A Robust Second-Order Multiple Balance Method and α-Weighted Multigroup Constants for Time-Dependent Nuclear Reactor Simulations

by

Ilham Variansyah

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
2021

Doctoral Committee:

Professor William R. Martin, Co-Chair
Professor Thomas J. Downar, Co-Chair
Dr. Benjamin R. Betzler, Oak Ridge National Laboratory
Professor Robert Krasny
Professor Edward W. Larsen

Ilham Juanda

Ilham Variansyah

ilhamv@umich.edu

ORCID ID: 0000-0003-3426-7160

To my parents,
for their unparalleled love, and
for teaching me the value and joy of intellectual pursuits

To my beloved wife, Hana Khairunnisa,
whom I cannot thank enough for loving me back

And to my son, Abimana,
who patiently waited every day to play with me
when I worked on this dissertation

ACKNOWLEDGEMENTS

First and foremost, I would like to thank Prof. William Martin, my advisor, for *everything*: for taking me as his student despite the problematic situation that I was in, for his profound wisdom, guidance, patience, support, and more. Prof. Martin, I truly appreciate all you have done for me.

Second, I would like to thank my committee for their invaluable support. To Prof. Thomas Downar, for co-chairing and engaging me to further explore potential applications of the methods I developed. To Prof. Edward Larsen, for helping me develop the MBTD method since day one; his perspective has shifted mine and led me to this work. I truly enjoyed all of the short and hours-long discussions with him, and I genuinely look up to his enthusiasm for researching and teaching. To Dr. Benjamin Betzler, for being an exemplary mentor since my summer internship at Oak Ridge National Laboratory and throughout my doctoral study, and for helping me in exploring applications of and methods for solving the α-eigenvalue neutron transport problems. To Prof. Robert Krasny, for serving in my committee, providing his profound mathematical expertise, further shaping the MBTD method that I developed.

I would also like to express gratitude to (1) the Consortium for Advanced Simulation of Light Water Reactors (www.casl.gov), an Energy Innovation Hub (http://www.energy.gov/hubs) for Modeling and Simulation of Nuclear Reactors, under U.S. Department of Energy Contract No. DE-AC05-00OR22725, and (2) the U.S. Department of Energy, through the Nuclear Energy University Program (Award No. NEUP-IRP-15-8761), for supporting and funding my researches during my doctoral study. And, to the Department of Nuclear Engineering and Radiological Sciences and the College of Engineering, for supporting me through the teaching opportunities and generous awards and fellowships. Particularly, I would like to thank Dr. Frederick W. and Marion K. Buckman, the generous donors of the John C. Lee Fellowship.

My graduate study journey might not be as "smooth" and might not have happened at all without the support of and the following. I would like to thank Dr. Andang Widi Harto, my advisor during my bachelor's study at Universitas Gadjah Mada, for his inspirational teachings and lecture notes/books in Advanced Reactor Technology and Nuclear Cogeneration, which had sparked my enthusiasm for nuclear energy technology. I would like to express gratitude to Lembaga Pengelola Dana Pendidikan (LPDP, or Indonesia Endowment Fund for Education) for making studying abroad a reality for me. LPDP fully supported my Master's study here, which had opened up doors to many great opportunities that had led me up to this point, and beyond. I would also like to thank Dr. Topan Setiadipura, who voluntarily took me as his mentee at BATAN (Indonesia’s National Nuclear Energy Agency) during my transition to graduate study; his mentorship on nuclear reactor computation could not have been more timely.

Finally, and most importantly, I would like to thank my wife, Hana Khairunnisa. Hana, thank you for being with me, going all the way to the other side of the earth, being away from your family and friends, leaving all dreams that you might have had, just so I could achieve mine. Thank you so much for your patience, unwavering support, and loving encouragement whenever I am feeling down. You are essential to my life, and to everything I accomplish. Also, Abim, thank you so much for coming to this world; you are the most precious gift I have ever had. And thank you for patiently waiting to play with me whenever I work overtime. Hana and Abim, you flood my life with joy and color my days with happiness. You are at the core of my motivation. Being with you has taught me that one could push beyond the limit and strive for the loved ones. I cannot express in words how grateful and how lucky I am to have you in my life.

Ilham, Hana, and Abim

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

α-MGXS	α-Weighted Multigroup Constants
BC	Boundary Condition
BE	Backward Euler
CN	Crank-Nicholson
DD	Diamon-Difference Method
DMD	Domain Mode Decomposition
DSA	Diffusion Synthetic Acceleration
FA	Fourier Analysis
FDM	Finite Difference Method
FE	Forward Euler
GMRES	Generalized Minimal Residual
GS	Gauss-Seidel
ILU	Incomplete LU Factorization
JFNK	Jacobian-Free Newton Krylov
k-MGXS	k-Weighted Multigroup Constants
LHS	Left-hand-side
MB	Multiple Balance
MBTD	Multiple Balance Time-Discretization
MGXS	Multigroup Constants
MOC	Method of Characteristics
NR	Newton Raphson
ODE	Ordinary Differential Equation
PCQM	Predictor-Corrector Quasi-Static Method
PDE	Partial Differential Equation
RHS	Right-hand-side

RK	Runge-Kutta
SI	Source Iteration
SN	Discrete Ordinate Method
SOR	Successive Over-Relaxation
TML	Transient Multi-Level
TR-BDF2	Trapezoidal/Second-Order Backward Difference Formula
TRMM	Transition Rate Matrix Method

ABSTRACT

Time-dependent nuclear reactor simulations are essential in improving the safety, effectiveness, and efficiency of nuclear reactor designs, experiments, and operations. This thesis proposes, implements, and tests two new methods designed to improve two different aspects of time-dependent reactor simulation: (1) *Multiple Balance Time-Discretization* (MBTD), a robust second-order accurate time-stepping method, an alternative to the highly reliable Backward Euler (BE); and (2) *α-Weighted Multigroup Constants* ($\boldsymbol{\alpha}$-MGXS), an alternative formulation of multigroup constants that offers advantages over the traditionally-used k-Weighted Multigroup Constants ($\boldsymbol{k}$-MGXS) for time-dependent neutron transport simulations.

Despite being only first-order accurate, BE has been the primary time-discretization method in reactor simulations due to its simplicity and robustness (unconditionally stable and free of spurious oscillations). The *Multiple Balance method* [Morel & Larsen 1990] was originally introduced as a spatial discretization for neutron transport methods. We show that its application to time-discretization (MBTD) yields a method that is not only robust like BE but also second-order accurate. MBTD consists of solving two coupled balance equations at each time step. In this thesis, three general strategies for solving these coupled equations are explored. MBTD adaptations are made for (1) the finite difference method (FDM) applied to the neutron diffusion equation and for (2) several techniques for the neutron transport equation, including Source Iteration (SI), applied to the Diamond-Difference (SN-DD) and Method of Characteristics (MOC). By exploiting the results of Fourier convergence analysis, an effective Diffusion Synthetic Acceleration (DSA) method for MBTD-SI is developed. Four representative kinetic problems are devised to test and assess the relative efficiency of MBTD versus BE. It is found that MBTD is about 2, 2.5, and 3 times computationally more expensive than BE for neutron diffusion with FDM, neutron transport DSA with SN-DD, and MOC, respectively, given the same uniform time-step size. However, due to its higher-order accuracy, MBTD is generally more efficient than BE: a larger time step can be used to achieve a certain accuracy. Finally, a similar trend is observed in a neutronics/thermal-hydraulics tight-coupling multi-physics application, where MBTD is more efficient than BE for reasonably accurate simulations (relative error less than ~10%).

Multigroup neutron transport methods remain as essential tools for reactor simulations, but their accuracy can only be as good as their multigroup constants (MGXS). Estimation of MGXS is traditionally based on the solution of the k-eigenvalue neutron transport calculation. However, the k-eigenfunction is not physically representative for systems that are far from critical, which is the case in many reactor transient simulations. Representing the asymptotic behavior of time-dependent transport problems, the α-eigenfunction may be a better alternative for the calculation of MGXS. In this thesis, physics-preserving MGXS for time-stepping methods are derived. A review of α-eigenvalue iteration methods is presented. A relaxed α-k Iteration developed to simulate the fundamental α-mode is implemented in the open-source Monte Carlo code OpenMC and verified with several benchmark problems. Results from four kinetics problems simulating absorber injection and removal to initially-critical infinite-medium fast and thermal systems emphasize that the fundamental α-eigenfunction—as a multigroup constant weighting spectrum—offers physical characteristics that make it advantageous (in producing accurate solutions) over the typically used fundamental k-eigenfunction.

CHAPTER 1

Introduction

The heart of a nuclear power plant or nonpower reactor is the nuclear reactor [1], where the neutron-induced nuclear fission chain reaction is maintained and controlled to generate energy or produce useful radiations and radioactive materials. Computational simulations of nuclear reactors [2] are essential in improving the safety, effectiveness, and efficiency of nuclear reactor designs, experiments, and operations. A reactor simulation essentially solves a mathematical model that may involve complex interactions between multiple physics in the reactor, including thermal-hydraulics, material structure, and chemistry. Nevertheless, the essence of a nuclear reactor simulation lies in the neutron transport, a key element of physics that keeps track of the free neutrons—the main actor of the fission chain reaction—in the reactor: how they are spatially distributed, how fast and in which direction they move, and how their population varies in time. Steady-state simulation is of primary interest to the nuclear reactor community because nuclear reactors are typically operated in steady-state mode. However, time-dependent simulation is important in assessing the performance of a reactor during transients and over its operational cycles.

This thesis proposes, implements, and tests two new methods to improve two different aspects of time-dependent reactor simulation. The two methods are (1) *Multiple Balance Time-Discretization* (*MBTD*), a robust second-order accurate time-discretization method, an alternative to the highly reliable *Backward Euler* (*BE*); and (2) *α-Weighted Multigroup Constants* ($\boldsymbol{\alpha}$-MGXS), an alternative formulation of multigroup constants that offers advantageous physical characteristics over the traditionally used *k-Weighted Multigroup Constants* ($\boldsymbol{k}$-MGXS) for time-dependent neutron transport simulation.

We note that all equation numbers, acronyms, table numbers, figure numbers, chapter/section numbers, and citations are hyperlinked to their source or definition for those viewing this document as an electronic PDF. Additionally, Python scripts used to generate and present numerical results in this thesis are provided in the author's Github repository (https://github.com/ilhamv/ilham_variansyah_dissertation)[3].

1.1 Time-Discretization and MBTD

In a time-dependent reactor simulation, we usually discretize the time dependence of the simulated physics. This time dependence is typically characterized by a first-order temporal derivative, which yields an initial value problem. Finite difference, or *time-stepping*, often becomes the preferred time-discretization method due to its simplicity, which allows the straightforward adaptation of existing steady-state simulation methods and computational programs. Even advanced time-discretization methods such as PCQM (Predictor-Corrector Quasi-Static Method) [4] and TML (Transient Multi-Level) [5] essentially employ some basic time-stepping methods in advancing a solution from one time-point to the next.

Standard, widely used finite time-stepping methods are the *Forward Euler* (*FE*), *Backward Euler* (*BE*), and *Trapezoid* (or *Crank-Nicholson*, *CN*) methods [6]. FE is simple but requires a very small time-step to guarantee stability; this makes it inappropriate for the *stiff* problems typically encountered in reactor dynamics. CN is favorable because of its higher (second) order of accuracy; however, it suffers from persisting spurious oscillations if the time step is insufficiently small. Despite being only first-order accurate, BE remains as the favorite method in practice [5][4] due to its *robustness*—being free from producing spurious oscillating solutions regardless of time-step size.

It is reasonable for a reactor analyst to choose the first-order accurate BE over the second-order accurate CN. BE's robustness is indispensable because spurious oscillations may yield unphysical solutions and may jeopardize the prediction of multi-physics feedback. CN indeed damps any occurring spurious oscillations in time; however, this damping is effective only if the time step is sufficiently small, and it is not always possible to predetermine if a time step is small enough. This argument is relevant even if we use an adaptive time-stepping technique. Most, if not all, adaptive time-stepping techniques are based on using information from solution estimates to determine the optimal time-step size to be used [6]. If the solution estimates are generated from a method that suffers from spurious oscillations, the calculated time-step size may not be as optimal as predicted. Therefore, even adaptive time-stepping techniques can benefit from a robust method such as BE.

Is a higher order of accuracy necessary? If yes, why stop at second-order accuracy? A higher-order method (third-order or higher) can always be formulated, theoretically at least. However, second-order methods typically achieve the right compromise between accuracy and complexity [7]. First-order methods are often not accurate enough. Third- and higher-order methods are more accurate, but they have issues associated with stability or being too complicated to implement. Therefore, despite all the reasoning for the robust BE, it is always tempting to use CN, especially because we can easily switch from BE to CN for "free". For example, in a computational program that adapts the θ*-Method* time-stepping formulation [8][4], we just need to change the parameter from $\theta = 1$ for BE into $\theta = 0.5$ for CN. There has always been a dilemma of choosing CN for accuracy or BE for robustness. Even though robustness often wins, it would be a significant advance if a method could be found that is not only robust (like BE) but also second-order accurate (like CN).

The *Multiple Balance* (*MB*) method was first proposed in 1990 as a novel spatial discretization for the discrete ordinate neutron transport method (SN) [9]. A simple analysis, verified with numerical experiments, shows that MB is second-order accurate and robust—it produces strictly non-negative, non-oscillatory solutions regardless of the spatial mesh size. MB spatial discretization has been further studied in [10] and [11]. However, MB application as time-discretization (MBTD) has not previously been investigated.

MBTD is not an entirely new idea [12]. It can be straightforwardly demonstrated that by applying MB formulation as a time-discretization to a simple initial value problem, one obtains a robust second-order accurate method. The problem is, when we implement MBTD to a practical problem, such as the time-dependent neutron transport equation, we obtain two coupled balance equations—each of which has similar complexity to the original balance equation—that must be solved simultaneously at each time step. It is unknown if there is an efficient strategy to solve the coupled balance equations so that the benefit of higher accuracy outweighs the additional computational complexity per time step. The first application of MBTD to time-dependent neutron transport was done by the author in [13].

In this thesis, the application of MBTD to time-dependent reactor simulation is investigated. In Chapter 2, the MBTD formulation and its basic features are introduced. Chapter 3, Chapter 4, and Chapter 5 respectively investigate the application of MBTD to neutron diffusion, neutron transport, and a typical multi-physics reactor problem. Each investigation is centered on comparing the formulation, computational complexity, theoretical convergence, and numerical results of the BE and MBTD methods. The objective is to assess the relative effectiveness of MBTD to BE, which will help in concluding whether MBTD is a worthy higher-order alternative to the robust method BE. Finally, conclusions are presented in Chapter 8, along with discussions of planned future work.

1.2 Multigroup Constants Calculation and α-MGXS

Accurately solving the time-dependent neutron transport equation is essential in a reactor transient simulation. The neutron transport equation can be solved using either a Monte Carlo or a deterministic method. The Monte Carlo method offers high-fidelity, continuous-energy, neutron transport simulation with minimal approximation. However, it suffers from stochastic uncertainties requiring expensive calculation of a large number of samples (neutron histories). Applications of the Monte Carlo method for solving the time-dependent neutron transport equation include those in the Monte Carlo code TART [14], Serpent [15], MCATK [16], TRIPOLI-4 [17], McCARD [18], and MCNP [19].

Deterministic methods continue to be essential tools in reactor transient simulations. Bounded with approximations, these methods give reasonably good solutions with less computational effort than the Monte Carlo method. Time-stepping methods, as discussed in the previous section, are often used in deterministic reactor transient simulations. In any of those time-stepping methods, the system properties (e.g., cross-sections) are discretized or homogenized over the phase space. These homogenized properties are known as the *multigroup constants*.

The multigroup constants should be appropriately weighted to preserve the underlying physics of the problem. There is a growing interest in using the Monte Carlo method as a means to generate the multigroup constants [20][21]. An approach typically performed in this hybrid methodology is to run a k-eigenvalue Monte Carlo simulation over the whole (or a portion) of the reactor core and use the resulting neutron flux distribution—the fundamental k-eigenfunction—as the weighting spectrum for the multigroup constants. However, the k-eigenvalue equation only represents the actual physical problem—i.e. time-dependent neutron transport—if the system is precisely critical. The fundamental k-eigenfunction makes a reasonable weighting spectrum for a system that is close to critical. However, this idea is challenged if the system is far from critical, which is often the situation in reactor transients.

The fundamental eigenfunction of the α-eigenvalue problem may be a better alternative weighting spectrum. This eigenfunction represents the asymptotic-in-time behavior of the system. This does not necessarily make the fundamental α-eigenfunction a superior weighting spectrum as it may not be important in the earlier time of a transient (as demonstrated by McClarren [22], for example). Nevertheless, the fundamental α-eigenfunction has a physical characteristic that is relevant to the actual time-dependent problem, something that the fundamental k-eigenfunction lacks. Several studies [23][24][25] emphasize that the fundamental k- and α-eigenfunctions are markedly different for systems that are far from critical, and the discrepancies become more pronounced in heterogeneous systems. This may lead to considerably different integral parameters, including the average fission production, neutron lifetime, and infinite multiplication factor. Furthermore, more recent studies [25][26] demonstrate the effectiveness of using the fundamental α-eigenfunction instead of the k- as the weighting spectrum in the hybrid eigenfunction expansion method Transition Rate Matrix Method (TRMM [27][28][29][30]).

Methods that calculate α-modes (or eigenpairs) of the time-dependent neutron transport problem without actually solving the time-dependent problem (like TRMM and Dynamic Mode Decomposition [22]) are needed so that one can use the resulting α-eigenfunctions as the multigroup constants weighting spectra. The α-eigenvalue Monte Carlo method has been an active research area [31][32][24][33][34][35][36][37][38][39][25]. Methods presented in the earlier works [32][24][33][36][39] are specifically devised to calculate the prompt α-mode, while later works [37][38][25] generalize these methods to include contributions from the delayed neutrons. Furthermore, methods for calculating higher α-modes are investigated in [34].

In the second part of this thesis, we investigate the effectiveness of the fundamental α-eigenfunction ($\boldsymbol{\alpha}$-MGXS) compared to the k-eigenfunction ($\boldsymbol{k}$-MGXS) as the multigroup constants weighting spectrum for time-dependent neutron transport calculations. This work is based on the author's accepted manuscript of an article [40] published in *Nuclear Science and Engineering*. Chapter 6 presents a review of α-eigenvalue neutron transport iteration methods, which are the key ingredient in generating the α-eigenfunction weighting spectrum for calculating $\boldsymbol{\alpha}$-MGXS. In

Chapter 7, $\boldsymbol{\alpha}$-MGXS for the time-stepping neutron transport method is derived, and its effectiveness is compared to that of $\boldsymbol{k}$-MGXS by simulations of multi-group kinetic test problems. Conclusions are presented in Chapter 8, along with discussions of planned future work.

CHAPTER 2

Multiple Balance Time-Discretization (MBTD)

In this chapter, the MBTD formulation and its basic features are discussed. In Section 2.1, we present several well-known first- and second-order time-stepping methods for solving general linear initial value problems. Section 2.2 introduces the MBTD formulation. In Section 2.3, a simple analysis is performed to reveal the accuracy and stability of the methods discussed in the previous sections. Section 2.4 discusses strategies for solving the MBTD coupled balance equations introduced in Section 2.2. Finally, an MBTD formulation for non-linear problems, which is particularly essential for multi-physics reactor simulations, is given in Section 2.5. A summary of the chapter is presented in Section 2.6.

2.1 Time-Stepping Methods

Let us consider an initial value problem with an autonomous (time-independent) linear operator:

$$\frac{\partial u}{\partial t} + \boldsymbol{L}u(t) = 0, \qquad t \geq 0, \tag{2.1}$$

with initial condition $u(0) = u_0$. In a typical time-stepping method, Eq. (2.1) is operated with $\Delta t^{-1}\int_{t_{k-1}}^{t_k}(\cdot)dt$ to obtain

$$\frac{u_k - u_{k-1}}{\Delta t} + \boldsymbol{L}\bar{u}_k = 0, \qquad k = 1, 2, \dots, \tag{2.2}$$

$$u_k = u(t_k), \qquad \bar{u}_k = \frac{1}{\Delta t}\int_{t_{k-1}}^{t_k} u(t)dt\,, \qquad t_k = k\Delta t, \tag{2.3}$$

with the aim of being able to calculate the *time-step solution* u_k from the given previous solution u_{k-1}. However, Eq. (2.2) introduces an additional unknown: the *time-average solution* $\bar{u}_k$. To proceed, we need to make an approximation for $\bar{u}_k$. Figure 2.1 illustrates the temporal grid of the numerical solution, where at each time step k we need to solve two unknowns ($\bar{u}_k$ and u_k) per Eq. (2.2).

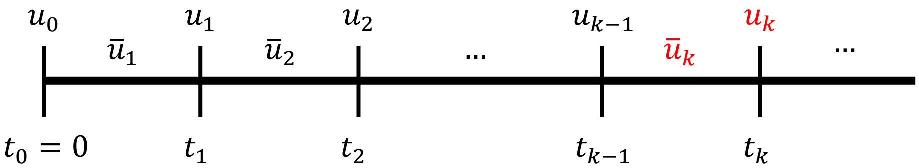


Figure 2.1: Temporal grid for the numerical solution of time-stepping methods

The standard time-stepping methods include Forward Euler (FE, or explicit Euler), Backward Euler (BE, or implicit Euler), and Crank-Nicholson (CN, or Trapezoid), which respectively approximate $\bar{u}_k$ as follows:

$$\bar{u}_k = \begin{cases} u_{k-1}, & \text{FE,} \\ u_k, & \text{BE,} \\ \dfrac{u_{k-1} + u_k}{2}, & \text{CN.} \end{cases} \qquad (2.4)$$

Equation (2.4) is often referred to as the *auxiliary equation* of the method, which is introduced to solve the *original balance equation*, Eq. (2.2).

In an *explicit* method like FE, all computational nodes (e.g., in the spatial variable) of the time-step solution u_k can be calculated independently, because each of them depends only on the previous time-step solution u_{k-1}. This means explicit methods can take full advantage of massively parallel computation. However, this embarrassingly parallel feature and the overall simplicity of FE (and other explicit methods) come with very limited stability (discussed in Section 2.3): an extremely small time-step size Δt is required to produce a meaningful converging solution, especially for a *stiff problem* [6] (which is typically the case in reactor simulations). On the other hand, *implicit* methods like BE and CN need to perform an operator solve to generate the solution u_k at each time-step, because the operator $\boldsymbol{L}$ acts upon the unknown u_k. However, this makes implicit methods like BE and CN stable and more suitable for the typically stiff problems in reactor simulation.

