, the scattering cross section to scattering events, and the total cross section to all events. The cross sections are generally assumed a known quantity in nuclear reactor physics[1].

When fission occurs, a small number neutrons are released. For the transport equation we use the mean number of neutron emitted per fission ν. Some fission neutrons are promptly emitted, while others are delayed because they are produced through the decay of fission products. We typically define a delayed neutron fraction β which is the fraction the total fission neutrons that are delayed. The prompt neutrons and delayed neutrons are emitted with a different distribution of energies (χ_p and χ_d respectively). While these parameters are not formally cross sections, they are material dependent and often handled like cross sections in neutron transport computer codes. The fission spectra are also considered a known quantity.

The time-dependent neutron transport equation is also a function of the neutron velocity v which is an energy-dependent scalar quantity in this formulation. While the Latin character "v" looks similar to the Greek character "nu" (i.e. the mean number of neutrons emitted per fission), in this thesis the neutron velocity will only occur in equations in the denominator of time derivative terms (e.g. $\frac{1}{v}\frac{d\varphi}{dt}$) while the mean number of neutrons emitted per fission will exclusively appear in the numerator of the fission source (i.e. $\nu\Sigma_f\varphi$).

The total neutron source q includes neutrons from fission and scattering events, and delayed neutrons. The delayed neutron source S_d will be discussed in detail later in this chapter. Because we use an analytic precursor integration method in this work, this term is assumed to be known in Equation 2.2.

The time-dependent transport equation can be rewritten as the steady-state transport equation with minor modifications. First, the angular flux time derivative is by definition zero. Second, the delayed neutrons can be combined with the prompt

[1] Formally, the neutron cross sections are a function of the temperature distribution of materials which may not be known because the temperature distribution is an indirect function of the neutron flux. In this case, an initial guess for the cross sections is employed and they are updated iteratively until the solution converges.

neutrons because the delayed neutrons will be in equilibrium. Finally, the neutron fission source is divided by the eigenvalue k_{eff} to ensure a critical system.

Techniques for solving the neutron transport equation are typically classified based on their treatment of the angular variables and in some cases the spatial variable. In the following sections we review several common techniques for solving the neutron transport equation.

2.1.2 Spherical Harmonics (P$_N$)

In the spherical harmonics method (P$_N$) the angular variable is expanded in terms of spherical harmonics functions, which are in turn described by Legendre polynomials. The series of spherical harmonic functions is infinite, and in practice it is necessary to truncate the series after a finite number of terms; if the series is truncated after N+1 terms, the approximation is referred to as a P$_N$ method. The P$_N$ method does not specify how the spatial derivatives are treated, but finite difference approximations are common.

The spherical harmonics functions are given by:

$$Y_n^m(\mathbf{\Omega}) = \sqrt{\frac{2n+1}{4\pi}\frac{(n-|m|)!}{(n+|m|)!}}\, P_n^{|m|}(\mu)e^{im\alpha}, \qquad 0 \le n \le |m| \le N, \tag{2.4}$$

where the associated Legendre functions are:

$$P_n^m(\mu) = (1-\mu^2)^{\frac{m}{2}}\frac{d^m}{d\mu^m}P_n(\mu), \qquad 0 \le n \le m \le N, \tag{2.5}$$

with the Legendre polynomials given for $n > 0$:

$$P_n(\mu) = \frac{1}{2^n n!}\frac{d^n}{d\mu^n}[(\mu^2-1)^n], \qquad 0 \le n \le N, \tag{2.6}$$

and where the angular variable $\mathbf{\Omega}$ for polar angle is θ and azimuthal angle α is:

$$\mathbf{\Omega} = \sqrt{1-\mu^2}\cos\alpha\,\mathbf{i} + \sqrt{1-\mu^2}\sin\alpha\,\mathbf{j} + \mu\mathbf{k}, \tag{2.7}$$

and $\mu = \cos\theta$.

For P_N methods, the angular flux is approximated using a truncated expansion in terms of the spherical harmonics functions:

$$\varphi(\mathbf{x},\mathbf{\Omega},E,t) \approx \sum_{n=0}^{N}\sum_{m=-n}^{n} \varphi_n^m(\mathbf{x},E,t)\,Y_n^m(\mathbf{\Omega}), \qquad 2.8$$

where φ_n^m are the n^{th} moments of the neutron flux, and thus the angular flux is expanded into $(N+1)^2$ terms.

To derive the $(N+1)^2$ equations for P_N, we multiply Equation 2.1 by each complex conjugate of the spherical harmonics functions (i.e. $Y_n^m(\mathbf{\Omega})$ where $0 \leq |m| \leq n \leq N$). Then we integrate the resulting equation over all directions in the unit sphere. Finally, we apply Equation 2.8 to each of the $(N+1)^2$ equations.

This results in a complex system of first-order partial differential equations. As N goes to ∞, the P_N solution converges to the neutron transport solution. However, it is impractical to solve the P_N equations for high N because as N increases, the number of unknowns grows quadratically. As a result, high-order P_N methods are not as widely used in reactor physics as the discrete ordinates methods described later in this chapter.

On the other hand, low-order P_N methods are widespread in reactor physics, especially P_1 methods. These methods are attractive because of their simple structure and low solution cost. One convenient feature of P_1 methods is that the zeroth and first moments of the angular flux are the neutron scalar flux and the neutron currents, which are useful quantities. However, for some problems the P_1 method is not sufficiently accurate. This is also true of the diffusion method and simplified spherical harmonics methods which are described briefly later in this section.

P_N methods may be adapted to solve time-dependent neutron transport problems. Simple time integration techniques like backward Euler have been preferred for transient P_N methods because of the cost and complexity of the transport method. The work of McClarren et al. provides recent examples of time-dependent P_N methods [McC07a, McC07b]. These methods used Backward Euler or a second-order semi-implicit Runge-Kutta method to treat the time derivative. The papers explored the use of Riemann solvers for time-dependent P_N, building on previous work in time-dependent P_N methods by Brunner and Holloway [Bru05].

2.1.2.1 Diffusion and Simplified Spherical Harmonics (SP_N)

Neutron diffusion theory has long been the workhorse of nuclear reactor physics as well as nuclear reactor kinetics [Sut96]. Diffusion theory can be derived from the P_1 equations with an approximation called Fick's Law. Here the neutron current $\boldsymbol{J}$ (the first moment of the angular flux) is approximated using the gradient of neutron scalar flux ϕ (the zeroth moment of the angular flux):

$$\boldsymbol{J}(\mathbf{x}, E, t) \approx -\frac{1}{3\Sigma_{tr}(\mathbf{x}, E, t)} \boldsymbol{\nabla}\phi(\mathbf{x}, E, t). \qquad 2.9$$

This permits the elimination of the equations for the first moment of the angular flux, resulting in equations only in terms of the scalar flux. Although diffusion theory is formally an approximation to the P_1 equations, it is often distinguished from neutron transport methods because it is inaccurate for problems involving strong absorbers or streaming pathways. However, for many problems in reactor physics, diffusion theory is sufficiently accurate.

Simplified spherical harmonics (SP_N) is also an approximation to the P_N method. While the formal derivation is more complex, the equations can be "derived" by observing the relationship between the diffusion operator in the 1D and 3D P_1 equations (i.e. $\frac{d}{dx}\frac{1}{3\Sigma_{tr}}\frac{d}{dx}$ vs. $\boldsymbol{\nabla}\frac{1}{3\Sigma_{tr}}\boldsymbol{\nabla}$) and applying an analogous replacement to the diffusion operator in the 1D P_N equations which results in the 3D SP_N equations. In 1D geometries, SP_N and P_N are equivalent. Although early derivations were suspect [Gel61, Smi97], the SP_N method has since been shown to have a valid mathematical foundation.

The SP_N method can be understood as a "super diffusion method." It has a similar mathematical structure, and for problems where diffusion is accurate, SP_N methods are more accurate. However, like diffusion and P_1 methods, the SP_N methods may be inaccurate for problems with strong absorbers and streaming.

Nonetheless, diffusion and SP$_N$ methods have been extensively used for nuclear reactor kinetics. A wide range of time integration techniques and approaches to addressing the delayed neutron precursor equations have been applied to these methods. As a result, much of the literature on time-dependent diffusion and SP$_N$ methods is applicable to time-dependent transport methods. Sutton and Aviles [Sut96] provided an excellent overview of contemporary research in time-dependent neutron diffusion for nuclear reactor kinetics.

2.1.3 Discrete Ordinates (S$_N$)

In the discrete ordinates method (S$_N$), the angular domain is discretized into a finite number of directions M. Each of these directions Ω_m has an angular weight w_m which corresponds to the surface area on the unit sphere represented by that direction. For rectangular Cartesian geometries, neutrons travel in these directions only; in effect, neutrons that would travel in other directions are forced to travel along the discrete ordinates, a process which is accounted for with the angular weights. The S$_N$ method does not specify the treatment of the spatial derivatives, and finite difference approximations are common.

A set of discrete ordinates and angular weights is called an angular quadrature set. The definition of an angular quadrature set is important to the accuracy of the S$_N$ method. For 1D geometries $M = N$, but for higher dimensionality $M > N$.

After the applying the S$_N$ approximation, the transport equation is replaced by M equations for each unique direction $\mathbf{\Omega}_m$:

$$\begin{aligned}\frac{1}{v(E)}\frac{d\varphi_m(\mathbf{x}, E, t)}{dt} \\ &= -\mathbf{\Omega}_m \cdot \mathbf{\nabla}\varphi_m(\mathbf{x}, E, t) - \Sigma_t(\mathbf{x}, E, t)\varphi_m(\mathbf{x}, E, t) \\ &+ q_m(\mathbf{x}, E, t),\end{aligned} \tag{2.10}$$

where the directions are coupled through the source term, which includes scattering and fission.

The S$_N$ equations are typically much easier to implement and solve than P$_N$ equations of comparable accuracy. As the number of angles increases, the S$_N$ solution converges to the transport solution. However, low-order S_N methods may be inaccurate in problems with optically-thin streaming paths.

S_N methods have been extended to treat time-dependent transport problems. Because of the computational expense of the S_N equations, simple time integration methods such as Backward Euler have been preferred. In an early example, Goluoglu and Dodds used the 3D S_N computer code TORT to calculate the shape function for a time-dependent quasistatic-based reactor kinetics method [Gol01]. This method employed Backward Euler to approximate time derivatives.

Later in an unrelated work, Pautz and Birkhofer incorporated time dependence into the 2D S_N computer code DORT [Pau03] and later TORT [Seu11]. Both implementations used Backward Euler to treat the time derivatives. In recognition of the large memory requirements of storing the previous angular flux to represent the angular flux time derivative, Pautz and Birkhofer explored the option of approximating the previous angular flux using a low-order spherical harmonics expansion. However, they observed that this method was insufficiently accurate and concluded that all the angular fluxes needed to be stored [Pau03].

The Los Alamos S_N computer code PARTISN [Alc11] can model transient problems. PARTISN uses the Theta method to treat the angular flux time derivative.

A transient method has been incorporated into the S_N computer code Denovo by Banfield et. al. [Ban12]. This implementation also uses Backward Euler to treat the angular flux time derivative. However, because of the considerable memory requirements of storing the angular flux, Denovo approximates the previous angular flux in the time derivative using the scalar flux. Banfield et. al. justify this approximation by comparison to the time-dependent MOC computer code DeCART [Cho05], but DeCART approximates *both* angular fluxes in the time derivative using the scalar flux, not just the previous angular flux. Based on our previous work, we concluded that this is an important distinction [Hof13b].

Also, there has been limited use of S_N methods for space-time transport. This is discussed in the time integration section of this chapter.

2.1.4 Collision Probability Method (CPM)

The Collision Probability Method (CPM) is an integral neutron transport method. Unlike S$_N$ and P$_N$ methods, CPM solves the integral form of the neutron transport equation rather than the differential form. The formal derivation of CPM with a realistic treatment of spatial heterogeneities is complex and beyond the scope of this chapter. However, the integral transport equation can be solved succinctly with some approximations, and this solution provides some insight into the application of CPM. Integrating the transport equation over all angles gives the integral transport equation:

$$\begin{aligned}&\int_{4\pi} \frac{1}{v(E)} \frac{d\varphi(\mathbf{x},\mathbf{\Omega},E,t)}{dt} d\mathbf{\Omega} \\ &\qquad = \int_{4\pi} -\mathbf{\Omega}\cdot\nabla\varphi(\mathbf{x},\mathbf{\Omega},E,t) - \Sigma_t(\mathbf{x},E,t)\varphi(\mathbf{x},\mathbf{\Omega},E,t) \\ &\qquad + q(\mathbf{x},\mathbf{\Omega},E,t)\, d\mathbf{\Omega}.\end{aligned} \tag{2.11}$$

If the neutron source and angular flux time derivative are both assumed to be isotropic, they can be written in terms of an isotropic transient source:

$$\int_{4\pi} \mathbf{\Omega}\cdot\nabla\varphi(\mathbf{x},\mathbf{\Omega},E,t) + \Sigma_t(\mathbf{x},E,t)\varphi(\mathbf{x},\mathbf{\Omega},E,t)\, d\mathbf{\Omega} = \hat{q}(\mathbf{x},E,t), \tag{2.12}$$

where the isotropic transient source is defined in terms of the neutron scalar flux:

$$\begin{aligned}\hat{q}(\mathbf{x},E,t) &= \int_{4\pi} q(\mathbf{x},\mathbf{\Omega},E,t) - \frac{1}{v(E)}\frac{d\varphi(\mathbf{x},\mathbf{\Omega},E,t)}{dt} d\mathbf{\Omega} \\ &= (1-\beta)\frac{\chi_p(\mathbf{x},E)}{4\pi}\int_0^\infty \nu\Sigma_f(\mathbf{x},E',t)\phi(\mathbf{x},E',t)dE' \\ &+ \frac{1}{4\pi}\int_0^\infty \Sigma_s(\mathbf{x},E'\to E,t)\phi(\mathbf{x},E',t)dE' + \frac{S_d(\mathbf{x},E,t)}{4\pi} \\ &- \frac{1}{4\pi v(E)}\frac{d\phi(\mathbf{x},E,t)}{dt}.\end{aligned} \tag{2.13}$$

If we also assume that the total cross section is spatially invariant near $\mathbf{x}$ we can solve Equation 2.12 for the scalar flux[2]:

$$\phi(\mathbf{x}, E, t) = \int\limits_{\mathbf{x}'} \frac{e^{-\Sigma_t(\mathbf{x},E,t)|\mathbf{x}-\mathbf{x}'|}}{|\mathbf{x}-\mathbf{x}'|^2} \hat{q}(\mathbf{x}', E, t) d\mathbf{x}'. \qquad 2.14$$

This is the solution to the integral transport equation with isotropic sources and spatially-invariant cross sections. Thus the spatially-dependent scalar flux at any given position is a function of the neutron source at every other position in the problem. A similar equation is derived in CPM. In practice it is necessary to limit the effective range of sources to some number of neutron mean-free-paths.

Unlike P_N and S_N, CPM methods imply the use of certain types of spatial discretizations. In CPM, the problem is customarily divided into “flat source” regions where the scalar flux, neutron source, and cross sections are assumed to be spatially invariant. Sets of parallel and equally-spaced “rays” are defined over the problem geometry at various angles; these angles are analogous to discrete ordinates. Equation 2.14 is used to define “collision probabilities”, which are the likelihood that a neutron from a source in a particular flat source region will have its first collision within another region along a ray.

The collision probabilities are used to construct a system of linear equations for the scalar flux in each region in terms of the neutron source in nearby regions (where the distance is limited to some number of mean-free-paths). Since the neutron source is a function of the scalar flux, the customary approach is to use some initial guess for the scalar flux and iteratively update the source with each new scalar flux solution until the solutions have converged.

One of the major advantages of CPM is that is allows for complex geometries (e.g. cylinders) without homogenization; this is in contrast to most P_N and S_N

[2] CPM methods typically define fine material regions where the cross section is spatially invariant, but this introduces complexities to the derivation that are beyond the scope of this chapter.

implementations. This simply requires a sophisticated geometric treatment for determining the intercepts of the rays with the boundaries of spatial regions.

However, CPM has some significant disadvantages. First, it requires the assumption that the neutron source is isotropic. This is a reasonable approximation for the fission source, which is relatively isotropic, and it may be an acceptable approximation for the scattering source if transport-corrected scattering is employed. However, for time-dependent transport, approximating the angular flux time derivative as isotropic may be too inaccurate.

Another disadvantage of CPM is that it requires forming large systems of equations which requires large allocations of memory. As a result of these shortcomings, while CPM has been widely used for small, assembly-level steady-state neutron transport [Lou99, Smi00, Jon00], it may not be practical for time-dependent neutron transport for large problems. While we are not aware of any time-dependent CPM methods that employ conventional time integration techniques (e.g. Backward Euler), CPM has been used for 1D space-time neutron transport; this work is discussed in Section 2.2.5 .

2.1.5 Method of Characteristics (MOC)

Method of Characteristics (MOC) is a neutron transport method that has similarities to S_N and CPM. Like S_N, MOC treats the angular dependence of the neutron flux by defining discrete ordinates along which neutrons travel. Like CPM, MOC allows for complex geometries and defines rays (which are called “characteristics” in MOC) along which the neutron flux is propagated. Thus MOC implies some details about both the angular treatment and spatial discretization.

MOC uses a general mathematical technique called method of characteristics which is used to rewrite a multi-dimensional partial differential equation (PDE) as a 1D ordinary differential equation (ODE) along a characteristic curve[3]. The resulting ODE is often easier to solve than the original PDE. Appendix A provides a general derivation of

[3] In neutron transport for rectangular Cartesian geometries, these “characteristic curves” are straight lines in space.

MOC. The MOC solution along the characteristic may either be used alone to provide insight into the nature of the PDE or be combined with solutions along other characteristics to form an approximation to the solution of the PDE.

For the time-dependent 3D neutron transport equation, we use MOC to rewrite the spatial derivatives as a single spatial derivative along a characteristic:

$$\mathbf{\Omega} \cdot \mathbf{\nabla} \varphi(\mathbf{x}, \mathbf{\Omega}, E, t) \rightarrow \frac{d}{ds} \varphi_k(s, \mathbf{\Omega}_m, E, t), \tag{2.15}$$

where k is the index of a characteristic which is uniquely specified by a discrete ordinate $\mathbf{\Omega}_m$ and an arbitrary starting position $\mathbf{x_0}$ while s is the spatial coordinate along that characteristic which is related to the starting position by $\mathbf{x} = \mathbf{x_0} + \mathbf{\Omega}_m s$.

By applying Equation 2.15 to Equation 2.1, the neutron transport equation is rewritten as a characteristic equation:

$$\begin{aligned} &\frac{1}{v(E)} \frac{d\varphi_k(s, \mathbf{\Omega}_m, E, t)}{dt} \\ &\quad = -\frac{d\varphi_k(s, \mathbf{\Omega}_m, E, t)}{ds} - \Sigma_t(s, E, t) \varphi_k(s, \mathbf{\Omega}_m, E, t) \\ &\quad + q_k(s, \mathbf{\Omega}_m, E, t). \end{aligned} \tag{2.16}$$

This is the equation for the angular flux along a characteristic. As with CPM, the geometry is customarily discretized into fine regions where the spatial dependence of the cross sections and neutron source may be approximated. We also define "segments" as the portion of a characteristic within a region. If the source and cross section are assumed to be spatially-invariant within regions and the angular flux time derivative is assumed to be spatially-flat along each segment, the characteristic equation can be solved using the integrating factor:

$$
\begin{aligned}
&\varphi_k(s, \mathbf{\Omega}_m, E, t) \\
&\qquad = \varphi_k\left(s_0^{r,k}, \mathbf{\Omega}_m, E, t\right) e^{-\Sigma_t^r(E,t)(s-s_0)} \\
&\qquad + \left(q_k^r(\mathbf{\Omega}_m, E, t) - \frac{1}{v(E)} \frac{d\varphi_k^r(\mathbf{\Omega}_m, E, t)}{dt}\right) \\
&\qquad \times \left(\frac{1 - e^{-\Sigma_t^r(E,t)(s-s_0)}}{\Sigma_t^r(E,t)}\right),
\end{aligned}
\qquad 2.17
$$

where r is the index of the region that corresponds to the position s along characteristic k, and $s_0^{r,k}$ is the position where the characteristic enters the region.

Equation 2.17 is used to propagate the angular flux along the characteristic and to calculate the segment-wise average angular flux. The latter is used for all of the segments within a region to numerically integrate the scalar fluxes by region.

While MOC could be used to form a linear system of equations which was solved in a manner similar to CPM, instead the angular flux is propagated along the characteristics sequentially using a sweeping routine. This provides some advantages in terms of parallelizability and avoids the need to create and solve a large system of equations. Because of these advantages, MOC has been widely used for many years for steady-state neutron transport for small, assembly-level problems [Hal80, Jon00, Smi00]. More recently, due to increases in computing power MOC has been used for large, steady-state reactor-level problems as well [Joo04, Col13]. There have also been several MOC computer codes with transient capability [Cho05, Hur08, Tay09, Tsu13, Tal13].

The primary disadvantage in using MOC for time-dependent transport is the necessity of storing angular fluxes for each segment to represent the angular flux time derivative. Since there are many segments in a large problem, this can entail prohibitively large memory requirements. A common resolution to these issues has been to assume that the angular flux time derivative is isotropic and approximate it using the scalar flux [Cho05, Hur08, Tsu13, Tal13]. While this approximation is computationally efficient, it may be inaccurate for problems where the angular distribution of the flux is changing quickly. We, among others, have empirically investigated the accuracy of this approximation [Tsu13, Tal13, Hof13b]. The results suggest that this approximation is reasonably accurate for small problems, but we have been unable to test large problems due to the high memory requirements of storing the angular flux for the reference solution. Other researchers have simply stored the angular flux and thus limited their practical problem size [Tay09].

In this thesis we describe a new class of time-dependent MOC formulations which avoids the need to make this approximation. In this class of methods, an equation for the angular flux time derivative along the characteristic is defined, which is used to propagate the angular flux time derivative along the characteristic in addition to the angular flux; hence the class is called MOC with Angular Flux Time Derivative Propagation (TDP). This thesis focuses on a subset of these methods which effectively recasts the angular flux time derivative in terms of the propagated effects of source derivatives and is thus called the Source Derivative Propagation (SDP) method. We originally developed an equivalent but less general SDP method based on an approximate space-time MOC method [Hof13a].

We assessed the accuracy of the SDP methods by comparison to a conventional time-dependent MOC method that approximates the angular flux time derivative locally using a backward difference approximation. We refer to this approach as backward difference time-dependent MOC (BDC). We employed two difference BDC methods: a reference BDC method that stored the angular flux for the time derivative (RBDC), and an efficient BDC method assumes that the angular flux time derivative is isotropic (IBDC). When a first-order backward difference approximation is used, RBDC is equivalent to Talamo's Method II [Tal13], while IBDC is equivalent to the isotropic approximations used for DeCART's original transient method [Cho05], Tsujita's isotropic method [Tsu13], and Talamo's Method I [Tal13].

Several researchers have investigated the use of space-time MOC for low dimensionality space-time transport, and this is described in detail in Section 2.2.5 .

2.1.5.1 Method of Characteristic Direction Probability (CDP)

The MOC equations can be used to form a linear system of equations for the angular fluxes in a manner similar to CPM; this is called the Method of Characteristic Direction Probabilities (CDP) [Hon99, Liu13]. This method is the subject of current research because CDP provides some advantages relative to MOC. In particular, solving the system of linear equations for the angular fluxes in CDP is often faster than evaluating the angular fluxes sequentially using MOC sweeps. However, when the computational expense of setting up the system of equations is included, CDP may be slower. To our knowledge, no one has implemented a time-dependent CDP method, but it would face the same obstacles as conventional time-dependent MOC. The SDP methods described in this thesis could be applied to CDP as well.

2.2 Implicit Time Integration Methods

In this section we will provide a brief overview of implicit time-integration methods that have been used for nuclear reactor kinetics. Because time-dependent neutron transport has only recently been applied to nuclear reactor kinetics, some of the following methods have only been applied to time-dependent diffusion. However, all methods in this section are applicable to time-dependent neutron transport methods in general.

For illustration, each of the methods in this section will be applied to the following basic ODE:

$$\frac{dy(t)}{dt} = f[y(t), t]. \tag{2.18}$$

The time integration methods described in this section are implicit, in contrast to explicit methods. The practical distinction between explicit and implicit methods is that for explicit methods the function in Equation 2.18 $f(y, t)$ is evaluated at a previous point in time (i.e. $t < t_n$), while for implicit methods the function is evaluated at the present time (i.e. $t = t_n$). As a result, for explicit methods it is generally possible to algebraically isolate the solution variable $y(t_n)$. For implicit methods this is not possible, and it is typically necessary to solve a linear equation.

Although implicit methods have a higher computational cost than explicit methods, they are preferable for solving stiff differential equations like the time-dependent neutron transport equation. Stiffness occurs when differential equations feature multiple time-dependent phenomena with very different time scales (e.g. the inverse neutron velocity and the delayed neutron precursor decay constants in the time-dependent neutron transport equation). Explicit methods are often unstable for stiff equations except when we employ unacceptably small time steps which resolve the faster time scale. Implicit methods may be stable for much larger time steps and are thus more computationally efficient.

With the exception of the space-time transport methods discussed at the end of this section, the time integration methods in this chapter can be understood as either linear multi-step methods, multi-stage methods, or both. Figure 2-1 provides a graphical representation of the relationship between the implicit time integration methods described in this section. The dashed lines indicate methods that are not discussed in this chapter, including general linear methods (GLM), which is the unification of multistep and multistage methods. The textbook "Numerical Methods for Ordinary Differential Equations" by J. C. Butcher provides thorough examination of GLM [But08].

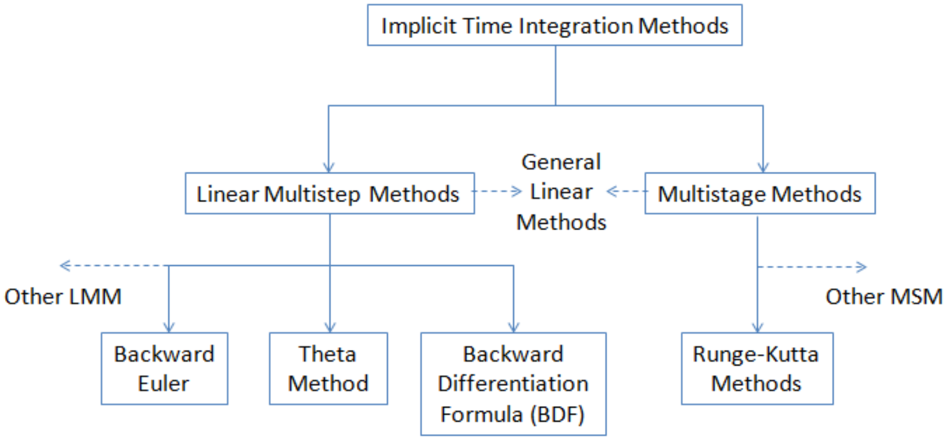


Figure 2-1. The implicit time integration methods described in this chapter

Linear multi-step methods approximate Equation 2.18 using a linear combination of the solution variable y and the function f evaluated at one or more previous points in time. In other words, they approximate Equation 2.18 with an equation of the form:

$$\sum_{i=0}^{I} A_i y_{n-i} = \sum_{j=0}^{J} B_j f\left(y_{n-j}, t_{n-j}\right), \qquad 2.19$$

where t_n is the time at which we are solving the equation and for succinctness we define $y_n \equiv y(t_n)$. As a result, multi-step methods seek to achieve high-order representation of the ODE by storing additional data from previous points in time.

By contrast, multi-stage methods achieve high-order representation of the ODE by evaluating the equation at points in time that are intermediate to the current and previous solution, and then discarding that data when the solution is determined. Runge-Kutta methods are a major class of multi-stage methods.

An *S*-stage Runge-Kutta method produces a solution to the differential equation using:

$$y_n = y_{n-1} + \frac{1}{\Delta t_n} \sum_{i=0}^{S} B_i k_i, \qquad 2.20$$

where the time step size $\Delta t_n \equiv t_n - t_{n-1}$, and k_i are coefficients calculated for the Runge-Kutta method using:

$$k_i = \Delta t_n f\left(y_n + \sum_{j=1}^{S} A_{i,j}\, k_j\, , t_{n-1} + C_i \Delta t_n \right), \qquad 2.21$$

and where the coefficients $A_{i,j}$, B_i, and C_i are specified in a tableau (e.g. Table 2-1).

Table 2-1. A tableau for an arbitrary Runge-Kutta method

C_1	$A_{1,1}$	$A_{1,2}$	$\cdots$	$A_{1,S}$
C_2	$A_{2,1}$	$A_{2,2}$	$\cdots$	$A_{2,S}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
C_S	$A_{S,1}$	$A_{S,2}$	$A_{S,1}$	$A_{S,S}$
	B_1	B_2	$\cdots$	B_S

These coefficients are unique to each Runge-Kutta method and are generated as needed. Depending on which and how many coefficients are defined in the tableau, one or more equations (i.e. Equation 2.21) will be solved sequentially or simultaneously to model one time step with a Runge-Kutta method. Thus linear multi-step methods generally require more memory while multi-stage methods require greater computational expense.

2.2.1 Backward Euler Method

The Backward Euler method (or the implicit Euler method) is a first-order, implicit time integration method. It is one of the most basic time integration methods, and it is widely used because it is easy to implement and has good stability properties. It provides a natural starting point for new reactor kinetics methods.

For the Backward Euler method, the time-dependent equation is evaluated at time t_n and the time derivative is approximated using a backward difference:

$$\frac{y_n - y_{n-1}}{\Delta t_n} = f(y_n, t_n). \qquad 2.22$$

To understand why the Backward Euler method is first-order accurate in time, consider the backward Taylor series expansion of $y(t)$ near t_n:

$$y_{n-1} = y_n - \Delta t_n \frac{dy_n}{dt} + \frac{\Delta t_n^2}{2!}\frac{d^2 y_n}{dt^2} - \frac{\Delta t_n^3}{3!}\frac{d^3 y_n}{dt^3} + \cdots. \qquad 2.23$$

We can algebraically isolate $\frac{dy_n}{dt}$:

$$\frac{dy_n}{dt} = \frac{y_n - y_{n-1}}{\Delta t_n} + \frac{\Delta t_n}{2!}\frac{d^2 y_n}{dt^2} - \frac{\Delta t_n^2}{3!}\frac{d^3 y_n}{dt^3} + \cdots. \qquad 2.24$$

If we apply Equation 2.24 to Equation 2.18, we recover the Backward Euler method with a truncated error term:

$$\frac{y_n - y_{n-1}}{\Delta t_n} = f(y_n, t_n) + \left[-\frac{\Delta t_n}{2!}\frac{d^2 y_n}{dt^2} + \frac{\Delta t_n^2}{3!}\frac{d^3 y_n}{dt^3} - \cdots\right], \qquad 2.25$$

where the truncated error term is in brackets. The leading error term scales with the time step size, and so Backward Euler is said to be first-order accurate in time.

Many high-order, implicit time integration methods are closely related to the Backward Euler method because either (a) with the correct selection of a parameter the high-order method becomes equivalent to Backward Euler, or (b) Backward Euler is the first-order version of the high-order method. In either case, with the correct input selection, a high-order method can reproduce the results of Backward Euler.

Many time-dependent neutron diffusion and neutron transport codes use the Backward Euler method [Gel01, Pau03, McC07a, Tay09, Ban12, Tsu13, Tal13].

2.2.2 Theta Method

The Theta method is a variable-order, implicit, linear multi-step method. In the Theta method, either the user or a computer code determines a theta parameter which can range from zero to one:

$$\frac{y_n - y_{n-1}}{\Delta t_n} = \theta f(y_n, t_n) + (1-\theta) f(y_{n-1}, t_{n-1}). \qquad 2.26$$

There are several special cases for theta. When theta is one, the Theta method is identical to Backward Euler. When theta is zero, the result is identical to the Forward Euler method (i.e. the Euler method or the Explicit Euler method). When theta is one-half, the result is identical to the second-order accurate Trapezoidal Rule:

$$\frac{y_n - y_{n-1}}{\Delta t_n} = \frac{1}{2}\big(f(y_n, t_n) + f(y_{n-1}, t_{n-1})\big). \qquad 2.27$$

We can also illustrate why the Trapezoidal Rule is second-order accurate in time using the Taylor series expansion. We begin with Equation 2.25, but we approximate the second derivative using a backward difference:

$$\frac{d^2y_n}{dt^2} = \frac{\frac{dy_n}{dt} - \frac{dy_{n-1}}{dt}}{\Delta t_n} = \frac{f(y_n, t_n) - f(y_{n-1}, t_{n-1})}{\Delta t_n}. \qquad 2.28$$

When this is applied to the second derivative in Equation 2.25, we recover the Trapezoidal Rule:

$$\frac{y_n - y_{n-1}}{\Delta t_n} = \frac{1}{2}\big(f(y_n, t_n) + f(y_{n-1}, t_{n-1})\big) + \left[+\frac{\Delta t_n^2}{3!}\frac{d^3y_n}{dt^3} - \cdots\right], \qquad 2.29$$

where now the leading error term is proportional to the square of the time step and thus the method is said to be second-order accurate in time[4].

Compared to the Backward Euler method, the Theta method requires that one additional quantity is stored from the previous point in time (i.e. $f(y_{n-1}, t_{n-1})$). This is typical of linear multi-step methods, where higher-order accuracy is achieved by retaining additional data from previous points in time.

The Theta method has been used in several neutron diffusion and transport computer codes [Cho05, Alc11]. While the implementation of the Theta method is straightforward, explicitly calculating and storing the function $f(y_n, t_n)$ may be cumbersome because some methods do not explicitly construct this term. Further, this term is angularly-dependent, and storing this value further increases the memory requirements.

2.2.3 Backward Differentiation Formula (BDF)

The Backward Differentiation Formula (BDF) or Gear's method is a variable-order, implicit linear multi-step method. In BDF, the time derivative is approximated as a linear function of the solution at one or more previous points in time:

[4] If we repeat this process and use backward differences to approximate the other higher derivatives (e.g. $\frac{d^3y_n}{dt^3}$, etc.), we derive the Implicit Adams methods. Backward Euler and the Trapezoidal Rule are equivalent to the first- and second-order accurate Implicit Adams methods. The Implicit Adams methods were not considered for this work because they are not stable for stiff equations beyond second order.

$$\sum_{i=0}^{O_{\mathrm{BDF}}} A_i y_{n-i} = f(y_n, t_n), \qquad 2.30$$

where O_{BDF} is the order of the BDF method and A is the coefficient for each solution from the current and previous points in time.

The coefficients for BDF are derived using the backward Taylor series expansion in Equation 2.23. As for the Trapezoidal Rule, the high-order derivatives are approximated using backward difference; unlike the Trapezoidal Rule, the backward differences are repeated until the high-order derivative is approximated in terms of the solution variable rather than the first derivative. For example, for a constant time step Δt the second derivative is approximated as:

$$\frac{d^2 y_n}{dt^2} = \frac{\frac{dy_n}{dt} - \frac{dy_{n-1}}{dt}}{\Delta t} = \frac{\frac{y_n - y_{n-1}}{\Delta t} - \frac{y_{n-1} - y_{n-2}}{\Delta t}}{\Delta t} = \frac{y_n - 2y_{n-1} + y_{n-2}}{\Delta t^2}. \qquad 2.31$$

By truncating the Taylor series expansion to the desired order of accuracy, the corresponding BDF method is derived. For example, for second-order BDF (BDF2) with a constant time step we apply Equation 2.31 to Equation 2.25:

$$\frac{\frac{3}{2}y_n - 2y_{n-1} + \frac{1}{2}y_{n-2}}{\Delta t} = f(y_n, t_n) - \left[+\frac{\Delta t^2}{3!}\frac{d^3 y_n}{dt^3} - \cdots\right], \qquad 2.32$$

where the error term is again in brackets and the leading error is second-order. Higher-order methods are derived in a similar fashion. The coefficients are dependent on the previous time step sizes.

BDF methods up to sixth-order are potentially stable for stiff equations, but methods beyond sixth-order are unstable. The first-order BDF method (BDF1) is equivalent to Backward Euler.

Like the Theta method, BDF achieves high-order representation of the time derivative by storing additional data from previous points in time. However, whereas the Theta method stores the first derivative of the solution from the previous time step, BDF stores the solution from many previous points in time. The memory requirements for BDF and the Theta method are equal for the same order of accuracy[5], but BDF can be used to achieve higher accuracy methods. Also, BDF methods are generally less computationally expensive than the Theta method because they do not require explicitly constructing the function $f(y_n, t_n)$. By contrast, BDF methods are easy and inexpensive to implement because they only require calculating the A coefficients and retaining the earlier neutron fluxes for additional time steps.

Unlike the Theta method, BDF methods of order two and higher are not self-starting. On the first time step (i.e. solving for y_1) there is only one previous solution (i.e. y_0, the steady-state solution). As a practical matter, we address this issue by using low-order BDF methods and allowing the order to ascend sequentially as more data is available.

BDF methods have not been widely used for nuclear reactor spatial kinetics. Ginestar et al. [Gin98] used BDF to solve the time-dependent neutron diffusion equation for both a constant time step as well as an adaptive time step algorithm using the step-doubling method [Cro96]. Garcia et al. [Gar05] used a publically-available ODE solver FCVODE to solve the time-dependent diffusion equation. This solver employed BDF, although Garcia did not discuss the accuracy of the method relative to any other time integrators as the paper was more focused on parallelization. More recently, Shim et al. [Shi11] implemented BDF in the neutron diffusion code RENUS. This implementation included a sophisticated adaptive time stepping algorithm based on a comparison of the BDF-N and BDF-N+1 methods.

BDF methods have not been employed for neutron transport prior to the work described in this thesis. This is likely because of the excessive memory expense of storing the angular flux from one previous point in time (e.g. for Backward Euler). This expense increases linearly with the order of the BDF method. However, if low-order angular approximations to the angular flux time derivative are employed, the memory

[5] This is also true when comparing a particular order BDF method to the Implicit Adams method of the same order. The Theta method is a specialized subset of the Implicit Adams method.

requirements of high-order BDF are limited while the time integration may still be accurate.

The BDF method was used in this research for two reasons. First, BDF1 is equivalent to Backward Euler, which is well understood and has favorable stability properties. Thus we used BDF1 with a fine time step to establish reference solutions for reactor transients. Second, although higher-order BDF methods require excessive memory to store additional angular fluxes, in this work we developed a new time-dependent transport method which effectively recasts the angular flux time derivative in terms of the neutron source time derivative. When we approximate the source time derivatives to high-order accuracy using backward differences, the solution is closely-related to the solution of the transport equation using the same order BDF to approximate the angular flux time derivative.

2.2.4 Runge-Kutta Methods

Runge-Kutta is a class of multi-stage time integration methods. Different Runge-Kutta methods are specified by their unique tableaus (e.g. Table 2-1 and Table 2-2), where higher-order methods typically have more stages and correspondingly larger tableaus. For explicit Runge-Kutta methods, the tableau is a lower-triangular matrix, e.g.:

Table 2-2. A tableau for an arbitrary explicit Runge-Kutta method

$$\begin{array}{c|ccccc}
0 & & & & & \\
C_2 & A_{2,1} & & & & \\
C_3 & A_{3,1} & A_{3,2} & & & \\
\vdots & \vdots & \vdots & \ddots & & \\
C_S & A_{S,1} & A_{S,2} & \cdots & A_{S,S-1} & \\
\hline
 & B_1 & B_2 & \cdots & B_{S-1} & B_S
\end{array}$$

This allows the evaluation of each stage sequentially. Explicit Runge-Kutta is efficient because it entails a linear increase in computational requirements with the number of stages. These methods are popular for solving time-dependent differential equations where explicit methods are appropriate, especially when advanced features are incorporated into the method such as lower-order embedded methods for adaptive time stepping or error monitoring.

However, the time-dependent neutron transport equation is a stiff differential equation. When explicit methods are applied to stiff equations, very small time steps are necessary to maintain stability. Consequently, implicit methods are preferred for stiff equations. For implicit Runge-Kutta methods (e.g. Table 2-1), the S stages are implicitly coupled and must be solved simultaneously. This results in a super-linear increase in computing requirements with the number of stages, which rapidly becomes more expensive than linear multi-step methods of the same order. In addition, for neutron transport methods that are not conventionally solved by setting up a system of equations (e.g. MOC) this requires a substantial modification of the solution procedure. Because of these disadvantages, implicit Runge-Kutta methods are not as widely-used for the solution of stiff differential equations as linear multi-step methods.

Implicit Runge-Kutta methods have been used occasionally for neutron diffusion-based reactor kinetics. In a recent example, Aboanber and Hamada [Abo08] implemented a fourth-order accurate Rosenbrock Runge-Kutta method in the neutron diffusion computer code TGRK. Rodrigues de Lima et al. [Rod09] used the same fourth-order Rosenbrock Runge-Kutta method for time-dependent neutron diffusion.

Runge-Kutta methods have been used rarely for neutron transport, and only for low dimensionality problems. Yang and Jevremovic [Yan10] used a Rosenbrock Runge-Kutta method for time-dependent MOC in 1D, but the equations are substantially more complex than other time-dependent MOC methods due to the implicit coupling between stages along characteristics. Further, unlike conventional MOC methods, the equations were solved by constructing a linear system of equations representing the characteristics rather than using MOC sweeps.

2.2.5 Space-Time Transport Methods

Space-time transport methods are an accurate but expensive alternative to linear multi-step methods and multi-stage methods for neutron transport. Whereas in linear multi-step methods and multi-stage methods the time dependence is treated by discretizing the equation in time and approximating the time derivative using a finite differencing technique, in space-time methods the transport equation is allowed to vary continuously in time over short intervals. This allows for explicit tracking of the neutron time-of-flight and ensures the preservation of causality.

In space-time transport methods, the technique that is used to treat the spatial derivatives (e.g. CPM or MOC) is also used to treat the time derivative. For example, whereas in steady-state MOC the characteristics are defined to traverse the spatial dimensions, in space-time MOC the characteristics traverse space and time. The solution to the space-time transport equation for a particular transport method is similar in form to the steady-state solution, but for space-time transport methods neutrons are tracked within an additional dimension which requires the evaluation of many more equations.

Space-time transport methods typically track neutron transport along discrete ordinates (e.g. S_N, CPM, or MOC, but not P_N). There are two significant differences between transport in the spatial dimensions and the temporal dimension. First, whereas neutrons travel both directions along an ordinate in space (e.g. forward and backward), the neutrons only travel forward in time. Second, while there is a large degree of freedom to specify the discrete ordinates in space, the angles in the space-time plane are defined by the neutron velocity; when the widely-used multi-group approximation[6] is employed, these angles are fixed by the energy group structure.

While space-time transport methods are in principle very accurate, they are substantially more computationally expensive and memory intensive than linear multi-step methods and multi-stage methods. Further, for most problems in nuclear reactor kinetics, the error from not explicitly tracking the neutron time-of-flight is very small.

[6] The multi-group approximation is discussed in Chapter 3.

This is demonstrated by observing that the neutron mean-free-path is much smaller than the distance that a neutron can travel within one time step.

Consequently, space-time transport methods have not been employed to solve large multi-dimensional nuclear reactor kinetics problems. However, space-time transport methods have been used for 1D reactor kinetics and 3D radiative transport for non-reactor geometries. Space-time S_N methods were used to understand ray effects in papers by Zerr and Baker [Zer11] and Barbarina et al. [Bar12]. Keller and Lee [Kel98] developed a space-time CPM method in 1D for nuclear reactor kinetics. Pandya and Adams [Pan09] developed a 3D space-time MOC method for radiative transfer, while Tsujita et al. [Tsu12] implemented a 1D space-time MOC method for reactor kinetics.

To limit the computational and memory expense of space-time neutron transport, we developed a space-time MOC method that employed approximations in the space-time planes [Hof13a]. These approximations brought the computational expense of the new method in line with conventional time-dependent MOC methods using linear multi-step methods. However, while studying this Analytically-integrated Space-Time MOC method (ASTC), we discovered that ASTC represented a subset of a more general class of methods based on Angular Flux Time Derivative Propagation (TDP). ASTC is derived in Appendix B, while the TDP methods are the focus of this thesis.

2.3 Approximations to Angular Dependence of the Angular Flux Time Derivative

Storing the angular flux from one or more previous points in time to represent the angular flux time derivative is prohibitively memory-intensive for large reactor problems. Several researchers have investigated options for approximating the angular-dependence of the angular flux time derivative to reduce the memory requirements of time-dependent neutron transport. The following sections summarize some options for addressing this problem.

2.3.1 Low-Order Angular Approximations

A common resolution to the angular flux storage problem is to use a low-order angular approximation to the angular flux time derivative. The most popular approximation [Cho05, Tsu13, Tal13] has been to assume that the angular flux time derivative is isotropic and approximate it with the scalar flux:

$$\frac{d\varphi(\mathbf{x},\mathbf{\Omega},E,t)}{dt} \approx \frac{1}{4\pi}\frac{d\phi(\mathbf{x},E,t)}{dt}. \tag{2.33}$$

Storing the scalar flux from previous points in time requires substantially less memory than the angular flux. This approximation is very efficient and is attractive when combined with the isotropic source approximation[7]. However, this approximation results in an error term which is proportional to the time derivative of the first moment of the angular flux (i.e. the time derivative of the neutron current)[8], and thus it may not be adequate for some transients.

This approximation can be generalized as approximating the angular flux time derivative using a low-order spherical harmonics expansion:

$$\frac{d\varphi(\mathbf{x},\mathbf{\Omega},E,t)}{dt} \approx \sum_{n=0}^{N}\sum_{m=-n}^{n}\frac{d\varphi_n^m(\mathbf{x},E,t)}{dt}Y_n^m(\mathbf{\Omega}). \tag{2.34}$$

Although this approximation is slightly more memory- and computationally-expensive than the isotropic approximation, it better preserves the angular dependence of the time derivative. For low-order expansions, it requires less memory than storing the angular flux. This approximation is more attractive if the spherical harmonics expansion of the angular flux is being calculated in any case (e.g. because P_N is the neutron transport method or spherical harmonics are used to expand the scattering source); if the spherical harmonics expansion of the angular flux is not otherwise being calculated, this adds an additional computation expense to the approximation.

A third option is to approximate the angular flux time derivative using the time derivative of the angular flux on a coarser spatial mesh, e.g.:

$$\frac{d\varphi(\mathbf{x},\mathbf{\Omega},E,t)}{dt} \approx \frac{d\bar{\varphi}^r(\mathbf{\Omega},E,t)}{dt}, \tag{2.35}$$

[7] This approximation is described in Chapter 3.

[8] This is demonstrated in Chapter 6.

where $\bar{\varphi}^r$ is the average angular flux for a region r in which the spatial position $\mathbf{x}$ lies. This approximation may be more or less accurate than the previous approximation depending on the coarseness of the mesh and the number of flux moments used to approximate the time derivative.

Prior to developing the new method described in this thesis, we tested each of these three options [Hof13b]. While these approximations were reasonably accurate for small reactor kinetics problems, we sought to develop a new time-dependent neutron transport method that would circumvent the angular flux storage problem. This new method is briefly described in Section 2.3.3 .

2.3.1.1 Asymmetric Approximations of the Angular Flux Time Derivative

It is noteworthy that some researchers have considered or directly used the low-order angular approximations listed above (e.g. the scalar flux or spherical harmonics approximation) but only applied the approximation to the previous term in the finite-differenced angular flux time derivative [Pau03, Tay09, Ban12]. In other words, they have used an asymmetric angular flux time derivative approximation like:

$$\frac{d\varphi(\mathbf{x},\mathbf{\Omega},E,t)}{dt} \approx \frac{\varphi(\mathbf{x},\mathbf{\Omega},E,t) - \frac{1}{4\pi}\phi(\mathbf{x},E,t-\Delta t)}{\Delta t}. \qquad 2.36$$

With simple error analysis we showed that this results in an error term which is proportional to the error in the angular flux approximation; this is in contrast to when all terms in the time derivative are similarly approximated, which results in an error term proportional to the *first time derivative* of the error in the angular flux approximation [Hof13b]. In addition, the asymmetric approximation of the angular flux time derivative results in an inconsistent equation. As a result, it is preferable to use the same approximation with all terms in the time derivative if possible.

2.3.2 On-the-Fly Angular Flux Recalculation

An alternative to storing the angular flux from one or more previous points in time is to recalculate the angular flux as needed and discard the value when it is no longer required. This necessitates storing the required parameters from the previous point in time (e.g. cross sections, boundary conditions, and neutron sources). Storing these parameters may require less memory than storing the angular flux. This is the conceptual basis for the on-the-fly method, which was recently developed for MOC by Tsujita et al. [Tsu13].

However, this introduces a new problem. Suppose that Backward Euler is used to approximate the time derivative, i.e.:

$$\frac{d\varphi(t_n)}{dt} \approx \frac{\varphi(t_n) - \varphi(t_{n-1})}{\Delta t}, \qquad 2.37$$

where the spatial, angular, and energy dependence of the angular flux is suppressed for brevity.

For the on-the-fly method with Backward Euler, to approximate the time derivative at time t_n, $\varphi(t_{n-1})$ is recalculated on-the-fly using data at time t_{n-1}. However, if Backward Euler is used for time t_{n-1}, then $\varphi(t_{n-1})$ is a function of $\varphi(t_{n-2})$, so that angular flux $\varphi(t_{n-2})$ should also be recalculated on-the-fly. This dependency continues back to the beginning of the transient. In principle the on-the-fly method could be used to solve the time-dependent transport equation without storing the angular flux and without approximating the angular dependence of the angular flux time derivative, but this results in a linear increase in the computational and memory requirements with the number of time steps. With enough time steps, the on-the-fly method will require more memory than storing the angular flux.

As a compromise to limit this expense, Tsujita proposed limiting the number N of previous angular fluxes that are recalculated on-the-fly. For the earliest angular flux that is recalculated $\varphi(t_{n-N})$, the isotropic approximation to the angular flux time derivative is applied. Although this introduces some error, the error is mitigated because with each successive previous angular flux that is recalculated on-the-fly, the error term is again divided by the neutron velocity. For example, if the isotropic approximation is applied to $\varphi(t_{n-N})$, the error term in $\varphi(t_n)$ is proportional to $\frac{1}{v^{N+1}}$. Since the neutron velocity is large, the error rapidly decreases with N, and so the user can adjust N to achieve the desired accuracy. When $N = 0$, the method is equivalent to the familiar isotropic approximation.

Tsujita implemented an on-the-fly method in an MOC-based neutron transport code, which was referred to as an “on-flight” method. Tsujita tested the on-flight method for a few transients and concluded that the isotropic approximation was reasonably accurate, and that with higher N the on-flight method becomes more accurate.

2.3.3 Analytically-Integrated Space-Time Characteristics

In this thesis, we develop a new time-dependent MOC method that circumvents the angular flux storage issue by propagating the angular flux time derivative along characteristics. In steady-state MOC, the angular flux is propagated along characteristics in space. The angular flux at any point along a characteristic can be written as a function of the neutron sources along that characteristic which are attenuated by the intervening materials. In the new method for time-dependent MOC, we devise equations for propagating the first time derivative of the angular flux along characteristics in terms of the first time derivatives of the neutron sources along the characteristic which are similarly attenuated by intervening materials. This is the basis for Angular Flux Time Derivative Propagation (TDP) methods. This thesis focuses on a subset of these methods which can recast the angular flux time derivative in terms of the propagated effects of source-derivatives, which are hence called Source Derivative Propagation (SDP) methods.

A limited form of SDP methods was originally developed using space-time MOC with approximations to the space-time characteristic in the time domain. This method was called analytically-integrated space-time characteristics (ASTC) [Hof13a]. However, this thesis presents a more general form of SDP methods. These methods are presented and analyzed in detail in the following chapters.

The SDP methods have similar memory requirements as the on-flight methods developed by Tsujita. The SDP methods also have a similar leading error term, which will be proportional to some power of the inverse velocity. When Backward Euler is employed for on-flight with $N = 1$ and SDP methods, they have similar computational expense. The primary advantage of SDP methods over on-flight methods is that SDP methods can accommodate high-order approximations for the time derivatives (e.g. using backward differences) without a substantial increase in computational or memory expense. By contrast, when the time derivatives are approximated to high-order accuracy using the on-flight method, the computational and memory requirements increase linearly with the order of accuracy.

2.4 Delayed Neutron Precursor Equation

When fission occurs a few neutrons are released. Most of these neutrons are emitted promptly, but some neutrons are delayed. These delayed neutrons are released when certain fission products (i.e. delayed neutron precursor nuclides) decay by neutron emission. For steady-state neutron transport, the delayed neutrons are in equilibrium and do not require explicit treatment. However, for time-dependent problems the delayed neutrons are important.

The buildup and decay of the delayed neutron precursor concentration C_i for the precursor i is described by the differential equation:

$$\frac{dC_i(t)}{dt} = \beta_i \psi(t) - \lambda_i C_i(t), \qquad 2.38$$

where ψ is the total neutron fission source, β_i is the fractional delayed neutron precursor yield, λ_i is the decay constant, and the spatial, angular, and energy dependences have been suppressed for brevity.

The total delayed neutron source S_d is the sum of the decay rates of the isotopes multiplied by the delayed neutron spectrum $\chi_{d,i}$:

$$S_d(t) = \sum_i \chi_{d,i} \lambda_i C_i(t). \qquad 2.39$$

While in principle we could explicitly track the buildup and decay of the dozens of unique delayed neutron precursors, this is not done in practice in reactor kinetics. This is because the precursor yields and decay constants are not accurately known, and this explicit treatment is computationally expensive. Instead, we define a small number of artificial delayed neutron precursor groups. These groups combine many delayed neutron precursors with similar decay constants. The group parameters (i.e. yields and decay constants) are more easily measured experimentally than those of the individual isotopes.

The delayed neutron precursor group equation is similar to Equation 2.38:

$$\frac{dC_j(t)}{dt} = \beta_j \psi(t) - \lambda_j C_j(t), \tag{2.40}$$

where j is the index for the delayed neutron group.

Equation 2.40 is generally solved by one of two ways in time-dependent neutron transport. The first approach is to discretize the precursor equation in time and treat the time derivative using a conventional time integration technique. In this approach, the precursor equation is solved in conjunction with the transport equation. The second option is to make some assumption about the temporal variation of the neutron fission source (e.g. it is linear in time). This allows the precursor equations to be eliminated from the transport equation. In the following section we will derive each of these approaches.

2.4.1 Solution by Time Integration

The solution of the delayed neutron precursor equation by time integration is straightforward. The equation is discretized in time and a time integration technique is applied. If Backward Euler is used, the precursor equation becomes:

$$\frac{C_j(t_n) - C_j(t_{n-1})}{\Delta t} = \beta_j \psi(t_n) - \lambda_j C_j(t_n). \tag{2.41}$$

The equation is solved by isolating the delayed neutron precursor concentration:

$$C_j(t_n) = \frac{\beta_j \psi(t_n) + \frac{C_j(t_{n-1})}{\Delta t}}{\frac{1}{\Delta t} + \lambda_j}. \tag{2.42}$$

Equation 2.42 is coupled to the neutron transport equation by the fission source, which is a function of the angular flux. Similarly, the neutron transport equation is dependent upon the precursor group concentrations through the delayed neutron source term. As a result, the solutions to the delayed neutron precursor equations are implicitly coupled to the solutions for the neutron transport equation, and the two must be solved simultaneously.

2.4.2 Analytical Precursor Integration

Analytical precursor integration is a common technique to avoid explicitly solving the delayed neutron precursor equations simultaneously with the neutron transport equation. By assuming that the fission source has a known temporal variation, the dependency of the neutron transport equation on the delayed neutron precursor concentration can be eliminated. In this section we derive analytical precursor integration with a quadratic fission source.

With a linear approximation, the fission source near time t_n is:

$$\begin{aligned}\psi(t) &= \psi^n + (t - t_n)\frac{\psi^n - \psi^{n-1}}{\Delta t_n} \\ &= \psi^n - t_n\frac{\psi^n - \psi^{n-1}}{\Delta t_n} + t\frac{\psi^n - \psi^{n-1}}{\Delta t_n},\end{aligned} \tag{2.43}$$

where $\psi^n \equiv \psi(t_n)$ and $\Delta t_n \equiv t_n - t_{n-1}$.

We apply this approximation Equation 2.40:

$$\frac{dC_j(t)}{dt} + \lambda_j C_j(t) = \beta_j\left[\psi^n - t_n\frac{\psi^n - \psi^{n-1}}{\Delta t_n} + t\frac{\psi^n - \psi^{n-1}}{\Delta t_n}\right]. \tag{2.44}$$

Next we apply the integrating factor $e^{\lambda_j(t-t_n)}$ to each side:

$$\frac{d}{dt}\left(C_j(t)e^{\lambda_j(t-t_n)}\right) = e^{\lambda_j(t-t_n)}\beta_j\left[\psi^n - t_n\frac{\psi^n - \psi^{n-1}}{\Delta t_n} + t\frac{\psi^n - \psi^{n-1}}{\Delta t_n}\right]. \tag{2.45}$$

We solve this equation by integrating both sides over the time step:

$$\int_{t_{n-1}}^{t_n} \frac{d}{dt}\left(C_j(t)e^{\lambda_j(t-t_n)}\right)dt$$
$$= \beta_j \int_{t_{n-1}}^{t_n} e^{\lambda_j(t-t_n)}\left(\psi^n - t_n\frac{\psi^n - \psi^{n-1}}{\Delta t_n}\right) + te^{\lambda_j(t-t_n)}\frac{\psi^n - \psi^{n-1}}{\Delta t_n}dt, \tag{2.46}$$

which evaluates to:

$$C_j(t_n) - C_j(t_{n-1})e^{-\lambda_j \Delta t_n}$$
$$= \frac{\beta_j}{\lambda_j}\left[\left(1 - e^{-\lambda_j \Delta t_n}\right)\left(\psi^n - t_n\frac{\psi^n - \psi^{n-1}}{\Delta t_n}\right) + \left(t_n - t_{n-1}e^{-\lambda_j \Delta t_n} - \frac{1 - e^{-\lambda_j \Delta t_n}}{\lambda_j}\right)\frac{\psi^n - \psi^{n-1}}{\Delta t_n}\right]. \tag{2.47}$$

With some algebra we isolate the new delayed precursor group concentration in terms of the previous delayed neutron precursor concentration and the new and previous fission sources:

$$C_j(t_n) = C_j(t_{n-1})e^{-\lambda_j \Delta t_n} + \frac{\beta_j}{\lambda_j}\left[\psi^n\left(1 - \frac{1 - e^{-\lambda_j \Delta t_n}}{\lambda_j \Delta t_n}\right) - \psi^{n-1}\left(e^{-\lambda_j \Delta t_n} - \frac{1 - e^{-\lambda_j \Delta t_n}}{\lambda_j \Delta t_n}\right)\right]. \tag{2.48}$$

In analytical precursor integration, Equation 2.48 is used to eliminate the dependence of the delayed neutron source on the new delayed neutron precursor concentration.

2.5 Summary

This chapter provides a summary of neutron transport and time integration methods relevant to time-dependent neutron transport for nuclear reactor kinetics. This is the context in which new time-dependent neutron transport methods were developed.

In the following chapters, we present several neutron transport methods based on MOC using BDF for time integration. Chapter 3 provides a steady-state MOC method which is used to initialize the transient problem and provide context for typical MOC conventions. Chapter 4 provides a conventional time-dependent MOC method that uses a finite difference technique to treat the angular flux time derivative. Finally, Chapter 5 provides the new time-dependent MOC method developed for this thesis.

Chapter 3

Steady-State MOC (SSC) Derivation

Method of Characteristics (MOC) has been widely used to solve the steady-state neutron transport equation for 2D assembly models [Hal80, Jon00, Smi00]. More recently, Steady-State Method of Characteristics (SSC) has been applied to solving larger 3D problems, either using 3D MOC or as a part of coupled 2D/1D formulations [Joo04, Col13, Tal13]. SSC methods have been used to perform criticality eigenvalue calculations, drive depletion models, and initialize reactor transients. This chapter provides a derivation of SSC in 3D for a criticality eigenvalue model which is representative of typical applications of MOC to reactor physics. In this work, the SSC method is used to initialize reactor transients which are modeled using the time-dependent MOC methods in the following chapters.

Although this derivation is formally for 3D MOC, one of the convenient features of characteristic methods is that if we carefully define our characteristic transform for any given dimensionality, the characteristic equations will be essentially identical regardless of the dimension; the only difference will be the definition of some quantities and the transform. Thus the equations in this derivation can also be easily adapted to problems in 1D and 2D with minor modifications.

This derivation begins with the steady-state Boltzmann transport equation. We will then discuss each of the approximations applied to the transport equation and illustrate how the transport equation changes with each approximation. When the characteristic transform is applied, the resulting differential equation is called the SSC characteristic equation. Finally, we show how the characteristic equation is solved and how the solutions are used in SSC.

3.1 Steady-State Boltzmann Transport Equation

The Boltzmann transport equation in steady-state can be written as:

$$\mathbf{\Omega} \cdot \mathbf{\nabla}\varphi(\mathbf{x}, \mathbf{\Omega}, E) + \Sigma_t(\mathbf{x}, E)\varphi(\mathbf{x}, \mathbf{\Omega}, E) = q(\mathbf{x}, \mathbf{\Omega}, E), \qquad 3.1$$

where the source q includes fission and scattering:

$$q(\mathbf{x}, \mathbf{\Omega}, E) = \frac{\chi(\mathbf{x}, E)}{4\pi k_{\text{eff}}} \int_0^\infty \int_{4\pi} \nu\Sigma_f(\mathbf{x}, E')\varphi(\mathbf{x}, \mathbf{\Omega}', E')d\mathbf{\Omega}'dE' \\ + \int_0^\infty \int_{4\pi} \Sigma_s(\mathbf{x}, \mathbf{\Omega}' \cdot \mathbf{\Omega}, E' \to E)\varphi(\mathbf{x}, \mathbf{\Omega}', E')d\mathbf{\Omega}'dE', \qquad 3.2$$

where

$\mathbf{x} =$ position vector $= x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$

$\mathbf{\Omega} =$ [unit] direction of flight vector $= \sqrt{1-\mu^2}\cos\alpha\,\mathbf{i} + \sqrt{1-\mu^2}\sin\alpha\,\mathbf{j} + \mu\mathbf{k}$

$\alpha =$ azimuthal angle

$\mu = \cos\theta =$ cosine of the polar angle

$E =$ neutron kinetic energy

$\varphi =$ neutron angular flux

$\Sigma_t =$ macroscopic total cross section

$q =$ total neutron source

$\chi =$ total fission neutron energy distribution

$k_{\text{eff}} =$ k-eigenvalue

$\nu =$ number of neutrons released per fission

$\Sigma_f =$ macroscopic fission cross section

$\Sigma_s(\mathbf{x}, \mathbf{\Omega}' \cdot \mathbf{\Omega}, E' \to E) =$ macroscopic differential scattering cross section.

3.2 Approximations to the Neutron Transport Equation for SSC

We employ several approximations to solve the steady-state neutron transport equation using MOC. In this section we discuss and incorporate each approximation to the Boltzmann equation in turn.

3.2.1 Multi-group Approximation

To treat the energy dependence, we will apply the widely-used multi-group approximation. We discretize the energy domain into G discrete groups, where the upper and lower boundaries for group g are E_g and E_{g-1}. It is customary to associate the lower group numbers with higher energies because neutrons are produced by fission at high energies and lose energy by scattering.

If we integrate Equation 2.1 over an arbitrary energy group g, the result is the steady-state, multi-group neutron transport equation:

$$\mathbf{\Omega} \cdot \mathbf{\nabla}\varphi_g(\mathbf{x}, \mathbf{\Omega}) + \Sigma_{t,g}(\mathbf{x})\varphi_g(\mathbf{x}, \mathbf{\Omega}) = q_g(\mathbf{x}, \mathbf{\Omega}), \tag{3.3}$$

where the multi-group source is:

$$\begin{aligned} q_g(\mathbf{x}, \mathbf{\Omega}) = \sum_{g'=1}^{G} \Bigg(& \frac{\chi_g(\mathbf{x})}{4\pi k_{\text{eff}}} \int\limits_{4\pi} \nu\Sigma_{f,g'}(\mathbf{x})\varphi_{g'}(\mathbf{x}, \mathbf{\Omega}')d\mathbf{\Omega}' \\ & + \int\limits_{4\pi} \Sigma_{s,g' \to g}(\mathbf{x}, \mathbf{\Omega}' \cdot \mathbf{\Omega})\varphi_{g'}(\mathbf{x}, \mathbf{\Omega}')d\mathbf{\Omega}' \Bigg). \end{aligned} \tag{3.4}$$

The full steady-state transport equation is represented by G equations representing G energy groups. These equations are coupled through the production of fission neutrons and the scattering process.

3.2.2 Isotropic Source Approximation

The computational efficiency of MOC is substantially improved if we approximate the source terms as isotropic. This approximation is frequently applied to the fission source in reactor physics because it is nearly isotropic:

$$\begin{aligned} & \sum_{g'=1}^{G} \left(\frac{\chi_g(\mathbf{x})}{4\pi k_{\text{eff}}} \int\limits_{4\pi} \nu\Sigma_{f,g'}(\mathbf{x})\varphi_{g'}(\mathbf{r}, \mathbf{\Omega}')d\mathbf{\Omega}' \right) \\ & \qquad = \sum_{g'=1}^{G} \left(\frac{\chi_g(\mathbf{x})}{4\pi k_{\text{eff}}} \nu\Sigma_{f,g'}(\mathbf{x})\phi_{g'}(\mathbf{x}) \right), \end{aligned} \tag{3.5}$$

where the scalar flux ϕ is defined as the integral of the angular flux over all angles by:

$$\phi_{g'}(\mathbf{x}) = \int\limits_{4\pi} \varphi_{g'}(\mathbf{x}, \mathbf{\Omega}')d\mathbf{\Omega}'. \tag{3.6}$$

However, the scattering source may exhibit significant anisotropy. Many MOC methods employ an approximation called transport-corrected scattering to maintain the computational benefits of an isotropic source while preserving the first-order anisotropy. This approximation is easy to implement because it only requires replacing the total cross section with the transport cross section and the differential scattering cross section with the transport-corrected scattering cross section. We define the transport cross section in terms of the total cross section and the first moment of the differential scattering cross section:

$$\Sigma_{tr,g}(\mathbf{x}) = \Sigma_{t,g}(\mathbf{x}) - \Sigma_{s1,g}(\mathbf{x}) = \Sigma_{t,g}(\mathbf{x}) - \sum_{g'} \Sigma_{s1,g'\to g}(\mathbf{x}), \tag{3.7}$$

and the transport-corrected scattering cross section is defined in terms of the zeroth and first moment of the differential scattering cross section:

$$\Sigma_{s,g'\to g}^{tr}(\mathbf{x}) = \begin{cases} \Sigma_{s0,g'\to g}(\mathbf{x}) & g \neq g' \\ \Sigma_{s0,g\to g}(\mathbf{x}) - \Sigma_{s1,g}(\mathbf{x}) & g = g' \end{cases}. \tag{3.8}$$

When these approximations are applied to the steady-state multi-group neutron transport equation, the source may be treated as isotropic:

$$\mathbf{\Omega} \cdot \mathbf{\nabla} \varphi_g(\mathbf{x}, \mathbf{\Omega}) + \Sigma_{tr,g}(\mathbf{x}) \varphi_g(\mathbf{x}, \mathbf{\Omega}) = q_g(\mathbf{x}), \tag{3.9}$$

where the angular integral of the scattering source has been resolved:

$$q_g(\mathbf{x}) = \frac{1}{4\pi} \sum_{g'=1}^{G} \left(\frac{\chi_g(\mathbf{x})}{k_{\text{eff}}} \nu\Sigma_{f,g'}(\mathbf{x}) \phi_{g'}(\mathbf{x}) + \Sigma_{s,g'\to g}^{tr}(\mathbf{x}) \phi_{g'}(\mathbf{x}) \right). \tag{3.10}$$

3.2.3 Discrete Ordinates Approximation

Next we will discretize the angular domain into $L \cdot M$ directions described by L azimuthal angles α_l and M polar angles θ_m. Each azimuthal and polar angle has a corresponding angular weight w_l and w_m which reflects the area on the unit sphere that the discrete ordinate $\mathbf{\Omega}_{l,m}$ represents. A set of angular directions and weights is called a quadrature set. The optimization of quadrature sets is a field of active research and beyond the scope of this thesis.

With this approximation, we rewrite the steady-state neutron transport equation in terms of the discrete ordinates:

$$\mathbf{\Omega} \cdot \mathbf{\nabla}\varphi_g\left(\mathbf{x}, \mathbf{\Omega}_{l,m}\right) + \Sigma_{tr,g}(\mathbf{x})\varphi_g\left(\mathbf{x}, \mathbf{\Omega}_{l,m}\right) = q_g(\mathbf{x}). \qquad 3.11$$

3.2.4 Characteristic Transform

Next we use MOC to rewrite the partial differential equation as an ordinary differential equation along a characteristic[9] in the spatial domain. Here we begin with the definition of the characteristic as informed by the definition of the total derivative with respect to partial derivatives, e.g.:

$$\frac{d\varphi(x,y,z)}{ds} = \frac{\partial\varphi}{\partial x}\frac{dx}{ds} + \frac{\partial\varphi}{\partial y}\frac{dy}{ds} + \frac{\partial\varphi}{\partial z}\frac{dz}{ds}. \qquad 3.12$$

Thus if we define the characteristic using:

$$\begin{gathered} \frac{dx}{ds} = \sqrt{1-\mu_m^2}\cos\alpha_l \\ \frac{dy}{ds} = \sqrt{1-\mu_m^2}\sin\alpha_l, \\ \frac{dz}{ds} = \mu_m \end{gathered} \qquad 3.13$$

where s is the dimension along the characteristic, then we can rewrite the steady-state, multi-group, neutron transport equation as an ordinary differential equation along a characteristic uniquely defined by an arbitrary position $\mathbf{x_0}$ and a discrete ordinate $\mathbf{\Omega}_{l,m}$:

$$\begin{aligned} \frac{d\varphi_g\left(\mathbf{x_0} + s\mathbf{\Omega}_{l,m}, \mathbf{\Omega}_{l,m}\right)}{ds} &+ \Sigma_{tr,g}\left(\mathbf{x_0} + s\mathbf{\Omega}_{l,m}\right)\varphi_g\left(\mathbf{x_0} + s\mathbf{\Omega}_{l,m}, \mathbf{\Omega}_{l,m}\right) \\ &= q_g\left(\mathbf{x_0} + s\mathbf{\Omega}_{l,m}\right). \end{aligned} \qquad 3.14$$

For succinctness we will introduce the index k for each unique characteristic:

$$\frac{d\varphi_g^k(s)}{ds} + \Sigma_{tr,g}^k(s)\varphi_g^k(s) = q_g^k(s), \qquad 3.15$$

where $k = f\left(\mathbf{\Omega}_{l,m}, \mathbf{x_0}\right)$.

[9] Characteristics are also occasionally called “rays” or “tracks” in literature on MOC for reactor physics.

Equation 3.15 is the characteristic equation for SSC. When we apply MOC to neutron transport we span the spatial domain with many unique characteristics at different angles and positions in space. Each characteristic is representative of the volume of space around that characteristic, and by carefully recombining the solutions to all of the characteristics we can solve the neutron transport equation over the spatial and angular domain.

3.2.5 Spatial Discretization and Step Characteristics

Equation 3.15 can be analytically solved if we make approximations to the spatial dependence of the source and transport cross section along the characteristic. Since the spatial variation of these quantities is complex, we will discretize our problem domain into many fine spatial regions r. Figure 3-1 shows an example spatial meshing for a single pin cell. One of the significant advantages of MOC over other neutron transport methods is that MOC can easily accommodate curved surfaces without homogenization.

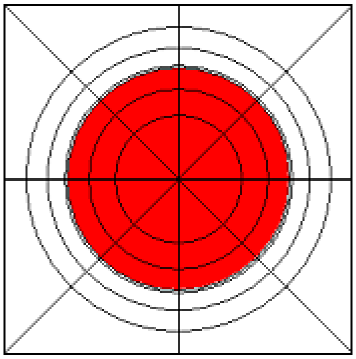


Figure 3-1. Example region meshing for a pin cell in MOC [Hur08]

When we span the spatial domain with characteristics, we define a segment as the portion of a characteristic within a region. Since all of a characteristic lies within one region or another, the characteristic can be thought of as a series of segments. Figure 3-2 displays many characteristics at a particular angle within a discretized pin cell. By approximating the spatial dependence of the source and cross section within each region, we can analytically solve the characteristic equation for each segment.

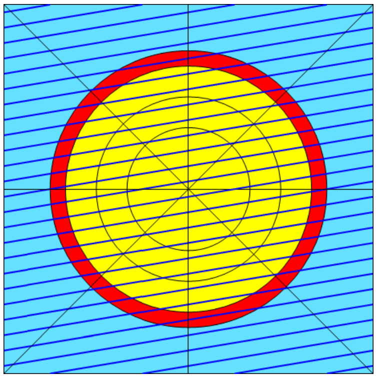


Figure 3-2. Example pin cell meshing with characteristics [adapted from Hur08]

For the source approximation, we will use the “step characteristics” method: the source is assumed to be spatially-invariant within the region:

$$q_g^k(s) = q_g^r, \tag{3.16}$$

where the region r corresponds to the spatial position s along characteristic k.

Step characteristics is popular for MOC because it is very computationally-efficient. An alternative is to allow the source to have some low-order spatial variation within the region, e.g. “linear characteristics,” which assumes that the source varies linearly in space. While this might allow us to use larger spatial regions, calculating the spatial distribution of the source within the region adds substantial cost and complexity to the method.

Further, in order to analytically solve the characteristic equation it is necessary to assume that the transport cross section is also spatially-invariant within the region:

$$\Sigma_{tr,g}^k(s) = \Sigma_{tr,g}^r. \tag{3.17}$$

This approximation is a bounding criterion for determining how large spatial regions can be, and it limits the advantage of employing linear characteristics.

When these approximations are applied to the characteristic equation, the result is the step characteristic equation for steady-state neutron transport:

$$\frac{d\varphi_g^k(s)}{ds} + \Sigma_{tr,g}^r \varphi_g^k(s) = q_g^r. \tag{3.18}$$

3.2.6 Other Assumptions and Approximations

There are a few other common assumptions implicit in this derivation and the use of the Boltzmann transport equation which deserve mentioning. First, we have assumed that the reactor is in a steady-state, equilibrium, critical condition. This describes the state of a nuclear reactor throughout most of the operating cycle, where the neutron flux, material composition, reaction rates, and other properties are changing very slowly. The solution of this equation is representative of a "snapshot" of the reactor at a particular moment. These assumptions are not necessarily valid immediately after a perturbation (e.g. control rod movement or a change in the incoming coolant properties), at which point more formal treatment of time-dependence is necessary.

Based on the assumed steady-state condition, we can ignore the angular flux time derivative because it is negligible. Similarly, we can also ignore the time dependence of the quantities in the transport equation because they are changing very slowly. Also, we can ignore the distinction between prompt and delayed fission neutrons because the delayed neutrons are in equilibrium.

There are also some assumptions inherent in the use of the Boltzmann transport equation for reactor physics. First, we note that the interaction of an individual neutron with matter is inherently stochastic, but the neutron transport equation is strictly deterministic; the use of the transport equation requires that there are enough neutrons in the domain of interest such that the transport equation describes their mean behavior and statistical fluctuations are not important. It also requires that there are not so many neutrons that they change the medium over short time scales (e.g. by rapidly changing the material composition or properties). Finally, we have assumed that the material composing the problem are isotropic.

3.3 Solution of the Step Characteristic Equation

The step characteristic equation can be analytically solved along a segment, e.g. using an integrating factor of $e^{\Sigma_{tr,g}^{r}\left(s-s_0^{k,r}\right)}$:

$$\frac{d}{ds}\left(\varphi_g^k(s)e^{\Sigma_{tr,g}^r\left(s-s_0^{k,r}\right)}\right) = q_g^r e^{\Sigma_{tr,g}^r\left(s-s_0^{k,r}\right)}, \qquad 3.19$$

where $s_0^{k,r}$ is the incoming position[10] of the segment along s where characteristic k enters region r.

We integrate the equation along the segment:

$$\int_{s_0^{k,r}}^{s} \frac{d}{ds}\left(\varphi_g^k(s)e^{\Sigma_{tr,g}^r\left(s-s_0^{k,r}\right)}\right) ds' = \int_{s_0^{k,r}}^{s} q_g^r e^{\Sigma_{tr,g}^r\left(s'-s_0^{k,r}\right)}\, ds', \qquad 3.20$$

which evaluates to:

$$\varphi_g^k(s)e^{\Sigma_{tr,g}^r\left(s-s_0^{k,r}\right)} - \varphi_{g,0}^{k,r} = q_g^r\left(\frac{e^{\Sigma_{tr,g}^r\left(s-s_0^{k,r}\right)}-1}{\Sigma_{tr,g}^r}\right), \qquad 3.21$$

where $\varphi_{g,0}^{k,r} \equiv \varphi_g^k\left(s_0^{k,r}\right)$.

With some algebra we isolate the solution of the equation along the segment, which is the spatially-dependent segment-wise angular flux:

$$\varphi_g^k(s) = \varphi_{g,0}^{k,r}e^{-\Sigma_{tr,g}^r\left(s-s_0^{k,r}\right)} + q_g^r\left(\frac{1-e^{-\Sigma_{tr,g}^r\left(s-s_0^{k,r}\right)}}{\Sigma_{tr,g}^r}\right). \qquad 3.22$$

This is the solution to the step characteristics equation. In SSC, this equation is used for two purposes: (1) to propagate the angular flux from segment to segment along each characteristic, and (2) to numerically integrate the region-wise scalar flux.

3.3.1 Angular Flux Propagation Along the Characteristic

To solve Equation 3.22 we assumed that we knew the incoming angular flux at the beginning of the segment $\varphi_{g,0}^{k,r}$. We can make this assumption because the incoming angular at the problem boundary will be specified by the boundary condition, and we

[10] Note that if the geometry includes curved surfaces, it is possible for a characteristic to enter a region more than once. In practice this occurs very rarely, and although this special case may complicate the implementation of MOC, it does not present a difficulty from a theoretical standpoint. To avoid excessive complexity we will avoid the formal addition of another index to cover this special case (e.g. $s_0^{k,r,i}$).

will calculate the outgoing angular flux for each segment. Since we will evaluate the segments sequentially starting at the problem boundary, we will use the outgoing angular flux from the previous segment as the incoming angular flux for the next segment. Calculating the outgoing angular flux is simply a matter of evaluating Equation 3.22 at the end of the segment:

$$\varphi_g^k\left(s_0^{k,r} + \Delta s^{k,r}\right) = \varphi_{g,0}^{k,r} e^{-\Sigma_{tr,g}^r \Delta s^{k,r}} + q_g^r \left(\frac{1 - e^{-\Sigma_{tr,g}^r \Delta s^{k,r}}}{\Sigma_{tr,g}^r}\right), \qquad 3.23$$

where $\Delta s^{k,r}$ is the length of the segment for characteristic k in region r.

3.3.2 Numerical Integration of the Region-wise Scalar Flux

While the fundamental solution of SSC is the segment-wise angular flux, we prefer not to store these values because it would require excessive memory. Instead, we will numerically integrate the region-wise scalar fluxes by weighting and summing the segment-wise average angular fluxes by region as we generate them.

Thus for each segment—in addition to calculating the outgoing angular flux—we will calculate the segment-wise average angular flux. This is just the spatially-dependent segment-wise angular flux averaged over the segment length:

$$\begin{aligned}\bar{\varphi}_g^{k,r} &= \frac{1}{\Delta s^{k,r}} \int_{s_0^{k,r}}^{s_0^{k,r}+\Delta s^{k,r}} \varphi_g^k(s) ds \\ &= \left(\varphi_{g,0}^{k,r} - \frac{q_g^r}{\Sigma_{tr,g}^r}\right)\left(\frac{1 - e^{-\Sigma_{tr,g}^r \Delta s^{k,r}}}{\Sigma_{tr,g}^r \Delta s^{k,r}}\right) + \frac{q_g^r}{\Sigma_{tr,g}^r}.\end{aligned} \qquad 3.24$$

The region-wise scalar flux is calculated using the segment-wise average angular fluxes for each segment in a region at every angle. Figure 3-3 provides a 2D example for seven segments within a single region where the spacing between characteristics is ΔK_l.

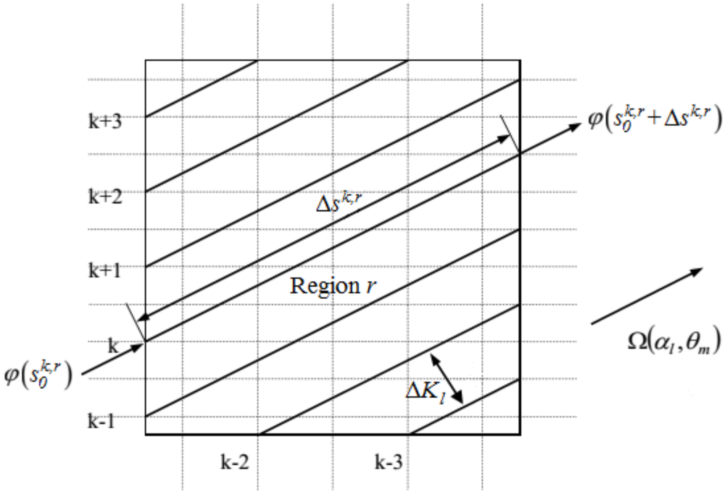


Figure 3-3. Characteristic segments in an example region [adapted from Hur08].

To calculate the region-wise scalar flux, we first need to calculate the region-wise average angular fluxes for each angle, which is the volume-weighted average of the segment-wise average angular fluxes for every segment in a region at a particular angle. The volume[11] that each segment represents is a function of the segment length and the spacing between the characteristics. Using Figure 3-3 as an example, the represented volume of each characteristic is the product of the segment length and the characteristic spacing interval:

$$V^{k,r} = \Delta s^{k,r} \Delta K_l. \tag{3.25}$$

The region-wise average angular flux is the volume-weighted sum of each of the segment-wise average angular fluxes for each segment within the region:

$$\bar{\varphi}_{g,l,m}^{r} = \frac{\sum_{k\in r} \bar{\varphi}_g^{k,r} V^{k,r}}{\sum_{k\in r} V^{k,r}}. \tag{3.26}$$

The region-wise scalar flux is numerically integrated from the region-wise average angular fluxes using the angular weighting factors corresponding to the quadrature set:

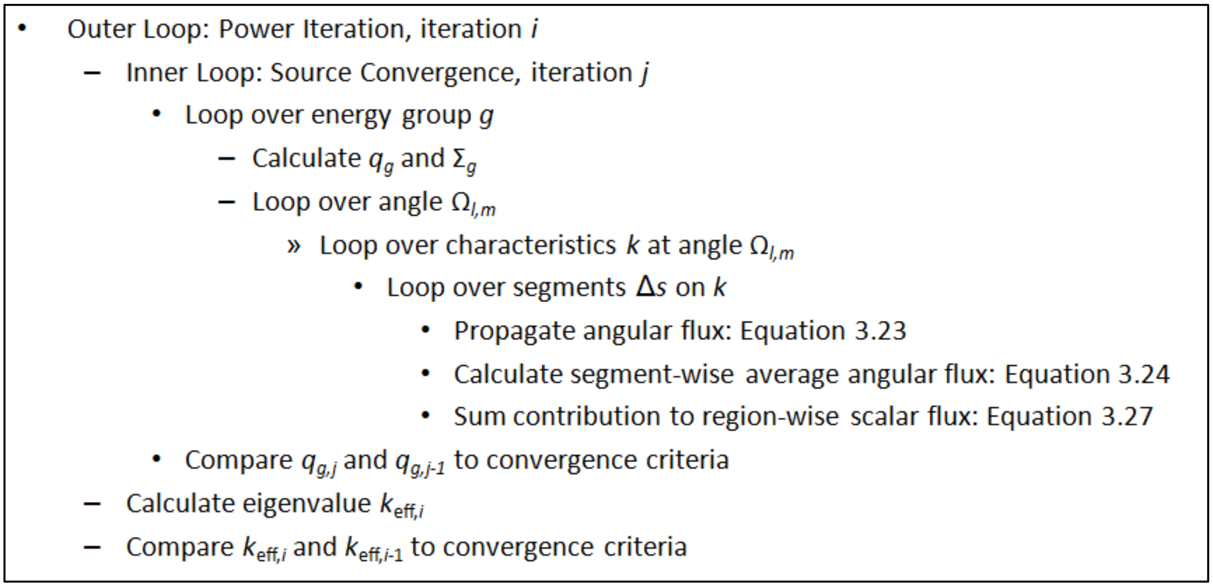
[11] Or area for 2D problems.

$$\phi_g^r = \sum_m^M w_m \sum_l^L w_l \, \bar{\varphi}_{g,l,m}^r . \tag{3.27}$$

Thus in SSC the storage of the angular flux throughout the spatial domain is avoided by numerically integrating the angular flux as it is generated to calculate the scalar flux. When every segment has been evaluated, the scalar fluxes are integrated.

3.4 Summary of SSC Algorithm

MOC methods are an iterative, sweeping process. Using an initial estimate of the scalar flux, the source is calculated. This source is used with Equation 3.23 to propagate the angular flux along each characteristic at each energy group and with Equation 3.27 to calculate a new scalar flux for each region. Once the scalar flux has been calculated, the convergence criteria are checked (e.g. by comparing the scalar flux, fission source, total source, etc.), and if the convergence criteria are not satisfied, another iteration is performed. If an eigenvalue calculation is being performed, the eigenvalue is calculated to balance the neutron production and loss terms. If the eigenvalue has not converged, another iteration is performed. This process is summarized in Figure 3-4. Some technical details are suppressed for clarity (e.g. inner loops for upscattering, etc.).



```
• Outer Loop: Power Iteration, iteration $i$
  – Inner Loop: Source Convergence, iteration $j$
    • Loop over energy group $g$
      – Calculate $q_g$ and $\Sigma_g$
      – Loop over angle $\Omega_{l,m}$
        » Loop over characteristics $k$ at angle $\Omega_{l,m}$
          • Loop over segments $\Delta s$ on $k$
            • Propagate angular flux: Equation 3.23
            • Calculate segment-wise average angular flux: Equation 3.24
            • Sum contribution to region-wise scalar flux: Equation 3.27
    • Compare $q_{g,j}$ and $q_{g,j-1}$ to convergence criteria
  – Calculate eigenvalue $k_{\text{eff},i}$
  – Compare $k_{\text{eff},i}$ and $k_{\text{eff},i-1}$ to convergence criteria
```

Figure 3-4. Nested algorithm for steady-state eigenvalue MOC

In Chapter 7 we provide details on the computer code DeCART which was used to test the new methods developed in this work. SSC is used to initialize the time-dependent MOC methods described in Chapters 4 and 5 for numerical test problems in Chapter 8.

Chapter 4

Time-Dependent MOC with BDF (BDC) Derivation

The typical approach to solving the time-dependent neutron transport equation is to discretize the equation in time and apply a time integration technique to treat the time derivative. Although there are many time integration methods available, they generally result in one or more equations of similar form to the steady-state transport equation but with a transient fixed source term resulting from a finite difference approximation of the time derivative. This is true of the backward differentiation formula (BDF), which is the time integration technique used in this chapter. When BDF is applied to the time-dependent transport equation solved using Method of Characteristics (MOC), the resulting equation is solved using a similar approach to the Steady-State Method of Characteristics (SSC).

This chapter follows a similar structure to the previous chapter, providing a derivation for Time-Dependent MOC with BDF (BDC). We will begin by introducing all of the approximations applied to the time-dependent neutron transport equation. To facilitate comparison between the different MOC methods, in this chapter and other chapters we will discuss and apply each approximation in a deliberate order. We will first consider approximations that are also applied to SSC. Then we will introduce approximations that are common to all of the time-dependent MOC methods. Finally, we will detail approximations specific to BDC.

When the BDC characteristic equations have been derived, this chapter will branch to consider two options for the treatment of the angular flux terms in the angular flux time derivative: an accurate reference method where the angular flux time derivative is approximated using segment-wise angular fluxes (RBDC), and an efficient low-order method that assumes the angular flux time derivative is isotropic (IBDC). The latter approximation uses the scalar flux in place of the angular flux to treat the angular flux time derivative. The relationship between the BDC methods and the other methods investigated in this thesis is graphically-represented in Figure 4-1. This figure shows all of the methods begin with the same equation and shared approximations, but different approximations are used to treat the spatial and temporal derivatives.

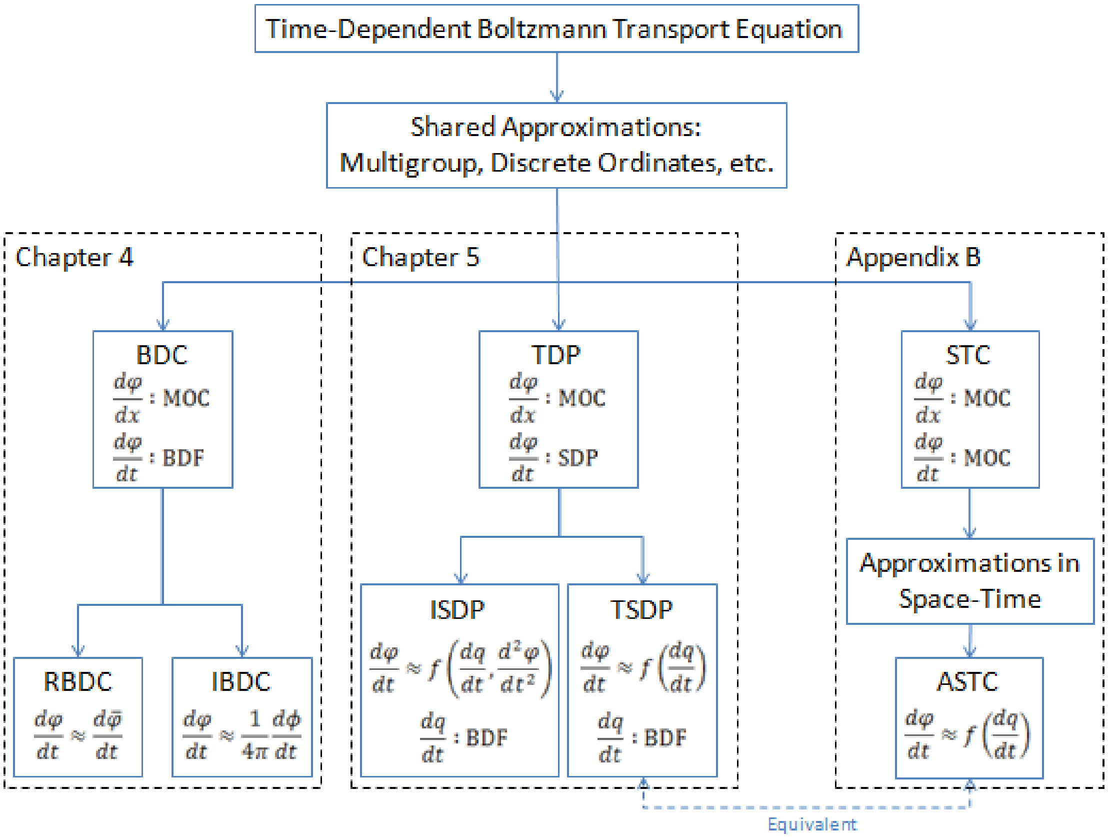

Time-Dependent Boltzmann Transport Equation
Shared Approximations: Multigroup, Discrete Ordinates, etc.
Chapter 4
Chapter 5
Appendix B
BDC
$\frac{d\varphi}{dx}$: MOC
$\frac{d\varphi}{dt}$: BDF
TDP
$\frac{d\varphi}{dx}$: MOC
$\frac{d\varphi}{dt}$: SDP
STC
$\frac{d\varphi}{dx}$: MOC
$\frac{d\varphi}{dt}$: MOC
Approximations in Space-Time
RBDC
$\frac{d\varphi}{dt} \approx \frac{d\bar{\varphi}}{dt}$
IBDC
$\frac{d\varphi}{dt} \approx \frac{1}{4\pi}\frac{d\phi}{dt}$
ISDP
$\frac{d\varphi}{dt} \approx f\left(\frac{dq}{dt}, \frac{d^2\varphi}{dt^2}\right)$
$\frac{dq}{dt}$: BDF
TSDP
$\frac{d\varphi}{dt} \approx f\left(\frac{dq}{dt}\right)$
$\frac{dq}{dt}$: BDF
ASTC
$\frac{d\varphi}{dt} \approx f\left(\frac{dq}{dt}\right)$
Equivalent

Figure 4-1. Relationship between the time-dependent MOC methods in this thesis

After the derivations there is a brief discussion on the application of the BDC equations to solve time-dependent neutron transport. This section places emphasis on the similarity of the BDC algorithm to the SSC algorithm.

4.1 Time-Dependent Boltzmann Transport Equation

The time-dependent Boltzmann transport equation can be written as:

$$\begin{aligned}\frac{1}{v(E)}\frac{d\varphi(\mathbf{x},\mathbf{\Omega},E,t)}{dt}&\\ &= -\mathbf{\Omega}\cdot\nabla\varphi(\mathbf{x},\mathbf{\Omega},E,t)-\Sigma_t(\mathbf{x},E,t)\varphi(\mathbf{x},\mathbf{\Omega},E,t)\\ &+q(\mathbf{x},\mathbf{\Omega},E,t),\end{aligned} \tag{4.1}$$

where the source q includes fission, scattering, and delayed neutrons:

$$\begin{aligned}q(\mathbf{x},\mathbf{\Omega},E,t) = (1-\beta)\frac{\chi_p(\mathbf{x},E)}{4\pi}\int\limits_0^\infty\int\limits_{4\pi}\nu\Sigma_f(\mathbf{x},E',t)\varphi(\mathbf{x},\mathbf{\Omega}',E',t)d\mathbf{\Omega}'dE'\\ +\int\limits_0^\infty\int\limits_{4\pi}\Sigma_s(\mathbf{x},\mathbf{\Omega}'\cdot\mathbf{\Omega},E'\to E,t)\varphi(\mathbf{x},\mathbf{\Omega}',E',t)d\mathbf{\Omega}'dE'\\ +\frac{S_d(\mathbf{x},E,t)}{4\pi},\end{aligned} \tag{4.2}$$

and where the delayed neutron source S_d is the total contribution of the delayed neutron precursors[12]:

$$S_d(\mathbf{x},E,t) = \sum_i \chi_{d,i}(\mathbf{x},E)\lambda_i C_i(\mathbf{x},t), \tag{4.3}$$

where i is the index for each delayed neutron precursor, and where the delayed neutron precursor concentration is described by the differential equation:

$$\frac{dC_i(\mathbf{x},t)}{dt} = \int_0^\infty\int_{4\pi}\frac{\beta_i\nu\Sigma_f(\mathbf{x},E',t)}{k_{\text{eff}}}\varphi(\mathbf{x},\mathbf{\Omega}',E',t)d\mathbf{\Omega}'\,dE' - \lambda_i C_i(\mathbf{x},t), \tag{4.4}$$

and where the variables are defined as for SSC in the previous chapter except:

$t =$ time

$v =$ neutron velocity

$\chi_p =$ prompt neutron fission spectrum

[12] Delayed neutrons are emitted when certain fission products decay into a daughter nuclei with excitation energy in excess of the neutron binding energy. These daughters may emit neutrons which were delayed by the radioactive decay of the parent fission product. While the term "precursor" may be used to indicate the parent irrespective of whether it decays by neutron emission, since we are concerned about reactor kinetics, we will define "precursor" to only include those daughters that actually emit neutrons.

β = total delayed neutron fraction

S_d = total delayed neutron source

$\chi_{d,i}$ = delayed neutron spectrum for precursor i

λ_i = decay constant for delayed neutron precursor i

C_i = delayed neutron precursor concentration for i

β_i = delayed neutron precursor yield for precursor i.

4.2 Approximations to the Neutron Transport Equation for BDC

We employ several approximations to solve the time-dependent neutron transport equation using MOC. Many of these approximations are similar or identical to those applied to SSC, except that they now must address the angular flux time derivative, delayed neutron source, and delayed neutron precursor differential equation. In this section we discuss each approximation to the time-dependent Boltzmann transport equation and illustrate how it is applied. We will first discuss approximations that are analogous to the approximations we applied in SSC. Then we will detail the approximations that are applied to this and the other time-dependent MOC methods in this work. Finally, we will cover the approximations that are unique to BDC methods.

4.2.1 Multi-group Approximation

The multi-group approximation is widely used in reactor kinetics. In general, the same neutron energy groups structures are applicable to time-dependent neutron transport as steady-state neutron transport, but we now also require group-averaged neutron velocities:

$$\frac{1}{v_g}\frac{d\varphi_g(\mathbf{x},\mathbf{\Omega},t)}{dt} = -\mathbf{\Omega}\cdot\nabla\varphi_g(\mathbf{x},\mathbf{\Omega},t) - \Sigma_{t,g}(\mathbf{x},t)\varphi_g(\mathbf{x},\mathbf{\Omega},t) + q_g(\mathbf{x},\mathbf{\Omega},t), \qquad 4.5$$

where the multi-group source is:

$$q_g(\mathbf{x},\mathbf{\Omega},t) = \sum_{g'=1}^{G}\left((1-\beta)\frac{\chi_{p,g}(\mathbf{x},t)}{4\pi k_{\text{eff}}}\int_{4\pi} \nu\Sigma_{f,g'}(\mathbf{x},t)\varphi_{g'}(\mathbf{x},\mathbf{\Omega}',t)d\mathbf{\Omega}' + \int_{4\pi} \Sigma_{s,g'\to g}(\mathbf{x},\mathbf{\Omega}'\cdot\mathbf{\Omega},t)\varphi_{g'}(\mathbf{x},\mathbf{\Omega}',t)d\mathbf{\Omega}'\right) + \frac{S_{d,g}(\mathbf{x},t)}{4\pi}, \tag{4.6}$$

and where the delayed neutron source is:

$$S_{d,g}(\mathbf{x},t) = \sum_i \chi_{d,g,i}(\mathbf{x})\lambda_i C_i(\mathbf{x},t), \tag{4.7}$$

and where the fission source in the precursor equations is also converted to multi-group form:

$$\frac{dC_i(\mathbf{x},t)}{dt} = \sum_{g'=1}^{G}\int_{4\pi} \beta_i \nu\Sigma_{f,g'}(\mathbf{x},t)\varphi_{g'}(\mathbf{x},\mathbf{\Omega}',t)d\mathbf{\Omega}' - \lambda_i C_i(\mathbf{x},t), \tag{4.8}$$

4.2.2 Isotropic Source Approximation

As in SSC, we will apply the isotropic source approximation to BDC. This includes assuming that the fission source is isotropic and applying the transport-corrected scattering approximation so that the scattering source is isotropic:

$$\frac{1}{v_g}\frac{d\varphi_g(\mathbf{x},\mathbf{\Omega},t)}{dt} = -\mathbf{\Omega}\cdot\mathbf{\nabla}\varphi_g(\mathbf{x},\mathbf{\Omega},t) - \Sigma_{tr,g}(\mathbf{x},t)\varphi_g(\mathbf{x},\mathbf{\Omega},t) + q_g(\mathbf{x},t), \tag{4.9}$$

where the multi-group total source is:

$$q_g(\mathbf{x},t) = \frac{1}{4\pi}\sum_{g'=1}^{G}\left[\frac{(1-\beta)\chi_{p,g}(\mathbf{x})\nu\Sigma_{f,g'}(\mathbf{x},t)}{k_{\text{eff}}}\phi_{g'}(\mathbf{x},t) + \Sigma_{s,g'\to g}^{tr}(\mathbf{x},t)\phi_{g'}(\mathbf{x},t)\right] + \frac{S_{d,g}(\mathbf{x},t)}{4\pi}. \tag{4.10}$$

By assuming that the fission source is isotropic, we eliminate the angular integral in the delayed neutron precursor equation:

$$\frac{dC_i(\mathbf{x},t)}{dt} = \sum_{g'=1}^{G} \beta_i \nu \Sigma_{f,g'}(\mathbf{x},t)\phi_{g'}(\mathbf{x},t) - \lambda_i C_i(\mathbf{x},t). \qquad 4.11$$

4.2.3 Discrete Ordinates Approximation

We will also use the discrete ordinates approximation for BDC. In general, the same quadrature sets that are used for SSC are applicable to BDC. The time-dependent transport equation with the discrete ordinates approximation is simply:

$$\begin{aligned} \frac{1}{v_g}\frac{d\varphi_g\left(\mathbf{x},\mathbf{\Omega}_{l,m},t\right)}{dt} \\ &= -\mathbf{\Omega}\cdot\nabla\varphi_g\left(\mathbf{x},\mathbf{\Omega}_{l,m},t\right) - \Sigma_{tr,g}(\mathbf{x},t)\varphi_g\left(\mathbf{x},\mathbf{\Omega}_{l,m},t\right) \\ &+ q_g(\mathbf{x},t). \end{aligned} \qquad 4.12$$

4.2.4 Delayed Neutron Group Approximation

Next we apply the delayed neutron group approximation which was discussed in Chapter 2:

$$S_{d,g}(\mathbf{x},t) = \sum_j \chi_{d,g,j}(\mathbf{x})\lambda_j C_j(\mathbf{x},t), \qquad 4.13$$

where *j* is the index for the delayed neutron groups, and the precursor equations are also given in terms of the delayed neutron groups:

$$\frac{dC_j(\mathbf{x},t)}{dt} = \sum_{g'=1}^{G} \beta_j \nu \Sigma_{f,g'}(\mathbf{x},t)\phi_{g'}(\mathbf{x},t) - \lambda_j C_j(\mathbf{x},t). \qquad 4.14$$

4.2.5 Characteristic Transform

Next we perform the characteristic transform to rewrite the spatial derivatives as a single derivative along a characteristic. As with SSC—but in contrast to space-time neutron transport methods—this transform is defined with respect to the spatial dimensions only; as a result the time derivative will remain in our characteristic equation.

The definition of the characteristic is unchanged from SSC:

$$\frac{dx}{ds} = \sqrt{1-\mu_m^2}\cos\alpha_l$$
$$\frac{dy}{ds} = \sqrt{1-\mu_m^2}\sin\alpha_l \qquad 4.15$$
$$\frac{dz}{ds} = \mu_m,$$

where again the spatial dimension along the characteristic is s.

When we apply the characteristic transform to the time-dependent neutron transport equation, the result is similar to the steady-state neutron transport equation but with an angular flux time derivative along the characteristic:

$$\begin{aligned}\frac{1}{v_g}\frac{d\varphi_g(\mathbf{x_0}+s\mathbf{\Omega}_{l,m},\mathbf{\Omega}_{l,m},t)}{dt} \\ = -\frac{d\varphi_g(\mathbf{x_0}+s\mathbf{\Omega}_{l,m},\mathbf{\Omega}_{l,m},t)}{ds} \\ -\Sigma_{tr,g}(\mathbf{x_0}+s\mathbf{\Omega}_{l,m},t)\varphi_g(\mathbf{r_0}+s\mathbf{\Omega}_{l,m},\mathbf{\Omega}_{l,m},t) \\ +q_g(\mathbf{x_0}+s\mathbf{\Omega}_{l,m},t).\end{aligned} \qquad 4.16$$

For succinctness we will introduce the index k for each unique characteristic:

$$\frac{1}{v_g}\frac{d\varphi_g^k(s,t)}{dt} = -\frac{d\varphi_g^k(s,t)}{ds} - \Sigma_{tr,g}^k(s,t)\varphi_g^k(s,t) + q_g^k(s,t), \qquad 4.17$$

where $k = f(\mathbf{\Omega}_{l,m},\mathbf{x_0})$.

4.2.6 Spatial Discretization and Step Characteristics

As in SSC, we discretize our spatial domain into fine spatial regions. Within these regions we assume that there is no spatial dependence of the transport cross section and the neutron source; this approximation is called step characteristics.

When these approximations are applied to the characteristic equation, the result is similar to the steady-state step characteristic equation but with time dependence:

$$\frac{1}{v_g}\frac{d\varphi_g^k(s,t)}{dt} = -\frac{d\varphi_g^k(s,t)}{ds} - \Sigma_{tr,g}^r(t)\varphi_g^k(s,t) + q_g^r(t), \qquad 4.18$$

where r is the index for the spatial regions.

4.2.7 Time Discretization

Next we discretize the time-dependent step characteristic equation in time. The equation is evaluated at time t_n:

$$\frac{1}{v_g}\frac{d\varphi_g^k(s,t_n)}{dt} = -\frac{d\varphi_g^k(s,t_n)}{ds} - \Sigma_{tr,g}^r(t_n)\varphi_g^k(s,t_n) + q_g^r(t_n). \tag{4.19}$$

The transient will be modeled by solving the time-discretized equation at discrete time points separated by time steps Δt_n which may vary in duration. For succinctness we will indicate the time using the index n:

$$\frac{1}{v_g}\frac{d\varphi_g^{k,n}(s)}{dt} = -\frac{d\varphi_g^{k,n}(s)}{ds} - \Sigma_{tr,g}^{r,n}\varphi_g^{k,n}(s) + q_g^{r,n}. \tag{4.20}$$

4.2.8 Approximation of Angular Flux Time Derivative with BDF

We approximate the angular flux time derivative using BDF as the time integration method. BDF is a variable-order, implicit, linear multi-step method for time integration suitable for solving stiff differential equations. The first-order BDF is equivalent to backward Euler method. BDF is derived in Chapter 2.

In BDF, the time derivative is approximated using the weighted sum of the solution variable at the current time point along with one or more previous time points, i.e.:

$$\frac{d\varphi_g^{k,n}(s)}{dt} \approx \sum_{i=0}^{O_{BDF}} A^i \varphi_g^{k,n-i}(s), \tag{4.21}$$

where the coefficients A^i depend upon the order of the BDF method and size of the time steps between each time point.

Since BDF is an implicit method, time t_n corresponds to the time at which we are solving the transport equation. Previous time points (e.g. t_{n-1}, $t_{n-2,}$ …) correspond to earlier points at which the transport equation was solved.

The number of coefficients and neutron fluxes that are used for BDF depends on the desired order of accuracy; the number of previous fluxes that needs to be stored is equal to the order of accuracy. For example, for first-order BDF (i.e. BDF1) only one previous flux is required:

$$\left(\frac{d\varphi_g^{k,n}(s)}{dt}\right)_{\mathrm{BDF1}} = \sum_{i=0}^{O_{BDF}=1} A^i \varphi_g^{k,n-i}(s) = \frac{1}{\Delta t_n}\varphi_g^{k,n}(s) + \frac{-1}{\Delta t_n}\varphi_g^{k,n-1}(s), \qquad 4.22$$

where $A^0 = \frac{1}{\Delta t_n}$ and $A^1 = \frac{-1}{\Delta t_n}$ for BDF1.

When BDF is applied to the time-discretized neutron transport equation, the angular flux time derivative is replaced by the weighted sum of angular fluxes from previous points in time:

$$\frac{1}{v_g}\sum_{i=0}^{O_{BDF}} A^i \varphi_g^{k,n-i}(s) = -\frac{d\varphi_g^{k,n}(s)}{ds} - \Sigma_{tr,g}^{r,n}\varphi_g^{k,n}(s) + q_g^{r,n}, \qquad 4.23$$

where we have avoided combining the first angular flux term in the summation with the transport term because of the approximation described in the following section.

4.2.9 Approximation of the Spatial Dependence of the Time Derivative Terms

Formally, the angular flux terms in the time derivative should be represented by the spatially-dependent segment-wise angular flux, which is the solution variable for MOC. Since the MOC equations describe the spatially-dependent angular flux, we could in principle store enough data to exactly represent the spatial dependence. However, we show in Appendix C that this requires storing in memory segment-wise quantities for every previous time step, resulting in a linear increase in the memory requirements of the method with the number of time steps modeled.

In fact for existing computers, just storing a single value for the angular flux at each segment (e.g. the segment-wise average angular flux) for one previous point in time is already prohibitively memory-intensive for large reactor models; maintaining even low-order approximations of the spatial dependence of the angular flux requires even more memory. In light of this, as a practical necessity we will approximate the angular flux terms at previous points in time in the angular flux time derivative with the segment-wise average angular flux.

Although we might elect to only approximate the previous terms in the time derivative while treating the new angular flux term implicitly (as many time-dependent neutron transport codes have done) in doing so we would needlessly introduce additional error and ensure that the approximation of the angular flux time derivative is not consistent. Taking that into account, we approximate all terms in the time derivative using the segment-wise average angular flux:

$$\sum_{i=0}^{O_{BDF}} A^i \varphi_g^{k,n-i}(s) \approx \sum_{i=0}^{O_{BDF}} A^i \bar{\varphi}_g^{k,r,n-i}, \qquad 4.24$$

where $\bar{\varphi}_g^{k,r,n}$ is the segment-wise average angular flux for characteristic k within region r at time t_n.

With this approximation, the characteristic equation recovers a form similar to the SSC but with an additional source term based on the angular flux time derivative:

$$\frac{d\varphi_g^{k,n}(s)}{ds} + \Sigma_{tr,g}^{r,n}\varphi_g^{k,n}(s) = q_g^{r,n} - \frac{1}{v_g}\sum_{i=0}^{O_{BDF}} A^i \bar{\varphi}_g^{k,r,n-i}. \qquad 4.25$$

4.2.10 Analytic Precursor Integration

Next we apply the second-order (quadratic) analytic precursor integration technique to treat the delayed neutrons [Hur08, Joo98]. The analytic precursor technique is derived in Chapter 2. The delayed neutron source is rewritten in terms of the integrated delayed neutron source from previous time steps $\tilde{S}_{d,g}^{r,<n}$ and the fission source at the current time step:

$$S_{d,g}^{r,n} = \tilde{S}_{d,g}^{r,<n} + \chi_{d,g}\omega^{r,n}\sum_{g'}\frac{\nu\Sigma_{f,g'}^{r,n}}{k_{\text{eff}}}\phi_{g'}^{r,n}, \qquad 4.26$$

where the cumulative source and coefficients for quadratic precursor integration are defined as:

$$
\begin{gathered}
\tilde{S}_{d,g}^{r,<n} = \chi_{d,g}\left[\sum_j \kappa_j^n \lambda_j C_j^{r,n-1} + \sum_j \beta_j^r \sum_{i=1}^{2}\left(D_j^{n-i}\sum_{g'}\frac{\nu\Sigma_{f,g'}^{r,n-i}}{k_{\text{eff}}}\phi_{g'}^{r,n-i}\right)\right] \\
\omega^{r,n} = \sum_j \beta_j^r D_j^n \\
D_j^{n-2} = \frac{1}{\lambda_j \Delta t_{n-1}(\gamma^n + 1)}\left[\frac{2\left(1-\kappa_j^n\right)}{\lambda_j \Delta t_{n-1}} - \kappa_j^n\left(\kappa_j^n + 1\right)\right] \\
D_j^{n-1} = \frac{1}{\lambda_j \Delta t_{n-1}}\left[\kappa_j^n + 1 + \frac{1-\kappa_j^n}{\gamma^n}\left(1 - \frac{2}{\lambda_j \Delta t_{n-1}}\right)\right] - \kappa_j^n \\
D_j^n = 1 - \frac{2}{(\gamma^n + 1)\lambda_j \Delta t_{n-1}} + \frac{1-\kappa_j^n}{\gamma^n(\gamma^n+1)\lambda_j \Delta t_{n-1}}\left(\frac{2}{\lambda_j \Delta t_{n-1}} - 1\right) \\
\kappa_j^n = e^{-\lambda_j \Delta t_n} \\
\gamma^n = \frac{\Delta t_n}{\Delta t_{n-1}}.
\end{gathered}
\tag{4.27}
$$

After applying the quadratic analytic precursor integration approximation, the source can be rewritten without dependence on the precursor concentration at the present time step:

$$
\begin{aligned}
q_g^{r,n} = \frac{1}{4\pi}\sum_{g'=1}^{G}&\left[\left(\chi_{p,g}^r(1-\beta) + \chi_{d,g}^r \omega^{r,n}\right)\frac{\nu\Sigma_{f,g'}^{r,n}}{k_{\text{eff}}}\phi_{g'}^{r,n} + \Sigma_{s,g'\to g}^{tr,r,n}\phi_{g'}^{r,n}\right] \\
&+ \frac{\tilde{S}_{d,g}^{r,<n}}{4\pi}.
\end{aligned}
\tag{4.28}
$$

4.2.11 Other Assumptions and Approximations

There are a few other assumptions and approximations in BDC that warrant mentioning. First, as in SSC, we have assumed that the Boltzmann equation adequately represents the behavior of the neutrons; there is neither so few neutrons in the domain that statistical variations are important nor so many that they significantly alter the material composition.

In SSC, we assumed that the problem was in a critical, steady-state configuration; in BDC, we assume that the transient begins from a critical configuration. Even if the model specifications should result in a critical reactor, simulation may not be exactly critical due to errors in the model. Thus the initial critical condition is enforced by dividing the neutron source through by the steady-state k-eigenvalue throughout the transient. This is a common practice in nuclear reactor kinetics.

We assumed that the transients begin from a critical configuration because this is representative of the transients of interest for reactor kinetics. If the transient begins from a dynamic or non-critical state, the methods would require modification.

We have also assumed that the delayed neutron precursors are not mobile; rather the delayed neutrons are produced at the sight of the fission event. The effects of mobile delayed neutron precursors cannot be accurately captured in the neutron transport equation. As a result, this assumption is essentially universally applied for reactor kinetics for solid fuel reactor systems.

We assumed that the delayed neutron group parameters are independent of position. This is generally a reasonable assumption for problems where there is one principal fissioning isotope. While it was not formally necessary for the derivation, it simplifies the notation and is applicable for the test problems considered in this work.

We should also note that one of the consequences of using a conventional time integration method to represent the angular flux time derivative is that we cannot capture neutron time-of-flight. In principal, with these methods (e.g. backward Euler, BDF, Runge-Kutta, etc.) a neutron source in one part of the domain can influence the angular flux in another part of the domain that is outside of the range of a neutron over the duration of the time step. In practice this is not a problem in reactor kinetics for typical time steps because the range of even the slowest neutrons is many neutron mean free paths, so the impact of this approximation is limited. Neutron time-of-flight could be preserved for neutron transport if space-time transport methods are used, but they are even more computationally- and memory-intensive than BDC.

4.3 Solution of the Step Characteristic Equation - RBDC

In this chapter, we provide derivations for two BDC methods: a reference method where the segment-wise angular flux is used for the angular flux time derivative (RBDC) and a low-order approximate method where the region-wise scalar flux is used for the angular flux time derivative (IBDC). While the derivations for these two methods have been identical up to this point, it is now necessary to differentiate between the two. We will first provide the derivation for RBDC.

As for SSC, the RBDC step characteristic equation can be analytically solved along the segment using an integrating factor of $e^{\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}$:

$$\frac{d}{ds}\left(\varphi_g^{k,n}(s)e^{\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}\right)=\left(q_g^{r,n}-\frac{1}{v_g}\sum_{i=0}^{O_{BDF}}A^i\bar{\varphi}_g^{k,r,n-i}\right)e^{\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}. \tag{4.29}$$

The solution of the equation along the segment is the spatially-dependent segment-wise angular flux at time t_n:

$$\begin{aligned}\varphi_g^{k,n}(s)=\varphi_{g,0}^{k,r,n}&e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}\\&+\left(q_g^{r,n}-\frac{1}{v_g}\sum_{i=0}^{O_{BDF}}A^i\bar{\varphi}_g^{k,r,n-i}\right)\left(\frac{1-e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{\Sigma_{tr,g}^{r,n}}\right),\end{aligned} \tag{4.30}$$

where $\varphi_{g,0}^{k,r,n}\equiv\varphi_g^{k,n}\left(s_0^{k,r}\right)$.

The solution to the RBDC equation is similar to the solution of the SSC equation, except that delayed neutrons are treated separately from prompt neutrons and the angular flux time derivative functions as an additional source term. As in SSC, this equation is used to propagate the angular flux along the characteristic from segment to segment, and to numerically integrate the region-wise scalar flux. In addition, the segment-wise average angular flux is stored for use treating the angular flux time derivative on subsequent time steps. In the following sections, we derive the equations for these quantities.

4.3.1 Angular Flux Propagation Along the Characteristic - RBDC

When we solved Equation 3.22, we assumed that we knew the incoming angular flux at the beginning of the segment $\varphi_{g,0}^{k,r,n}$. We can make this assumption for a general segment because we will propagate the angular fluxes along characteristics. This entails calculating the outgoing angular flux for each segment to serve as the incoming angular flux for the following segment.

To calculate the outgoing angular flux, we evaluate Equation 3.22 at the end of the segment:

$$\begin{aligned}\varphi_g^{k,n}\left(s_0^{k,r}+\Delta s^{k,r}\right) &\\ &= \varphi_{g,0}^{k,r,n} e^{-\Sigma_{tr,g}^{r,n}\Delta s^{k,r}} \\ &+ \left(q_g^{r,n} - \frac{1}{v_g}\sum_{i=0}^{O_{BDF}} A^i \bar{\varphi}_g^{k,r,n-i}\right)\left(\frac{1-e^{-\Sigma_{tr,g}^{r,n}\Delta s^{k,r,n}}}{\Sigma_{tr,g}^{r,n}}\right).\end{aligned} \tag{4.31}$$

This result is very similar to the analogous equation for SSC, but with the time derivative approximation as an additional source. Note that the time derivative includes the segment-wise average angular flux at time t_n; this is calculated in the next section.

4.3.2 Numerical Integration of the Region-wise Scalar Flux - RBDC

We also use Equation 3.22 to calculate the segment-wise average angular flux. As in SSC, this will be used to numerically integrate the region-wise scalar flux. In addition, the segment-wise average angular flux was used in the previous section to calculate the outgoing angular flux. Finally, keep in mind that we will need to store the segment-wise average angular flux for one or more time steps for use in representing the time derivative; as mentioned previously, this requires extremely large amounts of memory and is the primary disadvantage to RBDC.

To calculate the segment-wise average angular flux, we integrate the spatially-dependent segment-wise angular flux averaged over the segment length:

$$\begin{aligned}\bar{\varphi}_g^{k,r,n} &= \left(\varphi_{g,0}^{k,r,n} - \frac{q_g^{r,n}}{\Sigma_{tr,g}^{r,n}} + \frac{1}{\Sigma_{tr,g}^{r,n} v_g}\sum_{i=0}^{O_{BDF}} A^i \bar{\varphi}_g^{k,r,n-i}\right)\left(\frac{1-e^{-\Sigma_{tr,g}^{r,n}\Delta s^{k,r}}}{\Sigma_{tr,g}^{r,n}\Delta s^{k,r}}\right) \\ &+ \frac{q_g^{r,n}}{\Sigma_{tr,g}^{r,n}} - \frac{1}{\Sigma_{tr,g}^{r,n} v_g}\sum_{i=0}^{O_{BDF}} A^i \bar{\varphi}_g^{k,r,n-i}.\end{aligned} \tag{4.32}$$

However, Equation 4.32 gives the segment-wise average angular flux in terms of itself because the segment-wise average angular flux was used to represent the angular flux time derivative. With some algebra we can isolate the segment-wise average angular flux at time t_n:

$$\begin{aligned}\bar{\varphi}_g^{k,r,n} = \Bigg[&\left(\varphi_{g,0}^{k,r,n} - \frac{q_g^{r,n}}{\Sigma_{tr,g}^{r,n}} + \frac{1}{\Sigma_{tr,g}^{r,n} v_g} \sum_{i=1}^{O_{BDF}} A^i \bar{\varphi}_g^{k,r,n-i}\right)\left(\frac{1 - e^{-\Sigma_{tr,g}^{r,n} \Delta s^{k,r}}}{\Sigma_{tr,g}^{r,n} \Delta s^{k,r}}\right) \\ &+ \frac{q_g^{r,n}}{\Sigma_{tr,g}^{r,n}} - \frac{1}{\Sigma_{tr,g}^{r,n} v_g} \sum_{i=1}^{O_{BDF}} A^i \bar{\varphi}_g^{k,r,n-i}\Bigg] \\ &\times \left[1 + \frac{A^0}{\Sigma_{tr,g}^{r,n} v_g}\left(1 - \frac{1 - e^{-\Sigma_{tr,g}^{r,n} \Delta s^{k,r}}}{\Sigma_{tr,g}^{r,n} \Delta s^{k,r}}\right)\right]^{-1}.\end{aligned} \tag{4.33}$$

Note that, whereas the BDF sums in Equation 4.33 begin at the index 0, the sums in Equation 4.33 begin at the index 1 and thus only include the angular flux from previous points in time.

The region-wise scalar flux is calculated by numerically integrating the segment-wise average angular fluxes over space and angle as in SSC (Equations 3.26 and 3.27):

$$\phi_g^{r,n} = \sum_m w_m \sum_l w_l \frac{\sum_{k \in r} \bar{\varphi}_g^{k,r,n} \Delta s^{k,r} \Delta K_l}{\sum_{k \in r} \Delta s^{k,r} \Delta K_l}, \tag{4.34}$$

where the only difference between the time-dependent and steady-state equations for integrating the scalar flux is the presence of the time index.

4.4 Solution of the Step Characteristic Equation - IBDC

In this chapter we also solve the BDC equations using the scalar flux approximation. We begin by further approximating the angular flux time derivative in terms of the region-wise scalar flux:

$$\sum_{i=0}^{O_{BDF}} A^i \bar{\varphi}_g^{k,r,n-i} \approx \sum_{i=0}^{O_{BDF}} \frac{A^i \phi_g^{r,n-i}}{4\pi}. \tag{4.35}$$

We should note that once this approximation is applied, the solution is no longer expected to converge to the Boltzmann transport equation as the angular, spatial, and temporal discretizations are refined. This approximation avoids the requirement of storing the segment-wise average angular flux. The resulting equation can be solved using the integrating factor $e^{\Sigma_{tr,g}^{r,n} s}$:

$$\frac{d}{ds}\left(\varphi_g^{k,n}(s)e^{\Sigma_{tr,g}^{r,n}s}\right) = \left(q_g^{r,n} - \frac{1}{v_g}\sum_{i=0}^{O_{BDF}}\frac{A^i\phi_g^{r,n-i}}{4\pi}\right)e^{\Sigma_{tr,g}^{r,n}s}. \qquad 4.36$$

The solution of the equation along the segment is the spatially-dependent segment-wise angular flux for IBDC:

$$\varphi_g^{k,n}(s) = \varphi_{g,0}^{k,r,n}e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)} + \left(q_g^{r,n} - \frac{1}{v_g}\sum_{i=0}^{O_{BDF}}\frac{A^i\phi_g^{r,n-i}}{4\pi}\right)\left(\frac{1-e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{\Sigma_{tr,g}^{r,n}}\right). \qquad 4.37$$

In IBDC Equation 4.37 is used to propagate the angular flux along characteristics and to calculate the segment-wise average angular flux for numerical integration into the region-wise scalar flux. This illustrates an obvious difficulty; Equation 4.37 is used to calculate the scalar flux at time t_n, but it assumes that the scalar flux is known to represent the angular flux time derivative. While it is possible to algebraically eliminate this dependency in a manner analogous to Equation 4.33 for RBDC, it requires a substantial modification of the MOC algorithm.

On the other hand, to start the iterations for MOC at each time step we require an estimate of the scalar flux at time t_n to calculate an initial estimate of the total neutron source; if we use this estimate to provide an initial guess for the angular flux time derivative, then we have enough information to solve Equation 4.37. Further, this estimate provides a substantial computational benefit; we can combine the time derivative term with the total source q into a single isotropic source defined on the region-wise mesh, i.e.:

$$\varphi_g^{k,n}(s) = \varphi_{g,0}^{k,r,n}e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)} + \hat{q}_g^{r,n}\left(\frac{1-e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{\Sigma_{tr,g}^{r,n}}\right). \qquad 4.38$$

The resulting equation is identical in form to the characteristic equation of SSC, so we could use the steady-state algorithm with minimal modification. In light of this, we will adopt this approach and assume that an estimate of the scalar flux at the end of the current time step is known and used to calculate the time derivative. MOC is an iterative process, and for each additional iteration we will use the scalar flux from the previous iteration as an estimate. Details on the calculation of the initial estimate of the scalar flux are provided in Chapter 7.

4.4.1 Angular Flux Propagation Along the Characteristic - IBDC

Equation 4.38 is used to propagate the angular flux along the characteristic by evaluating it at the end of the segment:

$$\varphi_g^{k,n}\left(s_0^{k,r} + \Delta s^{k,r}\right) = \varphi_{g,0}^{k,r,n} e^{-\Sigma_{tr,g}^{r,n} \Delta s^{k,r}} + \hat{q}_g^{r,n} \left(\frac{1 - e^{-\Sigma_{tr,g}^{r,n} \Delta s^{k,r,n}}}{\Sigma_{tr,g}^{r,n}} \right). \qquad 4.39$$

4.4.2 Numerical Integration of the Region-wise Scalar Flux - IBDC

We also use Equation 4.38 to calculate the region-wise scalar flux. The region-wise scalar flux is based on the segment-wise average angular flux:

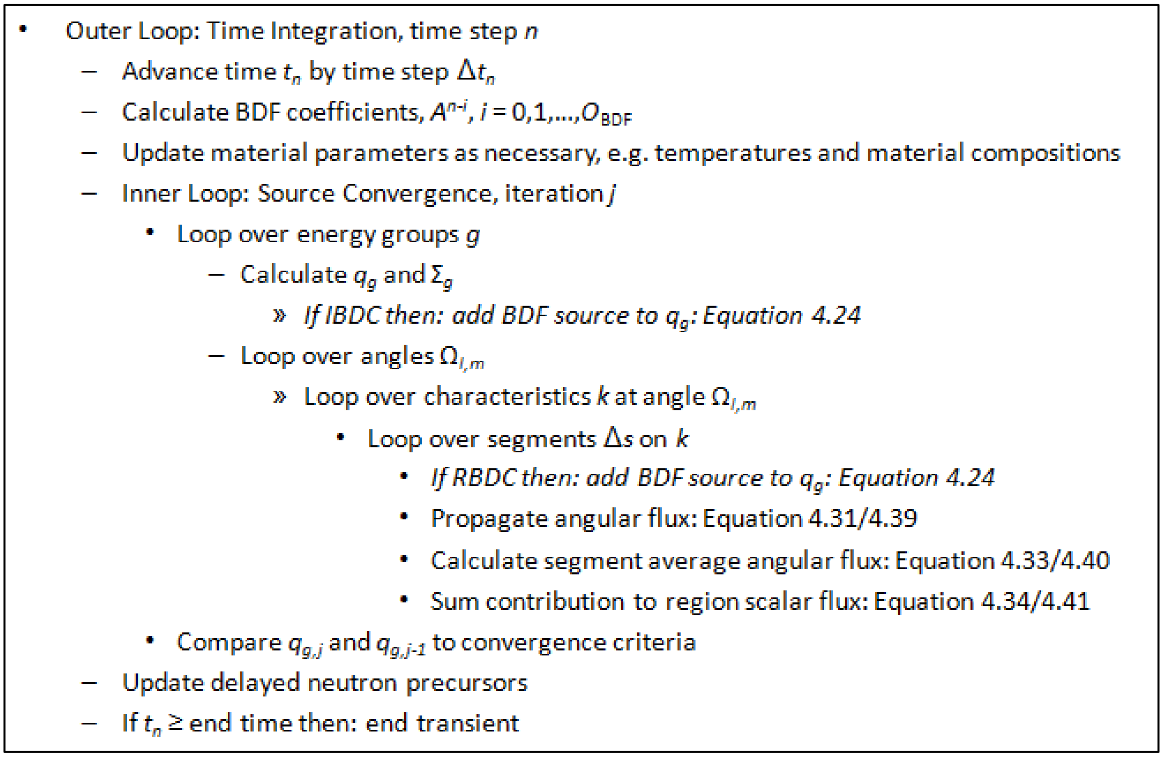
$$\bar{\varphi}_g^{k,r,n} = \left[\varphi_{g,0}^{k,r,n} - \frac{\hat{q}_g^{r,n}}{\Sigma_{tr,g}^{r,n}} \right] \left(\frac{1 - e^{-\Sigma_{tr,g}^{r,n} \Delta s^{k,r}}}{\Sigma_{tr,g}^{r,n} \Delta s^{k,r}} \right) + \frac{\hat{q}_g^{r,n}}{\Sigma_{tr,g}^{r,n}}. \qquad 4.40$$

As for RBDC, the region-wise scalar flux is calculated by numerically integrating the segment-wise average angular flux over volume and angle using Equation 4.34.

4.5 Summary of BDC and Algorithm

The algorithm for a single time step of BDC is similar to the algorithm for SSC. BDC begins from a steady-state solution which was determined using SSC. Some perturbation in the problem domain occurs (e.g. a material or temperature changes, which changes the cross sections in part of the problem), which causes the neutron angular flux to change over time.

For each time step, an initial estimate of the scalar flux at the end of the time step is determined (details for the calculation of the estimate using the computer code DeCART are provided in Chapter 7), and the scalar flux is used to estimate the source and the angular flux time derivative if IBDC is used. With the source estimate, the relevant equations are used to propagate the angular flux along characteristics from the problem boundary inward and to calculate the region-wise scalar flux. This process repeats until the solution has converged. If the current time is less than the end time of the transient, the method advances one time step and repeats the process. This algorithm is depicted in Figure 4-2.



```
• Outer Loop: Time Integration, time step $n$
  – Advance time $t_n$ by time step $\Delta t_n$
  – Calculate BDF coefficients, $A^{n-i}$, $i = 0,1,\ldots,O_{\mathrm{BDF}}$
  – Update material parameters as necessary, e.g. temperatures and material compositions
  – Inner Loop: Source Convergence, iteration $j$
      • Loop over energy groups $g$
          – Calculate $q_g$ and $\Sigma_g$
              » If IBDC then: add BDF source to $q_g$: Equation 4.24
          – Loop over angles $\Omega_{l,m}$
              » Loop over characteristics $k$ at angle $\Omega_{l,m}$
                  • Loop over segments $\Delta s$ on $k$
                      • If RBDC then: add BDF source to $q_g$: Equation 4.24
                      • Propagate angular flux: Equation 4.31/4.39
                      • Calculate segment average angular flux: Equation 4.33/4.40
                      • Sum contribution to region scalar flux: Equation 4.34/4.41
      • Compare $q_{g,j}$ and $q_{g,j-1}$ to convergence criteria
  – Update delayed neutron precursors
  – If $t_n \geq$ end time then: end transient
```

Figure 4-2. Nested algorithm for time-dependent MOC with BDF

We will analyze the error in the treatment of the time derivative for RBDC and IBDC in Chapter 6 and numerically test these methods in Chapter 8. Details of the implementation of the MOC methods in the computer code DeCART are provided in Chapter 7.

Chapter 5
Derivation of Time-Dependent MOC with Source-Derivative Propagation

5.1 Introduction

In Chapter 4, the time-dependent neutron transport equation was solved using method of characteristics (MOC) to treat the spatial derivatives and the backward difference formula (BDF) to treat the time derivatives. This time-dependent MOC method with BDF (BDC) method represents a conventional approach to solving the time-dependent neutron transport equation for nuclear reactor kinetics. However, it has a significant disadvantage in that storing the angular flux to represent the angular flux time derivative for the reference method (RBDC) requires excessive memory and limits the size and resolution of reactor kinetics problems that can be modeled. Chapter 4 also presents an approximate method (IBDC) which has modest memory requirements, but the isotropic approximation may not always be appropriate.

Because of these issues, we developed a new time-dependent MOC method. This method circumvents the angular flux storage through angular flux time derivative propagation (TDP) along the characteristics in space rather than using a local finite difference approximation for the derivative. To accomplish this, we derive a characteristic equation for the angular flux time derivative which is used alongside the characteristic equation for the angular flux to propagate both the angular flux and its time derivative along the characteristic. In much the same way that the angular flux equation in steady-state MOC (SSC) gives the angular flux in terms of the propagated effects of the neutron source terms along a characteristic, this TDP method gives the angular flux time derivative in terms of the propagated effects of the source time derivatives along a characteristic; this TDP method is thus called Source Derivative

Propagation (SDP). It requires substantially less memory to represent the source time derivatives than the angular flux time derivatives.

While one might identify other TDP methods that define the characteristic equation for the angular flux time derivative differently, this thesis will focus on SDP and other closely-related methods because they have favorable mathematical properties. For this method, we take a time derivative of the characteristic equation for the angular flux. This gives an equation for the angular flux time derivative in terms of the time derivative of the incoming angular flux, the source time derivative, and the second time derivative of the angular flux. If we approximate the second time derivative of the angular flux, we can calculate the first time derivative of the angular flux along the characteristic in terms of the first source derivatives along the characteristic.

The treatment of the second time derivative of the angular flux is an important consideration. While we could store the angular flux at previous points in time, this would require excessive memory and eliminate a major advantage of SDP. Instead we focus on two options for treating the second derivative of the angular flux in SDP: (1) truncating the second derivative (TSDP), or (2) assuming that it is isotropic (ISDP). Both approximations result in methods which require comparable memory and computational requirements to IBDC but are expected to better capture the angular dependence of the angular flux time derivative. Derivations of TSDP and ISDP are provided in this chapter.

The relationship between the SDP methods and the other time-dependent MOC methods in this thesis is represented graphically in Figure 5-1. This figure illustrates that all of the time-dependent MOC methods begin with the Boltzmann transport equation and apply a common set of assumptions. The methods differ in their treatment of the spatial and temporal derivatives. The figure also foreshadows some of the SDP methods that will be developed in this chapter.

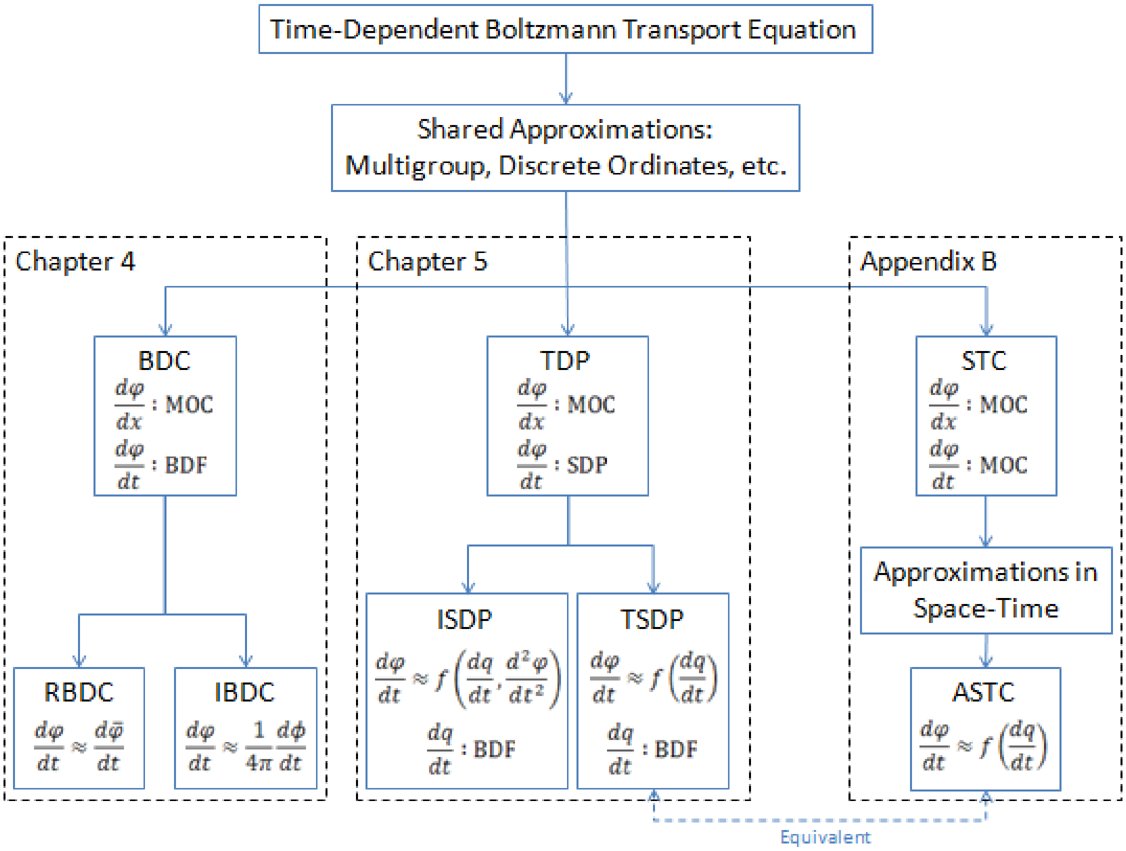

Time-Dependent Boltzmann Transport Equation
Shared Approximations: Multigroup, Discrete Ordinates, etc.
Chapter 4
Chapter 5
Appendix B
BDC
$\frac{d\varphi}{dx}$: MOC
$\frac{d\varphi}{dt}$: BDF
TDP
$\frac{d\varphi}{dx}$: MOC
$\frac{d\varphi}{dt}$: SDP
STC
$\frac{d\varphi}{dx}$: MOC
$\frac{d\varphi}{dt}$: MOC
Approximations in Space-Time
RBDC
$\frac{d\varphi}{dt} \approx \frac{d\bar{\varphi}}{dt}$
IBDC
$\frac{d\varphi}{dt} \approx \frac{1}{4\pi}\frac{d\phi}{dt}$
ISDP
$\frac{d\varphi}{dt} \approx f\left(\frac{dq}{dt}, \frac{d^2\varphi}{dt^2}\right)$
$\frac{dq}{dt}$: BDF
TSDP
$\frac{d\varphi}{dt} \approx f\left(\frac{dq}{dt}\right)$
$\frac{dq}{dt}$: BDF
ASTC
$\frac{d\varphi}{dt} \approx f\left(\frac{dq}{dt}\right)$
Equivalent

Figure 5-1. Relationship between SDP and other time-dependent MOC methods

The derivation for TSDP and ISDP use the assumption that the cross sections are changing slowly during the transient to drastically simplify the derivation. While this assumption is valid throughout most of the time and space domain within reactor transients, it is crucially invalid near regions were material compositions are changing (e.g. near moving control rods or voiding moderator). To address this limitation, we also developed a version of SDP which allows the cross sections to change linearly in time, which is called Source & Cross Section Derivative Propagation (SCDP). This method is slightly more complex than SDP, but it is only required when cross sections are changing quickly. Like SDP, SCDP results in a second angular flux derivative term which can either be truncated or assumed to be isotropic (ISCDP), but we will focus on the latter.

A General Formulation for Arbitrary Nth Source-Derivative Propagation

An alternative technique is to treat the second time derivative of the angular flux by also propagating it along the characteristic. If we define this equation using the approach taken for SDP, the result is a method of Second Source Derivative Propagation (2SDP). However, the characteristic equations for 2SDP gain a third derivative of the angular flux which requires similar consideration (e.g. a truncated or isotropic approximation). SDP and 2SDP can be understood as the first two members of a family of SDP methods which could be expanded to an arbitrarily-high number of source-derivatives. In this chapter, we will show how 2SDP is derived and provide characteristic equations for an arbitrary N^{th} source derivative propagation (NSDP) method. Figure 5-2 illustrates the relationship between SDP, 2SDP, and NSDP as a family of TDP methods. While these higher-derivative methods have unfavorable qualities for implementation as reactor kinetics methods, we present their derivation because they provide insight into the topic of SDP.