As will be discussed in Section 2.3, FE and BE are first-order accurate, while CN is second-order accurate. An advanced second-order accurate method, TR-BDF2, is suggested in [41]:

$$\begin{cases} \dfrac{u_k - u_{k-1}}{\Delta t} + \boldsymbol{L}\dfrac{u_{k-1} + u_{k-1/2} + u_k}{3} = 0, \\ \dfrac{u_{k-1/2} - u_{k-1}}{\Delta t/2} + \boldsymbol{L}\dfrac{u_{k-1} + u_{k-1/2}}{2} = 0. \end{cases} \tag{2.5}$$

TR-BDF2 offers better stability and improved accuracy compared to CN, with an additional cost of performing an extra operator solve at each time-step. In Eq. (2.5), we sequentially solve for $u_{k-1/2}$ and then u_k in the second and the first equation, respectively. We note that the second equation is essentially a CN method solving for $u_{k-1/2}$ from the given u_{k-1} with half of the time-step size Δt, and the first equation is a second-order backward-difference formula solving for u_k from the given $u_{k-1/2}$ and u_{k-1}.

2.2 MBTD

Strictly speaking, the auxiliary equations of FE, BE, and CN—shown in Eq. (2.4)—are not based on physics. In MBTD, a *physics-based* auxiliary equation is used instead. This auxiliary equation is obtained by finite-differencing the original continuous equation, Eq. (2.1), with the following criteria: (1) only the unknowns (u_k and $\bar{u}_k$) should be used, (2) the auxiliary equation should limit to Eq. (2.1) as $\Delta t \rightarrow 0$, and (3) should be as "implicit" as possible. We call the auxiliary equation that meets these criteria the *balance-like equation*:

$$\frac{u_k - \bar{u}_k}{\Delta t/2} + \boldsymbol{L}u_k = 0. \tag{2.6}$$

For convenience, the original balance Eq. (2.2) and MBTD's auxiliary (or balance-like) Eq. (2.6) are presented below:

$$\begin{cases} \dfrac{u_k - u_{k-1}}{\Delta t} + \boldsymbol{L}\bar{u}_k = 0, \\ \\ \dfrac{u_k - \bar{u}_k}{\Delta t/2} + \boldsymbol{L}u_k = 0. \end{cases} \tag{2.7}$$

Equation (2.7) is MBTD. The two equations in Eq. (2.7) are coupled; they cannot be sequentially solved like those in Eq. (2.5) of TR-BDF2. It was unknown whether there is an efficient strategy for solving the *coupled balance equations* of MBTD, Eq. (2.7). Before we present proposed strategies for solving the MBTD coupled equations, accuracy and stability of the described methods are discussed next.

2.3 Accuracy and Stability

Accuracy and stability of the time-stepping methods discussed in the previous sections (FE, BE, CN, TR-BDF2, and MBTD) can be characterized by considering the following simple time-dependent neutron transport problem (a more general time-dependent neutron transport is discussed in Chapter 4):

$$\frac{1}{v}\frac{d\psi}{dt} + \Sigma_t \psi(t) = 0, \qquad t \geq 0, \tag{2.8}$$

with initial condition $\psi(0) = \psi_0$ and total macroscopic cross-section $\Sigma_t > 0$. The solution $\psi(t)$ exponentially decays in time, and we seek a one-step solution $\psi(\Delta t) = \psi_1$:

$$\psi_1^{\text{Exact}} = \psi(\Delta t) = A_{\text{Exact}}(\eta)\psi_0, \tag{2.9}$$

where

$$A_{\text{Exact}}(\eta) = e^{-\eta} = 1 - \eta + \frac{1}{2}\eta^2 - \frac{1}{6}\eta^3 + O(\eta^4), \tag{2.10}$$

$$\eta = v\Sigma_t \Delta t. \tag{2.11}$$

We note that the solution ψ_1 is presented in terms of the *amplification factor* $A(\eta)$. The non-dimensional parameter η represents the number of *optical-thicknesses* (or mean-free paths Σ_t^{-1})

traveled by the neutrons per time-step size Δt; it is a measure of the numerical resolution or how effective the time-dependent physics is captured by the chosen time step Δt. The lower η is, the better the numerical resolution and the better the time-dependent physics is captured. By solving for ψ_1 from the given ψ_0 using the discussed time-stepping methods, we can determine the corresponding amplification factors $A(\eta)$. These are summarized in Table 2.1 and Figure 2.2 and are useful for accuracy and stability analysis.

Table 2.1: Summary for simple stability analysis of time-stepping methods

Method	Amplification factor: $A(\eta)$	Time-step error factor: $A_{\text{Exact}}(\eta) - A(\eta)$	Stability: $\lvert A(\eta)\rvert \leq 1$	Robustness: $0 \leq \lvert A(\eta)\rvert \leq 1$
FE	$1 - \eta$	$\frac{1}{2}\eta^2 + O(\eta^3)$	$\eta \leq 2$	$\eta \leq 1$
BE	$\frac{1}{1+\eta}$	$-\frac{1}{2}\eta^2 + O(\eta^3)$	$\eta \geq 0$	$\eta \geq 0$
CN	$\frac{1-\frac{1}{2}\eta}{1+\frac{1}{2}\eta}$	$\frac{1}{12}\eta^3 + O(\eta^4)$		$\eta \leq 2$
TR-BDF2	$\frac{12-5\eta}{12+7\eta+\eta^2}$	$\frac{1}{24}\eta^3 + O(\eta^4)$		$\eta \leq 2.4$
MBTD	$\frac{1}{1+\eta+\frac{1}{2}\eta^2}$	$-\frac{1}{6}\eta^3 + O(\eta^4)$		$\eta \geq 0$

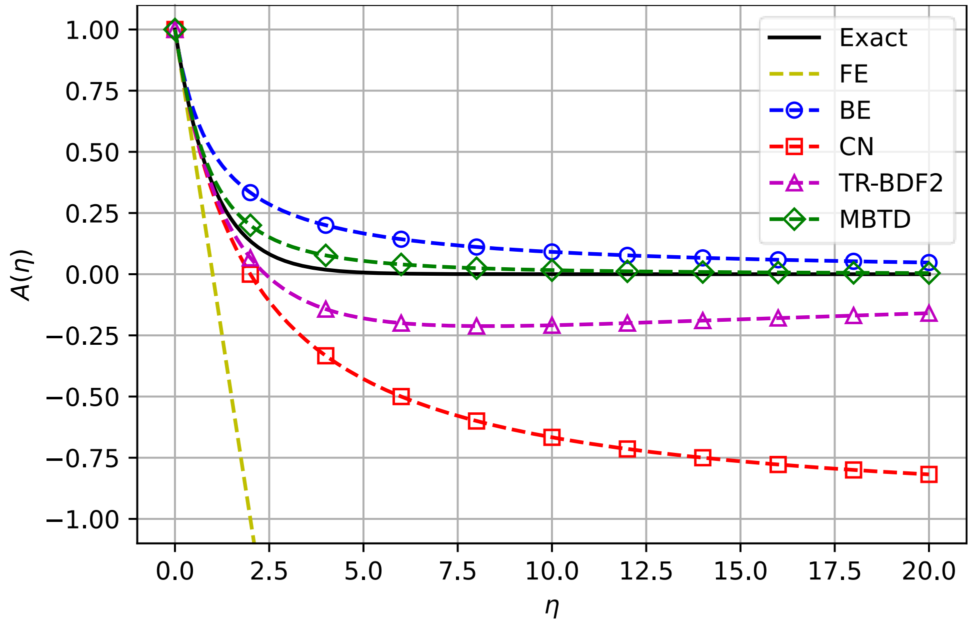


Figure 2.2: Amplification factor of the time-stepping methods

In general, the closer $A(\eta)$ is to $A_{\text{Exact}}(\eta)$ (which can be observed in Figure 2.2), the better the method is in replicating the exact solution for different values of η. Accordingly, small values of η are typically preferred for sufficient accuracy. The error factors introduced per time step, $[A_{\text{Exact}}(\eta) - A(\eta)]$, are shown in Table 2.1 for the five time-stepping methods. It is shown that FE and BE are first-order accurate, while CN, TR-BDF2, and MBTD are second-order accurate. Additionally, it is worth mentioning that in general (by looking at the leading coefficients), TR-BDF2 is twice as accurate as CN, and CN is twice as accurate as MBTD.

The sign and magnitude of $A(\eta)$ characterize the stability of a method at a given η. If $|A(\eta)| \leq 1$, the method is *stable* and produces a non-diverging solution (numerical error not growing per time step). If $A(\eta) < 0$, *spurious oscillations* are produced in the solution (a solution ψ_k will have a different sign from the previous solution ψ_{k-1}). These spurious oscillations will be damped (with a damping ratio of $|A(\eta)|$) only if $-1 < A(\eta) < 0$. Otherwise, if $A(\eta) \leq -1$, the method becomes unstable, and its solution diverges.

A sufficiently small η is required for FE to guarantee stability. This strict condition makes the method not appropriate for a typically stiff problem, which is characterized by high $v\Sigma_t$ for this example. For CN, it is found that $A(\eta)$ approaches -1 as $\eta \to \infty$. This means that CN always damps any occurring spurious oscillation, since $|A(\eta)| \leq 1$ regardless of Δt. However, the damping will be inefficient if η is insufficiently small, because the damping ratio $|A(\eta)|$ will be very close to one.

For TR-BDF2, $A(\eta)$ approaches 0 as $\eta \to \infty$, which makes TR-BDF2 an *L-Stable* method [6], a highly desirable stability property for solving a stiff problem. Furthermore, it is guaranteed that the damping ratio of TR-BDF2 is always smaller than ~ 0.212, as shown in Figure 2.2. Nevertheless, even though well-damped, spurious oscillations are still possible with TR-BDF2 for relatively large η.

Amplification factors for BE and MBTD decay to 0 as $\eta \to \infty$. This makes both methods not only L-Stable, but also free from spurious oscillation, because $A(\eta) > 0$ regardless of Δt. Given these, we consider BE and MBTD as *robust* time-stepping methods. (We define a robust method as one that is not only L-Stable but also free from spurious oscillations.)

It is worth mentioning that the amplification factor of BE, CN, and TR-BDF2 may be negative in the case of $\Sigma_t < 0$ (exponentially growing solution, or supercritical system); however, this is not the case for MBTD. Nevertheless, Δt is typically very small in a supercritical system to achieve acceptable accuracy and convergence of the neutron transport source iteration (discussed in Chapter 4).

To demonstrate the robustness of BE and MBTD, we apply the described methods (FE, BE, CN, TR-BDF2, and MBTD) to solve the following simple stiff problem:

$$\begin{cases} \dfrac{dy}{dt} + \dfrac{1}{2} y(t) = 20x(t), \qquad y(0) = 0, \\ \\ \dfrac{dx}{dt} + 20x(t) = 0, \qquad x(0) = 1. \end{cases} \tag{2.12}$$

In general, a problem is called *stiff* if we are attempting to compute a solution that is smooth and slowly varying (relative to the time interval of the computation), but in a context where the nearby solution curves are much more rapidly varying [6]. The stiffness of a problem is often characterized by the significantly different time scale of its solutions. For example, in Eq. (2.12), the solution $x(t)$ is much more rapidly varying than $y(t)$. A more practical example would be the significant different time scales of solutions of the neutron and delayed neutron precursor equations (discussed more in Chapter 3).

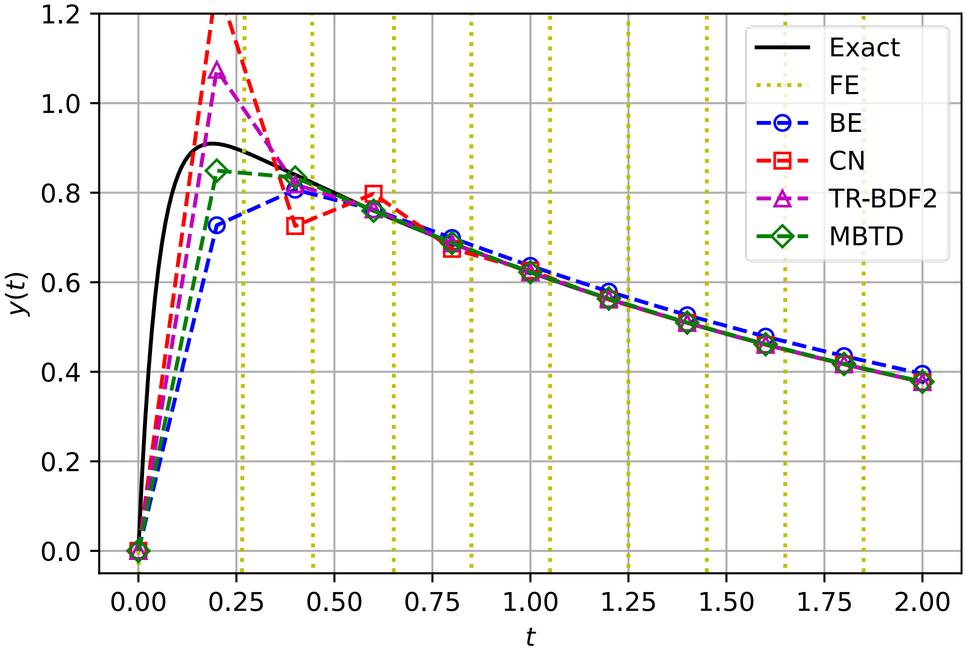


Figure 2.3: Time-stepping method solutions for the simple linear stiff problem

The analytical solution of $y(t)$ and its numerical time-stepping solutions with $\Delta t = 0.2$

are shown in Figure 2.3. It is found that the FE solution diverges. CN and TR-BDF2 produce oscillating solutions; however, the oscillations are gradually damped, where TR-BDF2 damps the oscillation better than CN. Meanwhile, solutions of BE and MBTD are free from spurious oscillations, but MBTD produces a more accurate solution than BE—both in the initial "boundary layer" and the asymptotic solution—due to its higher-order accuracy.

2.4 Strategies for Solving the Coupled Balance Equations

In MBTD, the balance-like auxiliary equation Eq. (2.6) is introduced to solve the original balance equation Eq. (2.2); this results in the coupled balance equations [Eq. (2.7), rewritten below for convenience]:

$$\begin{cases} \dfrac{u_k - u_{k-1}}{\Delta t} + \boldsymbol{L}\bar{u}_k = 0, \\[2ex] \dfrac{u_k - \bar{u}_k}{\Delta t/2} + \boldsymbol{L}u_k = 0. \end{cases} \tag{2.13}$$

In relatively simple problems like those in Eqs. (2.8) and (2.12), the coupled equations can be straightforwardly solved by substituting the second equation into the first one. However, in the more practical problem (such as the neutron transport), the linear operator $\boldsymbol{L}$ is not as simple, making direct substitution prohibitive.

One of the significant merits of using the standard time-stepping methods—in particular BE, CN, and TR-BDF2—is that we can leverage the standard steady-state operator solver of the physics of interest, which is typically represented in the form of

$$\boldsymbol{L}u = Q, \tag{2.14}$$

(or in the generic linear algebra form: $Ax = b$) where Q is a right-hand-side (RHS) independent source. For example, in BE, Eq. (2.2) reduces to

$$\frac{u_k - u_{k-1}}{\Delta t} + \boldsymbol{L}u_k = 0, \tag{2.15}$$

$$\left(\boldsymbol{L} + \frac{1}{\Delta t}\right)u_k = \frac{1}{\Delta t}u_{k-1}. \tag{2.16}$$

This can be easily re-cast into the steady-state form of Eq. (2.14), where we introduce the so-called *time-absorption* term $1/\Delta t$ into the steady-state operator $\boldsymbol{L}$ and introduce the previous solution term u_{k-1} into the RHS independent source Q. Such straightforward adaptation of the steady-state solver applies for CN and TR-BDF2 as well. However, it is less obvious if the same approach can be efficiently used to solve the coupled equations of MBTD, Eq. (2.13). Three strategies to solve Eq. (2.13) are investigated in this section: (1) Substitution, (2) Iterative Solve, and (3) Simultaneous Solve.

2.4.1 Substitution

In Substitution, we substitute the second equation (balance-like auxiliary equation) of Eq. (2.13) into the first one (original balance equation); this gives

$$\frac{u_k - u_{k-1}}{\Delta t} + \left(\boldsymbol{L} + \frac{1}{2}\Delta t \boldsymbol{L}^2\right) u_k = 0. \tag{2.17}$$

The merit of Substitution is that we reduce the two coupled balance equations [Eq. (2.13)] having two unknowns $\bar{u}_k$ and u_k into a single combined equation [Eq. (2.17)] having one unknown u_k.

MBTD with the Substitution strategy [Eq. (2.17)] is structurally more comparable to BE [Eq. (2.15)]. They share the same memory complexity for the unknown u_k, and the only difference is the operator $\boldsymbol{L}^2$ term in Eq. (2.17). It is found that the $\boldsymbol{L}^2$ term serves as a correction term to BE, which allows MBTD to achieve a higher order of accuracy while maintaining robustness. However, this Substitution strategy is advantageous only if the operator $\boldsymbol{L}$ is simple, such that numerically handling $\boldsymbol{L}^2$ is practical.

2.4.2 Iterative Solve

The motivation of the Iterative Solve is to make use of the standard steady-state operator solver as a black box, similar to what is adopted in the standard time-stepping methods [in particular, BE Eq. (2.15)]. The main idea is to lag some terms in Eq. (2.13) such that we obtain some forms of the standard steady-state Eq. (2.14), which then can be solved iteratively. Five iterative schemes are formulated: (1) Simple Lag, (2) Parallel Lag, (3) Lagged Substitution, (4) Lagged

$O(\Delta t^2)$, and (5) Lagged $O(\Delta t)$. Later in this section, a simple convergence analysis is performed to assess the convergence rate of the proposed iteration schemes.

Simple Lag

In Simple Lag, we lag u_k in the first equation of Eq. (2.13):

$$\begin{cases} \dfrac{u_k^{(l-1)} - u_{k-1}}{\Delta t} + \boldsymbol{L}\bar{u}_k^{(l)} = 0, \\ \\ \dfrac{u_k^{(l)} - \bar{u}_k^{(l)}}{\Delta t/2} + \boldsymbol{L}u_k^{(l)} = 0. \end{cases} \tag{2.18}$$

The superscripts $^{(l)}$ indicate the iteration index. First, we solve the first equation of Eq. (2.18) to get $\bar{u}_k^{(l)}$ from a given u_{k-1} and an initial guess (or the previous iterate solution) of $u_k^{(l-1)}$. Later, by using the newly solved $\bar{u}_k^{(l)}$, we solve the second equation to obtain the new iterate solution of $u_k^{(l)}$. The iteration is performed until the solutions $\bar{u}_k$ and u_k converge.

Parallel Lag

In Parallel Lag, we respectively lag u_k and $\bar{u}_k$ in the first and the second equation of Eq. (2.13):

$$\begin{cases} \dfrac{u_k^{(l-1)} - u_{k-1}}{\Delta t} + \boldsymbol{L}\bar{u}_k^{(l)} = 0, \\ \\ \dfrac{u_k^{(l)} - \bar{u}_k^{(l-1)}}{\Delta t/2} + \boldsymbol{L}u_k^{(l)} = 0. \end{cases} \tag{2.19}$$

Since the first and the second equations are now independent, they can be simultaneously solved in parallel; this gives Parallel Lag a computational advantage over the other iterative strategies.

Lagged Substitution

This method modifies the combined equation of the Substitution strategy by lagging the BE-correction term in Eq. (2.17):

$$\begin{cases} \dfrac{u_k^{(l)} - u_{k-1}}{\Delta t} + \boldsymbol{L}u_k^{(l)} + \zeta^{(l-1)} = 0, \\ \\ \zeta^{(l)} = \dfrac{1}{2}\Delta t \boldsymbol{L}^2 u_k^{(l)}. \end{cases} \tag{2.20}$$

By lagging the BE-correction term ζ, we avoid directly solving the $\boldsymbol{L}^2$ operator, which is the main limitation of the Substitution strategy. The new iterate solution of ζ can be evaluated as the following:

$$\zeta^{(l)} = \frac{1}{2}\Delta t \boldsymbol{L}\left(\boldsymbol{L}u_k^{(l)}\right) = -\frac{1}{2}\Delta t \boldsymbol{L}\left(\frac{u_k^{(l)} - u_{k-1}}{\Delta t} + \zeta^{(l-1)}\right). \tag{2.21}$$

<u>*Lagged* $O(\Delta t^2)$</u>

It is a rule of thumb [12] to lag a small quantity in formulating an iterative method. In this method, the following small quantity,

$$u_k - 2\bar{u}_k + u_{k-1} = O(\Delta t^2), \tag{2.22}$$

is introduced into the first equation of Eq. (2.13) as the lagged term in the iteration scheme:

$$\begin{cases} \dfrac{\bar{u}_k^{(l)} - u_{k-1}}{\Delta t/2} + \boldsymbol{L}\bar{u}_k^{(l)} = -\dfrac{1}{\Delta t}\left(u_k^{(l-1)} - 2\bar{u}_k^{(l-1)} + u_{k-1}\right), \\ \\ \dfrac{u_k^{(l)} - \bar{u}_k^{(l)}}{\Delta t/2} + \boldsymbol{L}u_k^{(l)} = 0. \end{cases} \tag{2.23}$$

By careful observation, one can find that Eq. (2.23) does not actually lag the small quantity that we hoped for [which is the LHS of Eq. (2.22)]; it just lags a part of it [the first two terms, shown with superscript $^{(l-1)}$ in Eq. (2.23)], which is not $O(\Delta t^2)$, and may result to a sub-optimally converging iteration scheme.

<u>*Lagged* $O(\Delta t)$</u>

This iterative method is formulated to anticipate the issue found in the formulation of the Lagged $O(\Delta t^2)$. Instead of Eq. (2.22), the lagged quantity introduced into Eq. (2.13) is

$$u_k - \bar{u}_k = O(\Delta t), \tag{2.24}$$

which yields the following iteration scheme:

$$\begin{cases} \dfrac{\bar{u}_k^{(l)} - u_{k-1}}{\Delta t} + \boldsymbol{L}\bar{u}_k^{(l)} = -\dfrac{1}{\Delta t}\left(u_k^{(l-1)} - \bar{u}_k^{(l-1)}\right), \\ \dfrac{u_k^{(l)} - \bar{u}_k^{(l)}}{\Delta t/2} + \boldsymbol{L}u_k^{(l)} = 0. \end{cases} \tag{2.25}$$

It is worth mentioning that, compared to BE [Eq. (2.15)], all the five Iterative Solve strategies have about twice the memory complexity, because they need to store both u_k and $\bar{u}_k$ (or ζ in case of Lagged Substitution) during the iteration.