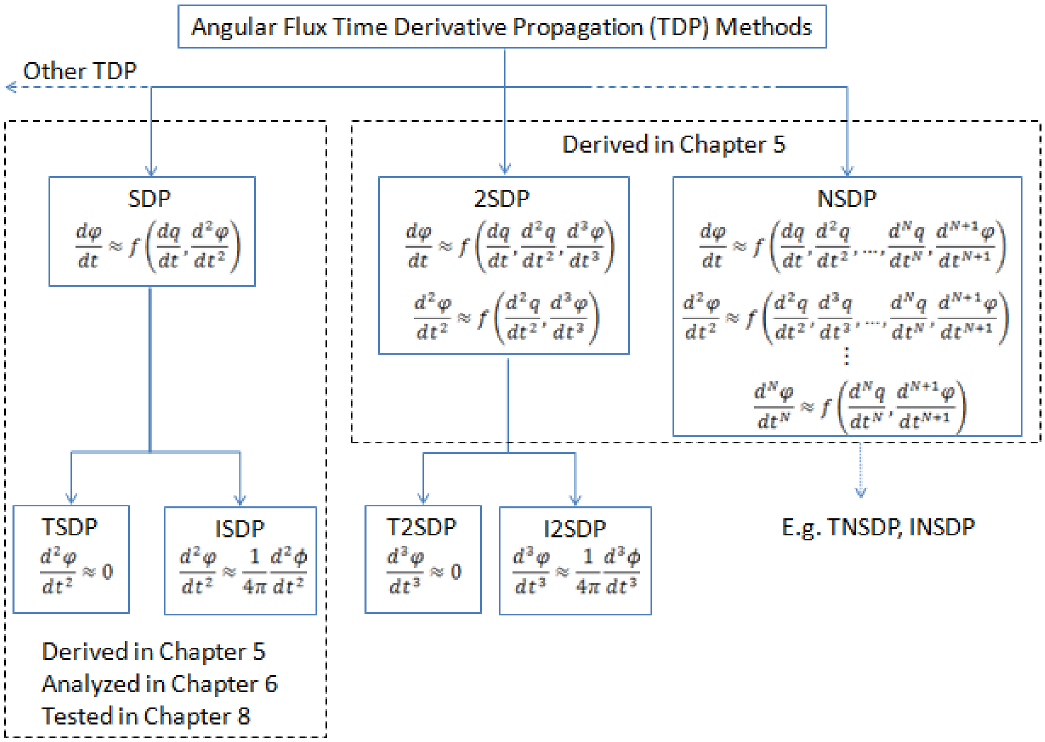

Angular Flux Time Derivative Propagation (TDP) Methods
Other TDP
Derived in Chapter 5
SDP
$\frac{d\varphi}{dt} \approx f\left(\frac{dq}{dt}, \frac{d^2\varphi}{dt^2}\right)$
2SDP
$\frac{d\varphi}{dt} \approx f\left(\frac{dq}{dt}, \frac{d^2q}{dt^2}, \frac{d^3\varphi}{dt^3}\right)$
$\frac{d^2\varphi}{dt^2} \approx f\left(\frac{d^2q}{dt^2}, \frac{d^3\varphi}{dt^3}\right)$
NSDP
$\frac{d\varphi}{dt} \approx f\left(\frac{dq}{dt}, \frac{d^2q}{dt^2}, \dots, \frac{d^Nq}{dt^N}, \frac{d^{N+1}\varphi}{dt^{N+1}}\right)$
$\frac{d^2\varphi}{dt^2} \approx f\left(\frac{d^2q}{dt^2}, \frac{d^3q}{dt^3}, \dots, \frac{d^Nq}{dt^N}, \frac{d^{N+1}\varphi}{dt^{N+1}}\right)$
$\vdots$
$\frac{d^N\varphi}{dt^N} \approx f\left(\frac{d^Nq}{dt^N}, \frac{d^{N+1}\varphi}{dt^{N+1}}\right)$
TSDP
$\frac{d^2\varphi}{dt^2} \approx 0$
ISDP
$\frac{d^2\varphi}{dt^2} \approx \frac{1}{4\pi}\frac{d^2\phi}{dt^2}$
T2SDP
$\frac{d^3\varphi}{dt^3} \approx 0$
I2SDP
$\frac{d^3\varphi}{dt^3} \approx \frac{1}{4\pi}\frac{d^3\phi}{dt^3}$
E.g. TNSDP, INSDP
Derived in Chapter 5
Analyzed in Chapter 6
Tested in Chapter 8

Figure 5-2. Relationship between SDP, 2SDP, and NSDP as a family of TDP methods

5.1.1 Chapter Overview

The derivations for the various SDP methods are the primary focus of this chapter. Before we begin the derivations, we will briefly discuss the history of the development of the TDP methods. Specifically, we explain the relationship between TSDP and the analytically-integrated space-time characteristic (ASTC) method we developed previously. Figure 5-1 illustrates this relationship.

As with Chapters 3 and 4, we begin the derivations by addressing the approximations to the time-dependent Boltzmann equation for these methods. Since many of the approximations are identical to those in the Chapter 4, we will briefly review those and then skip immediately to the time-discretized, multi-group, step characteristic equation with analytically integrated delayed neutron precursors.

Next we discuss two new approximations used for the SDP methods. The first concerns the derivation of the characteristic equation for the angular flux time derivative, while the second regards the treatment of the source time derivatives using backward differences. The latter is closely related to BDF, which is one reason why we use BDF for time integration for the BDC methods described in Chapter 4.

After we establish the approximations, we derive the solutions to the characteristic equation for TSDP and ISDP. While these two methods are only a narrow subset of TDP methods, they are the primary focus of this dissertation because they are efficient and accurate throughout most of the space and time domain of the reactor transients we modeled.

Next we derive the characteristic equations for ISCDP. This derivation is similar to that of ISDP, but it permits the cross sections to vary over the time step. This method is slightly more computationally-expensive than the SDP methods, but it may be important when cross sections are changing quickly. When cross sections are not changing, this method is identical to ISDP, and the ISDP algorithm can be used for time steps and in parts of the reactor where cross sections are not changing quickly to enhance computational efficiency.

In the next section we build on the derivation for SDP to derive the characteristic equations for 2DSP, which are similar to SDP but involve the propagation of the second time derivative of the angular flux in addition to the first. Still higher derivative methods can be derived by repeating the process, resulting in systematic changes to the characteristic equations. Based on these systematic changes, we provide characteristic equations for an arbitrary N^{th} derivative SDP method.

Finally, this chapter will provide a summary of the implementation of the propagated source methods, paying special attention to how the method differs from the BDC methods in the Chapter 4.

5.2 History of TDP Development and Relationship with ASTC

The TDP concept was discovered while analyzing a time-dependent MOC methods based on space-time MOC, where the characteristics are defined not only in space but also time. Space-time transport methods are generally more computationally and memory intensive than time-discretized transport methods that store the angular flux (e.g. RBDC). In light of the memory issues of RBDC, a space-time method would seem a step in the wrong direction, but when further approximations are applied in the time domain the method is drastically simplified. This approximation allows the many discrete space-time characteristics to be replaced by a single function which represents the space-time characteristics within a time step. This function can be solved to determine the angular flux along a spatial segment at the end of the time step, which corresponds to the spatial segments employed by BDC and propagated source-derivative methods. Consequently, the method simplifies to one that is comparable to IBDC in computational and memory requirements.

This Analytically-integrated Space-Time Characteristic (ASTC) method provides several advantages. First, it provides a basis for the propagation of the angular flux time derivative in space along characteristics, avoiding the need to store the angular flux to represent the time derivative. Second, it recasts the angular flux time derivative in terms of source time derivatives, which can be represented to a high order using backward differences in time without requiring excessive memory. We derive the ASTC method in Appendix B.

While investigating ASTC to understand its error, an equivalent derivation was identified which does not require the definition of characteristics in space-time. There are several advantages to this derivation: (a) it is much less complex than the space-time derivation, (b) it is more general than the space-time derivation, (c) it clearly identifies the leading error terms, and (d) the error terms suggest options for refinement. TSDP is the equivalent derivation to ASTC, while ISDP is similar but with a correction term. For the reasons listed above, the SDP methods are the focus of this chapter rather than ASTC.

5.3 Approximations to the Boltzmann Transport Equation for SDP

5.3.1 Approximations Shared with BDC

The derivation of SDP begins with the time-dependent Boltzmann transport equation and applies many of the approximations that were used when deriving BDC in Chapter 4. These include: the multi-group approximation, the isotropic source approximation, the discrete ordinates approximation, the delayed neutron group approximation, the spatial characteristic transform, step characteristics, and analytic precursor integration. After these approximations are applied, the result is the time-discretized, multi-group, step characteristics equation with analytic precursor integration:

$$\frac{1}{v_g}\frac{d\varphi_g^{k,n}(s)}{dt} = -\frac{d\varphi_g^{k,n}(s)}{ds} - \Sigma_{tr,g}^{r,n}\varphi_g^{k,n}(s) + q_g^{r,n}, \qquad 5.1$$

where the source includes fission, scattering, and analytically-integrated precursors:

$$q_g^{r,n} = \frac{1}{4\pi}\sum_{g'=1}^{G}\left[\left(\chi_{p,g}^{r}(1-\beta) + \chi_{d,g}^{r}\omega^{r,n}\right)\frac{\nu\Sigma_{f,g'}^{r,n}}{k_{\text{eff}}}\phi_{g'}^{r,n} + \Sigma_{s,g'\rightarrow g}^{tr,r,n}\phi_{g'}^{r,n}\right] + \frac{\tilde{S}_{d,g}^{r,<n}}{4\pi}. \qquad 5.2$$

As in Chapter 4, we assumed that the transient is beginning from a steady-state, critical configuration. This is representative of the transients of interest in nuclear reactor kinetics, and if this is not true the SDP methods will require modification.

5.3.2 Propagation of the Angular Flux Time Derivative

First, we will derive a characteristic equation for the spatially-dependent segment-wise angular flux time derivative in terms of the angular flux time derivative at the incoming position and the source time derivative along the segment. To derive this equation, we will assume that the angular flux time derivative is known, and group it with the neutron source:

$$\frac{d\varphi_g^{k,n}(s)}{ds} + \Sigma_{tr,g}^{r,n}\varphi_g^{k,n}(s) = q_g^{r,n} - \frac{1}{v_g}\frac{d\varphi_g^{k,n}(s)}{dt}. \qquad 5.3$$

Next we apply the integrating factor $e^{\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}$:

$$\frac{d}{ds}\left(e^{\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}\varphi_g^{k,n}(s)\right) = \left(q_g^{r,n} - \frac{1}{v_g}\frac{d\varphi_g^{k,n}(s)}{dt}\right)e^{\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}. \qquad 5.4$$

We can partially solve for the segment-wise angular flux by integration:

$$\int_{s_0^{k,r}}^{s}\frac{d}{ds}\left(e^{\Sigma_{tr,g}^{r,n}\left(s'-s_0^{k,r}\right)}\varphi_g^{k,n}(s')\right)ds'$$
$$= \int_{s_0^{k,r}}^{s}\left(q_g^{r,n} - \frac{1}{v_g}\frac{d\varphi_g^{k,n}(s')}{dt}\right)e^{\Sigma_{tr,g}^{r,n}\left(s'-s_0^{k,r}\right)}ds'. \qquad 5.5$$

The integral can be evaluated for every term except for the time derivative:

$$\varphi_g^{k,n}(s)e^{\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)} - \varphi_{g,0}^{k,r,n}$$
$$= q_g^{r,n}\left(\frac{e^{\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}-1}{\Sigma_{tr,g}^{r,n}}\right) - \frac{1}{v_g}\int_{s_0^{k,r}}^{s}\frac{d\varphi_g^{k,n}(s')}{dt}e^{\Sigma_{tr,g}^{r,n}\left(s'-s_0^{k,r}\right)}ds', \qquad 5.6$$

where $\varphi_{g,0}^{k,r,n} \equiv \varphi_g^{k,n}\left(s_0^{k,r}\right)$.

With some algebra the spatially-dependent segment-wise angular flux is isolated giving an equation for the angular flux along the segment:

$$\varphi_g^{k,n}(s) = \varphi_{g,0}^{k,r,n} e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)} + q_g^{r,n}\left(\frac{1-e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{\Sigma_{tr,g}^{r,n}}\right) - \frac{e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{v_g}\int_{s_0^{k,r}}^{s}\frac{d\varphi_g^{k,n}(s')}{dt}e^{\Sigma_{tr,g}^{r,n}\left(s'-s_0^{k,r}\right)}ds'. \qquad 5.7$$

This is a partial solution of the BDC equations; if we approximate the angular flux time derivative in Equation 5.7 with the segment-wise average angular flux or the region-wise scalar flux the result will be the solution to the RBDC or IBDC equations respectively. If the angular flux is not changing, the result is identical to SSC.

In contrast to BDC, to derive the characteristic equation for the angular flux time derivative, we will take a time derivative of Equation 5.7 with the assumption that the transport cross section is changing slowly with time near time t_n (i.e. $\frac{d}{dt}\Sigma_{tr,g}^{r,n} \approx 0$)[13]:

$$\frac{d\varphi_g^{k,n}(s)}{dt} = \frac{d\varphi_{g,0}^{k,r,n}}{dt} e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)} + \frac{dq_g^{r,n}}{dt}\left(\frac{1-e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{\Sigma_{tr,g}^{r,n}}\right) - \frac{e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{v_g}\int_{s_0^{k,r}}^{s}\frac{d^2\varphi_g^{k,n}(s')}{dt^2}e^{\Sigma_{tr,g}^{r,n}\left(s'-s_0^{k,r}\right)}ds'. \qquad 5.8$$

This results in an equation for the spatially-dependent segment-wise angular flux time derivative which will be used for the SDP methods. The angular flux time derivative is given in terms of the incoming angular flux time derivative at the beginning of the segment, a source time derivative within the region, and a second angular flux time derivative. Note that if this approximation is not valid, the solution of this method will not converge to the solution of the Boltzmann equation in the limit as the time step goes to zero.

For each segment, the incoming angular flux time derivative is calculated using the outgoing angular flux time derivative from the previous segment as calculated using

[13] In Section 5.6 we derive SDP while assuming that the transport cross section varies linearly in time (i.e. SCDP), which results in a minor modification to the characteristic equations.

Equation 5.8. The source time derivative is approximated using backward differences in time, which is described in the following section. Two options for the treatment of the second angular flux derivative are described in the following sections: (1) truncating the term (TSDP) or (2) assuming the term is isotropic (ISDP). In both cases, the net result is that the angular flux time derivative is effectively represented by a source time derivative or a modified source time derivative whose effects are propagated in space along the characteristics.

To derive the SDP characteristic equation for the angular flux, we apply Equation 5.8 to Equation 5.3:

$$\begin{aligned}\frac{d\varphi_g^{k,n}(s)}{ds} &+ \Sigma_{tr,g}^{r,n}\varphi_g^{k,n}(s) \\ &= q_g^{r,n} - \frac{1}{v_g}\frac{d\varphi_{g,0}^{k,r,n}}{dt}e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)} \\ &- \frac{1}{v_g}\frac{dq_g^{r,n}}{dt}\left(\frac{1-e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{\Sigma_{tr,g}^{r,n}}\right) \\ &+ \frac{e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{v_g^2}\int_{s_0^{k,r}}^{s}\frac{d^2\varphi_g^{k,n}(s')}{dt^2}e^{\Sigma_{tr,g}^{r,n}\left(s'-s_0^{k,r}\right)}ds'.\end{aligned} \qquad 5.9$$

In the following sections we approximate the source time derivative and angular flux second derivative. Then we can solve Equation 5.9, and the solution will be used to propagate the angular flux along the characteristic.

5.3.3 Approximation of Source Time Derivative with Backward Differences

We can approximate the source time derivative to a varying order of accuracy using backward differences. These approximations are derived in an analogous fashion to BDF, using a Taylor series expansion of the source in time:

$$q_g^{r,n-1} = q_g^{r,n} - \frac{\Delta t_n}{1!}\frac{dq_g^{r,n}}{dt} + \frac{\Delta t_n^2}{2!}\frac{d^2q_g^{r,n}}{dt^2} - \frac{\Delta t_n^3}{3!}\frac{d^3q_g^{r,n}}{dt^3} + \cdots. \qquad 5.10$$

Using algebra we isolate the first source derivative, which we are approximating:

$$\frac{dq_g^{r,n}}{dt} = \frac{q_g^{r,n} - q_g^{r,n-1}}{\Delta t_n} + \frac{\Delta t_n}{2}\frac{d^2 q_g^{r,n}}{dt^2} - \frac{\Delta t_n^2}{3!}\frac{d^3 q_g^{r,n}}{dt^3} + \cdots. \tag{5.11}$$

We approximate the first derivative of the source to the desired accuracy by truncating the expansion at the corresponding order and approximating the remaining derivatives using backward differences in terms of lower derivatives of the source. For example, we approximate the second derivative using backward difference in terms of the first derivatives:

$$\begin{aligned}\frac{d^2 q_g^{r,n}}{dt^2} &= \frac{\frac{dq_g^{r,n}}{dt} - \frac{dq_g^{r,n-1}}{dt}}{\Delta t_n} = \frac{\frac{q_g^{r,n} - q_g^{r,n-1}}{\Delta t_n} - \frac{q_g^{r,n-1} - q_g^{r,n-2}}{\Delta t_{n-1}}}{\Delta t_n} \\ &= \frac{q_g^{r,n} - q_g^{r,n-1}}{\Delta t_n^2} - \frac{q_g^{r,n-1} - q_g^{r,n-2}}{\Delta t_n \Delta t_{n-1}}.\end{aligned} \tag{5.12}$$

We apply the equations for the higher derivatives (e.g. Equation 5.12) to Equation 5.11, which results in an equation for the first time derivative of the source in terms of the source at previous points in time. For example, if a constant time step is assumed, the second order approximation is:

$$\frac{dq_g^{r,n}}{dt} = \frac{\frac{3}{2}q_g^{r,n} - 2q_g^{r,n-1} + \frac{1}{2}q_g^{r,n-2}}{\Delta t_n} - O(\Delta t_n^2). \tag{5.13}$$

While the coefficients for the source derivative approximation with backward differences are identical to those that arise in BDF for the same order and time step size, it is an oversimplification that this *is* BDF. Nonetheless, for many of the test problems considered in this research, SDP methods using backward differences for the source derivative closely reproduce the results of BDC methods using BDF of the same order and time step size.

The source derivative approximation can be written as the weighted sum of the neutron source at previous points in time in an analogous fashion to BDF:

$$\frac{dq_g^{r,n}}{dt} = \sum_{i=0}^{O_{\mathrm{BD}}} A^i q_g^{r,n-i}. \tag{5.14}$$

Applying Equation 5.14 to Equation 5.9 gives the SDP step characteristic equation prior to the treatment of the second time derivative of the angular flux:

$$\begin{aligned}\frac{d\varphi_g^{k,n}(s)}{ds} &+ \Sigma_{tr,g}^{r,n}\varphi_g^{k,n}(s)\\ &= q_g^{r,n} - \frac{1}{v_g}\frac{d\varphi_{g,0}^{k,r,n}}{dt}e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}\\ &-\frac{1}{v_g}\sum_{i=0}^{O_{\mathrm{BD}}}A^i q_g^{r,n-i}\left(\frac{1-e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{\Sigma_{tr,g}^{r,n}}\right)\\ &+\frac{e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{v_g^2}\int_{s_0^{k,r}}^{s}\frac{d^2\varphi_g^{k,n}(s')}{dt^2}e^{\Sigma_{tr,g}^{r,n}\left(s'-s_0^{k,r}\right)}ds'.\end{aligned} \qquad 5.15$$

In the following sections, this equation will be solved two different ways: by truncating the angular flux second derivative (TSDP) and by approximating the angular flux second derivative with the scalar flux (ISDP). Both approximations introduce an error term which will prevent the methods from converging to the solution of the Boltzmann equation in the limit as the time step goes to zero; however, for large time steps this error is expected to be small due to division by the neutron velocity.

5.4 Solution of the Step Characteristic Equation with Truncated Second Derivative

For TSDP, we will assume that the angular flux second derivative term is small:

$$\frac{e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{v_g}\int_{s_0^{k,r}}^{s}\frac{d^2\varphi_g^{k,n}(s')}{dt^2}e^{\Sigma_{tr,g}^{r,n}\left(s'-s_0^{k,r}\right)}ds' \approx 0. \qquad 5.16$$

While the second derivative of the angular flux itself may not necessarily be small, in the characteristic equation for the angular flux (e.g. Equation 5.15) this term is divided by the square of the neutron velocity, which is large and limits the impact of this approximation.

This allows us to rewrite the equation for the spatially-dependent angular flux time derivative in terms of known quantities:

$$\frac{d\varphi_g^{k,n}(s)}{dt} = \frac{d\varphi_{g,0}^{k,r,n}}{dt} e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)} + \sum_{i=0}^{O_{\mathrm{BD}}} A^i q_g^{r,n-i} \left(\frac{1-e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{\Sigma_{tr,g}^{r,n}}\right). \qquad 5.17$$

When Equation 5.16 is applied to Equation 5.9, the result is identical to the step characteristic equation that occurs in ASTC:

$$\begin{aligned} &\frac{d\varphi_g^{k,n}(s)}{ds} + \Sigma_{tr,g}^{r,n}\varphi_g^{k,n}(s) \\ &\qquad = q_g^{r,n} - \frac{1}{v_g}\frac{d\varphi_{g,0}^{k,r,n}}{dt} e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)} \\ &\qquad - \frac{1}{v_g}\sum_{i=0}^{O_{\mathrm{BD}}} A^i q_g^{r,n-i}\left(\frac{1-e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{\Sigma_{tr,g}^{r,n}}\right). \end{aligned} \qquad 5.18$$

As with the other MOC methods, this differential equation is solved using an integrating factor $e^{\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}$, which eliminates some of the exponentials on the right-hand side of Equation 5.18:

$$\begin{aligned} &\frac{d}{ds}\left(e^{\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}\varphi_g^{k,n}(s)\right) \\ &\qquad = q_g^{r,n} e^{\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)} - \frac{1}{v_g}\frac{d\varphi_{g,0}^{k,r,n}}{dt} \\ &\qquad - \frac{1}{v_g}\sum_{i=0}^{O_{\mathrm{BD}}} A^i q_g^{r,n-i}\left(\frac{e^{\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}-1}{\Sigma_{tr,g}^{r,n}}\right). \end{aligned} \qquad 5.19$$

Equation 5.19 is solved by integrating over the segment to position s:

$$\begin{aligned} &\int_{s_0^{k,r}}^{s} \frac{d}{ds}\left(e^{\Sigma_{tr,g}^{r,n}\left(s'-s_0^{k,r}\right)}\varphi_g^{k,n}(s')\right) ds' \\ &\qquad = \int_{s_0^{k,r}}^{s} q_g^{r,n} e^{\Sigma_{tr,g}^{r,n}\left(s'-s_0^{k,r}\right)} - \frac{1}{v_g}\frac{d\varphi_{g,0}^{k,r,n}}{dt} \\ &\qquad - \frac{1}{v_g}\sum_{i=0}^{O_{\mathrm{BD}}} A^i q_g^{r,n-i}\left(\frac{e^{\Sigma_{tr,g}^{r,n}\left(s'-s_0^{k,r}\right)}-1}{\Sigma_{tr,g}^{r,n}}\right) ds'. \end{aligned} \qquad 5.20$$

which evaluates to:

$$\begin{aligned}\varphi_g^{k,n}(s)e^{\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)} &- \varphi_{g,0}^{k,r,n} \\ &= q_g^{r,n}\left(\frac{e^{\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}-1}{\Sigma_{tr,g}^{r,n}}\right)-\frac{1}{v_g}\frac{d\varphi_{g,0}^{k,r,n}}{dt}\left(s-s_0^{k,r}\right) \\ &-\frac{1}{v_g}\sum_{i=0}^{O_{\mathrm{BD}}}A^i q_g^{r,n-i}\left(\frac{e^{\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}-1}{\left(\Sigma_{tr,g}^{r,n}\right)^2}\right) \\ &-\frac{1}{v_g}\sum_{i=0}^{O_{\mathrm{BD}}}A^i q_g^{r,n-i}\left(\frac{s-s_0^{k,r}}{\Sigma_{tr,g}^{r,n}}\right).\end{aligned} \qquad 5.21$$

With some algebra, Equation 5.21 can be rewritten as the spatially-dependent segment-wise angular flux for a segment using TSDP:

$$\begin{aligned}\varphi_g^{k,n}(s) = &\left[\varphi_{g,0}^{k,r,n}-\frac{s-s_0^{k,r}}{v_g}\left(\frac{d\varphi_{g,0}^{k,r,n}}{dt}-\sum_{i=0}^{O_{\mathrm{BD}}}\frac{A^i q_g^{r,n-i}}{\Sigma_{tr,g}^{r,n}}\right)\right]e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)} \\ &+\left(q_g^{r,n}-\frac{1}{v_g}\sum_{i=0}^{O_{\mathrm{BD}}}\frac{A^i q_g^{r,n-i}}{\Sigma_{tr,g}^{r,n}}\right)\left(\frac{1-e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{\Sigma_{tr,g}^{r,n}}\right).\end{aligned} \qquad 5.22$$

This is the solution to the TSDP characteristic equation. As with the other MOC methods, it is used to propagate the angular flux along characteristics in space and to integrate the region-wise scalar flux with the segment-wise average angular fluxes.

5.4.1 Angular Flux Propagation Along the Characteristic - TSDP

We know the incoming angular flux at the beginning of a segment because we calculated the outgoing angular flux for the previous segment. This is found by evaluating Equation 5.22 at the end of the segment:

$$\begin{aligned}\varphi_g^{k,n}&\left(s_0^{k,r}+\Delta s^{k,r}\right) \\ &= \left[\varphi_{g,0}^{k,r,n}-\frac{\Delta s^{k,r}}{v_g}\left(\frac{d\varphi_{g,0}^{k,r,n}}{dt}-\sum_{i=0}^{O_{\mathrm{BD}}}\frac{A^i q_g^{r,n-i}}{\Sigma_{tr,g}^{r,n}}\right)\right]e^{-\Sigma_{tr,g}^{r,n}\Delta s^{k,r}} \\ &+\left(q_g^{r,n}-\frac{1}{v_g}\sum_{i=0}^{O_{\mathrm{BD}}}\frac{A^i q_g^{r,n-i}}{\Sigma_{tr,g}^{r,n}}\right)\left(\frac{1-e^{-\Sigma_{tr,g}^{r,n}\Delta s^{k,r}}}{\Sigma_{tr,g}^{r,n}}\right).\end{aligned} \qquad 5.23$$

5.4.2 Numerical Integration of the Region-wise Scalar Flux - TSDP

In order to numerically integrate the region-wise scalar flux, we will calculate the segment-wise average angular flux. This is calculated by integrating the segment-wise spatially-dependent angular flux over the segment:

$$\begin{aligned}\bar{\varphi}_g^{k,r,n} = \Bigg[&\varphi_{g,0}^{k,r,n} - \frac{\Delta s^{k,r}}{v_g}\left(\frac{d\varphi_{g,0}^{k,r,n}}{dt} - \sum_{i=0}^{O_{\mathrm{BD}}} \frac{A^i q_g^{r,n-i}}{\Sigma_{tr,g}^{r,n}}\right) - \frac{q_g^{r,n}}{\Sigma_{tr,g}^{r,n}} + \frac{2}{v_g}\sum_{i=0}^{O_{\mathrm{BD}}} \frac{A^i q_g^{r,n-i}}{\left(\Sigma_{tr,g}^{r,n}\right)^2} \\ &- \frac{1}{\Sigma_{tr,g}^{r,n} v_g}\frac{d\varphi_{g,0}^{k,r,n}}{dt}\Bigg]\left(\frac{1-e^{-\Sigma_{tr,g}^{r,n}\Delta s^{k,r}}}{\Sigma_{tr,g}^{r,n}\Delta s^{k,r}}\right) + \frac{q_g^{r,n}}{\Sigma_{tr,g}^{r,n}} \\ &- \frac{2}{v_g}\sum_{i=0}^{O_{\mathrm{BD}}} \frac{A^i q_g^{r,n-i}}{\left(\Sigma_{tr,g}^{r,n}\right)^2} + \frac{1}{\Sigma_{tr,g}^{r,n} v_g}\frac{d\varphi_{g,0}^{k,r,n}}{dt}.\end{aligned} \tag{5.24}$$

The region-wise scalar flux is calculated by numerically integrating the segment-wise average angular fluxes over space and angle as in SSC:

$$\phi_g^{r,n} = \sum_m w_m \sum_l w_l \frac{\sum_{k\in r} \bar{\varphi}_g^{k,r,n} \Delta s^{k,r} \Delta R_l}{\sum_{k\in r} \Delta s^{k,r} \Delta R_l}. \tag{5.25}$$

5.4.3 Propagation of the Angular Flux Time Derivative on Characteristic - TSDP

We propagate the angular flux along the characteristic so that the incoming angular flux for every segment will be known. We will similarly propagate the angular flux time derivative along the characteristic. This simply involves evaluating Equation 5.17 at the end of the segment:

$$\frac{d\varphi_g^{k,n}\left(s_0^{k,r} + \Delta s^{k,r}\right)}{dt} = \frac{d\varphi_{g,0}^{k,r,n}}{dt} e^{-\Sigma_{tr,g}^{r,n}\Delta s^{k,r}} + \sum_{i=0}^{O_{\mathrm{BD}}} A^i q_g^{r,n-i}\left(\frac{1-e^{-\Sigma_{tr,g}^{r,n}\Delta s^{k,r}}}{\Sigma_{tr,g}^{r,n}}\right). \tag{5.26}$$

This equation is identical to the equation for the outgoing angular flux time derivative that arises in ASTC. Since Equations 5.23, 5.24, and 5.26 are identical for TSDP and ASTC, the two methods are equivalent.

5.5 Solution of the Step Characteristic Equation for SDP with Isotropic Correction

In the previous section we assumed that the angular flux second derivative term in the characteristic equation was small so that it could be truncated. As we will see in Chapter 6, there may be conditions where TSDP is less accurate than IBDC (e.g. when the angular distribution of the neutron flux is changing slowly but the change in the scalar flux is accelerating or decelerating quickly, or equivalently $\frac{d\boldsymbol{J}}{dt}$ is small but $\frac{d^2\phi}{dt^2}$ is large).

To ensure that SDP is systematically more accurate than IBDC, we will also consider an SDP method where the angular flux second derivative is approximated as isotropic using the region-wise scalar flux:

$$\frac{e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{v_g}\int_{s_0^{k,r}}^{s}\frac{d^2\varphi_g^{k,n}(s')}{dt^2}e^{\Sigma_{tr,g}^{r,n}\left(s'-s_0^{k,r}\right)}ds' \approx \frac{e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{4\pi v_g}\frac{d^2\phi_g^{r,n}}{dt^2}\int_{s_0^{k,r}}^{s}e^{\Sigma_{tr,g}^{r,n}\left(s'-s_0^{k,r}\right)}ds', \tag{5.27}$$

which allows us to evaluate the integral over the segment:

$$\frac{e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{v_g}\int_{s_0^{k,r}}^{s}\frac{d^2\varphi_g^{k,n}(s')}{dt^2}e^{\Sigma_{tr,g}^{r,n}\left(s'-s_0^{k,r}\right)}ds' \approx \frac{1}{4\pi v_g}\frac{d^2\phi_g^{r,n}}{dt^2}\left(\frac{1-e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{\Sigma_{tr,g}^{r,n}}\right). \tag{5.28}$$

However, now we require an approximation for the scalar flux second derivative. To remain consistent with our work so far, we will approximate the second derivative using backward differences:

$$\frac{d^2\phi_g^{r,n}}{dt^2} \approx \sum_{i=0}^{O_{\mathrm{BD}}+1} B^i\phi_g^{r,n-i}, \tag{5.29}$$

where the coefficients B^i are derived in a similar fashion to the coefficients for the first derivative. The coefficients are a function of the time step size of the current and previous time steps, and they have the dimension of inverse-time squared (i.e. t^{-2}).

This allows us to rewrite the equation for the spatially-dependent angular flux time derivative with the scalar flux second derivative as an additional source-derivative term:

$$\begin{aligned}\frac{d\varphi_g^{k,n}(s)}{dt} &= \frac{d\varphi_{g,0}^{k,r,n}}{dt} e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)} \\ &\quad + \left(\sum_{i=0}^{O_{\mathrm{BD}}} A^i q_g^{r,n-i} - \sum_{i=0}^{O_{\mathrm{BD}}+1} \frac{B^i \phi_g^{r,n-i}}{4\pi v_g}\right)\left(\frac{1-e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{\Sigma_{tr,g}^{r,n}}\right).\end{aligned} \tag{5.30}$$

When Equation 5.30 is applied to Equation 5.9, the result is the step characteristic equation for ISDP:

$$\begin{aligned}&\frac{d\varphi_g^{k,n}(s)}{ds} + \Sigma_{tr,g}^{r,n}\varphi_g^{k,n}(s) \\ &\quad = q_g^{r,n} - \frac{1}{v_g}\frac{d\varphi_{g,0}^{k,r,n}}{dt} e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)} \\ &\quad - \frac{1}{v_g}\left(\sum_{i=0}^{O_{\mathrm{BD}}} A^i q_g^{r,n-i} - \sum_{i=0}^{O_{\mathrm{BD}}+1} \frac{B^i \phi_g^{r,n-i}}{4\pi v_g}\right)\left(\frac{1-e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{\Sigma_{tr,g}^{r,n}}\right).\end{aligned} \tag{5.31}$$

This differential equation only differs from the equivalent equation for TSDP by the presence of the scalar flux second derivative source term. To simplify the characteristic equation, we will define a transient source term which encompasses the source derivative and the scalar flux second derivative:

$$T_{g,0}^{r,n} \equiv \frac{1}{\Sigma_{tr,g}^{r,n}}\left(\sum_{i=0}^{O_{\mathrm{BD}}} A^i q_g^{r,n-i} - \sum_{i=0}^{O_{\mathrm{BD}}+1} \frac{B^i \phi_g^{r,n-i}}{4\pi v_g}\right), \tag{5.32}$$

where the 0 subscript is to distinguish this transient fixed source term from those that are used for methods derived later in the chapter.

This allows us to write the ISDP characteristic equation more succinctly:

$$\begin{aligned}\frac{d\varphi_g^{k,n}(s)}{ds} &+ \Sigma_{tr,g}^{r,n}\varphi_g^{k,n}(s) \\ &= q_g^{r,n} - \frac{1}{v_g}\frac{d\varphi_{g,0}^{k,r,n}}{dt}e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)} - \frac{T_{g,0}^{r,n}}{v_g}\left(1-e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}\right).\end{aligned} \tag{5.33}$$

The process to solve this equation is similar to TSDP; an integrating factor is applied, the equation is integrated over the segment length, and after some algebra the solution is the spatially-dependent segment-wise angular flux:

$$\begin{aligned}\varphi_g^{k,n}(s) = &\left[\varphi_{g,0}^{k,r,n} - \frac{s-s_0^{k,r}}{v_g}\left(\frac{d\varphi_{g,0}^{k,r,n}}{dt} - T_{g,0}^{r,n}\right)\right]e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)} \\ &+ \left(q_g^{r,n} - \frac{T_{g,0}^{r,n}}{v_g}\right)\left(\frac{1-e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{\Sigma_{tr,g}^{r,n}}\right).\end{aligned} \tag{5.34}$$

This is the solution to the ISDP characteristic equation. It is used to propagate the angular flux along characteristics in space and to numerically integrate the region-wise scalar flux using the segment-wise average angular fluxes. In addition, Equation 5.30 will be used to propagate the angular flux time derivative along the characteristics.

5.5.1 Angular Flux Propagation Along the Characteristic - ISDP

To propagate the angular flux along the characteristic, we will calculate the outgoing angular flux for each segment. This is calculated by evaluating Equation 5.34 at the end of the segment:

$$\begin{aligned}\varphi_g^{k,n}&\left(s_0^{k,r} + \Delta s^{k,r}\right) \\ &= \left[\varphi_{g,0}^{k,r,n} - \frac{\Delta s^{k,r}}{v_g}\left(\frac{d\varphi_{g,0}^{k,r,n}}{dt} - T_{g,0}^{r,n}\right)\right]e^{-\Sigma_{tr,g}^{r,n}\Delta s^{k,r}} \\ &+ \left(q_g^{r,n} - \frac{T_{g,0}^{r,n}}{v_g}\right)\left(\frac{1-e^{-\Sigma_{tr,g}^{r,n}\Delta s^{k,r}}}{\Sigma_{tr,g}^{r,n}}\right).\end{aligned} \tag{5.35}$$

5.5.2 Numerical Integration of the Region-wise Scalar Flux - ISDP

We calculate the segment-wise average angular flux in order to obtain the region-wise scalar flux. This is calculated by integrating the segment-wise spatially-dependent angular flux over the segment:

$$\begin{aligned}\bar{\varphi}_g^{k,r,n} = \Bigg[&\varphi_{g,0}^{k,r,n} - \frac{\Delta s^{k,r}}{v_g}\left(\frac{d\varphi_{g,0}^{k,r,n}}{dt} - T_{g,0}^{r,n}\right) - \frac{q_g^{r,n}}{\Sigma_{tr,g}^{r,n}} + 2\frac{T_{g,0}^{r,n}}{\Sigma_{tr,g}^{r,n} v_g} \\ &- \frac{1}{\Sigma_{tr,g}^{r,n} v_g}\frac{d\varphi_{g,0}^{k,r,n}}{dt}\Bigg]\left(\frac{1 - e^{-\Sigma_{tr,g}^{r,n}\Delta s^{k,r}}}{\Sigma_{tr,g}^{r,n}\Delta s^{k,r}}\right) + \frac{q_g^{r,n}}{\Sigma_{tr,g}^{r,n}} - 2\frac{T_{g,0}^{r,n}}{\Sigma_{tr,g}^{r,n} v_g} \\ &+ \frac{1}{\Sigma_{tr,g}^{r,n} v_g}\frac{d\varphi_{g,0}^{k,r,n}}{dt}.\end{aligned} \qquad 5.36$$

The region-wise scalar flux is calculated by numerically integrating the segment-wise average angular fluxes over space and angle using Equation 5.25.

5.5.3 Propagation of the Angular Flux Time Derivative on Characteristic - ISDP

To propagate the angular flux time derivative along the characteristic we evaluated Equation 5.30 at the end of the segment:

$$\frac{d\varphi_g^{k,n}\left(s_0^{k,r} + \Delta s^{k,r}\right)}{dt} = \frac{d\varphi_{g,0}^{k,r,n}}{dt} e^{-\Sigma_{tr,g}^{r,n}\Delta s^{k,r}} + T_{g,0}^{r,n}\left(1 - e^{-\Sigma_{tr,g}^{r,n}\Delta s^{k,r}}\right). \qquad 5.37$$

5.6 Solution of Characteristic Equation with Cross Section Derivative Propagation

The previous derivations for SDP have assumed that the transport cross section changes slowly in time so its time derivative can be neglected. This is a common assumption in time-dependent transport methods because it is true throughout most of the space and time domain. The exception is near volumes in which the material composition changes quickly during the transient, such as a control rod guide tube or voiding moderator.

This approximation is also a matter of practicality. Allowing the cross sections to vary in space and/or time substantially complicates the solution of the transport equation. For MOC or collision probability methods (CPM) the resulting differential or integral equation is not analytically solvable without the flat cross section approximation. To accommodate this approximation, the regions are typically just defined to be fine enough that the approximation is acceptable.

Nonetheless, to understand the implications of allowing the transport cross section to vary in time, we have developed the Source and Cross Section Derivative Propagation (SCDP) method.

We begin with the characteristic equation prior to time-discretization without resolving the angular flux time derivative integral, which is similar to Equation 5.9:

$$\begin{aligned}\varphi_g^k(s,t) = \varphi_{g,0}^{k,r}(t)e^{-\Sigma_{tr,g}^r(t)\left[s-s_0^{k,r}\right]} + q_g^r(t)\left[\frac{1-e^{-\Sigma_{tr,g}^r(t)\left[s-s_0^{k,r}\right]}}{\Sigma_{tr,g}^r(t)}\right]\\ - \frac{e^{-\Sigma_{tr,g}^r(t)\left[s-s_0^{k,r}\right]}}{v_g}\int_{s_0^{k,r}}^{s}\frac{d\varphi_g^k(s',t)}{dt}e^{\Sigma_{tr,g}^r(t)\left[s'-s_0^{k,r}\right]}ds',\end{aligned} \tag{5.38}$$

where for clarity we will temporarily use parentheses exclusively to indicate functional dependence and square brackets to clarify arithmetic order precedence.

We assume that the transport cross section varies linearly with time near the present point in time t_n:

$$\Sigma_{tr,g}^r(t) \approx \Sigma_{tr,g}^r(t_n) + [t-t_n]\frac{d\Sigma_{tr,g}^r(t_n)}{dt}. \tag{5.39}$$

We take a time derivative of Equation 5.38:

$$\begin{aligned}\frac{d\varphi_g^k(s,t)}{dt} = \frac{d}{dt}\left[\varphi_{g,0}^{k,r}(t)e^{-\Sigma_{tr,g}^r(t)\left[s-s_0^{k,r}\right]}\right]\\ + \frac{d}{dt}\left[q_g^r(t)\left[\frac{1-e^{-\Sigma_{tr,g}^r(t)\left[s-s_0^{k,r}\right]}}{\Sigma_{tr,g}^r(t)}\right]\right]\\ - \frac{d}{dt}\left[\frac{e^{-\Sigma_{tr,g}^r(t)\left[s-s_0^{k,r}\right]}}{v_g}\int_{s_0^{k,r}}^{s}\frac{d\varphi_g^k(s',t)}{dt}e^{\Sigma_{tr,g}^r(t)\left[s'-s_0^{k,r}\right]}ds'\right].\end{aligned} \tag{5.40}$$

We evaluate each of the derivatives using the chain rule:

$$\begin{aligned}\frac{d}{dt}&\left[\varphi_{g,0}^{k,r}(t)e^{-\left[\Sigma_{tr,g}^{r}(t_n)+[t-t_n]\frac{d\Sigma_{tr,g}^{r}(t_n)}{dt}\right]\left[s-s_0^{k,r}\right]}\right]\\&= \frac{d\varphi_{g,0}^{k,r}(t)}{dt}e^{-\Sigma_{tr,g}^{r}(t)\left[s-s_0^{k,r}\right]}\\&- \frac{d\Sigma_{tr,g}^{r}(t_n)}{dt}\left[s-s_0^{k,r}\right]\varphi_g^k\left(s_0^{k,r}\right)e^{-\Sigma_{tr,g}^{r}\left[s-s_0^{k,r}\right]},\end{aligned} \tag{5.41}$$

and:

$$\begin{aligned}\frac{d}{dt}&\left[q_g^r(t)\left[\frac{1-e^{-\left[\Sigma_{tr,g}^{r}(t_n)+[t-t_n]\frac{d\Sigma_{tr,g}^{r}(t_n)}{dt}\right]\left[s-s_0^{k,r}\right]}}{\Sigma_{tr,g}^{r}(t_n)+[t-t_n]\frac{d\Sigma_{tr,g}^{r}(t_n)}{dt}}\right]\right]\\&= \frac{dq_g^r(t)}{dt}\left[\frac{1-e^{-\Sigma_{tr,g}^{r}(t)\left[s-s_0^{k,r}\right]}}{\Sigma_{tr,g}^{r}(t)}\right]\\&- q_g^r(t)\frac{d\Sigma_{tr,g}^{r}(t)}{dt}\left[\frac{1-e^{-\Sigma_{tr,g}^{r}(t)\left[s-s_0^{k,r}\right]}}{\Sigma_{tr,g}^{r}(t)^2}\right]\\&+ \frac{q_g^r(t)}{\Sigma_{tr,g}^{r}(t)}\frac{d\Sigma_{tr,g}^{r}(t)}{dt}\left[s-s_0^{k,r}\right]e^{-\Sigma_{tr,g}^{r}(t)\left[s-s_0^{k,r}\right]},\end{aligned} \tag{5.42}$$

and:

$$\begin{aligned}&\frac{d}{dt}\left[\frac{e^{-\Sigma_{tr,g}^{r}(t)\left[s-s_0^{k,r}\right]}}{v_g}\int_{s_0^{k,r}}^{s}\frac{d\varphi_g^k(s',t)}{dt}e^{\Sigma_{tr,g}^{r}(t)\left[s'-s_0^{k,r}\right]}ds'\right]\\&= -\frac{d\Sigma_{tr,g}^{r}(t)}{dt}\frac{\left[s-s_0^{k,r}\right]}{v_g}e^{-\Sigma_{tr,g}^{r}(t)\left[s-s_0^{k,r}\right]}\int_{s_0^{k,r}}^{s}\frac{d\varphi_g^k(s',t)}{dt}e^{\Sigma_{tr,g}^{r}(t)\left[s'-s_0^{k,r}\right]}ds'\\&+ \frac{e^{-\Sigma_{tr,g}^{r}(t)\left[s-s_0^{k,r}\right]}}{v_g}\int_{s_0^{k,r}}^{s}\frac{d^2\varphi_g^k(s',t)}{dt^2}e^{\Sigma_{tr,g}^{r,n}(t)\left[s'-s_0^{k,r}\right]}ds'\\&+ \frac{d\Sigma_{tr,g}^{r}(t)}{dt}\frac{e^{-\Sigma_{tr,g}^{r}(t)\left[s-s_0^{k,r}\right]}}{v_g}\int_{s_0^{k,r}}^{s}\frac{d\varphi_g^k(s',t)}{dt}\left[s'-s_0^{k,r}\right]e^{\Sigma_{tr,g}^{r}(t)\left[s'-s_0^{k,r}\right]}ds'.\end{aligned} \tag{5.43}$$

We evaluate all of the terms at time t_n and apply the results to Equation 5.40:

$$
\begin{aligned}
\frac{d\varphi_g^{k,n}(s)}{dt} = {} & \left[\frac{d\varphi_{g,0}^{k,r,n}}{dt} - \left(s - s_0^{k,r}\right)\frac{d\Sigma_{tr,g}^{r,n}}{dt}\left(\varphi_{g,0}^{k,r,n} - \frac{q_g^{r,n}}{\Sigma_{tr,g}^{r,n}}\right)\right] e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)} \\
& + \left(\frac{dq_g^{r,n}}{dt} - \frac{q_g^{r,n}}{\Sigma_{tr,g}^{r,n}}\frac{d\Sigma_{tr,g}^{r,n}}{dt}\right)\left(\frac{1 - e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{\Sigma_{tr,g}^{r,n}}\right) \\
& + \frac{d\Sigma_{tr,g}^{r,n}}{dt}\frac{s - s_0^{k,r}}{v_g} e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)} \int\limits_{s_0^{k,r}}^{s} \frac{d\varphi_g^{k,n}(s')}{dt} e^{\Sigma_{tr,g}^{r,n}\left(s'-s_0^{k,r}\right)} ds' \\
& - \frac{e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{v_g} \int\limits_{s_0^{k,r}}^{s} \frac{d^2\varphi_g^{k,n}(s')}{dt^2} e^{\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)} ds' \\
& - \frac{d\Sigma_{tr,g}^{r,n}}{dt}\frac{e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{v_g} \\
& \times \int\limits_{s_0^{k,r}}^{s} \frac{d\varphi_g^{k,n}(s')}{dt}\left(s' - s_0^{k,r}\right) e^{\Sigma_{tr,g}^{r,n}\left(s'-s_0^{k,r}\right)} ds',
\end{aligned}
\qquad 5.44
$$

where we have returned to our convention of using a mix of brackets and parentheses to indicate arithmetic order precedence.

Equation 5.44 is used in SCDP to propagate the angular flux time derivative along the characteristic. When we substitute it into Equation 5.38 we get the characteristic equation for SCDP:

$$\begin{aligned}
\frac{d\varphi_g^{k,n}(s)}{ds} + \Sigma_{tr,g}^{r,n}\varphi_g^{k,n}(s)
&= q_g^{r,n}\\
&- \frac{1}{v_g}\left[\frac{d\varphi_{g,0}^{k,r,n}}{dt} - \frac{d\Sigma_{tr,g}^{r,n}}{dt}\varphi_{g,0}^{k,r,n}\left(s-s_0^{k,r}\right)\right.\\
&\left.+\frac{d\Sigma_{tr,g}^{r,n}}{dt}\frac{q_g^{r,n}}{\Sigma_{tr,g}^{r,n}}\left(s-s_0^{k,r}\right)\right]e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}\\
&-\frac{1}{v_g}\left(\frac{dq_g^{r,n}}{dt}-\frac{d\Sigma_{tr,g}^{r,n}}{dt}\frac{q_g^{r,n}}{\Sigma_{tr,g}^{r,n}}\right)\left(\frac{1-e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{\Sigma_{tr,g}^{r,n}}\right)\\
&-\frac{d\Sigma_{tr,g}^{r,n}}{dt}\frac{s-s_0^{k,r}}{v_g^2}e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}\int_{s_0^{k,r}}^{s}\frac{d\varphi_g^{k,n}(s')}{dt}e^{\Sigma_{tr,g}^{r,n}\left(s'-s_0^{k,r}\right)}ds'\\
&+\frac{e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{v_g^2}\int_{s_0^{k,r}}^{s}\frac{d^2\varphi_g^{k,n}(s')}{dt^2}e^{\Sigma_{tr,g}^{r,n}\left(s^{'}-s_0^{k,r}\right)}ds'\\
&+\frac{d\Sigma_{tr,g}^{r,n}}{dt}\frac{e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{v_g^2}\\
&\times\int_{s_0^{k,r}}^{s}\frac{d\varphi_g^{k,n}(s')}{dt}\left(s'-s_0^{k,r}\right)e^{\Sigma_{tr,g}^{r,n}\left(s'-s_0^{k,r}\right)}ds'.
\end{aligned}\tag{5.45}$$

The solution to this characteristic equation is:

$$
\begin{aligned}
\varphi_g^{k,n}(s) = \Bigg[& \varphi_{g,0}^{k,r,n} - \frac{s - s_0^{k,r}}{v_g}\left(\frac{d\varphi_{g,0}^{k,r,n}}{dt} - \frac{1}{\Sigma_{tr,g}^{r,n}}\frac{dq_g^{r,n}}{dt}\right) \\
& + \frac{\left(s - s_0^{k,r}\right)^2}{2v_g}\frac{d\Sigma_{tr,g}^{r,n}}{dt}\left(\varphi_{g,0}^{k,r,n} - \frac{q_g^{r,n}}{\Sigma_{tr,g}^{r,n}}\right)\Bigg] e^{-\Sigma_{tr,g}^{r,n}\left(s - s_0^{k,r}\right)} \\
& + \left[q_g^{r,n} - \frac{1}{\Sigma_{tr,g}^{r,n} v_g}\left(\frac{dq_g^{r,n}}{dt} - \frac{d\Sigma_{tr,g}^{r,n}}{dt}\frac{q_g^{r,n}}{\Sigma_{tr,g}^{r,n}}\right)\right]\left(\frac{1 - e^{-\Sigma_{tr,g}^{r,n}\left(s - s_0^{k,r}\right)}}{\Sigma_{tr,g}^{r,n}}\right) \\
& - \frac{d\Sigma_{tr,g}^{r,n}}{dt}\int_{s_0^{k,r}}^{s}\frac{s' - s_0^{k,r}}{v_g^2}\int_{s_0^{k,r}}^{s'}\frac{d\varphi_g^{k,n}(s'')}{dt}e^{\Sigma_{tr,g}^{r,n}\left(s'' - s_0^{k,r}\right)}ds''ds' \\
& + \int_{s_0^{k,r}}^{s}\frac{e^{-\Sigma_{tr,g}^{r,n}\left(s' - s_0^{k,r}\right)}}{v_g^2}\int_{s_0^{k,r}}^{s'}\frac{d^2\varphi_g^{k,n}(s'')}{dt^2}e^{\Sigma_{tr,g}^{r,n}\left(s'' - s_0^{k,r}\right)}ds''ds' \\
& + \frac{d\Sigma_{tr,g}^{r,n}}{dt}\int_{s_0^{k,r}}^{s}\frac{e^{-\Sigma_{tr,g}^{r,n}\left(s' - s_0^{k,r}\right)}}{v_g} \\
& \times \int_{s_0^{k,r}}^{s'}\frac{d\varphi_g^{k,n}(s'')}{dt}\left(s'' - s_0^{k,r}\right)e^{\Sigma_{tr,g}^{r,n}\left(s'' - s_0^{k,r}\right)}ds''\,ds'.
\end{aligned}
\tag{5.46}
$$

We need to address the second derivative terms. While we can truncate these terms as we did for TDSP, assuming that the angular flux time derivatives are isotropic imposes only a minor computational and memory expense but is expected to provide enhanced accuracy. As a result, we will only consider the SCDP method with an isotropic correction (ISCDP).

After applying the isotropic approximation to approximate the time derivatives, we can evaluate all of the integrals in Equations 5.44 and 5.46. The first integral evaluates to:

$$\frac{d\Sigma_{tr,g}^{r,n}}{dt}\frac{s-s_0^{k,r}}{v_g}e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}\int_{s_0^{k,r}}^{s}\frac{d\varphi_g^{k,n}(s')}{dt}e^{\Sigma_{tr,g}^{r,n}\left(s'-s_0^{k,r}\right)}ds'$$

$$=\frac{d\Sigma_{tr,g}^{r,n}}{dt}\frac{d\phi_g^{r,n}}{dt}\frac{s-s_0^{k,r}}{4\pi v_g}\left(\frac{1-e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{\Sigma_{tr,g}^{r,n}}\right). \qquad 5.47$$

The second integral evaluates to:

$$\frac{e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{v_g}\int_{s_0^{k,r}}^{s}\frac{d^2\varphi_g^{k,n}(s')}{dt^2}e^{\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}ds'$$

$$=\frac{1}{4\pi v_g}\frac{d^2\phi_g^{r,n}}{dt^2}\left(\frac{1-e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{\Sigma_{tr,g}^{r,n}}\right), \qquad 5.48$$

The third integral evaluates to:

$$\frac{d\Sigma_{tr,g}^{r,n}}{dt}\frac{e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{v_g}\int_{s_0^{k,r}}^{s}\frac{d\varphi_g^{k,n}(s')}{dt}s'e^{\Sigma_{tr,g}^{r,n}\left(s'-s_0^{k,r}\right)}ds'$$

$$=\frac{d\Sigma_{tr,g}^{r,n}}{dt}\frac{d\phi_g^{r,n}}{dt}\frac{1}{4\pi\Sigma_{tr,g}^{r,n}v_g}\left(s-\frac{1-e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{\Sigma_{tr,g}^{r,n}}\right), \qquad 5.49$$

When we apply the integrals to Equation 5.44 along with backward differences to approximate the time derivatives, the resulting characteristic equation for the angular flux time derivative is:

$$\frac{d\varphi_g^{k,n}(s)}{dt}=\left[\frac{d\varphi_{g,0}^{k,r,n}}{dt}-\left(s-s_0^{k,r}\right)T_{g,1}^{r,n}\right]e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}$$

$$+T_{g,2}^{r,n}\left(\frac{1-e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{\Sigma_{tr,g}^{r,n}}\right), \qquad 5.50$$

where the two new transient fixed source terms are defined as:

$$\begin{aligned} T_{g,1}^{r,n} &\equiv \sum_{i=0}^{O_{\mathrm{BD}}} A^{i} \Sigma_{tr,g}^{r,n-i} \left(\varphi_{g,0}^{k,r,n} - \frac{q_g^{r,n}}{\Sigma_{tr,g}^{r,n}} + \frac{1}{4\pi \Sigma_{tr,g}^{r,n} v_g} \sum_{i=0}^{O_{\mathrm{BD}}} A^{i} \phi_g^{r,n-i} \right), \\ T_{g,2}^{r,n} &\equiv \sum_{i=0}^{O_{\mathrm{BD}}} A^{i} q_g^{r,n-i} - \sum_{i=0}^{O_{\mathrm{BD}}+1} \frac{B^{i} \phi_g^{r,n-i}}{4\pi v_g} - \frac{q_g^{r,n}}{\Sigma_{tr,g}^{r,n}} \sum_{i=0}^{O_{\mathrm{BD}}} A^{i} \Sigma_{tr,g}^{r,n-i} \\ &\quad + \frac{1}{4\pi \Sigma_{tr,g}^{r,n} v_g} \sum_{i=0}^{O_{\mathrm{BD}}} A^{i} \Sigma_{tr,g}^{r,n-i} \sum_{i=0}^{O_{\mathrm{BD}}} A^{i} \phi_g^{r,n-i}. \end{aligned} \tag{5.51}$$

With the same assumptions, the resulting characteristic equation for the angular flux is:

$$\begin{aligned} \varphi_g^{k,n}(s) = &\left[\varphi_{g,0}^{k,r,n} - \frac{s - s_0^{k,r}}{v_g} \left(\frac{d\varphi_{g,0}^{k,r,n}}{dt} - T_{g,2}^{r,n} \right) + \frac{\left(s - s_0^{k,r}\right)^2}{2 v_g} T_{g,1}^{r,n} \right] \\ &\times e^{-\Sigma_{tr,g}^{r,n}\left(s - s_0^{k,r}\right)} + \left(q_g^{r,n} - \frac{T_{g,2}^{r,n}}{\Sigma_{tr,g}^{r,n} v_g} \right) \left(\frac{1 - e^{-\Sigma_{tr,g}^{r,n}\left(s - s_0^{k,r}\right)}}{\Sigma_{tr,g}^{r,n}} \right). \end{aligned} \tag{5.52}$$

This illustrates that the computational impact of ISCDP is minor relative to the SDP methods. The angular flux time derivative equation has gained a linear term and the angular flux equation has gained a quadratic term. It is also noteworthy that when the cross section derivative goes to zero, the ISCDP equations become equivalent to the ISDP equations.

5.6.1 Angular Flux Propagation Along the Characteristic – ISCDP

To propagate the angular flux along the characteristic, we evaluate Equation 5.52 at the end of the segment:

$$\begin{aligned} \varphi_g^{k,n}(\Delta s^{k,r}) = &\left[\varphi_{g,0}^{k,r,n} - \frac{\Delta s^{k,r}}{v_g} \left(\frac{d\varphi_{g,0}^{k,r,n}}{dt} - T_{g,2}^{r,n} \right) + \frac{{\Delta s^{k,r}}^2}{2 v_g} T_{g,1}^{r,n} \right] e^{-\Sigma_{tr,g}^{r,n} \Delta s^{k,r}} \\ &+ \left[q_g^{r,n} - \frac{T_{g,2}^{r,n}}{\Sigma_{tr,g}^{r,n} v_g} \right] \left(\frac{1 - e^{-\Sigma_{tr,g}^{r,n} \Delta s^{k,r}}}{\Sigma_{tr,g}^{r,n}} \right). \end{aligned} \tag{5.53}$$

5.6.2 Numerical Integration of the Region-wise Scalar Flux – ISCDP

We calculate the segment-wise average angular flux to numerically integrate the region-wise scalar flux. The average angular flux is calculated by integrating over the segment-wise angular flux:

$$\begin{aligned}\bar{\varphi}_g^{k,r,n} = \Bigg[\varphi_{g,0}^{k,r,n} - \frac{\Delta s^{k,r}}{v_g}\left(\frac{d\varphi_{g,0}^{k,r,n}}{dt} - T_{g,2}^{r,n}\right) + \frac{\Delta s^{k,r^2}}{2v_g} T_{g,1}^{r,n} - \frac{q_g^{r,n}}{\Sigma_{tr,g}^{r,n}} \\ + \frac{T_{g,2}^{r,n}}{\Sigma_{tr,g}^{r,n\;2} v_g}\Bigg]\left(\frac{1 - e^{-\Sigma_{tr,g}^{r,n}\Delta s^{k,r}}}{\Sigma_{tr,g}^{r,n}}\right) + \frac{q_g^{r,n}}{\Sigma_{tr,g}^{r,n}} - \frac{T_{g,2}^{r,n}}{\Sigma_{tr,g}^{r,n\;2} v_g}.\end{aligned} \tag{5.54}$$

The region-wise scalar flux is calculated by numerically integrating the segment-wise average angular flux over space and angle using Equation 5.25.

5.6.3 Propagation of the Angular Flux Time Derivative on Characteristic - ISCDP

To propagate the angular flux time derivative along the characteristic, we evaluate Equation 5.50 at the end of the segment:

$$\frac{d\varphi_g^{k,n}(\Delta s^{k,r})}{dt} = \left(\frac{d\varphi_{g,0}^{k,r,n}}{dt} - \Delta s^{k,r} T_{g,1}^{r,n}\right) e^{-\Sigma_{tr,g}^{r,n}\Delta s^{k,r}} + T_{g,2}^{r,n}\left(\frac{1 - e^{-\Sigma_{tr,g}^{r,n}\Delta s^{k,r}}}{\Sigma_{tr,g}^{r,n}}\right). \tag{5.55}$$

5.7 Derivation for Second Source Derivative Propagation (2SDP)

In the previous derivations, the characteristic equation for the angular flux time derivative is given in terms of the first derivative of the source. However, through repeated differentiation, we can define characteristic equations for higher time derivatives of the angular flux. In this section we will derive the characteristic equation for a second-order source expansion (2SDP).

We begin by taking another time derivative of Equation 5.8 while assuming that the transport cross section is changing slowly, providing an equation for the second angular flux time derivative:

$$\begin{aligned}\frac{d^2\varphi_g^{k,n}(s)}{dt^2} &= \frac{d^2\varphi_{g,0}^{k,r,n}}{dt^2} e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)} + \frac{d^2 q_g^{r,n}}{d^2t}\left(\frac{1-e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{\Sigma_{tr,g}^{r,n}}\right) \\ &\quad - \frac{e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{v_g}\int_{s_0^{k,r}}^{s} \frac{d^3\varphi_g^{k,n}(s')}{dt^3} e^{\Sigma_{tr,g}^{r,n}\left(s'-s_0^{k,r}\right)} ds'.\end{aligned} \tag{5.56}$$

We substitute Equation 5.56 into Equation 5.8:

$$\begin{aligned}\frac{d\varphi_g^{k,n}(s)}{dt} &= \frac{d\varphi_{g,0}^{k,r,n}}{dt} e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)} + \frac{dq_g^{r,n}}{dt}\left(\frac{1-e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{\Sigma_{tr,g}^{r,n}}\right) \\ &\quad - \frac{e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{v_g}\int_{s_0^{k,r}}^{s}\left[\frac{d^2\varphi_{g,0}^{k,r,n}}{dt^2} + \frac{d^2 q_g^{r,n}}{d^2t}\left(\frac{e^{\Sigma_{tr,g}^{r,n}\left(s'-s_0^{k,r}\right)}-1}{\Sigma_{tr,g}^{r,n}}\right)\right. \\ &\quad \left. - \frac{1}{v_g}\int_{s_0^{k,r}}^{s'} \frac{d^3\varphi_g^{k,n}(s'')}{dt^3} e^{\Sigma_{tr,g}^{r,n}\left(s''-s_0^{k,r}\right)} ds''\right] ds'.\end{aligned} \tag{5.57}$$

We can evaluate the integral for the second derivative terms, resulting in an equation for the angular flux time derivative in terms of the first and second derivatives of the incoming angular flux and the source along with the third derivative of the angular flux:

$$\begin{aligned}\frac{d\varphi_g^{k,n}(s)}{dt} &= \frac{d\varphi_{g,0}^{k,r,n}}{dt} e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)} + \frac{dq_g^{r,n}}{dt}\left(\frac{1-e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{\Sigma_{tr,g}^{r,n}}\right) \\ &\quad - \left(\frac{d^2\varphi_{g,0}^{k,r,n}}{dt^2} - \frac{1}{\Sigma_{tr,g}^{r,n}}\frac{d^2 q_g^{r,n}}{d^2t}\right)\frac{s-s_0^{k,r}}{v_g} e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)} \\ &\quad - \frac{1}{\Sigma_{tr,g}^{r,n} v_g}\frac{d^2 q_g^{r,n}}{d^2t}\left(\frac{1-e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{\Sigma_{tr,g}^{r,n}}\right) \\ &\quad + \frac{e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{v_g^2}\int_{s_0^{k,r}}^{s}\int_{s_0^{k,r}}^{s'} \frac{d^3\varphi_g^{k,n}(s'')}{dt^3} e^{\Sigma_{tr,g}^{r,n}\left(s''-s_0^{k,r}\right)} ds''\, ds'.\end{aligned} \tag{5.58}$$

For 2SDP, Equation 5.56 is used to propagate the second derivative of the angular flux along the characteristic in addition to Equation 5.58 which is used to propagate the first derivative of the angular flux.

When we substitute Equation 5.58 into Equation 5.3, the result is the characteristic equation for 2SDP:

$$\begin{aligned}
&\frac{d\varphi_g^{k,n}(s)}{ds}+\Sigma_{tr,g}^{r,n}\varphi_g^{k,n}(s)\\
&\qquad=q_g^{r,n}-\frac{1}{v_g}\frac{d\varphi_{g,0}^{k,r,n}}{dt}e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}\\
&\qquad-\frac{1}{v_g}\frac{dq_g^{r,n}}{dt}\left(\frac{1-e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{\Sigma_{tr,g}^{r,n}}\right)\\
&\qquad+\left(\frac{d^2\varphi_{g,0}^{k,r,n}}{dt^2}-\frac{1}{\Sigma_{tr,g}^{r,n}}\frac{d^2q_g^{r,n}}{d^2t}\right)\frac{s-s_0^{k,r}}{v_g^2}e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)} \qquad 5.59\\
&\qquad+\frac{1}{\Sigma_{tr,g}^{r,n}v_g^2}\frac{d^2q_g^{r,n}}{d^2t}\left(\frac{1-e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{\Sigma_{tr,g}^{r,n}}\right)\\
&\qquad-\frac{e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{v_g^3}\int_{s_0^{k,r}}^{s}\int_{s_0^{k,r}}^{s'}\frac{d^3\varphi_g^{k,n}(s'')}{dt^3}e^{\Sigma_{tr,g}^{r,n}\left(s''-s_0^{k,r}\right)}ds''\,ds'.
\end{aligned}$$

This differential equation is solved along a segment like other MOC methods:

$$
\begin{aligned}
\varphi_g^{k,n}(s) = \Bigg[\varphi_g^{k,n}(s) &- \frac{s}{v_g}\left(\frac{d\varphi_{g,0}^{k,r,n}}{dt} - \frac{1}{\Sigma_{tr,g}^{r,n}} \frac{dq_g^{r,n}}{dt} + \frac{1}{\left(\Sigma_{tr,g}^{r,n}\right)^2 v_g} \frac{d^2 q_g^{r,n}}{d^2 t} \right) \\
&+ \frac{s^2}{v_g^2}\left(\frac{d^2\varphi_{g,0}^{k,r,n}}{dt^2} - \frac{1}{\Sigma_{tr,g}^{r,n}} \frac{d^2 q_g^{r,n}}{d^2 t} \right) \Bigg] e^{-\Sigma_{tr,g}^{r,n}\left(s - s_0^{k,r}\right)} \\
&+ \left(q_g^{r,n} - \frac{1}{\Sigma_{tr,g}^{r,n} v_g} \frac{dq_g^{r,n}}{dt} + \frac{1}{\Sigma_{tr,g}^{r,n} v_g^2} \frac{d^2 q_g^{r,n}}{d^2 t} \right) \qquad 5.60 \\
&\times \left(\frac{1 - e^{-\Sigma_{tr,g}^{r,n}\left(s - s_0^{k,r}\right)}}{\Sigma_{tr,g}^{r,n}} \right) \\
&- \frac{e^{-\Sigma_{tr,g}^{r,n}\left(s - s_0^{k,r}\right)}}{v_g^3} \int_{s_0^{k,r}}^{s} \int_{s_0^{k,r}}^{s'} \frac{d^3 \varphi_g^{k,n}(s'')}{dt^3} e^{\Sigma_{tr,g}^{r,n}\left(s'' - s_0^{k,r}\right)} ds''\, ds'.
\end{aligned}
$$

For 2SDP, Equation 5.60 is employed to propagate the angular flux along characteristics. It is also used to calculate the segment-wise average angular flux.

Also, it is noteworthy that if the third derivative term is truncated (i.e. T2SDP), the resulting equation is identical to that of ASTC when the source is assumed to vary quadratically in time; this further reinforces the general relationship between the propagated source methods and the ASTC methods.

While we might consider approximating the second source derivative using backward differences, this requires at minimum the source at two previous points in time. As a result, 2SDP using backward differences would not be self-starting.

5.8 Characteristic Equations for Nth Derivative Propagation (NSDP)

This approach can be extended to still higher derivatives, which results in systematic changes to the characteristic equation for each derivative. We can determine these equations by inspection of the equations for the linear, quadratic, cubic, etc. methods. The equations can be written succinctly using summation syntax.

For an N^{th} source derivative propagation (NSDP) method, the characteristic equation for the segment-wise angular flux in terms of the derivatives of the incoming angular flux and neutron source is:

$$\begin{aligned}\varphi_g^{k,n}(s) = \left(\varphi_g^{k,n}(s) + \Delta\varphi_{g,N}^{k,r,n}(s)\right) e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)} + \left(q_g^{r,n} + \Delta q_{g,N}^{r,n}\right) \\ \times \left(\frac{1-e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{\Sigma_{tr,g}^{r,n}}\right) + \left(\frac{-1}{v_g}\right)^{N+1} e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)} \\ \times \int^{(N+1)} \frac{d^{N+1}\varphi_g^{k,n}(s)}{dt^{N+1}} e^{\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)} ds^{(N+1)},\end{aligned} \qquad 5.61$$

where the $(N+1)$ on the integral and differential indicates $(N+1)$ nested integrals over the segment, and where the coefficients are:

$$\begin{aligned}\Delta\varphi_{g,N}^{k,r,n}(s) \equiv \sum_{i=1}^{N} \frac{1}{i!}\left(\frac{-s}{v_g}\right)^i \left(\frac{d^i\varphi_{g,0}^{k,r,n}}{dt^i} + \sum_{j=i}^{N}\left(\frac{-1}{\Sigma_{tr,g}^{r,n}}\right)^j \left(\frac{1}{v_g}\right)^{j-i} \frac{d^j q_g^{r,n}}{dt^j}\right), \\ \Delta q_{g,N}^{r,n} \equiv \sum_{i=1}^{N}\left(\frac{-1}{\Sigma_{tr,g}^{r,n} v_g}\right)^i \frac{d^i q_g^{r,n}}{dt^i}.\end{aligned} \qquad 5.62$$

Note that when N is zero, the result is the BDC characteristic equation before the angular flux time derivative is approximated.

Similarly, the equation for the i^{th} angular flux time derivative for an N^{th} order source expansion is:

$$\begin{aligned}\frac{d^i\varphi_g^{k,n}(s)}{dt^i} = \left(\frac{d^i\varphi_{g,0}^{k,r,n}}{dt^i} + \frac{\Delta\varphi_{g,N}^{k,r,n}(s)}{\Delta t}\right) e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)} + \left(\frac{d^i q_g^{r,n}}{dt^i} + \frac{\Delta q_{g,N}^{r,n}}{\Delta t}\right) \\ \times \left(\frac{1-e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)}}{\Sigma_{tr,g}^{r,n}}\right) + \left(\frac{-1}{v_g}\right)^{N+1-i} e^{-\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)} \\ \times \int^{(N+1-i)} \frac{d^{N+1-i}\varphi_g^{k,n}(s)}{dt^{N+1}} e^{\Sigma_{tr,g}^{r,n}\left(s-s_0^{k,r}\right)} ds^{(N+1-i)},\end{aligned} \qquad 5.63$$

where the coefficients are:

$$\frac{\Delta\varphi_{g,N}^{k,r,n}(s)}{\Delta t} \equiv \sum_{j=i+1}^{N} \frac{1}{(j-i)!}\left(\frac{-s}{v_g}\right)^{j-i}\left[\frac{d^j\varphi_{g,0}^{k,r,n}}{dt^j} + \sum_{p=j}^{N}\left(\frac{-1}{\Sigma_{tr,g}^{r,n}}\right)^{p+1-j}\left(\frac{1}{v_g}\right)^{p-j}\frac{d^p q_g^{r,n}}{dt^p}\right] \tag{5.64}$$

$$\frac{\Delta q_{g,N}^{r,n}}{\Delta t} \equiv \sum_{j=1}^{N}\left(\frac{-1}{\Sigma_{tr,g}^{r,n} v_g}\right)^{j-i}\frac{d^j q_g^{r,n}}{dt^j}$$

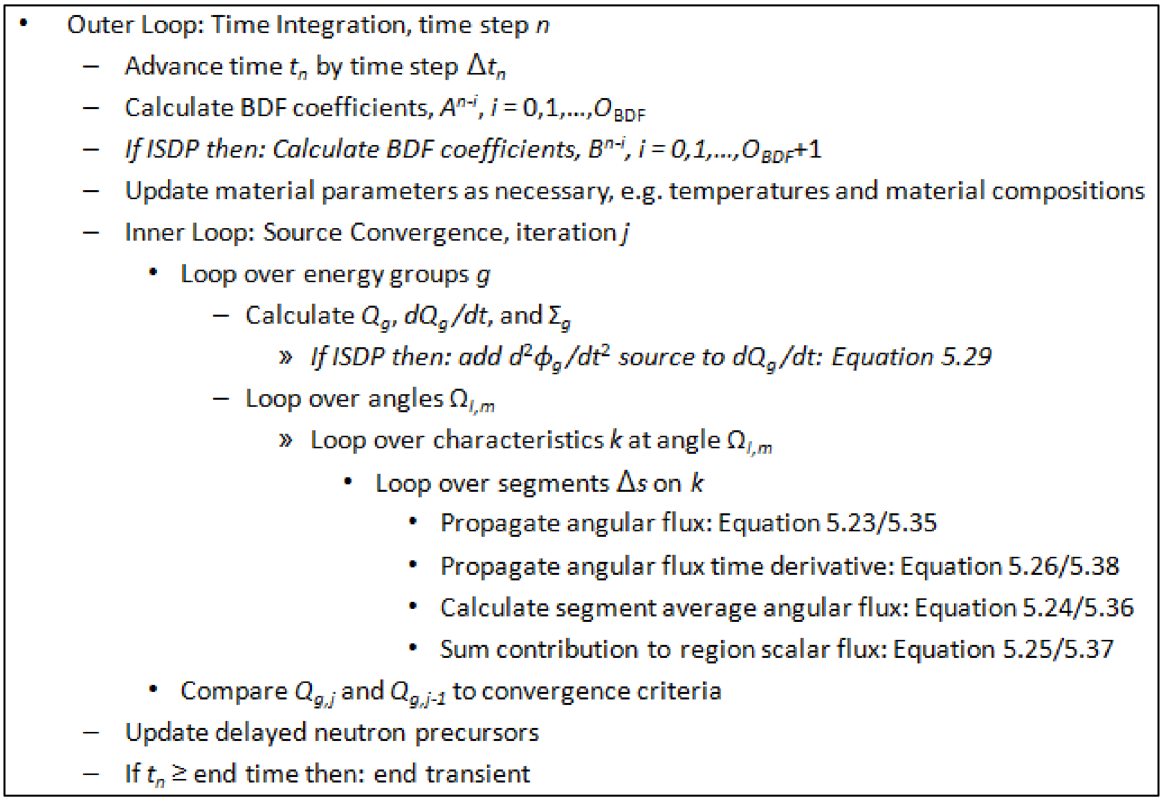
Together, these equations can replicate the characteristic equations N^{th} source derivative propagation methods. However, higher derivative methods are of only theoretical value because they face the same non-self-starting limitation as 2SDP. Nonetheless, we can see that the angular flux time derivative can be understood in terms of the neutron source time derivatives. However, the effect of each higher source time derivative is mitigated by subsequent divisions by the neutron velocity.

5.9 Summary, Algorithm, and Discussion of SDP Methods

The algorithm for a single time step for the SDP methods is similar to that of BDC. One major difference is that the SDP methods require approximations for the source derivatives rather than the angular flux derivative. This is performed when the source terms are being constructed, and it has a limited impact on the total run-time because most of the computational expense is incurred while evaluating quantities along the characteristics. Although the solutions to the SDP method characteristic equations look formidable, many terms are isotropic and defined at the region level rather than the segment level. This allows us to pre-calculate groups of terms while the source is being constructed which drastically simplifies the characteristic equations and reduces the number of arithmetic operations per segment; as a consequence, TSDP and ISDP are about twice as fast as RBDC even though the latter has deceivingly simpler equations. The difference is even more pronounced when high-order backward differences or BDF methods are used to estimate the time derivatives.

Another difference is that the SDP methods evaluate equations to propagate angular flux derivatives along the characteristics. While these equations are relatively simple when we only propagate the first derivatives, they quickly become complex as additional higher derivatives are propagated. As a result, the higher derivative methods are expected to rapidly escalate in computational expense. In addition, the higher derivative methods are not self-starting. As a result, we only investigated first-derivative-based SDP methods.

Otherwise, the SDP and SCDP algorithms are essentially identical to that of the BDC methods; the methods iteratively estimate the scalar flux, use the scalar flux to calculate sources, and propagate angular fluxes along characteristics to update the scalar flux. This process repeats until convergence, at which point the method advances one time step. The algorithm for the SDP methods is summarized in Figure 5-3; the algorithm for SCDP is similar, but involves the calculation and use of the cross section time derivative as well.



```
• Outer Loop: Time Integration, time step $n$
    – Advance time $t_n$ by time step $\Delta t_n$
    – Calculate BDF coefficients, $A^{n-i}$, $i = 0,1,\ldots,O_{BDF}$
    – If ISDP then: Calculate BDF coefficients, $B^{n-i}$, $i = 0,1,\ldots,O_{BDF}+1$
    – Update material parameters as necessary, e.g. temperatures and material compositions
    – Inner Loop: Source Convergence, iteration $j$
        • Loop over energy groups $g$
            – Calculate $Q_g$, $dQ_g/dt$, and $\Sigma_g$
                » If ISDP then: add $d^2\phi_g/dt^2$ source to $dQ_g/dt$: Equation 5.29
            – Loop over angles $\Omega_{l,m}$
                » Loop over characteristics $k$ at angle $\Omega_{l,m}$
                    • Loop over segments $\Delta s$ on $k$
                        • Propagate angular flux: Equation 5.23/5.35
                        • Propagate angular flux time derivative: Equation 5.26/5.38
                        • Calculate segment average angular flux: Equation 5.24/5.36
                        • Sum contribution to region scalar flux: Equation 5.25/5.37
        • Compare $Q_{g,j}$ and $Q_{g,j-1}$ to convergence criteria
    – Update delayed neutron precursors
    – If $t_n \geq$ end time then: end transient
```

Figure 5-3. Nested algorithm of time-dependent MOC with SDP methods

The SDP methods are expected to accurately solve the time-dependent transport equation without necessitating the storage of the angular flux to represent the angular flux time derivative. Although the SDP methods require the truncation or approximation of a higher angular flux time derivative term, the impact of this approximation is mitigated by the repeated division of these terms by the neutron velocity. Like IBDC, the SDP methods will not converge to the solution of the time-dependent Boltzmann equation as the time step is reduced to zero; the SDP methods may nonetheless be accurate for transients reasonably large time steps.

A secondary benefit of the SDP methods is that they recast the angular flux time derivative in terms of a source derivative. The source derivatives can be approximated to high-order accuracy using backward differences with limited computational and memory expense; this stands in contrast to methods for approximating the angular flux time derivative to a high-order accuracy, which are computationally- and/or memory-intensive.

We analyze SDP and SCDP methods in Chapter 6 and numerically test these methods in Chapter 8. We provide the details of their implementation in DeCART in Chapter 7.

Chapter 6
Error Analysis of Angular Flux Time Derivative Approximations

6.1 Introduction

In this chapter, we will provide brief, simplified error analyses for the angular flux time derivative options used in the time-dependent Method of Characteristics (MOC) methods is in this thesis. The purpose of this chapter is to identify the sources of error that explain the differences between the methods. This chapter will focus on the angular flux time derivatives specifically rather than the solutions for the angular flux (which are a function of the time derivative) to limit complexity. Specifically, we will derive an equation for each method for the angular flux time derivative that clearly identifies the leading error terms in time, space, and angle.

The purpose of this chapter is not to provide a detailed account of all of the sources of error introduced by MOC when solving the transport equation, but only to provide insight into the sources of error that are introduced by our treatment of the angular flux time derivative. To limit the complexity of this chapter, we will ignore the sources of error inherent in the approximations used by MOC (e.g. angular error resulting from the use of discrete ordinates, energy error from the use of the multi-group approximation, spatial error from the use of the flat source approximation, etc.). Further, since the error accumulates as we advance along segments and over time steps, we will limit the analysis to the first segment and time step to limit the complexity. Finally, we assumed that the methods begin from a critical, steady-state configuration because this is representative of the reactor transients of interest.

We will rely on Taylor series expansions for this error analysis. First, this chapter will review the Taylor series expansion to illustrate how it can be used to understand the error in the approximation of a time derivative for a differential equation.

All of the time-dependent MOC methods in this work can approximate time derivatives to high-order accuracy using backward differences. However, rather than discussing high-order approximations in the context of each time-dependent MOC method, we will discuss the use of backward differences to approximate time derivatives more generally. With the understanding that this concept applies to each of the time-dependent MOC methods, we can limit the analysis for each unique method to the first-order backward difference method.

We will begin the error analysis sections with the reference time-dependent MOC method (RBDC) derived in Chapter 4. This method stores the segment-wise average angular flux to represent the angular flux time derivative, which require extensive memory. Following RBDC we will analyze the time-dependent MOC method with an isotropic approximation for the time derivative (IBDC), which was also derived in Chapter 4. This method is computationally- and memory-efficient, but it introduces a new angular and spatial error term.

Finally we will evaluate the three new angular flux time derivative propagation (TDP) methods tested in this work. The first is the Source Derivative Propagation method with a Truncated second-derivative term (TSDP), the second is the Source Derivative Propagation method with an isotropic second-derivative term (ISDP), and the third is the Source and Cross Section Derivative Propagation method with Isotropic Correction (ISCDP). These methods were derived in Chapter 5. The error analysis for the latter methods builds upon previous methods.

6.2 Review of Taylor Series Expansions for Error Analysis

Taylor series expansions are a useful tool for understanding the error that is introduced when finite differences are used to approximate derivatives. In this section we will limit our consideration to time derivatives because we treat spatial dependence using MOC in this work. We begin with the assumption that the variable in question y is continuously differentiable and continuous near a point in time t_n. The variable near that point in time can be represented by an infinite series of higher derivatives:

$$y(t) = y(t_n) + (t - t_n)\frac{dy(t_n)}{dt} + \frac{(t - t_n)^2}{2!}\frac{d^2y(t_n)}{dt^2} + \sum_{i=3}^{\infty}\frac{(t - t_n)^i}{i!}\frac{d^iy(t_n)}{dt^i}. \tag{6.1}$$

In practice it is usually necessary to truncate a Taylor series expansion at some derivative, resulting in truncation error that scales with that derivative. For example, if we truncate the series at the third derivative, the approximation includes an error term that scales with the third power of the time:

$$y(t) = y(t_n) + (t - t_n)\frac{dy(t_n)}{dt} + \frac{(t - t_n)^2}{2!}\frac{d^2y(t_n)}{dt^2} + O(t^3), \tag{6.2}$$

where the O notation indicates the order of the leading error term.

Backward Euler is a simple method for time integration. In this method, the time derivative is approximated using a backward difference:

$$\frac{dy_n}{dt} \approx \frac{y_n - y_{n-1}}{\Delta t}, \tag{6.3}$$

where the subscripts indicate the time (e.g. $y_n \equiv y(t_n)$) and the time step is defined as $\Delta t_n \equiv t_n - t_{n-1}$.

We can show that Backward Euler is first-order accurate in time using the Taylor series expansion near t_n:

$$y_{n-1} = y_n - \Delta t_n\frac{dy_n}{dt} + \frac{\Delta t_n^2}{2!}\frac{d^2y_n}{dt^2} + O(\Delta t_n^3). \tag{6.4}$$

Using algebra we isolate the first derivative:

$$\frac{dy_n}{dt} = \frac{y_n - y_{n-1}}{\Delta t_n} + \frac{\Delta t_n}{2!}\frac{d^2y_n}{dt^2} + O(\Delta t_n^2). \tag{6.5}$$

We see that the first term on the right hand side is the approximation used for Backward Euler, while the following terms are the error. Since the leading error term is proportional to the time step, Backward Euler is first-order accurate in time.

6.3 High-Order Accurate Approximations using Backward Differences

We can approximate derivatives to high-order accuracy by using backward differences to approximate higher derivatives. This is the basis for the time integration method called the Backward Differentiation Formula (BDF). We approximate the higher derivatives using successive backward differences in terms of lower derivatives. For example, we approximate the second derivative in terms of the backward difference of the first derivatives, and then we approximate the first derivatives in terms of the solution variable:

$$\frac{d^2y_n}{dt^2} \approx \frac{\frac{dy_n}{dt} - \frac{dy_{n-1}}{dt}}{\Delta t_n} \approx \frac{\frac{y_n - y_{n-1}}{\Delta t_n} - \frac{y_{n-1} - y_{n-2}}{\Delta t_{n-1}}}{\Delta t_n}. \quad 6.6$$

When we apply this to the higher derivatives for the Taylor series expansion, we can derive a time integration method with accuracy to the selected order. For example, to derive the second-order accurate BDF method, we substitute Equation 6.6 into Equation 6.5:

$$\begin{aligned} \frac{dy_n}{dt} &= \frac{y_n - y_{n-1}}{\Delta t} + \frac{1}{2}\left(\frac{y_n - y_{n-1}}{\Delta t} - \frac{y_{n-1} - y_{n-2}}{\Delta t}\right) + O(\Delta t^2) \\ &= \frac{\frac{3}{2}y_n - 2y_{n-1} + \frac{1}{2}y_{n-2}}{\Delta t} + O(\Delta t^2), \end{aligned} \quad 6.7$$

where we have assumed a constant time step for simplicity (i.e. $\Delta t \equiv \Delta t_n = \Delta t_{n-1}$).

Higher-order methods are derived similarly by eliminating additional higher derivatives. Although BDF methods may be derived to arbitrary order, only methods with orders six and lower have good stability properties. A discussion of BDF stability is beyond the scope of this thesis. The leading temporal error for BDF order *N* is given by:

$$O(\Delta t^N) = (-1)^{N+1} \frac{\Delta t^N}{(N+1)!} \frac{d^{N+1}y_n}{dt^{N+1}} + O(\Delta t^{N+1}), \quad 6.8$$

where we should also note that high-order methods can be less accurate than low-order methods if the time step is very large.

We can assess the temporal order of convergence by defining a reference solution and calculating the “error” in an approximate solution as a function of the time step size. For an N^{th} order method the error should scale with the N^{th} power of the time step size, which is difficult to identify on a plot. If we take a logarithm of the error equation (or plot the error on a log-log plot), the error will scale linearly with slope N:

$$\log\big(\epsilon(\Delta t)\big) = N\log(\Delta t) + \log\left(\frac{(-1)^{N+1}}{(N+1)!}\frac{d^{N+1}y_n}{dt^{N+1}}\right), \qquad 6.9$$

where $\epsilon(\Delta t)$ is the error as a function of time step size. We will use this observation in Chapter 8 to determine the order of convergence to the reference solution for high-order time integration methods.

All of the methods in this thesis may be employed with a user-selected order of accuracy by using backward differences to approximate the time derivatives. However, for succinctness, we will perform the method-specific error analysis using the first-order backward difference method:

$$\frac{d\varphi_k^n(s)}{dt} = \frac{\varphi_k^n(s) - \varphi_k^{n-1}(s)}{\Delta t} + O(\Delta t), \qquad 6.10$$

where φ is the angular flux, k is the index for the characteristic, s is the spatial position along the characteristic, and the energy group index is suppressed for clarity. As a reminder, note that the characteristic index contains an embedded angular definition (i.e. $k = f\big(\mathbf{\Omega}_{l,m}\big)$ where l and m are the indices for the azimuthal and polar angles). This will serve as the starting point for the analysis in subsequent sections. Our goal is to derive an equation for the angular flux time derivative for each method that clearly identifies the error terms. Error Analysis for RBDC

The first method we consider is RBDC, which is the reference MOC solution. In this method, we approximate the angular flux time derivative at the segment-wise average angular flux:

$$\frac{d\varphi_k^n(s)}{dt} \approx \frac{\bar{\varphi}_{k,r}^n - \bar{\varphi}_{k,r}^{n-1}}{\Delta t}, \qquad 6.11$$

where $\bar{\varphi}_{k,r}^n$ indicates the segment-wise average angular flux for characteristic k in region r at time t_n when s is within r.

Our goal is to rewrite Equation 6.10 in terms of the approximation expressed in Equation 6.11 plus any error terms. We note that the segment-wise average angular flux is related to the spatially-dependent angular flux by the integral:

$$\bar{\varphi}_{k,r}^{n} \equiv \frac{1}{\Delta s}\int_{0}^{\Delta s} \varphi_{k}^{n}(s)\, ds, \tag{6.12}$$

where we have arbitrarily defined the incoming position of the characteristic in the region to be $s = 0$ in this chapter for succinctness, and Δs is the length of the segment.

We can relate the segment-wise angular flux to the segment-wise average angular flux using a simple difference:

$$\varphi_{k}^{n}(s) \equiv \bar{\varphi}_{k,r}^{n} - \delta_{k}^{n}(s). \tag{6.13}$$

Figure 6-1 shows the relationship between the angular flux, average angular flux, and the difference for a representative segment at steady-state. This segment is characterized by a strong incoming angular flux relative to the source, resulting in a solution that is dominated by a negative exponential function. The reverse situation has a $1 - e^{-\Sigma s}$ shape.

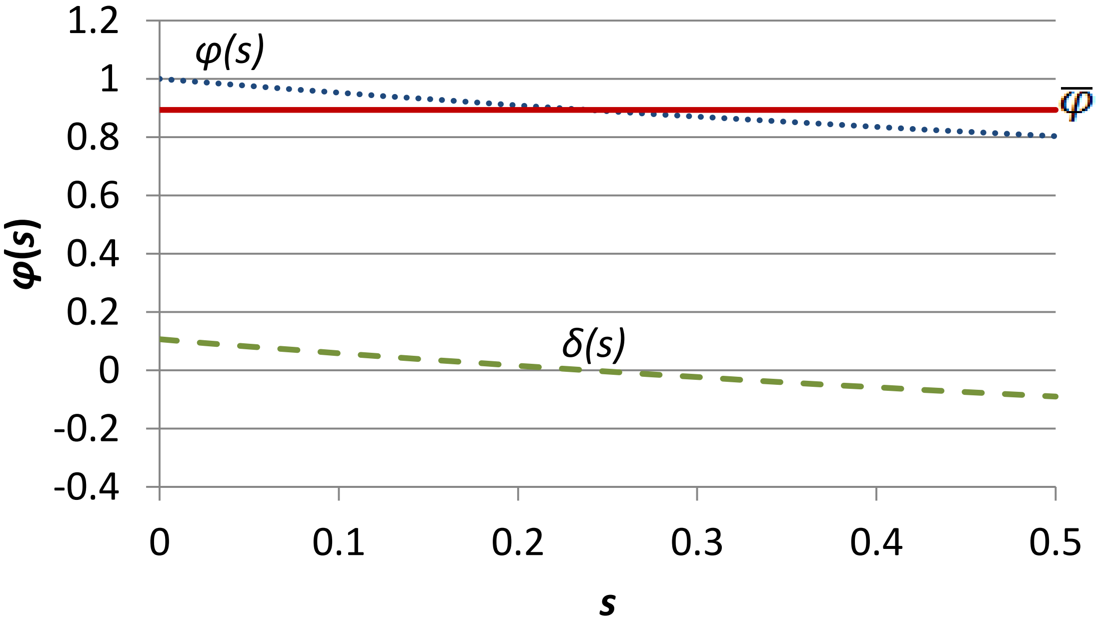


Figure 6-1. Angular flux for a representative segment

We can rewrite Equation 6.10 using Equation 6.13:

$$\frac{d\varphi_k^n(s)}{dt} = \frac{\bar{\varphi}_{k,r}^n - \bar{\varphi}_{k,r}^{n-1}}{\Delta t} + \frac{\delta_k^n(s) - \delta_k^{n-1}(s)}{\Delta t} + O(\Delta t). \tag{6.14}$$

The error introduced by this approximation is related to how the difference is changing in time; if the difference is constant, the approximation introduces no additional error.

If we assume that a transient begins from a steady-state condition, we can derive the time-dependent MOC angular flux for an arbitrary time step which is shown in Appendix C. However, these equations are complex, and using the first time step as an example is adequate for this chapter.

To evaluate the new error term, we will rewrite it in terms of the angular fluxes and average angular fluxes for the first time step (t_1) and steady-state (t_0):

$$\frac{d\varphi_k^1(s)}{dt} = \frac{\bar{\varphi}_{k,r}^1 - \bar{\varphi}_{k,r}^0}{\Delta t} + \frac{1}{\Delta t}\left(\varphi_k^1(s) - \varphi_k^0(s) - \bar{\varphi}_{k,r}^1 + \bar{\varphi}_{k,r}^0\right) + O(\Delta t). \tag{6.15}$$

We will substitute the equations for the angular flux terms in parentheses. The steady-state angular flux was derived in Chapter 3 and is:

$$\varphi_k^0(s) = \varphi_k^0(0)e^{-\Sigma_r^0 s} + \frac{q_r^0}{\Sigma_r^0}\left(1 - e^{-\Sigma_r^0 s}\right), \tag{6.16}$$

while the steady-state average angular flux is:

$$\bar{\varphi}_k^0 = \left(\varphi_k^0(0) - \frac{q_r^0}{\Sigma_r^0}\right)\left(\frac{1 - e^{-\Sigma_r^0 \Delta s}}{\Sigma_r^0 \Delta s}\right) + \frac{q_r^0}{\Sigma_r^0}. \tag{6.17}$$

The angular flux at the end of the first time step is derived in Appendix C by using Backward Euler and using Equation 6.16 for the segment-wise angular flux:

$$\begin{aligned}\varphi_k^1(s) = \varphi_k^1(0)e^{-\hat{\Sigma}_r^1 s} &+ \frac{q_r^1 + \frac{1}{v\Delta t}\frac{q_r^0}{\Sigma_r^0}}{\hat{\Sigma}_r^1}\left(1 - e^{-\hat{\Sigma}_r^1 s}\right) \\ &+ \frac{1}{v\Delta t}\left(\varphi_k^0(0) - \frac{q_r^0}{\Sigma_r^0}\right)\left(\frac{e^{-\Sigma_r^0 s} - e^{-\hat{\Sigma}_r^1 s}}{\hat{\Sigma}_r^1 - \Sigma_r^0}\right),\end{aligned} \tag{6.18}$$

which results in a segment-wise average angular flux of:

$$\bar{\varphi}_k^1 = \left(\varphi_k^1(0) - \frac{q_r^1 + \frac{1}{v\Delta t}\frac{q_r^0}{\Sigma_r^0}}{\hat{\Sigma}_r^1}\right)\left(\frac{1 - e^{-\hat{\Sigma}_r^1 \Delta s}}{\hat{\Sigma}_r^1 \Delta s}\right)$$
$$+ \frac{1}{v\Delta t}\left(\varphi_k^0(0) - \frac{q_r^0}{\Sigma_r^0}\right)\left[\frac{\left(\hat{\Sigma}_r^1 - \Sigma_r^0\right) - \left(\hat{\Sigma}_r^1 e^{-\Sigma_r^0 \Delta s} - \Sigma_r^0 e^{-\hat{\Sigma}_r^1 \Delta s}\right)}{\left(\hat{\Sigma}_r^1 - \Sigma_r^0\right)\Sigma_r^0 \hat{\Sigma}_r^1 \Delta s}\right], \qquad 6.19$$

where:

$$\hat{\Sigma}_r^1 = \Sigma_r^1 + \frac{1}{v\Delta t}. \qquad 6.20$$

If we substitute Equations 6.16—6.19 into Equation 6.14, we find that the error introduced by the RBDC1 approximation is:

$$\frac{d\varphi_k^1(s)}{dt} = \frac{\bar{\varphi}_{k,r}^1 - \bar{\varphi}_{k,r}^0}{\Delta t}$$
$$+ \frac{1}{\Delta t}\left[\varphi_k^1(0) e^{-\hat{\Sigma}_r^1 s} + \frac{q_r^1 + \frac{1}{v\Delta t}\frac{q_r^0}{\Sigma_r^0}}{\hat{\Sigma}_r^1}\left(1 - e^{-\hat{\Sigma}_r^1 s}\right)\right.$$
$$+ \frac{1}{v\Delta t}\left(\varphi_k^0(0) - \frac{q_r^0}{\Sigma_r^0}\right)\left(\frac{e^{-\Sigma_r^0 s} - e^{-\hat{\Sigma}_r^1 s}}{\hat{\Sigma}_r^1 - \Sigma_r^0}\right) - \varphi_k^0(0) e^{-\Sigma_r^0 s}$$
$$- \frac{q_r^0}{\Sigma_r^0}\left(1 - e^{-\Sigma_r^0 s}\right) - \left(\varphi_k^1(0) - \frac{q_r^1 + \frac{1}{v\Delta t}\frac{q_r^0}{\Sigma_r^0}}{\hat{\Sigma}_r^1}\right)\left(\frac{1 - e^{-\hat{\Sigma}_r^1 \Delta s}}{\hat{\Sigma}_r^1 \Delta s}\right) \qquad 6.21$$
$$- \frac{1}{v\Delta t}\left(\varphi_k^0(0) - \frac{q_r^0}{\Sigma_r^0}\right)\left(\frac{\left[\hat{\Sigma}_r^1 - \Sigma_r^0\right] - \left[\hat{\Sigma}_r^1 e^{-\Sigma_r^0 \Delta s} - \Sigma_r^0 e^{-\hat{\Sigma}_r^1 \Delta s}\right]}{\left[\hat{\Sigma}_r^1 - \Sigma_r^0\right]\Sigma_r^0 \hat{\Sigma}_r^1 \Delta s}\right)$$
$$\left. + \left(\varphi_k^0(0) - \frac{q_r^0}{\Sigma_r^0}\right)\left(\frac{1 - e^{-\Sigma_r^0 \Delta s}}{\Sigma_r^0 \Delta s}\right) + \frac{q_r^0}{\Sigma_r^0}\right] + O(\Delta t).$$

We note that all of the spatially-dependent terms are described using exponential functions. We can expand all the exponential functions using truncated power series:

$$\begin{aligned} e^{-\Sigma_r s} &= 1 - \Sigma_r s + O(s^2), \\ e^{-\Sigma_r \Delta s} &= 1 - \Sigma_r \Delta s + O(\Delta s^2). \end{aligned} \tag{6.22}$$

With this expansion, many terms cancel and we can identify the leading linear spatial error term:

$$\begin{aligned} \frac{d\varphi_k^1(s)}{dt} = \frac{\bar{\varphi}_{k,r}^1 - \bar{\varphi}_{k,r}^0}{\Delta t} + \frac{s}{\Delta t}\left[\left(\frac{q_r^1}{\hat{\Sigma}_r^1} - \varphi_k^1(0)\right)\hat{\Sigma}_r^1 - \left(\frac{q_r^0}{\hat{\Sigma}_r^0} - \varphi_k^0(0)\right)\hat{\Sigma}_r^0\right] + O(s^2) \\ + O(\Delta s^2) + O(\Delta t), \end{aligned} \tag{6.23}$$

where:

$$\hat{\Sigma}_r^0 = \Sigma_r^0 + \frac{1}{v\Delta t}. \tag{6.24}$$

We can rewrite the leading error term using Taylor series expansions in terms of the source and incoming angular flux:

$$\begin{aligned} \frac{d\varphi_k^1(s)}{dt} = \frac{\bar{\varphi}_{k,r}^1 - \bar{\varphi}_{k,r}^0}{\Delta t} + s\left[\frac{dq_r^1}{dt} - \frac{d}{dt}(\varphi_k^1(0)\Sigma_r^1) - \frac{1}{v\Delta t}\frac{d\varphi_k^1(0)}{dt}\right] + O(s\Delta t) \\ + O(s^2) + O(\Delta s^2) + O(\Delta t). \end{aligned} \tag{6.25}$$

This demonstrates that the error in the RBDC1 approximation is proportional to the position along the segment in the region; as the segment length goes to zero, the error also goes to zero. This is consistent with other sources of error in MOC when step characteristics is used (i.e. the flat source and flat cross section approximation). It is also clear that the method is more accurate when the angular flux is relatively flat across the segment (i.e. $\frac{q_r^n}{\hat{\Sigma}_r^n} - \varphi_k^n(0) \approx 0$) for both time steps.

We can determine the error that the approximation to the angular flux time derivative contributes to the error to the angular flux by substituting Equation 6.21 into Equation 5.7:

$$\begin{aligned}\varphi_g^{k,1}(s) = \left[\varphi_g^{k,1}(s)\right]_{\mathrm{RBDC1}} \\ -\frac{e^{-\Sigma_r^1 s}}{v}\int_0^s \Bigg(s'\left[\frac{dq_r^1}{dt} - \frac{d}{dt}(\varphi_k^1(0)\Sigma_r^1) - \frac{1}{v\Delta t}\frac{d\varphi_k^1(0)}{dt}\right] + \frac{\Delta t}{2}\frac{d^2\bar{\varphi}_k^1}{dt^2} \\ + O(s'^2) + O(\Delta s^2) + O(s'\Delta t) + O(\Delta t^2)\Bigg) e^{\Sigma_r^1 s'} ds',\end{aligned} \qquad 6.26$$

where $\left[\varphi_g^{k,1}(s)\right]_{\mathrm{RBDC1}}$ is the angular flux calculated using RBDC1 (i.e. Equation 4.30) and we have expanded the $O(\Delta t)$ error using Equation 6.8:

$$O(\Delta t) = \frac{\Delta t}{2!}\frac{d^2\bar{\varphi}_k^1}{dt^2} + O(\Delta t^2). \qquad 6.27$$

To remain consistent with our treatment of exponentials in this chapter, we approximate the exponentials using a truncated Taylor series and evaluate the integral:

$$\begin{aligned}\varphi_g^{k,1}(s) = \left[\varphi_g^{k,1}(s)\right]_{\mathrm{RBDC1}} + \frac{s\Delta t}{2v}\frac{d^2\bar{\varphi}_k^1}{dt^2} \\ -\frac{s^2}{v}\left[\frac{dq_r^1}{dt} - \frac{d}{dt}(\varphi_k^1(0)\Sigma_r^1) - \frac{1}{v\Delta t}\frac{d\varphi_k^1(0)}{dt}\right] + O(s^3) \\ + O(s\Delta s^2) + O(s^2\Delta t) + O(\Delta t^2).\end{aligned} \qquad 6.28$$

This illustrates that the leading spatial error term for RBDC is linear and proportional to the leading temporal error, which is related to the order of BDF used to approximate the time derivative. The quadratic spatial error term is related to the “flatness” of the angular flux time derivative along the segment, which is the basic assumption of RBDC.

Although this chapter focuses on the first-order temporal methods (e.g. RBDC1, IBDC1, etc.), we will provide one example of a high-order method. If BDFN was used to approximate the angular flux time derivative for RBDC, the $O(\Delta t)$ error term in Equation 6.25 would be replaced with an $O(\Delta t^N)$ error term based on Equation 6.8:

$$O(\Delta t^N) = \frac{\Delta t^N}{(N+1)!}\frac{d^{N+1}\bar{\varphi}_k^n}{dt^{N+1}} + O(\Delta t^{N+1}). \qquad 6.29$$

We can substitute Equation 6.29 into Equation 5.7 to identify the error for RBDC*N*:

$$\begin{aligned}\varphi_g^{k,1}(s) = \left[\varphi_g^{k,1}(s)\right]_{\mathrm{RBDC}N} + \frac{s\Delta t^N}{(N+1)!\,v}\frac{d^{N+1}\bar{\varphi}_k^1}{dt^{N+1}} \\ -\frac{s^2}{v}\left[\frac{dq_r^1}{dt} - \frac{d}{dt}(\varphi_k^1(0)\Sigma_r^1) - \frac{1}{v\Delta t}\frac{d\varphi_k^1(0)}{dt}\right] + O(s^3) \\ + O(s\Delta s^2) + O(s^2\Delta t^N) + O(\Delta t^{N+1}).\end{aligned} \quad 6.30$$

Thus we see that the principal difference in the error between RBDC1 and RBDC*N* lies in the leading temporal error term. When the time step is small, the higher-order BDF methods are more accurate than the lower-order BDF methods because the temporal error term is smaller. However, the high-order methods are often suspect because they may introduce instability under some circumstances. In Chapter 8 we will use RBDC1 with a very small time step to generate reference solutions for this reason, but it is worth bearing in mind that the high-order methods may nonetheless be more accurate than the reference solution. Expressed as an equation, this condition would be:

$$\frac{s\Delta t_{\mathrm{REF}}}{2v}\frac{d^2\bar{\varphi}_k^n}{dt^2} > \frac{s\Delta t^N}{(N+1)!\,v}\frac{d^{N+1}\bar{\varphi}_k^n}{dt^{N+1}}, \quad 6.31$$

where Δt_{REF} is the fine time step used to generate the reference solution and Δt is a larger time step for an N^{th}order solution.

6.4 Error Analysis for IBDC

In IBDC, the segment-wise angular flux time derivative is approximated using the region-wise scalar flux time derivative for the region where the segment lies:

$$\frac{d\varphi_k^n(s)}{dt} \approx \frac{\phi_r^n - \phi_r^{n-1}}{4\pi\Delta t}. \quad 6.32$$

This approximation introduces additional spatial and angular error. However, error analysis for IBDC is more difficult than for RBDC because the scalar flux is defined on a different spatial mesh than the segment-wise angular flux. In MOC, we calculate the region-wise scalar flux by numerically integrating the segment-wise average angular fluxes over angle and space within the region. We will use this process to understand the error, building on the result from the previous section.

The first step to calculating the scalar flux is to find the region-wise angular flux. The region-wise angular flux is calculated by integrating the segment-wise average angular flux over space for every segment within a region at the same angle:

$$\bar{\varphi}^{n}_{r,l,m} = \int_{\mathbf{x}\in r} \bar{\varphi}^{n}_{k,r} d\mathbf{x} \approx \sum_{k\in r} w_{k,r}\bar{\varphi}^{n}_{k,r}, \qquad 6.33$$

where the weighting factor $w_{k,r}$ is based on the fractional volume represented by the segment within the region and depends on the specific geometry and dimensionality.

Figure 6-2 shows seven characteristic segments within a region at the same angle ($\mathbf{\Omega}_{l,m}$). The segments are separated by an angle-dependent ray spacing of ΔK_l. The angular flux along each segment is assumed to be representative of a band of space around the segment with a half-width of $\frac{\Delta K_l}{2}$.

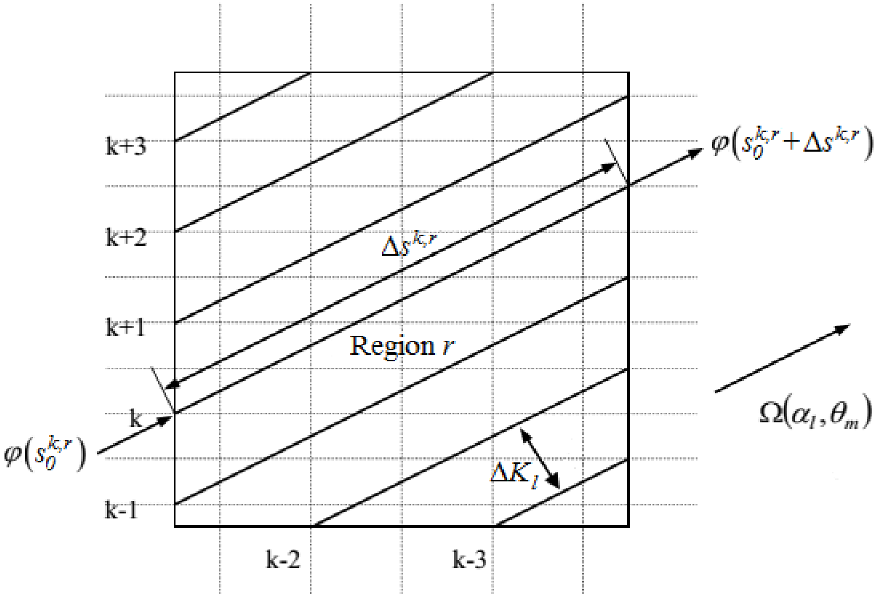


Figure 6-2. Example of characteristic segments within a region [adapted from Hur08]

When we approximated the angular flux time derivative using the segment-wise average angular flux, we introduced an error that was proportional to the position along the segment s. When we integrate in space perpendicular to the segment within the region to calculate the region-wise angular flux, we expect to introduce an error that is proportional to some effective width of the region $W_{k,r}$. Deriving this width in a general fashion is complex because regions are not regularly-shaped, and the segments are not of constant length. Note that this error is not diminished by reducing the ray spacing interval, but it is diminished as the regions become smaller. A more detailed investigation of this error is beyond the scope of this chapter, and we will be satisfied designating this error as $O\left(W_{k,r}\right)$:

$$\frac{d\varphi_k^1(s)}{dt} = \frac{\bar{\varphi}_{r,l,m}^1 - \bar{\varphi}_{r,l,m}^0}{\Delta t} + O\left(W_{k,r}\right) + O(s) + O(\Delta s^2) + O(\Delta t), \qquad 6.34$$

The next step to determine the region-wise scalar flux is to integrate the region-wise angular flux over angle:

$$\phi_r^n = \int_{4\pi} \bar{\varphi}_{r,l,m}^n d\mathbf{\Omega} \approx \sum_{l,m} w_{l,m} \bar{\varphi}_{r,l,m}^n, \qquad 6.35$$

If we ignore the error resulting from the use of a finite number of discrete ordinates (which is inherent in MOC), we can relate the region-wise scalar flux to the region-wise angular flux using spherical harmonics expansions:

$$\bar{\varphi}_{r,l,m}^n = \frac{\phi_r^n}{4\pi} + \frac{3\mathbf{\Omega}_{l,m} \cdot \boldsymbol{J}_r^n}{4\pi} + \sum_{i=2}^{\infty} \sum_{j=-i}^{i} \varphi_{r,i,j} Y_i^j\left(\mathbf{\Omega}_{l,m}\right), \qquad 6.36$$

where $\boldsymbol{J}_r^n$ are the neutron currents for region r at time t_n, $\varphi_{r,i,j}$ are the moments of the angular flux order two and higher, and Y_i^j are the spherical harmonics functions for order two and higher. The scalar flux and neutron current are the first two moments of the angular flux, and we have separated these terms from the summation so we can relate the scalar flux and angular flux to understand the error.

If we apply Equation 6.36 to Equation 6.34, we can see that the scalar flux approximation introduces an error term proportional to region-wise neutron currents:

$$\frac{d\varphi_k^1(s)}{dt} = \frac{\phi_r^1 - \phi_r^0}{4\pi\Delta t} + \frac{3\mathbf{\Omega}_{l,m}}{4\pi} \cdot \left(\frac{\boldsymbol{J}_r^1 - \boldsymbol{J}_r^0}{\Delta t}\right) + O\left(Y_2^j(\mathbf{\Omega}_{l,m})\right) + O(W_{k,r}) + O(s) \\ + O(\Delta s^2) + O(\Delta t). \tag{6.37}$$

If we use a Taylor series expansion of the region-wise neutron current in time:

$$\boldsymbol{J}_r^0 = \boldsymbol{J}_r^1 - \Delta t \frac{d\boldsymbol{J}_r^1}{dt} + O(\Delta t^2), \tag{6.38}$$

we can rewrite Equation 6.37 in terms of the neutron current time derivative, with the new error terms for IBDC in brackets:

$$\frac{d\varphi_k^1(s)}{dt} = \frac{\phi_r^1 - \phi_r^0}{4\pi\Delta t} + \left[\frac{3\mathbf{\Omega}_{l,m}}{4\pi} \cdot \frac{d\boldsymbol{J}_r^1}{dt} + O\left(Y_2^j(\mathbf{\Omega}_{l,m})\right) + O(W_{k,r})\right] + O(s) \\ + O(\Delta s^2) + O(\Delta t). \tag{6.39}$$

Equation 6.39 identifies the new error terms that results from the isotropic approximation of the angular flux time derivative using the region-wise scalar flux. As the regions become smaller, the $O(W_{k,r})$ error term diminishes, and the primary new source of error is proportional to the time derivative of the neutron current and other high order moments of the angular flux.

This analysis indicates that, when the regions are sufficiently small, the scalar flux approximation is accurate provided that the angular dependence of the neutron flux is changing slowly. This explains why IBDC methods have been observed to be accurate for many transients [Tsu13, Hof13b, Tal13]. This is also observed in Chapter 8.

The error in the angular flux solution for IBDC can be determined by substituting Equation 6.39 into Equation 5.7:

$$\varphi_g^{k,1}(s) = \left[\varphi_g^{k,1}(s)\right]_{\text{IBDC1}} \\ - \frac{s}{v}\left[\frac{3\mathbf{\Omega}_{l,m}}{4\pi} \cdot \frac{d\boldsymbol{J}_r^1}{dt} + O\left(Y_2^j(\mathbf{\Omega}_{l,m})\right) + O(W_{k,r}) + \frac{\Delta t}{2}\frac{d^2\bar{\varphi}_k^1}{dt^2}\right] \\ + O(s^2) + O(s\Delta s^2) + O(s^2\Delta t) + O(\Delta t^2). \tag{6.40}$$

Like RBDC, the leading spatial error term for IBDC is linear and contains the leading temporal error term. However, it also contains an error term based on the neutron current time derivative and region size, and these error terms will not diminish as the time step is refined.

6.5 Error Analysis for TSDP

The error analysis for the SDP methods is more straightforward because the error terms are evident in the derivation. For TSDP, the angular flux time derivative is approximated using Equation 5.17:

$$\frac{d\varphi_k^n(s)}{dt} \approx \frac{d\varphi_k^n(0)}{dt} e^{-\Sigma_r^n s} + \frac{q_r^n - q_r^{n-1}}{\Delta t}\left(\frac{1-e^{-\Sigma_r^n s}}{\Sigma_r^n}\right), \qquad 6.41$$

where q_r^n is the total neutron source and Σ_r^n is the macroscopic transport cross section.

Equation 6.41 is derived by starting with the partially-integrated characteristic equation for time-dependent neutron transport before the angular flux time derivative is approximated:

$$\varphi_k^n(s) = \varphi_k^n(0)e^{-\Sigma_r^n s} + q_r^n\left(\frac{1-e^{-\Sigma_r^n s}}{\Sigma_r^n}\right) - \frac{e^{-\Sigma_r^n s}}{v}\int_0^s \frac{d\varphi_k^n(s')}{dt} e^{\Sigma_r^n s'} ds'. \qquad 6.42$$

The equation for the time-dependent angular flux is derived by taking a derivative of Equation 6.42 with respect to time:

$$\begin{aligned}\frac{d\varphi_k^n(s)}{dt} = {} & \frac{d\varphi_k^n(0)}{dt}\, e^{-\Sigma_r^n s} + \frac{dq_r^n}{dt}\left(\frac{1-e^{-\Sigma_r^n s}}{\Sigma_r^n}\right) \\ & + \left[\varphi_k^n(0)\frac{d}{dt}\left(e^{-\Sigma_r^n s}\right) - q_r^n \frac{d}{dt}\left(\frac{1-e^{-\Sigma_r^n s}}{\Sigma_r^n}\right)\right. \\ & - \frac{d}{dt}\left(\frac{e^{-\Sigma_r^n s}}{v}\right)\int_0^s \frac{d\varphi_k^n(s')}{dt} e^{\Sigma_r^n s'} ds' \\ & - \frac{e^{-\Sigma_r^n s}}{v}\int_0^s \frac{d\varphi_k^n(s')}{dt}\frac{d}{dt}\left(e^{\Sigma_r^n s}\right) ds' \\ & \left. - \frac{e^{-\Sigma_r^n s}}{v}\int_0^s \frac{d^2\varphi_k^n(s')}{dt^2} e^{\Sigma_r^n s'} ds'\right],\end{aligned} \qquad 6.43$$

where the terms in square brackets are truncated for TSDP.

The source time derivative is approximated using backward differences, which results in a time-dependent error. For consistency with the earlier section in this chapter, we will assume a first-order backward difference approximation:

$$\begin{aligned}\frac{d\varphi_k^n(s)}{dt} = {} & \frac{d\varphi_k^n(0)}{dt}\, e^{-\Sigma_r^n s} + \frac{q_r^n - q_r^{n-1}}{\Delta t}\left(\frac{1-e^{-\Sigma_r^n s}}{\Sigma_r^n}\right) \\ & + \left[\varphi_k^n(0)\frac{d}{dt}\left(e^{-\Sigma_r^n s}\right) + q_r^n \frac{d}{dt}\left(\frac{1-e^{-\Sigma_r^n s}}{\Sigma_r^n}\right)\right. \\ & - \frac{d}{dt}\left(\frac{e^{-\Sigma_r^n s}}{v}\right)\int_0^s \frac{d\varphi_k^n(s')}{dt} e^{\Sigma_r^n s'} ds' \\ & - \frac{e^{-\Sigma_r^n s}}{v}\int_0^s \frac{d\varphi_k^n(s')}{dt}\frac{d}{dt}\left(e^{\Sigma_r^n s}\right) ds' \\ & \left. - \frac{e^{-\Sigma_r^n s}}{v}\int_0^s \frac{d^2\varphi_k^n(s')}{dt^2} e^{\Sigma_r^n s'} ds' + O(\Delta t)\right].\end{aligned} \qquad 6.44$$

The first two error terms depend upon the time-dependence of the transport cross section. In practice, the cross sections change slowly during transients except near regions where material compositions are changing rapidly (e.g. near moving control rods or voiding moderator).

To resolve the time derivatives in the first two terms, we rewrite the exponential functions as power series:

$$\begin{gathered}\varphi_k^n(0)\frac{d}{dt}\left(e^{-\Sigma_r^n s}\right)=\varphi_k^n(0)\frac{d}{dt}\left(1-\Sigma_r^n s+O(s^2)\right),\\ q_r^n\frac{d}{dt}\left(\frac{1-e^{-\Sigma_r^n s}}{\Sigma_r^n}\right)=q_r^n\frac{d}{dt}\left(\frac{1-1+\Sigma_r^n s-\frac{(\Sigma_r^n s)^2}{2}+O(s^3)}{\Sigma_r^n}\right),\end{gathered} \tag{6.45}$$

which allows us to identify the spatial dependence of the first two error terms:

$$\begin{aligned}\frac{d\varphi_k^n(s)}{dt}=\frac{d\varphi_k^n(0)}{dt}\,e^{-\Sigma_r^n s}&+\frac{q_r^n-q_r^{n-1}}{\Delta t}\left(\frac{1-e^{-\Sigma_r^n s}}{\Sigma_r^n}\right)\\ &+\Bigg[\varphi_k^n(0)\frac{d\Sigma_r^n}{dt}s-q_r^n\frac{s^2}{2}\frac{d\Sigma_r^n}{dt}\\ &-\frac{d}{dt}\left(\frac{e^{-\Sigma_r^n s}}{v}\right)\int_0^s\frac{d\varphi_k^n(s')}{dt}e^{\Sigma_r^n s'}ds'\\ &-\frac{e^{-\Sigma_r^n s}}{v}\int_0^s\frac{d\varphi_k^n(s')}{dt}\frac{d}{dt}\left(e^{\Sigma_r^n s'}\right)ds'\\ &-\frac{e^{-\Sigma_r^n s}}{v}\int_0^s\frac{d^2\varphi_k^n(s')}{dt^2}e^{\Sigma_r^n s'}ds'+O(s^2)+O(\Delta t)\Bigg].\end{aligned} \tag{6.46}$$

Thus we see that the leading parts of the first two error terms are proportional to the time derivative of the cross section and scale with the length along the segment. When the cross section is changing slowly or the regions are very small, these error terms diminish.

The remaining three error terms are more difficult to resolve because they require evaluating the integral of angular flux time derivatives. We will evaluate each sequentially. We rewrite the exponential function in the time derivative using a power series:

$$\frac{d}{dt}\left(\frac{e^{-\Sigma_r^n s}}{v}\right)=\frac{d}{dt}\left(\frac{1-\Sigma_r^n s+O(s^2)}{v}\right)=-\frac{s}{v}\frac{d\Sigma_r^n}{dt}+O(s^2). \tag{6.47}$$

To resolve the integral, we approximate the angular flux time derivative using the segment-wise average angular flux and a power series for the exponential function:

$$\begin{aligned}\int_0^s \frac{d\varphi_k^n(s')}{dt}e^{\Sigma_r^n s'}ds' &= \int_0^s \left(\frac{d\bar{\varphi}_k^n}{dt}+O(s')\right)\left(1+\Sigma_r^n s'+O\left(s'^2\right)\right)ds' \\ &= \frac{d\bar{\varphi}_k^n}{dt}s+O(s^2).\end{aligned} \tag{6.48}$$

We follow a similar approach for the second term:

$$\begin{aligned}&\frac{e^{-\Sigma_r^n s}}{v}\int_0^s \frac{d\varphi_k^n(s')}{dt}\frac{d}{dt}\left(e^{\Sigma_r^n s}\right)ds' \\ &\qquad=\left(\frac{1-\Sigma_r^n s+O(s^2)}{v}\right) \\ &\qquad\times\int_0^s\left(\frac{d\bar{\varphi}_k^n}{dt}+O(s')\right)\frac{d}{dt}\left(1+\Sigma_r^n s'+O\left(s'^2\right)\right)ds' \\ &\qquad=\left(\frac{1-\Sigma_r^n s+O(s^2)}{v}\right)\left(\frac{d\bar{\varphi}_k^n}{dt}\frac{d\Sigma_r^n}{dt}\frac{s^2}{2}+O(s^3)\right) \\ &\qquad=\frac{s^2}{2v}\frac{d\bar{\varphi}_k^n}{dt}\frac{d\Sigma_r^n}{dt}+O(s^3).\end{aligned} \tag{6.49}$$

We do the same for the third term:

$$\begin{aligned}&\frac{e^{-\Sigma_r^n s}}{v}\int_0^s \frac{d^2\varphi_k^n(s')}{dt^2}e^{\Sigma_r^n s'}ds' \\ &\qquad=\left(\frac{1+O(s)}{v}\right)\int_0^s\left(\frac{d^2\bar{\varphi}_k^n}{dt^2}+O(s')\right)\left(1+O(s')\right)ds' \\ &\qquad=\frac{s}{v}\frac{d^2\bar{\varphi}_k^n}{dt^2}+O(s^2).\end{aligned} \tag{6.50}$$

We apply Equations 6.48—6.50 to Equation 6.46 to derive the final error equation for TSDP:

$$
\begin{aligned}
\frac{d\varphi_k^n(s)}{dt} = \frac{d\varphi_k^n(0)}{dt}\, e^{-\Sigma_r^n s} + \frac{q_r^n - q_r^{n-1}}{\Delta t}\left(\frac{1-e^{-\Sigma_r^n s}}{\Sigma_r^n}\right) \\
+ \Bigg[s\left(\varphi_k^n(0)\frac{d\Sigma_r^n}{dt} - \frac{1}{v}\frac{d^2\bar{\varphi}_k^n}{dt^2}\right) - \frac{s^2}{2}\frac{d\Sigma_r^n}{dt}\left(q_r^n - \frac{1}{v}\frac{d\bar{\varphi}_k^n}{dt}\right) \\
+ O(s^2) + O(\Delta t)\Bigg].
\end{aligned}
\qquad 6.51
$$

TSDP approximations resulted in three error terms which are proportional to the cross section time derivative and one error term which is proportional to the second derivative of the angular flux. All four error terms diminish as the segment length is reduced, and the leading spatial error term is first-order in space. If the cross sections are changing slowly, the remaining error term is proportional to the second derivative of the angular flux. The error in the second derivative term is diminished by division by the neutron velocity, which is large even for thermal neutrons. This is an essential feature for this method to be reasonably accurate.

Based on this analysis, we expect TSDP to do well when the cross sections are changing slowly and when the angular flux is not accelerating or decelerating quickly. One implication of this analysis is that there could be conditions where IBDC is more accurate than TSDP; specifically, when the angular distribution of the flux is not changing quickly, but the total amplitude is accelerating or decelerating quickly. However, we have not observed this for any test transient, including those presented in Chapter 8. Nonetheless, recognition of this possibility informs the approximations used in ISDP and ISCDP, which are analyzed in the following sections.

The error in the angular flux that results from TSDP is determined by substituting Equation 6.51 into Equation 5.7:

$$\begin{aligned}\varphi_g^{k,n}(s) = \left[\varphi_g^{k,n}(s)\right]_{\mathrm{TSDP1}} \\ &- \frac{e^{-\Sigma_r^n s}}{v}\int_0^s \left[s\left(\varphi_k^n(0)\frac{d\Sigma_r^n}{dt} - \frac{1}{v}\frac{d^2\bar{\varphi}_k^n}{dt^2}\right)\right. \\ &- \frac{s^2}{2}\frac{d\Sigma_r^n}{dt}\left(q_r^n - \frac{1}{v}\frac{d\bar{\varphi}_k^n}{dt}\right) + \frac{\Delta t}{2\Sigma_r^n}\frac{d^2 q_r^n}{dt^2}\left(1 - e^{-\Sigma_r^1 s}\right) + O(s^2) \\ &\left. + O(\Delta t^2)\right] e^{\Sigma_r^n s'} ds',\end{aligned} \qquad 6.52$$

where we have expanded the $O(\Delta t)$ term in terms of the second derivative of the source. We evaluate the integral making use of the power series expansion of the exponentials:

$$\begin{aligned}\varphi_g^{k,n}(s) = \left[\varphi_g^{k,n}(s)\right]_{\mathrm{TSDP1}} \\ &- \frac{1}{v}\left[\frac{s^2}{2}\left(\varphi_k^n(0)\frac{d\Sigma_r^n}{dt} - \frac{1}{v}\frac{d^2\bar{\varphi}_k^n}{dt^2} + \frac{\Delta t}{2}\frac{d^2 q_r^n}{dt^2}\right)\right. \\ &\left. - \frac{s^3}{6}\frac{d\Sigma_r^n}{dt}\left(q_r^n - \frac{1}{v}\frac{d\bar{\varphi}_k^n}{dt}\right)\right] + O(s^3) + O(s^2\Delta t^2).\end{aligned} \qquad 6.53$$

Like RBDC and IBDC, the leading spatial error term is proportional to the leading temporal error term. However, in the case of TSDP the leading temporal error term resulting from the time derivative is quadratic in space[14]. Finally, we note that the second derivative terms are divided by the velocity twice, which substantially reduces the impact of this approximation.

6.6 Error Analysis of ISDP

The derivation for ISDP is similar to TSDP, but it approximates the second angular flux derivative term using the neutron scalar flux:

$$\frac{d\varphi_k^n(s)}{dt} \approx \frac{d\varphi_k^n(0)}{dt}e^{-\Sigma_r^n s} + \left(\frac{1}{\Sigma_r^n}\frac{q_r^n - q_r^{n-1}}{\Delta t} - \frac{\phi_r^n - 2\phi_r^{n-1} + \phi_r^{n-2}}{4\pi\Sigma_r^n v\Delta t^2}\right)\left(1 - e^{-\Sigma_r^n s}\right). \qquad 6.54$$

[14] Note that the use of the flat source approximation introduces a linear spatial error for all of the MOC methods in this work. In the parametric evaluation in Chapter 8 we observed that the spatial error for RBDC, IBDC, and TSDP scaled the same way, which indicates that this linear spatial error is dominant.

To simplify this analysis, we will build on Equation 6.51. We add the scalar flux time derivative term within the brackets and subtract the same term outside of the brackets:

$$\begin{aligned}\frac{d\varphi_k^n(s)}{dt} = \frac{d\varphi_k^n(0)}{dt}\, e^{-\Sigma_r^n s} &+ \left(\frac{q_r^n - q_r^{n-1}}{\Delta t} - \frac{\phi_r^n - 2\phi_r^{n-1} + \phi_r^{n-2}}{4\pi v \Delta t^2}\right)\left(\frac{1 - e^{-\Sigma_r^n s}}{\Sigma_r^n}\right) \\ &+ \left[\varphi_k^n(0)\frac{d\Sigma_r^n}{dt} s - q_r^n \frac{s^2}{2}\frac{d\Sigma_r^n}{dt} + \frac{s^2}{2v}\frac{d\Sigma_r^n}{dt}\frac{d\bar{\varphi}_k^n}{dt} - \frac{s}{v}\frac{d^2\bar{\varphi}_k^n}{dt^2}\right. \\ &\left. + \frac{\phi_r^n - 2\phi_r^{n-1} + \phi_r^{n-2}}{4\pi v \Delta t^2}\left(\frac{1 - e^{-\Sigma_r^n s}}{\Sigma_r^n}\right) + O(s^2) + O(\Delta t)\right].\end{aligned} \tag{6.55}$$

As in the previous section, we approximate the exponential functions in the brackets using a power series:

$$\begin{aligned}\frac{d\varphi_k^n(s)}{dt} = \frac{d\varphi_k^n(0)}{dt}\, e^{-\Sigma_r^n s} &+ \left(\frac{q_r^n - q_r^{n-1}}{\Delta t} - \frac{\phi_r^n - 2\phi_r^{n-1} + \phi_r^{n-2}}{4\pi v \Delta t^2}\right)\left(\frac{1 - e^{-\Sigma_r^n s}}{\Sigma_r^n}\right) \\ &+ \left[\varphi_k^n(0)\frac{d\Sigma_r^n}{dt} s - q_r^n \frac{s^2}{2}\frac{d\Sigma_r^n}{dt} + \frac{s^2}{2v}\frac{d\Sigma_r^n}{dt}\frac{d\bar{\varphi}_k^n}{dt} - \frac{s}{v}\frac{d^2\bar{\varphi}_k^n}{dt^2}\right. \\ &\left. + s\frac{\phi_r^n - 2\phi_r^{n-1} + \phi_r^{n-2}}{4\pi v \Delta t^2} + O(s^2) + O(\Delta t)\right].\end{aligned} \tag{6.56}$$

Next we approximate the second derivative of the angular flux with backward differences:

$$\frac{d^2\bar{\varphi}_k^n}{dt^2} = \frac{\bar{\varphi}_k^n - 2\bar{\varphi}_k^{n-1} + \bar{\varphi}_k^{n-2}}{\Delta t^2} + O(\Delta t). \tag{6.57}$$

We approximate the segment-wise angular fluxes using the region-wise angular flux as we did for IBDC:

$$\frac{d^2\bar{\varphi}_k^n}{dt^2} = \frac{\bar{\varphi}_{r,l,m}^n - 2\bar{\varphi}_{r,l,m}^{n-1} + \bar{\varphi}_{r,l,m}^{n-2}}{\Delta t^2} + O\left(W_{k,r}\right) + O(\Delta t). \tag{6.58}$$

Finally we approximate the region-wise angular fluxes using the region-wise scalar fluxes with Equation 6.38. If we apply Equations 6.57, 6.58, and 6.38 to Equation 6.56, the result is the final error equation for ISDP:

$$\begin{aligned}\frac{d\varphi_k^n(s)}{dt} = \frac{d\varphi_k^n(0)}{dt}\, e^{-\Sigma_r^n s} &+ \left(\frac{q_r^n - q_r^{n-1}}{\Delta t} - \frac{\phi_r^n - 2\phi_r^{n-1} + \phi_r^{n-2}}{4\pi v \Delta t^2}\right)\left(\frac{1 - e^{-\Sigma_r^n s}}{\Sigma_r^n}\right) \\ &+ \varphi_k^n(0)\frac{d\Sigma_r^n}{dt}s - q_r^n \frac{s^2}{2}\frac{d\Sigma_r^n}{dt} + \frac{s^2}{2v}\frac{d\Sigma_r^n}{dt}\frac{d\bar{\varphi}_k^n}{dt} - \frac{s}{v}\frac{3\mathbf{\Omega}}{4\pi}\cdot\frac{d^2\boldsymbol{J}_r^n}{dt^2} \\ &+ O\left(Y_2^j\left(\mathbf{\Omega}_{l,m}\right)\right) + O\left(W_{k,r}\right) + O(s^2) + O(\Delta t).\end{aligned} \qquad 6.59$$

As a result, the second angular flux time derivative error term in TSDP has been replaced by the second derivative of the neutron current, while the cross section derivative error terms are unchanged. This relaxes the condition for the accuracy of the method; instead of requiring that the angular flux is not accelerating quickly, it requires that the angular distribution of the flux is not accelerating quickly. On this basis we expect ISDP to be more accurate than TSDP, and for ISDP to be more accurate than IBDC except potentially when cross sections are changing quickly. This is confirmed in Chapter 8, where ISDP is uniformly more accurate than IBDC.

The error in the angular flux that results from ISDP is determined by substituting Equation 6.59 into Equation 5.7:

$$\begin{aligned}\varphi_g^{k,1}(s) = \left[\varphi_g^{k,1}(s)\right]_{\mathrm{ISDP1}} & \\ -\frac{1}{v}\Bigg[&\frac{s^2}{2}\left(\varphi_k^n(0)\frac{d\Sigma_r^n}{dt} - \frac{1}{v}\frac{3\mathbf{\Omega}}{4\pi}\cdot\frac{d^2\boldsymbol{J}_r^n}{dt^2} + \frac{\Delta t}{2}\left[\frac{d^2 q_r^n}{dt^2} + \frac{1}{v}\frac{d^3\varphi_k^n}{dt^3}\right]\right) \\ &- \frac{s^3}{6}\frac{d\Sigma_r^n}{dt}\left(q_r^n - \frac{1}{v}\frac{d\bar{\varphi}_k^n}{dt}\right)\Bigg] + O\left(Y_2^j\left(\mathbf{\Omega}_{l,m}\right)\right) + O\left(W_{k,r}\right) \\ &+ O(s^3) + O(s\Delta t^2).\end{aligned} \qquad 6.60$$

Like TSDP, the leading spatial error term is quadratic, and the leading temporal error term is quadratic in space. However, the angular flux second derivative error term has been replaced by a neutron current second derivative and an angular flux third derivative which is the result of the time derivative approximation.

6.7 Error Analysis of ISCDP

The error analysis for ISCDP proceeds in a similar fashion to TSDP. The approximation for the angular flux time derivative for ISCDP is:

$$\begin{aligned}\frac{d\varphi_k^n(s)}{dt} \approx {} & \left[\frac{d\varphi_k^n(0)}{dt} - s\frac{\Sigma_r^n - \Sigma_r^{n-1}}{\Delta t}\left(\varphi_k^n(0) - \frac{q_r^n}{\Sigma_r^n} + \frac{\phi_r^n - \phi_r^{n-1}}{4\pi\Sigma_r^n v\Delta t}\right)\right] e^{-\Sigma_r^n s} \\ & + \left(\frac{1}{\Sigma_r^n}\frac{dq_r^n}{dt} - \frac{\phi_r^n - 2\phi_r^{n-1} + \phi_r^{n-2}}{4\pi\Sigma_r^n v\Delta t^2} + \frac{\Sigma_r^n - \Sigma_r^{n-1}}{\Delta t}\frac{\phi_r^n - \phi_r^{n-1}}{4\pi\Sigma_r^n v\Delta t}\right) \\ & \times \left(1 - e^{-\Sigma_r^n s}\right).\end{aligned} \tag{6.61}$$

We derive this equation by taking the time derivative of the partially integrated characteristic equation for the angular flux:

$$\begin{aligned}\frac{d\varphi_k^n(s)}{dt} = {} & \frac{d\varphi_k^n(0)}{dt}\, e^{-\Sigma_r^n s} + \frac{dq_r^n}{dt}\left(\frac{1 - e^{-\Sigma_r^n s}}{\Sigma_r^n}\right) + \varphi_k^n(0)\frac{d}{dt}\left(e^{-\Sigma_r^n s}\right) \\ & - q_r^n \frac{d}{dt}\left(\frac{1 - e^{-\Sigma_r^n s}}{\Sigma_r^n}\right) - \frac{d}{dt}\left(\frac{e^{-\Sigma_r^n s}}{v}\right)\int_0^s \frac{d\varphi_k^n(s')}{dt} e^{\Sigma_r^n s'} ds' \\ & - \frac{e^{-\Sigma_r^n s}}{v}\int_0^s \frac{d\varphi_k^n(s')}{dt}\frac{d}{dt}\left(e^{\Sigma_r^n s}\right) ds' - \frac{e^{-\Sigma_r^n s}}{v}\int_0^s \frac{d^2\varphi_k^n(s')}{dt^2} e^{\Sigma_r^n s'} ds'.\end{aligned} \tag{6.62}$$

To evaluate the cross section derivative terms, we rewrite the cross section using a Taylor series expansion:

$$\Sigma_r(t) = \Sigma_r^n + (t - t_n)\frac{d\Sigma_r^n}{dt} + \frac{(t - t_n)^2}{2}\frac{d^2\Sigma_r^n}{dt^2} + O(t^3). \tag{6.63}$$

This allows us to evaluate the cross section time derivatives at time t_n:

$$\begin{aligned}\frac{d\varphi_k^n(s)}{dt} = {} & \left[\frac{d\varphi_k^n(0)}{dt} - s\frac{d\Sigma_r^n}{dt}\left(\varphi_k^n(0) - \frac{q_r^n}{\Sigma_r^n}\right)\right] e^{-\Sigma_r^n s} \\ & + \left(\frac{dq_r^n}{dt} - \frac{q_r^n}{\Sigma_r^n}\frac{d\Sigma_r^n}{dt}\right)\left(\frac{1 - e^{-\Sigma_r^n s}}{\Sigma_r^n}\right) + \frac{d\Sigma_r^n}{dt}\frac{s e^{-\Sigma_r^n s}}{v}\int_0^s \frac{d\varphi_k^n(s')}{dt} e^{\Sigma_r^n s'} ds' \\ & - \frac{d\Sigma_r^n}{dt}\frac{e^{-\Sigma_r^n s}}{v}\int_0^s \frac{d\varphi_k^n(s')}{dt} s e^{\Sigma_r^n s} ds' - \frac{e^{-\Sigma_r^n s}}{v}\int_0^s \frac{d^2\varphi_k^n(s')}{dt^2} e^{\Sigma_r^n s'} ds'.\end{aligned} \tag{6.64}$$

Next we approximate the angular flux time derivatives using the segment-wise average angular fluxes as for RBDC, which introduce a linear spatial error term:

$$\frac{d\varphi_k^n(s)}{dt} = \frac{d\bar{\varphi}_k^n}{dt} + O(s). \tag{6.65}$$

This allows us to resolve the integrals and results in a quadratic spatial error:

$$\begin{aligned}\frac{d\varphi_k^n(s)}{dt} &= \Bigg[\frac{d\varphi_k^n(0)}{dt} - s\,\frac{d\Sigma_r^n}{dt}\bigg(\varphi_k^n(0) - \frac{q_r^n}{\Sigma_r^n} + \frac{1}{\Sigma_r^n v}\frac{d\bar{\varphi}_k^n}{dt}\bigg)\Bigg]\, e^{-\Sigma_r^n s} \\ &\quad + \Bigg[\frac{dq_r^n}{dt} - \frac{d\Sigma_r^n}{dt}\bigg(\frac{q_r^n}{\Sigma_r^n} - \frac{1}{\Sigma_r^n v}\frac{d\bar{\varphi}_k^n}{dt}\bigg) - \frac{1}{v}\frac{d^2\bar{\varphi}_k^n}{dt^2}\Bigg]\bigg(\frac{1-e^{-\Sigma_r^n s}}{\Sigma_r^n}\bigg) + O(s^2).\end{aligned} \tag{6.66}$$

Next we approximate the segment-wise angular flux time derivatives in terms of the region-wise scalar flux, introducing error terms based on the neutron current derivatives:

$$\begin{aligned}\frac{d\varphi_k^n(s)}{dt} &= \Bigg[\frac{d\varphi_k^n(0)}{dt} - s\,\frac{d\Sigma_r^n}{dt}\bigg(\varphi_k^n(0) - \frac{q_r^n}{\Sigma_r^n} + \frac{1}{4\pi\Sigma_r^n v}\frac{d\phi_r^n}{dt}\bigg)\Bigg]\, e^{-\Sigma_r^n s} \\ &\quad + \Bigg[\frac{dq_r^n}{dt} - \frac{d\Sigma_r^n}{dt}\bigg(\frac{q_r^n}{\Sigma_r^n} - \frac{1}{4\pi\Sigma_r^n v}\frac{d\phi_r^n}{dt}\bigg) - \frac{1}{4\pi v}\frac{d^2\phi_r^n}{dt^2}\Bigg] \times \bigg(\frac{1-e^{-\Sigma_r^n s}}{\Sigma_r^n}\bigg) \\ &\quad - \Bigg[s\,\frac{d\Sigma_r^n}{dt}\frac{3\mathbf{\Omega}_k}{4\pi\Sigma_r^n v}\cdot\frac{d\boldsymbol{J}_r^n}{dt}\; e^{-\Sigma_r^n s} \\ &\quad - \bigg(\frac{d\Sigma_r^n}{dt}\frac{3\mathbf{\Omega}_k}{4\pi\Sigma_r^n v}\cdot\frac{d\boldsymbol{J}_r^n}{dt} - \frac{1}{v}\frac{3\mathbf{\Omega}_k}{4\pi v}\cdot\frac{d^2\boldsymbol{J}_r^n}{dt^2}\bigg)\bigg(\frac{1-e^{-\Sigma_r^n s}}{\Sigma_r^n}\bigg) \\ &\quad + O\Big(Y_2^j\big(\mathbf{\Omega}_{l,m}\big)\Big) + O\big(W_{k,r}\big) + O(s^2)\Bigg].\end{aligned} \tag{6.67}$$

We approximate the time derivatives using backward differences:

$$\begin{aligned}\frac{d\varphi_k^n(s)}{dt} &= \Bigg[\frac{d\varphi_k^n(0)}{dt} - s\,\frac{\Sigma_r^n - \Sigma_r^{n-1}}{\Delta t}\bigg(\varphi_k^n(0) - \frac{q_r^n}{\Sigma_r^n} + \frac{\phi_r^n - \phi_r^{n-1}}{4\pi\Sigma_r^n v\Delta t}\bigg)\Bigg]\, e^{-\Sigma_r^n s} \\ &\quad + \Bigg[\frac{q_r^n - q_r^{n-1}}{\Delta t} - \frac{\Sigma_r^n - \Sigma_r^{n-1}}{\Delta t}\bigg(\frac{q_r^n}{\Sigma_r^n} - \frac{\phi_r^n - \phi_r^{n-1}}{4\pi\Sigma_r^n v\Delta t}\bigg) \\ &\quad - \frac{\phi_r^n - 2\phi_r^{n-1} + \phi_r^{n-2}}{4\pi v\Delta t^2}\Bigg]\bigg(\frac{1-e^{-\Sigma_r^n s}}{\Sigma_r^n}\bigg) \\ &\quad - \Bigg[s\,\frac{d\Sigma_r^n}{dt}\frac{3\mathbf{\Omega}_k}{4\pi\Sigma_r^n v}\cdot\frac{d\boldsymbol{J}_r^n}{dt}\; e^{-\Sigma_r^n s} \\ &\quad - \bigg(\frac{d\Sigma_r^n}{dt}\frac{3\mathbf{\Omega}_k}{4\pi\Sigma_r^n v}\cdot\frac{d\boldsymbol{J}_r^n}{dt} - \frac{3\mathbf{\Omega}_k}{4\pi v}\cdot\frac{d^2\boldsymbol{J}_r^n}{dt^2}\bigg)\bigg(\frac{1-e^{-\Sigma_r^n s}}{\Sigma_r^n}\bigg) + O\Big(Y_2^j\big(\mathbf{\Omega}_{l,m}\big)\Big) \\ &\quad + O\big(W_{k,r}\big) + O(s^2) + O(\Delta t)\Bigg].\end{aligned} \tag{6.68}$$

Finally, we expand the exponentials in the bracketed error term to produce a final ISCDP error equation that is analogous to the equation for ISDP:

$$\begin{aligned}
\frac{d\varphi_k^n(s)}{dt} = {} & \left[\frac{d\varphi_k^n(0)}{dt} - s\frac{\Sigma_r^n - \Sigma_r^{n-1}}{\Delta t}\left(\varphi_k^n(0) - \frac{q_r^n}{\Sigma_r^n} + \frac{\phi_r^n - \phi_r^{n-1}}{4\pi\Sigma_r^n v\Delta t}\right)\right] e^{-\Sigma_r^n s} \\
& + \left[\frac{q_r^n - q_r^{n-1}}{\Delta t} - \frac{\Sigma_r^n - \Sigma_r^{n-1}}{\Delta t}\left(\frac{q_r^n}{\Sigma_r^n} - \frac{\phi_r^n - \phi_r^{n-1}}{4\pi\Sigma_r^n v\Delta t}\right)\right. \\
& \left. - \frac{\phi_r^n - 2\phi_r^{n-1} + \phi_r^{n-2}}{4\pi v\Delta t^2}\right]\left(\frac{1 - e^{-\Sigma_r^n s}}{\Sigma_r^n}\right) \\
& - \left[\frac{3\mathbf{\Omega}_k}{4\pi v}\cdot\left(s\frac{d^2\boldsymbol{J}_r^n}{dt^2} + \frac{s^2}{2}\frac{d\Sigma_r^n}{dt}\frac{d\boldsymbol{J}_r^n}{dt}\right) + O\left(Y_2^j(\mathbf{\Omega}_{l,m})\right) + O\left(W_{k,r}\right)\right. \\
& \left. + O(s^2) + O(\Delta t)\right].
\end{aligned} \tag{6.69}$$

As for ISDP, there is a linear error term for ISCDP that is proportional to the second derivative of the neutron current; this is the result of approximating the second derivative of the angular flux with the scalar flux. However, in contrast to TSDP and ISDP, the two cross section derivative error terms proportional to the incoming angular flux and the neutron source have been eliminated. In addition, the cross section derivative error term in TSDP and ISDP that is proportional to the angular flux time derivative is now proportional to the neutron current time derivative. As a result, we expect ISCDP to be more accurate than TSDP and ISDP. Further, unlike TSDP and ISDP, ISCDP is expected to still be accurate when the cross sections are changing quickly. However, in Chapter 8 we observe that the difference between ISDP and ISCDP is negligible, indicating that the assumption of slowly-changing cross sections is reasonable for the transients we tested.