Convergence analysis

A simple convergence analysis is performed to assess the convergence rates for the five formulated Iterative Solve strategies: (1) Simple Lag, (2) Parallel Lag, (3) Lagged Substitution, (4) Lagged $O(\Delta t^2)$, and (5) Lagged $O(\Delta t)$. The simple problem of Eq. (2.8) is considered. We apply MBTD by using the five Iterative Solve strategies, and then re-casting the resulting numerical equations into the typical iteration matrix form [6]:

$$\boldsymbol{M}u^{(l)} = \boldsymbol{N}u^{(l-1)} + s, \tag{2.26}$$

where $u = [\bar{u}_k \quad u_k]$ and $s = [u_{k-1} \quad 0]/\Delta t$ for Simple Lag, Parallel Lag, Lagged $O(\Delta t^2)$, and Lagged $O(\Delta t)$, and $u = u_k$ and $s = u_{k-1}/\Delta t$ for Lagged Substitution. Given the matrices $\boldsymbol{M}$ and $\boldsymbol{N}$, we can determine the iteration matrix $\boldsymbol{G} = \boldsymbol{M}^{-1}\boldsymbol{N}$ and its corresponding *spectral radius* $\rho(\boldsymbol{G})$ (the largest absolute value of eigenvalues of a matrix). The resulting spectral radii for the Iterative Solve strategies are shown in Figure 2.4.

The spectral radius ρ of an iteration matrix represents the worst-case estimate for the error reduction factor per iteration; the smaller ρ, the faster the convergence; while $\rho > 1$ indicates that the iteration scheme diverges. From Figure 2.4, it is found that Simple Lag generally converges more quickly than Parallel Lag; yet, if the two equations in Eq. (2.19) are simultaneously solved in parallel, Parallel Lag potentially reduces the required time in producing iterate solutions by a half. Nevertheless, both Simple Lag and Parallel Lag diverge for smaller η. On the other hand, Lagged Substitution works very well for smaller η; however, it diverges for larger η. Since the whole point of MBTD is to obtain a higher-order robust method, we need an iterative strategy that converges regardless of η. Such requirement is satisfied only by Lagged $O(\Delta t^2)$ and Lagged $O(\Delta t)$, where Lagged $O(\Delta t)$ performs better due to its generally lower ρ; this finding demonstrates the utility of consistently lagging a small quantity in an iterative scheme.

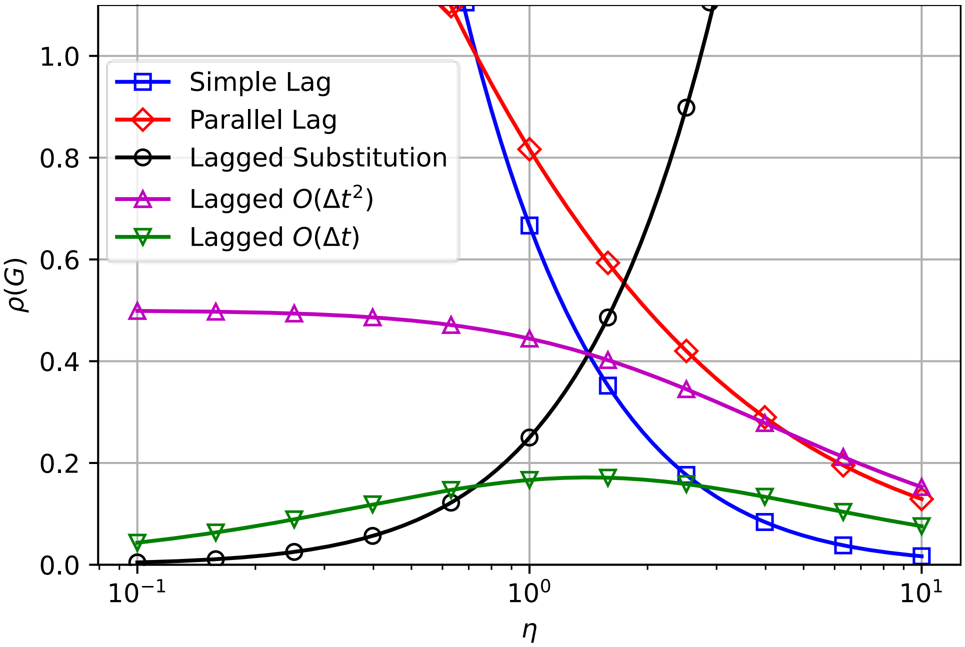


Figure 2.4: Spectral radius of the MBTD Iterative Solve strategies

Based on the simple convergence analysis, it is found that Lagged $O(\Delta t)$ [Eq. (2.25)] is the best Iterative Solve strategy for solving the MBTD coupled equations. Nevertheless, despite the relatively low spectral radius of Lagged $O(\Delta t)$ (i.e., ρ is always lower than 0.2), the Iterative Solve strategy may make MBTD significantly more expensive than BE. In BE [Eq. (2.15)], we only need to perform one steady-state solve to obtain the next time-step solution u_k. In MBTD with Lagged $O(\Delta t)$ strategy [Eq. (2.25)], we need to perform two steady-state solves at each iteration. In two iterations, MBTD with the Iterative Solve is already four times more expensive than BE. This significant extra computational cost may outweigh the higher-order accuracy, which could make MBTD with Iterative Solve strategy less efficient than BE.

Exploiting the steady-state solver may result in a more competitive overall iteration scheme, particularly if the steady-state problem is solved iteratively. This can be done by considering the MBTD Iterative Solve as the *outer iteration* and the iterative steady-state solver as the *inner iteration*. The idea is to avoid unnecessary inner iteration convergence in the earlier stage of the outer iteration; thus, instead of converging the inner iteration, a certain number of inner iterations (e.g., one) is performed per outer iteration. Since the outer and the inner iterations are intertwined together, a problem/solver-specific convergence analysis needs to be done to determine the overall convergence rate, and it is possible that the outcome may be different than the one shown in Figure 2.4, to the extent that an originally non-converging method (i.e., Simple Lag, Parallel Lag, and Lagged Substitution,) may converge when it is intertwined with the steady-state iterative solver.

2.4.3 Simultaneous Solve

In this MBTD strategy, we simultaneously solve for $\bar{u}_k$ and u_k in Eq. (2.13). Simultaneous Solve is analog to that performed in BE per Eq. (2.16). To show this, we re-cast Eq. (2.13) into the following

$$\begin{cases} \boldsymbol{L}\bar{u}_k + \dfrac{1}{\Delta t}u_k = \dfrac{1}{\Delta t}u_{k-1}, \\ -\dfrac{2}{\Delta t}\bar{u}_k + \left(\boldsymbol{L} + \dfrac{2}{\Delta t}\right)u_k = 0. \end{cases} \tag{2.27}$$

Once the rest of the system's phase space (in case of neutron transport: space, direction of flight, and kinetic energy) is discretized, Eq. (2.16) of BE and Eq. (2.27) of MBTD are reduced to systems of algebraic equations, which are illustrated in Figure 2.5.

In Figure 2.5, $\boldsymbol{I}$ is the diagonal identity matrix, and $\boldsymbol{L}$ represents the typical steady-state matrix operator. It is evident that MBTD with Simultaneous Solve strategy consists of solving a linear problem with a “four-block” matrix, with a doubled number of unknowns (and thus memory complexity) compared to that of BE. The diagonal blocks are structurally identical to the “single-block” matrix of BE; they only differ in the magnitude of the time-absorption Δt^{-1}. Meanwhile, the off-diagonal blocks are just uniform-valued diagonal matrices.

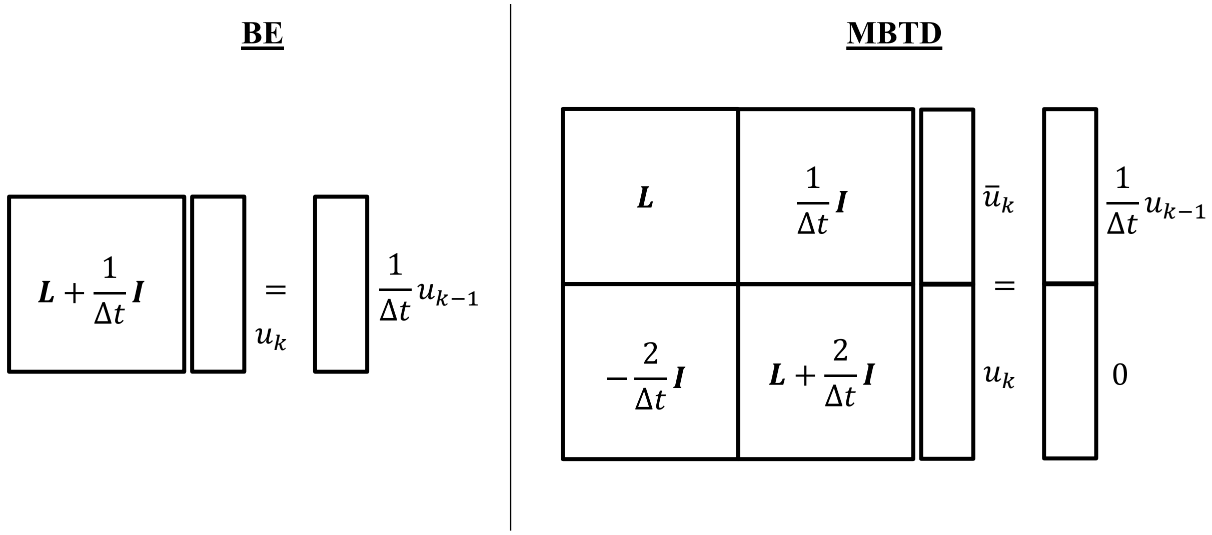

BE
$L + \frac{1}{\Delta t} I$
u_k
=
$\frac{1}{\Delta t} u_{k-1}$
MBTD
L
$\frac{1}{\Delta t} I$
$-\frac{2}{\Delta t} I$
$L + \frac{2}{\Delta t} I$
$\bar{u}_k$
u_k
=
$\frac{1}{\Delta t} u_{k-1}$
0

Figure 2.5: Systems of algebraic equations of BE and MBTD

Once the four-block matrix (and also the long RHS vector) is constructed, one can use a standard linear solver to find the unknown $[\bar{u}_k \quad u_k]$, similarly to what would be done in BE. However, some methods, particularly the iterative ones, may need a slight modification to be able to solve the MBTD four-block matrix. For example, the Gauss-Seidel (GS) method requires a strictly diagonally dominant matrix to guarantee convergence. For some physical problems, such as neutron diffusion (discussed in Chapter 3), this condition for convergence is always satisfied by the single-block matrix of BE. However, it may not be satisfied by the four-block matrix of MBTD if the Δt is very small such that the elements of the upper-right block dominate the diagonal element of the upper-left block of the matrix. This issue can be remedied by modifying the iteration scheme. Let us re-cast the single-block matrix problem of BE as the following:

$$\boldsymbol{A}x = (\boldsymbol{A_L} + \boldsymbol{A_D} + \boldsymbol{A_U})x = b, \tag{2.28}$$

where the unknown $x = u_k$ and A_L, A_D, and A_U respectively denote the lower diagonal, diagonal, and upper diagonal parts of a BE matrix A. GS iteratively solves Eq. (2.28) as follows:

$$(\boldsymbol{A_L} + \boldsymbol{A_D})x^{(l)} = b - (\boldsymbol{A_U})x^{(l-1)}, \tag{2.29}$$

where the superscripts $^{(l)}$ denote the iteration index. Now, let us similarly re-cast the four-block matrix problem of MBTD:

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} x = \begin{bmatrix} \boldsymbol{A_{11,L}} + \boldsymbol{A_{11,D}} + \boldsymbol{A_{11,U}} & \boldsymbol{A_{12,D}} \\ \boldsymbol{A_{21,D}} & \boldsymbol{A_{22,L}} + \boldsymbol{A_{22,D}} + \boldsymbol{A_{22,U}} \end{bmatrix} x = b, \qquad (2.30)$$

where the unknown $x = [\bar{u}_k \quad u_k]$ and we note that the off-diagonal blocks (A_{12} and A_{21}) are diagonal. To avoid the non-diagonally dominance issue of the four-block MBTD matrix, the modified-GS iteratively solves Eq. (2.30) as follows:

$$\begin{bmatrix} \boldsymbol{A_{11,L}} + \boldsymbol{A_{11,D}} & \boldsymbol{A_{12,D}} \\ \boldsymbol{A_{21,D}} & \boldsymbol{A_{22,L}} + \boldsymbol{A_{22,D}} \end{bmatrix} x^{(l)} = b - \begin{bmatrix} \boldsymbol{A_{11,U}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A_{22,U}} \end{bmatrix} x^{(l-1)}. \qquad (2.31)$$

It is illustrative to observe that, instead of sequentially solving the whole unknowns from the top to bottom as is done in the standard GS, the modified-GS *simultaneously* solve the two unknowns—sequentially—from their respective top to bottom, which is shown in Figure 2.6. This discussion about modifying standard GS serves as a guideline for adapting other iterative methods—that are also characterized with "sweeping" solutions from one end to the other—for solving the four-block matrix of MBTD Simultaneous Solve strategy. Such iterative methods include SOR (Successive Over-Relaxation), and more importantly, the *neutron transport source iteration* (discussed in Chapter 4).

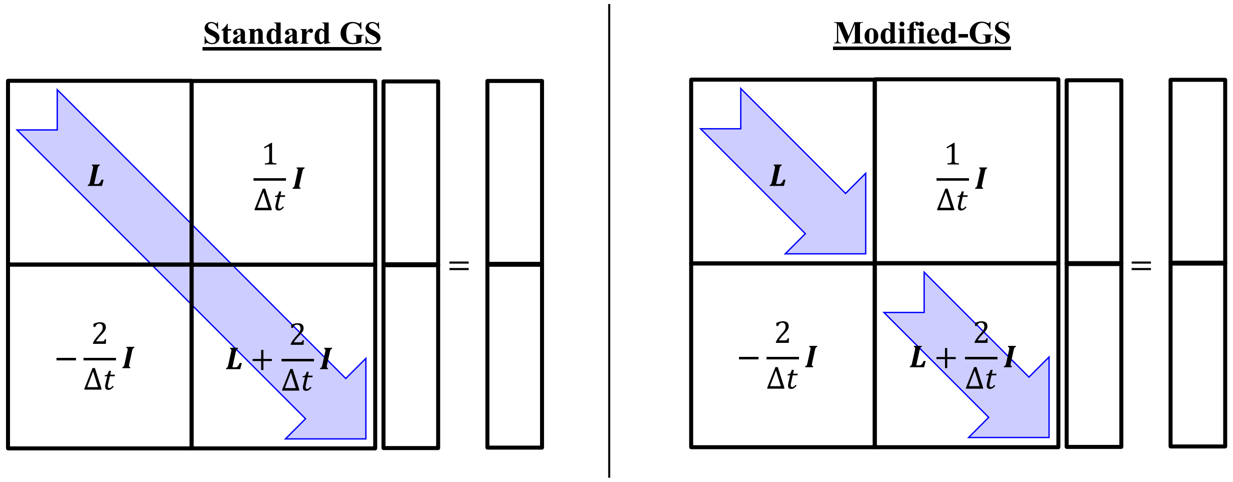


Figure 2.6: GS vs. the modified-GS for MBTD Simultaneous Solve strategy

2.5 MBTD for Non-Linear Problems

The previous sections discuss the application of MBTD for solving a linear time-dependent problem with an autonomous (time-independent) operator Eq. (2.1), and the resulting MBTD coupled balance equations are shown in Eq. (2.13). In this section, we discuss the application of MBTD for solving a non-linear time-dependent problem, which is essential for multi-physics reactor simulation (discussed in Chapter 5).

Let us consider a generic non-linear problem:

$$\frac{\partial u}{\partial t} = f[t, u(t)], \qquad t \geq 0, \tag{2.32}$$

with initial condition $u(0) = u_0$. Similar to Section 2.1, Eq. (2.32) is operated with $\Delta t^{-1} \int_{t_{k-1}}^{t_k} (\cdot) dt$ to obtain

$$\frac{u_k - u_{k-1}}{\Delta t} = \frac{1}{\Delta t} \int_{t_{k-1}}^{t_k} f[t, u(t)] dt\,, \qquad k = 1, 2, \dots. \tag{2.33}$$

Since $f[t, u(t)]$ is generally unknown, we need to make an approximation for the integral in Eq. (2.33). Time-stepping methods essentially differ from how they approximate this integral. Given this observation, time-stepping methods are essentially numerical integration methods.

The standard time-stepping methods (FE, BE, and CN) approximate Eq. (2.33) as follows:

$$\frac{u_k - u_{k-1}}{\Delta t} = \begin{cases} f(t_{k-1}, u_{k-1}), & \text{FE,} \\ f(t_k, u_k), & \text{BE,} \\ \dfrac{f(t_{k-1}, u_{k-1}) + f(t_k, u_k)}{2}, & \text{CN.} \end{cases} \tag{2.34}$$

By referring to the balance-like auxiliary equation of the linear problem Eq. (2.6), one can formulate a balance-like auxiliary equation for the non-linear problem of Eq. (2.32):

$$\frac{u_k - \bar{u}_k}{\Delta t/2} = f(t_k, u_k). \tag{2.35}$$

However, the original balance equation Eq. (2.33) and the derived balance-like auxiliary equation Eq. (2.35) do not make solvable coupled balance equations. This is because Eq. (2.35) does not satisfy the first criteria (discussed in Section 2.2) of the MBTD balance-like auxiliary equation, as it introduces a new unknown of $\bar{u}_k$.

By closely observing Eq. (2.35) [and Eq. (2.6) for the linear problem], it is evident that $\bar{u}_k$ is essentially a solution that is located (in time) $\Delta t/2$ prior to u_k. This means, MBTD essentially suggests that the time-average solution approximately equals the *mid-point solution*, or $\bar{u}_k = u_{k-1/2} = u(t_{k-1} + \Delta t/2)$. By following this underlying idea, one would approximate the time-average of $f[t, u(t)]$ as its mid-point value, which yields solvable MBTD coupled balance equations for the non-linear problem:

$$\begin{cases} \dfrac{u_k - u_{k-1}}{\Delta t} = f\left(t_{k-1/2}, u_{k-1/2}\right), \\ \dfrac{u_k - u_{k-1/2}}{\Delta t/2} = f(t_{k-1}, u_{k-1}). \end{cases} \tag{2.36}$$

We note that Eq. (2.36) reduces to Eq. (2.7) if the operator f is linear and autonomous.

The MBTD coupled balance equations for the non-linear problem shown in Eq. (2.36) is essentially a *mid-point method*, which (1) approximates the time-average of $f[t, u(t)]$ to its mid-point value $f\left(t_{k-1/2}, u_{k-1/2}\right)$ and (2) introduces an auxiliary equation to describe the mid-point solution $u_{k-1/2}$. For MBTD, the mid-point solution is approximated by a *right-implicit* (or right-BE) equation. As a comparison, the more widely known mid-point method, which is also known as the two-stage explicit Runge-Kutta (RK) method [6], approximates the mid-point solution $u_{k-1/2}$ with a left-explicit (or left-FE) equation:

$$\begin{cases} \dfrac{u_k - u_{k-1}}{\Delta t} = f\left(t_{k-1/2}, u_{k-1/2}\right), \\ \dfrac{u_{k-1/2} - u_{k-1}}{\Delta t/2} = f(t_{k-1}, u_{k-1}). \end{cases} \tag{2.37}$$

Any reasonably formulated time-stepping method (or one-step method for initial value problem ODEs [6]), including MBTD, can be represented in the s-stage RK form:

$$\begin{cases} u_k = u_{k-1} + \Delta t \displaystyle\sum_{i=1}^{s} b_i f(t_{k-1} + c_i \Delta t, Y_i)\,, \\[2ex] Y_i = u_{k-1} + \Delta t \displaystyle\sum_{j=1}^{s} a_{ij} f\big(t_{k-1} + c_j \Delta t, Y_j\big)\,, \qquad i = 1, 2, \dots, s, \end{cases} \tag{2.38}$$

whose coefficients are often displayed in the so-called Butcher tableau:

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array} \tag{2.39}$$

For example, the Butcher tableau for FE, BE, CN [Eq. (2.34)], two-stage explicit RK [or left-explicit mid-point, Eq. (2.37)], and MBTD [or right-implicit mid-point, Eq. (2.36)] are respectively shown below:

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array} \tag{2.40}$$

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array} \tag{2.41}$$

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1/2 & 1/2 \\ \hline & 1/2 & 1/2 \end{array} \tag{2.42}$$

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1/2 & 1/2 & 0 \\ \hline & 0 & 1 \end{array} \tag{2.43}$$

$$\begin{array}{c|cc} 1/2 & 1 & -1/2 \\ 1 & 1 & 0 \\ \hline & 1 & 0 \end{array} \tag{2.44}$$

Furthermore, for RK methods [6], consistency requires

$$1 = \sum_{i=1}^{s} b_i\,, \qquad c_i = \sum_{j=1}^{s} a_{ij}\,, \qquad i = 1, 2, \dots, s, \tag{2.45}$$

while second-order accuracy requires

$$\frac{1}{2} = \sum_{i=1}^{s} b_i c_i\,, \tag{2.46}$$

which is satisfied by CN, two-stage explicit RK, and MBTD.

To test the non-linear formulation of MBTD and further demonstrate its robustness compared to the widely-used methods BE and CN, a simple non-linear stiff problem from [6] is considered:

$$\frac{du}{dt} = \lambda[u(t) - \cos(t)] - \sin(t)\,, \tag{2.47}$$

with $\lambda = -10^6$, initial condition $u(0) = 1.5$, analytical solution

$$u(t) = e^{\lambda}[u(0) - 1] - \cos(t)\,. \tag{2.48}$$

The problem is numerically solved with $\Delta t = 0.1$, and the numerical solutions are shown in Figure 2.7. It is evident that even though stable, CN does not well handle under-resolved transients. On the other hand, the robust methods BE and MBTD still produce meaningful solutions even with a relatively large time-step size.

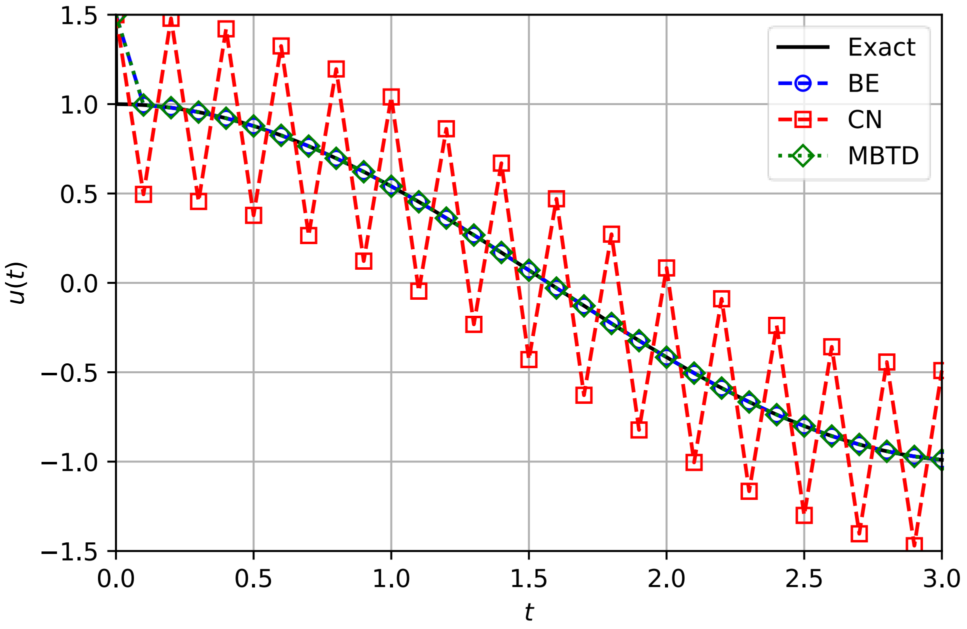


Figure 2.7: Time-stepping method solutions for the simple non-linear stiff problem

2.6 Summary

In this chapter, we have discussed the formulation and basic features of the proposed time-stepping method MBTD. MBTD consists of solving two coupled balance equations [Eq. (2.13)]—the original balance equation and the physics-based balance-like auxiliary equation—to obtain the next time-step solution u_k from the previous solution u_{k-1}. A simple analysis, summarized in Table 2.1 and Figure 2.2, shows that MBTD is not only second-order accurate, but also robust (i.e., L-stable and free from spurious oscillation regardless of time-step size Δt). Furthermore, a simple stiff problem Eq. (2.12)—whose analytical and numerical solutions are shown in Figure 2.3—is devised to demonstrate the second-order accuracy and robustness of MBTD.