The error in the angular flux that results from ISCDP is determined by substituting Equation 6.69 into Equation 5.7:

$$\begin{aligned}\varphi_g^{k,1}(s) = {} & \left[\varphi_g^{k,1}(s)\right]_{\mathrm{ISCDP1}} \\ & -\frac{1}{v}\left[\frac{s^2}{2}\left(\frac{\Delta t}{2}\left[\frac{d^2\Sigma_r^n}{dt^2}\left(\varphi_k^n(0) - 2\frac{q_r^n}{\Sigma_r^n} + \frac{1}{2\pi\Sigma_r^n v}\frac{d\phi_r^n}{dt}\right) + \frac{d^2 q_r^n}{dt^2}\right.\right.\right. \\ & \left.\left.\left. - \frac{1}{4\pi v}\frac{d^3\phi_r^n}{dt^3}\right] + \frac{3\mathbf{\Omega}_k}{4\pi v}\cdot\frac{d^2\boldsymbol{J}_r^n}{dt^2}\right) + \frac{s^3}{6}\frac{3\mathbf{\Omega}_k}{4\pi v}\cdot\frac{d\Sigma_r^n}{dt}\frac{d\boldsymbol{J}_r^n}{dt}\right] \\ & + O\left(Y_2^j(\mathbf{\Omega}_{l,m})\right) + O(W_{k,r}) + O(s^3) + O(s\Delta t^2) + O(\Delta t^2).\end{aligned} \qquad 6.70$$

The error equation for ISCDP is more complex than TSDP and ISDP, but now every error term is proportional to a second temporal derivative or higher. Like TSDP and ISDP, the leading spatial error term is quadratic, and the leading temporal error term is also quadratic in space.

6.8 Summary and Conclusions

In this chapter we identified the leading error terms for the angular flux time derivative approximations for each time-dependent MOC method. All of the methods had similar temporal error which was determined by the order of the approximation for the time derivatives. One noteworthy difference between the BDC and SDP methods was that the spatial error was linear for BDC and quadratic for SDP. However, based on the parametric study in Chapter 8, it appears that the linear spatial error that results from the flat source approximation dominates the spatial error for all methods.

The methods that approximated the segment-wise angular flux time derivatives with the region-wise scalar flux introduced a new spatially-dependent error term which reflects the fact that the scalar flux is defined on a different mesh than the angular flux. However, this error term becomes small if the regions are small. Further, the scalar flux approximation preserves the zeroth moment of the angular flux and relaxes the criteria for accuracy; if the angular dependence of the neutron flux is changing slowly, the approximation is accurate.

This observation was leveraged in the development of the ISDP method. The original SDP method, TSDP, incurs an error term which is proportional to the second derivative of the angular flux. In ISDP, this term is approximated using the second derivative of the scalar flux, which reduces the error.

The other major sources of error in TSDP and ISDP are the result of neglecting the time derivatives of the cross sections. ISCDP addresses these error terms by incorporating the cross section derivatives into the angular flux time derivative equation. As a result, ISCDP is expected to be the most accurate of the new SDP methods. We confirm these trends with the numerical experiments in Chapter 8.

Chapter 7
Overview of DeCART and Implementation of MOC Methods

7.1 Introduction

This chapter provides background on the computer code DeCART [Joo04, Cho05, Hur08] and the implementation of the time-dependent method of characteristics (MOC) methods in DeCART for this thesis. DeCART is a three-dimensional (3D) whole core neutron transport code for light water reactor analysis. This chapter will first provide an overview of DeCART, including a summary of the important features used in this work. It will then discuss in more detail the use of coarse mesh finite difference (CMFD) based diffusion to accelerate source convergence in DeCART. Finally, it will discuss the steady-state and transient MOC algorithms with more detail than provided in Chapters 3—5.

7.2 Overview of DeCART

DeCART models 3D reactor problems using a coupled 2D/1D approach. Most of the spatial heterogeneity within a light water reactor is in the planar dimension, and in these dimensions DeCART uses 2D MOC to resolve the fine spatial details of the nuclear fuel assemblies. By contrast, there is greater uniformity in the axial dimension, so DeCART uses a neutron diffusion-based nodal expansion method (NEM) for this dimension. Since this thesis exclusively considers 2D problems, NEM is not employed in this work.

The 2D/1D model in DeCART is coupled using a 3D diffusion-based coarse mesh finite difference (CMFD) method. CMFD and MOC operate synergistically in DeCART: CMFD is used to accelerate the neutron source convergence in MOC, and MOC provides the radial neutron currents for CMFD’s coarse mesh cells to generate correction factors that preserve the MOC current using neutron diffusion. CMFD is roughly an order of magnitude less computationally-expensive than MOC, and it can significantly accelerate the convergence of the neutron flux. As a result, CMFD has been commonly used to accelerate neutron transport methods [Col13].

DeCART solves the neutron transport equation by alternating between CMFD and MOC. CMFD is used to calculate coarse-mesh (cell-wise) neutron scalar flux, which provides an initial estimate of the neutron source term for MOC. MOC is used to calculate the fine-mesh (region-wise) scalar flux as well as the neutron currents for the CMFD cells. CMFD then incorporates a current-correction factor so that it preserves the MOC currents at the cell boundaries. CMFD is used to recalculate the coarse-mesh scalar flux, and the change in the coarse mesh scalar flux is used to update the neutron source term for MOC. This process repeats until the solution converges.

In this work, DeCART is used to solve two types of problems: criticality eigenvalue problems and transient problems. In criticality eigenvalue problems, the neutron fission source is modified by an eigenvalue which ensures that the fission source is balanced by the neutron loss terms. By contrast, transient problems evaluate the change in the neutron flux distribution over time beginning from a critical configuration as a result of some perturbation.

The MOC algorithm is similar for these two problems. The primary difference between MOC for the steady-state and transient problem is the incorporation of the angular flux time derivative in the latter. The time-dependent MOC methods employed in this thesis use different approaches to approximating the time derivative, but they incorporate the time derivative in a fashion that does not result in a large departure from steady-state MOC conventions.

7.3 CMFD Acceleration and MOC Coupling

To accelerate the convergence of the neutron transport-based MOC model, DeCART forms an equivalent neutron diffusion-based model using CMFD. The CMFD model uses Cartesian spatial “cells” which are coarser than irregularly shaped MOC “regions” and typically correspond to a single homogenized pin cell. Figure 7-1 shows a comparison of the MOC regions and a corresponding CMFD cell.

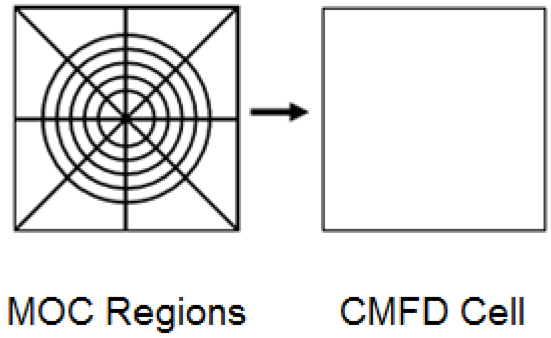


Figure 7-1. Comparison of MOC regions and CMFD cells [adapted from Hur08]

In order to make the CMFD model equivalent to the MOC model, DeCART homogenizes the MOC region-wise cross sections for the CMFD cells using scalar flux weighting:

$$\Sigma_{X,g}^{c} = \frac{\sum_{r\in c} V^{r}\Sigma_{X,g}^{r}\phi_{g}^{r}}{V^{c}\phi_{g}^{c}}, \tag{7.1}$$

where c is the index for the CMFD cells, r is the index for the MOC regions, Σ_X is the macroscopic cross section in question, and V is the volume of the cell or region.

The homogenized cross sections are used to solve a 2D or 3D spatially-discretized diffusion equation. Here we will use a 1D equation for simplicity:

$$\frac{J_g^{c+} - J_g^{c-}}{\Delta c} + \Sigma_{r,g}^{c}\phi_g^{c} = \frac{\chi_g^{c}}{k_{\text{eff}}}\sum_{g'} \nu\Sigma_{f,g'}^{c}\phi_{g'}^{c} + \sum_{g'\neq g}\Sigma_{s,g'\to g}^{c}\phi_{g'}^{c}, \tag{7.2}$$

where $J^{c\pm}$ is the neutron current at each cell face as calculated by MOC, Δc is the width of the cell, Σ_r is the removal cross section, and the other quantities are defined in the previous chapters.

To avoid storing the MOC neutron currents, corrective factors $\widehat{D}$ are defined that preserve the neutron currents in terms of the CMFD scalar fluxes:

$$\begin{aligned} J_g^{c+} &= -\widetilde{D}_g^{c+}\left(\phi_g^{c+1} - \phi_g^{c}\right) - \widehat{D}_g^{c+}\left(\phi_g^{c+1} + \phi_g^{c}\right), \\ J_g^{c-} &= +\widetilde{D}_g^{c-}\left(\phi_g^{c-1} - \phi_g^{c}\right) - \widehat{D}_g^{c-}\left(\phi_g^{c-1} + \phi_g^{c}\right), \end{aligned} \tag{7.3}$$

where $\widetilde{D}$ is the coupling coefficient determined using finite differences in terms of the standard diffusion coefficient:

$$\widetilde{D}_g^{c\pm} = \frac{2D_g^{c\pm 1}D_g^c}{\left(D_g^{c\pm 1} + D_g^c\right)\Delta c}, \tag{7.4}$$

and D is the diffusion coefficient:

$$D_g^c \equiv \frac{1}{3\Sigma_{tr,g}^c}. \tag{7.5}$$

For the first CMFD iteration before the MOC neutron currents are known, the corrective factor is assumed to be zero. This is equivalent to CMFD without a correction factor.

This permits us to rewrite Equation 7.2 with only the scalar fluxes as unknowns:

$$\begin{aligned}\frac{1}{\Delta c}\Big[-\widetilde{D}_g^{c+}\left(\phi_g^{c+1} - \phi_g^c\right) - \widehat{D}_g^{c+}\left(\phi_g^{c+1} + \phi_g^c\right) - \widetilde{D}_g^{c-}\left(\phi_g^{c-1} - \phi_g^c\right)\\ + \widehat{D}_g^{c-}\left(\phi_g^{c-1} + \phi_g^c\right)\Big] + \Sigma_{r,g}^c\phi_g^c\\ = \frac{\chi_g^c}{k_{\text{eff}}}\sum_{g'}\nu\Sigma_{f,g'}^c\phi_{g'}^c + \sum_{g'\neq g}\Sigma_{s,g'\to g}^c\phi_{g'}^c.\end{aligned} \tag{7.6}$$

In DeCART, this multi-group linear system is solved using source iteration while a Krylov subspace method is used for within group solutions. Once the cell-wise scalar flux is determined, it is used to update the region-wise scalar flux:

$$\phi_g^{r,i+1/2} = \frac{\phi_g^{c,i+1/2}}{\phi_g^{c,i}}\phi_g^{r,i}, \tag{7.7}$$

where i is the iteration index, and $i + 1/2$ corresponds to the intermediate scalar flux solution after the CMFD update but before MOC iterations.

The updated region-wise scalar fluxes are used to calculate the region-wise neutron source for MOC. An MOC iteration is then performed, which results in an updated region-wise scalar flux and cell-wise neutron currents. If the solution has not converged, the cell-wise neutron current corrective factor is recalculated, and another CMFD/MOC iteration is performed. Convergence is evaluated based on several measures including the convergence of the eigenvalue, the residual, the fission source, the fuel temperature, and the transient residual. This process is summarized in Figure 7-2.

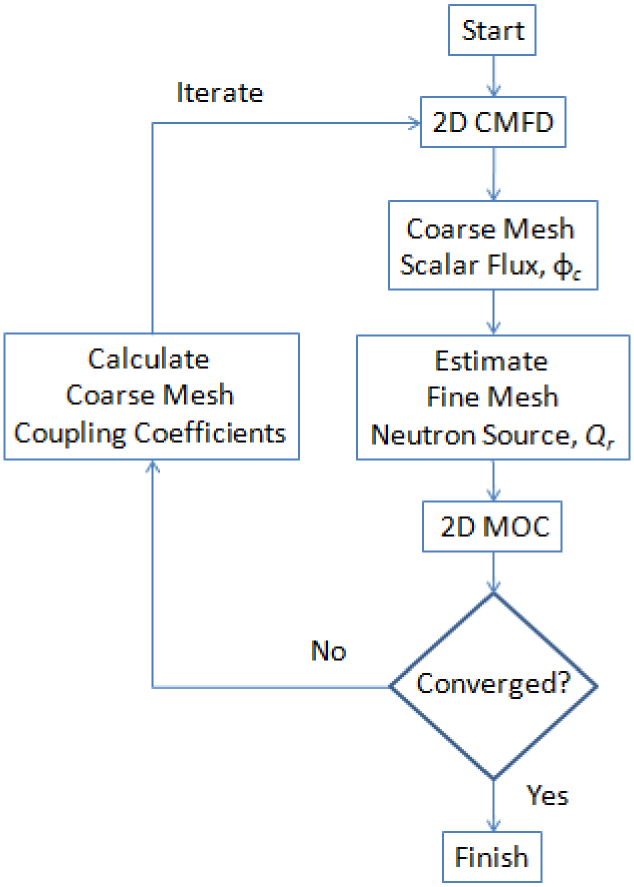


Figure 7-2. Simplified representation of DeCART CMFD/MOC algorithm

The process is similar for time-dependent problems, except that delayed neutrons are incorporated using analytic precursor integration, the time derivative is incorporated as a transient fixed source term, and the fission source is divided by the steady-state eigenvalue to ensure an initially-critical system. The time-dependent equations are used to model a single time step, and CMFD/MOC iterate until the solution at the end of the time step has converged. Then the process continues with advancing time steps until the end of the transient.

7.4 MOC Initialization and Algorithm

MOC is used to derive 1D transport equations for the angular flux along characteristics with different angles and positions in the problem domain. These 1D equations are used both to solve for the angular flux distribution along the characteristics as well as to numerically integrate the scalar flux within fine mesh regions. The MOC equations were derived in Chapters 3—5 and the time-dependent MOC equations were analyzed in Chapter 6. In this section we will discuss the setup of the MOC algorithm and application of the MOC equations in DeCART. Note that this section does not outline all of the capabilities of DeCART; rather, we describe only the capabilities used in this work.

7.4.1 Spatial Discretization and Initialization

The spatial discretization of the MOC regions in DeCART is based on a user input. DeCART defines the MOC spatial meshing within the confines of the CMFD Cartesian spatial meshing as shown in Figure 7-1. While DeCART is able to model a variety of arbitrary geometric shapes, the test problems in this work used either subdivided the CMFD cell using square regions (e.g. the TWIGL transients) or circular regions centered in the CMFD cell (e.g. the C5G7 transient). The circular regions are typically used to represent nuclear fuel pins, guide tubes, or similar structures, and they may be subdivided using radial divisions. Figure 7-3 provides an example of the C5G7 spatial meshing for a CMFD cell with 32 MOC regions.

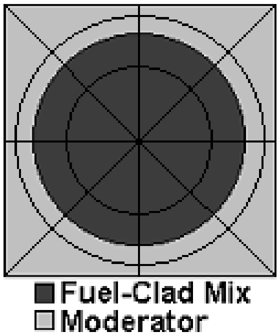


Figure 7-3. MOC spatial meshing used for C5G7 problem [adapted from NEA03]

The positions and angles of the characteristics are also based on a user input. The user specifies the number of azimuthal and polar angles modeled as well as the desired spacing between parallel characteristics. However, if the specifications for the angles and characteristic spacing were allowed to be completely arbitrary, the memory requirements for storing the geometric data representing the characteristics would be enormous; this is similar to the memory problem that arises when we need to store the angular flux to represent the time derivative for MOC. To avoid this issue, DeCART and many other MOC computer codes define the characteristics in terms of reusable characteristic modules[15]. These modules have a repeating pattern of characteristics that are reproduced in every instance of a module so that the geometric information only needs to be stored once. Figure 7-4 provides an example of a cell-based characteristic module for a single pin-cell with characteristics at one angle.

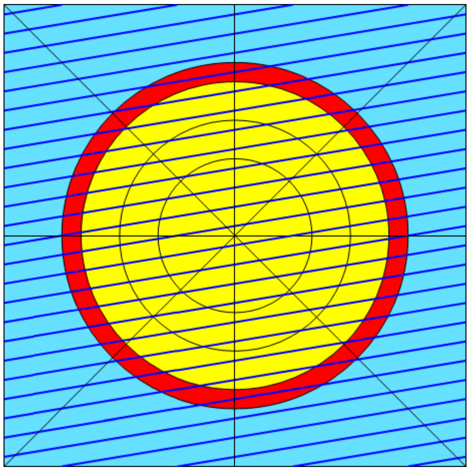


Figure 7-4. Characteristic module for a pin cell [adapted from Hur08]

The characteristic modules are defined such that the characteristics in one module are continuous with the characteristics in the adjacent modules at the same angle. This allows the angular flux to be propagated continuously across the module boundaries. Figure 7-5 shows a single characteristic that spans several characteristic

[15] These are often referred to as "ray tracing modules", where the term "ray" is synonymous with characteristic.

modules. In addition, the characteristics in the module are defined such that characteristics at supplementary angles are continuous at module boundaries to treat reflecting boundary conditions.

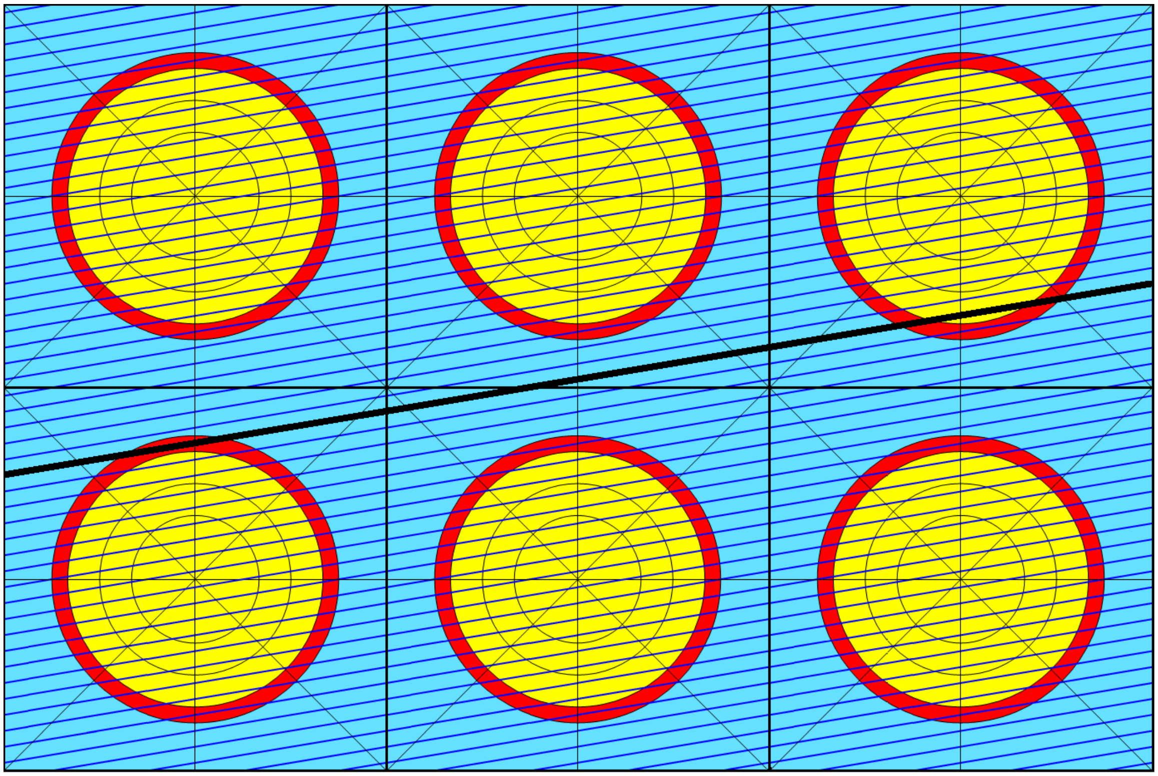


Figure 7-5. A characteristic spanning four modules [adapted from Hur08]

7.4.2 Steady-State MOC Algorithm

With the MOC geometry and characteristics defined, DeCART can begin the iterative process of CMFD and MOC to solve the neutron transport equation. As described in the previous section, CMFD is used first to solve for the scalar flux distribution on the coarse cell-wise mesh. The cell-wise scalar flux is used to update the region-wise scalar flux using Equation 7.7. This scalar flux will be used to calculate the fission and scattering neutron sources for each region for each energy group.

Next DeCART begins the inner loop of region-wise source convergence. Starting with the fast neutrons, DeCART loops over energy groups. For each energy group, the neutron fission and scattering sources are calculated using the most recent region-wise scalar flux.

Next the one group characteristic tracing algorithm begins by looping over all angles and then all characteristics at each angle. For each characteristic, DeCART advances segment-by-segment and calculates the outgoing angular flux for the segment and the contribution of the angular flux along the segment to the scalar flux in the region. Note that DeCART models “long” characteristics which span the entire geometry rather than “short” characteristics which only span a characteristic module. Figure 7-6 shows characteristics in a pin cell for angle **Ω.** The characteristics are evaluated starting with the bottom right corner, and the segments for the first three characteristics are numbered.

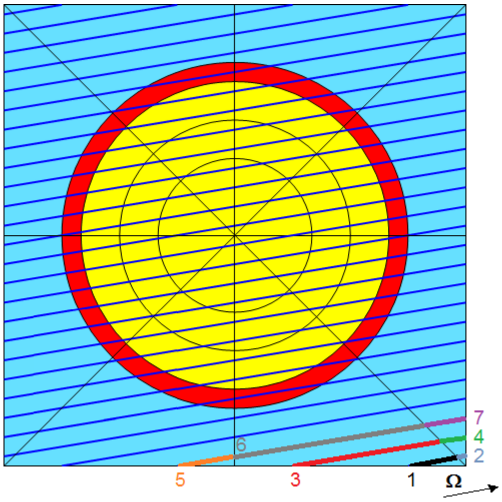


Figure 7-6. First three characteristics evaluated for a module [adapted from Hur08]

We will use Figure 7-6 as an example with the assumption that it represents the entire problem domain. Starting with Segment 1, the incoming angular flux is assumed to be known from the boundary condition. The segment-wise average angular flux is calculated using the known incoming angular flux, the neutron source in the region, and the cross section. The precise equation depends on the MOC method employed, and we will use the steady-state equation for this example:

$$\bar{\varphi}_g^{k,r} = \left(\varphi_{g,\mathrm{in}}^{k,r} - \frac{q_g^r}{\Sigma_{tr,g}^r}\right)\left(\frac{1 - e^{-\Sigma_{tr,g}^r \Delta s^{k,r}}}{\Sigma_{tr,g}^r \Delta s^{k,r}}\right) + \frac{q_g^r}{\Sigma_{tr,g}^r}, \tag{7.8}$$

where k is the index for the characteristic, $\varphi_{g,\mathrm{in}}^{k,r}$ is the incoming angular flux, q_g^r is the total neutron source, $\Sigma_{tr,g}^r$ is the macroscopic transport cross section, and $\Delta s^{k,r}$ is the length of the segment.

Using a weighting factor based on the area or volume within the region that the segment represents and a weighting factor angular quadrature, the contribution of the segment-wise average angular flux to the region-wise scalar flux is tallied:

$$\phi_g^r = \sum_m^M w_m \sum_l^L w_l \frac{\sum_{k \in r} \bar{\varphi}_g^{k,r} V^{k,r}}{\sum_{k \in r} V^{k,r}}, \tag{7.9}$$

where l and m are the indices for the azimuthal and polar components of angle $\Omega_{l,m}$.

The outgoing angular flux for Segment 1 is also calculated:

$$\varphi_{g,\mathrm{out}}^{k,r} = \varphi_{g,\mathrm{in}}^{k,r} e^{-\Sigma_{tr,g}^r \Delta s^{k,r}} + q_g^r \left(\frac{1 - e^{-\Sigma_{tr,g}^r \Delta s^{k,r}}}{\Sigma_{tr,g}^r}\right). \tag{7.10}$$

The outgoing angular flux for Segment 1 serves as the incoming angular flux for Segment 2. This process continues, advancing segment-by-segment along a characteristic until reaching a problem boundary. If this boundary has a reflecting boundary condition, the outgoing angular flux at the boundary is stored as the incoming angular flux for the supplementary angle characteristic at the outgoing position.

Although we described this process for a single direction, DeCART evaluates the angular flux along a characteristic in the forward and backward direction at the same time. This approach is more computationally efficient because some quantities are identical for each direction (e.g. the segment lengths and the exponential terms), and this results in those quantities only being looked-up or calculated once.

At this point, the incoming and outgoing angular fluxes for each segment are still stored in memory. The outgoing angular fluxes that correspond to cell boundaries for

CMFD are used to calculate the cell-wise currents. These currents will be used to generate the correction terms for CMFD.

When all of the segments on a characteristic have been evaluated, the angular fluxes along the characteristic are discarded, and the next characteristic is evaluated segment-by-segment. In this example, Segment 3 would be next. Once all of the characteristics at all of the angles have been evaluated for a particular energy group, the region-wise scalar flux for that energy group is fully integrated. The source is updated for this energy group and the process is repeated once to ensure that angular fluxes have been fully propagated across all reflecting boundaries for the first iteration.

At this point DeCART advances to the next energy group and repeats this process. This continues until all energy groups have been evaluated. For the low energy groups that experience significant neutron upscattering, the process is repeated again to capture this effect. Now an MOC iteration is complete. The convergence criteria are evaluated, and another CMFD/MOC iteration is performed if necessary.

7.4.3 Differences for Transient MOC

The MOC algorithm for the time-dependent problem in DeCART is similar to the steady-state problem. The major differences are the incorporation of the angular flux time derivative, the treatment of delayed neutrons, and the division of the fission source by the eigenvalue to ensure that the initial system is critical. Transients are initiated by changing material compositions, cross sections, or other properties (e.g. density), which results in an imbalance in the neutron production and loss terms.

Transients are modeled by solving the time-dependent transport equation for a series of user-defined time steps over a user-defined duration. Solving the transport equation for each time step is roughly analogous to converging the steady-state transport equation, except that the fluxes change relatively little over a time step compared to the typical change in the flux from the initial guess to the converged steady-state solution. As a result, whereas a steady-state eigenvalue problem may require many MOC iterations to converge, during a transient most time steps converge with one or two MOC iterations. As a result, there is limited opportunity for transient-specific transport acceleration schemes in DeCART.

The two types of time-dependent MOC methods investigated in this thesis treat the angular flux time derivative differently. The more conventional approach (i.e. the BDC methods) is to approximate the angular flux time derivative using a local finite difference approximation. This results in a new source term for each segment or region depending on the mesh where the approximate time derivative is defined. When the segment-wise angular flux time derivative is approximated using the region-wise scalar flux (i.e. IBDC), the time derivative is fully incorporated into the region-wise neutron source. When the segment-wise angular flux time derivative is approximated using the segment-wise average angular fluxes (i.e. RDBC), a unique MOC algorithm is required to construct the segment-wise time derivative approximations. However, in either case, the time derivative is incorporated as a transient source term, and the MOC equations are used to calculate the region-wise scalar flux and to propagate the angular flux along characteristics in a similar fashion to steady-state MOC.

The new time-dependent MOC methods developed in this thesis (i.e. the SDP methods) treat the segment-wise angular flux time derivative differently. For these methods, characteristic equations are defined for the angular flux time derivative along the characteristic. These equations are used to propagate the angular flux time derivative along with the propagated angular flux. As a result, an additional equation is evaluated for each segment. Otherwise the transient MOC algorithm is similar to the steady-state algorithm.

7.5 Summary

This chapter provided an overview of the computer code DeCART which was used to test the new time-dependent MOC methods developed for this thesis. The chapter began by providing a summary of DeCART and its major features. The chapter then described the use of CMFD to solve the neutron diffusion equations to accelerate the convergence of MOC. Finally, the chapter discussed the MOC algorithms.

Chapter 8

Test Problems and Numerical Results

8.1 Introduction

In this chapter we present three reactor transient test problems that were used to empirically evaluate the accuracy and performance of the new Source Derivative Propagation (SDP) methods described in Chapter 5 and analyzed in Chapter 6. This chapter focuses on the SDP method with the Truncated second-derivative (TSDP) primarily, but results are provided for selected cases of SDP with the Isotropic correction (ISDP) and with cross section propagation (ISCDP). The SDP methods were implemented in the neutron transport computer code DeCART as described in Chapter 7.

8.1.1 Approach to Evaluate the Accuracy of SDP

We assess the accuracy of the SDP methods by comparison to solutions generated using a reference method[16]. The reference method—described in Chapter 4—uses method of characteristics (MOC) to treat the spatial derivatives and the backward differentiation formula (BDF) to treat the time derivative. The reference MOC method uses the designation RBDC*N*, where *N* corresponds to the order of the BDF solution employed. The orders of SDP methods are indicated similarly (e.g. TSDP*N*).

The reference method is employed in two different ways. First, we use the first-order reference method (RBDC1) with a fine time step to generate a primary reference solution. This solution is used to verify that the SDP methods converge to the reference solution as the time step is decreased and that the high-order SDP methods increase in accuracy as the order increases. Second, we will use the RBDC with various orders and time step sizes to determine whether the SDP methods reproduce the RBDC solution for

[16] We are not using published reference solutions for these transients because either the problem is new and does not have published solutions, or the published solutions were calculated using neutron diffusion theory rather than neutron transport.

the same order and time step size. In other words, we will compare the solution of RBDC*N* with time step Δt to SDP*N* with the same time step; if these results are in good agreement, it demonstrates that the SDP methods accurately reproduce the solution using the conventional method.

While the RBDC methods provide a reference solution for assessing accuracy, we will also use a time-dependent MOC method with an isotropic approximation for the angular flux time derivative (IBDC) to assess the efficiency of the SDP methods. IBDC is memory-efficient because it does not require storing any angularly-dependent data, and it is computationally-efficient because it does not require any additional arithmetic operations at the segment-level as compared to steady-state MOC. This is in contrast to RBDC which requires the formation of a unique angular flux time derivative approximation for every segment.

8.1.2 Numerical Test Problems

In this chapter, we present results from three reactor kinetics test problems. These three transients are modeled in two dimensions (2D) because DeCART only performs neutron transport in 2D. The models are also necessarily small because of the high memory expense of storing the neutron angular flux for RBDC.

The first two transients are based on the TWIGL reactor model [Yas65]. TWIGL is a 2D reactor with large homogenous regions with specified cross sections. Transients are driven by changing the cross sections in some regions. Two transient problems are specified: a slow transient based on a linear cross section ramp, and a faster transient using a step-change in the cross section. The TWIGL transients are still widely used for benchmarking early in the development of reactor kinetics methods because of their simple modeling requirements. The TWIGL transient is also much less computationally expensive than the other transients allowing us to generate many solutions for different time step sizes. This includes solutions using RBDC6, which required storing the angular flux from six previous points in time.

The third transient is based on the C5G7 mixed-oxide (MOX) benchmark problem [Lew01]. The C5G7 problem is a small pressurized water reactor (PWR) core with conventional UO2 and MOX assemblies. The C5G7 problem geometry is more complex than the TWIGL reactor because it explicitly represents the fuel rods and surrounding moderator. While the C5G7 benchmark problem only provided details for a steady-state calculation, we have defined a C5G7 transient based on an instantaneous control rod ejection. This transient does not incorporate thermal-hydraulic feedback which limits realism but allows us to focus on the solution of the transport equation.

8.2 TWIGL Transient Problem

8.2.1 TWIGL Problem Specification

The TWIGL reactor is a 2D seed-and-blanket geometry 1.6 m along each side. It is surrounded by a vacuum and typically modeled with one-quarter or one-eighth core symmetry. The problem geometry is displayed in Figure 8-1. The problem specifies one delayed neutron group and cross sections for two energy groups within three different material regions: Regions 1 and 2 are seed regions while Region 3 is the blanket region. The transient was driven by varying the thermal absorption cross section in Region 1.

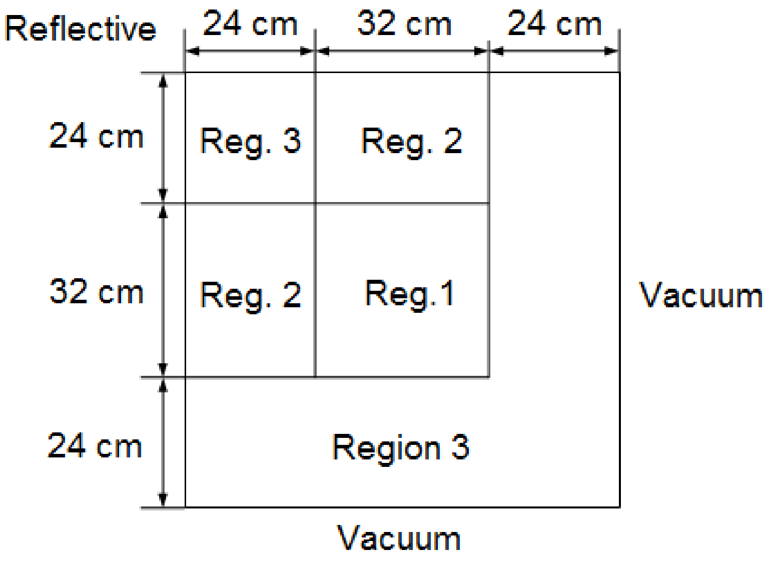


Figure 8-1. Geometry for the south-east quadrant of TWIGL [adapted from ANL05]

Table 8-1 provides both the TWIGL reactor macroscopic cross sections by region and the kinetics parameters. The arrows in the table indicate the cross sections that are linearly-ramped or step-changed in the TWIGL transients.

Table 8-1. Cross sections and kinetics parameters for the TWIGL Transients

Region	Group, g	Σ_a (cm^{-1})	$\nu\Sigma_f$ (cm^{-1})	Σ_t (cm^{-1})	$\Sigma_{s,g\to g}$ (cm^{-1})	$\Sigma_{s,g\to g'}$ (cm^{-1})
1	1	0.01	0.007	0.238095	0.218095	0.01
	2	0.15 → 0.1465	0.2	0.83333 → 0.82983	0.68333	0.0
2	1	0.01	0.007	0.238095	0.218095	0.01
	2	0.15	0.2	0.83333	0.68333	0.0
3	1	0.008	0.003	0.25641	0.23841	0.01
	2	0.05	0.06	0.666667	0.616667	0.0

$v_1 = 1 \times 10^7$ cm/s, $v_2 = 2 \times 10^5$ cm/s, $\chi_1 = 1$, $\chi_2 = 0$, $\beta = 0.0075$, $\lambda = 0.08\ \text{s}^{-1}$

We modeled the TWIGL reactor in DeCART using quarter-core symmetry with 100 square "assemblies" that were 32 cm on each side. These assemblies contained 4×4 cells that were 2 cm long and contained 25 square flat-source regions each[17]. The MOC ray spacing was 0.02 cm, and the number of azimuthal angles and polar angles in 90° were 8 and 4 respectively. The initial core power was normalized to one.

8.2.2 Steady-State Eigenvalue Solution

Each of the TWIGL transients begins from an assumed steady-state critical condition[18]. This state is initialized using an eigenvalue problem. We calculated the eigenvalue $k_{\text{eff}} = 0.916051$. This is in good agreement with results reported using the computer code VARIANT-K [Rin97] with comparable options ($k_{\text{eff}} = 0.916074$) [ANL05].

The steady-state power distribution by assembly for the TWIGL reactor is provided in Table 8-2. This power distribution is color-coded by magnitude with borders to indicate the three regions. The same data is presented in a 3D column plot in Figure 8-2, although this is color-coded by region.

[17] In section 8.1.1.1 we perform a parametric study of the sensitivity of SDP to these input parameters.

[18] The steady-state results for each transient method are identical regardless of the method in question.

Table 8-2. Steady-state power distribution for the SE quadrant TWIGL reactor

1.21	1.24	1.20	2.31	2.13	1.94	1.69	0.66	0.45	0.19
1.24	1.27	1.22	2.32	2.12	1.92	1.66	0.65	0.44	0.19
1.20	1.22	1.16	2.30	2.09	1.86	1.60	0.62	0.43	0.18
2.31	2.32	2.30	2.15	2.00	1.76	1.50	0.58	0.40	0.17
2.13	2.12	2.09	2.00	1.85	1.61	1.36	0.52	0.36	0.15
1.94	1.92	1.86	1.76	1.61	1.38	1.16	0.45	0.30	0.13
1.69	1.66	1.60	1.50	1.36	1.16	0.97	0.36	0.24	0.10
0.66	0.65	0.62	0.58	0.52	0.45	0.36	0.28	0.18	0.07
0.45	0.44	0.43	0.40	0.36	0.30	0.24	0.18	0.11	0.05
0.19	0.19	0.18	0.17	0.15	0.13	0.10	0.07	0.05	0.02

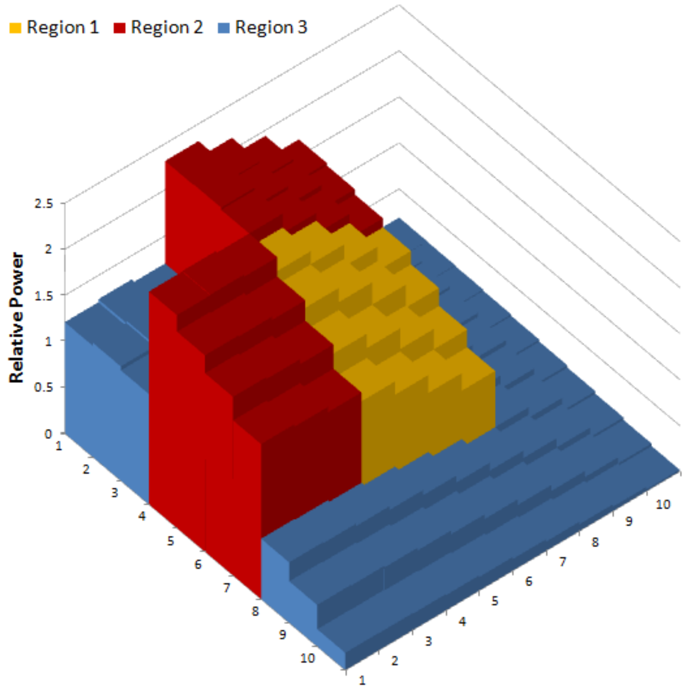


Figure 8-2. Steady-state power distribution for SE quadrant of TWIGL reactor

8.2.3 Linear Ramp Transient

The first transient evaluated is the TWIGL linear ramp. In this transient, the thermal absorption cross section in Region 1 is linearly decreased from 0.15 cm^{-1} to 0.1465 cm^{-1} over a 0.2 s period. This results in a slow exponential power increase which can be accurately modeled with a very large range of time steps, which is useful for error scaling. Note that the original specification of the TWIGL ramp transient involved modeling the transient out to 0.5 s even though the linear ramp ends at 0.2 s. We limited the transient to 0.2 s to limit the computational requirements of the model.

The reference solution used RBDC1 with 0.01 ms time steps. RBDC1 requires approximately 4 GB of RAM to store the angular fluxes for the current and previous point in time. A power trace for the TWIGL core is presented in Figure 8-3, with the final relative power of 1.980412. The reference solution required approximately 20 hours using eight 2.53 GHz Intel Xeon processors on the University of Michigan Center for Advanced Computing (CAC) network.

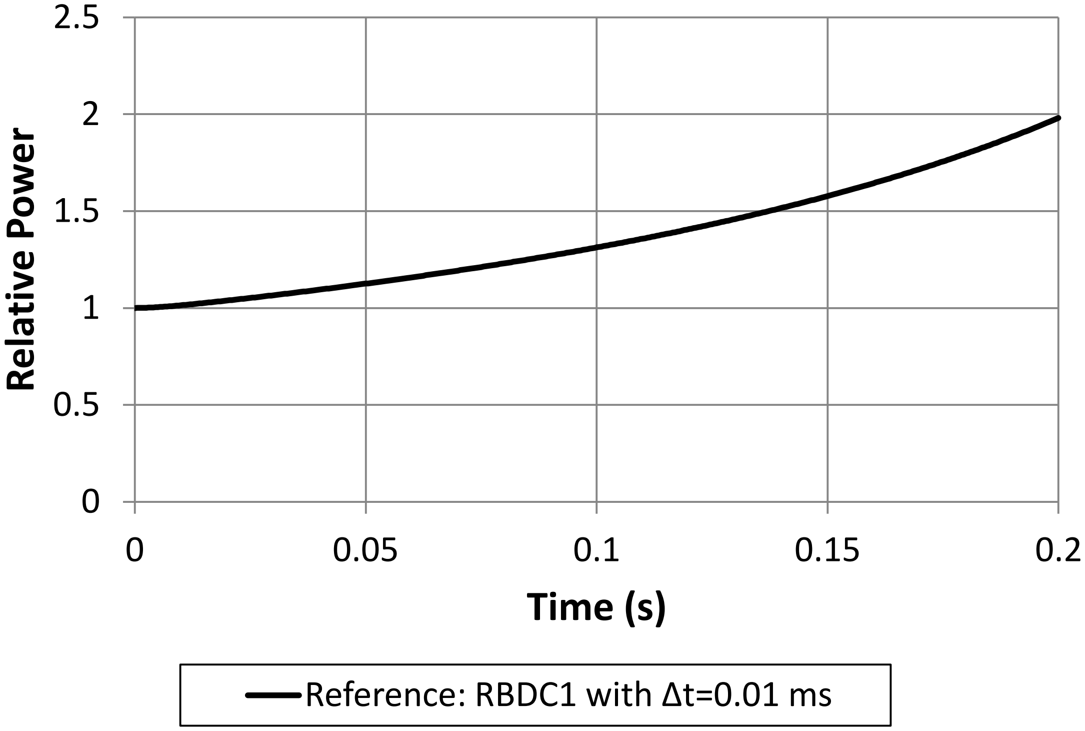


Figure 8-3. Relative core power for the ramp transient reference solution

In the following subsections we used RBDC to evaluate the performance of the SDP methods. In the first subsection we provided a brief parametric study for the TSDP1 to confirm that the method converges toward the RBDC1 solution as the input parameters are refined and that the accuracy of TSDP1 is not unduly sensitive to the input parameters. The next three subsections focused on the accuracy of the time integration of the SDP methods. The fifth subsection concerned the ability of the SDP method to capture the spatial variation of the solution. The last subsection compared the computational and memory requirements of the methods.

8.2.3.1 Input Parameter Sensitivity Evaluation

In this section we evaluated the sensitivity of the SDP methods to the input parameters to confirm that the SDP methods converge to the RBDC solution as the parameters are refined. Here the parameters of interest represent the angular and spatial meshing for MOC; the sensitivity of the methods to the temporal meshing is evaluated in detail in later sections. To minimize the sensitivity of the methods to the source convergence, in this section all of the methods performed two MOC iterations at every time step regardless of the source convergence[19].

For the TWIGL transients, the three SDP methods are in close agreement, and in this section we will only evaluate TSDP1. We assessed the sensitivity of the input parameters for TSDP1 to the RBDC1 solution by comparing the total power at the end of the transient. The four parameters we investigated included the number of azimuthal angles in 90°, the number of polar angles in 90°, the spacing between the characteristics (ΔK_l), and the number of regions per cell.

The results of the parametric study are presented in Figure 8-4 with the perturbations from the reference parameters above displayed on the x-axis. The first feature we note is that TSDP1 accurately replicates the RBDC1 solution for every adjustment of input parameters. Second, we note that while IBDC1 under-predicted the

[19] We determined two MOC iterations to be sufficient by performing a scoping calculation with RBDC1 using a source convergence criteria of 1×10^{-7}. The convergence criteria is still used for CMFD, but it only has a minor impact on the transient result.

final power for each case, the error for IBDC scales with RBDC and TSDP; this indicates that the methods had the same leading error terms in angle and space. In Chapter 6 we noted that the leading spatial component of the error for RBDC and IBDC was linear (Equations 6.27 and 6.40), but the leading spatial component of the error for the SDP methods was quadratic (Equations 6.53, 6.60, and 6.70). Nonetheless, the flat source approximation used with every method introduced a linear spatial error which is apparently dominant.

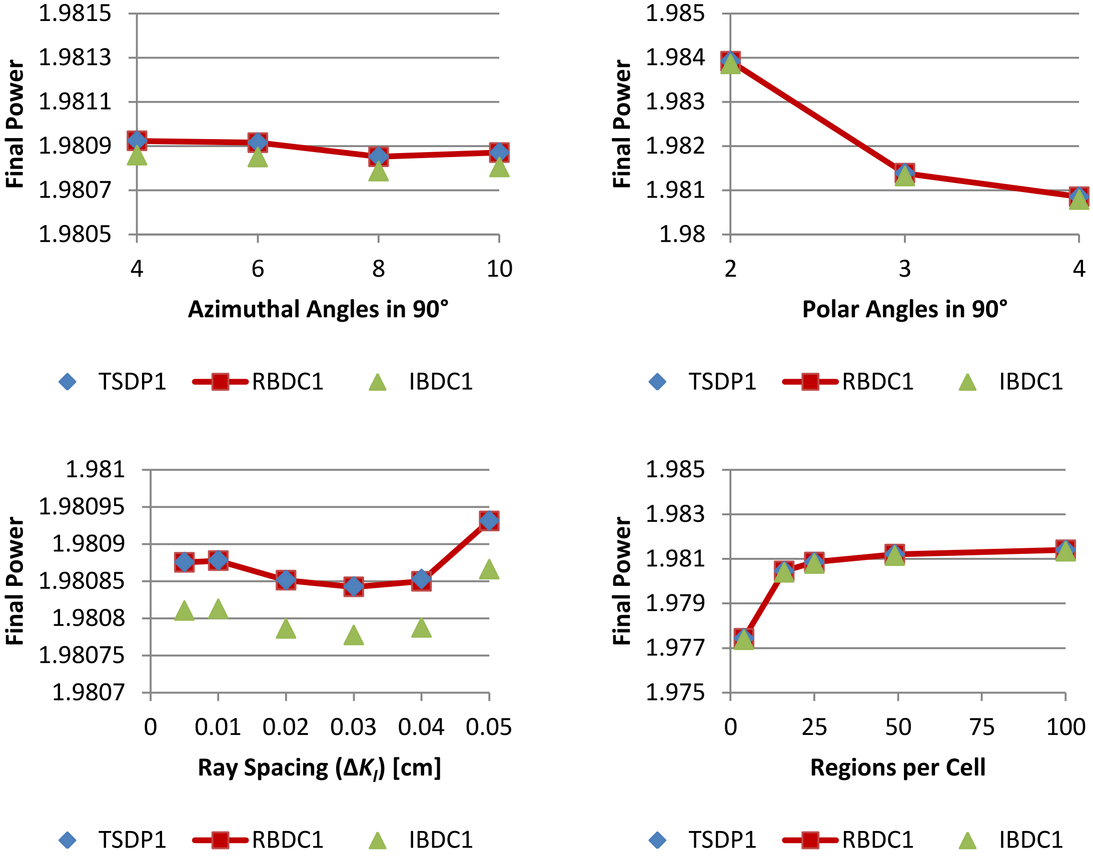


Figure 8-4. Results of parametric study for TSDP1, RBDC1, & IBDC1

8.2.3.2 SDP1 Convergence to Reference Solution as Time Step Decreases

In this section we assessed whether the SDP1 methods converge to the reference solution as the time step size decreases. To limit the sheer amount of data in each of the solutions while still being able to compare the results for many different time step sizes, we used the final power for the core at the end of the transient for comparison. Specifically, we compared the absolute magnitude of the relative difference in the final power:

$$\epsilon_X(\Delta t) = \left|\frac{P_X(\Delta t) - P_{\mathrm{RBDC1}}(0.01\ \mathrm{ms})}{P_{\mathrm{RBDC1}}(0.01\ \mathrm{ms})}\right|, \qquad 8.1$$

where $P_X(\Delta t)$ is the final power for method X with time step Δt. The absolute magnitude of the error is used because negative values cannot be plotted on a logarithmic scale.

In Figure 8-5 we plotted the error for TSDP1 on a log-log plot as a function of the time step size. The time steps varied over three orders of magnitude. We see that the error in the TSDP1 solution declines as the time step size decreases, indicating that the TSDP1 method is converging to the reference solution.

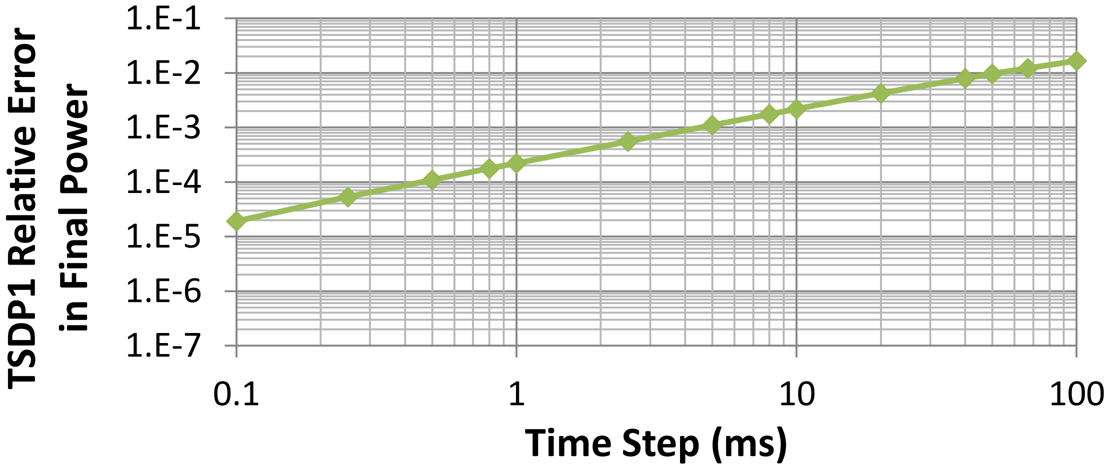


Figure 8-5. Relative error in the final power for TSDP1 as a function of time step size

The slope of the error in Figure 8-5 is indicative of the order of convergence of TSDP1 to the reference solution. Based on Equation 6.9 we calculate the slope of the error using the equation:

$$S_X(\Delta t_i) = \frac{\ln \epsilon_X(\Delta t_i) - \ln \epsilon_X(\Delta t_{i+1})}{\ln(\Delta t_i) - \ln(\Delta t_{i+1})}, \qquad 8.2$$

where Δt_i is one time step size and Δt_{i+1} is the next smaller time step size.

Using Equation 8.2, we determined that the average slope of the TSDP1 error is 0.951. This indicates that TSDP1 has first order convergence to the reference solution, which is consistent with the expectations for BDF1. This also suggests that the other error terms identified in Chapter 6 in Equation 6.53 (e.g. resulting from the neglect of the cross section derivatives and the second angular flux time derivative) are very small.

We observed similar trends for ISDP1 and ISCDP1, which were in very good agreement with TSDP1. Plots of the error for ISDP1 and ISCDP1 are visually indistinguishable from Figure 8-5.

8.2.3.3 SDPN Convergence to the Reference Solution as the Order Increases

Next we evaluated whether the high-order SDP methods converge to the reference solution as the order increases, and as the time step size decreases. Again, we compared the final power for each of the SDP methods to the reference solution using Equation 8.1. In Figure 8-6 the error is plotted on a log-log plot for all of the TSDP methods up to order six.

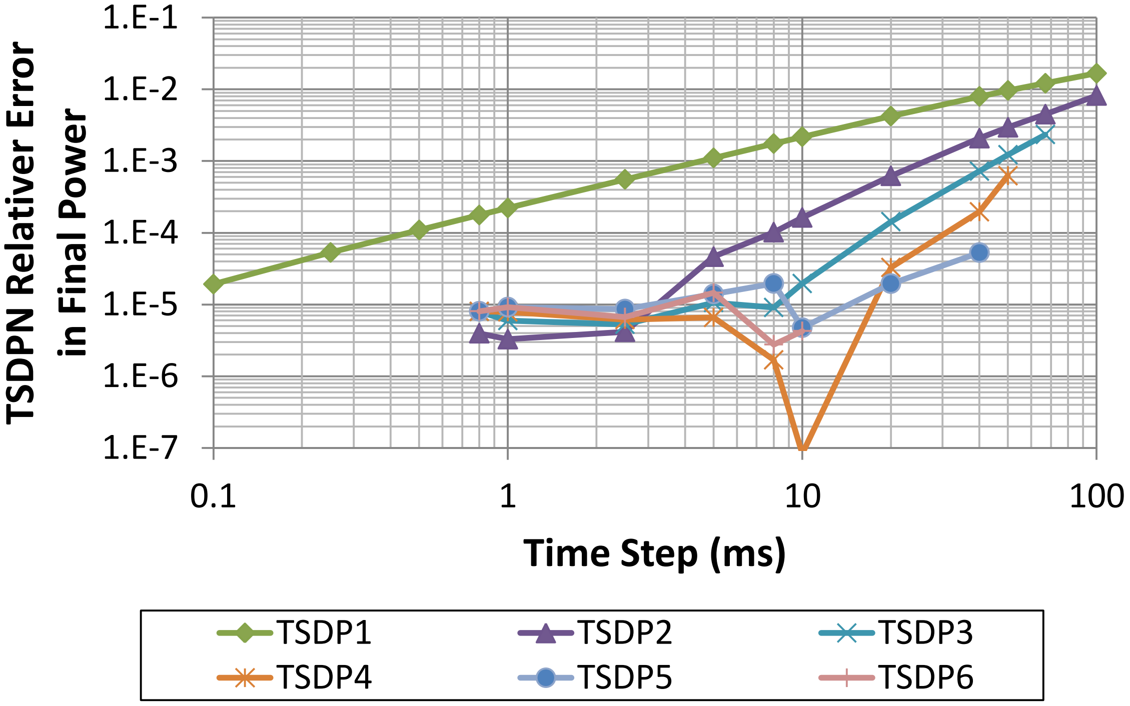


Figure 8-6. Relative error in the final power for TSDPN as a function of time step size

Here we observed that the error for each high-order TSDP method decreased with time step size until each method reaches a transition point after which the error briefly increases and stabilizes around 1×10^{-5}. For example, the transition point for TSDP4 was 10 ms. Upon closer inspection of the high-order TSDP solutions we noted that they were converging downward like RBDC. For each TSDP method, the error transition corresponds to the point where the high-order solutions are reporting a final power that is smaller than the reference solution (1.980412). The solutions of third-order and greater TSDP methods all ultimately converge to a final power of 1.980396. Thus the apparent increase in error in Figure 8-6 for the high-order methods at the transition reflects when the error goes from positive to negative. This is illustrated in Figure 8-7, which shows the final power for several TSDP methods as a function of time step size. The same trend is observed for the high-order RBDC methods and is illustrated in Section 8.2.3.4, confirming that this phenomena is not the result of the SDP methods.

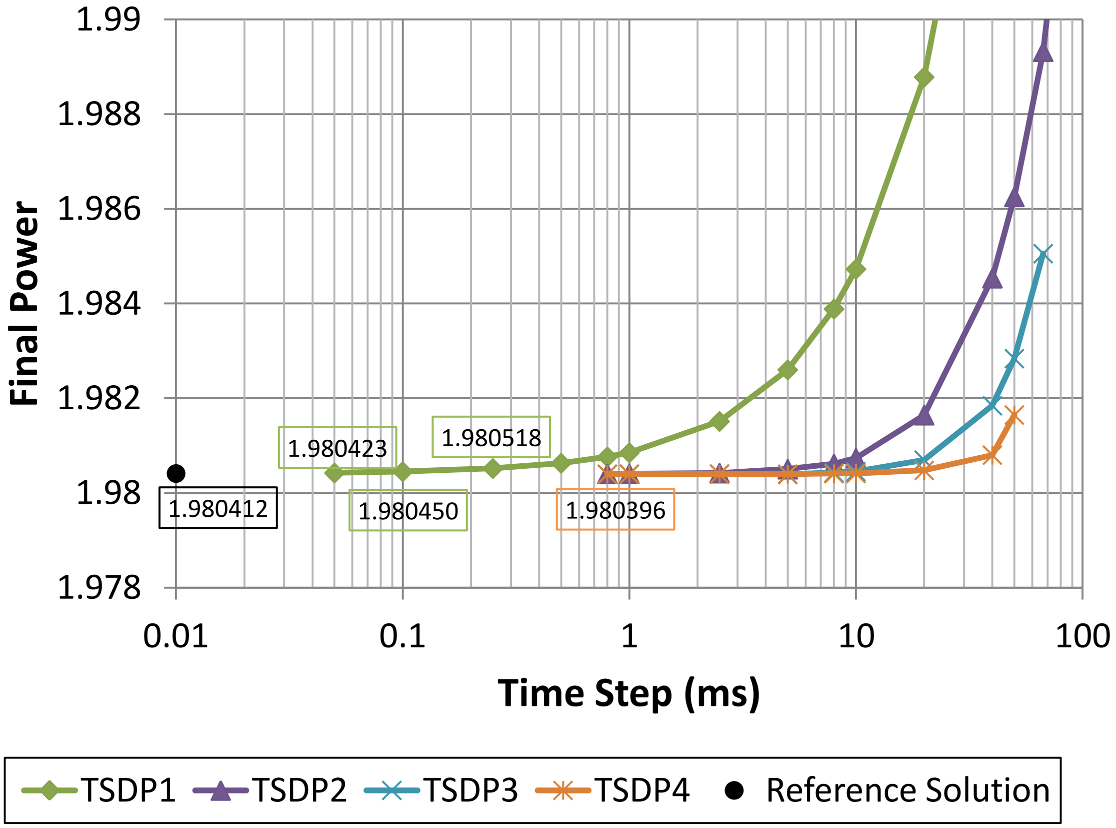


Figure 8-7. Final power as a function of time step size for TSDP1-4

Thus, the transition appears to correspond to the point that the high-order TSDP method becomes more accurate than the reference solution. This can only be confirmed with RBDC1 by generating a reference solution with a yet smaller time step, but doing so requires prohibitive run-times. However, we note that when the time step for the reference solution is adjusted upwards (e.g. to 0.025 ms or 0.05 ms) the transition point for each method shift upwards as well, which is consistent with this explanation.

Further, the possibility that the high-order time integration methods are able to accurately replicate the reference solution with very large time steps is not surprising. The transient behavior of the TWIGL ramp is characterized by a slow exponential, and it is not difficult to model the higher derivatives with backward differences. We will return to this topic in Section 8.2.3.4 where we use the assumption that the solution is exponential in time with the leading temporal error terms identified in Chapter 6 (e.g. Equation 6.27 and 6.30) to predict the transition time steps where the high-order methods become more accurate than the reference solution.

Note that some of the high-order TSDP methods do not have solutions for large time step sizes (e.g. the largest time step size for a TSDP4 solution is 50 ms). This is because the order of the approximation for the time derivative with backward differences is limited by the number of previous data available. As a result, when a high-order method is specified, DeCART begins the transient with a first-order method and ascends in method order as additional data becomes available. Since the transient is 200 ms long, 50 ms is the largest time step that allows for a fourth-order method to be used for at least one time step.

We can calculate the order of convergence to the reference solution for the high-order TSDP methods using Equation 8.2 averaged over all time step sizes. However, the error results below the transition are not meaningful, and thus they should be discarded from this measurement. Instead, we calculated the error slopes using only the data above the transition. The slopes of the TSDP methods are provided in Table 8-3. No slope could be calculated for TSDP-6 because the solution with the largest time step (20 ms) was smaller than the reference solution.

Table 8-3. Logarithmic error for TSDP methods discarding data below transition

Method	TSDP1	TSDP2	TSDP3	TSDP4	TSDP5	TSDP6
Slope	0.95	1.91	2.92	4.09	5.30	no data

As expected, the slopes increased for the higher-order TSDP methods. The slopes are close to the BDF order for each method, although there were few data points for the TSDP4 and TSDP5. The slopes for the higher-order TSDP methods are smaller for the transients with very few time steps because the error from the lower-order methods that are used to initialize the high-order method contaminate the final solution. Similar trends were observed for ISDP and ISCDP, which were in close agreement with the TSDP. Error plots for ISDP and ISCDP are visually indistinguishable from Figure 8-6.

8.2.3.4 SDPN Comparison to RBDCN for Equal Time Step Size

In this section we compared the TSDP methods with RBDC methods of the same order and time step. Figure 8-8 shows the relative error for the high-order RBDC methods plotted as dashed lines with the error for the high-order TSDP methods plotted as symbols in the same color as Figure 8-6. The error is nearly identical for each order

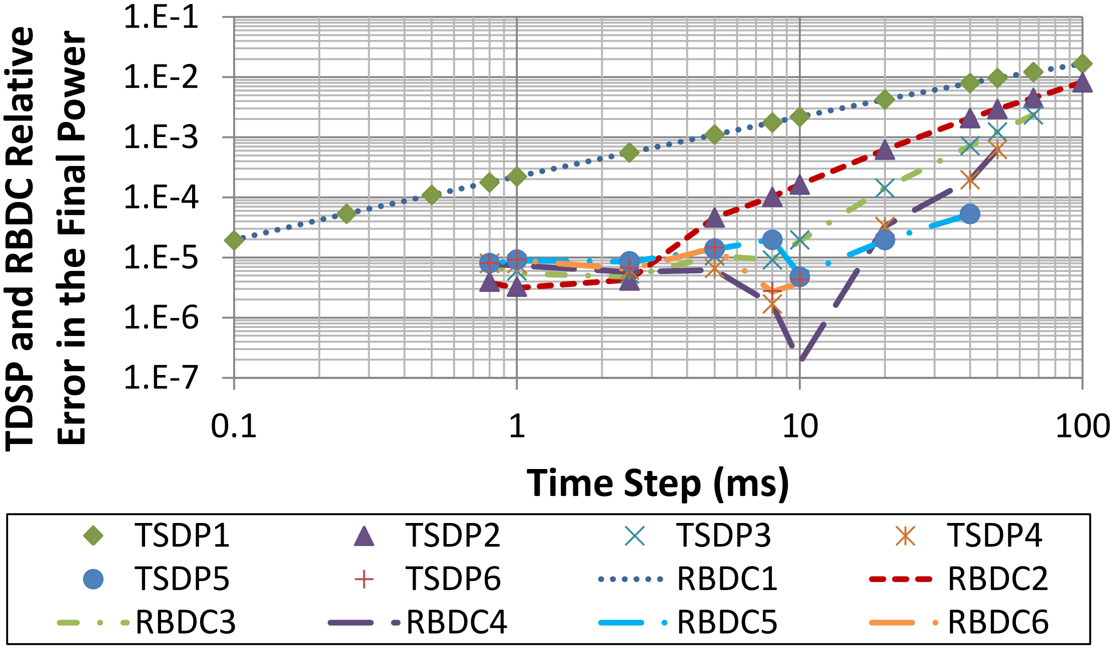


Figure 8-8. Relative error in the final power for TSDPN and RBDCN as a function of Δt

and time step size because the RBDC and TSDP methods are in excellent agreement. This also confirms that the "error transition" in Figure 8-6 is not unique to the SDP methods. We can estimate when the error transition will occur by comparing the leading temporal error term for the reference solution with that of a high-order method with a larger time step. Although it is difficult to compare the leading temporal error terms for RBDC to TSDP because they are expressed in terms of different quantities, it is easier to compare the leading temporal error terms for RBDC methods of different orders. The error transition for RBDCN will occur when the leading temporal error term for the reference solution is equal to the leading temporal error for RBDC*N* (i.e. Equation 6.31 with $\Delta t_{\mathrm{REF}} = 1 \times 10^{-5}$ s).

If we assume that the angular flux is exponential in time, we can estimate the higher derivatives of the angular flux using:

$$\frac{d^N \varphi(t)}{dt^N} = \alpha^N \varphi(0) e^{\alpha t}, \qquad 8.3$$

where α is the inverse period of flux. The inverse period was calculated to be 3.415 s^{-1} using a final relative power of 1.980412 at 0.2 s.

The error transition for RBDC*N* is expected when the following is satisfied:

$$\frac{\left(1 \times 10^{-5}\ \mathrm{s}\right)}{2} \alpha^2 = \frac{\Delta t^N}{(N+1)!} \alpha^N, \qquad 8.4$$

where we have applied Equation 8.3 to Equation 6.31 and eliminated the common terms on each side. Table 8-4 shows the solution of this equation for each N as well as the actual error transition observed in this transient. We note that the actual error transition for RBDC2 was very close, and the other methods were reasonably close. The actual transitions were lower than expected because the high-order methods are initialized using low-order method until enough previous data points are available to make a high-order approximation of the time derivative. These initial low-order time steps prevent the high-order methods from achieving genuine high-order accuracy, and the effect is greatest when there are relatively few time steps in the transient. In any case, this reiterates that the cause of the error transition is that the reference solution is less accurate than some of the higher-order solutions.

Table 8-4. Predicted and actual error transition time steps for RBDC and TSDP

N	2	3	4	5	6
RBDC*N* Predicted	3 ms	20 ms	60 ms	120 ms	190 ms
RBDC*N* Actual	2.5 ms	8 ms	10 ms	40 ms	> 20 ms
TSDP*N* Actual	2.5 ms	8 ms	20 ms	40 ms	> 20 ms

In Figure 8-8 we compared the TSDPN solutions to the RBDCN solutions indirectly by observing that their relative error was approximately the same for each order and time step size. To compare the results more directly, we calculated the absolute difference in final power for the RBDC and TSDP methods:

$$\Delta P_N(\Delta t) = |P_{\mathrm{RBDC}N}(\Delta t) - P_{\mathrm{TSDP}N}(\Delta t)|, \tag{8.5}$$

where we again use the absolute magnitude of the difference because we will plot the result on a log-log plot.

To reduce the amount of visual clutter, we plotted these results in two graphs. Figure 8-9 shows the absolute difference for the odd-order methods while Figure 8-10 shows the difference for even order methods. Both plots also show the difference for IBDC1, which is the popular isotropic approximation for time-dependent MOC.

The absolute difference in the final power for the RBDC and TSDP methods is in the range of 1x10^{-7} to 1x10^{-6}. Closer observation of the output files for the ramp transient has shown that the RBDC and TSDP methods are in excellent agreement throughout the transient with occasional deviations. These deviations occur when a convergence criterion for one method (e.g. RBDC) falls just above the acceptance threshold whereas for the other method (e.g. TSDP) it falls just below. This results in one method having an extra iteration which results in an increased discrepancy; this is the explanation for the small oscillations in the absolute difference for some methods in

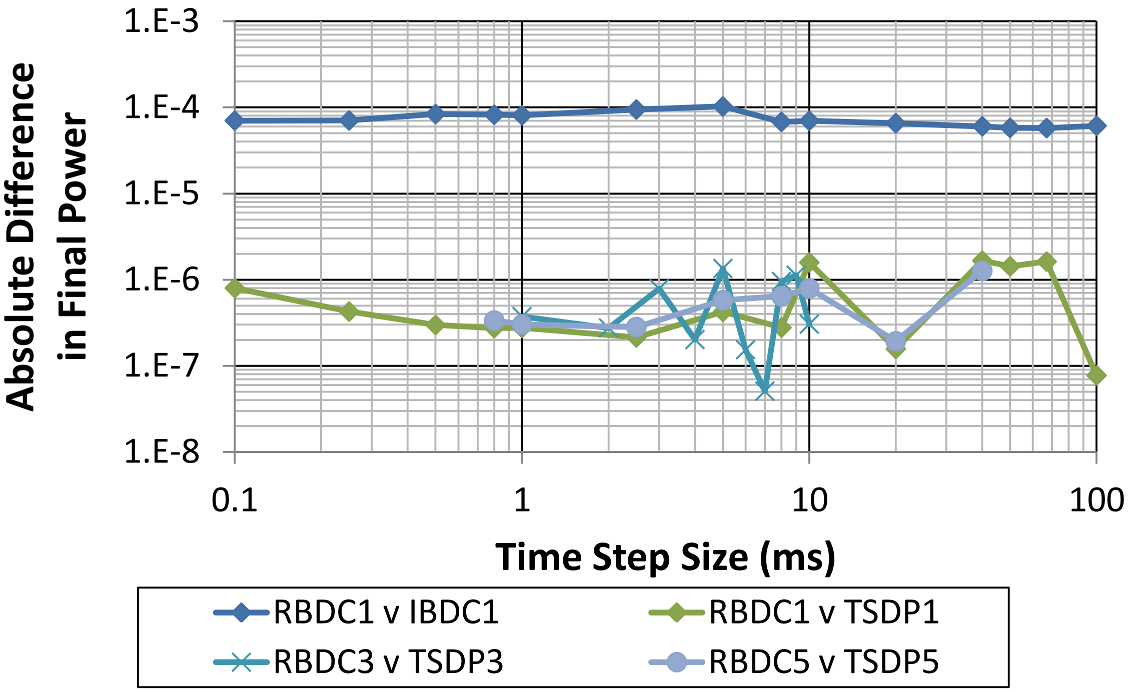


Figure 8-9. The absolute difference in peak power for TSDP/IBDC and RBDC with the same order and time step size: odd order methods

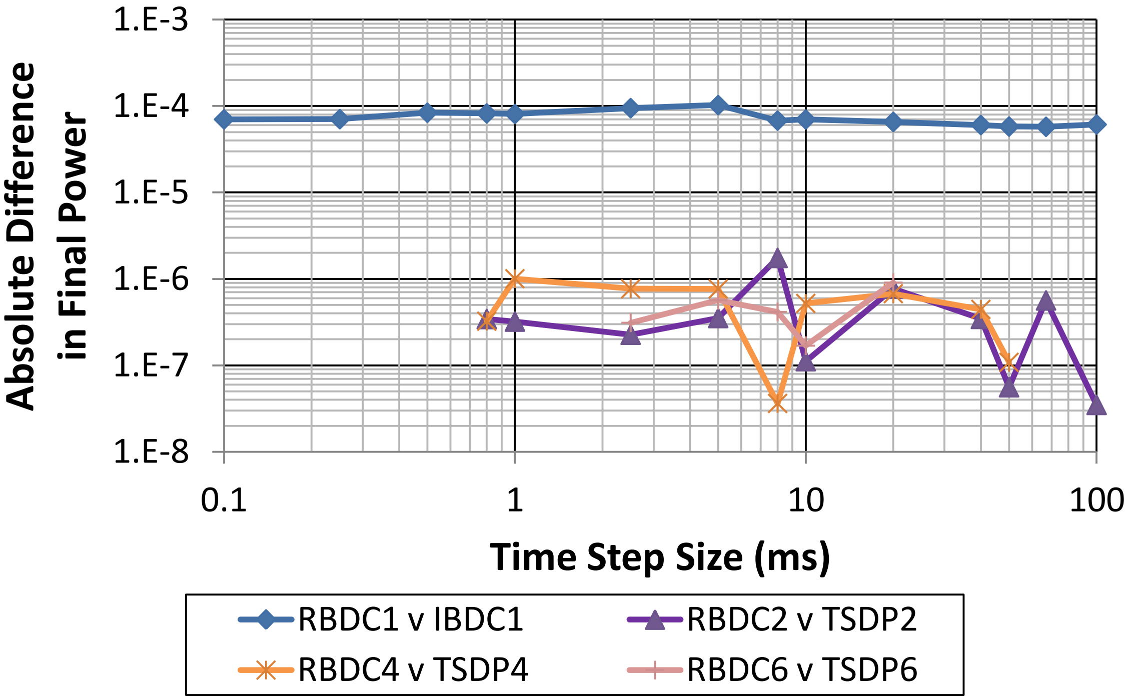


Figure 8-10. The absolute difference in peak power for TSDP/IBDC and RBDC with the same order and time step size: even order methods

Figure 8-9 and Figure 8-10. If the SDP and RBDC methods are hardcoded to use the same number of iterations, the discrepancy diminishes; this was demonstrated in the results for the TWIGL step change transient. The TSDP and ISDP/ISCDP methods also deviated from each other by a similar margin for the same reason.

The TSDP methods more accurately replicated the RBDC solution to the ramp transient than the IBDC methods. The difference in the final power for the IBDC methods is about two orders of magnitude larger than the TSDP methods. This primarily reflects the error in neglecting the angular-dependence of the angular flux time derivative. The error for IBDC is nonetheless very small.

It is well established [But08] that BDF methods of order 7 and higher are unstable, and we have observed this instability for RBDC and SDP methods. While these methods are not practically useful, it is noteworthy that the close agreement between the SDP and RBDC methods extends to these higher-order methods. Figure 8-11 and Figure 8-12 illustrate this agreement for the seventh and eighth order methods respectively.

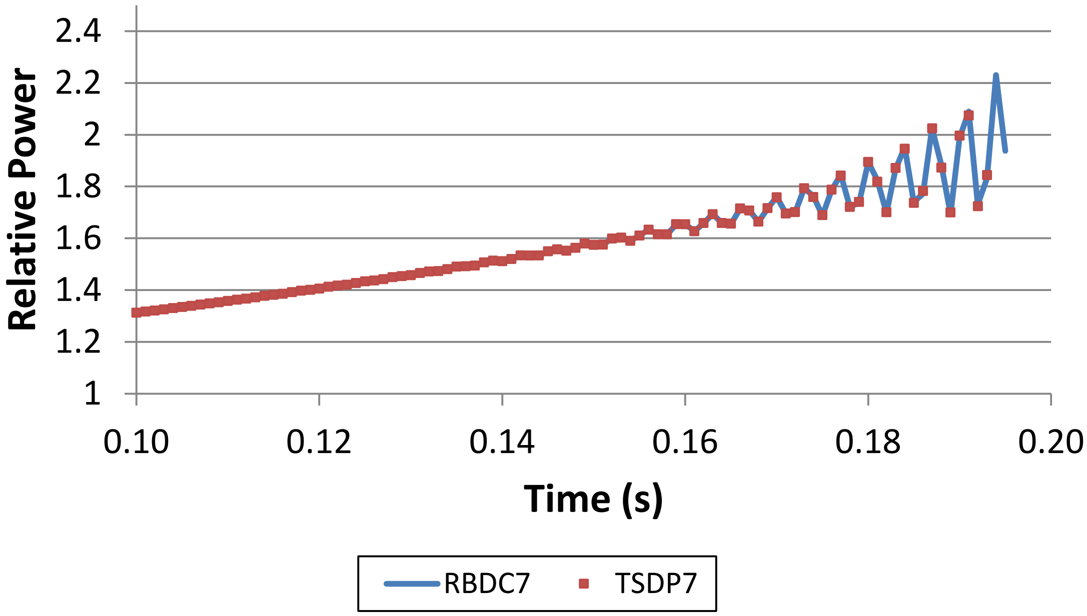


Figure 8-11. Time-dependent relative power for RBDC7 and TSDP7

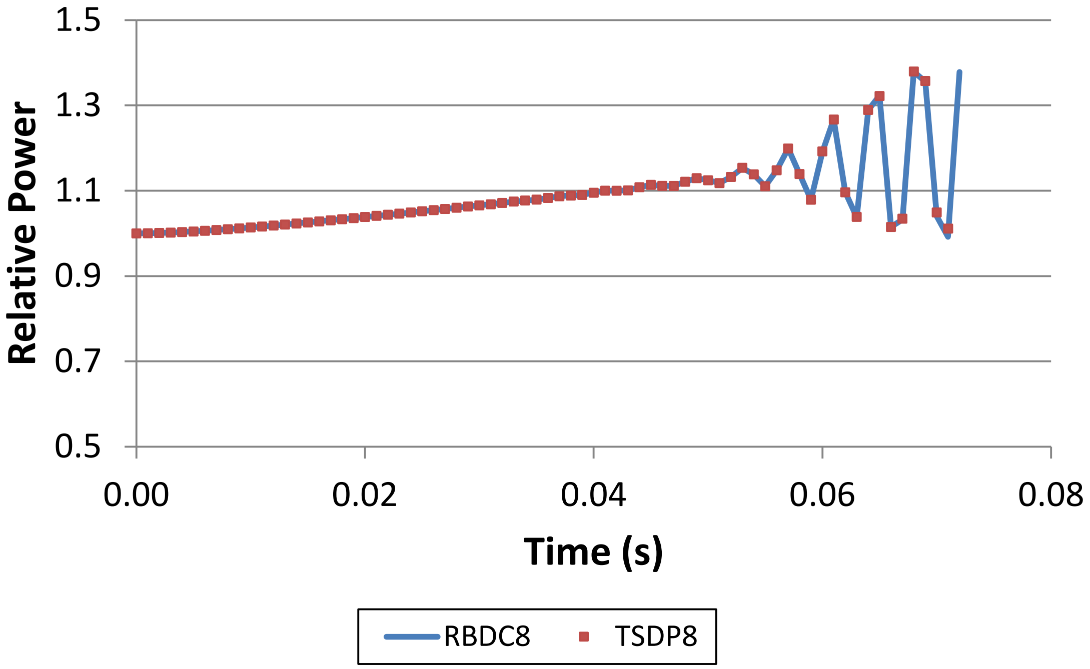


Figure 8-12. Time-dependent relative power for RBDC8 and TSDP8

8.2.3.5 Comparison of SDP and RBDC Spatial Power Distribution

To this point we have used the total core power at the end of the transient as a metric for assessing accuracy. In this subsection we focused on the cell-wise reactor power distribution to assess whether the SDP methods are accurately capturing the spatial variation in the solution. As a reminder, in our TWIGL geometry the quarter-core symmetric south-east quadrant of the reactor is composed of 1600 cells, where the cell dimensions are 2 cm × 2 cm. The cells consist of 25 equal size square flat source regions. The cell-wise powers were edited to the DeCART output file to seven decimal places.

In this subsection we used the results of the TWIGL ramp transient with a time step size of 1 ms; these results are typical of other time step sizes. For the reference solution, we used the RBDC method of the same order as the SDP or IBDC method. In other words, to assess the accuracy of SDP*N* and IBDC*N*, we used the RBDC*N* solution.

Figure 8-13 provides the color-coded cell-wise relative power distribution for RBDC1 at time 0.025 s as an example. The peak power is 2.60966 at position (8,13) and (13,8) as measured from the center of the core; this position is along the interior face of the seed region facing the inner blanket.

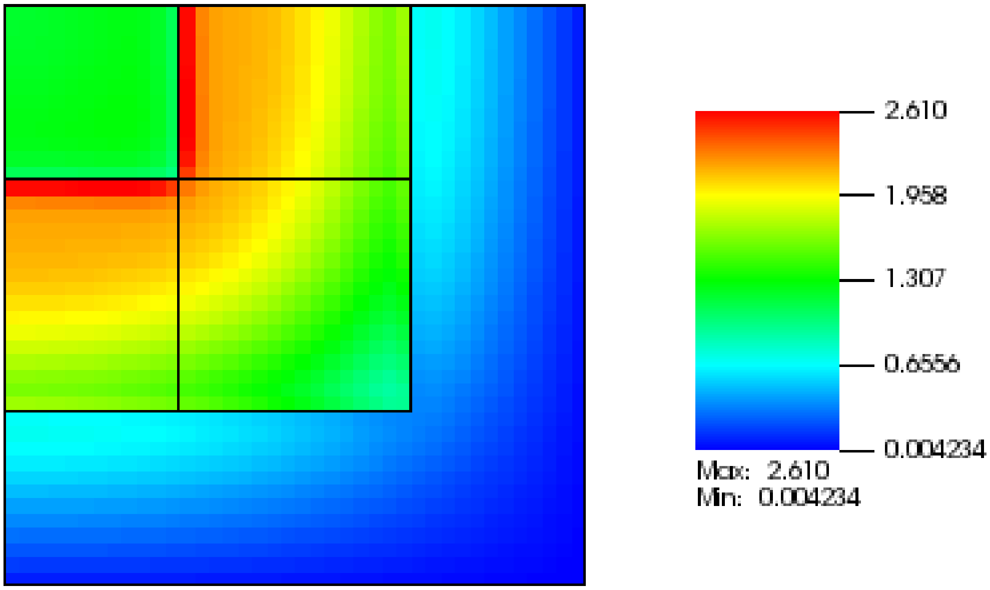


Figure 8-13. The cell-wise relative power for the reference solution at 0.025 s

We assessed the spatial accuracy using several measures of error: the average relative error, the L2 error, the maximum absolute error, the maximum relative error, the absolute error for the peak cell, and the relative error for the peak cell. The average error for method X is calculated using:

$$\epsilon_{X,\text{avg}}(t) = \frac{1}{C}\sum_{c=1}^{C}\left|\frac{P_{X,c}(t) - P_{\text{Ref},c}(t)}{P_{\text{Ref},c}(t)}\right|, \qquad 8.6$$

where $P_{X,c}(t)$ is the power for method X in cell c at time t and C is the total number of cells.

The L2 error is calculated using an L2 norm of the relative error:

$$\epsilon_{X,\text{L2}}(t) = \sqrt{\frac{\sum_c^C \left(P_{X,c}(t) - P_{\text{Ref},c}(t)\right)^2}{\sum_c^C P_{\text{Ref},c}(t)^2}}. \qquad 8.7$$

The maximum absolute error is simply the absolute error in the cell with the largest absolute error:

$$\epsilon_{X,\text{maxA}}(t) = \max_c\left|P_{X,c}(t) - P_{\text{Ref},c}(t)\right|. \qquad 8.8$$

Similarly, the maximum relative error is the relative error in the cell with the largest relative error:

$$\epsilon_{X,\mathrm{maxR}}(t) = \max_c \left| \frac{P_{X,c}(t) - P_{\mathrm{Ref},c}(t)}{P_{\mathrm{Ref},c}(t)} \right|. \tag{8.9}$$

The peak absolute error is the absolute error in the cell that is producing the most power in the reference solution:

$$\epsilon_{X,\mathrm{peakA}}(t) = P_{X,\mathrm{peak}}(t) - P_{\mathrm{Ref,peak}}(t), \tag{8.10}$$

where the peak cell power is:

$$P_{\mathrm{Ref,peak}}(t) = \max_c\left[P_{\mathrm{Ref},c}(t)\right]. \tag{8.11}$$

Similarly, the peak relative error is the relative error in the cell that is producing the most power in the reference solution:

$$\epsilon_{X,\mathrm{peakR}}(t) = \frac{P_{X,\mathrm{peak}}(t) - P_{\mathrm{Ref,peak}}(t)}{P_{\mathrm{Ref,peak}}(t)}. \tag{8.12}$$

Using these definitions, we calculated the time-dependent errors for TSDP1 and present the results in Table 8-5. The average and L2 error in the cell-wise power are small. They are also comparable in magnitude to the error in the total power at the end of the transient, which indicates that the error in the final power is representative of the

Table 8-5. Time-dependent error in the relative cell-wise power for TSDP1

Time (s)	$\epsilon_{X,\mathrm{avg}}$	$\epsilon_{X,\mathrm{L2}}$	$\epsilon_{X,\mathrm{maxA}}$	$\epsilon_{X,\mathrm{maxR}}$	$\epsilon_{X,\mathrm{peakA}}$	$\epsilon_{X,\mathrm{peakR}}$
0.025	4.86E-08	7.46E-08	3.00E-07	9.97E-07	-1.00E-07	-3.83E-08
0.050	5.03E-08	7.21E-08	3.00E-07	8.00E-07	-1.00E-07	-3.84E-08
0.075	4.93E-08	7.26E-08	3.00E-07	1.31E-06	-1.00E-07	-3.85E-08
0.100	5.08E-08	7.15E-08	3.00E-07	2.26E-06	-1.00E-07	-3.86E-08
0.125	4.69E-08	7.08E-08	3.00E-07	1.14E-06	-1.00E-07	-3.86E-08
0.150	5.22E-08	7.30E-08	3.00E-07	1.65E-06	-1.00E-07	-3.87E-08
0.175	5.17E-08	7.37E-08	3.00E-07	1.22E-06	-2.00E-07	-7.76E-08
0.200	5.37E-08	7.44E-08	3.00E-07	1.45E-06	-1.00E-07	-3.89E-08

average error. The error is relatively constant through-out the transient. The error in the peak cell is approximately 1×10^{-7}, which was the last decimal edited by DeCART.

To illustrate the spatial variation of the error, in Figure 8-14 we present a pseudo-color plot of the relative error cell-wise relative power at the end of the transient for TSDP1 for the southeast quadrant; this is equivalent to evaluating Equation 8.12 for every cell. We see that the largest relative error is scattered in the outer blanket region. This occurred because these cells have a small absolute power, and those sporadic erroneous cells only differ in power by ±1×10^{-7}. The largest absolute error occurred within the ramped seed region, where the power was slightly over predicted. The error is very small near the peak cells, which do not change over the transient.

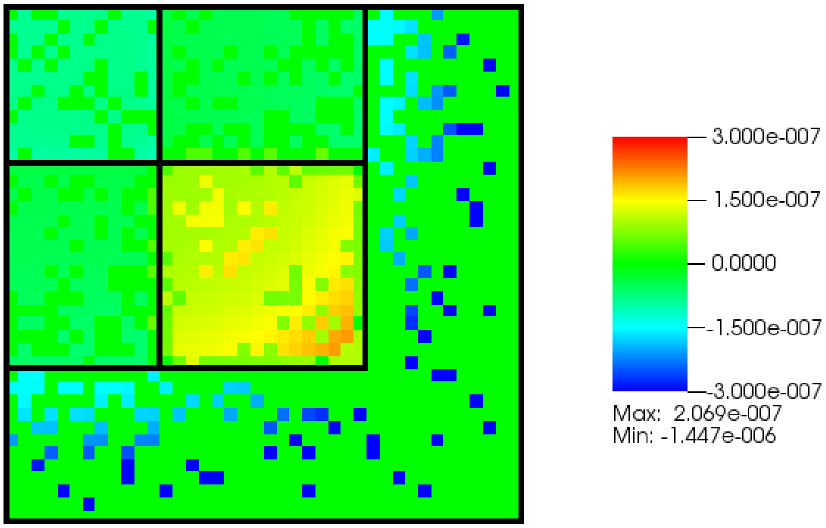


Figure 8-14. Relative error in the relative cell-wise power distribution for TSDP1

We also evaluated the errors for ISDP1 and presented the results in Table 8-6. We see that the incorporation of the second derivative approximation had a very minor effect on the accuracy of the SDP method, although the accuracy was strictly improved. This is because the TSDP1 was already accurate, and the second derivative of the angular flux is small for this transient. The spatially-dependent error for ISDP1 is essentially indistinguishable from that of TSDP1, so we will not present it graphically.

Table 8-6. Time-dependent error in the relative cell-wise power for ISDP1

Time (s)	$\epsilon_{X,\text{avg}}$	$\epsilon_{X,\text{L2}}$	$\epsilon_{X,\text{maxA}}$	$\epsilon_{X,\text{maxR}}$	$\epsilon_{X,\text{peakA}}$	$\epsilon_{X,\text{peakR}}$
0.025	4.71E-08	7.27E-08	3.00E-07	9.97E-07	-1.00E-07	-3.83E-08
0.050	5.07E-08	7.27E-08	3.00E-07	8.00E-07	-1.00E-07	-3.84E-08
0.075	5.01E-08	7.21E-08	3.00E-07	1.31E-06	-1.00E-07	-3.85E-08
0.100	5.03E-08	7.08E-08	3.00E-07	2.26E-06	-1.00E-07	-3.86E-08
0.125	4.49E-08	7.08E-08	3.00E-07	7.97E-07	-1.00E-07	-3.86E-08
0.150	5.23E-08	7.31E-08	3.00E-07	1.65E-06	-1.00E-07	-3.87E-08
0.175	5.20E-08	7.26E-08	3.00E-07	1.22E-06	-1.00E-07	-3.88E-08
0.200	5.17E-08	7.43E-08	3.00E-07	1.45E-06	-1.00E-07	-3.89E-08

We also calculated the spatial error for ISCDP1, which unlike the SDP methods does not assume that the cross section is changing slowly with time. The results are displayed in Table 8-7. The results are very similar to ISDP1, indicating that the assumption that the cross section is changing slowly for those methods has a limited impact. Like ISDP1, the spatially-dependent error for ISCDP1 is visually indistinguishable from that of TSDP1.

Table 8-7. Time-dependent error in the relative cell-wise power for ISCDP1

Time (s)	$\epsilon_{X,\text{avg}}$	$\epsilon_{X,\text{L2}}$	$\epsilon_{X,\text{maxA}}$	$\epsilon_{X,\text{maxR}}$	$\epsilon_{X,\text{peakA}}$	$\epsilon_{X,\text{peakR}}$
0.025	4.72E-08	7.32E-08	3.00E-07	9.97E-07	-1.00E-07	-3.83E-08
0.050	4.91E-08	7.20E-08	3.00E-07	8.00E-07	-1.00E-07	-3.84E-08
0.075	4.92E-08	7.24E-08	3.00E-07	1.31E-06	-1.00E-07	-3.85E-08
0.100	5.04E-08	7.15E-08	3.00E-07	2.26E-06	-1.00E-07	-3.86E-08
0.125	4.51E-08	7.09E-08	3.00E-07	7.97E-07	-1.00E-07	-3.86E-08
0.150	5.23E-08	7.31E-08	3.00E-07	1.65E-06	-1.00E-07	-3.87E-08
0.175	5.08E-08	7.25E-08	3.00E-07	1.22E-06	-1.00E-07	-3.88E-08
0.200	5.24E-08	7.40E-08	3.00E-07	1.45E-06	-1.00E-07	-3.89E-08

Table 8-8 displays the error for IBDC1. The maximum and peak error for IBDC1 is an order of magnitude larger than the SDP methods, while the average and L2 errors are one to two orders of magnitude larger. In addition, unlike the SDP methods, these errors systematically increase over the transient, which explains the larger discrepancy in accuracy for IBDC observed in the previous section. The errors for IBDC1 are nonetheless very small, and the accuracy is probably adequate for many applications.

Table 8-8. Time-dependent error in the relative cell-wise power for IBDC1

Time (s)	$\epsilon_{X,\mathrm{avg}}$	$\epsilon_{X,\mathrm{L2}}$	$\epsilon_{X,\mathrm{maxA}}$	$\epsilon_{X,\mathrm{maxR}}$	$\epsilon_{X,\mathrm{peakA}}$	$\epsilon_{X,\mathrm{peakR}}$
0.025	7.16E-07	5.33E-07	2.70E-06	2.36E-05	2.00E-06	7.66E-07
0.050	7.81E-07	5.81E-07	2.90E-06	2.36E-05	2.20E-06	8.45E-07
0.075	8.55E-07	6.30E-07	3.20E-06	2.35E-05	2.30E-06	8.85E-07
0.100	9.03E-07	6.87E-07	3.50E-06	5.91E-06	2.60E-06	1.00E-06
0.125	1.02E-06	7.55E-07	3.80E-06	9.22E-06	2.50E-06	9.66E-07
0.150	1.13E-06	8.33E-07	4.20E-06	2.33E-05	2.70E-06	1.05E-06
0.175	1.25E-06	9.31E-07	4.70E-06	1.29E-05	3.00E-06	1.16E-06
0.200	1.39E-06	1.05E-06	5.30E-06	9.15E-06	3.40E-06	1.32E-06

In Figure 8-15 we present the relative error in the cell-wise relative power for IBDC1. Note that the scale is an order of magnitude larger than that of Figure 8-14; the error for the IBDC and SDP methods cannot be meaningfully compared on the same

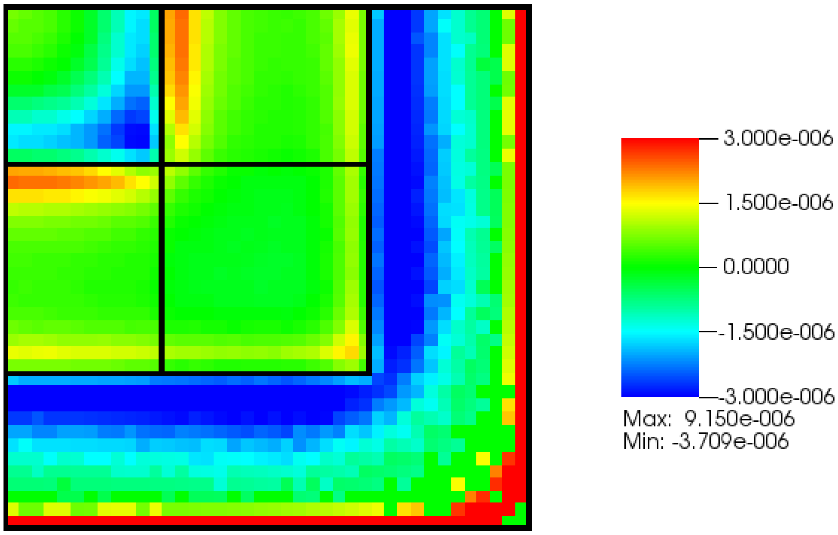


Figure 8-15. Relative error in the relative cell-wise power distribution for IBDC1

scale. The relative error for IBDC1 is greatest at the core periphery where the relative power is low. However, there is also substantial error at the interfaces between the seed and blanket regions. This includes the peak cells, which are near the interface of the seed and internal blanket. This error is likely the result of IBDC's inability to capture the changing angular flux distribution at the material interfaces. However, the error is nonetheless relatively small throughout the problem.

We also evaluated the error for the high-order SDP methods by comparison to the RBDC methods of the same order. The errors for the high-order methods are all comparable to their low-order counterparts. As an example, we presented the error for TSDP6 in Table 8-9. The spatial dependence of the error for the high-order SDP methods is similar to the spatial error for the first-order methods.

Table 8-9. Time-dependent error in the relative cell-wise power for TSDP6

Time (s)	$\epsilon_{X,\mathrm{avg}}$	$\epsilon_{X,\mathrm{L2}}$	$\epsilon_{X,\mathrm{maxA}}$	$\epsilon_{X,\mathrm{maxR}}$	$\epsilon_{X,\mathrm{peakA}}$	$\epsilon_{X,\mathrm{peakR}}$
0.025	5.67E-08	7.99E-08	3.00E-07	1.26E-06	-1.00E-07	-3.83E-08
0.050	5.60E-08	8.14E-08	3.00E-07	7.86E-07	-1.00E-07	-3.84E-08
0.075	5.51E-08	8.18E-08	3.00E-07	1.14E-06	-1.00E-07	-3.85E-08
0.100	5.30E-08	8.04E-08	3.00E-07	9.16E-07	-1.00E-07	-3.86E-08
0.125	5.76E-08	8.12E-08	3.00E-07	1.75E-06	-2.00E-07	-7.73E-08
0.150	5.56E-08	8.19E-08	3.00E-07	7.54E-07	-2.00E-07	-7.74E-08
0.175	5.97E-08	8.21E-08	3.00E-07	8.27E-07	-2.00E-07	-7.76E-08
0.200	5.95E-08	8.28E-08	3.00E-07	3.88E-06	-1.00E-07	-3.89E-08

8.2.3.6 Comparison of Computational and Memory Requirements

There are substantial differences between the computational and memory requirements of the SDP and BDC methods. In this section, we compared the relative run-time and memory requirements of the SDP and BDC methods for a representative TWIGL ramp transient. These results are representative of the TWIGL step change transient in the next section, and thus we did not provide additional run-time and memory results for that transient.

To assess the relative computational expense of the different MOC methods, we compared the run-time for several methods for the ramp transient with a 0.5 ms time step using eight Intel Xeon processors. The run-times are compared in Figure 8-16.

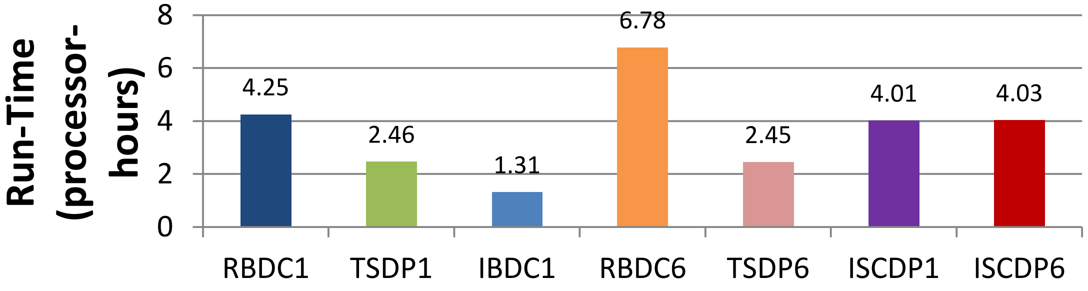


Figure 8-16. Run-time for a TWIGL ramp transient with a 0.5 ms time step

The IBDC methods represent a baseline for efficient time-dependent MOC because they incorporate the time derivative into an isotropic, modified source. This allows the IBDC method to use the same MOC equations as steady-state MOC. The SDP and RBDC methods all involve additional terms or equations that are evaluated for each segment. As a result, the SDP and RBDC methods are necessarily more computationally expensive.

Although the TSDP equations appear more complex than the RBDC equations, it is noteworthy that TSDP1 requires roughly two-thirds as much run-time as RBDC-1[20]. This is because so many of the TSDP terms are isotropic and can be pre-constructed by region before beginning MOC sweeps. As a result, TSDP requires about two-thirds as many arithmetic operations as RBDC1 per segment. In fact, the relative run-times correlate closely with the number of arithmetic operations performed per segment because this dominates the computational expense of DeCART. The run-time for ISDP methods and TSDP methods are indistinguishable, and thus TSDP methods are not shown in Figure 8-16.

[20] Because the RBDC methods involve very large variables for storing the angular flux, this could potentially increase the run-time if the memory is not managed well. We have taken care to avoid this by (a) storing the angular fluxes in memory in the order that they are required rather than by region and (b) using pointers to avoid moving the angular flux variables in memory at the end of each time step.

It is also noteworthy that the high-order representation of the source time derivative for TSDP6 does not impact the run-time over TSDP1. By contrast, RBDC6 increases the run-time over RBDC1 by about 60%. This is because it requires several arithmetic operations for every segment to construct the angularly-dependent BDF approximation of the time-derivative for RBDC6, but for TSDP6 the high-order isotropic source derivative approximation is pre-constructed by region. This is a major advantage of having the time derivative defined at the region-level rather than the segment-level.

We also note that incorporating the cross section derivative using ISCDP substantially increased the computation requirements over TSDP and ISDP. However, for many transients the cross sections are not changing quickly in most of the space and time domain, and we could limit the use of ISCDP to time steps and locations where the cross sections have changed substantially and use ISDP otherwise. This would substantially reduce the computational requirements without sacrificing accuracy. In this case, the run-time for ISCDP will converge toward ISDP as the number of time steps and locations that require ISCDP decrease.

We used the same transient to evaluate the memory requirements for the various methods. The memory requirements are plotted in Figure 8-17 in units of GB RAM. Note that the vertical scale is logarithmic due to the large difference in memory required for the various methods.

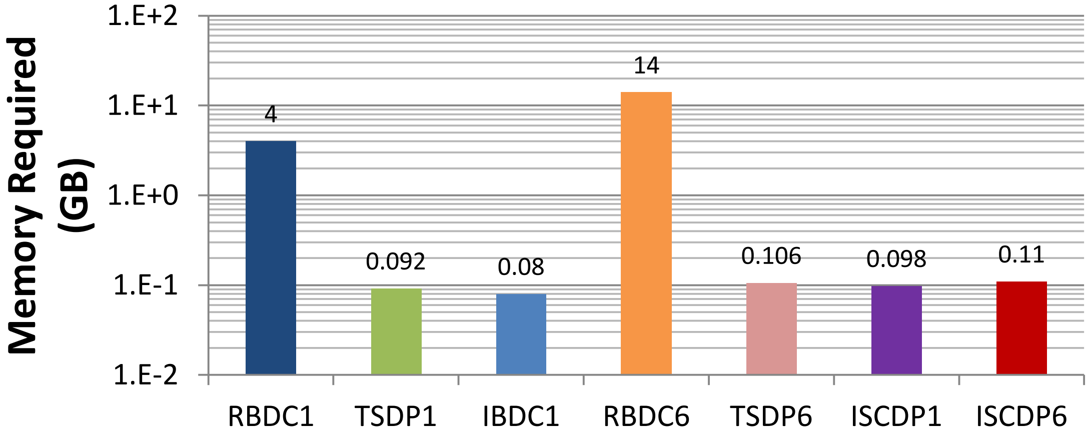


Figure 8-17. Memory requirements for a TWIGL ramp transient

Again, the IBDC method forms a baseline for the memory requirements of time-dependent MOC. The TSDP methods require somewhat more memory than IBDC because they necessitate the storage of the previous neutron sources as well as the angular flux along the problem boundary if the problem has non-vacuum boundary conditions. This increase in memory is comparable to storing the first moment of the neutron flux for each region, and it is small compared to the memory cost of storing the angular flux for RBDC. Storing the angular flux increases the memory requirements of RBDC over IBDC and TSDP by two orders of magnitude for the TWIGL transients.

While storing additional previous angular fluxes for high-order representation of the angular flux time derivative can substantially improve the accuracy of time-dependent MOC using RBDC, it linearly increases the memory required to model the transient. This is because the storage of the angular flux dominates the memory required for the model. By contrast, storing additional previous neutron sources or scalar fluxes to improve the accuracy of the time derivative for TSDP and IBDC only has a modest impact on the memory requirements of the transient. This is another major advantage to using terms defined at the region-level rather than the segment-level to represent the time derivative.

8.2.4 Step Change Transient

A step change transient is also defined for the TWIGL problem. This transient begins with a step change in the thermal absorption cross section in region 1 equal to the total change in the linear ramp transient. While the original TWIGL problem specified that the transient was to be modeled to 0.5 s, we limited the model to 0.2 s to reduce the computational burden.

The reference solution was generated using RBDC1 with a 0.025 ms time step with a final power of 2.105263. Figure 8-18 shows the power trace for the reference.

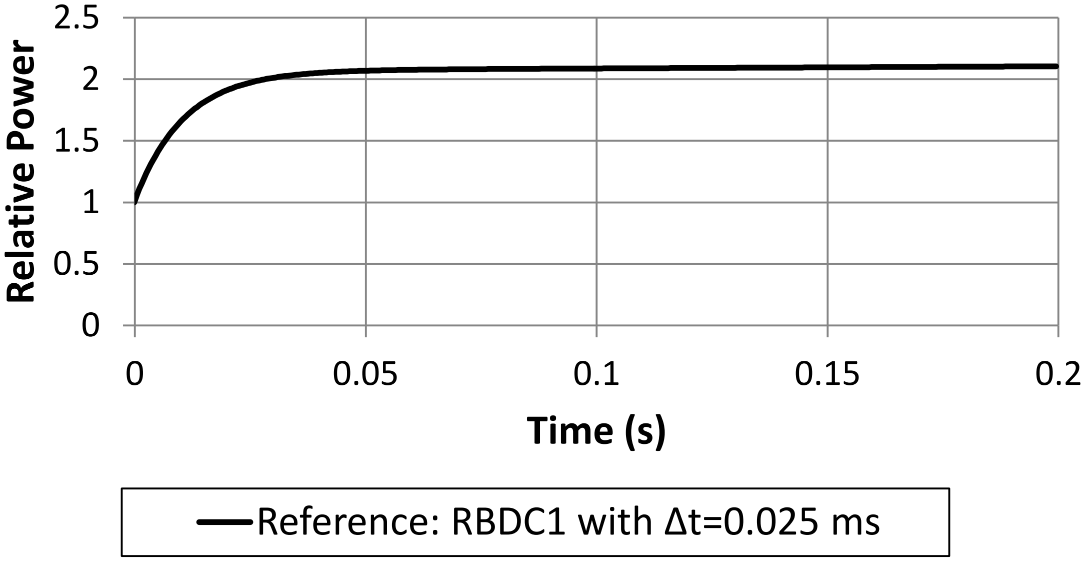


Figure 8-18. Relative core power for the step transient reference solution

The step change transient is faster than the linear ramp transient and should be more difficult for the SDP methods. In the following subsections we will assess whether the SDP methods converge to the reference solution and whether they replicate the RBDC solution for the same order and time step size. We will not discuss the memory and run-time requirements for the step change transient, because the results are essentially identical to the ramp transient.

8.2.4.1 SDP1 Convergence to Reference Solution as Time Step Decreases

In this section we assessed whether the TSDP1 solution converges to the reference solution as the time step decreases. We used the reactor power at the end of the transient to assess the solution accuracy as for the TWIGL ramp. The relative error in the final power is defined using Equation 8.1. Figure 8-19 provides a log-log plot of the error as a function of the time step size. Unlike the TWIGL ramp, here we see that the solution converges linearly when the time step is below a critical value of about 10 ms; above this threshold the solution is inaccurate. This trend was observed for all methods (SDP, RBDC, and IBDC), and reflects the inability of large time steps to capture the fast step change at the beginning of the transient.

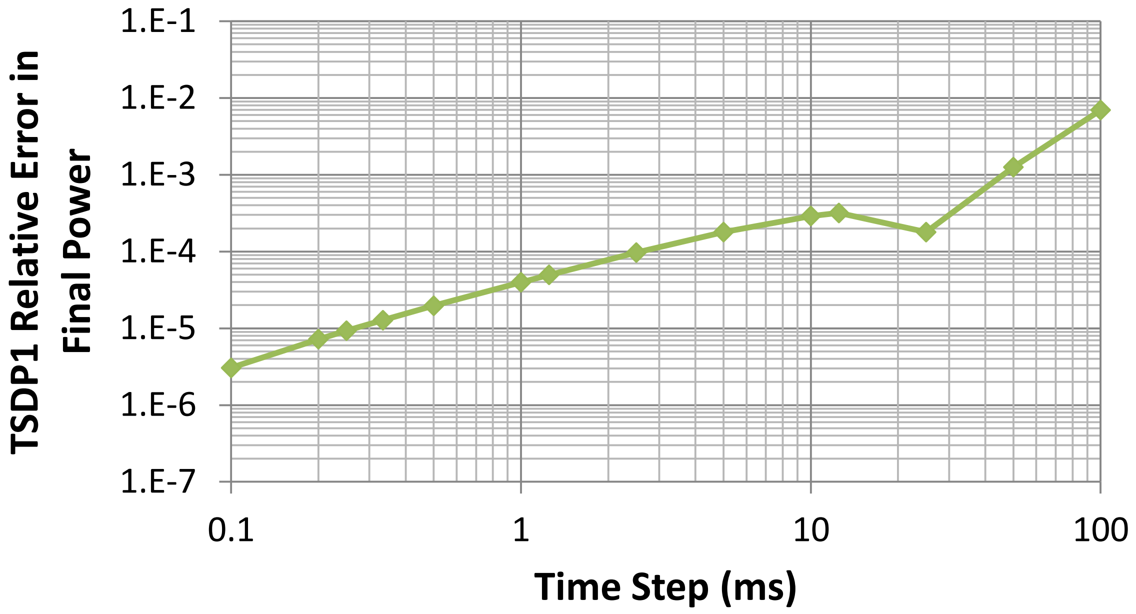


Figure 8-19. Relative error in the final power for TSDP1 as a function of Δ*t*

As with the TWIGL ramp, we can calculate the order of convergence of TSDP1 to the reference solution based on the slope of Figure 8-19 using Equation 8.2. If we exclude the results for time steps greater than 10 ms, the average slope is 1.021, indicating first-order convergence to the reference solution. As for the TWIGL ramp, the solutions using ISDP1 and ISCDP1 were close to the solution for TSDP1, and error plots for ISDP1 and ISCDP1 are visually indistinguishable from Figure 8-19.

8.2.4.2 SDPN Convergence to the Reference Solution as the Order Increases

Next we assessed whether the SDP solutions increase in accuracy as the method order increases. Again we used the relative error in the final power as a metric. Figure 8-20 presents the error for the high-order TSDP methods. Like TSDP1, the high-order TSDP methods exhibit nonlinear convergence to the reference solution with time steps larger than 10 ms. In fact, when the time steps are larger than 10 ms the high-order methods may be less accurate than the first-order method, which was a possibility we discussed in Chapter 6. For time step sizes below 10 ms, we observe that the high-order TSDP methods become more accurate as the time step size reduces until they reach a transition where the error drops and then stabilizes near 1×10^{-6}. This transition corresponds to the point where the high-order method becomes more accurate than the reference solution. A similar feature was observed for the TWIGL ramp transient.

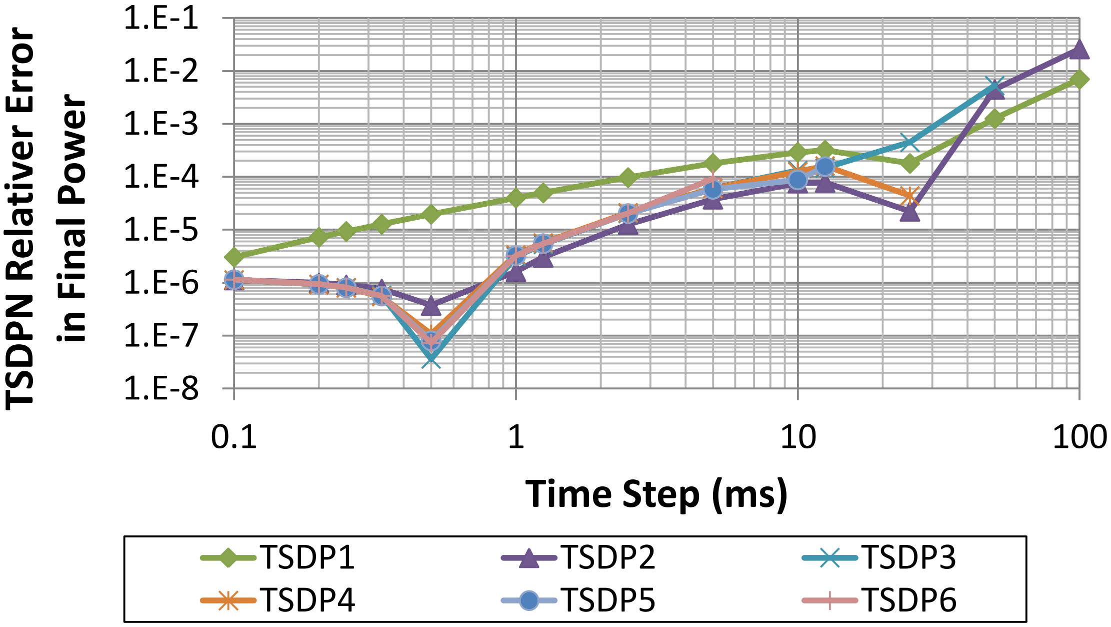


Figure 8-20. Relative error in the final power for TSDPN as a function of Δ*t*

Unlike the TWIGL ramp, for this transient the high-order TSDP methods do not converge quickly to the reference with large time steps. Instead, their error closely tracks the error of TSDP2. This is because the fast elements of the transient occur in the first few time steps. The high-order methods require the solution from many previous data points to accurately represent the higher derivatives. As a result, unless the time steps are very small, the fast aspect of the transient is over before high-order approximations are possible. This limitation affects all BDF methods, including RBDC.

We may nonetheless calculate the order of convergence for the high-order methods using Equation 8.2. Table 8-10 presents the error slopes when the data with time steps above 10 ms and below 1 ms is excluded. This confirms that the error for the high-order methods in this region is roughly quadratic. Similar trends were observed for ISDP and ISCDP methods.

Table 8-10. Logarithmic error slopes for TSDP methods, 1 ms ≤ Δt ≤ 10 ms

Method	TSDP1	TSDP2	TSDP3	TSDP4	TSDP5	TSDP6
Slope	1.021	1.971	2.730	2.397	2.424	2.694

8.2.4.3 SDPN Comparison to RBDCN For Same Time Step Size

In this subsection we compared the TSDP methods with RBDC methods of the same time step and order. The relative error for the high-order methods is displayed in Figure 8-21 with the error for the high-order RBDC methods plotted as dashed lines and the error for the high-order TSDP methods plotted as symbols. The error is nearly identical for each time step and method because the RBDC and TSDP methods are in excellent agreement. This agreement includes the inaccurate behavior for all methods with time steps above 10 ms, confirming that this is a feature of the transient rather than the SDP methods.

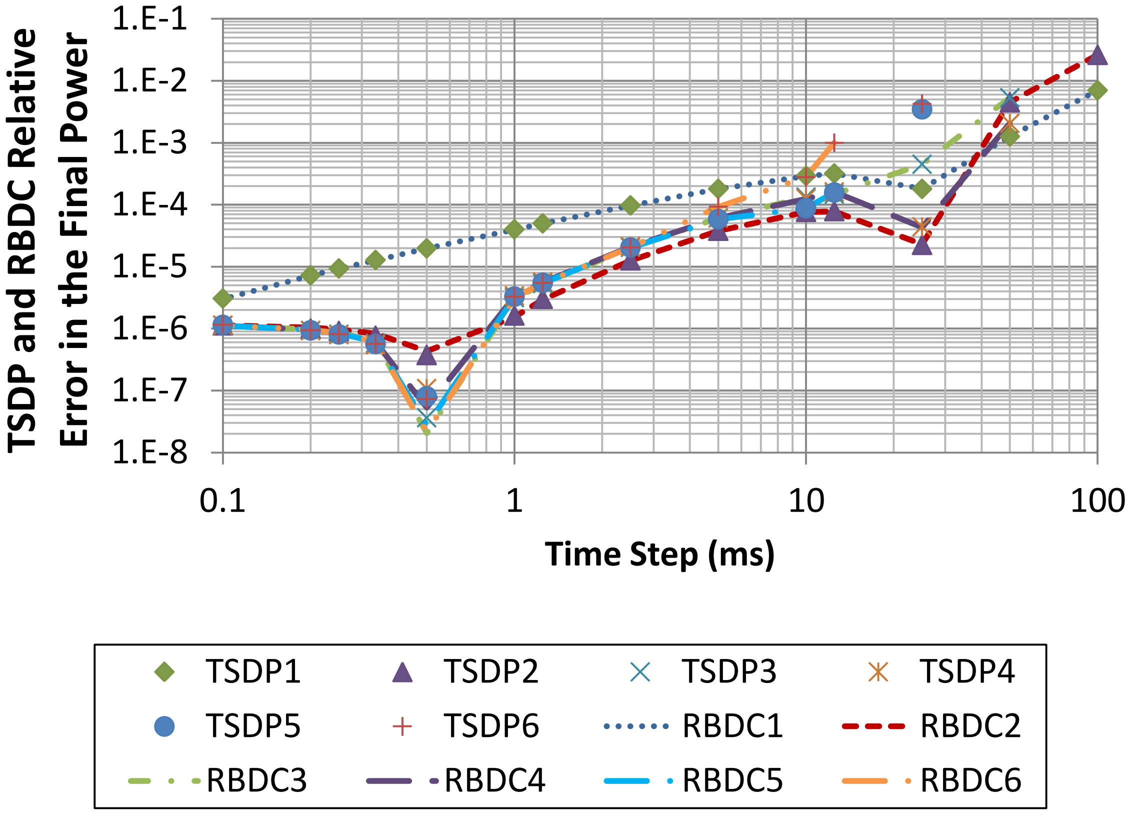


Figure 8-21. Relative error in the final power for TSDPN & RBDCN vs Δt

We compared the TSDP and RBDC methods directly by calculating the absolute difference in final power using Equation 8.5. These results are presented in Figure 8-22 for the odd-order methods and Figure 8-23 for the even-order methods.

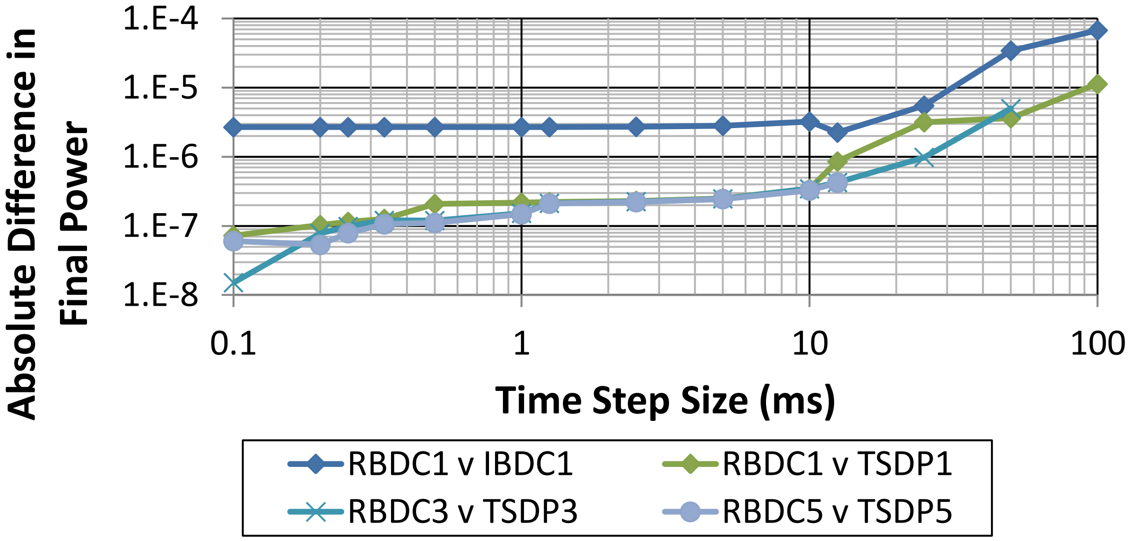


Figure 8-22. The absolute difference in peak power for TSDP/IBDC and RBDC with the same order and time step size: odd-order methods

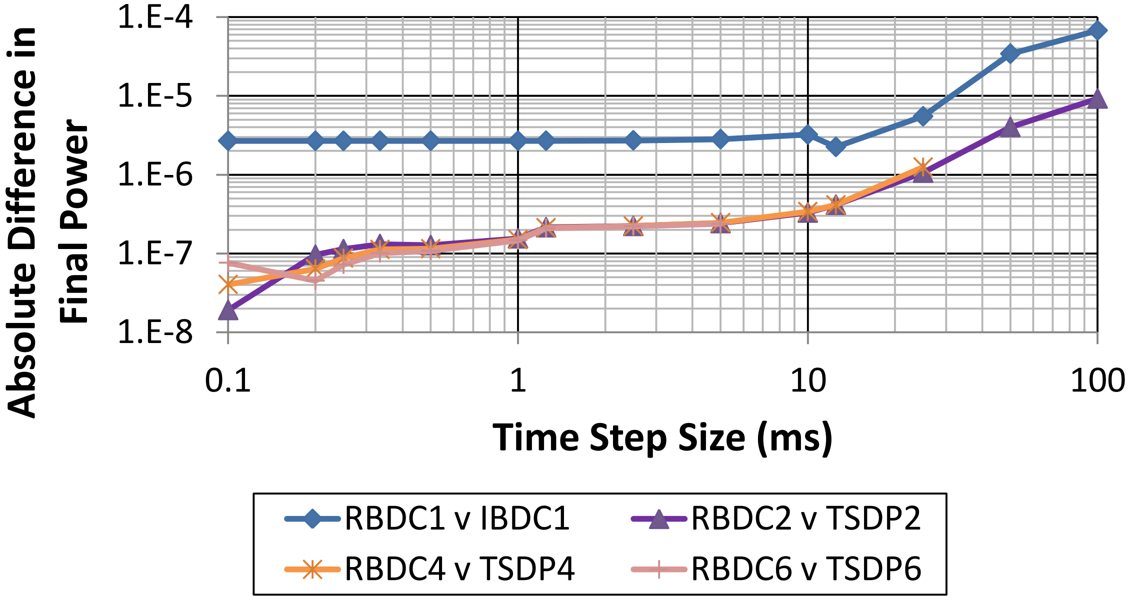


Figure 8-23. The absolute difference in peak power for TSDP/IBDC and RBDC with the same order and time step size: even-order methods

As for the TWIGL ramp, we see that the TSDP methods accurately capture the results of the RBDC methods for the same time step size and order. The discrepancy is largest for time steps greater than 10 ms and diminishes slowly as the time step is reduced. In contrast to Figure 8-9 and Figure 8-10 for the TWIGL ramp, the absolute difference does not oscillate with time step; this is because we forced all methods to use the same number of MOC iterations as the RBDC methods. Similar differences were observed for ISDP and ISCDP. We also note that the IBDC method did better for the TWIGL step change transient than it did for the TWIGL ramp transient; this is because the step change transient is characterized by a large amplitude change in contrast the slow continuous change for the TWIGL ramp transient.

8.2.4.4 Comparison of SDP and RBDC Spatial Power Distribution

Next we assessed the spatial accuracy of the SDP methods by comparing the cell-wise powers. In this subsection we used the results of the TWIGL step transient with a 1 ms time step. As for the TWIGL ramp, we calculated several measures of error for each cell: the average relative error, the L2 error, the maximum absolute error, the maximum relative error, the absolute error for the peak cell, and the relative error for the peak cell. These errors are calculated using Equations 8.6—8.12.

The time-dependent errors for TSDP1 are presented in Table 8-11. The spatial errors are even smaller than the errors for the TWIGL ramp transient. There is a general trend for the errors to decrease with time which reflects the decay of the higher derivatives as the reactor power trace flattens. As the higher derivatives decrease, all of the leading error terms in Equation 6.53 diminish as well. The error in the peak cell is less than 1×10^{-7}.

Table 8-11. Time-dependent error in the relative cell-wise power for TSDP1

Time (s)	$\epsilon_{X,\text{avg}}$	$\epsilon_{X,\text{L2}}$	$\epsilon_{X,\text{maxA}}$	$\epsilon_{X,\text{maxR}}$	$\epsilon_{X,\text{peakA}}$	$\epsilon_{X,\text{peakR}}$
0.025	1.38E-08	2.21E-08	1.00E-07	1.51E-06	0.00E+00	0.00E+00
0.050	2.62E-09	7.70E-09	1.00E-07	1.23E-06	0.00E+00	0.00E+00
0.075	4.13E-10	5.62E-09	1.00E-07	1.53E-07	0.00E+00	0.00E+00
0.100	3.06E-10	4.87E-09	1.00E-07	9.29E-08	0.00E+00	0.00E+00
0.125	1.06E-10	2.81E-09	1.00E-07	8.51E-08	0.00E+00	0.00E+00
0.150	2.99E-10	3.98E-09	1.00E-07	1.56E-07	0.00E+00	0.00E+00
0.175	1.57E-10	3.98E-09	1.00E-07	6.40E-08	0.00E+00	0.00E+00
0.200	2.88E-10	3.44E-09	1.00E-07	1.86E-07	0.00E+00	0.00E+00

We also analyzed the time-dependent error for ISDP1, which was very similar to TSDP1. These results are presented in Table 8-12. Note that the cell-wise powers were identical for ISDP1 and RBDC1 at 0.125 s. These results indicate that the impact of incorporating the second time derivative of the scalar flux was limited for this transient.

Table 8-12. Time-dependent error in the relative cell-wise power for ISDP1

Time (s)	$\epsilon_{X,\text{avg}}$	$\epsilon_{X,\text{L2}}$	$\epsilon_{X,\text{maxA}}$	$\epsilon_{X,\text{maxR}}$	$\epsilon_{X,\text{peakA}}$	$\epsilon_{X,\text{peakR}}$
0.025	1.49E-08	2.53E-08	1.00E-07	1.51E-06	0.00E+00	0.00E+00
0.050	2.66E-09	7.70E-09	1.00E-07	1.23E-06	0.00E+00	0.00E+00
0.075	2.92E-10	3.98E-09	1.00E-07	1.53E-07	0.00E+00	0.00E+00
0.100	3.06E-10	4.87E-09	1.00E-07	9.29E-08	0.00E+00	0.00E+00
0.125	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.150	1.96E-10	2.81E-09	1.00E-07	1.56E-07	0.00E+00	0.00E+00
0.175	2.16E-10	4.87E-09	1.00E-07	6.40E-08	0.00E+00	0.00E+00
0.200	5.59E-11	1.99E-09	1.00E-07	8.95E-08	0.00E+00	0.00E+00

Table 8-13 includes the time-dependent error results for ISCDP1. As for TSDP1, the error was small throughout the transient. At time 0.100 s and 0.125 s, the cell-wise powers were identical for ISCDP1 and RBDC1. Nonetheless, the results indicate that the impact of neglecting the cross section time derivative for ISDP1 was limited; TSDP1 was nearly as accurate as ISCDP1.

Table 8-13. Time-dependent error in the relative cell-wise power for ISCDP1

Time (s)	$\epsilon_{X,\text{avg}}$	$\epsilon_{X,\text{L2}}$	$\epsilon_{X,\text{maxA}}$	$\epsilon_{X,\text{maxR}}$	$\epsilon_{X,\text{peakA}}$	$\epsilon_{X,\text{peakR}}$
0.025	2.19E-08	3.02E-08	1.00E-07	1.51E-06	0.00E+00	0.00E+00
0.050	3.04E-09	9.94E-09	1.00E-07	1.23E-06	0.00E+00	0.00E+00
0.075	1.92E-10	2.81E-09	1.00E-07	1.53E-07	0.00E+00	0.00E+00
0.100	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.125	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.150	1.96E-10	2.81E-09	1.00E-07	1.56E-07	0.00E+00	0.00E+00
0.175	8.00E-11	2.81E-09	1.00E-07	6.40E-08	0.00E+00	0.00E+00
0.200	5.59E-11	1.99E-09	1.00E-07	8.95E-08	0.00E+00	0.00E+00

We also calculated the spatial error for IBDC1, which is provided in Table 8-14. The error is substantially higher than for the SDP methods, but the error is not as high as IBDC1 for the TWIGL ramp. This is consistent with our observations for the error in the final power, and it indicates that the TWIGL step transient was more forgiving for the isotropic approximation.

Table 8-14. Time-dependent error in the relative cell-wise power for IBDC1

Time (s)	$\epsilon_{X,\text{avg}}$	$\epsilon_{X,\text{L2}}$	$\epsilon_{X,\text{maxA}}$	$\epsilon_{X,\text{maxR}}$	$\epsilon_{X,\text{peakA}}$	$\epsilon_{X,\text{peakR}}$
0.025	1.67E-06	1.21E-06	5.40E-06	1.29E-05	3.40E-06	1.32E-06
0.050	2.04E-07	1.53E-07	6.00E-07	4.30E-06	4.00E-07	1.55E-07
0.075	5.60E-08	4.63E-08	2.00E-07	3.39E-06	0.00E+00	0.00E+00
0.100	2.79E-08	3.27E-08	1.00E-07	2.80E-06	0.00E+00	0.00E+00
0.125	2.12E-08	3.04E-08	1.00E-07	1.97E-06	1.00E-07	3.89E-08
0.150	3.49E-08	3.09E-08	1.00E-07	9.14E-06	0.00E+00	0.00E+00
0.175	2.39E-08	3.12E-08	1.00E-07	3.79E-06	0.00E+00	0.00E+00
0.200	2.46E-08	3.13E-08	1.00E-07	1.77E-06	1.00E-07	3.89E-08

We observed similar trends for the high-order SDP methods when compared to RBDC methods of the same order. Table 8-15 provides the error results for TSDP6, which was representative of the high-order methods. As for the other SDP methods, the error TSDP6 was larger earlier in the transient and tended to decrease with time.

Table 8-15. Time-dependent error in the relative cell-wise power for TSDP6

Time (s)	$\epsilon_{X,\text{avg}}$	$\epsilon_{X,\text{L2}}$	$\epsilon_{X,\text{maxA}}$	$\epsilon_{X,\text{maxR}}$	$\epsilon_{X,\text{peakA}}$	$\epsilon_{X,\text{peakR}}$
0.025	1.10E-07	5.06E-08	1.00E-07	3.40E-06	-1.00E-07	-3.89E-08
0.050	7.87E-09	1.98E-08	1.00E-07	6.73E-07	1.00E-07	3.89E-08
0.075	2.15E-09	1.01E-08	1.00E-07	4.66E-07	0.00E+00	0.00E+00
0.100	1.81E-10	4.45E-09	1.00E-07	6.68E-08	0.00E+00	0.00E+00
0.125	2.62E-10	4.87E-09	1.00E-07	8.18E-08	0.00E+00	0.00E+00
0.150	5.88E-11	2.81E-09	1.00E-07	4.70E-08	0.00E+00	0.00E+00
0.175	2.16E-10	3.98E-09	1.00E-07	9.22E-08	0.00E+00	0.00E+00
0.200	6.40E-10	6.29E-09	1.00E-07	1.76E-07	0.00E+00	0.00E+00

8.2.5 Summary for TWIGL Transients

The SDP methods accurately reproduced the RBDC solutions for the TWIGL transients. The SDP methods converged to the reference solution as the time step decreased and the order increased. This is a basic requirement for the SDP methods to be effective for solving time-dependent neutron transport.

The high-order SDP methods also accurately reproduced the RBDC methods when the same order and time step size was employed. This indicates that the angular flux time derivative propagation (TDP) approach using SDP was able to accurately reproduce the angular flux time derivative. The SDP methods were also more accurate than the IBDC method, although IBDC was nonetheless reasonably accurate. This is consistent with the error analysis in Chapter 6, and it suggests that all of the error terms for the SDP methods were smaller than the error terms for IBDC.

The SDP methods were computationally and memory efficient compared to the RBDC methods. The RBDC methods required more computing resources and substantially more memory than the SDP methods. The SDP methods did require more memory and run-time than the IBDC methods, although the difference was modest and the SDP methods were more accurate.

8.3 C5G7 Benchmark Transient Problem

8.3.1 C5G7 Benchmark Transient Problem Specification

The C5G7 benchmark problem [Lew01] was defined to assess the accuracy of homogenization approaches to neutron transport. Many neutron transport codes homogenize spatial regions (e.g. fuel pins and their surrounding moderator) so that the geometry can be represented using regular Cartesian cells, and previous benchmark neutron transport problems often simply provided homogenized cross sections. Instead, the C5G7 problem specifies cross sections for the fuel and moderator separately, which allows users to test the accuracy of homogenization techniques.

One of the advantages of MOC is that it can treat complex geometries without homogenization. Thus, compared to the TWIGL model, the C5G7 problem has a more realistic geometry. In addition, the C5G7 problem contains both UO_2 and MOX fuel, which have different properties. As a result, the C5G7 problem is expected to provide more spatial and angular heterogeneity than the TWIGL problem and thus be more challenging for approximations to the angular flux time derivative.

The C5G7 problem is a small reactor core with sixteen fuel assemblies: eight UO_2 assemblies and eight MOX assemblies. The assemblies are 21.42 cm square. The reactor is surrounded by a water reflector 21.42 cm thick. Figure 8-24 shows the south-east quadrant of the C5G7 reactor.

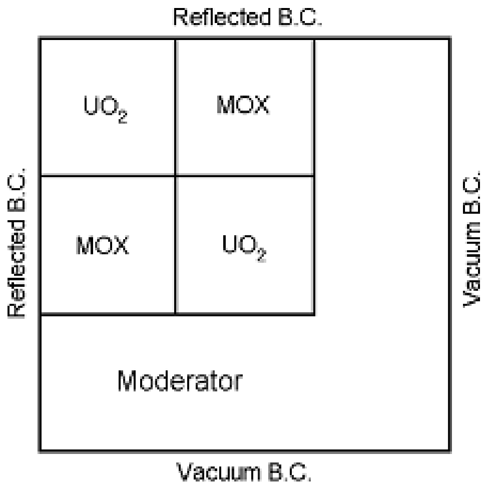


Figure 8-24. Layout of south-east quadrant of C5G7 benchmark by assembly [NEA03]

Each of the assemblies is composed of 17×17 fuel pins or guide tubes which are surrounded by moderator. Each pin cell is 1.26 cm square. The fuel pins have a radius of 0.54 cm and are centered in the pin cell. The layout of the assemblies for the south-east quadrant is depicted in Figure 8-25. The pin cell geometry is displayed in Figure 8-26.

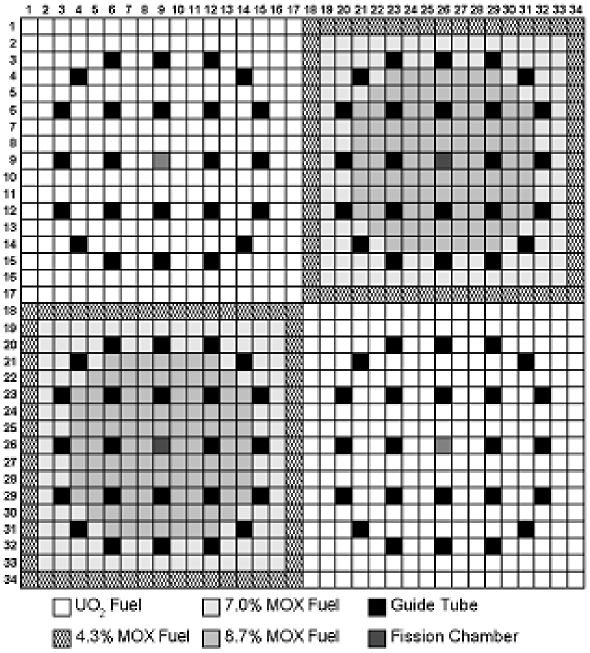


Figure 8-25. C5G7 fuel assembly layout by pin cell [NEA03]

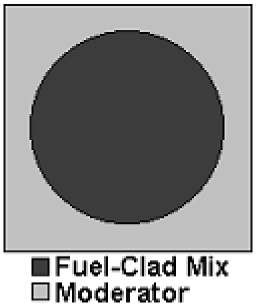


Figure 8-26. C5G7 pin cell layout [NEA03]

The C5G7 problem uses a seven-group cross section library. The seven-group library was defined using the collision probability method (CPM) computer code WIMS-AECL based on a 69-group library. The seven-group library was selected by the designers of C5G7 because it was identified as the most challenging group structure investigated. The cross sections for the C5G7 transient are provided in Appendix D.

While the C5G7 problem defines the cross sections for the steady-state problem, it does not specify conditions for a transient problem. In order to define a transient problem, we have incorporated control rods into the model, and we will eject control rods in one assembly to initiate a transient. Because C5G7 is a very small reactor, ejecting control rods with realistic cross sections would result in too great of a reactivity insertion. As a compromise, the control rod cross sections were adjusted such that a single control rod drive (CRD) would have approximately 1$ reactivity [Tsu13]. In addition, although it would typically require 0.1 s for the full ejection of the control rods, we are not able to accurately reflect partially-inserted control rods using a 2D DeCART model; thus we approximate the ejection as a step change in material composition.