It is unknown if there is an efficient strategy to solve the MBTD coupled balance equations. Three strategies are proposed: (1) Substitution, (2) Iterative Solve, and (3) Simultaneous Solve. Substitution solves the coupled balance equations by directly substituting the auxiliary equation into the original balance equation, which reduces the problem into a single combined equation [Eq. (2.17)]. However, Substitution is advantageous only if numerically handling the $\boldsymbol{L}^2$ operator is practical.

Iterative Solve iteratively solves the coupled balance equations by lagging some of the terms. Six Iterative Solve strategies are proposed: (1) Simple Lag, (2) Parallel Lag, (3) Lagged Substitution, (4) Lagged $O(\Delta t^2)$, and (5) Lagged $O(\Delta t)$. A simple analysis, summarized in Figure 2.4, shows that Lagged $O(\Delta t)$ [Eq. (2.25)] is the most optimum method because of its always converging iteration scheme ($\rho < 0$, regardless of Δt) and relatively low spectral radius ρ. To mitigate the significant additional cost (relative to that of BE) due to the multiple solves required in the iteration, it is suggested to exploit the steady-state solver. For example, if the steady-state problem is also solved iteratively, we can set a certain number of inner iteration (steady-state solve) per outer iteration, such that unnecessarily converging the steady-state problem in the earlier stage of the outer iteration can be avoided.

Simultaneous Solve solves for the two unknowns ($\bar{u}_k$ and u_k) in the coupled balance equations, which make a four-block matrix problem as shown in Figure 2.5. An adaptation of a standard operator solver may be required to appropriately solve the MBTD four-block matrix problem. For an iterative operator solver characterized by sweeping solutions from one end of the computational nodes to the other—such as GS, SOR, and more importantly, the *neutron transport source iteration* (discussed in Chapter 4)—we need to modify the iteration scheme. This is done by *simultaneously* solving for the two unknowns, as illustrated in Figure 2.6.

Finally, MBTD coupled balance equations for non-linear problems, which is essential for multi-physics reactor simulation (discussed in Chapter 5), are formulated and shown in Eq. (2.36). It is found that MBTD is essentially a variant of the mid-point method, where the mid-point solution is approximated by a right-implicit auxiliary equation. Furthermore, MBTD can be re-cast into a two-stage implicit RK method [Eq. (2.38)], whose Butcher tableau is shown in Eq. (2.44). Numerical solutions of a simple non-linear stiff problem shown in Figure 2.7 further emphasizes the robustness of MBTD compared to the widely-used implicit methods BE and CN.

The formulation and strategies discussed in this chapter serve as general guidance for applying MBTD to a time-dependent problem. In Chapter 3 and Chapter 4, MBTD application to neutron diffusion and neutron transport is respectively discussed, followed by its application to a typical multi-physics (neutronics + thermal-hydraulics) reactor problem in Chapter 5.

CHAPTER 3

MBTD Application to Neutron Diffusion

The neutron diffusion method has been an essential tool for reactor simulations [8]. It approximates the physical neutron transport equation (discussed in Chapter 4) by assuming that neutrons “diffuse” from high to low neutron density according to *Fick’s Law* [2]:

$$\vec{J} = \int_{4\pi} \hat{\Omega}\psi(\hat{\Omega})d\Omega \approx -D\nabla\left[\int_{4\pi} \psi(\hat{\Omega})d\Omega\right] = -D\nabla\phi. \tag{3.1}$$

Here ψ is the directional angular flux, ϕ is the neutron flux, and $\vec{J}$ is the neutron current. This reduces the neutron transport problem into a neutron diffusion problem, which is much easier to solve—not to mention that diffusion problems have been well-studied in many other fields [6]. The diffusion coefficient D is often defined as the isotropic-scattering $1/3\Sigma_t$ or the transport-corrected formulation $1/3\Sigma_{tr}$, where Σ_t and Σ_{tr} are the total and transport cross-section, respectively. However, definitions for the neutron diffusion coefficient that preserve physical properties of neutron transport are typically the most effective [42].

Fick’s Law or Eq. (3.1) is a reasonable approximation for many reactor problems. However, solutions of the neutron diffusion equation may be inaccurate near boundaries or interfaces of materials with very different neutronic properties, and in regions with strong absorption, high neutron streaming, highly anisotropic scattering, or an independent neutron source [2]. Despite its lack of resolution in angle, neutron diffusion well captures the nearly isotropic features of the neutron flux distributions. This further extends its usefulness as a basis for accelerating neutron transport iteration methods [43][4][5].

In this chapter, the application of MBTD to the neutron diffusion equation is investigated; in particular, its numerical formulation and efficiency are compared to those of BE. In Section 3.1, we present the time-dependent neutron diffusion equation and its other forms, including steady-state and eigenvalue problems. A representative mono-energetic 1-D slab problem is considered. This simple problem yields informative results, and its numerical analysis and results reveal strong incentives for generalization to more practical problems. In Section 3.2, the spatial discretization Finite Difference Method (FDM) is presented. The application of the time-stepping θ-Method (in particular BE) is discussed in Section 3.3, which is followed by that of MBTD in Section 3.4. A stability analysis for the discussed methods is performed in Section 3.5. In Section 3.6, standard and traditional approximations for the delayed neutron source are presented, and their adaptations for MBTD are formulated. Four analytical test problems are devised in Section 0 to verify the characteristics and assess the efficiency of the methods. Numerical results are discussed in Section 0. Finally, Section 3.9 summarizes the chapter.

3.1 Neutron Diffusion Equations

We consider a mono-energetic 1D-slab time-dependent neutron diffusion with one delayed neutron precursor:

$$\begin{cases} \frac{1}{v}\frac{\partial \phi}{\partial t} - \frac{\partial}{\partial x} D(x,t)\frac{\partial \phi}{\partial x} + \Sigma_a(x,t)\phi(x,t) = \nu_p \Sigma_f(x,t)\phi(x,t) + \lambda C(x,t) + Q(x,t), \\ \qquad \frac{\partial C}{\partial t} + \lambda C(x,t) = \nu_d \Sigma_f(x,t)\phi(x,t), \qquad x \in [0,X], \qquad t > 0, \end{cases} \tag{3.2}$$

$$\nu = \nu_p + \nu_d, \tag{3.3}$$

with *albedo* (or reflectance) boundary conditions

$$\frac{J^+(0,t)}{J^-(0,t)} = \alpha_L(t), \qquad \frac{J^-(X,t)}{J^+(X,t)} = \alpha_R(t), \qquad t > 0, \tag{3.4}$$

$$J^\pm(x,t) = \frac{\phi(x,t)}{4} \mp \frac{D(x,t)}{2}\frac{\partial \phi}{\partial x}\,, \tag{3.5}$$

and initial conditions

$$\phi(x,0) = \phi_{init}(x), \qquad C(x,0) = C_{init}(x), \qquad x \in [0,X]. \tag{3.6}$$

Eq. (3.2) describes the balance equations for the *neutron flux* $\phi(x,t)$ and the *delayed neutron* *precursor concentration* $C(x,t)$. In the neutron balance equation, we see that neutrons diffuse in space with *diffusion coefficient* $D(x,t)$, are absorbed with *absorption cross-section* $\Sigma_a(x,t)$, and are produced via *prompt fission production* with cross-section $\nu_p\Sigma_f(x,t)$, from a *delayed source* by means of precursor radioactive decay $\lambda C(x,t)$, and an *independent source* $Q(x,t)$. Considering the precursor balance equation, precursors decay with *precursor decay constant* λ and are produced via *delayed fission production* with cross-section $\nu_d\Sigma_f(x,t)$.

In operator form, Eq. (3.2) becomes

$$\begin{cases} \frac{1}{v}\frac{\partial \phi}{\partial t} + \boldsymbol{M}(x,t)\phi(x,t) = \boldsymbol{F_p}(x,t)\phi(x,t) + \lambda C(x,t) + Q(x,t), \\ \frac{\partial C}{\partial t} + \lambda C(x,t) = \boldsymbol{F_d}(x,t)\phi(x,t), \end{cases} \tag{3.7}$$

with *migration operator*

$$\boldsymbol{M}(x,t) = -\frac{\partial}{\partial x}D(x,t)\frac{\partial}{\partial x} + \Sigma_a(x,t), \tag{3.8}$$

and *prompt* and *delayed fission production operators*

$$\boldsymbol{F_p}(x,t) = \nu_p\Sigma_f(x,t), \qquad \boldsymbol{F_d}(x,t) = \nu_d\Sigma_f(x,t), \tag{3.9}$$

respectively.

It is instructive to consider a time-dependent neutron diffusion equation *without delayed neutrons*. One way to achieve this is to assume $\partial C/\partial t \ll \lambda C(x,t)$ (instantaneous precursor decay) in Eq. (3.2):

$$\frac{1}{v}\frac{\partial \phi}{\partial t} + \boldsymbol{M}(x,t)\phi(x,t) = \boldsymbol{F}(x,t)\phi(x,t) + Q(x,t), \tag{3.10}$$

$$\boldsymbol{F}(x,t) = \boldsymbol{F_p}(x,t) + \boldsymbol{F_d}(x,t). \tag{3.11}$$

A similar problem without delayed neutrons can be obtained by assuming that $\nu_d = 0$ and $C_{init}(x) = 0$, or that $\lambda C(x,t)$ is negligible to the rest of the terms in the neutron balance equation. In Section 3.3 and 3.4, the θ-Method and MBTD are respectively applied to solve this time-dependent problem without delayed neutrons [Eq. (3.10)]. Later in Section 3.6, the methods are generalized to include the delayed neutron precursor balance equation for solving the actual problem of Eq. (3.2).

In Section 2.4, it is mentioned that one of the significant merits of using time-stepping methods is that we can leverage the standard steady-state operator solver, which is typically represented in the form of Eq. (2.14). The *steady-state* neutron diffusion equation is derived by setting $\partial\phi/\partial t \approx 0$ and $\partial C/\partial t \approx 0$ in Eq. (3.2):

$$\boldsymbol{M}(x)\phi(x) = \boldsymbol{F}(x)\phi(x) + Q(x), \tag{3.12}$$

$$C(x) = \frac{\boldsymbol{F_d}(x)\phi(x)}{\lambda}. \tag{3.13}$$

Additionally, let us consider the *eigenvalue* diffusion equations. The usual k-*eigenvalue* diffusion equation is derived by setting $\partial\phi/\partial t = 0$, $\partial C/\partial t = 0$, $Q(x) = 0$, and introducing a scaling factor $1/k$ (inverse of the eigenvalue) in front of the fission neutron sources in Eq. (3.2):

$$\boldsymbol{M}(x)\phi(x) = \frac{1}{k}\boldsymbol{F}(x)\phi(x), \tag{3.14}$$

$$C(x) = \frac{1}{k}\frac{\boldsymbol{F_d}(x)\phi(x)}{\lambda}. \tag{3.15}$$

The α-*eigenvalue* (or time-eigenvalue) diffusion equation is derived by assuming a solution in the form of $\phi(x,t) = \phi(x)e^{\alpha t}$ and $C(x,t) = C(x)e^{\alpha t}$ for Eq. (3.2), and removing the independent source $Q(x,t)$:

$$\left[\boldsymbol{M}(x) + \frac{\alpha}{v}\right]\phi(x) = \boldsymbol{F_p}(x)\phi(x) + \frac{\lambda}{\alpha+\lambda}\boldsymbol{F_d}(x)\phi(x), \tag{3.16}$$

$$C(x) = \frac{\boldsymbol{F_d}(x)\phi(x)}{\alpha+\lambda}. \tag{3.17}$$

For specific cases in which $|\alpha| \gg \lambda$, the simpler *prompt* α-*eigenvalue* diffusion equation is preferred instead:

$$\left[\boldsymbol{M}(x)+\frac{\alpha}{v}\right]\phi(x)=\boldsymbol{F_p}(x)\phi(x). \tag{3.18}$$

The k-eigenvalue Eq. (3.14) is useful in determining the off-criticality of a system (how far k deviates from unity), while the α-eigenvalue Eqs. (3.16) and (3.18) represent the asymptotic-in-time behavior of neutrons in the system. We note that $\alpha > 0$ when a system is *supercritical* ($k > 1$), $\alpha < 0$ when a system is *subcritical* ($k < 1$), and $\alpha = 0$ when a system is *critical* ($k = 1$).

3.2 Finite Difference Method

Here we discretize the spatial variable of the neutron diffusion equations. It is instructive to start by considering the steady-state problem shown in Eqs. (3.12) and (3.13):

$$-\frac{d}{dx}D(x)\frac{d\phi}{dx}+\Sigma_a(x)\phi(x)=\nu\Sigma_f(x)\phi(x)+Q(x), \tag{3.19}$$

$$C(x)=\frac{1}{\lambda}\nu_d\Sigma_f(x)\phi(x), \tag{3.20}$$

with boundary conditions

$$\frac{J^+(0)}{J^-(0)}=\alpha_L, \qquad \frac{J^-(X)}{J^+(X)}=\alpha_R, \tag{3.21}$$

$$J^\pm(x)=\frac{\phi(x)}{4}\mp\frac{D(x)}{2}\frac{d\phi}{dx}. \tag{3.22}$$

Let us consider a system with piece-wise cross-sections within non-uniform spatial meshes of size Δx_j:

$$\begin{cases}\Sigma_*(x)=\Sigma_{*,j}, & x\in\left[x_{j-1/2},x_{j+1/2}\right], \quad j=1,2,\dots,J, \quad \Sigma_*=\Sigma_a,\nu\Sigma_f,D,\dots, \\ x_{1/2}=0, & x_{J+1/2}=X, \quad x_{j+1/2}=x_{j-1/2}+\Delta x_j, \quad \sum_{j=1}^{J}\Delta x_j=X.\end{cases} \tag{3.23}$$

Then, we define the *cell-averaged* quantities:

$$\phi_j = \frac{1}{\Delta x_j}\int_{x_{j-1/2}}^{x_{j+1/2}} \phi(x)dx\,, \qquad C_j = \frac{1}{\Delta x_j}\int_{x_{j-1/2}}^{x_{j+1/2}} C(x)dx\,, \qquad Q_j = \frac{1}{\Delta x_j}\int_{x_{j-1/2}}^{x_{j+1/2}} Q(x)dx\,, \quad (3.24)$$

where subscripts j and $j \pm 1/2$ respectively denote cell-averaged and *cell-edge* quantities. Figure 3.1 illustrates the spatial grid of the numerical solution.

Next, we adapt the *box-scheme* FDM by (1) approximating the cell-edge *neutron currents* with *left- and right-handed derivatives*

$$\left[-D(x)\frac{d\phi}{dx}\right]_{x=x_{j+1/2}} = -D_j\frac{\phi_{j+1/2}-\phi_j}{\Delta x_j/2} = -D_{j+1}\frac{\phi_{j+1}-\phi_{j+1/2}}{\Delta x_{j+1}/2}, \qquad (3.25)$$

and operating Eqs. (3.19) and (3.20) with $\Delta x_j^{-1}\int_{x_{j-1/2}}^{x_{j+1/2}}(\cdot)dx$:

$$-\Sigma_j^{(-)}\phi_{j-1} + \left[\Sigma_{a,j} + \Sigma_j^{(-)} + \Sigma_j^{(+)}\right]\phi_j - \Sigma_j^{(+)}\phi_{j+1} = \nu\Sigma_{f,j}\phi_j + Q_j, \qquad (3.26)$$

$$C_j = \frac{1}{\lambda}\nu_d\Sigma_{f,j}\phi_j, \qquad (3.27)$$

where

$$\Sigma_j^{(+)} = \frac{1}{\Delta x_j}\left(\frac{D}{\Delta x}\right)_{j+1/2}, \qquad \Sigma_j^{(-)} = \frac{1}{\Delta x_j}\left(\frac{D}{\Delta x}\right)_{j-1/2}, \qquad (3.28)$$

$$\left(\frac{D}{\Delta x}\right)_{j+1/2} = \begin{cases} \dfrac{2}{\Delta x_1/D_1}, & j = 0, \\ \dfrac{2}{\Delta x_j/D_j + \Delta x_{j+1}/D_{j+1}}, & j = 1,\dots,J-1, \\ \dfrac{2}{\Delta x_J/D_J}, & j = J, \end{cases} \qquad (3.29)$$

$$\phi_0 = B_L\phi_1, \qquad \phi_{J+1} = B_R\phi_J, \qquad (3.30)$$

$$B_L = \frac{1}{\dfrac{1-\alpha_L}{1+\alpha_L}\dfrac{\Delta x_1/D_1}{4}+1}, \qquad B_R = \frac{1}{\dfrac{1-\alpha_R}{1+\alpha_R}\dfrac{\Delta x_J/D_J}{4}+1}. \qquad (3.31)$$

Re-casting Eqs. (3.26), (3.30), and (3.27) into matrix operator form, we obtain

$$(\boldsymbol{M}-\boldsymbol{F})\boldsymbol{\phi}=\boldsymbol{Q}, \tag{3.32}$$

$$\boldsymbol{C}=\frac{1}{\lambda}\boldsymbol{F_d}\boldsymbol{\phi}, \tag{3.33}$$

$$\boldsymbol{F}=\boldsymbol{F_p}+\boldsymbol{F_d}. \tag{3.34}$$

Here $\boldsymbol{\phi}$, $\boldsymbol{C}$**,** and $\boldsymbol{Q}$ are column vectors of size $(J\times 1)$ with elements of ϕ_j, C_j, and Q_j. $\boldsymbol{F_p}$ and $\boldsymbol{F_d}$ are diagonal matrices of size $(J\times J)$ with elements of $\nu_p\Sigma_{f,j}$ and $\nu_d\Sigma_{f,j}$, respectively. $\boldsymbol{M}$ is a tridiagonal matrix of size $(J\times J)$ with lower and upper diagonals of $-\Sigma_j^{(-)}$ and $-\Sigma_j^{(+)}$, and with diagonal elements of $\Sigma_{a,j}+\Sigma_j^{(-)}+\Sigma_j^{(+)}$, except for the left- and right-most elements which respectively have additional terms of $-\Sigma_1^{(-)}B_L$ and $-\Sigma_J^{(+)}B_R$, representing the two boundary conditions.

Eq. (3.32) is essentially a sparse linear algebra problem of the form $Ax=b$, where $A=\boldsymbol{M}-\boldsymbol{F}$, $x=\boldsymbol{\phi}$, and $b=\boldsymbol{Q}$. Therefore, we can solve for $\boldsymbol{\phi}$ in Eq. (3.32) by using any working linear algebra solver. In this thesis, ILU-preconditioned Krylov method GMRES [44] is used to solve such sparse linear algebra problems. Once the neutron flux solution $\boldsymbol{\phi}$ is obtained, the delayed neutron precursor concentration $\boldsymbol{C}$ can be straightforwardly calculated per Eq. (3.33).

We can apply the same FDM formulation to the time-dependent diffusion equations. Equation (3.2) becomes:

$$\begin{cases}\frac{1}{v}\frac{d\boldsymbol{\phi}}{dt}+\left[\boldsymbol{M}(t)-\boldsymbol{F_p}(t)\right]\boldsymbol{\phi}(t)=\lambda\boldsymbol{C}(t)+\boldsymbol{Q}(t),\\ \qquad\qquad \frac{d\boldsymbol{C}}{dt}+\lambda\boldsymbol{C}(t)=\boldsymbol{F_d}(t)\boldsymbol{\phi}(t),\end{cases} \tag{3.35}$$

$$\boldsymbol{\phi}(0)=\boldsymbol{\phi_{init}},\qquad \boldsymbol{C}(0)=\boldsymbol{C_{init}}. \tag{3.36}$$

Similarly, the time-dependent diffusion without delayed neutrons Eq. (3.10) becomes

$$\frac{1}{v}\frac{d\boldsymbol{\phi}}{dt}+[\boldsymbol{M}(t)-\boldsymbol{F}(t)]\boldsymbol{\phi}(t)=\boldsymbol{Q}(t). \tag{3.37}$$

The time-dependent vectors and matrices in Eqs. (3.35)–(3.37) follow the definition of those in the steady-state problem Eqs. (3.32) and (3.33), except that each of their elements is time-dependent: $\phi_j(t)$, $C_j(t)$, $Q_j(t)$, $\alpha_L(t)$, $\alpha_R(t)$, and $\Sigma_{*,j}(t)$.