We placed the control rods in the central UO_2 assemblies. Although C5G7 is typically modeled with quarter-core symmetry, this would be equivalent to ejecting four CRD at once, which introduces an unrealistically-large reactivity. Instead, we model the C5G7 transient with using the full core geometry, and we eject the control rods from only one assembly. This full core geometry is illustrated in Figure 8-27.

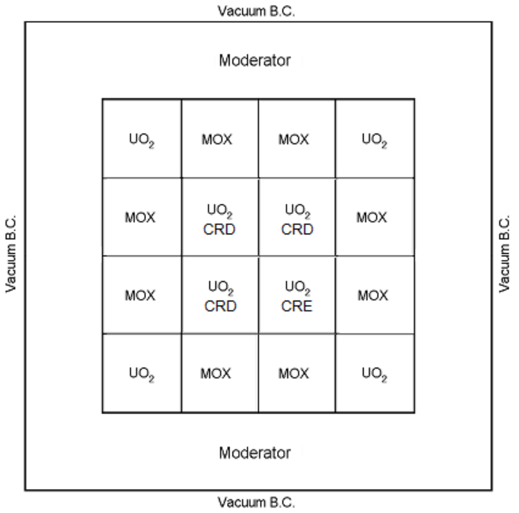


Figure 8-27. C5G7 with control rod drives (CRD) & ejected CRD (CRE) [NEA03]

We modeled the C5G7 reactor using a 12 by 12 grid of assemblies: 8 MOX assemblies, 4 UO_2 assemblies, 4 UO_2 assemblies with CRD, and 20 moderator "assemblies" that surround the reactor. Each assembly consists of 17 by 17 pin cells. In the outer moderator zone, the cells were entirely composed of moderator. In the fuel assemblies, the cells were composed of two materials: moderator outside of a radius of 0.54 cm, and either fuel, fission chamber, guide tube, or control rod inside the radius. The cell was divided into 32 regions divided evenly between the outer moderator and inner non-moderator material. The spatial discretization of the pin cell is displayed in Figure 8-28. The corresponding radii for Figure 8-28 are: r1 = 0.27 cm, r2 = 0.54 cm, and r3 = 0.60 cm. The pin pitch is 1.26 cm. For MOC we used a ray spacing of 0.04 cm, and we used 8 azimuthal angles and 3 polar angles in 90°.

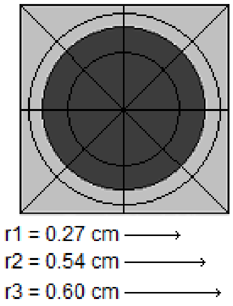


Figure 8-28. Spatial discretization of the pin cell into regions [adapted from NEA03]

The C5G7 benchmark problem also does not provide kinetics parameters or delayed neutron data. We used representative kinetics parameters for a pressurized water reactor [Tsu13]. The delayed neutron spectrum was assumed to be equal to the prompt neutron fission spectrum. The neutron energy group structure and velocities are provided in Table 8-16, while the delayed neutron precursor group data is provided in Table 8-17.

Table 8-16. Neutron energy group structure for C5G7 benchmark

Energy Group	**1**	**2**	**3**	**4**	**5**	**6**	**7**
$\boldsymbol{E}_{\mathbf{top}}$	20 MeV	1 MeV	500 keV	3 eV	0.625 eV	0.1 eV	0.02 eV
$\boldsymbol{E}_{\mathbf{bottom}}$	1 MeV	500 keV	3 eV	0.625 eV	0.1 eV	0.02 eV	10 μeV
$\boldsymbol{v}_{\mathbf{midpoint}}$ **(cm/s)**	4.48×10^{9}	1.20×10^{9}	6.92×10^{8}	1.86×10^{6}	8.33×10^{5}	3.39×10^{5}	1.85×10^{5}

Table 8-17. Delayed neutron group parameters for C5G7 benchmark

Delayed Group	**1**	**2**	**3**	**4**	**5**	**6**
β	2.470×10^{-4}	1.3845×10^{-3}	1.222×10^{-3}	2.6455×10^{-3}	8.320×10^{-4}	1.690×10^{-4}
λ (s^{-1})	0.0127	0.0317	0.115	0.311	1.40	3.87

8.3.2 Steady-state Eigenvalue Solution

The C5G7 transient begins from an assumed steady-state critical condition, which is initialized using an eigenvalue problem. We calculated the eigenvalue to be $k_{\text{eff}} = 1.148013$. The power distribution by fuel assembly for the C5G7 reactor at steady-state is depicted in Table 8-18. This table is color-coded by the magnitude of the power. Figure 8-29 provides the steady-state power distribution by fuel cell for the south-east quadrant of the reactor.

Table 8-18. Steady-state relative power distribution by assembly

0.617	0.861	0.861	0.617
0.861	1.66	1.66	0.861
0.861	1.66	1.66	0.861
0.617	0.861	0.861	0.617

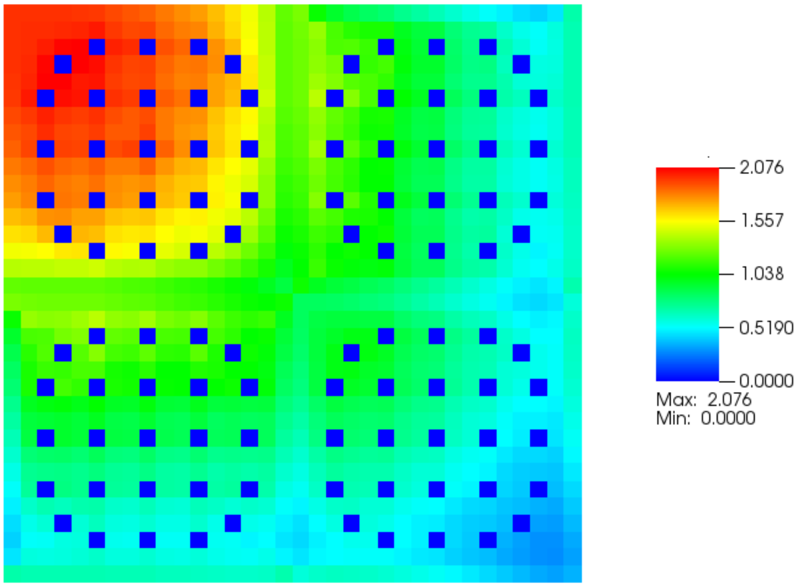


Figure 8-29. Steady-state power distribution by pin cell in south-east quadrant

8.3.3 Control Rod Ejection Transient

The C5G7 transient was driven by replacing the control rod material in the guide tubes of the south-east, central UO_2 assembly with the moderator-filled guide tube material. This resulted in the introduction of about 1.25$ of reactivity which causes a fast exponential increase in power. The transient was simulated for 0.05 s, over which period the power increases by two orders of magnitude from the initial power of one.

We generated a reference solution using RBDC1 with a 0.01 ms time step, which required about 3.5 days using eight 2.53 GHz Intel Xeon processors and approximately 30 GB of RAM to store the angular flux. A power trace for the transient is presented in Figure 8-30, with a final power of 126.030.

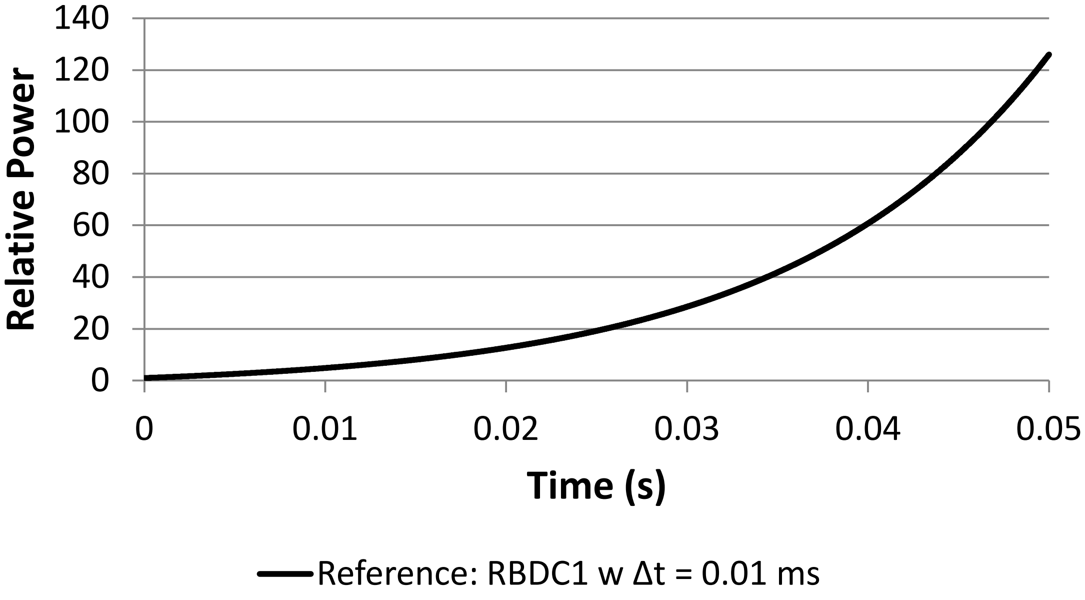


Figure 8-30. Relative core power for C5G7 transient reference solution

In the following subsections we used the reference solution to assess the accuracy and efficiency of the SDP methods. The first subsection focuses on the ability of SDP methods to accurately integrate the total power, while the second subsection assesses how well the SDP methods to replicate the RBDC methods of the same order and time step. The third subsection evaluates the ability of SDP to capture the relative pin power distribution. The fourth subsection evaluates the scalar flux solution at the sub-pin level. Finally, the fifth subsection compares the computational and memory resources required for each method.

8.3.3.1 SDP Convergence to the Reference Solution as Time Step Decreases

In this subsection we evaluated the convergence of the SDP methods to the reference solution as the time step size decreases and the order of the source time derivative approximation increases. As for the TWIGL transients, we used the final power for the core as a metric for the accuracy of the methods. This was used to calculate the relative error in the final power as defined in Equation 8.1.

Figure 8-31 is a plot of the relative error in the final power for TSPD1 as a function of time step size on a log-log graph. The C5G7 transient was amenable to a narrower range of time step sizes than the TWIGL transients, and the error was much larger for a given time step size due to the larger time derivatives. We calculate the order of convergence using Equation 8.2 to be 1.091, indicating linear convergence as expected. Similar trends were observed for ISDP1 and ISCDP1.

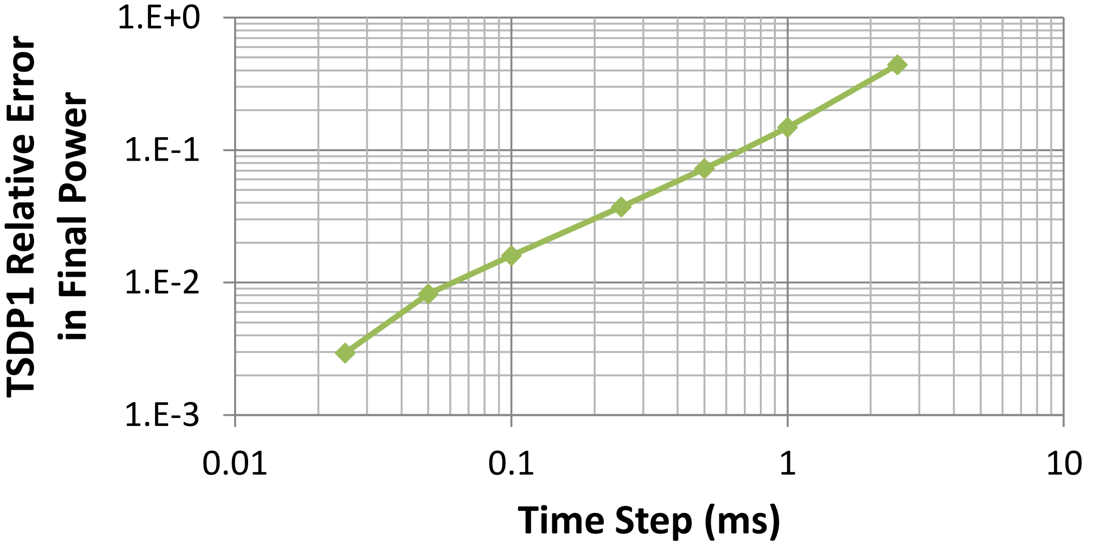


Figure 8-31. Relative error in the final power for TSDP1

Next we evaluated the accuracy of the high-order SDP methods as the order increases and the time step size decreases. The relative error in the final power for the TSDP methods is presented in Figure 8-32. As expected, as the step size decreases, the high-order TSDP methods become increasingly accurate. However, much like the TWIGL step change transient, for large time steps the error slope for the high-order methods closely tracks the TSDP2 results. This is because the control rod ejection is approximated using a step change, and by the time several previous solutions are available to approximate the time derivatives to high-order accuracy, significant error form low order methods is incurred. This is a limitation for all BDF methods including RBDC. For more realistic transients, control rod ejections are not step changes, and the high-order methods will better capture this element of the transient.

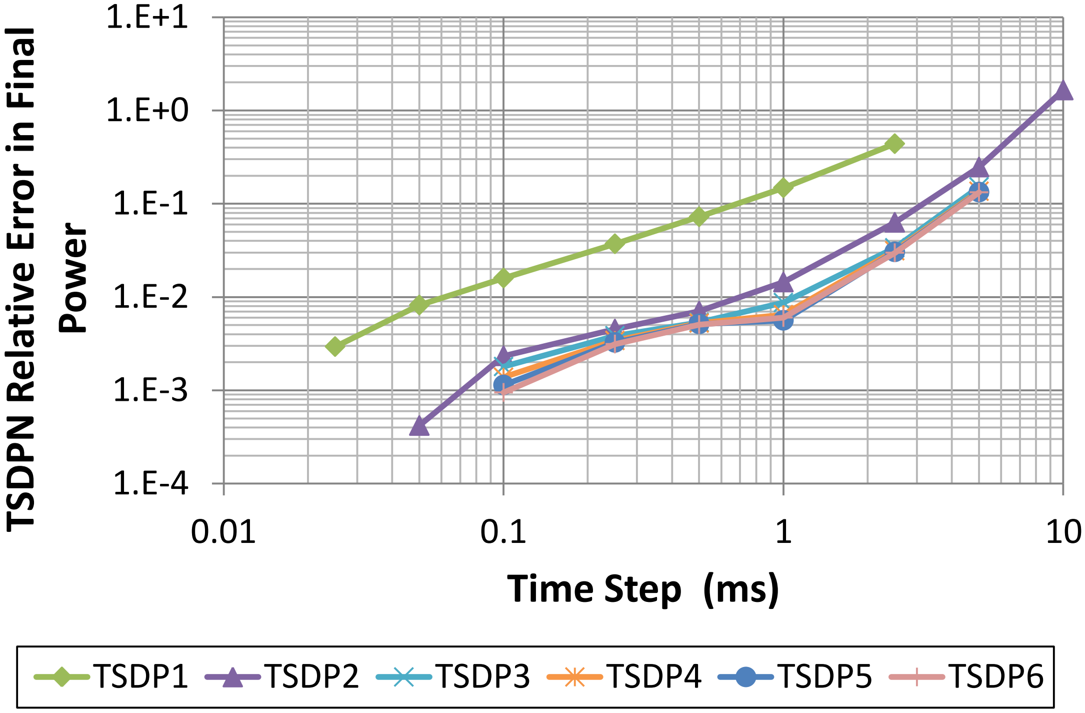


Figure 8-32. Relative error in final power for TSDPN as a function of time step size

The order of convergence for the high-order methods is calculated using Equation 8.2, and the results are summarized in Table 8-19. This demonstrates that the order of convergence to the reference solution is limited for the high-order methods. Similar trends were observed for ISDP1 and ISCDP1, and there error plots are visually indistinguishable from Figure 8-32.

Table 8-19. Logarithmic error slopes for TSDP methods

Method	**TSDP1**	**TSDP2**	**TSDP3**	**TSDP4**	**TSDP5**	**TSDP6**
Slope	1.091	1.599	1.111	1.151	1.181	1.223

8.3.3.2 SDPN Comparison to RBDCN for Same Time Step Size

In this subsection we assessed whether the SDPN methods are in good agreement with the RBDCN methods of the same order and time step size. Again, we used the relative error in the final power at the end of the transient as a metric. However, instead of comparing to the fine time step reference solution, we compared the SDP methods to the RBDC method of the same order and time step size. However, because of the high memory cost of storing the angular fluxes for the C5G7 transient, we were only able to generate RBDC methods of a maximum of order three, which required approximately 60 GB of RAM. The relative errors for the RBDC methods are compared to TSDP in Figure 8-33. We see that while the error is comparable for the RBDC methods and the TSDP methods, the agreement is not as close as it was for the TWIGL transients.

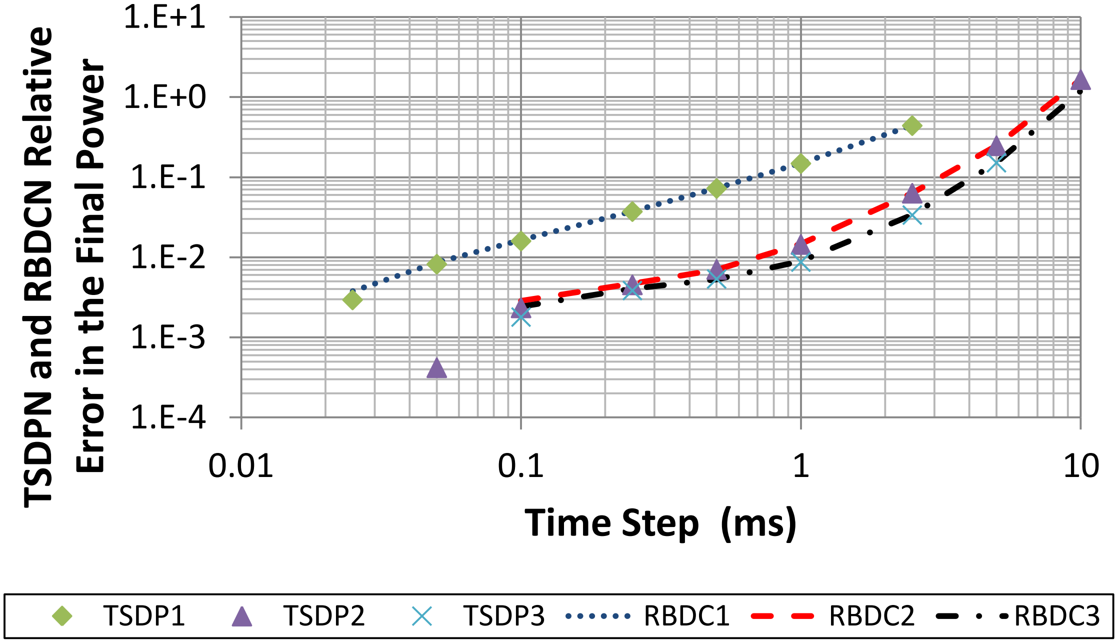


Figure 8-33. Error in the final power for TSDPN and RBDCN as a function of Δt

We compared the RBDCN solutions directly to TSDPN using the absolute difference in final power as calculated with Equation 8.5. These results are presented in Figure 8-34 for TSDPN alongside the results IBDC1. IBDC1 uses a dashed line for emphasis. Here we see that the TSDP and IBDC methods offer comparable agreement to RBDC, in spite of the isotropic approximation for the angular flux time derivative in the latter. However, we see in the following sections that the SDP methods do a better job of representing the spatial solution and are more accurate than IBDC.

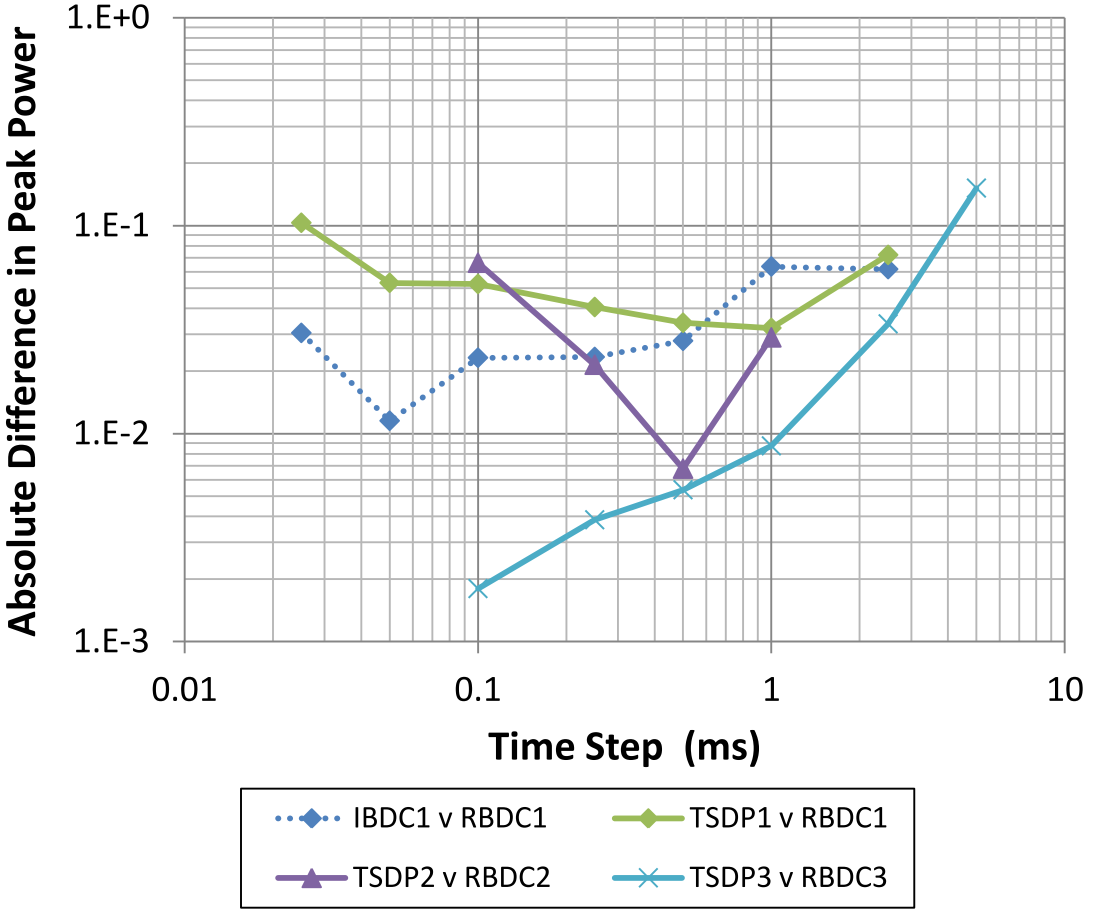


Figure 8-34. The absolute difference in peak power for TSDP/IBDC and RBDC with the same order and time step size

We also compared the absolute difference in power for the ISDPN methods with respect to the RBDCN methods of the same time step and order. These results are presented in Figure 8-35. Here we see a systematic improvement for the ISDP methods over the TSDP methods, with all of the ISDP methods outperforming IBDC1, albeit by a narrow margin. This suggests that the incorporation of the second time derivative has a small but positive impact on the accuracy of ISDP.

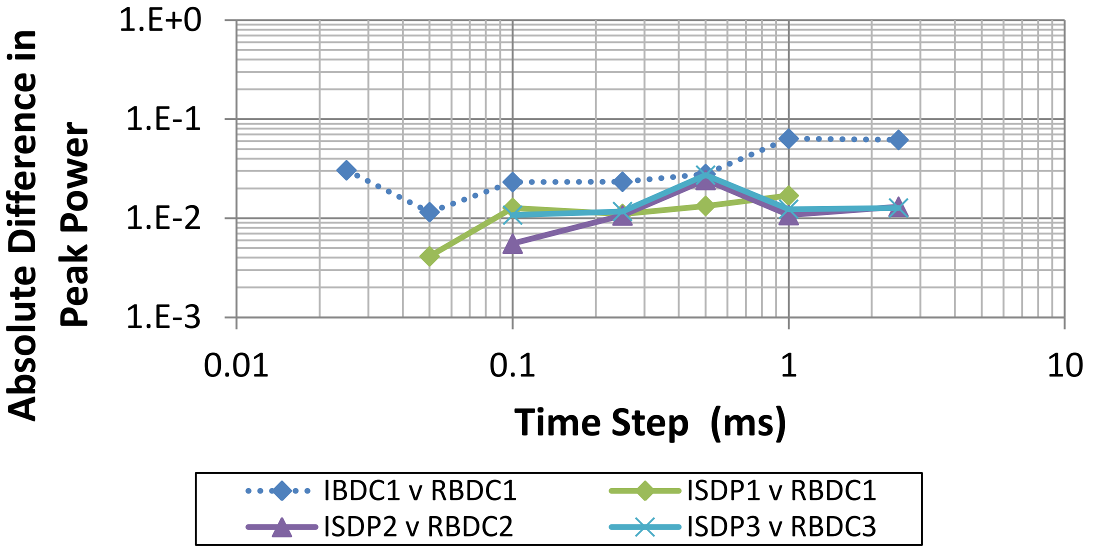


Figure 8-35. The absolute difference in peak power for ISDP/IBDC and RBDC with the same order and time step size

We also compared the absolute difference in peak power for ISCDPN with RBDCN with the same order and time step size. The result is similar to the TSDPN and ISDPN methods, with the results being comparable in magnitude to IBDC.

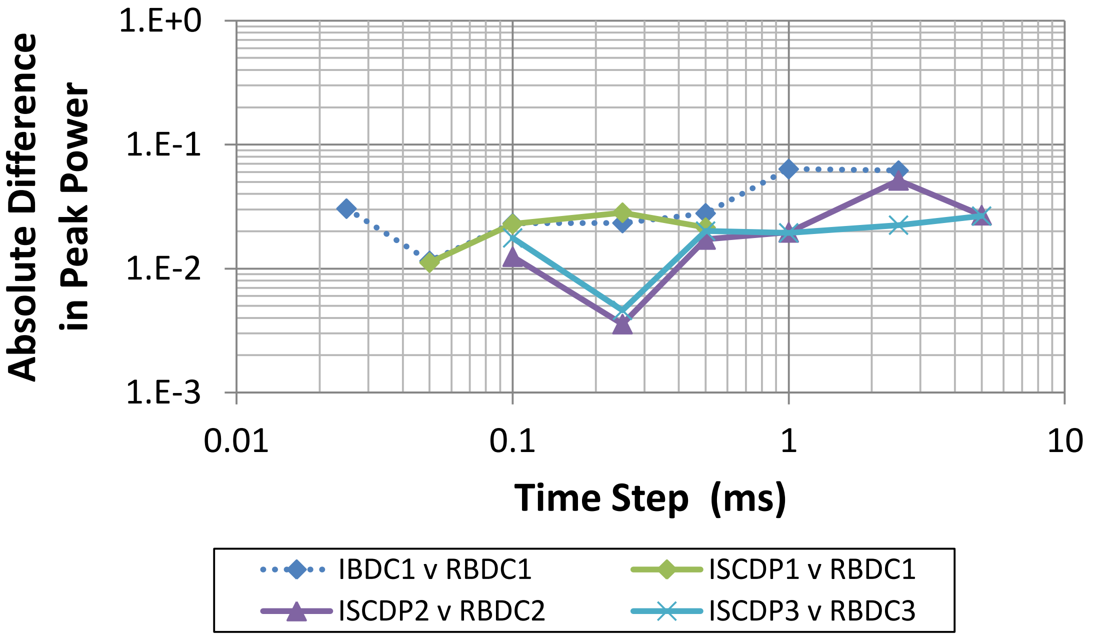


Figure 8-36. The absolute difference in peak power for ISCDP/IBDC and RBDC with the same order and time step size

8.3.3.3 Comparison of SDP and RBDC Spatial Power Distribution

Next we evaluated the spatial accuracy of the SDP methods by comparing the relative pin powers for each cell. We used the results for the C5G7 transient with a 1 ms time step, comparing the average relative error, the L2 error, the maximum absolute error, the maximum relative error, the absolute error for the peak pin, and the relative error for the peak pin. These errors are calculated using Equations 8.6—8.12. For the C5G7 transient, pin cells with no fission (e.g. in the moderator of guide tubes) are not included in the totals for the average and L2 errors because they produce no power.

The time-dependent errors for TSDP1 are displayed in Table 8-20. Here we see that relative pin powers for TSDP1 were in excellent agreement with those of RBDC1. This is in contrast to the observation of the previous subsection, where the agreement in the total power was at the fourth significant figure. This suggests that the TSDP1 is accurately capturing the angular and spatial distribution of the angular flux time derivative, but the magnitude of the time derivative is overestimated. This could be explained by the truncation of the second time derivative of the angular flux.

Table 8-20. Time-dependent error in the relative cell-wise power for TSDP1

Time (s)	$\epsilon_{X,\mathrm{avg}}$	$\epsilon_{X,\mathrm{L2}}$	$\epsilon_{X,\mathrm{maxA}}$	$\epsilon_{X,\mathrm{maxR}}$	$\epsilon_{X,\mathrm{peakA}}$	$\epsilon_{X,\mathrm{peakR}}$
0.005	9.33E-07	1.06E-06	4.00E-06	2.81E-06	-3.90E-06	-1.59E-06
0.010	6.93E-07	7.67E-07	2.90E-06	2.29E-06	-2.90E-06	-1.18E-06
0.015	5.77E-07	6.33E-07	2.40E-06	1.86E-06	-2.30E-06	-9.35E-07
0.020	5.12E-07	5.59E-07	2.00E-06	1.73E-06	-2.00E-06	-8.12E-07
0.025	4.79E-07	5.18E-07	1.90E-06	1.65E-06	-1.90E-06	-7.71E-07
0.030	4.56E-07	4.91E-07	1.80E-06	1.52E-06	-1.80E-06	-7.31E-07
0.035	4.41E-07	4.72E-07	1.70E-06	1.44E-06	-1.70E-06	-6.90E-07
0.040	4.30E-07	4.60E-07	1.70E-06	1.46E-06	-1.70E-06	-6.90E-07
0.045	4.25E-07	4.53E-07	1.60E-06	1.40E-06	-1.60E-06	-6.49E-07
0.050	4.16E-07	4.45E-07	1.60E-06	1.41E-06	-1.50E-06	-6.09E-07

To illustrate where the error occurs for TSDP1, in Figure 8-37 we plotted the relative error in the relative pin power at the end of the transient. We see that the largest relative error is at the edge of the reactor at the interface of the MOX and moderator assemblies; however these are low power pins, and the absolute error is modest. The largest absolute error occurs near the northwest-most ejected control rod, which is in the southeast quadrant of the core. The power in the rodded assemblies is slightly under predicted, including the assembly with the ejected control rods.

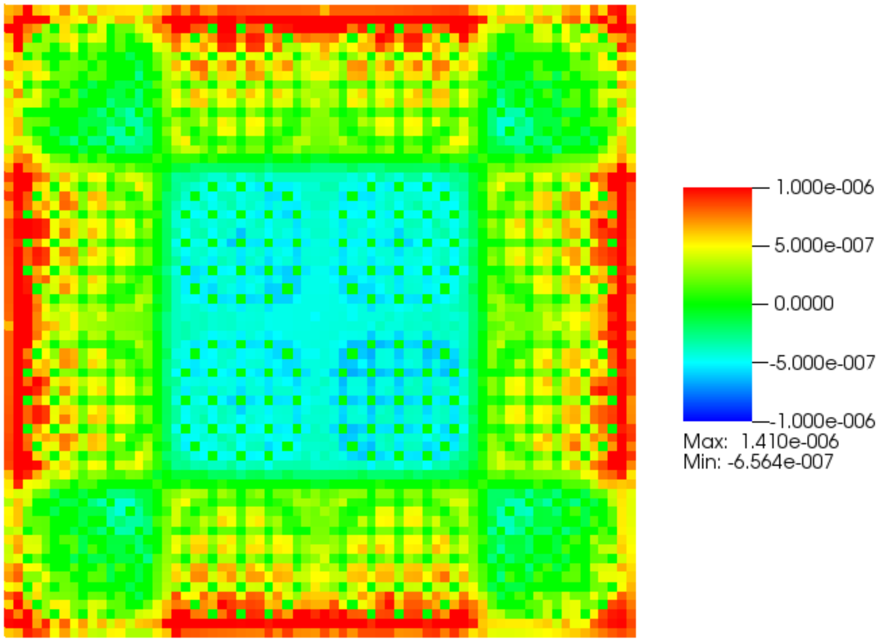


Figure 8-37. Relative error in the relative pin power distribution for TSDP1 at 0.05 s

The time-dependent errors for ISDP1 are presented in Table 8-21. We see that the errors in the relative pin power are generally slightly lower for ISDP1 than TSDP1. When this is coupled with ISDP1's more accurate prediction of the total power, the result is that ISDP1 systematically better predicts the absolute pin powers than TSDP1.

Table 8-21. Time-dependent error in the relative cell-wise power for ISDP1

Time (s)	$\epsilon_{X,\mathbf{avg}}$	$\epsilon_{X,\mathbf{L2}}$	$\epsilon_{X,\mathbf{maxA}}$	$\epsilon_{X,\mathbf{maxR}}$	$\epsilon_{X,\mathbf{peakA}}$	$\epsilon_{X,\mathbf{peakR}}$
0.005	6.27E-07	5.60E-07	1.70E-06	3.66E-06	-1.30E-06	-5.30E-07
0.010	4.70E-07	4.44E-07	1.50E-06	2.32E-06	-1.50E-06	-6.10E-07
0.015	4.08E-07	3.95E-07	1.40E-06	2.00E-06	-1.40E-06	-5.69E-07
0.020	3.74E-07	3.66E-07	1.30E-06	1.65E-06	-1.30E-06	-5.28E-07
0.025	3.56E-07	3.49E-07	1.30E-06	1.62E-06	-1.20E-06	-4.87E-07
0.030	3.42E-07	3.35E-07	1.30E-06	1.65E-06	-1.20E-06	-4.87E-07
0.035	3.30E-07	3.25E-07	1.20E-06	1.60E-06	-1.20E-06	-4.87E-07
0.040	3.28E-07	3.21E-07	1.20E-06	1.60E-06	-1.20E-06	-4.87E-07
0.045	3.23E-07	3.18E-07	1.20E-06	1.60E-06	-1.10E-06	-4.46E-07
0.050	3.17E-07	3.12E-07	1.10E-06	1.60E-06	-1.10E-06	-4.46E-07

Figure 8-38 shows the spatially-dependent relative error in the relative pin power for ISDP1 at the end of the transient. The error distribution is similar to TSDP1, but the error is smaller near the core periphery and in the interior MOX assemblies. In general the error is more uniform for ISDP1 than TSDP1.

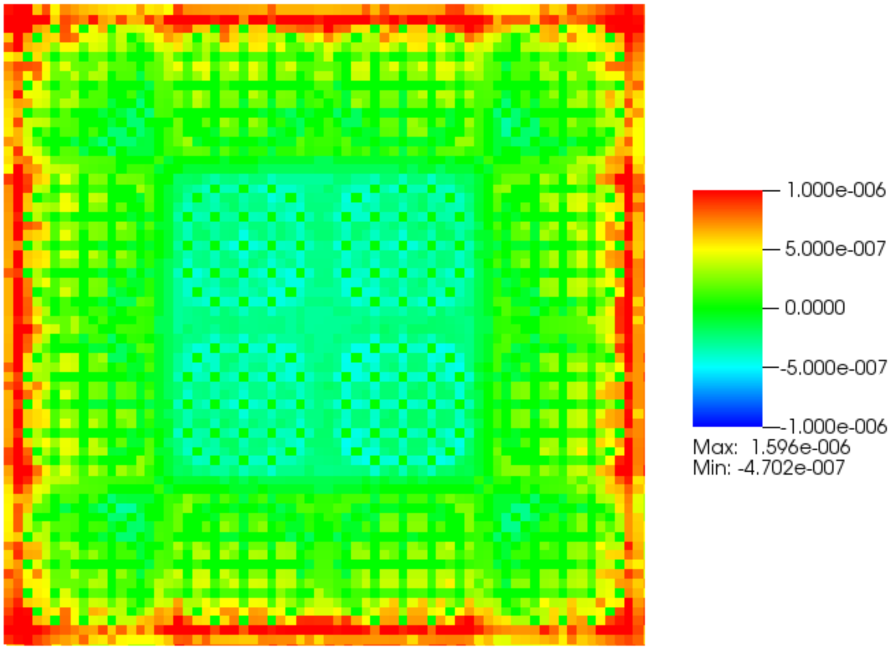


Figure 8-38. Relative error in the relative pin power distribution for ISDP1 at 0.05 s

Table 8-22 presents the time-dependent errors for ISCDP1. The spatial error for ISCDP1 is small and comparable to ISDP1. In addition, the spatially-dependent relative error in the pin power for ISCDP1 is essentially indistinguishable from the that of ISDP1 presented in Figure 8-38. This confirms that the error resulting from the neglect of the cross section derivative is small for this problem.

Table 8-22. Time-dependent error in the relative cell-wise power for ISCDP1

Time (s)	$\epsilon_{X,\mathbf{avg}}$	$\epsilon_{X,\mathbf{L2}}$	$\epsilon_{X,\mathbf{maxA}}$	$\epsilon_{X,\mathbf{maxR}}$	$\epsilon_{X,\mathbf{peakA}}$	$\epsilon_{X,\mathbf{peakR}}$
0.005	6.53E-07	6.55E-07	2.50E-06	2.80E-06	-2.40E-06	-9.78E-07
0.010	4.98E-07	5.00E-07	1.90E-06	2.01E-06	-1.90E-06	-7.73E-07
0.015	4.25E-07	4.26E-07	1.60E-06	1.73E-06	-1.60E-06	-6.50E-07
0.020	3.83E-07	3.83E-07	1.40E-06	1.65E-06	-1.40E-06	-5.69E-07
0.025	3.61E-07	3.60E-07	1.40E-06	1.60E-06	-1.30E-06	-5.28E-07
0.030	3.45E-07	3.42E-07	1.30E-06	1.60E-06	-1.30E-06	-5.28E-07
0.035	3.34E-07	3.30E-07	1.20E-06	1.60E-06	-1.20E-06	-4.87E-07
0.040	3.30E-07	3.24E-07	1.20E-06	1.60E-06	-1.20E-06	-4.87E-07
0.045	3.25E-07	3.20E-07	1.20E-06	1.60E-06	-1.10E-06	-4.46E-07
0.050	3.18E-07	3.14E-07	1.10E-06	1.60E-06	-1.10E-06	-4.46E-07

Table 8-23 presents the time-dependent error for IBDC1. Although IBDC1 offered similar accuracy to the SDP methods for the total core power, the spatial error in the cell-wise relative pin power is about two order of magnitude greater for IBDC1 than the SDP methods. This is true throughout the transient in essentially every error category, although the error for IBDC1 is nonetheless very small.

Table 8-23. Time-dependent error in the relative cell-wise power for IBDC1

Time (s)	$\epsilon_{X,\mathrm{avg}}$	$\epsilon_{X,\mathrm{L2}}$	$\epsilon_{X,\mathrm{maxA}}$	$\epsilon_{X,\mathrm{maxR}}$	$\epsilon_{X,\mathrm{peakA}}$	$\epsilon_{X,\mathrm{peakR}}$
0.005	4.42E-05	4.43E-05	1.92E-04	2.02E-04	-1.92E-04	-7.83E-05
0.010	3.35E-05	3.37E-05	1.46E-04	1.53E-04	-1.46E-04	-5.94E-05
0.015	2.88E-05	2.89E-05	1.25E-04	1.31E-04	-1.25E-04	-5.09E-05
0.020	2.62E-05	2.63E-05	1.14E-04	1.20E-04	-1.14E-04	-4.64E-05
0.025	2.49E-05	2.52E-05	1.10E-04	1.12E-04	-1.10E-04	-4.47E-05
0.030	2.39E-05	2.41E-05	1.05E-04	1.08E-04	-1.05E-04	-4.28E-05
0.035	2.33E-05	2.35E-05	1.02E-04	1.05E-04	-1.02E-04	-4.15E-05
0.040	2.28E-05	2.30E-05	1.00E-04	1.03E-04	-1.00E-04	-4.06E-05
0.045	2.25E-05	2.27E-05	9.85E-05	1.02E-04	-9.85E-05	-4.00E-05
0.050	2.23E-05	2.25E-05	9.78E-05	1.01E-04	-9.78E-05	-3.97E-05

The spatially-dependent relative error in the pin power for IBDC1 is presented in Figure 8-39. Note that the scale for this plot is an order of magnitude larger than the scale of Figure 8-37 and Figure 8-39; the SDP and IBDC results cannot be meaningfully compared when the scale is the same. Like the SDP methods, IBDC1 has the greatest

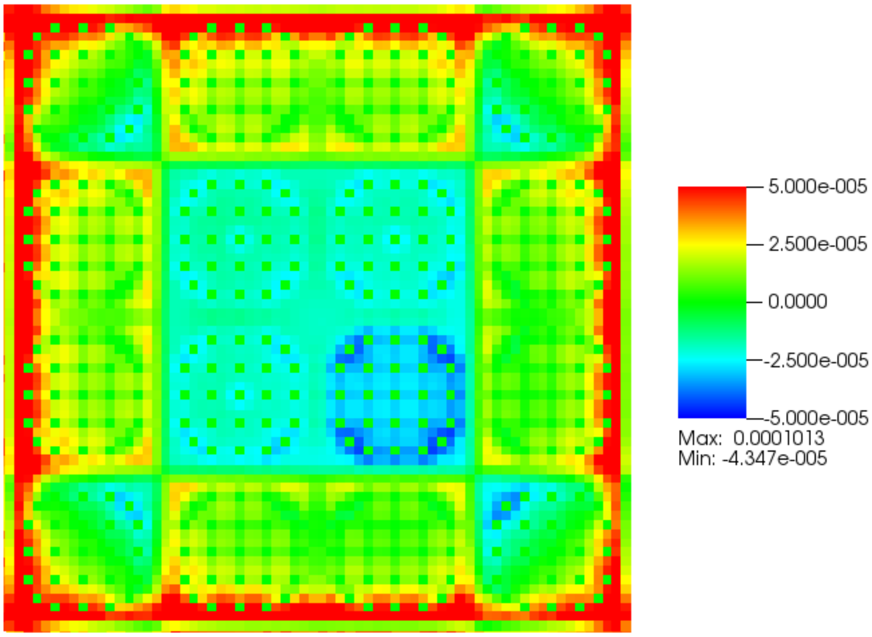


Figure 8-39. Relative error in the relative pin power distribution for IBDC1 at 0.05 s

relative error near the core periphery and within the rodded assemblies. However, for IBDC1 the error near the ejected rods is more than double that of the remaining control rods. By contrast, the error for the SDP methods near the ejected and remaining control rods was comparable.

8.3.3.4 Comparison of SDP and RBDC Sub-pin Scalar Flux Distribution

In this subsection we assessed how well the SDP methods replicated the scalar flux distribution with respect to the RBDC solution when the same order and time step size is used. As in the previous section, we used the results from the transient with a 1 ms time step. We assessed the accuracy using the same measures of error described by Equations 8.6—8.12, although the region-wise neutron scalar fluxes were used instead of the cell-wise relative pin powers. As a result, this comparison incorporates the results in the moderator regions and reflector, in contrast to the previous section. Although we assessed the error for all energy groups, we limit this evaluation to the thermal group (7), which was the most important group flux and is representative of the other groups.

The time-dependent error in the thermal group scalar flux for TSDP1 is presented in Table 8-24. Here we see that the error is similar in magnitude to the relative error in the total power at the end of the transient. This reconciles the results

Table 8-24. Time-dependent error in the thermal group scalar flux for TSDP1

Time (s)	$\epsilon_{X,\mathrm{avg}}$	$\epsilon_{X,\mathrm{L2}}$	$\epsilon_{X,\mathrm{maxA}}$	$\epsilon_{X,\mathrm{maxR}}$	$\epsilon_{X,\mathrm{peakA}}$	$\epsilon_{X,\mathrm{peakR}}$
0.005	1.08E-04	1.10E-04	4.56E-04	1.14E-04	4.56E-04	1.11E-04
0.010	1.09E-04	1.11E-04	8.78E-04	1.14E-04	8.78E-04	1.12E-04
0.015	1.18E-04	1.19E-04	1.58E-03	1.22E-04	1.58E-03	1.20E-04
0.020	1.29E-04	1.30E-04	2.74E-03	1.33E-04	2.74E-03	1.31E-04
0.025	1.42E-04	1.43E-04	4.62E-03	1.46E-04	4.62E-03	1.44E-04
0.030	1.56E-04	1.58E-04	7.62E-03	1.60E-04	7.62E-03	1.58E-04
0.035	1.71E-04	1.73E-04	1.24E-02	1.75E-04	1.24E-02	1.73E-04
0.040	1.87E-04	1.89E-04	1.99E-02	1.91E-04	1.99E-02	1.89E-04
0.045	2.04E-04	2.05E-04	3.16E-02	2.07E-04	3.16E-02	2.05E-04
0.050	2.21E-04	2.22E-04	4.98E-02	2.24E-04	4.98E-02	2.22E-04

from the previous two sections showing that while TSDP1 accurately captures the spatial distribution of the solution, there is more error in the magnitude of the solution. We also note that the error is evenly distributed throughout which is why the maximum relative error is comparable to the average error. The error is nonetheless small throughout the transient.

The time-dependent error in the thermal group scalar flux for ISDP1 is presented in Table 8-25. This result is also consistent with the results in the previous two sections, and it shows that the incorporation of the scalar flux second derivative improves the accuracy of the method. The error for ISCDP1 was similar to IDSP1, confirming that the cross section derivative had a small impact on the solution.

Table 8-25. Time-dependent error in the thermal group scalar flux for ISDP1

Time (s)	$\epsilon_{X,\mathrm{avg}}$	$\epsilon_{X,\mathrm{L2}}$	$\epsilon_{X,\mathrm{maxA}}$	$\epsilon_{X,\mathrm{maxR}}$	$\epsilon_{X,\mathrm{peakA}}$	$\epsilon_{X,\mathrm{peakR}}$
0.005	4.42E-05	4.43E-05	1.92E-04	2.02E-04	-1.92E-04	-7.83E-05
0.010	3.35E-05	3.37E-05	1.46E-04	1.53E-04	-1.46E-04	-5.94E-05
0.015	2.88E-05	2.89E-05	1.25E-04	1.31E-04	-1.25E-04	-5.09E-05
0.020	2.62E-05	2.63E-05	1.14E-04	1.20E-04	-1.14E-04	-4.64E-05
0.025	2.49E-05	2.52E-05	1.10E-04	1.12E-04	-1.10E-04	-4.47E-05
0.030	2.39E-05	2.41E-05	1.05E-04	1.08E-04	-1.05E-04	-4.28E-05
0.035	2.33E-05	2.35E-05	1.02E-04	1.05E-04	-1.02E-04	-4.15E-05
0.040	2.28E-05	2.30E-05	1.00E-04	1.03E-04	-1.00E-04	-4.06E-05
0.045	2.25E-05	2.27E-05	9.85E-05	1.02E-04	-9.85E-05	-4.00E-05
0.050	2.23E-05	2.25E-05	9.78E-05	1.01E-04	-9.78E-05	-3.97E-05

Finally we present the time-dependent error in the thermal group scalar flux for IBDC1 in Table 8-26. We see that although the error in the relative pin power for IBDC1 was much greater than the SDP methods, the error in the scalar flux is only slightly larger. This is because the error in the region-wise scalar flux is dominated by the error in the total magnitude of the solution. It is also noteworthy that the error for IBDC1 was more concentrated in the moderator regions than the SDP methods.

Table 8-26. Time-dependent error in the thermal group scalar flux for IBDC1

Time (s)	$\epsilon_{X,\mathrm{avg}}$	$\epsilon_{X,\mathrm{L2}}$	$\epsilon_{X,\mathrm{maxA}}$	$\epsilon_{X,\mathrm{maxR}}$	$\epsilon_{X,\mathrm{peakA}}$	$\epsilon_{X,\mathrm{peakR}}$
0.005	2.56E-04	1.69E-04	4.60E-04	6.22E-04	1.77E-04	4.30E-05
0.010	2.61E-04	1.94E-04	1.10E-03	5.40E-04	7.90E-04	1.01E-04
0.015	2.82E-04	2.23E-04	2.22E-03	5.20E-04	1.91E-03	1.45E-04
0.020	3.10E-04	2.56E-04	4.16E-03	5.26E-04	3.87E-03	1.85E-04
0.025	3.36E-04	2.84E-04	7.36E-03	5.43E-04	6.97E-03	2.17E-04
0.030	3.50E-04	3.00E-04	1.19E-02	5.48E-04	1.14E-02	2.36E-04
0.035	3.71E-04	3.22E-04	1.93E-02	5.63E-04	1.86E-02	2.60E-04
0.040	3.97E-04	3.49E-04	3.13E-02	5.85E-04	3.03E-02	2.88E-04
0.045	4.32E-04	3.85E-04	5.13E-02	6.18E-04	5.00E-02	3.25E-04
0.050	4.70E-04	4.23E-04	8.32E-02	6.54E-04	8.14E-02	3.63E-04

8.3.3.5 Comparison of Computational and Memory Requirements

There are substantial differences in the computational and memory requirements of the SDP and BDC methods. In this section we compared the run-time and memory required for the C5G7 transient.

Assessing the run-time for the C5G7 transient was more difficult than expected because there was a large amount of fluctuation run-time per time step for each method. This was result of using shared machines on the CAC Flux network, and the diversity of machines on that network. While reviewing the run-time for all methods with all time step sizes, it was observed that every time step size had a result that was uncharacteristic of the typical run-times. For example, Figure 8-40 shows the run-times for the C5G7 transient with a 1 ms time step. Here the results are typical except for the TSDP3 result, which was faster than IBDC1. The SDP and RBDC methods should never be faster than IBDC because IBDC has the minimum number of arithmetic operations.

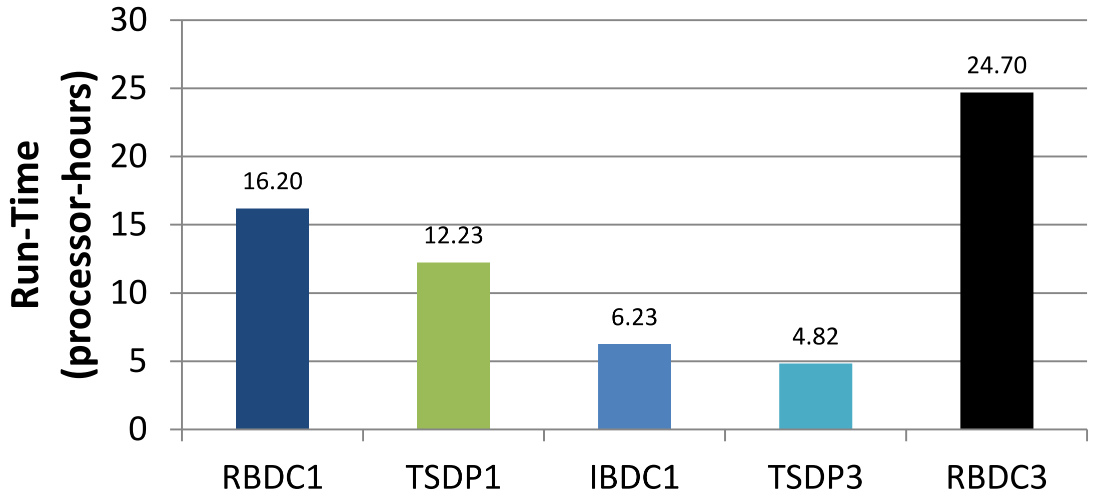


Figure 8-40. The run-time for the C5G7 transient with a 1 ms time step

To counteract this in an unbiased way, we calculated the average run-time per time step for all of the methods. Figure 8-41 displays this result, which is representative of the relative computational requirements for the various methods. As was the case for the TWIGL transient, the SDP methods were intermediate in computational expense to the RBDC1 and IBDC1 method. The high-order SDP methods had a similar computational expense to the first-order SDP methods.

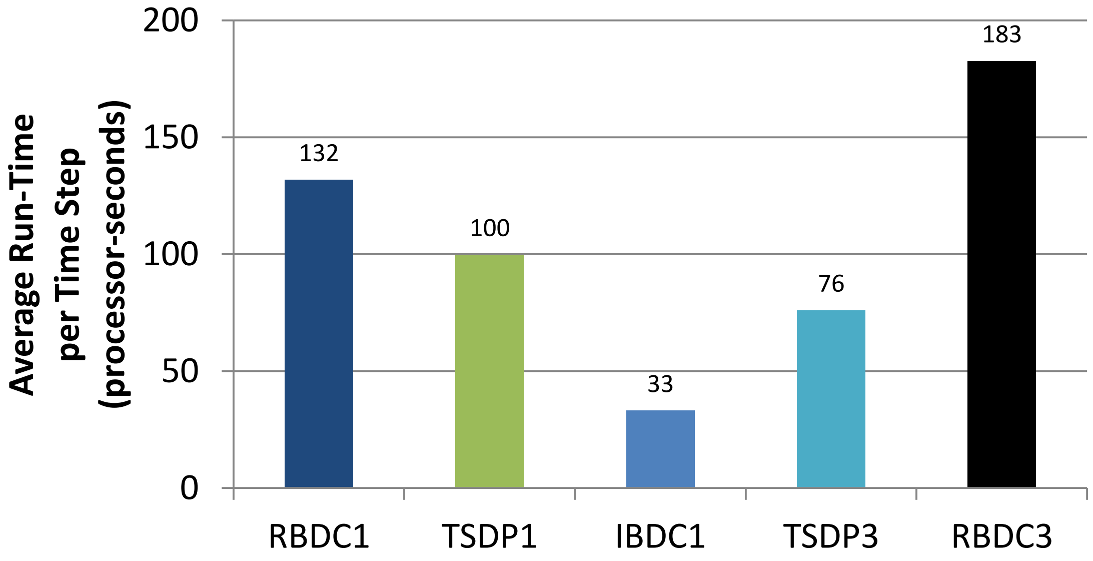


Figure 8-41. The average run-time per time step for all C5G7 transient results

The C5G7 transient with a 1 ms time step was also used to determine the memory required for the various methods. These results are summarized in Figure 8-42. Note that the figure uses a log scale because the memory requirements for the RBDC methods were two orders of magnitude larger than for the SDP and IBDC methods.

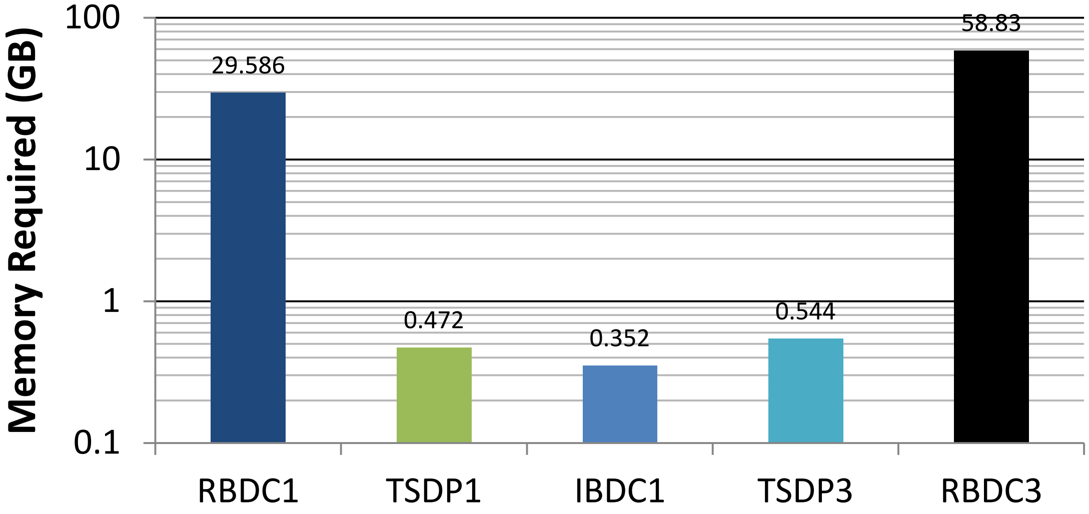


Figure 8-42. Memory requirements for the C5G7 transient

8.3.4 C5G7 Summary

The SDP methods accurately reproduced the RBDC solutions of the C5G7 transient. As the time steps were refined and the order of the method was increased, the SDP methods converged toward the reference solution.

When the SDP methods were compared to the RBDC methods with the same order and time step size, the agreement was good. This agreement included capturing the spatial distribution of the power. This indicates that the SDP methods effectively captured the angular flux time derivative throughout the space and time domain of the transient. In addition, the value of incorporating the isotropic approximation for the second derivative was more clear for the C5G7 transient. Consequently, the ISDP methods were uniformly more accurate than the IBDC methods, which were nonetheless accurate. The difference in accuracy between the SDP methods and IBDC was less pronounced for the C5G7 transient than the TWIGL transients.

The SDP methods required substantially less memory than the RBDC methods, especially when high-order time integration was performed. The SDP methods were also more computationally efficient than RBDC, but required somewhat more run-time and memory than IBDC.

8.4 Summary and Conclusion

We modeled three transients using DeCART to test the new SDP methods. The first two transients were based on the TWIGL reactor which includes a relatively simple geometry and slow-to-moderately-fast changes in the neutron flux. The third transient is based on the C5G7 problem, and it is more challenging because it has greater spatial heterogeneity and is a faster transient.

The SDP methods accurately represented the reference solution for all three transients. The SDP methods converged toward the reference solution as the time step was refined and the order of the method was increased. For the TWIGL ramp transient, the methods exhibited the BDF order of convergence. The other two transients began with a step change, which limited the accuracy improvement of the high-order methods.

The SDP methods also accurately captured the solution using the reference method (RBDC) when the same time step and order was employed. This indicates that the propagated angular flux using SDP is representative of the local finite differenced angular flux time derivative approximation used in RBDC.

The SDP methods with the isotropic correction were more accurate for all transients than IBDC, which is a popular approximate method. In particular, the ISDP methods captured the spatial distribution of the power better than the IBDC methods, although for the C5G7 transient the SDP methods only did slightly better at capturing the total magnitude of the solution. The SDP methods did require slightly more memory and computational resources than IBDC, but the SDP methods also required substantially less memory than RBDC. Nonetheless the IBDC methods are probably adequate for many applications.

Chapter 9

Summary, Conclusions, and Future Work

9.1 Summary of Work

In this thesis, we developed a new time-dependent method of characteristics (MOC) formulation that treats the angular flux time derivative by propagation along characteristics. This is an application of a new concept called angular flux time derivative propagation (TDP). While alternate definitions of the angular flux time derivative along the characteristic can be made, we focused on a definition that recasts the angular flux time derivative in terms of neutron source time derivatives, and hence this method is called source derivative propagation (SDP).

SDP was developed in light of the challenges introduced by conventional time integration techniques for time-dependent neutron transport. Specifically, conventional techniques treat the time derivative using a local finite difference approximation which implies that the angular flux should be stored. Storing the angular flux for reactor problems entails prohibitively large memory requirements. We developed the SDP methods to circumvent this obstacle because approximating the neutron source time derivative requires substantially less memory than directly storing the angular flux.

A further advantage of this approach is that the source time derivatives can be approximated to high-order accuracy using backward differences without requiring excessive memory. This is analogous to the Backward Differentiation Formula (BDF), and when this technique is conventionally applied to time-dependent neutron transport the memory requirements increase linearly with the order of the method.

We investigated three SDP methods in this thesis. The first method was originally derived [Hof13a] using analytically-integrated space-time characteristics (ASTC), although we identified an alternate derivation while analyzing ASTC. The alternate derivation was presented in this thesis and was preferred because it is more succinct, it clearly identified the leading error terms, and it suggests options for refinement.

From the alternate derivation, it is clear that ASTC is equivalent to an SDP method where the angular flux second-derivative error term is truncated, and thus that method is also called truncated SDP (TSDP). An alternative to truncating the error term that does not sacrifice the advantages of SDP is to approximate the error term as isotropic (ISDP). As expected, ISDP was found to be more accurate than TSDP without incurring substantial additional expense.

TSDP and ISDP were derived using the assumption that the macroscopic cross sections are changing slowly. While this approximation is true throughout most of the space and time domain of most nuclear reactor transients, there may be important cases where it is not applicable. In light of this, we derived an SDP methods which allowed the cross sections to vary linearly in time, resulting in a method that propagated the angular flux time derivative in terms of the source and cross section derivatives. We combined this with the isotropic second derivative approximation to derive the third method, ISCDP.

We assessed the SDP methods by comparison to two conventional time-dependent MOC methods. These two methods treated the angular flux time derivative using a local backward difference and thus they were referred to as backward difference time-dependent MOC (BDC). The first method approximated the angular flux time derivative using the segment-wise average angular flux. This reference method (RBDC) required substantial memory which limited the size of the problems that could be modeled, but it served as a reference for accuracy. The second method assumed that the angular flux time derivative was isotropic (IBDC). This is a popular approximation for time-dependent neutron transport, and it serves as a benchmark for computational efficiency.

We performed error analysis on the BDC and SDP methods to understand the leading error terms for their angular flux time derivative approximation. All of the methods had similar error scaling in time and space, but the coefficients of the error terms were different. Based on the error analysis, we identified that there may be conditions when TSDP and ISDP are less accurate than IBDC. Specifically, TSDP and ISDP could be less accurate than IBDC when the cross sections are changing quickly, although ISCDP should be more accurate in this situation. In addition, TSDP may be less accurate than IBDC when angular distribution of the neutron flux is changing slowly, but the second derivative of the scalar flux is large in magnitude. However, these two conditions were considered unlikely because the cross sections generally change slowly, and the SDP error terms were divided by the neutron velocity which reduced their magnitude. Neither situation was observed in the test problems.

We empirically evaluated the accuracy of the SDP methods using three test transients using the computer code DeCART. The test problems were modeled in 2D due to the limitations of the DeCART transport method, and they were limited in size due to the memory requirements of storing the angular flux for RBDC.

The first two transients were based on the TWIGL reactor benchmark [Yas65]. The first TWIGL transient was a slow exponential power increase driven by a linear cross section ramp. The TWIGL ramp permitted a wide range of time steps and was well-suited to evaluating the order of accuracy of the new method. The second TWIGL transient was driven by a step change and it resulted in a faster, more challenging transient. The third transient was based on the C5G7 problem [Lew01] with a control rod ejection. This transient was more challenging because it was faster and incorporated greater spatial heterogeneity.

We generated a reference solution for each transient using RBDC with a fine time step. We confirmed that the SDP methods converged to the reference solution as the time step size was reduced and the order of the source derivative approximation was increased. For the TWIGL ramp transient, the SDP methods exhibited the theoretical order of convergence, although the order of convergence for high-order methods was limited for the TWIGL step change transient and the C5G7 transient because of the step change. The high-order methods were nonetheless more accurate than the low-order methods when the time steps were not unreasonably large.

We also compared the SDP methods with RBDC methods of the same order and time step. The SDP and RBDC solutions were in good agreement for all transients, indicating that the propagated angular flux derivative of the SDP methods was close to the angular flux time derivative approximation used in RBDC. The IBDC methods were similarly evaluated, and the agreement between the SDP methods and the RBDC methods was at least one order of magnitude better than the agreement between IBDC and RBDC, indicating that SDP was more accurate than IBDC.

Among the SDP methods, ISDP was more accurate than TSDP by most measures for most transients, although the difference was small in all cases. ISCDP was also more accurate than TSDP and generally more accurate than ISDP, but this difference was small as well. This indicates that the TSDP approximation for the angular flux time derivative incorporated most of the necessary detail to represent the angular flux time derivative for the transients we investigated.

The efficiency of the SDP methods was assessed by comparing the run-time and memory requirements of the SDP methods to the BDC methods. The SDP methods required 50-100% more run-time than the IBDC methods, which are the baseline for efficiency. However, the SDP methods were all faster than the RBDC methods. In contrast to the RBDC methods, the SDP methods did not require additional run-time when high-order derivative approximations were used.

The SDP methods required about 10% more memory than the IBDC methods. This is comparable to the additional memory requirements of storing the neutron currents on the transport mesh. Both SDP and IBDC required about two orders of magnitude less memory than the RBDC methods, and the discrepancy increased when high-order derivative approximations were used.

9.2 Assessment of SDP Compared to Talamo & Tsujita's MOC Methods

Alberto Talamo [Tal13] and Kosuke Tsujita [Tsu13] developed time-dependent MOC formulations for reactor kinetics around the same time that SDP was developed. These methods are described in Chapter 2, but here we will assess SDP in light of them.

Talamo developed three time-dependent MOC formulations. The first two methods (methods I and II) are identical to IBDC1 and RBDC1 respectively. The equations for method III are equivalent to method I, but used a different numerical structure which improves the stability of the method when very small time steps are used (e.g. time steps of 1 μs and smaller). However, method III is more computationally expensive and requires significant modifications to the MOC algorithm.

Talamo's methods are representative of the conventional approach to time-dependent neutron transport where the time derivative is treated using a finite difference approximation. By contrast, the SDP methods treat the angular flux time derivative by propagation along characteristics. As we discussed in Chapters 4 and 8, RBDC1/method II is in principle accurate but requires excessive memory. IBDC1/method I avoids the memory requirement but introduces an approximation which may be inaccurate. The SDP methods were designed with these issues in mind, and our results suggest that the SDP methods provide improved accuracy over IBDC while avoiding the extreme memory requirements of RBDC.

Tsujita's on-the-fly (OTF) time-dependent MOC formulation was also motivated by the high memory cost of RBDC1 and the uncertain validity of the approximation in IBDC1. Like Talamo's formulations, the OTF method also treats the angular flux time derivative using a local finite difference approximation. However, instead of storing or approximating the angular flux from previous points in time, it is recalculated on-the-fly using data stored from previous time steps (e.g. the neutron sources, macroscopic cross sections, etc.). Thus, whereas the SDP methods propagate the angular flux time derivatives concurrently with the angular flux, the OTF methods propagate angular fluxes from previous points in time concurrently with the angular flux. The OTF method solves the equation:

$$\frac{d\varphi_g^{k,n}(s)}{ds} + \left(\Sigma_{tr,g}^{r,n} + \frac{1}{v_g \Delta t_n}\right)\varphi_g^{k,n}(s) = q_g^{r,n} - \frac{\bar{\varphi}_{g,\mathrm{OTF}}^{k,n-1}}{v_g \Delta t_n}, \qquad 9.1$$

where:

$$\begin{aligned}\bar{\varphi}_{g,\mathrm{OTF}}^{k,n-1} = {} & \left(\varphi_{g,0}^{k,n-1} - \frac{q_g^{r,n-1}}{\hat{\Sigma}_{tr,g}^{r,n-1}} + \frac{\bar{\varphi}_{g,\mathrm{OTF}}^{k,n-2}}{\hat{\Sigma}_{tr,g}^{r,n-1} v_g \Delta t_{n-1}}\right)\left(\frac{1 - e^{-\hat{\Sigma}_{tr,g}^{r,n-1}\Delta s^{k,r}}}{\hat{\Sigma}_{tr,g}^{r,n-1}\Delta s^{k,r}}\right) \\ & + \frac{q_g^{r,n-1}}{\hat{\Sigma}_{tr,g}^{r,n-1}} - \frac{\bar{\varphi}_{g,\mathrm{OTF}}^{k,n-2}}{\hat{\Sigma}_{tr,g}^{r,n-1} v_g \Delta t_{n-1}},\end{aligned} \qquad 9.2$$

where $\hat{\Sigma}_{tr,g}^{r,n-1} \equiv \Sigma_{tr,g}^{r,n-1} + \frac{1}{v_g \Delta t_{n-1}}$ and $\varphi_{g,0}^{k,n-1}$ is solved by propagating the angular flux from the previous time step using Equation 9.1.

While this can be done without approximation, it results in linearly increasing memory and computational requirements with the number of time steps. To avoid this problem, Tsujita truncates the OTF recalculation at some number of previous time steps and applies the isotropic approximation at that level. When the OTF method is truncated to N previous time steps (i.e. OTFN), the equation for the average angular flux for that previous point in time is:

$$\begin{aligned}\bar{\varphi}_{g,\mathrm{OTF}}^{k,n-N} = {} & \left(\varphi_{g,0}^{k,n-N} - \frac{q_g^{r,n-N}}{\Sigma_{tr,g}^{r,n-N}} + \frac{\phi_g^{r,n-N} - \phi_g^{r,n-N-1}}{4\pi \Sigma_{tr,g}^{r,n-N} v_g \Delta t_{n-N}}\right)\left(\frac{1 - e^{-\Sigma_{tr,g}^{r,n-N}\Delta s^{k,r}}}{\Sigma_{tr,g}^{r,n-N}\Delta s^{k,r}}\right) \\ & + \frac{q_g^{r,n-N}}{\Sigma_{tr,g}^{r,n-N}} - \frac{\phi_{g,\mathrm{OTF}}^{r,n-N} - \phi_{g,\mathrm{OTF}}^{r,n-N-1}}{4\pi \Sigma_{tr,g}^{r,n-N} v_g \Delta t_{n-N}}.\end{aligned} \qquad 9.3$$

This introduces an error term in the angular flux solution which is proportional to ${v_g}^{-N-1}$. Tsujita found that OTF1 was very accurate and OTF0 (i.e. IBDC1) was reasonably accurate for the transients tested.

OTFN has the same memory requirements as ISCDPN because it requires the storage of the same values from previous points in time. However, whereas the OTF methods use the extra data to diminish the error of the angular approximation, the SDP methods use the additional data to reduce the temporal error through high-order derivative approximations.

Because the OTF1 method requires propagating the angular flux at one previous point in time, it has similar computational cost to the SDP1 methods. However, the computational expense of the OTFN methods increases linearly with *N* because it is necessary to propagate additional previous angular fluxes. By contrast, the high-order SDPN methods do not increase in computational expense with *N*.