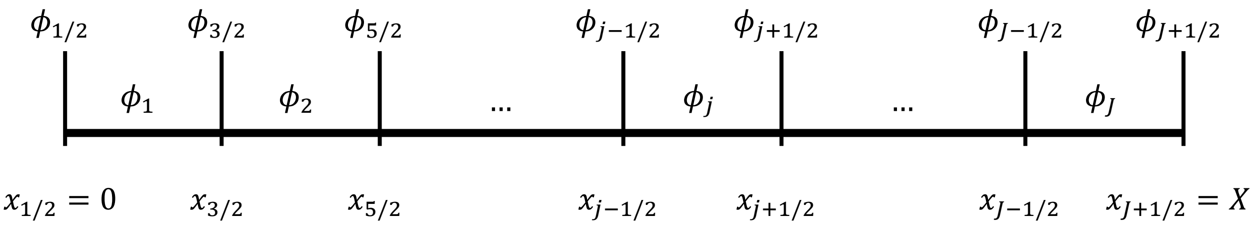


Figure 3.1: Spatial grid for the numerical solution of FDM

Additionally, we can apply the same FDM formulation to the k-eigenvalue [Eq. (3.14)], α-eigenvalue [Eq. (3.16)], and prompt α-eigenvalue [Eq. (3.18)], and respectively obtain

$$\boldsymbol{M\phi} = \frac{1}{k}\boldsymbol{F\phi}, \qquad \boldsymbol{C} = \frac{1}{k}\frac{\boldsymbol{F_d\phi}}{\lambda}, \tag{3.38}$$

$$\left(\boldsymbol{M} + \frac{\alpha}{v}\boldsymbol{I}\right)\boldsymbol{\phi} = \boldsymbol{F_p\phi} + \frac{\lambda}{\alpha+\lambda}\boldsymbol{F_d\phi}, \qquad \boldsymbol{C} = \frac{\boldsymbol{F_d\phi}}{\alpha+\lambda}, \tag{3.39}$$

$$\left(\boldsymbol{M} + \frac{\alpha}{v}\boldsymbol{I}\right)\boldsymbol{\phi} = \boldsymbol{F_p\phi}. \tag{3.40}$$

3.3 θ-Method

Here we apply the time-stepping θ-Method [8][4] to the neutron diffusion equation without delayed neutrons shown in Eq. (3.37). In a time-stepping method, we set the following:

$$\boldsymbol{\phi_k} = \boldsymbol{\phi}(t_k), \qquad \boldsymbol{Q_k} = \boldsymbol{Q}(t_k), \qquad k = 0, 1, \dots, \tag{3.41}$$

$$t_0 = 0, \qquad t_k = t_{k-1} + \Delta t_k, \tag{3.42}$$

$$\boldsymbol{\phi_0} = \boldsymbol{\phi_{init}}. \tag{3.43}$$

Now, let us re-cast Eq. (3.37) into

$$\frac{1}{v}\frac{d\boldsymbol{\phi}}{dt} = f_{diff}[t, \boldsymbol{\phi}(t)], \tag{3.44}$$

where the *diffusion operator* is defined as

$$f_{diff}[t, \boldsymbol{\phi}(t)] = -[\boldsymbol{M}(t) - \boldsymbol{F}(t)]\boldsymbol{\phi}(t) + \boldsymbol{Q}(t). \tag{3.45}$$

The θ-Method solves Eq. (3.44) as follows:

$$\frac{1}{v}\frac{\boldsymbol{\phi_k} - \boldsymbol{\phi_{k-1}}}{\Delta t_k} = (1-\theta)f_{diff}(t_{k-1}, \boldsymbol{\phi}_{k-1}) + \theta f_{diff}(t_k, \boldsymbol{\phi}_k). \tag{3.46}$$

For BE ($\theta = 1$),

$$\left(\boldsymbol{M_k} - \boldsymbol{F_k} + \frac{1}{v\Delta t_k}\boldsymbol{I}\right)\boldsymbol{\phi_k} = \boldsymbol{Q_{eff,k}}, \tag{3.47}$$

with the *effective independent source* at t_k defined as

$$\boldsymbol{Q_{eff,k}} = \boldsymbol{Q_k} + \frac{1}{v\Delta t_k}\boldsymbol{\phi_{k-1}}. \tag{3.48}$$

We note that the time-index subscript k similarly applies to the operators: $\boldsymbol{M_k} = \boldsymbol{M}(t_k)$ and $\boldsymbol{F_k} = \boldsymbol{F}(t_k)$.

It is evident that for FE ($\theta = 0$), the next time-step solution $\boldsymbol{\phi_k}$ can be obtained via straightforward matrix and vector operations based on the previous time-step solution $\boldsymbol{\phi_{k-1}}$ and the cross-section data at the previous time t_{k-1}. For the implicit methods BE [Eq. (3.47)] and CN ($\theta = 1/2$), a linear operator solve needs to be performed to obtain the solution $\boldsymbol{\phi_k}$. However, those linear problems have the same form as that of the tridiagonal steady-state problem [Eq. (3.32)], except for the additional *time-absorption* ($1/v\Delta t_k$) term and the effective independent source. In other words, the implicit methods BE and CN (or implicit time-stepping methods in general) reduce a time-dependent problem to solving a steady-state problem for each time step. In this thesis, ILU-preconditioned GMRES is used to solve the sparse linear BE systems [Eq. (3.47)].

3.4 MBTD

Here we apply MBTD to the time-dependent neutron diffusion equation without delayed neutrons shown in Eq. (3.37), and then compare the resulting numerical method to that of BE [Eq. (3.47)]. In addition to the time-stepping method setup shown in Eqs. (3.41)–(3.45), we define the mid-point quantities:

$$\boldsymbol{\phi}_{\boldsymbol{k-1/2}} = \boldsymbol{\phi}(t_{k-1/2}), \qquad \boldsymbol{Q}_{\boldsymbol{k-1/2}} = \boldsymbol{Q}(t_{k-1/2}), \qquad k = 1, 2, \dots, \tag{3.49}$$

where $t_{k-1/2} = t_{k-1} + \Delta t_k/2$. MBTD solves Eq. (3.44) as follows:

$$\begin{cases} \dfrac{1}{v}\dfrac{\boldsymbol{\phi}_{\boldsymbol{k}} - \boldsymbol{\phi}_{\boldsymbol{k-1}}}{\Delta t_k} = f_{diff}\left[t_{k-1/2}, \boldsymbol{\phi}_{\boldsymbol{k-1/2}}\right], \\ \quad \dfrac{1}{v}\dfrac{\boldsymbol{\phi}_{\boldsymbol{k}} - \boldsymbol{\phi}_{\boldsymbol{k-1/2}}}{\Delta t_k/2} = f_{diff}[t_k, \boldsymbol{\phi}_{\boldsymbol{k}}], \end{cases} \tag{3.50}$$

which gives

$$\begin{cases} \left(\boldsymbol{M}_{\boldsymbol{k-1/2}} - \boldsymbol{F}_{\boldsymbol{k-1/2}}\right)\boldsymbol{\phi}_{\boldsymbol{k-1/2}} + \dfrac{1}{v\Delta t_k}\boldsymbol{\phi}_{\boldsymbol{k}} = \boldsymbol{Q}_{\boldsymbol{eff,k-1/2}}, \\ -\dfrac{2}{v\Delta t_k}\boldsymbol{\phi}_{\boldsymbol{k-1/2}} + \left(\boldsymbol{M}_{\boldsymbol{k}} - \boldsymbol{F}_{\boldsymbol{k}} + \dfrac{2}{v\Delta t_k}\boldsymbol{I}\right)\boldsymbol{\phi}_{\boldsymbol{k}} = \boldsymbol{Q}_{\boldsymbol{eff,k}}, \end{cases} \tag{3.51}$$

with effective independent sources defined as

$$\boldsymbol{Q}_{\boldsymbol{eff,k-1/2}} = \boldsymbol{Q}_{\boldsymbol{k-1/2}} + \frac{1}{v\Delta t_k}\boldsymbol{\phi}_{\boldsymbol{k-1}}, \qquad \boldsymbol{Q}_{\boldsymbol{eff,k}} = \boldsymbol{Q}_{\boldsymbol{k}}. \tag{3.52}$$

Eq. (3.51) is a fully-discretized MBTD coupled balance equation for the time-dependent neutron diffusion without delayed neutrons Eq. (3.10).

We now consider the three strategies for solving MBTD coupled balance equations discussed in Section 2.4: (1) Substitution, (2) Iterative Solve, and (3) Simultaneous Solve. With Substitution, Eq. (3.51) reduces to

$$\left[\frac{v\Delta t_k}{2}(\boldsymbol{M}-\boldsymbol{F})^* + \left(\boldsymbol{M}_{\boldsymbol{k-1/2}} - \boldsymbol{F}_{\boldsymbol{k-1/2}} + \frac{1}{v\Delta t_k}\boldsymbol{I}\right)\right]\boldsymbol{\phi}_{\boldsymbol{k}} = \boldsymbol{Q}^*_{\boldsymbol{eff}}, \tag{3.53}$$

where

$$(\boldsymbol{M}-\boldsymbol{F})^* = \left(\boldsymbol{M}_{\boldsymbol{k-1/2}} - \boldsymbol{F}_{\boldsymbol{k-1/2}}\right)(\boldsymbol{M}_{\boldsymbol{k}} - \boldsymbol{F}_{\boldsymbol{k}}), \tag{3.54}$$

$$\boldsymbol{Q}^*_{\boldsymbol{eff}} = \boldsymbol{Q}_{\boldsymbol{eff,k-1/2}} + \left(\boldsymbol{M}_{\boldsymbol{k-1/2}} - \boldsymbol{F}_{\boldsymbol{k-1/2}}\right)\frac{v\Delta t_k}{2}\boldsymbol{Q}_{\boldsymbol{eff,k}}. \tag{3.55}$$

The operator $(\boldsymbol{M}-\boldsymbol{F})^*$ represents the square of the steady-state operator, which in this case is a fourth derivative operator having five-diagonal elements. We apply the migration and fission operator to the independent source, which is a consequence of squaring the whole operator: each term operates on itself and all the other terms. In this thesis, the resulting five-diagonal matrix problem of Eq. (3.53) is solved with ILU-preconditioned GMRES. Compared to BE [Eq. (3.47)], the major additional work of MBTD with the Substitution strategy [Eq. (3.53)] includes (1) preparing the five-diagonal matrix, which involves multiplying two tridiagonal steady-state operator matrices, and (2) solving a five-, instead of tri-, diagonal linear problem.

The Lagged $O(\Delta t)$ Iterative Solve strategy is considered. It reduces Eq. (3.51) to

$$\begin{cases} \left(\boldsymbol{M}_{\boldsymbol{k-1/2}} - \boldsymbol{F}_{\boldsymbol{k-1/2}} + \frac{1}{v\Delta t_k}\boldsymbol{I}\right)\boldsymbol{\phi}^{(l)}_{\boldsymbol{k-1/2}} = \boldsymbol{Q}_{\boldsymbol{eff,k-1/2}} - \frac{1}{v\Delta t_k}\left(\boldsymbol{\phi}^{(l-1)}_{\boldsymbol{k}} - \boldsymbol{\phi}^{(l-1)}_{\boldsymbol{k-1/2}}\right), \\ \left(\boldsymbol{M}_{\boldsymbol{k}} - \boldsymbol{F}_{\boldsymbol{k}} + \frac{2}{v\Delta t_k}\boldsymbol{I}\right)\boldsymbol{\phi}^{(l)}_{\boldsymbol{k}} = \boldsymbol{Q}_{\boldsymbol{eff,k}} + \frac{2}{v\Delta t_k}\boldsymbol{\phi}^{(l)}_{\boldsymbol{k-1/2}}. \end{cases} \tag{3.56}$$

Both equations are tridiagonal and similarly structured like that of the BE Eq. (3.47). It is evident that the memory complexity of this strategy is twice that of BE, and the additional work highly depends on the number of iterations required to converge the solutions $\boldsymbol{\phi}_{\boldsymbol{k-1/2}}$ and $\boldsymbol{\phi}_{\boldsymbol{k}}$. The convergence rate of the iteration scheme is estimated to correspond to the spectral radius of Lagged $O(\Delta t)$ shown in Figure 2.4.

Eq. (3.51) is structurally similar to the one shown in Figure 2.5, with $\boldsymbol{L}(t)$ in the form of tridiagonal steady-state operator [Eq. (3.32)]. With the Simultaneous Solve strategy, the unknowns $\boldsymbol{\phi}_{\boldsymbol{k-1/2}}$ and $\boldsymbol{\phi}_{\boldsymbol{k}}$ are simultaneously obtained by solving the four-block matrix of Eq.

(3.51) using a linear solver. Compared to BE, the additional work of MBTD with Simultaneous Solve strategy depends on how much faster the linear solver solves the tridiagonal problem of BE [Eq. (3.47)] compared to the four-block problem of MBTD [Eq. (3.51)].

3.5 Von Neumann Stability Analysis

In this section, Von Neumann analysis [6] is used to characterize the stability of the time-dependent neutron diffusion methods: (1) θ-Method with FDM, and (2) MBTD with FDM. Von Neumann analysis is based on Fourier analysis and is typically used to determine the stability of discretized linear PDEs with constant coefficients.

Let us consider the time-dependent neutron diffusion equation without delayed neutrons for a *non-supercritical* infinite medium:

$$\frac{1}{v}\frac{\partial\phi}{\partial t}-D\frac{\partial^2\phi}{\partial x^2}+\Sigma_a\phi(x,t)=\nu\Sigma_f\phi(x,t), \qquad x\in[-\infty,\infty], \qquad t>0, \tag{3.57}$$

where $\Sigma_a \geq \nu\Sigma_f$. The problem can be rearranged in terms of non-dimensional coefficients as

$$\frac{1}{v\Sigma_t}\frac{\partial\phi}{\partial t}=\frac{D}{\Sigma_t}\frac{\partial^2\phi}{\partial x^2}+\zeta\phi(x,t), \tag{3.58}$$

where ζ is a measure of off-criticality, which is defined as

$$\zeta=(k-1)(1-c), \tag{3.59}$$

and depends on *criticality* (or *multiplication factor*) $k=\nu\Sigma_f/\Sigma_a$ and *scattering ratio* $c=\Sigma_s/\Sigma_t$. For this stability analysis, since $\Sigma_a\geq\nu\Sigma_f$, we note that $-1\leq\zeta\leq 0$.

Applying FE and FDM, we obtain the following:

$$\frac{1}{\eta}\left(\phi_{k,j}-\phi_{k-1,j}\right)=\frac{\varsigma}{\tau^2}\left(\phi_{k-1,j-1}-2\phi_{k-1,j}+\phi_{k-1,j+1}\right)+\zeta\phi_{k-1,j}, \tag{3.60}$$

where we introduce the non-dimensional parameter $\eta=v\Sigma_t\Delta t$, and define two new non-dimensional parameters: $\tau=\Sigma_t\Delta x$ and $\varsigma=D\Sigma_t$, which are the spatial-mesh size Δx and diffusion coefficient D in the unit of optical thickness, respectively. Next, we introduce the

Fourier ansatz:

$$\phi_{k,j} = A(\omega)e^{i\omega\Sigma_t\Delta xj}, \qquad \phi_{k-1,j} = e^{i\omega\Sigma_t\Delta xj}, \tag{3.61}$$

where $A(\omega)$ is the *amplification factor* corresponding to *Fourier mode* ω. Substituting Eq. (3.61) into Eq. (3.60), we obtain

$$A(\omega) = 1 - z, \tag{3.62}$$

where

$$z = -\left\{\zeta\eta + \varsigma\frac{2\eta}{\tau^2}[\cos(\omega\tau) - 1]\right\}. \tag{3.63}$$

Since $-1 \le \zeta \le 0$ and $-1 \le \cos(\omega\tau) \le 1$ for any value of ω, we note that

$$0 \le -\zeta\eta \le z \le -\zeta\eta + \varsigma\frac{4\eta}{\tau^2}. \tag{3.64}$$

The amplification factor of FE in Eq. (3.62) is similar to the one in Table 2.1, except that we replace η with z, which is defined in Eq. (3.63). The amplification factors for BE, CN and MBTD can be similarly derived; nevertheless, it is worth to mention that the mid-point solution of MBTD can be expressed in terms of the Fourier ansatz:

$$\phi_{k-1/2,j} = \left(1 + \frac{1}{2}z\right)A(\omega)e^{i\omega\Sigma_t\Delta xj}. \tag{3.65}$$

A method is stable if $|A(\omega)| \le 1$, and free of spurious oscillations if $0 \le A(\omega) \le 1$, for all values of ω. By evaluating the inequalities, we can estimate the stability and robustness of a method based on the non-dimensional parameters ζ, ς, and more importantly, η and τ. The results of the Von Neumann stability analysis are summarized in Table 3.1. It is found that there is a severe restriction for FE to achieve stability: η (and thus Δt) must decrease at about the rate of τ^2 as we refine the spatial mesh Δx, and we note that τ^2 is much smaller than τ when τ is small. Furthermore, the same severe restriction applies to CN to guarantee free of spurious oscillations. Finally, it is again theoretically proven that BE and MBTD are unconditionally robust.

Table 3.1: Summary for Von Neumann stability analysis of neutron diffusion

Method	Stability	Free of spurious oscillations
FE	$\varsigma \frac{4\eta}{\tau^2} - \zeta\eta \leq 2$	$\varsigma \frac{4\eta}{\tau^2} - \zeta\eta \leq 1$
BE	Unconditional	Unconditional
CN		$\varsigma \frac{4\eta}{\tau^2} - \zeta\eta \leq 2$
MBTD		Unconditional

3.6 Approximations for Delayed Source

In this section, we generalize the formulation of BE [Eq. (3.47)] and MBTD [Eq. (3.51)] to include the delayed neutron precursor equation. Let us recast the time-dependent neutron diffusion with delayed neutrons Eq. (3.35) into

$$\begin{cases} \frac{1}{v}\frac{d\boldsymbol{\phi}}{dt} = f_{diff}[t, \boldsymbol{\phi}(t), \boldsymbol{C}(t)], \\ \frac{d\boldsymbol{C}}{dt} = -\lambda \boldsymbol{C}(t) + \boldsymbol{F_d}(t)\boldsymbol{\phi}(t), \end{cases} \tag{3.66}$$

where the diffusion operator is defined as

$$f_{diff}[t, \boldsymbol{\phi}(t), \boldsymbol{C}(t)] = -\left[\boldsymbol{M}(t) - \boldsymbol{F_p}(t)\right]\boldsymbol{\phi}(t) + \lambda\boldsymbol{C}(t) + \boldsymbol{Q}(t). \tag{3.67}$$

3.6.1 Standard and Traditional Approximations

By applying BE to the neutron equation in Eq. (3.66), one obtains

$$\left(\boldsymbol{M_k} - \boldsymbol{F_{p,k}} + \frac{1}{v\Delta t_k}\boldsymbol{I}\right)\boldsymbol{\phi_k} = \boldsymbol{Q_k} + \frac{1}{v\Delta t_k}\boldsymbol{\phi_{k-1}} + \lambda\boldsymbol{C_k}. \tag{3.68}$$

Three approximations for the delayed source $\lambda\boldsymbol{C_k}$ are considered: (1) *standard* BE formulation, and the traditionally used (2) *linear* and (3) *quadratic* delayed fission production [8][5][4]. By applying the standard BE formulation to the precursor equation in Eq. (3.66), one obtains

$$\lambda \boldsymbol{C}_{\boldsymbol{k}} = \frac{1}{1+\lambda\Delta t_k}\lambda \boldsymbol{C}_{\boldsymbol{k-1}} + \frac{\lambda\Delta t_k}{1+\lambda\Delta t_k}\boldsymbol{F}_{\boldsymbol{d,k}}\boldsymbol{\phi}_{\boldsymbol{k}}. \tag{3.69}$$

On the other hand, the two traditional approaches analytically solve the precursor equation in Eq. (3.66), which is essentially a heterogeneous first-order ODE,

$$\lambda \boldsymbol{C}_k = e^{-\lambda\Delta t_k}\lambda \boldsymbol{C}_{\boldsymbol{k-1}} + \lambda\int_{t_{k-1}}^{t_k} e^{-\lambda(t_k-t')}\boldsymbol{F}_{\boldsymbol{d}}(t')\boldsymbol{\phi}(t')dt', \tag{3.70}$$

and approximate the delayed fission production $\boldsymbol{F}_{\boldsymbol{d}}(t')\boldsymbol{\phi}(t')$ as a linear or quadratic function:

$$\boldsymbol{F}_{\boldsymbol{d}}(t')\phi(t') \approx \frac{t'-t_k}{-\Delta t_k}\boldsymbol{F}_{\boldsymbol{d,k-1}}\boldsymbol{\phi}_{\boldsymbol{k-1}} + \frac{t'-t_{k-1}}{\Delta t_k}\boldsymbol{F}_{\boldsymbol{d,k}}\boldsymbol{\phi}_{\boldsymbol{k}}, \tag{3.71}$$

or

$$\begin{aligned}\boldsymbol{F}_{\boldsymbol{d}}(t')\phi(t') \approx &\left[\frac{(t'-t_{k-1})(t'-t_k)}{\Delta t_{k-1}(\Delta t_{k-1}+\Delta t_k)}\right]\boldsymbol{F}_{\boldsymbol{d,k-2}}\boldsymbol{\phi}_{\boldsymbol{k-2}}\\ &+\left[\frac{(t'-t_{k-2})(t'-t_k)}{-\Delta t_{k-1}\Delta t_k}\right]\boldsymbol{F}_{\boldsymbol{d,k-1}}\boldsymbol{\phi}_{\boldsymbol{k-1}} + \left[\frac{(t'-t_{k-2})(t'-t_{k-1})}{(\Delta t_{k-1}+\Delta t_k)\Delta t_k}\right]\boldsymbol{F}_{\boldsymbol{d,k}}\boldsymbol{\phi}_{\boldsymbol{k}}.\end{aligned} \tag{3.72}$$

We note that the quadratic approximation Eq. (3.72) introduces a slight additional complexity to keep track of the quantities at t_{k-2}.

The resulting BE formulation with the three delayed neutron approximations can be represented in the following general form:

$$\left(\boldsymbol{M}_{\boldsymbol{k}} - \boldsymbol{F}_{\boldsymbol{eff,k}} + \frac{1}{v\Delta t_k}\boldsymbol{I}\right)\boldsymbol{\phi}_{\boldsymbol{k}} = \boldsymbol{Q}_{\boldsymbol{eff,k}}, \tag{3.73}$$

$$\boldsymbol{Q}_{\boldsymbol{eff,k}} = \boldsymbol{Q}_{\boldsymbol{k}} + \frac{1}{v\Delta t_k}\boldsymbol{\phi}_{\boldsymbol{k-1}} + \sum_{k'=k-2}^{k-1}\xi_{k'\to k}\boldsymbol{F}_{\boldsymbol{d,k'}}\boldsymbol{\phi}_{\boldsymbol{k'}} + \xi_{C\to k}\lambda\boldsymbol{C}_{\boldsymbol{k-1}}, \tag{3.74}$$

where the *effective fission production operator* $\boldsymbol{F}_{\boldsymbol{eff,k}}$ is defined as

$$\boldsymbol{F}_{\boldsymbol{eff,k}} = \boldsymbol{F}_{\boldsymbol{p,k}} + \xi_{k\to k}\boldsymbol{F}_{\boldsymbol{d,k}}, \tag{3.75}$$

and the constants $\xi_{k'\to k}$ and $\xi_{C\to k}$ are provided in Table 3.2. Finally, once the neutron flux solution $\boldsymbol{\phi_k}$ is obtained, the precursor solution $\boldsymbol{C_k}$ can be calculated as follows:

$$\boldsymbol{C_k} = \xi_{C\to k}\boldsymbol{C_{k-1}} + \frac{1}{\lambda}\sum_{k'=k-2}^{k} \xi_{k'\to k}\boldsymbol{F_{d,k'}}\boldsymbol{\phi_{k'}}\,. \tag{3.76}$$

The constant $\xi_{k'\to k}$ can be physically defined as the *effective contribution factor* of delayed fission production at $t_{k'}$ to delayed neutron source at t_k—i.e., the effective contribution of $\boldsymbol{F_{d,k'}\phi_{k'}}$ to $\lambda\boldsymbol{C_k}$. Similarly, $\xi_{C\to k}$ is the effective contribution factor of the previous time step delayed source $\lambda\boldsymbol{C_{k-1}}$. Even though the contribution factors of the linear and quadratic delayed fission production approximations are considerably more complicated than those of the standard BE (as observed in Table 3.2), they are expected to be more accurate because they exactly represent the contribution factor of $\lambda\boldsymbol{C_{k-1}}$, which is an exponential decay.

Table 3.2: Delayed source contribution factors of BE

	Standard BE	Linear $\boldsymbol{F_d\phi}$	Quadratic $\boldsymbol{F_d\phi}$
$\xi_{k-2\to k}$	0	0	$\frac{\Delta t_k}{(\Delta t_{k-1}+\Delta t_k)\lambda\Delta t_{k-1}}\left[\frac{2}{\lambda\Delta t_k}\left(1-e^{-\lambda\Delta t_k}\right) -\left(1+e^{-\lambda\Delta t_k}\right)\right]$
$\xi_{k-1\to k}$	0	$\frac{1-e^{-\lambda\Delta t_k}}{\lambda\Delta t_k} - e^{-\lambda\Delta t_k}$	$\frac{1}{\lambda\Delta t_{k-1}}\left[\left(1+e^{-\lambda\Delta t_k}\right)-e^{-\lambda\Delta t_k} +\left(\frac{\Delta t_{k-1}}{\Delta t_k}-\frac{2}{\lambda\Delta t_k}\right)\left(1-e^{-\lambda\Delta t_k}\right)\right]$
$\xi_{k\to k}$	$\frac{\lambda\Delta t_k}{1+\lambda\Delta t_k}$	$1-\frac{1-e^{-\lambda\Delta t_k}}{\lambda\Delta t_k}$	$1+\frac{1}{(\Delta t_{k-1}+\Delta t_k)\lambda}\left[\left(\frac{2}{\lambda\Delta t_k}-\frac{\Delta t_{k-1}}{\Delta t_k}\right)\left(1-e^{-\lambda\Delta t_k}\right) -2\right]$
$\xi_{C\to k}$	$\frac{1}{1+\lambda\Delta t_k}$	$e^{-\lambda\Delta t_k}$	$e^{-\lambda\Delta t_k}$

It is evident that the precursor equation is relatively simple such that the delayed source $\lambda\boldsymbol{C_k}$ can be explicitly represented as a function of the unknown neutron flux $\boldsymbol{\phi_k}$ which then can be substituted into the neutron equation. This allows us to reduce the neutron diffusion problem with delayed neutrons to Eq. (3.73), which has the same form as the one without delayed neutrons in Eq. (3.47), except for the additional terms which correspond to the effective delayed source and are collapsible into the effective independent source $\boldsymbol{Q_{eff}}$ and the effective fission production operator $\boldsymbol{F_{eff}}$. Ultimately, again, we manage to reduce the time-dependent problem into solving a form of steady-state problem Eq. (3.32) at each time step.

3.6.2 Adaptation for MBTD

Now we adapt the delayed source approximations described in the previous subsection for MBTD. By applying the MBTD formulation to the neutron equation in Eq. (3.66), one obtains

$$\begin{cases} (\boldsymbol{M_{k-1/2}} - \boldsymbol{F_{p,k-1/2}})\boldsymbol{\phi_{k-1/2}} + \dfrac{1}{v\Delta t_k}\boldsymbol{\phi_k} = \boldsymbol{Q_{k-1/2}} + \dfrac{1}{v\Delta t_k}\boldsymbol{\phi_{k-1}} + \lambda\boldsymbol{C_{k-1/2}}, \\ -\dfrac{2}{v\Delta t_k}\boldsymbol{\phi_{k-1/2}} + \left(\boldsymbol{M_k} - \boldsymbol{F_{p,k}} + \dfrac{2}{v\Delta t_k}\boldsymbol{I}\right)\boldsymbol{\phi_k} = \boldsymbol{Q_k} + \lambda\boldsymbol{C_k}. \end{cases} \quad (3.77)$$

It is found that MBTD coupled balance equations need to approximate not only $\lambda\boldsymbol{C_k}$ but also $\lambda\boldsymbol{C_{k-1/2}}$.

The previously-described approaches are considered: (1) the *standard* MBTD formulation, and (2) the *linear* and (3) *quadratic* delayed fission production. By applying the standard MBTD formulation to the precursor equation in Eq. (3.66), one can represent $\lambda\boldsymbol{C_k}$ and $\lambda\boldsymbol{C_{k-1/2}}$ as functions of $\boldsymbol{\phi_k}$ and $\boldsymbol{\phi_{k-1/2}}$. In the later approximations, in addition to the analytical expression of $\lambda\boldsymbol{C_k}$ in Eq. (3.70), we can also derive

$$\lambda\boldsymbol{C}_k = e^{-\lambda\Delta t_k/2}\lambda\boldsymbol{C_{k-1/2}} + \lambda\int_{t_{k-1/2}}^{t_k} e^{-\lambda(t_k-t')}\boldsymbol{F_d}(t')\boldsymbol{\phi}(t')dt'\,. \quad (3.78)$$

In the linear delayed fission production approach, similar to BE, MBTD uses the linear interpolation in Eq. (3.71). However, the quadratic delayed fission approach comes more naturally in MBTD. This is because instead of using and keeping track of the quantities at t_{k-2} as in Eq. (3.72), $\boldsymbol{F_{d,k-1/2}}\boldsymbol{\phi_{k-1/2}}$ can be used for the quadratic interpolation:

$$\begin{aligned}\boldsymbol{F_d}(t')\phi(t') \approx & \left[\frac{\big(t'-t_{k-1/2}\big)(t'-t_k)}{\Delta t_k^2/2}\right]\boldsymbol{F_{d,k-1}\phi_{k-1}} \\ & + \left[\frac{(t'-t_{k-1})(t'-t_k)}{-\Delta t_k^2/4}\right]\boldsymbol{F_{d,k-1/2}\phi_{k-1/2}} + \left[\frac{(t'-t_{k-1})\big(t'-t_{k-1/2}\big)}{\Delta t_k^2/2}\right]\boldsymbol{F_{d,k}\phi_k}.\end{aligned} \tag{3.79}$$

In addition to the three delayed source approximations described, another approach is developed; we call it MB+ delayed source formulation. This approach is similar to the linear and quadratic delayed fission production approximations in which the precursor heterogeneous ODE in Eq. (3.66) is analytically solved; however, instead of approximating the delayed fission production in Eqs. (3.70) and (3.78), we approximate the integrals by following the MBTD formulation:

$$\int_{t_{k-1}}^{t_k} e^{-\lambda(t_k-t')}\boldsymbol{F_d}(t')\boldsymbol{\phi}(t')dt' \approx \Delta t_k e^{-\lambda\Delta t_k/2}\boldsymbol{F_{d,k-1/2}\phi_{k-1/2}}, \tag{3.80}$$

$$\int_{t_{k-1/2}}^{t_k} e^{-\lambda(t_k-t')}\boldsymbol{F_d}(t')\boldsymbol{\phi}(t')dt' \approx \frac{\Delta t}{2}\boldsymbol{F_{d,k}\phi_k}. \tag{3.81}$$

The resulting MBTD coupled balance equations with the four delayed neutron approximations can be generalized in the following form:

$$\begin{cases} \big[\boldsymbol{M_{k-1/2}} - \boldsymbol{F_{eff,k-1/2}}\big]\boldsymbol{\phi_{k-1/2}} \\ \qquad + \left[\dfrac{1}{v\Delta t_k}\boldsymbol{I} - \xi_{k\to k-1/2}\boldsymbol{F_{d,k}}\right]\boldsymbol{\phi_k} = \boldsymbol{Q_{eff,k-1/2}}, \\ -\left[\dfrac{2}{v\Delta t_k}\boldsymbol{I} + \xi_{k-1/2\to k}\boldsymbol{F_{d,k-1/2}}\right]\boldsymbol{\phi_{k-1/2}} \\ \qquad + \left[\boldsymbol{M_k} - \boldsymbol{F_{eff,k}} + \dfrac{2}{v\Delta t_k}\boldsymbol{I}\right]\boldsymbol{\phi_k} = \boldsymbol{Q_{eff,k}}, \end{cases} \tag{3.82}$$

$$\boldsymbol{Q_{eff,k-1/2}} = \boldsymbol{Q_{k-1/2}} + \left[\frac{1}{v\Delta t_k}\boldsymbol{I} + \xi_{k-1\to k-1/2}\boldsymbol{F_{d,k-1}}\right]\boldsymbol{\phi_{k-1}} + \xi_{C\to k-1/2}\lambda\boldsymbol{C_{k-1}}, \tag{3.83}$$

$$\boldsymbol{Q_{eff,k}} = \boldsymbol{Q_k} + \xi_{k-1\to k}\boldsymbol{F_{d,k-1}\phi_{k-1}} + \xi_{C\to k}\lambda\boldsymbol{C_{k-1}}. \tag{3.84}$$

Table 3.3: Delayed source contribution factors of MBTD w/ standard formulation

	$\xi_{\to k-1/2}$	$\xi_{\to k}$
$\xi_{k-1\to}$	0	0
$\xi_{k-1/2\to}$	$1-\frac{1}{1+\lambda\Delta t_k+(\lambda\Delta t_k)^2/2}$	$\frac{\lambda\Delta t_k}{1+\lambda\Delta t_k+(\lambda\Delta t_k)^2/2}$
$\xi_{k\to}$	$-\frac{\lambda\Delta t_k/2}{1+\lambda\Delta t_k+(\lambda\Delta t_k)^2/2}$	$\frac{(\lambda\Delta t_k)^2/2}{1+\lambda\Delta t_k+(\lambda\Delta t_k)^2/2}$
$\xi_{C\to}$	$\frac{1}{\lambda\Delta t_k}\left[1-\frac{1}{1+\lambda\Delta t_k+(\lambda\Delta t_k)^2/2}\right]$	$\frac{1}{1+\lambda\Delta t_k+(\lambda\Delta t_k)^2/2}$

Table 3.4: Delayed source contribution factors of MBTD w/ linear $\boldsymbol{F_d\phi}$

	$\xi_{\to k-1/2}$	$\xi_{\to k}$
$\xi_{k-1\to}$	$\frac{1}{2}+\frac{1-e^{-\lambda\Delta t_k/2}}{\lambda\Delta t_k}-e^{-\lambda\Delta t_k/2}$	$\frac{1-e^{-\lambda\Delta t_k}}{\lambda\Delta t_k}-e^{-\lambda\Delta t_k}$
$\xi_{k-1/2\to}$	0	0
$\xi_{k\to}$	$\frac{1}{2}-\frac{1-e^{-\lambda\Delta t_k/2}}{\lambda\Delta t_k}$	$1-\frac{1-e^{-\lambda\Delta t_k}}{\lambda\Delta t_k}$
$\xi_{C\to}$	$e^{-\lambda\Delta t_k/2}$	$e^{-\lambda\Delta t_k}$

Table 3.5: Delayed source contribution factors of MBTD w/ quadratic $\boldsymbol{F_d\phi}$

	$\xi_{\to k-1/2}$	$\xi_{\to k}$
$\xi_{k-1\to}$	$-e^{-\lambda\Delta t_k/2}+\frac{1-3e^{-\lambda\Delta t_k/2}}{\lambda\Delta t_k}+\frac{1-e^{-\lambda\Delta t_k/2}}{(\lambda\Delta t_k)^2/4}$	$-e^{-\lambda\Delta t_k}-\frac{1+3e^{-\lambda\Delta t_k}}{\lambda\Delta t_k}+\frac{1-e^{-\lambda\Delta t_k}}{(\lambda\Delta t_k)^2/4}$
$\xi_{k-1/2\to}$	$1+\frac{e^{-\lambda\Delta t_k/2}}{\lambda\Delta t_k/4}-\frac{1-e^{-\lambda\Delta t_k/2}}{(\lambda\Delta t_k)^2/8}$	$\frac{1+e^{-\lambda\Delta t_k}}{\lambda\Delta t_k/4}-\frac{1-e^{-\lambda\Delta t_k}}{(\lambda\Delta t_k)^2/8}$
$\xi_{k\to}$	$-\frac{1+e^{-\lambda\Delta t_k/2}}{\lambda\Delta t_k}+\frac{1-e^{-\lambda\Delta t_k/2}}{(\lambda\Delta t_k)^2/4}$	$1-\frac{3+e^{-\lambda\Delta t_k}}{\lambda\Delta t_k}+\frac{1-e^{-\lambda\Delta t_k}}{(\lambda\Delta t_k)^2/4}$
$\xi_{C\to}$	$e^{-\lambda\Delta t_k/2}$	$e^{-\lambda\Delta t_k}$

Table 3.6: Delayed source contribution factors of MBTD w/ MB+ formulation

	$\xi_{\to k-1/2}$	$\xi_{\to k}$
$\xi_{k-1\to}$	0	0
$\xi_{k-1/2\to}$	$\lambda\Delta t_k$	$\lambda\Delta t_k e^{-\lambda\Delta t_k/2}$
$\xi_{k\to}$	$-\frac{\lambda\Delta t_k}{2}e^{\lambda\Delta t_k/2}$	0
$\xi_{C\to}$	$e^{-\lambda\Delta t_k/2}$	$e^{-\lambda\Delta t_k}$

The effective fission production operators $\boldsymbol{F_{eff}}$ are similarly defined in Eq. (3.75), and the constants $\xi_{k'\to k^*}$ and $\xi_{C\to k^*}$ (for each of the four methods) are provided in Table 3.3, Table 3.4, Table 3.5, and Table 3.6. This general form is structurally similar to that in Eq. (3.51), which means that we can apply the same MBTD coupled balance strategies for neutron diffusion without delayed neutrons discussed in Section 3.4. Finally, once the neutron flux solution $\boldsymbol{\phi_k}$ is obtained, the precursor solution $\boldsymbol{C_k}$ can be calculated by

$$\boldsymbol{C_k} = \xi_{C\to k}\boldsymbol{C_{k-1}} + \frac{1}{\lambda}\sum_{i=-2}^{0}\xi_{k+i/2\to k}\boldsymbol{F_d}\left(t_{k+i/2}\right)\boldsymbol{\phi}_{k+i/2}\,. \tag{3.85}$$

3.7 Analytical Test Problems

Table 3.7: The analytical kinetic test problems

Test Problem	Initial condition (steady-state)	Transient Part 1 ($0 < t \leq t_s$)	Transient Part 2 ($t_s < t \leq 2t_s$)
KP1 Subcritical (prompt only)	Subcritical without delayed neutrons with independent source: $\rho = -3.8\%$, $Q(t) = Q_0$ $\beta = 0$	Reactivity insertion by decreasing Σ_a, but still subcritical: $\rho = -1.15\%$, $t_s = 8$ ms	Decreasing independent source by a half: $Q(t) = Q_0/2$
KP2 Subcritical	Same as KP1, but with delayed neutron: $\rho = -5.92\$$, $\beta = 0.65\ \%$, $\lambda = 0.08\ \mathrm{s}^{-1}$	Same as KP1, $\rho = -1.78\$$, but $t_s = 50$ s	Same as KP1
KP3 Delayed supercritical	Critical without independent source: $\rho = 0$, $Q(t) = 0$ $\beta = 0.65\ \%$, $\lambda = 0.08\ \mathrm{s}^{-1}$	Delayed supercritical by decreasing Σ_a: $\rho = 0.36\$$, $t_s = 25$ s	Subcritical by increasing Σ_a: $\rho = -0.41\$$
KP4 Prompt supercritical	Same as KP3, but $\lambda = 0.4\ \mathrm{s}^{-1}$	Prompt supercritical by decreasing Σ_a: $\rho = 1.18\$$, $t_s = 18$ ms	Subcritical by increasing Σ_a: $\rho = -0.19\$$

The analytical test problems comprise four kinetic problems of mono-energetic *infinite* homogeneous media with one delayed neutron precursor group, which are summarized in Table 3.7. Each kinetic problem starts with an initial condition followed by two parts of transients (each with a duration of t_s): (1) Transient Part 1 and (2) Transient Part 2. In this infinite medium problem, the criticality of the system is defined as $k = \nu\Sigma_f/\Sigma_a$, and the corresponding *reactivity* is defined as $\rho = (k-1)/k$. This is often represented in the dollar unit of *delayed fission neutron fraction* $\beta = \nu_d/\nu = 1\$$. We note that a system is subcritical when $\rho < 0$, critical when $\rho = 0$, *delayed supercritical* when $0 < \rho < \beta$, and *prompt supercritical* when $\rho > \beta$. A typical value of $\beta = 0.65\%$ is used, and we use $\bar{\lambda}_{in} = 0.08\ \mathrm{s}^{-1}$ and $\bar{\lambda} = 0.4\ \mathrm{s}^{-1}$ [45] as the precursor decay constant λ for slow ($\rho < \beta$) and fast transients ($\rho > \beta$), respectively. Finally, by setting $D = 1/3\Sigma_t = 1.15\ \mathrm{cm}$ and $\nu\Sigma_f = 0.026\ \mathrm{cm}^{-1}$, and considering a typical *mean neutron generation time* of thermal reactors of $\Lambda = 1/(v\nu\Sigma_f) = 5\times 10^{-5}\ \mathrm{s}$, we set the neutron speed $v = 3.8\times 10^6\ \mathrm{cm/s}$.

Each transient part of the four kinetic problems described in Table 3.7 can be modeled as the following:

$$\begin{cases} \dfrac{1}{v}\dfrac{d\phi}{dt} + \Sigma_a\phi(t) = \nu_p\Sigma_f\phi(t) + \lambda C(t) + Q, \\ \\ \quad \dfrac{dC}{dt} + \lambda C(t) = \nu_d\Sigma_f\phi(t), \qquad t > 0, \end{cases} \tag{3.86}$$

with initial conditions $\phi(0) = \phi_{init}$, $C(0) = C_{init}$. This can be analytically solved by the method of normal mode decomposition [46]. We start by recasting the problem in matrix form:

$$\frac{d\boldsymbol{\Phi}}{dt} + \boldsymbol{A}\boldsymbol{\Phi}(t) = \begin{bmatrix} vQ \\ 0 \end{bmatrix}, \tag{3.87}$$

$$\boldsymbol{\Phi}(t) = \begin{bmatrix} \phi(t) \\ C(t) \end{bmatrix}, \qquad \boldsymbol{A} = \begin{bmatrix} v(\Sigma_a - \nu_p\Sigma_f) & -v\lambda \\ -\nu_d\Sigma_f & \lambda \end{bmatrix}. \tag{3.88}$$

Next, we define the normal vector $\boldsymbol{u}(t)$ that satisfies

$$\boldsymbol{V}\boldsymbol{u}(t) = \boldsymbol{\Phi}(t), \tag{3.89}$$

where $\boldsymbol{V}$ is the eigenvector matrix of $\boldsymbol{A}$, such that

$$\boldsymbol{A}\boldsymbol{V} = \boldsymbol{V}\begin{bmatrix} \omega_1 & 0 \\ 0 & \omega_2 \end{bmatrix}, \tag{3.90}$$

and ω_1 and ω_2 are the corresponding eigenvalues of $\boldsymbol{V}$. By introducing Eq. (3.89) into (3.87), and then using Eq. (3.90), we can reduce the original problem Eq. (3.87) into

$$\frac{d\boldsymbol{u}}{dt} + \begin{bmatrix} \omega_1 & 0 \\ 0 & \omega_2 \end{bmatrix} \boldsymbol{u}(t) = \boldsymbol{S}, \tag{3.91}$$

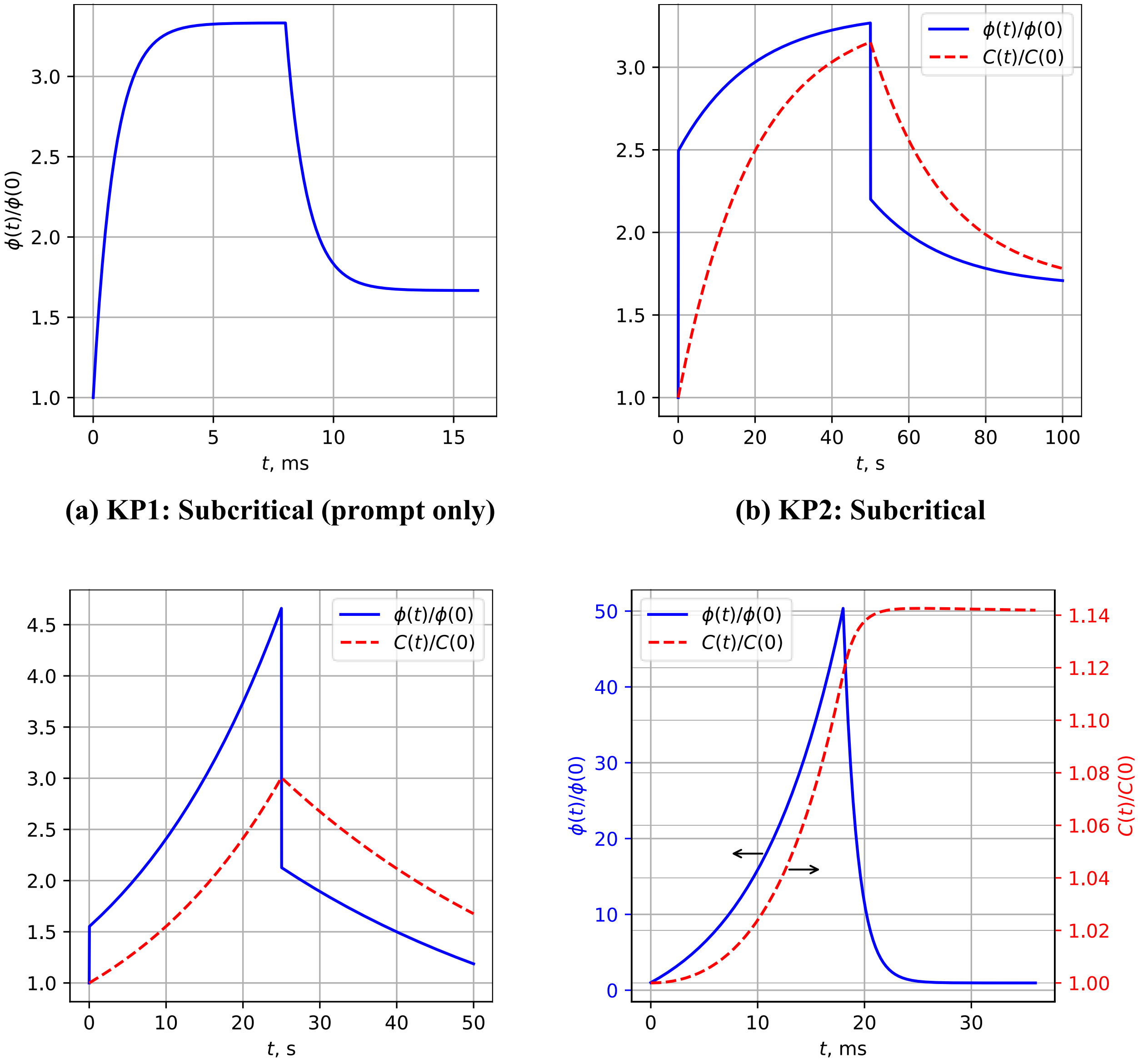
$$\boldsymbol{S} = \boldsymbol{V}^{-1} \begin{bmatrix} vQ \\ 0 \end{bmatrix}, \qquad \boldsymbol{u}(0) = \boldsymbol{V}^{-1}\boldsymbol{\Phi}(0). \tag{3.92}$$

This has the solutions

$$u_i(t) = \begin{cases} u_i(0) + S_i t, & \omega_i = 0, \\ u_i(0)e^{\omega_i t} + \dfrac{S_i}{\omega_i}(e^{\omega_i t} - 1), & \omega_i \neq 0, \end{cases} \tag{3.93}$$

with $i = 1, 2$. Finally, once we obtain $\boldsymbol{u}(t)$, we can determine the analytical solutions $\phi(t)$ and $C(t)$ per Eq. (3.89).