While we did not test the OTF methods in this work, it appears that both the OTF and SDP methods offer a viable option for accurately solving the time-dependent neutron transport equations for reactor kinetics. However, on the basis of this work and Tsujita's results, we concluded that the SDP methods are preferred for three reasons. First, the SDP methods are more easily and inexpensively extended to high-order accuracy. This is important because the time integration error for all but the shortest time steps is larger than the error resulting from the angular approximations. Second, it appears that SDP1 methods provide a greater improvement in accuracy over IBDC1 than OTF1; in Tsujita's work, OTF1 reduced the error by about one-half compared to IBDC1, while in this work SDP1 reduced the error by at least an order of magnitude. Finally, the SDP methods result in a smaller increase in computational expense over IBDC than the OTF methods.

9.3 Disadvantages and Limitations of SDP

Although the SDP methods accurately modeled the transients in this thesis, we can anticipate some disadvantages and limitations to the SDP methods which may limit their utility. In this section we will summarize some of these issues.

9.3.1 Non-convergence to the Solution of the Boltzmann Equation

It is clear that the SDP methods will not converge to the solution of the time-dependent Boltzmann equation as the angular, spatial, and temporal discretization. This is due to the error terms that result from the assumptions about the time-dependence of the cross sections and the approximation of the higher derivative term (e.g. by truncation or an isotropic assumption). Although we can identify these error terms and qualitatively describe the conditions where the methods will be accurate, it may be difficult in practice to state a priori whether the SDP methods will be sufficiently accurate for a given transient. IBDC and the OTF method have similar limitations.

A potential resolution to this problem would be to implement an arbitrary Nth derivative propagation method (i.e. as described in Section 5.8) and to simultaneously solve the SDP equations for an Nth derivative method and an N-1th derivative method. The two solutions can be compared to assess the error associated with approximating the Nth derivative. This concept is analogous to embedded adaptive time stepping methods, and it would be substantially more computationally and memory intensive.

9.3.2 Less Attractive with Higher-Order Spatial Variation of Source

The SDP methods were investigated because they have a convenient mathematical form which allows the time derivatives to be efficiently represented using the neutron source time derivative. However, while this work was limited to the step characteristics method (where the source is assumed to be spatially-invariant within regions), but there is interest in allowing the neutron source to vary spatially within regions (e.g. a linear variation). This will result in substantially more complex SDP equations, reducing the computational and memory efficiency of the method. On the other hand, this may result in a more accurate representation of the angular flux time derivative within the region. In any case, this warrants further investigation.

9.3.3 Less Attractive when High-order Moments are Required for Scattering

We investigated the SDP methods using transport-corrected scattering, which allows the scattering term to be treated as isotropic. However, if higher moments of the angular flux are calculated to represent anisotropic scattering, it may be more efficient to simply approximate the angular flux time derivative using a P_N expansion. On the other hand, the high-order moments can also be used with an SDP method to improve the approximation of the second derivative of the angular flux.

9.3.4 Error for IBDC is Modest for Transients in this Thesis

Although the SDP methods were generally more accurate than IBDC for the transients investigated in this thesis, the error for IBDC was nonetheless relatively small. When considering that IBDC is more computationally efficient and requires less memory than the SDP methods, one may conclude that IBDC offers a better balance of accuracy to computational expense than the SDP methods.

9.4 Suggested Future Work

The SDP methods show good promise for generating high-fidelity solutions to the time-dependent neutron transport equation for nuclear reactor kinetics. However, there are many opportunities for future research.

9.4.1 More Realistic and/or Challenging Transients

We were limited in the size of the transients we could model to test the SDP methods because of the high memory requirements for the reference solution. This is inevitable given the nature of the problem, but it could be overcome to a limited extent through parallel decomposition of the reactor model in space. DeCART is capable of distributed memory parallel decomposition in space, but only in the axial direction for modeling 3D problems. This was not employed in this work because the diffusion-based nodal expansion method (NEM) in the axial direction would contaminate the transport solution and make it difficult to assess the accuracy of the transport methods.

The lack of a third dimension and the constrained model size limits the realism of the transients we modeled. This is because small problems either incorporate very large spatial gradients (e.g. if they have vacuum boundary conditions), or they are relatively isotropic (e.g. if they have all reflecting boundary conditions). In addition, as a result of neglecting the axial dimension, we were unable to represent partially-inserted control rods, and the rod ejection for the C5G7 transient was approximated as a step change.

However, as a consequence of these limitations, the C5G7 transient was particularly challenging. It is promising that the SDP methods performed so well in spite of this. Nonetheless, it would be valuable to explore both more realistic transients to understand typical SDP performance as well as more challenging transients to understand the limits of the SDP methods.

9.4.2 Alternate Equations for Angular Flux Time Derivative Propagation

The equation for the angular flux time derivative along the characteristic was originally identified using ASTC, and it was investigated because of its favorable mathematical properties. However, alternate equations may be derived that also have favorable properties. For example, methods could be derived using finite differencing or other spatial methods.

Similarly, we might also consider alternative methods for source derivative approximation. We used backward differences to approximate the source derivatives to high-order accuracy because they are easy to implement, but these high-order methods are not as useful near step changes. We might consider alternate methods to approximate the time derivatives such as Runge-Kutta methods or an exponential transform.

In addition, there may be approximations for the SDP equations that can reduce the computational expense without substantially impairing the accuracy. For example, we could approximate the angular flux time derivative in the angular flux characteristic equation using the segment-wise average angular flux time derivative:

$$\frac{d\varphi_g^{k,n}(s)}{ds} + \Sigma_{tr,g}^{r,n}\varphi_g^{k,n}(s) = q_g^{r,n} - \frac{1}{v_g}\frac{d\bar{\varphi}_g^{k,n}}{dt}, \tag{9.4}$$

where the segment-wise average angular flux time derivative is calculated by integrating over the SDP equation along the segment:

$$\frac{d\bar{\varphi}_g^{k,n}}{dt} = \frac{1}{\Delta s}\int_{s_0}^{s_0+\Delta s} \frac{d\varphi_g^{k,n}(s)}{dt}\,ds, \tag{9.5}$$

and the spatially-dependent angular flux time derivative is propagated using the usual TSDP, ISDP, or ISCDP equation. This method may even be in better agreement with RBDC due to the similar approximation of the spatial dependence of the angular flux time derivative. It would entail the same memory requirements and have similar computation cost as the SDP methods described in Chapter 5.

9.4.3 Embedded Adaptive Time Stepping Based on *N* and *N*+1 Order Methods

Since this SDP method is capable of generating variable-order solutions by adjusting the number of previous solutions used to approximate the time derivative, this could provide the basis for an adaptive time stepping method. Adaptive time stepping is especially attractive for neutron transport because each time step is computationally expensive.

In this concept, the method would generate solutions of order N and $N+1$. The solutions would be compared, and the error would be used with the theoretical order of accuracy to adaptively adjust the time step size. Shim et al. [Shi11] used a similar approach in the BDF-based diffusion code RENUS to good effect.

9.4.4 Hybrid OTF/SDP Method

Tsujita et al. [Tsu13] developed an on-the-fly (OTF) time-dependent neutron transport method which recalculated angular fluxes from previous points in time to treat the angular flux time derivative. This method was discussed in Section 9.3.2. It has a similar motivation to the SDP methods, and it requires storing the same data (e.g. the previous neutron sources, cross sections, and boundary fluxes). Since both methods require that the same data is stored from previous points in time, it may be possible to develop a hybrid method that uses both concepts to greater effect.

9.4.5 Applicability of SDP to Other Transport Methods

The SDP concept could be applied to many of the other neutron transport methods described in Chapter 2. The application of the SDP concept to Method of Characteristic Direction Probabilities (CDP) is the most straightforward because the same equations are used; the equations are just solved using a linear system of equations rather than a sweeping procedure. For example, if TSDP1 was applied to CDP as implemented in MPACT [Liu13], the propagated angular flux time derivative would be solved in terms of transmission probabilities:

$$\frac{d\varphi_{g,\mathrm{OUT}}^{k,n}}{dt} = \frac{d\varphi_{g,\mathrm{IN}}^{k,n}}{dt} T_{g,\mathrm{IN}\to\mathrm{OUT}}^{k,n} + \sum_{j=1}^{J} \frac{1}{\Sigma_{tr,g}^{r(j),n}} \frac{dq_g^{r,n}}{dt} T_{g,j\to\mathrm{OUT}}^{k,n}, \qquad 9.6$$

where the transmission probabilities are given by:

$$T_{g,\mathrm{IN}\to\mathrm{OUT}}^{k,n} = e^{-\sum_{j=1}^{J}\left(\Sigma_{tr,g}^{r(j),n}\Delta s^{k,r(j)}\right)},$$
$$T_{g,j\to\mathrm{OUT}}^{k,n} = \left(1 - e^{-\Sigma_{tr,g}^{r(j),n}\Delta s^{k,r(j)}}\right) e^{-\sum_{l=j+1}^{J}\left(\Sigma_{tr,g}^{r(j),n}\Delta s^{k,r(l)}\right)}, \qquad 9.7$$

where J is the number of regions along the characteristic.

For other transport methods, the application of the concept is more subtle. For S_N methods, we would derive the angular flux time derivative equation in a similar fashion, but likely using finite differences or another spatial discretization technique to treat the propagation. For the Collision Probability Method (CPM) we would likely rewrite the neutron source derivatives in terms of the neutron scalar flux and solve for the new scalar flux as usual. However, these concepts need to be explored in detail to understand their applicability.

9.4.6 Applicability of SDP to the Axial Derivative Approximation for 2D/1D

When 3D neutron transport is done using a 2D/1D approach (e.g. DeCART), there is an angular flux axial derivative $\frac{d\varphi}{dz}$ that introduces an analogous memory problem to the angular flux time derivative. As for the time derivative, computer codes often use a low-order angular and spatial approximation to treat the axial derivative.

The SDP concept could also be applied to treat the axial derivative in planar MOC. This results in an equation for the propagation of the angular flux axial derivative along the segment in terms of the source axial derivatives, e.g.:

$$\frac{d\varphi_g^k(s)}{dz} = \frac{d\varphi_{g,0}^{k,r}}{dz} e^{-\Sigma_{tr,g}^{r}(s-s_0)} + \frac{dq_g^r}{dz}\left(\frac{1 - e^{-\Sigma_{tr,g}^{r}(s-s_0)}}{\Sigma_{tr,g}^{r}}\right), \qquad 9.8$$

where the axial source derivative could be approximated using finite differences or another technique. By combining this approach with SDP for the time derivative, 3D time-dependent MOC could be performed using a 2D planar MOC method. This is potentially attractive because 2D planar MOC methods are computationally efficient and well understood.

9.5 Closing Remarks

The goal of this research was to develop an accurate, time-dependent neutron transport method for nuclear reactor kinetics that avoids the memory challenge introduced by conventional time integration techniques but provides better accuracy than low-order spatial and angular approximations like the isotropic approximation. We developed the SDP methods with this goal in mind, and they fulfilled the goal for the test transients we numerically tested. The performance of the SDP methods was consistent with our expectations from error analysis, and we regard them as an favorable prospect for accurate, transport-based nuclear reactor kinetics.

Appendix A

General Method of Characteristics (MOC) Derivation

A.1 Introduction

Method of characteristics (MOC) is a general technique for solving partial differential equations by converting them into ordinary differential equations along characteristic curves in the problem domain. Although the technique may be applied to any hyperbolic partial differential equation, we will consider the limited case of a linear partial differential equation because it is applicable to MOC for neutron transport.

A.2 Derivation of MOC for a Linear Differential Equation

Consider the following linear differential equation:

$$a(x,y)\frac{\partial z}{\partial x} + b(x,y)\frac{\partial z}{\partial y} = c(x,y), \qquad \text{A.1}$$

with the solution $z(x,y)$ which is assumed known.

Consider the surface graph of the function $z = z(x,y)$. We can rewrite Equation A.1 as a dot product:

$$\big(a(x,y), b(x,y), c(x,y)\big) \cdot \left(\frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}, -1\right) = 0. \qquad \text{A.2}$$

We know that $\left(\frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}, -1\right)$ is a normal vector to the surface z at every point. This means that the vector (a, b, c) is tangent to the surface at every point, and thus the solution is the union of integral curves described by these vectors. These integral curves are called characteristic curves, and each may be uniquely defined as the solution of a system of ordinary differential equations:

$$\frac{dx}{ds} = a(x, y),$$
$$\frac{dy}{ds} = b(x, y), \tag{A.3}$$
$$\frac{dz}{ds} = c(x, y).$$

We rewrite the original partial differential equation in terms of Equation A.3 to get the characteristic equation in s:

$$\frac{dx}{ds}\frac{\partial z}{\partial x} + \frac{dy}{ds}\frac{\partial z}{\partial y} = \frac{dz}{ds}. \tag{A.4}$$

This is a general form of the MOC transform used in Chapter 3 (Equation 3.12) and Chapter 4 (Equation 4.15). The transform is also used with time dependence in Appendix B (Equation B.3).

Appendix B
Analytically-integrated Space-Time MOC (ASTC)

B.1 Introduction

In this appendix, we provide the derivation for a time-dependent neutron transport method that employed Method of Characteristics (MOC) to treat the temporal domain in addition to the spatial domain. This Analytically-integrated Space-Time MOC (ASTC) method[Hof13] was the stimulation for the discovery of the Source Derivative Propagation (SDP) methods that are the focus of this thesis. In particular, ASTC is equivalent to the SDP method with a truncated second derivative term (TSDP).

The derivation of SDP in Chapter 5 supersedes this derivation because it is more general, simpler, and more clearly identifies the sources of error in the method. However, the ASTC derivation provides some qualitative understanding of the terms in the angular flux time derivative characteristic equation.

This derivation shares many of the approximations that are employed for the other time-dependent MOC methods in Chapters 4 and 5. This chapter will begin by listing those approximations and then discussing the new approximations for this method. This will result in characteristic equations for the angular flux in time and space. This is the fundamental approach to deriving a space-time MOC (STC).

While STC is in principal very accurate, it is computationally- and memory-intensive because it requires the explicit modeling of many more characteristics than the other time-dependent MOC methods described in this thesis. To avoid this problem for ASTC, the time-dependence of the characteristics within space-time regions are approximated in a way that allows many characteristics to be modeled with a single function. This function is used to solve for the angular flux along spatial segments at the end of the time step, which results in a method that is equivalent to TSDP.

After the space-time characteristic equation is derived, we demonstrate how it is used to solve for the angular flux along spatial segments like other MOC methods. This illustrates that the ASTC equations are equivalent to TSDP.

B.2 Approximations to the Boltzmann Transport Equation for ASTC

B.2.1 Approximations Shared with BDC

The derivation of ASTC begins with the time-dependent Boltzmann transport equation and applies many of the approximations that were used when deriving BDC in Chapter 4. These include: the multi-group approximation, the isotropic source approximation, the discrete ordinates approximation, the delayed neutron group approximation, and analytic precursor integration. After these approximations are applied, the result is the multi-group, step characteristics equation with analytic precursor integration:

$$\begin{aligned}\frac{1}{v_g}\frac{d\varphi_g\left(\mathbf{x},\boldsymbol{\Omega}_{l,m},t\right)}{dt}&\\ &= -\boldsymbol{\Omega}\cdot\boldsymbol{\nabla}\varphi_g\left(\mathbf{x},\boldsymbol{\Omega}_{l,m},t\right) - \Sigma_{tr,g}(\mathbf{x},t)\varphi_g\left(\mathbf{x},\boldsymbol{\Omega}_{l,m},t\right)\\ &+ q_g(\mathbf{x},t),\end{aligned} \tag{B.1}$$

where the source is defined as:

$$\begin{aligned}q_g(\mathbf{x},t) = \frac{1}{4\pi}\sum_{g'=1}^{G}\Bigg[\Big(&[1-\beta]\chi_{p,g}(\mathbf{x})\\ &+ \chi_{d,g}(\mathbf{x})\omega(\mathbf{x},t)\Big)\frac{\nu\Sigma_{f,g'}(\mathbf{x},t)}{k_{\text{eff}}}\phi_{g'}(\mathbf{x},t)\\ &+ \Sigma_{s,g'\to g}^{tr}(\mathbf{x},t)\phi_{g'}(\mathbf{x},t)\Bigg] + \frac{\tilde{S}_{d,g}(\mathbf{x},t)}{4\pi}.\end{aligned} \tag{B.2}$$

B.2.2 Space-Time Characteristic Transformation

For the ASTC derivation, the MOC transform is defined in space and time:

$$
\begin{aligned}
\frac{dt}{d\varsigma} &= \frac{1}{v_g}, \\
\frac{dx}{d\varsigma} &= \sqrt{1-\mu_m^2}\cos\alpha_l, \\
\frac{dy}{d\varsigma} &= \sqrt{1-\mu_m^2}\sin\alpha_l, \\
\frac{dz}{d\varsigma} &= \mu_m,
\end{aligned}
\qquad \text{B.3}
$$

where ς is the dimension along the space-time characteristic in contrast to the dimension s for spatial characteristics.

This results in a characteristic equation that spans time and space:

$$
\frac{d\varphi_g^\kappa(\varsigma)}{d\varsigma} + \Sigma_{tr,g}^\kappa(\varsigma)\varphi_g^\kappa(\varsigma) = q_g^\kappa(\varsigma), \qquad \text{B.4}
$$

where κ is the index for the space-time characteristics, in contrast to k which was used for the spatial characteristics for the other MOC methods. κ and ς are used for space-time characteristics because we will eventually translate the method back to spatial characteristics.

B.2.3 Spatial and Temporal Discretization

Next we will discretize the space and time domain. As we did for the other MOC methods, the spatial domain is discretized into narrow regions were we can approximate the spatial dependence of the source and cross sections. Similarly, the time domain will be discretized into narrow temporal regions which are called "time steps." Together this four dimensional space is referred to as a space-time region. Figure B-1 shows an example of space-time characteristics for a spatially- and temporally-discretized pin cell in two spatial dimensions. The spatial characteristics used in the other time-dependent MOC methods can be understood as the projection of the space-time characteristics into a plane in space at a particular time.

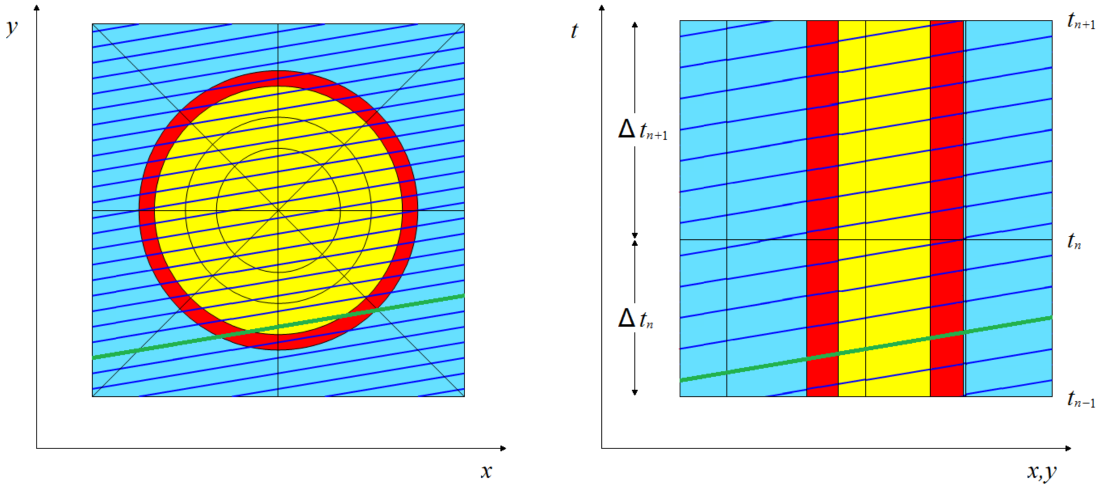


Figure B-1. Example of space-time characteristics for pin cell [adapted from Hur08]

B.2.4 Approximation of Source and Cross Section Time Dependence

We can analytically solve Equation B.4 within the space-time regions if we approximate the spatial dependence of the neutron source and cross section along the characteristics. In the previous chapters we assumed that the source and cross section were spatially-invariant along the segment within the narrow region. However, for ASTC the characteristics traverse time and space. Other researchers observed that the flat source approximation in time is inaccurate[Kel98, Pan11].

Instead, we will assume that the source varies linearly in time but is flat in space. As a result, the variation of the source along the characteristic is based exclusively on its changing temporal position. To create an implicit method, we will define the time dependence of the source in terms of the source at the end of the time step for each region:

$$q_g^{\kappa}(\varsigma) \approx q_g^{r,n} - \left(t_n - t^{\kappa}(\varsigma)\right)\frac{dq_g^{r,n}}{dt}, \qquad \text{B.5}$$

where r is the index for the region, n is the index for the time t_n which corresponds to the end of the time step, $q_g^{r,n}$ is the source at the end of the time step, and $t^{\kappa}(\varsigma)$ is the time corresponding to the position ς along the characteristic κ.

The cross section also requires an approximation. However, since the cross section varies slowly in time, we will assume that the cross section is spatially and temporally invariant within regions:

$$\Sigma_{tr,g}^{\kappa}(\varsigma) \approx \Sigma_{tr,g}^{r,n}. \qquad \text{B.6}$$

When we apply these approximations, the characteristic becomes:

$$\frac{d\varphi_g^{\kappa}(\varsigma)}{d\varsigma} + \Sigma_{tr,g}^{r,n}\varphi_g^{\kappa}(\varsigma) = q_g^{r,n} - \left(t_n - t^{\kappa}(\varsigma)\right)\frac{dq_g^{r,n}}{dt}, \qquad \text{B.7}$$

where the space-time region r corresponds to the position ς.

We note that the time corresponding to the spatial position can be rewritten in terms of the time where the characteristic enters the space-time region:

$$t^{\kappa}(\varsigma) = t^{\kappa}(\varsigma_0^{\kappa,r}) + \frac{\varsigma}{v_g}, \qquad \text{B.8}$$

where $\varsigma_0^{\kappa,r}$ is the position in s which corresponds to the entry of the characteristic κ in region r.

We apply this relation to Equation B.7:

$$\frac{d\varphi_g^{\kappa}(\varsigma)}{ds} + \Sigma_{tr,g}^{r,n}\varphi_g^{\kappa}(\varsigma) = q_g^{r,n} - \left(t_n - t^{\kappa}(\varsigma_0^{\kappa,r}) - \frac{\varsigma}{v_g}\right)\frac{dq_g^{r,n}}{dt}. \qquad \text{B.9}$$

B.2.5 Approximation of Source Time Derivative with Backward Differences

We can approximate the source time derivative to a varying order of accuracy using backward differences as described in Chapter 5. This allows us to write the source time derivative as a weighted sum of the source at previous points in time in an analogous fashion to BDF:

$$\frac{dq_g^{r,n}}{dt} = \sum_{i=0}^{O_{\mathrm{BD}}} A^i q_g^{r,n-i}. \qquad \text{B.10}$$

Applying Equation 5.14 to Equation B.9 gives the space-time characteristic equation for the angular flux with a linear source:

$$\frac{d\varphi_g^{\kappa}(\varsigma)}{ds} + \Sigma_{tr,g}^{r,n}\varphi_g^{\kappa}(\varsigma) = q_g^{r,n} - \left(t_n - t^{\kappa}(\varsigma_0^{\kappa,r}) - \frac{\varsigma}{v_g}\right)\sum_{i=0}^{o_{\mathrm{BD}}} A^i q_g^{r,n-i}. \tag{B.11}$$

We solve Equation B.11 for the angular flux distribution along a characteristic within a space-time region by using an integrating factor:

$$\begin{aligned}\varphi_g^{\kappa}(\varsigma) = \varphi_g^{\kappa}(\varsigma_0^{\kappa,r})e^{-\Sigma_{tr,g}^{r,n}(\varsigma-\varsigma_0^{\kappa,r})} &+ \frac{\varsigma}{\Sigma_{tr,g}^{r,n}v_g}\sum_{i=0}^{o_{\mathrm{BD}}} A^i q_g^{r,n-i} \\ &+ \left[q_g^{r,n} - \left(t_n - t^{\kappa}(\varsigma_0^{\kappa,r}) + \frac{1}{\Sigma_{tr,g}^{r,n}v_g}\right)\sum_{i=0}^{o_{\mathrm{BD}}} A^i q_g^{r,n-i}\right] \\ &\times \left(\frac{1 - e^{-\Sigma_{tr,g}^{r,n}(\varsigma-\varsigma_0^{\kappa,r})}}{\Sigma_{tr,g}^{r,n}}\right).\end{aligned} \tag{B.12}$$

For a conventional STC method, we would use Equation B.12 to propagate the angular fluxes along parallel characteristics within space-time planes and solve for the angular or scalar fluxes at the end of the time step. However, this approach is too computationally expensive and memory intensive for large reactor kinetics problems, and we require another approximation to derive a practical method.

B.2.6 Approximation of the Time Dependence of the Incoming Angular Flux

We note that for parallel characteristics in the same space-time region, in the only quantities Equation B.12 which differ between the characteristics are the incoming angular flux $\varphi_g^{\kappa}(\varsigma_0^{\kappa,r})$ and the time corresponding to the entry of the characteristic in the space-time region $\varsigma_0^{\kappa,r}$. All of the other quantities are isotropic and defined at the region-level.

In light of this, we will approximate the incoming angular flux in terms of the incoming time of the characteristic:

$$\varphi_g^{\kappa}(\varsigma_0^{\kappa,r}) \approx \varphi_{g,0}^{\kappa,n} - \left(t_n - t^{\kappa}(\varsigma_0^{\kappa,r})\right)\frac{d\varphi_{g,0}^{\kappa,n}}{dt}, \tag{B.13}$$

where $\varphi_{g,0}^{k,n}$ is the incoming angular flux for a space-time characteristic that enters the space-time region at time t_n (i.e. the end of the time step) and $\frac{d\varphi_{g,0}^{\kappa,n}}{dt}$ is the time derivative of that angular flux.

When we apply this approximation to Equation B.12, the result is a single characteristic equation that can represent any of the segments within a space-time region, where the only value that differs between the characteristics is the incoming time of the characteristic $\varsigma_0^{\kappa,r}$:

$$\begin{aligned}\varphi_g^{\kappa}(\varsigma) = \left[\varphi_{g,0}^{\kappa,n} - \left(t_n - t^{\kappa}\left(\varsigma_0^{\kappa,r}\right)\right)\frac{d\varphi_{g,0}^{\kappa,n}}{dt}\right] e^{-\Sigma_{tr,g}^{r,n}\left(\varsigma-\varsigma_0^{\kappa,r}\right)} + \frac{\varsigma}{\Sigma_{tr,g}^{r,n} v_g}\sum_{i=0}^{o_{\mathrm{BD}}} A^i q_g^{r,n-i} \\ + \left[q_g^{r,n} - \left(t_n - t^{\kappa}\left(\varsigma_0^{\kappa,r}\right) + \frac{1}{\Sigma_{tr,g}^{r,n} v_g}\right)\sum_{i=0}^{o_{\mathrm{BD}}} A^i q_g^{r,n-i}\right] \\ \times\left(\frac{1 - e^{-\Sigma_{tr,g}^{r,n}\left(\varsigma-\varsigma_0^{\kappa,r}\right)}}{\Sigma_{tr,g}^{r,n}}\right).\end{aligned} \tag{B.14}$$

B.3 Solution of the Characteristic Equations for ASTC

B.3.1 Translation of Space-Time Characteristics to Spatial Characteristics

Equation B.14 can be used to solve for the angular flux at any spatial and temporal position within the space-time region. However, we only require the angular flux at time t_n so that we can numerically integrate the region-wise scalar flux. This is the angular flux along all characteristics that exit the space-time region along the temporal boundary at time t_n.

The exiting position of these characteristics along the temporal boundary is linearly related to the incoming time by the equation:

$$s = v_g\left(t_n - t^{\kappa}\left(\varsigma_0^{\kappa,r}\right)\right), \tag{B.15}$$

where s is the position in space where the characteristic κ crosses the temporal boundary.

We rewrite Equation B.14 to give the angular flux at time t_n as a function of s:

$$\begin{aligned}\varphi_g^{\kappa,n}(s) &= \left[\varphi_{g,0}^{\kappa,n} - \frac{s}{v_g}\left(\frac{d\varphi_{g,0}^{\kappa,n}}{dt} - \frac{1}{\Sigma_{tr,g}^{r,n}}\sum_{i=0}^{O_{\mathrm{BD}}} A^i q_g^{r,n-i}\right)\right] e^{-\Sigma_{tr,g}^{r,n} s} \\ &+ \left(q_g^{r,n} - \frac{1}{\Sigma_{tr,g}^{r,n} v_g}\sum_{i=0}^{O_{\mathrm{BD}}} A^i q_g^{r,n-i}\right)\left(\frac{1 - e^{-\Sigma_{tr,g}^{r,n} s}}{\Sigma_{tr,g}^{r,n}}\right).\end{aligned} \tag{B.16}$$

Thus the dependency of the characteristic equation on the space-time dimension ς has been replaced by the spatial dimension s at time t_n. This is completely equivalent to the definition of the spatial characteristics we defined for the other MOC methods, and we can rewrite Equation B.16 in terms of the spatial characteristic k:

$$\begin{aligned}\varphi_g^{k,n}(s) &= \left[\varphi_{g,0}^{k,n} - \frac{s}{v_g}\left(\frac{d\varphi_{g,0}^{k,n}}{dt} - \frac{1}{\Sigma_{tr,g}^{r,n}}\sum_{i=0}^{O_{\mathrm{BD}}} A^i q_g^{r,n-i}\right)\right] e^{-\Sigma_{tr,g}^{r,n} s} \\ &+ \left(q_g^{r,n} - \frac{1}{\Sigma_{tr,g}^{r,n} v_g}\sum_{i=0}^{O_{\mathrm{BD}}} A^i q_g^{r,n-i}\right)\left(\frac{1 - e^{-\Sigma_{tr,g}^{r,n} s}}{\Sigma_{tr,g}^{r,n}}\right),\end{aligned} \tag{B.17}$$

where s is the spatial position along the characteristic k at time t_n.

This is the solution to the characteristic equation for ASTC, which is identical to the solution of the characteristic equation for TSDP in Chapter 5. Consequently, the equations for ASTC to propagate the angular flux and to calculate the segment-wise average angular flux are also identical. However, we still have to derive the equation to propagate the angular flux time derivative.

B.3.2 Propagation of the Angular Flux Time Derivative for ASTC

We can estimate the outgoing angular flux time derivative by taking a finite difference approximation of the outgoing angular flux for any two space-time characteristics using Equation B.14, e.g.:

$$\frac{d\varphi_g^{k,n}(\Delta s)}{dt} \approx \frac{\varphi_g^{\kappa,n}(\Delta\varsigma) - \varphi_g^{\kappa-1,n}(\Delta\varsigma)}{t^{\kappa}(\Delta\varsigma) - t^{\kappa-1}(\Delta\varsigma)}. \tag{B.18}$$

Regardless of which space-time characteristics we use, the non-derivative terms cancel and the result is:

$$\frac{d\varphi_g^{k,n}(\Delta s)}{dt} \approx \frac{d\varphi_{g,0}^{k,n}}{dt} e^{-\Sigma_{tr,g}^{r,n}\Delta s} + \sum_{i=0}^{O_{\mathrm{BD}}} A^i q_g^{r,n-i} \left(\frac{1 - e^{-\Sigma_{tr,g}^{r,n}\Delta s}}{\Sigma_{tr,g}^{r,n}} \right), \tag{B.19}$$

which is identical to Equation 5.26, which is used in TSDP to propagate the angular flux time derivative.

B.4 Summary and Conclusions

In this appendix we derived the characteristic equations for ASTC, which produce an identical method to TSDP. The development of ASTC stimulated the discovery of the SDP methods, one of which is equivalent to ASTC. The derivation requires the specification of characteristics in space-time and the approximation of the angular flux in the space-time plane. The ASTC derivation is less general and more complex than the SDP derivation, and thus the latter has been the focus of this thesis. The derivation for ASTC was provided for historical reasons and to provide qualitative insight into the SDP methods.

Appendix C
Time-Dependent MOC with Spatially-Dependent Angular Fluxes

C.1 Introduction

In Chapter 4, when deriving the reference time-dependent MOC method (RBDC), we approximated the finite differenced angular flux time derivative in terms of the segment-wise average angular flux. However, when we solve for angular fluxes in MOC, the segment-wise angular fluxes are spatially-dependent along the segment. As demonstrated in Chapter 6, using the segment-wise average angular flux for the time derivative introduces some spatial error. We could eliminate this error by using the spatially-dependent segment-wise angular flux for the time derivative. However, this is impractical because it requires storing a segment-wise quantity for every previous time step.

In this appendix, we derive the "exact" solution for the segment-wise angular flux in time-dependent MOC using Backward Euler to approximate the time derivative. We will solve for the angular flux at the end of the first two time steps and provide a general equation for the angular flux at an arbitrary time step.

C.2 Approximations to the Boltzmann Transport Equation

C.2.1 Approximations Shared with BDC

We begin by making all of the same assumptions that were used for the Backward Difference MOC (BDC) methods in Chapter 4 except for the spatial approximation of the angular flux time derivative. This includes: the multigroup approximation, the isotropic source approximation, the discrete ordinates approximation, the delayed neutron group approximation, the spatial characteristic transform, step characteristics, time discretization, and analytic precursor integration.

The resulting characteristic equation is:

$$\frac{1}{v_g}\frac{d\varphi_g^{k,n}(s)}{dt}+\frac{d\varphi_g^{k,n}(s)}{ds}+\Sigma_{tr,g}^{r,n}\varphi_g^{k,n}(s)=q_g^{r,n}. \tag{C.1}$$

C.2.2 Backward Euler

Next we approximate the angular flux time derivative using Backward Euler:

$$\frac{d\varphi_g^{k,n}(s)}{dt}=\frac{\varphi_g^{k,n}(s)-\varphi_g^{k,n-1}(s)}{\Delta t_n}. \tag{C.2}$$

Note that unlike the BDC chapter, the angular flux terms in the time derivative have retained their spatial dependence along the characteristic. When we apply Equation C.2 to Equation C.1, we have the characteristic equation for the angular flux for time-dependent MOC:

$$\frac{d\varphi_g^{k,n}(s)}{ds}+\left(\Sigma_{tr,g}^{r,n}+\frac{1}{v_g\Delta t_n}\right)\varphi_g^{k,n}(s)=q_g^{r,n}+\frac{\varphi_g^{k,n-1}(s)}{v_g\Delta t_n}. \tag{C.3}$$

C.3 Solution of Time-Dependent MOC for Time Step 1

To solve the Equation C.3 for any time step, we need to know the spatially-dependent angular flux from the previous time step $\varphi_g^{k,n-1}(s)$. For the first time step, the previous angular flux is the steady-state angular flux:

$$\varphi_g^{k,0}(s)=\varphi_{g,0}^{k,0}e^{-\Sigma_{tr,g}^{r,0}s}+\frac{q_g^{r,0}}{\Sigma_{tr,g}^{r,0}}\left(1-e^{-\Sigma_{tr,g}^{r,0}s}\right), \tag{C.4}$$

where $n=0$ corresponds to the steady-state solution, and we have arbitrarily defined the incoming spatial position as $s=0$ for succinctness.

We substitute Equation C.4 into Equation C.3 for $n=1$:

$$\frac{d\varphi_g^{k,1}(s)}{ds}+\hat{\Sigma}_{tr,g}^{r,1}\varphi_g^{k,1}(s)=\hat{q}_g^{r,1}+\Sigma_{\Delta t,g}^{1}\left(\varphi_{g,0}^{k,0}-\frac{q_g^{r,0}}{\Sigma_{tr,g}^{r,0}}\right)e^{-\Sigma_{tr,g}^{r,0}s}, \tag{C.5}$$

where:

$$\Sigma_{\Delta t,g}^{1}\equiv\frac{1}{v_g\Delta t_1}, \tag{C.6}$$

$$\hat{\Sigma}_{tr,g}^{r,1} \equiv \Sigma_{tr,g}^{r,1} + \Sigma_{\Delta t,g}^{1},$$

$$\hat{q}_g^{r,1} \equiv q_g^{r,0} + \Sigma_{\Delta t,g}^{1} \frac{q_g^{r,0}}{\Sigma_{tr,g}^{r,0}}.$$

We solve the equation using the integrating factor $e^{\hat{\Sigma}_{tr,g}^{r,1} s}$:

$$\varphi_g^{k,1}(s) = \varphi_{g,0}^{k,1} e^{-\hat{\Sigma}_{tr,g}^{r,1} s} + \frac{q_g^{r,1}}{\hat{\Sigma}_{tr,g}^{r,1}} \left(1 - e^{-\hat{\Sigma}_{tr,g}^{r,1} s}\right) + \Sigma_{\Delta t,g}^{1} A_{g,0}^{k,r} B_{g,0,0}^{k,r}(s), \qquad \text{C.7}$$

where we define the coefficients:

$$A_{g,0}^{k,r} \equiv \varphi_{g,0}^{k,0} - \frac{q_g^{r,0}}{\Sigma_{tr,g}^{r,0}},$$

$$B_{g,0,0}^{k,r}(s) \equiv \frac{e^{-\Sigma_{tr,g}^{r,0} s} - e^{-\hat{\Sigma}_{tr,g}^{r,1} s}}{\hat{\Sigma}_{tr,g}^{r,1} - \Sigma_{tr,g}^{r,0}}. \qquad \text{C.8}$$

This is the "exact" spatially-dependent angular flux at the end of the first time step when using Backward Euler. Note that this requires that the following data is stored from the steady-state solution: the incoming angular flux for every segment, the region-wise transport cross section, and the region-wise neutron source.

C.4 Solution of Time Dependent MOC for Time Step 2

For the second time step, we repeat this process, substituting Equation C.7 into Equation C.3 with $N = 2$:

$$\frac{d\varphi_g^{k,2}(s)}{ds} + \hat{\Sigma}_{tr,g}^{r,2} \varphi_g^{k,2}(s) = \hat{q}_g^{r,2} + \Sigma_{\Delta t,g}^{2} \left[A_{g,1}^{k,r} e^{-\hat{\Sigma}_{tr,g}^{r,1} s} + \Sigma_{\Delta t,g}^{1} A_{g,0}^{k,r} B_{g,0,0}^{k,r}(s)\right], \qquad \text{C.9}$$

where we define the coefficients for an arbitrary time step n:

$$\begin{gathered}
\Sigma_{\Delta t,g}^{n} \equiv \frac{1}{v_g \Delta t_n}, \\
\hat{\Sigma}_{tr,g}^{r,n} \equiv \Sigma_{tr,g}^{r,n} + \Sigma_{\Delta t,g}^{n}, \\
\hat{q}_g^{r,n} \equiv q_g^{r,n} + \Sigma_{\Delta t,g}^{n} \frac{\hat{q}_g^{r,n-1}}{\hat{\Sigma}_{tr,g}^{r,n-1}}, \\
A_{g,n}^{k,r} \equiv \varphi_{g,0}^{k,n} - \frac{\hat{q}_g^{r,n}}{\hat{\Sigma}_{tr,g}^{r,n}},
\end{gathered} \tag{C.10}$$

where Equations C.10 are applicable to steady-state (i.e. time step 0) if we define $\Sigma_{\Delta t,g}^{0} \equiv 0$.

The solution to Equation C.9 is:

$$\begin{gathered}
\varphi_g^{k,2}(s) = \varphi_{g,0}^{k,2} e^{-\hat{\Sigma}_{tr,g}^{r,2} s} + \frac{\hat{q}_g^{r,2}}{\hat{\Sigma}_{tr,g}^{r,2}} \left(1 - e^{-\hat{\Sigma}_{tr,g}^{r,1} s}\right) \\
+ \Sigma_{\Delta t,g}^{2} \left[A_{g,1}^{k,r} B_{g,1,1}^{k,r}(s) + \Sigma_{\Delta t,g}^{1} A_{g,0}^{k,r} B_{g,1,0}^{k,r}(s)\right],
\end{gathered} \tag{C.11}$$

where:

$$\begin{gathered}
B_{g,1,0}^{k,r}(s) \\
\equiv \frac{\left(\hat{\Sigma}_{tr,g}^{r,2} - \hat{\Sigma}_{tr,g}^{r,1}\right) e^{-\Sigma_{tr,g}^{r,0} s} - \left(\hat{\Sigma}_{tr,g}^{r,2} - \Sigma_{tr,g}^{r,0}\right) e^{-\hat{\Sigma}_{tr,g}^{r,1} s} + \left(\hat{\Sigma}_{tr,g}^{r,1} - \Sigma_{tr,g}^{r,0}\right) e^{-\hat{\Sigma}_{tr,g}^{r,2} s}}{\left(\hat{\Sigma}_{tr,g}^{r,2} - \hat{\Sigma}_{tr,g}^{r,1}\right)\left(\hat{\Sigma}_{tr,g}^{r,2} - \Sigma_{tr,g}^{r,0}\right)\left(\hat{\Sigma}_{tr,g}^{r,1} - \Sigma_{tr,g}^{r,0}\right)}.
\end{gathered} \tag{C.12}$$

The "exact" solution for the angular flux for the second time step is substantially more complex than the solution for the first time step, and it is clear that a pattern is emerging. In addition, this solution requires storing the same data as the previous time step, plus the following data for Δt_1: the segment-wise incoming angular fluxes, the region-wise cross sections, the region-wise neutron sources, and Δt_1.

C.5 Solution of Time-Dependent MOC for Time Step N

If we repeat this process for additional time steps, we can derive the general form of the segment-wise angular flux for an arbitrary time step N. The solution is:

$$\begin{aligned}\varphi_g^{k,N}(s) = \varphi_{g,0}^{k,N} e^{-\hat{\Sigma}_{tr,g}^{r,N} s} + \frac{\hat{q}_g^{r,N}}{\hat{\Sigma}_{tr,g}^{r,N}}\left(1 - e^{-\hat{\Sigma}_{tr,g}^{r,N} s}\right) \\ + \sum_{i=0}^{N-1}\left[A_{g,i}^{k,r} B_{g,N-1,i}^{k,r}(s) \prod_{j=N-i}^{N}\left(\Sigma_{\Delta t,g}^{j}\right)\right],\end{aligned} \tag{C.13}$$

where:

$$\begin{aligned} A_{g,i}^{k,r} &\equiv \varphi_{g,0}^{k,i} - \frac{\hat{q}_g^{r,i}}{\hat{\Sigma}_{tr,g}^{r,i}}, \\ B_{g,N-1,i}^{k,r}(s) &\equiv \frac{\sum_{j=i}^{N}\left([-1]^{j-i} e^{-\hat{\Sigma}_{tr,g}^{r,N} s} \prod_{\substack{l=i+1 \\ l\neq j}}^{N}\left[\prod_{\substack{m=i \\ m\neq l}}^{l-1}\left(\hat{\Sigma}_{tr,g}^{r,l} - \hat{\Sigma}_{tr,g}^{r,m}\right)\right]\right)}{\prod_{p=i+1}^{N}\left[\prod_{q=i}^{p-1}\left(\hat{\Sigma}_{tr,g}^{r,p} - \hat{\Sigma}_{tr,g}^{r,q}\right)\right]}, \\ \Sigma_{\Delta t,g}^{n} &\equiv \frac{1}{v_g \Delta t_n}, \\ \hat{\Sigma}_{tr,g}^{r,n} &\equiv \Sigma_{tr,g}^{r,n} + \Sigma_{\Delta t,g}^{n}, \\ \hat{q}_g^{r,n} &\equiv q_g^{r,n} + \Sigma_{\Delta t,g}^{n} \frac{\hat{q}_g^{r,n-1}}{\hat{\Sigma}_{tr,g}^{r,n-1}}. \end{aligned} \tag{C.14}$$

This illustrates the trend that was observed for the first two time steps: for every additional time step that is modeled, it is necessary to store another segment-wise incoming angular flux. Consequently, exactly representing the spatial dependence of the previous angular fluxes for time-dependent MOC is not practical for large problems because the memory requirements are large and increase linearly.

C.6 Summary

In this appendix we provided a derivation for the spatially-dependent angular flux for time-dependent MOC using Backward Euler without approximating the spatial dependence of the angular flux along the segment. This derivation illustrates that the memory requirements for such a method increase linearly with the number of time steps modeled. As a result, it is a practical necessity to approximate the spatial dependence of the angular flux for the time derivative in time-dependent MOC.

Appendix D

Macroscopic Cross Sections for C5G7 Transient

This appendix provides the macroscopic cross sections for the materials in the C5G7 transient [Lew01]. The cross sections are provided in units of cm^{-1} in the tables below.

Table D-1. UO$_2$ fuel-clad macroscopic cross sections

	$\Sigma_{tr,g}$	$\Sigma_{a,g}$	$\Sigma_{c,g}$	$\Sigma_{f,g}$	ν_g	χ_g
Group 1	1.7795E-01	8.0248E-03	8.1274E-04	7.2121E-03	2.7815E+00	5.8791E-01
Group 2	3.2981E-01	3.7174E-03	2.8981E-03	8.1930E-04	2.4744E+00	4.1176E-01
Group 3	4.8039E-01	2.6769E-02	2.0316E-02	6.4532E-03	2.4338E+00	3.3906E-04
Group 4	5.5437E-01	9.6236E-02	7.7671E-02	1.8565E-02	2.4338E+00	1.1761E-07
Group 5	3.1180E-01	3.0020E-02	1.2212E-02	1.7808E-02	2.4338E+00	0.0000E+00
Group 6	3.9517E-01	1.1126E-01	2.8225E-02	8.3035E-02	2.4338E+00	0.0000E+00
Group 7	5.6441E-01	2.8278E-01	6.6776E-02	2.1600E-01	2.4338E+00	0.0000E+00

$\Sigma_{s,g\prime\to g}$	To Group 1	To Group 2	To Group 3	To Group 4	To Group 5	To Group 6	To Group 7
Group 1	1.2754E-01	4.2378E-02	9.4374E-06	5.5163E-09	0.0000E+00	0.0000E+00	0.0000E+00
Group 2	0.0000E+00	3.2446E-01	1.6314E-03	3.1427E-09	0.0000E+00	0.0000E+00	0.0000E+00
Group 3	0.0000E+00	0.0000E+00	4.5094E-01	2.6792E-03	0.0000E+00	0.0000E+00	0.0000E+00
Group 4	0.0000E+00	0.0000E+00	0.0000E+00	4.5257E-01	5.5664E-03	0.0000E+00	0.0000E+00
Group 5	0.0000E+00	0.0000E+00	0.0000E+00	1.2525E-04	2.7140E-01	1.0255E-02	1.0021E-08
Group 6	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	1.2968E-03	2.6580E-01	1.6809E-02
Group 7	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8.5458E-03	2.7308E-01

Table D-2. 4.3% MOX fuel-clad macroscopic cross sections

	$\Sigma_{tr,g}$	$\Sigma_{a,g}$	$\Sigma_{c,g}$	$\Sigma_{f,g}$	ν_g	χ_g
Group 1	1.7873E-01	8.4339E-03	8.0686E-04	7.6270E-03	2.8521E+00	5.8791E-01
Group 2	3.3085E-01	3.7577E-03	2.8808E-03	8.7690E-04	2.8910E+00	4.1176E-01
Group 3	4.8377E-01	2.7970E-02	2.2272E-02	5.6984E-03	2.8549E+00	3.3906E-04
Group 4	5.6692E-01	1.0421E-01	8.1323E-02	2.2887E-02	2.8607E+00	1.1761E-07
Group 5	4.2623E-01	1.3994E-01	1.2918E-01	1.0764E-02	2.8545E+00	0.0000E+00
Group 6	6.7900E-01	4.0918E-01	1.7642E-01	2.3276E-01	2.8642E+00	0.0000E+00
Group 7	6.8285E-01	4.0935E-01	1.6038E-01	2.4897E-01	2.8678E+00	0.0000E+00

$\Sigma_{s,g\prime\to g}$	To Group 1	To Group 2	To Group 3	To Group 4	To Group 5	To Group 6	To Group 7
Group 1	1.2888E-01	4.1413E-02	8.2290E-06	5.0405E-09	0.0000E+00	0.0000E+00	0.0000E+00
Group 2	0.0000E+00	3.2545E-01	1.6395E-03	1.5982E-09	0.0000E+00	0.0000E+00	0.0000E+00
Group 3	0.0000E+00	0.0000E+00	4.5319E-01	2.6142E-03	0.0000E+00	0.0000E+00	0.0000E+00
Group 4	0.0000E+00	0.0000E+00	0.0000E+00	4.5717E-01	5.5394E-03	0.0000E+00	0.0000E+00
Group 5	0.0000E+00	0.0000E+00	0.0000E+00	1.6046E-04	2.7681E-01	9.3127E-03	9.1656E-09
Group 6	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	2.0051E-03	2.5296E-01	1.4850E-02
Group 7	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8.4948E-03	2.6501E-01

Table D-3. 7.0% MOX fuel-clad macroscopic cross sections

	$\Sigma_{tr,g}$	$\Sigma_{a,g}$	$\Sigma_{c,g}$	$\Sigma_{f,g}$	ν_g	χ_g
Group 1	1.8132E-01	9.0657E-03	8.1124E-04	8.2545E-03	2.8850E+00	5.8791E-01
Group 2	3.3437E-01	4.2967E-03	2.9711E-03	1.3257E-03	2.9108E+00	4.1176E-01
Group 3	4.9379E-01	3.2881E-02	2.4459E-02	8.4216E-03	2.8657E+00	3.3906E-04
Group 4	5.9122E-01	1.2203E-01	8.9157E-02	3.2873E-02	2.8706E+00	1.1761E-07
Group 5	4.7420E-01	1.8298E-01	1.6702E-01	1.5964E-02	2.8671E+00	0.0000E+00
Group 6	8.3360E-01	5.6846E-01	2.4467E-01	3.2379E-01	2.8666E+00	0.0000E+00
Group 7	8.5360E-01	5.8521E-01	2.2241E-01	3.6280E-01	2.8754E+00	0.0000E+00

$\Sigma_{s,g\prime\to g}$	To Group 1	To Group 2	To Group 3	To Group 4	To Group 5	To Group 6	To Group 7
Group 1	1.3046E-01	4.1792E-02	8.5105E-06	5.1329E-09	0.0000E+00	0.0000E+00	0.0000E+00
Group 2	0.0000E+00	3.2843E-01	1.6436E-03	2.2017E-09	0.0000E+00	0.0000E+00	0.0000E+00
Group 3	0.0000E+00	0.0000E+00	4.5837E-01	2.5331E-03	0.0000E+00	0.0000E+00	0.0000E+00
Group 4	0.0000E+00	0.0000E+00	0.0000E+00	4.6371E-01	5.4766E-03	0.0000E+00	0.0000E+00
Group 5	0.0000E+00	0.0000E+00	0.0000E+00	1.7619E-04	2.8231E-01	8.7289E-03	9.0016E-09
Group 6	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	2.2760E-03	2.4975E-01	1.3114E-02
Group 7	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8.8645E-03	2.5953E-01

Table D-4. 8.7% MOX fuel-clad macroscopic cross sections

	$\Sigma_{tr,g}$	$\Sigma_{a,g}$	$\Sigma_{c,g}$	$\Sigma_{f,g}$	ν_g	χ_g
Group 1	1.8305E-01	9.4862E-03	8.1411E-04	8.6721E-03	2.9043E+00	5.8791E-01
Group 2	3.3671E-01	4.6556E-03	3.0313E-03	1.6243E-03	2.9180E+00	4.1176E-01
Group 3	5.0051E-01	3.6240E-02	2.5968E-02	1.0272E-02	2.8699E+00	3.3906E-04
Group 4	6.0617E-01	1.3272E-01	9.3675E-02	3.9045E-02	2.8749E+00	1.1761E-07
Group 5	5.0275E-01	2.0840E-01	1.8914E-01	1.9258E-02	2.8718E+00	0.0000E+00
Group 6	9.2103E-01	6.5870E-01	2.8381E-01	3.7489E-01	2.8675E+00	0.0000E+00
Group 7	9.5523E-01	6.9017E-01	2.5957E-01	4.3060E-01	2.8781E+00	0.0000E+00

$\Sigma_{s,g\prime\to g}$	To Group 1	To Group 2	To Group 3	To Group 4	To Group 5	To Group 6	To Group 7
Group 1	1.3150E-01	4.2046E-02	8.6972E-06	5.1938E-09	0.0000E+00	0.0000E+00	0.0000E+00
Group 2	0.0000E+00	3.3040E-01	1.6463E-03	2.6006E-09	0.0000E+00	0.0000E+00	0.0000E+00
Group 3	0.0000E+00	0.0000E+00	4.6179E-01	2.4749E-03	0.0000E+00	0.0000E+00	0.0000E+00
Group 4	0.0000E+00	0.0000E+00	0.0000E+00	4.6802E-01	5.4330E-03	0.0000E+00	0.0000E+00
Group 5	0.0000E+00	0.0000E+00	0.0000E+00	1.8597E-04	2.8577E-01	8.3973E-03	8.9280E-09
Group 6	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	2.3916E-03	2.4761E-01	1.2322E-02
Group 7	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8.9681E-03	2.5609E-01

Table D-5. Fission chamber macroscopic cross sections

	$\Sigma_{tr,g}$	$\Sigma_{a,g}$	$\Sigma_{c,g}$	$\Sigma_{f,g}$	ν_g	χ_g
Group 1	1.2603E-01	5.1132E-04	5.1132E-04	4.7900E-09	2.7628E+00	5.8791E-01
Group 2	2.9316E-01	7.5813E-05	7.5807E-05	5.8256E-09	2.4624E+00	4.1176E-01
Group 3	2.8425E-01	3.1643E-04	3.1597E-04	4.6372E-07	2.4338E+00	3.3906E-04
Group 4	2.8102E-01	1.1675E-03	1.1623E-03	5.2441E-06	2.4338E+00	1.1761E-07
Group 5	3.3446E-01	3.3977E-03	3.3976E-03	1.4539E-07	2.4338E+00	0.0000E+00
Group 6	5.6564E-01	9.1886E-03	9.1879E-03	7.1497E-07	2.4338E+00	0.0000E+00
Group 7	1.1721E+00	2.3244E-02	2.3242E-02	2.0804E-06	2.4338E+00	0.0000E+00

$\Sigma_{s,g\prime\to g}$	To Group 1	To Group 2	To Group 3	To Group 4	To Group 5	To Group 6	To Group 7
Group 1	6.6166E-02	5.9070E-02	2.8334E-04	1.4622E-06	2.0642E-08	0.0000E+00	0.0000E+00
Group 2	0.0000E+00	2.4038E-01	5.2435E-02	2.4990E-04	1.9239E-05	2.9875E-06	4.2140E-07
Group 3	0.0000E+00	0.0000E+00	1.8343E-01	9.2288E-02	6.9365E-03	1.0790E-03	2.0543E-04
Group 4	0.0000E+00	0.0000E+00	0.0000E+00	7.9077E-02	1.6999E-01	2.5860E-02	4.9256E-03
Group 5	0.0000E+00	0.0000E+00	0.0000E+00	3.7340E-05	9.9757E-02	2.0679E-01	2.4478E-02
Group 6	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	9.1742E-04	3.1677E-01	2.3876E-01
Group 7	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	4.9793E-02	1.0991E+00

Table D-6. Guide tube macroscopic cross sections

	$\Sigma_{tr,g}$	$\Sigma_{a,g}$	$\Sigma_{c,g}$
Group 1	1.2603E-01	5.1132E-04	5.1132E-04
Group 2	2.9316E-01	7.5801E-05	7.5801E-05
Group 3	2.8424E-01	3.1572E-04	3.1572E-04
Group 4	2.8096E-01	1.1582E-03	1.1582E-03
Group 5	3.3444E-01	3.3975E-03	3.3975E-03
Group 6	5.6564E-01	9.1878E-03	9.1878E-03
Group 7	1.1722E+00	2.3242E-02	2.3242E-02

$\Sigma_{s,g'\to g}$	To Group 1	To Group 2	To Group 3	To Group 4	To Group 5	To Group 6	To Group 7
Group 1	6.6166E-02	5.9070E-02	2.8334E-04	1.4622E-06	2.0642E-08	0.0000E+00	0.0000E+00
Group 2	0.0000E+00	2.4038E-01	5.2435E-02	2.4990E-04	1.9239E-05	2.9875E-06	4.2140E-07
Group 3	0.0000E+00	0.0000E+00	1.8330E-01	9.2397E-02	6.9446E-03	1.0803E-03	2.0567E-04
Group 4	0.0000E+00	0.0000E+00	0.0000E+00	7.8851E-02	1.7014E-01	2.5881E-02	4.9297E-03
Group 5	0.0000E+00	0.0000E+00	0.0000E+00	3.7333E-05	9.9737E-02	2.0679E-01	2.4478E-02
Group 6	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	9.1726E-04	3.1677E-01	2.3877E-01
Group 7	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	4.9792E-02	1.0991E+00

Table D-7. Moderator macroscopic cross sections

	$\Sigma_{tr,g}$	$\Sigma_{a,g}$	$\Sigma_{c,g}$
Group 1	1.5921E-01	6.0105E-04	6.0105E-04
Group 2	4.1297E-01	1.5793E-05	1.5793E-05
Group 3	5.9031E-01	3.3716E-04	3.3716E-04
Group 4	5.8435E-01	1.9406E-03	1.9406E-03
Group 5	7.1800E-01	5.7416E-03	5.7416E-03
Group 6	1.2545E+00	1.5001E-02	1.5001E-02
Group 7	2.6504E+00	3.7239E-02	3.7239E-02

$\Sigma_{s,g'\to g}$	To Group 1	To Group 2	To Group 3	To Group 4	To Group 5	To Group 6	To Group 7
Group 1	4.4478E-02	1.1340E-01	7.2347E-04	3.7499E-06	5.3184E-08	0.0000E+00	0.0000E+00
Group 2	0.0000E+00	2.8233E-01	1.2994E-01	6.2340E-04	4.8002E-05	7.4486E-06	1.0455E-06
Group 3	0.0000E+00	0.0000E+00	3.4526E-01	2.2457E-01	1.6999E-02	2.6443E-03	5.0344E-04
Group 4	0.0000E+00	0.0000E+00	0.0000E+00	9.1028E-02	4.1551E-01	6.3732E-02	1.2139E-02
Group 5	0.0000E+00	0.0000E+00	0.0000E+00	7.1437E-05	1.3914E-01	5.1182E-01	6.1229E-02
Group 6	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	2.2157E-03	6.9991E-01	5.3732E-01
Group 7	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	1.3244E-01	2.4807E+00

Table D-8. Control rod macroscopic cross sections

	$\Sigma_{tr,g}$	$\Sigma_{a,g}$	$\Sigma_{c,g}$
Group 1	1.3511E-01	6.3068E-04	6.3068E-04
Group 2	3.1185E-01	9.0445E-04	9.0445E-04
Group 3	3.4445E-01	8.6632E-03	8.6632E-03
Group 4	3.4987E-01	4.0822E-02	4.0822E-02
Group 5	3.9204E-01	7.2934E-02	7.2934E-02
Group 6	6.2285E-01	1.0122E-01	1.0122E-01
Group 7	1.2390E+00	1.3875E-01	1.3875E-01

$\Sigma_{s,g\prime\to g}$	To Group 1	To Group 2	To Group 3	To Group 4	To Group 5	To Group 6	To Group 7
Group 1	7.6606E-02	5.7603E-02	2.6484E-04	1.3288E-06	1.8578E-08	0.0000E+00	0.0000E+00
Group 2	0.0000E+00	2.6344E-01	4.7260E-02	2.2491E-04	1.7315E-05	2.6888E-06	3.7926E-07
Group 3	0.0000E+00	0.0000E+00	2.4515E-01	8.3229E-02	6.2501E-03	9.7227E-04	1.8510E-04
Group 4	0.0000E+00	0.0000E+00	0.0000E+00	1.2804E-01	1.5327E-01	2.3293E-02	4.4367E-03
Group 5	0.0000E+00	0.0000E+00	0.0000E+00	4.0155E-05	1.1055E-01	1.8649E-01	2.2030E-02
Group 6	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	9.2796E-04	3.0534E-01	2.1537E-01
Group 7	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	4.5166E-02	1.0551E+00

References

[Abo08] A. E. ABOANBER and Y. M. HAMADA, "Generalized Runge-Kutta Method for Two- and Three-dimensional Space-Time Diffusion Equation," *Ann. Nucl. Eng.*, **35**, 1024-1040 (2008).

[Alc11] R. E. ALCOUFFE et al, "PARTISN: A Time-Dependent, Parallel Neutral Particle Transport Code System," LA-UR-08-7258, Los Alamos National Laboratory (2011).

[ANL05] "The Numerical Nuclear Reactor for High Fidelity Integrated Simulation of Neutronic, Thermal-Hydraulic and Thermo-Mechanical Phenomena, Final Report," International Nuclear Energy Research Initiative, Project Number 2002-010-K (2005).

[Ask72] J. R. ASKEW, "A Characteristics Formulation of the Neutron Transport Equation in Complicated Geometries," AAEW-M 1108, United Kingdom Atomic Energy Authority (1972).

[Ban12] J. E. BANFIELD et al., "A New Semi-Implicit Direct Kinetics Method with Analytical Representation of Delayed Neutrons," *Trans. Am. Nucl. Soc.*, **107**, 1111-1114 (2012).

[Bar12] A. BARBARINO et al., "On the Evaluation of Ray Effects in Multidimensional and Time-Dependent Transport Problems," *Trans. Am. Nucl. Soc.*, **106**, 369-371 (2012).

[Bor09] M. BOROUSHAKI, "Numerical Solution of the Neutron Transport Equation using Cellular Neural Networks," *Ann. Nuc. Energy*, **36**, 15-27 (2009).

[Bru05] T. A. BRUNNER and J. P. HOLLOWAY, "Two Dimensional Time Dependent Riemann Solvers for Neutron Transport," *J. Comput. Phys.*, **210**, 386 (2005).

[But08] J. C. BUTCHER, "Numerical Methods for Ordinary Differential Equations," John Wiley & Sons (2008).

[Cho05] J. Y. CHO et al., "Transient Capability for a MOC-Based Whole Core Transport Code DeCART," *Trans. Am. Nucl. Soc.*, **92**, 721 (2005).

[Col13] B. COLLINS et al., "Verification of MPACT: Michigan Parallel Characteristic Transport Code," *Trans. Am. Nucl. Soc.*, **108**, 795-798 (2013).

[Cro96] N. CROUZET and P. TURINSKY, "A Second-Derivative-Based Adaptive Time-Step Method for Spatial Kinetics Calculation," *Nucl. Sci. Eng.*, **123**, 206 (1996).

[Gar05] V. M. GARCIA et al., "Parallel Resolution of the Two-Group Time Dependent Neutron Diffusion Equation with Public Domain ODE Codes," *Lecture Note in Computer Science*, **3402**, 457-497 (2005).

[Gel61] E. M. GELBERT, "Application of Spherical Harmonics Method to Reactor Problems," WAPD-BT-20 (1960).

[Gin98] D. GINESTAR et al., "High Order Backward Discretization of the Neutron Diffusion Equation," *Ann. Nucl. Energy*, **25**, 47-64 (1998).

[Gol01] S. GOLUOGLU and H. L. DODDS, "A Time-Dependent, Three-Dimensional Neutron Transport Methodology," *Nucl. Sci. Eng.*, **139**, 248-261 (2001).

[Hal80] M. J. HALSALL, "Cactus, A Characteristic Solution to the Neutron Transport Equations in Complicated Geometries," AEEW-R-1291, United Kingdom Atomic Energy Authority (1980).

[Hof13a] A. J. HOFFMAN and J. C. LEE, "A Variable-Order Time-Dependent Neutron Transport Method for Nuclear Reactor Kinetics using Analytically-Integrated Space-Time Characteristics," *Proc. Intl. Conf. on Math. and Comput. Methods Applied to Nucl. Sci. Eng. (M&C2013)*, Sun Valley, USA, May 5-9 (2013).

[Hof13b] A. J. HOFFMAN and J. C. LEE, “Low-Order Approximations to the Angular Flux Time Derivative for Transport-based Reactor Kinetics,” *Trans. Am. Nucl. Soc.*, **108,** 777-780 (2013).

[Hon99] S. G. HONG and N. Z. CHO, “Method of Characteristic Direction Probabilities for Heterogeneous Lattice Calculation,” *Nucl. Sci. Eng.*, **132**, 65-77 (1999).

[Hur08] M. HURSIN et al., “DeCART v2.05 Theory Manual,” (2008).

[Joo98] H. G. JOO et al., “PARCS: Purdue Advanced Reactor Core Simulator,” PU/NE-98-26, Purdue University (1998).

[Joo04] H. G. JOO et al., “Methods and Performance of a Three-dimensional Whole-core Transport Code DeCART,” *Proc. PHYSOR2004*, Chicago, USA (2004).

[Jon00] D. B. JONES et al., “CPM-3 Computer Code Manual,” EPRI Project EP-P1295/C507 (2000).

[Kel98] P. M. KELLER and J. C. LEE, “A Time-Dependent Collision Probability Method for One-Dimensional Space-Time Nuclear Reactor Kinetics,” *Nucl. Sci. Eng.*, **129**, 124-128 (1998).

[Lew01] E. LEWIS et al., “Benchmark Specification for Deterministic 2D/3D MOX Fuel Assembly Transport Calculations without Spatial Homogenization,” Nuclear Energy Agency (2001).

[Liu13] Z. LIU et al., “The Method of Modular Characteristic Direction Probabilities in MPACT,” *Proc. Intl. Conf. on Math. and Comput. Methods Applied to Nucl. Sci. Eng. (M&C2013)*, Sun Valley, USA, May 5-9 (2013).

[Lou99] S. LOUBIERE et al., “APOLLO2 Twelve Years Later,” *Proc. Intl. Conf. on Math. and Comput. Reactor Phys. and Envir. Analysis in Nucl. App.*, Madrid, Spain, (1999).

[McC07a] R. G. MCCLARREN and J. P. HOLLOWAY, “A Quasilinear Implicit Riemann Solver for the Time-Dependent PN Equations,” *Nucl. Sci. Eng.*, **155**, 290-299 (2007).

[McC07b] R. G. MCCLARREN et al., "Semi-Implicit Time Integration for the Pn Equations," *Trans. Am. Nucl. Soc.*, **97**, 527-529 (2007).

[NEA03] "Benchmark on Deterministic Transport Calculations without Spatial Homogenization," NEA/NSC/DOC(2003)16 (2003).

[Pan09] T. M. PANDYA and M. L. ADAMS, "Method of Long Characteristics Applied in Space and Time," *Proc. Intl. Conf. on Math. and Comput. Methods Applied to Nucl. Sci. Eng. (M&C2009)*, Saratoga Springs, USA, May 3-7 (2009).

[Pan11] T. M. PANDYA, et al., "Long Characteristics with Piecewise Linear Sources Designed for Unstructured Grids," *Proc. Intl. Conf. on Math. and Comput. Methods Applied to Nucl. Sci. Eng. (M&C2011)*, Rio de Janeiro, Brazil, May 8-12 (2012).

[Pau03] A. PAUTZ and A. BIRKHOFER, "DORT-TD: A Transient Neutron Transport Code with Fully Implicit Time Integration," *Nucl. Sci. Eng.*, **145**, 299-319 (2003).

[Rin97] A. RINIEISKI et al., "Time-Dependent Neutron Transport with Variational Nodal Methods," *Proc. Joint. Intl. Conf. Math. Methods and Supercomput. for Nucl. Appl.*, Saratoga Springs, USA (1997).

[Rod09] Z. RODRIGUES DE LIMA et al., "A Modal Multidimensional Kinetics Method using Pseudo-Harmonics," *Ann. Nucl. Energy*, **36**, 752-759 (2009).

[Seu11] A. SEUBERT, "The Time-Dependent 3-D Transport Code TORT-TD: Recent Advances and Applications," *Trans. Am. Nucl. Soc.*, **104**, 879-882 (2011).

[Shi11] C. B. SHIM, et al., "Application of Backward Differentiation Formula to Spatial Reactor Kinetics Calculation with Adaptive Time Step Control," *Nucl. Eng. Tech.*, **43**, 531-546 (2011).

[Smi97] K. S. SMITH, "Multidimensional Nodal Transport Using the Simplified PL Method," *Proc. Joint. Intl. Conf. Math. Methods and Supercomput. for Nucl. Appl.*, Saratoga Springs, USA (1997).

[Smi00] K. S. SMITH and J. D. RHODES, "CASMO-4 Characteristics Method for Two-Dimensional PWR and BWR Core Calculations," *Trans. Am. Nucl. Soc.*, **83**, 294 (2000).

[Sut96] T. M. SUTTON and B. N. AVILES, "Diffusion Theory Methods for Spatial Kinetics Calculations," *Prog. Nucl. Energy*, **30**, 119-182 (1996).

[Tal13] A. TALAMO, "Numerical Solution of the Time Dependent Neutron Transport Equation by Method of Characteristics," *J. Comput. Phys.*, **240**, 248-267 (2013).

[Tay09] J. B. TAYLOR and A. J. BARATTA, "A Time-Dependent Method of Characteristics for 3D Nuclear Reactor Kinetics Applications," *Proc. Intl. Conf. on Math. and Comput. Methods Applied to Nucl. Sci. Eng. (M&C2009)*, Saratoga Springs, USA, May 3-7 (2009).

[Tsu12] K. TSUJITA, et al., "Kinetic Calculation Method in Space-Time Frame using Characteristic Line," *Trans. Am. Nucl. Soc.*, **106**, 743-746 (2012).

[Tsu13] K. TSUJITA, et al., "Higher Order Treatment on Temporal Derivative of Angular Flux for Time-Dependent MOC," *Proc. Intl. Conf. on Math. and Comput. Methods Applied to Nucl. Sci. Eng. (M&C2013)*, Sun Valley, USA, May 5-9 (2013).

[Yan10] X. YANG and T. JEVREMOVIC, "Spatial and Time-Dependent Reactor Kinetics Methodology Based on the Method of Characteristics," *PHYSOR2010*, Pittsburgh, USA, (2010).

[Yas65] J. B. YASINSKY and A. F. HENRY, "Some Numerical Experiments Concerning Space-Time Reactor Kinetics Behavior," *Nucl. Sci. Eng.*, **22**, 171 (1965).

[Zer11] R. J. ZERR and R. S. BAKER, "On the Matter of Time-Dependent Ray Effects," *Trans. Am. Nucl. Soc.*, **105**, 438-439 (2011).