The resulting analytical solutions for the four kinetic test problems are shown in Figure 3.2. With the absence of delayed neutrons, the solution of KP1 smoothly and rapidly (in milliseconds) grows and decays to new steady-state solutions, which are governed by the strength of the independent source. In KP2 and KP3, the two significantly different time responses of prompt and delayed neutrons are responsible for the *prompt jump* and *prompt drop* at the beginning of the first and the second part of the transients; these emphasize the stiffness of KP2 and KP3. Finally, KP4 simulates a system under prompt supercriticality, which is an inherent characteristic of typical reactivity accident simulations. These four kinetic test problems are used as benchmarks to investigate, verify accuracy and stability, and assess the efficiency of the discussed methods (particularly those of MBTD and BE). Later in Chapter 4, the same kinetic test problems are used to investigate the application of the methods in neutron transport problems.



(a) KP1: Subcritical (prompt only)

(b) KP2: Subcritical

(c) KP3: Delayed supercritical

(d) KP4: Prompt supercritical

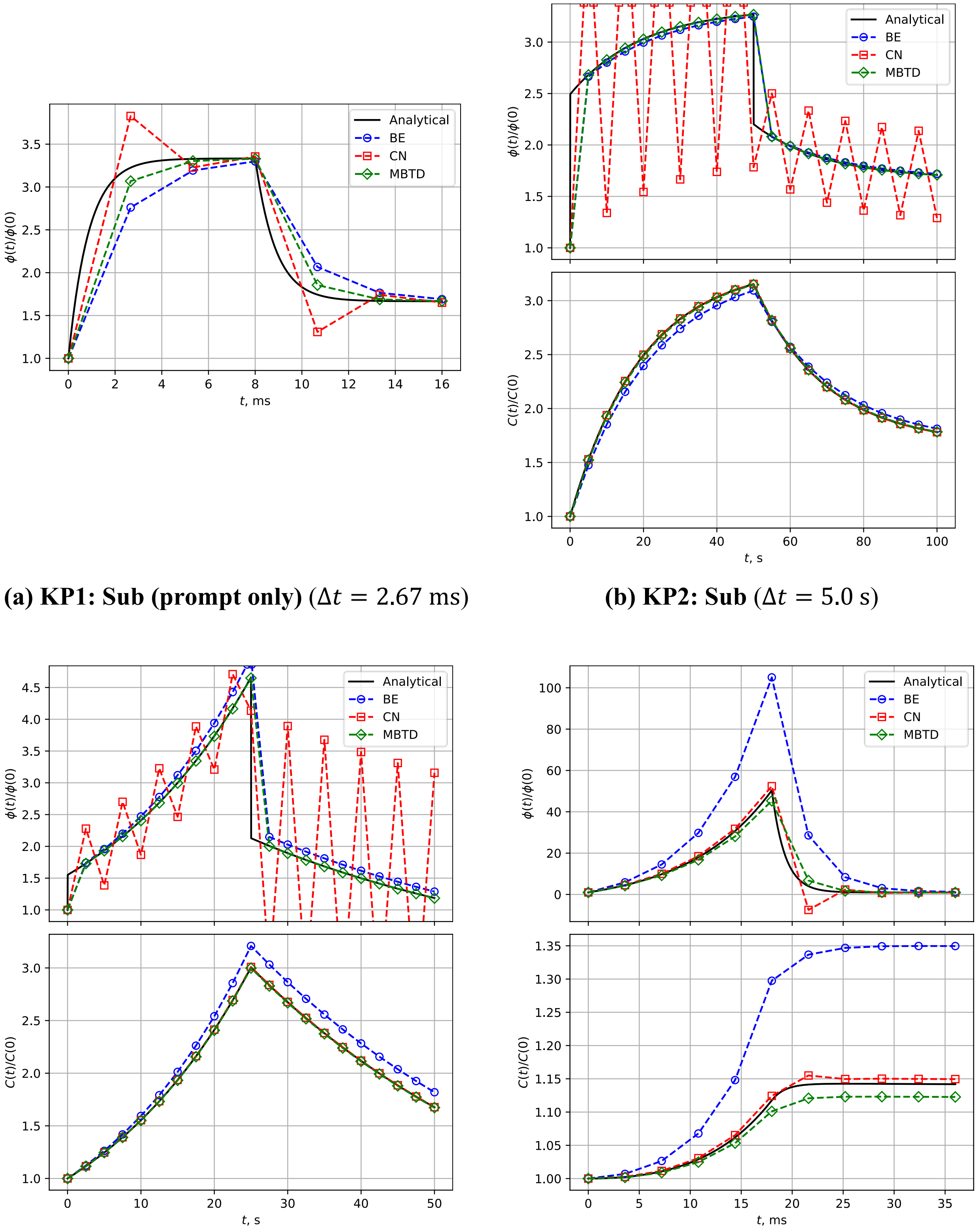
Figure 3.2: Analytical solutions of the kinetic problems

3.8 Numerical Results

First, in Section 3.8.1, the four kinetic test problems described in the previous section are numerically solved *as is* (infinite medium, no spatial dependency) with BE, CN, and MBTD by following their respective standard formulations in Section 2.1 and 2.2. The numerical solutions are then compared to the analytical solutions to determine and verify the accuracy and stability of the methods. Second, the delayed source approximations of BE and MBTD (derived and discussed in Section 3.6) are applied, and their numerical results are compared. Finally, in Section 3.8.3, the test problems are modeled as *very large* 1D-slab diffusion problems, such that the solutions at the center of the slab approximate the solutions of the infinite medium problems. These very large slab problems are then solved with BE and MBTD formulations for neutron diffusion problems derived in this chapter. This last numerical experiment is performed to test the formulation and assess the relative efficiency of the methods in solving diffusion problems: i.e., how much more expensive MBTD is compared to BE, and how much work (or time) is needed by each method to achieve a certain accuracy.

3.8.1 Accuracy and Stability

Figure 3.3 shows the resulting numerical solutions with relatively large uniform Δt. It is shown that (1) CN generally suffers from spurious oscillation, especially in the stiff problems KP2 and KP3; (2) the robust BE and MBTD well capture the overall features of the analytical solutions without spurious oscillation; (3) MBTD produces significantly more accurate solutions compared to BE due to its higher order of accuracy.



(a) KP1: Sub (prompt only) (Δt = 2.67 ms)

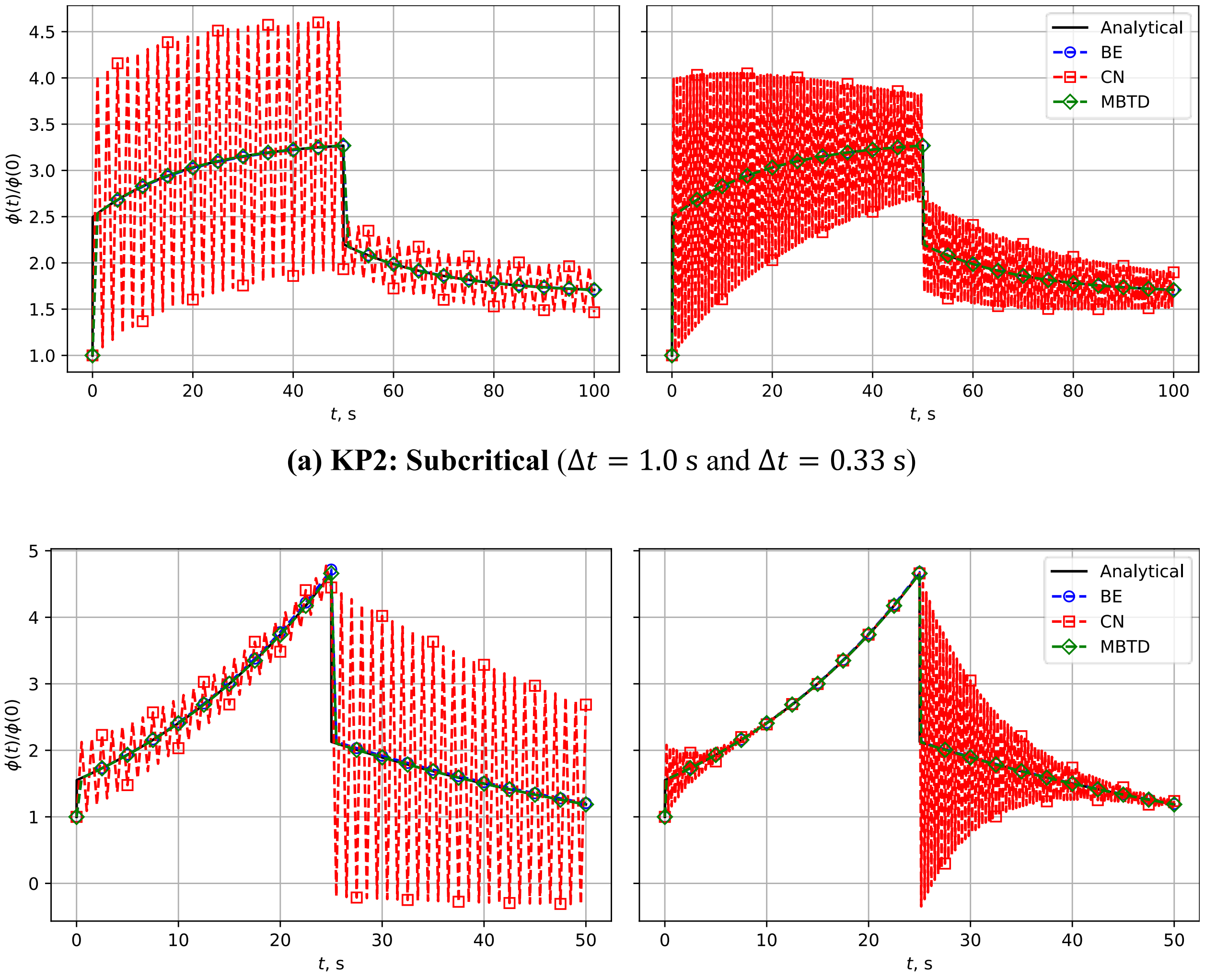
(b) KP2: Sub (Δt = 5.0 s)

(c) KP3: Delayed super (Δt = 2.5 s)

(d) KP4: Prompt super (Δt = 3.6 ms)

Figure 3.3: Numerical solutions (with coarse Δt) of the kinetic problems

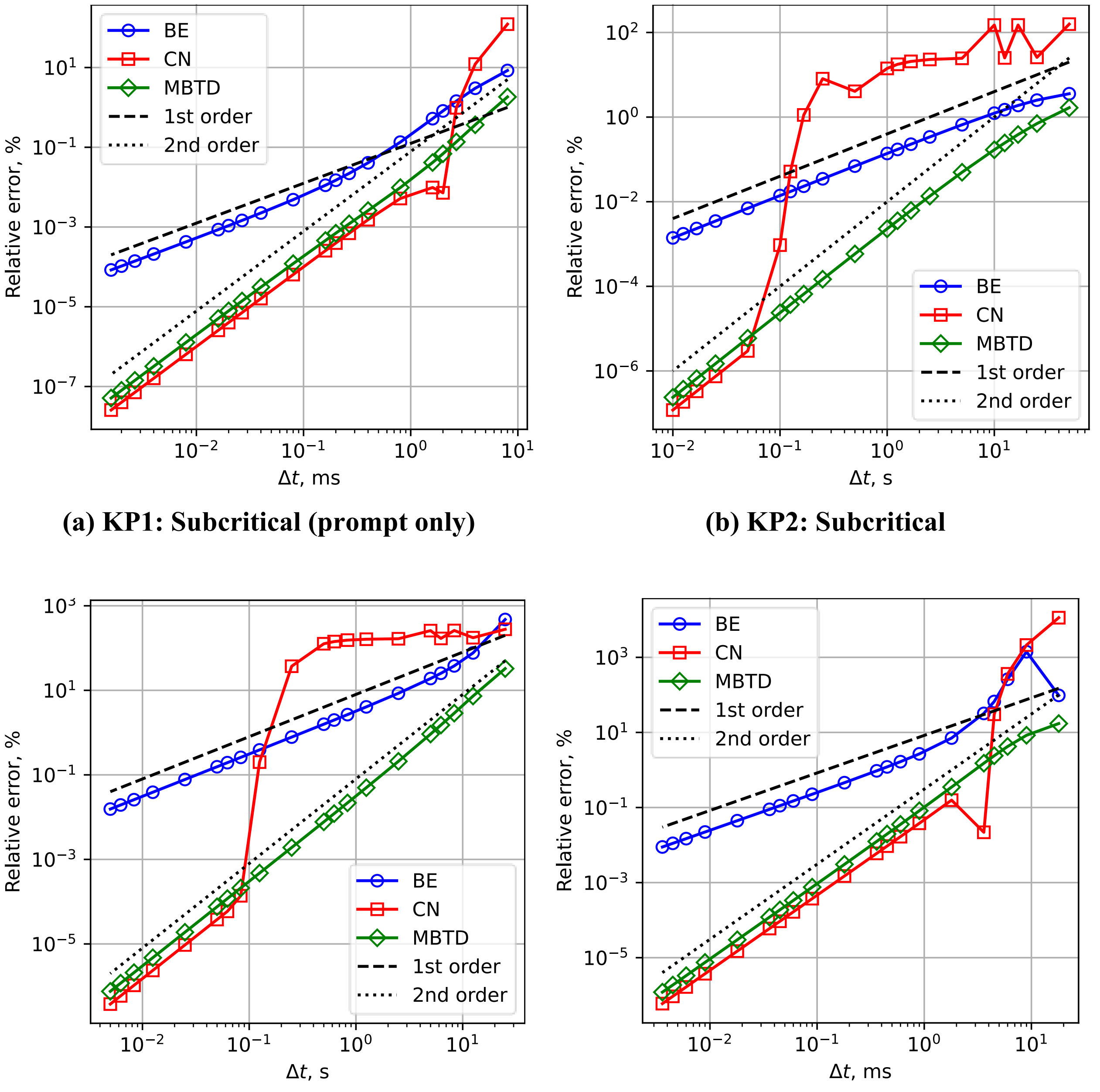
As we refine Δt, CN starts to damp its spurious oscillation. However, for solving the stiff problems KP2 and KP3, as shown in Figure 3.4, an extremely small Δt that well-resolves the prompt jumps and prompt drops is needed for CN to effectively damp the spurious oscillations and obtain physically meaningful results.



(a) KP2: Subcritical ($\Delta t = 1.0$ s and $\Delta t = 0.33$ s)

(b) KP3: Delayed supercritical ($\Delta \mathrm{t} = 0.5$ s and $\Delta \mathrm{t} = 0.167$ s)

Figure 3.4: Numerical solutions (with refining Δt) of the kinetic problems



(a) KP1: Subcritical (prompt only)

(b) KP2: Subcritical

(c) KP3: Delayed supercritical

(d) KP4: Prompt supercritical

Figure 3.5: Accuracy of the time-stepping methods for the kinetic problems

Relative errors of the method numerical solutions at the final time $t = 2t_s$ as a function of Δt are compared in Figure 3.5. This error metric is chosen to verify the methods' accuracy orders and stabilities in this section and in Section 3.8.2. The same error metric is used to assess the relative efficiency of MBTD to BE in solving the four kinetic test problems in Section 3.8.3 (for neutron diffusion) and Section 4.6.2 (for neutron transport). Later in Section 5.2, variants of error metrics will be used to assess the methods' relative efficiency in solving a multi-physics problem.

From Figure 3.5, it is evident that BE is first-order accurate (refer to broken lines $\propto \Delta t$), while CN and MBTD are second-order accurate (refer to the dotted lines $\propto \Delta t^2$). In addition, it is shown that CN is twice as accurate as MBTD, which is expected per Table 2.1. The conditional stability of CN is observable as well, especially for KP2 and KP3.

Another worth noting finding is that BE may suffer from spurious oscillations in the exponentially growing prompt supercritical system KP4, which leads to producing a non-physical negative solution, as shown in Figure 3.6. This is aligned with the observation of the BE amplification factor in Table 2.1: if Δt is not sufficiently small, the magnitude of the negative valued η (in exponentially growing system) will be larger than unity so that the amplification factor of BE becomes negative. On the other hand, MBTD preserves the positivity of its amplification factor so that it is still free from spurious oscillations even in such an exponentially growing problem.

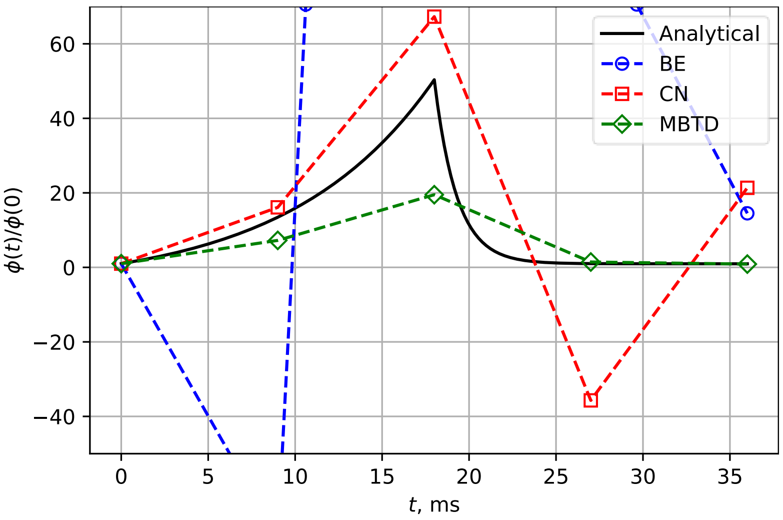
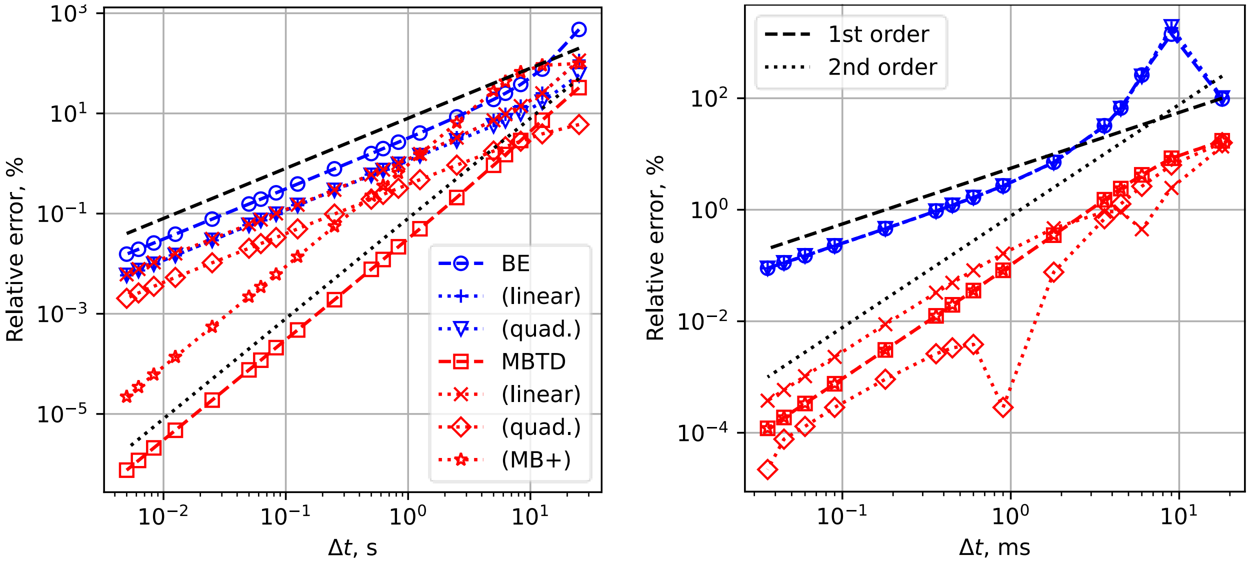


Figure 3.6: Possible spurious oscillation of BE in prompt supercritical problem KP4

3.8.2 *Delayed Source Approximations*

Figure 3.7 compares the resulting relative error of different delayed source approximations of BE and MBTD for kinetic problems KP3 and KP4.



(a) KP3: Delayed supercritical **(b) KP4: Prompt supercritical**

Figure 3.7: Accuracy of the delayed source approximations for the kinetic problems

For KP3 [Part (a) in Figure 3.7], it is found that all delayed source approximations of BE (blue lines) produce first-order accurate methods. The linear ("+" marker) and quadratic (down-triangle marker) delayed fission production approximations of BE produce considerably more accurate solutions than the standard BE delayed source formulation (circle marker). As for MBTD (red lines), it is evident that the linear (cross marker) and quadratic (diamond marker) delayed fission production approximations, unfortunately, produce first-order accurate methods. This indicates that the MBTD linear and quadratic delayed fission production approximations introduce a first-order inaccuracy, hindering the full potential of MBTD. Nevertheless, the standard MBTD delayed source formulation (square marker) preserves the second-order accuracy of MBTD and outperforms the other three MBTD delayed source approximations. It is worth mentioning that the fourth method MB+ (star marker) manages to preserve the second-order accuracy, but it introduces a large constant in front of the leading error term, making it considerably less accurate and inferior to the standard MBTD delayed source formulation.

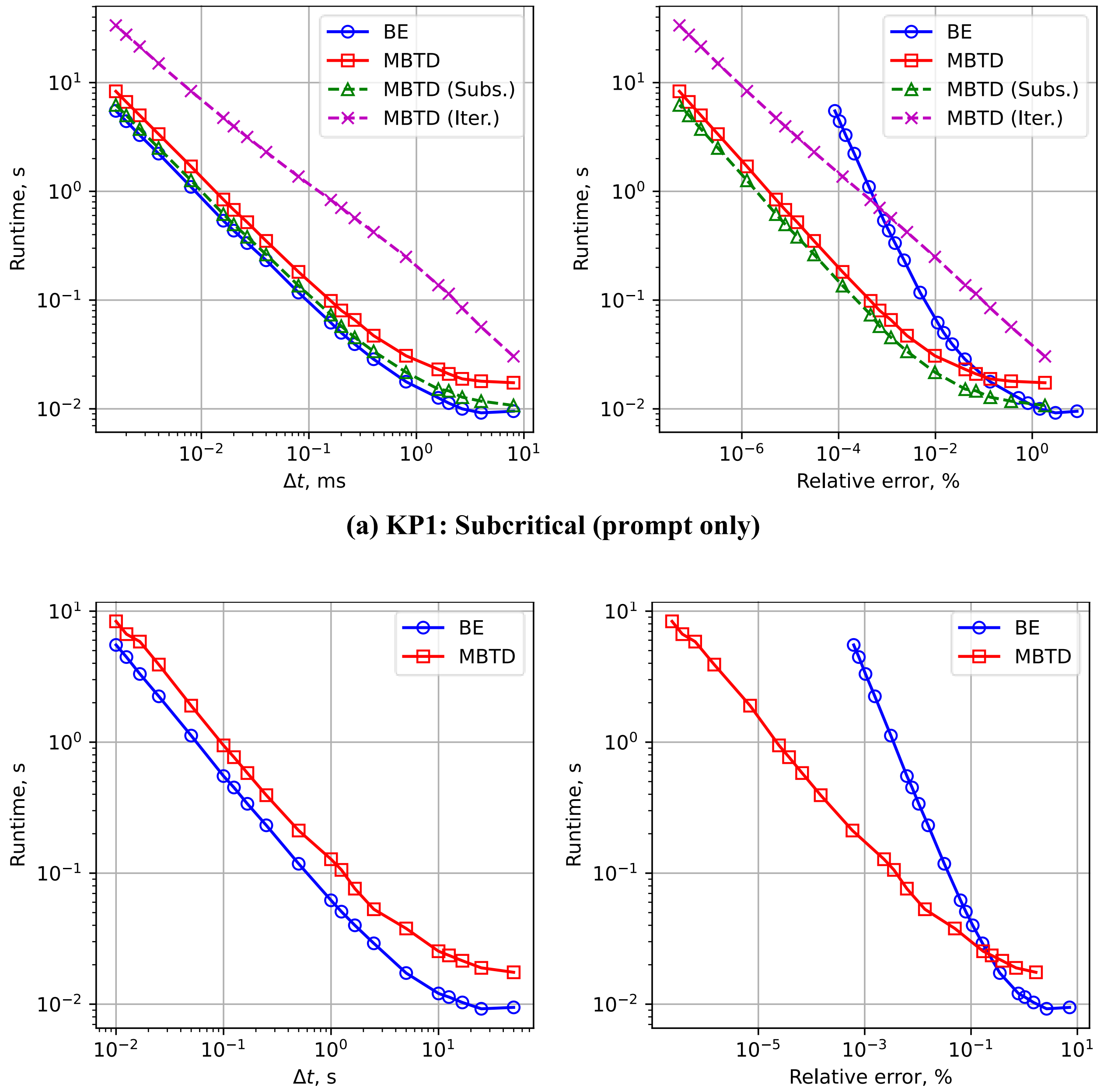
Different results are observed for KP4 [Part (b) in Figure 3.7]. This is mainly because prompt neutrons dominantly drive the transient of the prompt supercritical system. It is found that all delayed source approximations of BE have the same accuracy. As for MBTD, all of the delayed source approximations produce a second-order accurate method, where the linear and quadratic delayed fission production approximations are respectively less and more accurate than the standard MBTD formulation and MB+.

The main goal of this thesis work is to seek a robust second-order method. We concluded that the standard MBTD delayed source formulation is the most appropriate delayed source approximation for MBTD. As for BE, because of the lack of significant improvement from the quadratic approximation (at least for the test problems), the linear delayed fission production approximation is used for the rest of this thesis work in gauging the relative efficiency of MBTD.

3.8.3 Efficiency

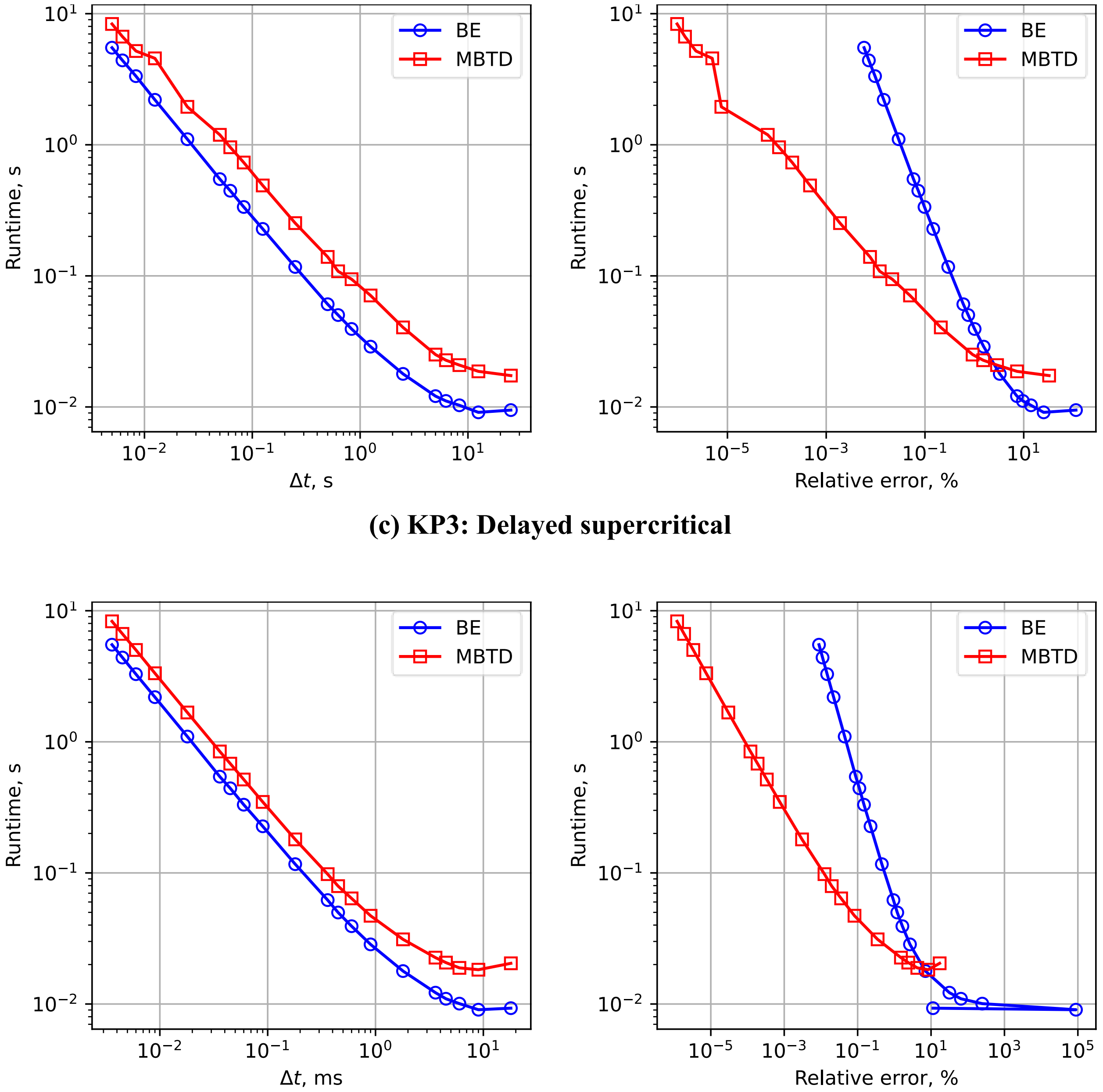
The infinite medium kinetic test problems in Table 3.7 are modeled as large 1D-slab diffusion problems such that the solutions at the center of the slab approximate the solutions of the original infinite medium problems. The diffusion problems are then numerically solved by using the neutron diffusion formulations of BE [Eq. (3.73), with linear delayed fission production] and MBTD [Eq. (3.82), with standard MBTD delayed source formulation Table 3.3]. Furthermore, we consider the three strategies of MBTD: (1) Substitution [described in Eq. (3.53)], (2) Lagged $O(\Delta t)$ Iterative Solve [described in Eq. (3.56), with relative error 2-norm convergence criterion of 10^{-5}], and (3) Simultaneous Solve. The simple diffusion coefficient $D = 1/3\Sigma_t$ is used, the full-width ($2X$) for the large slab model is chosen to be 1800 mean free paths $1/\Sigma_t$, and the spatial-mesh size Δx is taken to be one half of the mean free path. To take advantage of the symmetry and exercise the diffusion solvers with varying boundary conditions, left-vacuum ($\alpha_L = 0$) and right-reflecting ($\alpha_R = 1$) boundary conditions are used. Finally, SciPy's sparse ILU-preconditioned GMRES linear solver [47] is used to solve the resulting linear problems with a relative error tolerance of 10^{-7}. The simulations are performed with uniform time-step size $\Delta t = 2t_s/K$, with *number of time steps* K ranging from 2 to 10^4. The resulting relative errors and the simulation runtimes are shown in Figure 3.8.

In Figure 3.8, two plots are presented for each of the kinetic problems. The first plot (on the left) shows the calculation runtime for each Δt. It essentially informs the computational work required for the methods to generate solutions with a given time-step size. The smaller Δt, the more linear problems need to be solved during the whole simulation $t \in [0, 2t_s]$, thus the larger the computational cost and the longer the runtime. Nevertheless, the smaller Δt, the more accurate the solution, and how much more accurate the solution depends on the method's order of accuracy, as shown in Figure 3.5.



(a) KP1: Subcritical (prompt only)

(b) KP2: Subcritical



(c) KP3: Delayed supercritical

(d) KP4: Prompt supercritical

Figure 3.8: Runtime and efficiency of BE and MBTD for diffusion problems

The second plots (on the right) are similar to the first plots (on the left) except that they replace the x-axis with the resulting relative error of simulation with the given Δt. This second plot effectively compares the efficiency of the methods. If we draw a horizontal line at a fixed runtime, we can compare the achieved accuracy of the methods given the same computational effort (runtime). If we draw a vertical line at a fixed relative error, we can compare the computational cost of the methods to achieve that specified accuracy.

First, let us observe the results of KP2, KP3, and KP4 [Part (b), (c), and (d) in Figure 3.8]. From their first plots (on the left), it is found that the runtime of MBTD with Simultaneous Solve strategy (solid red line with square marker) is at most about two times larger than that of BE given the same value of Δt. Furthermore, from the second plots (on the right), it is found that MBTD's curve is always at the bottom-left side of the BE's, except for larger relative errors, which correspond to the small number of time steps K or low temporal resolution. This means, for reasonably accurate simulations, MBTD with the Simultaneous Solve strategy is more efficient than BE: (1) with the same computational effort, higher accuracy can be achieved, or (2) to achieve a certain accuracy, less computational effort is required.

Now let us turn to observe the results of KP1 [Part (a) of Figure 3.8], where we also compare the efficiency of the three strategies of MBTD. From the first plot (on the left), it is found that MBTD with the Substitution strategy [broken green line with star marker, labeled as MBTD (Subs.)] is slightly less expensive than the Simultaneous Solve strategy and almost as expensive as BE. This finding suggests that Substitution is a more efficient strategy, by a small margin than the Simultaneous Solve. Nevertheless, the applicability of the Substitution strategy for practical problems is limited. In this simple mono-energetic 1D-slab problem, we can straightforwardly construct the squared diffusion operator, which manifests as a five-diagonal matrix. However, in practical problems with multiple dimensions in space and multiple neutron energy groups, the squared diffusion operator is much more complicated; one can imagine that by squaring the operator, each of the terms is operated to itself and to all the other terms. Even if we can effectively construct such a complicated squared operator matrix, the efficiency will not be much improved from that of Simultaneous Solve, which—on the other hand—is much easier to adapt for more practical problems.

Finally, it is also observed from Part (a) of Figure 3.8 that the Lagged $O(\Delta t)$ Iterative Solve strategy [broken magenta line with cross marker, labeled as MBTD (Iter.)] is about six to seven times more expensive than BE. This indirectly indicates that only three iterations are needed to achieve convergence, since the Lagged $O(\Delta t)$ strategy solves two BE-like problems at each of the iterations. Despite the small number of iterations required for convergence and its ease of implementation (directly using BE routine as a black-box), the entailing six to seven times more work is enough to conclude that the Lagged $O(\Delta t)$ Iterative Solve strategy is too inefficient compared to the other two strategies.

3.9 Summary

In this chapter, we investigate MBTD application to time-dependent neutron diffusion problems. The MBTD formulation for neutron diffusion problems without delayed neutron precursors and the strategies for solving the resulting coupled balance equations are discussed in Section 3.4. A mono-energetic 1-D slab problem is considered in the formulation; however, its representation in terms of operators serves as a guideline for generalization to more practical problems.

A Von Neumann analysis is performed to characterize the stability and robustness of θ-Method (FE, CN, and BE) and MBTD with FDM spatial discretization in terms of non-dimensional parameters: the scattering ratio c, the criticality k, the ratio of diffusion coefficient to optical thickness ς, and—more importantly—the optical distance travelled per time step η and the optical thickness of spatial mesh τ. These are summarized in Table 3.1. While it is again theoretically proven that BE and MBTD are unconditionally robust, it is found that there is a severe restriction for FE to achieve stability—η must decrease at about the rate of τ^2—and the same severe restriction applies to CN to guarantee robustness (free of spurious oscillation).

Standard and traditional delayed source approximations of BE are presented, and their adaptations for MBTD are discussed in Section 3.6. These delayed sources allow us to generalize the method formulations to include the effect of delayed neutron precursors by means of effective contribution factors. These effective contribution factors are summarized in (1) Table 3.2 for BE Eq. (3.73), and (2) Table 3.3 to Table 3.6 for MBTD Eq. (3.82).

Four infinite-medium kinetic test problems are devised in Section 0 and summarized in Table 3.7. The problem set comprises source-driven subcritical systems, without and with delayed neutrons (KP1 and KP2, respectively), a delayed supercritical system (KP3), and a prompt supercritical system (KP4). Their analytic solutions are given in Figure 3.2. These four analytical test problems are used as benchmarks to investigate, verify accuracy and stability, and assess the efficiency of the discussed methods, along with their respective delayed source approximations and solving strategies (for MBTD).

Numerical experiments are discussed in Section 0, and the results are presented in Figure 3.3 to Figure 3.8. Figure 3.3 to Figure 3.5 (1) verify the accuracy order and robustness of BE and MBTD; (2) emphasize the persisting spurious oscillation of CN, especially in solving the challenging stiff problems KP2 and KP3, and (3) demonstrate the possible spurious oscillation of

BE in the exponentially growing prompt supercritical problem KP4. Figure 3.7 compares the accuracy and stability of the delayed source approximations of MBTD. It is found that the standard MBTD delayed source formulation is the most optimum approximation, as it effectively preserves second-order accuracy. Finally, it is found that Simultaneous Solve is the most efficient strategy for solving the MBTD neutron diffusion coupled balance equations, as it is easier to implement (relative to Substitution) and only about two times computationally more expensive than BE, given the same value of Δt as shown in Figure 3.8. This ultimately makes MBTD a more efficient method, compared to BE, for reasonably accurate simulations.

CHAPTER 4

MBTD Application to Neutron Transport

In this chapter, the application of MBTD to the neutron transport equation is investigated; in particular, its numerical formulation and efficiency are compared to those of BE. In Section 4.1, we present the time-dependent neutron transport equation and its other forms, including the steady-state and the eigenvalue problems. A representative mono-energetic 1-D slab problem with isotropic scattering is considered. Despite its simplicities, it serves as a guideline, and its numerical analysis and results reveal strong incentives for generalization to more practical problems. In Section 4.2, standard steady-state neutron transport methods are discussed; these include Source Iteration (SI) and the discretization methods discrete ordinate (SN) and Method of Characteristics (MOC). Section 4.3 presents how the steady-state neutron transport methods discussed in the preceding section are integrated and adapted into the time-stepping methods BE and MBTD for solving time-dependent neutron transports, where the Simultaneous Solve strategy is used to solve the resulting MBTD coupled balance equations. In Section 4.4, a Fourier analysis is performed to characterize and compare the convergence of BE-SI and MBTD-SI. In Section 4.5, Diffusion Synthetic Acceleration (DSA) for BE and MBTD are formulated, and the resulting iteration schemes are again Fourier-analyzed. Numerical experiments are performed in Section 4.6, where the four analytical test problems devised in Section 0 are used as benchmarks to verify the characteristics and assess the efficiency of the discussed neutron transport methods. Finally, Section 4.7 summarizes the contents of the chapter.

4.1 Neutron Transport Equations

Let us consider a mono-energetic 1D-slab time-dependent neutron transport equation with isotropic scattering and one delayed neutron precursor:

$$\begin{cases} \dfrac{1}{v}\dfrac{\partial\psi}{\partial t} + \mu\dfrac{\partial\psi}{\partial x} + \Sigma_t(x,t)\psi(x,\mu,t) = \\ \qquad\qquad \dfrac{1}{2}\left[\Sigma_s(x,t) + \nu_p\Sigma_f(x,t)\right]\phi(x,t) + \dfrac{1}{2}\lambda C(x,t) + Q(x,\mu,t), \\ \dfrac{\partial C}{\partial t} + \lambda C(x,t) = \nu_d\Sigma_f(x,t)\phi(x,t), \qquad x \in [0,X],\ \ \mu \in [-1,1],\ \ t > 0, \end{cases} \tag{4.1}$$

with boundary conditions

$$\begin{cases} \psi(0,\mu,t) = \psi_L(\mu,t), \qquad \mu \in [0,1], \\ \psi(X,\mu,t) = \psi_R(\mu,t), \qquad \mu \in [-1,0], \qquad t > 0, \end{cases} \tag{4.2}$$

and initial conditions

$$\psi(x,\mu,0) = \psi_{init}(x,\mu), \qquad C(x,0) = C_{init}(x), \qquad x \in [0,X],\ \ \mu \in [-1,1], \tag{4.3}$$

where neutron flux ϕ is the “directional sum” of the angular flux ψ:

$$\phi(x,t) = \int_{-1}^{1}\psi(x,\mu,t)d\mu\,. \tag{4.4}$$

Different from the neutron diffusion equation in Eq. (3.2) that approximates neutron streaming as a diffusion process, the neutron transport equation exactly models the neutron streaming process with respect to the neutron direction of flight. In this 1D-slab problem, this is represented by the polar angle cosine μ. In operator form, Eq. (4.1) reduces to

$$\begin{cases} \dfrac{1}{v}\dfrac{\partial\psi}{\partial t} + \boldsymbol{T}(x,\mu,t)\psi(x,\mu,t) = \left[\boldsymbol{S}(x,t) + \dfrac{\boldsymbol{F_p}(x,t)}{2}\right]\phi(x,t) + \dfrac{1}{2}\lambda C(x,t) + Q(x,\mu,t), \\ \dfrac{\partial C}{\partial t} + \lambda C(x,t) = \boldsymbol{F_d}(x,t)\phi(x,t), \end{cases} \tag{4.5}$$

where the *transport operator* T and the *isotropic scattering operator* are respectively defined as

$$\boldsymbol{T}(x,\mu,t) = \mu\frac{\partial}{\partial x} + \Sigma_t(x,t), \qquad \boldsymbol{S}(x,t) = \frac{1}{2}\Sigma_s(x,t). \tag{4.6}$$

The fission production operators $\boldsymbol{F_p}$ and $\boldsymbol{F_d}$ are identical to those defined in neutron diffusion in Section 3.1.

Similar to what is done in Section 3.1 for neutron diffusion, we consider the steady-state and eigenvalue equations for the neutron transport. The steady-state equation is:

$$\boldsymbol{T}(x,\mu)\psi(x,\mu) = \left[\boldsymbol{S}(x,t) + \frac{\boldsymbol{F}(x,t)}{2}\right]\phi(x,t) + Q(x,\mu), \qquad (4.7)$$

$$C(x) = \frac{\boldsymbol{F_d}(x,t)\phi(x)}{\lambda}. \qquad (4.8)$$

The k-eigenvalue equation is:

$$\boldsymbol{T}(x,\mu)\psi(x,\mu) = \left[\boldsymbol{S}(x,t) + \frac{1}{k}\frac{\boldsymbol{F}(x,t)}{2}\right]\phi(x), \qquad (4.9)$$

$$C(x) = \frac{1}{k}\frac{\boldsymbol{F_d}(x,t)\phi(x)}{\lambda}. \qquad (4.10)$$

The α-eigenvalue equation is:

$$\left[\boldsymbol{T}(x,\mu) + \frac{\alpha}{v}\right]\psi(x,\mu) = \left\{\boldsymbol{S}(x,t) + \frac{1}{2}\left[\boldsymbol{F_p}(x,t) + \frac{\lambda}{\alpha+\lambda}\boldsymbol{F_d}(x,t)\right]\right\}\phi(x), \qquad (4.11)$$

$$C(x) = \frac{\boldsymbol{F_d}(x)\phi(x)}{\alpha+\lambda}. \qquad (4.12)$$

Finally, the prompt α-eigenvalue diffusion equation is:

$$\left[\boldsymbol{T}(x,\mu) + \frac{\alpha}{v}\right]\psi(x,\mu) = \left[\boldsymbol{S}(x,t) + \frac{\boldsymbol{F_p}(x,t)}{2}\right]\phi(x), \qquad (4.13)$$

4.2 Source Iteration, SN, and MOC

Let us consider the steady-state neutron transport Eq. (4.7) with

$$\phi(x) = \int_{-1}^{1}\psi(x,\mu)d\mu\,, \qquad (4.14)$$

and boundary conditions

$$\begin{cases} \psi(0,\mu) = \psi_L(\mu), & \mu > 0, \\ \psi(X,\mu) = \psi_R(\mu), & \mu < 0. \end{cases} \tag{4.15}$$

It has been standard practice to iteratively solve this neutron transport problem by the following iteration scheme:

$$\boldsymbol{T}(x,\mu)\psi^{(l)}(x,\mu) = \left[\boldsymbol{S}(x,t) + \frac{\boldsymbol{F}(x,t)}{2}\right]\phi^{(l-1)}(x) + Q(x,\mu), \tag{4.16}$$

or

$$\mu\frac{\partial\psi^{(l)}}{\partial x} + \Sigma_t(x)\psi^{(l)}(x,\mu) = \frac{1}{2}\left[\Sigma_s(x) + \nu\Sigma_f(x)\right]\phi^{(l-1)}(x) + Q(x,\mu), \tag{4.17}$$

$$\phi^{(l)}(x) = \int_{-1}^{1}\psi^{(l)}(x,\mu)d\mu\,, \tag{4.18}$$

where the superscript $^{(l)}$ indicates the iteration index. The iteration starts by making an initial guess of $\phi^{(0)}$ to solve for $\psi^{(1)}$ per Eq. (4.17), which is then used to obtain the new iterate $\phi^{(1)}$ per Eq. (4.18). This iterative method, which is known as *Source Iteration* (SI), is advantageous because it decouples the angular neutron flux ψ in μ, such that we can independently solve $\psi(x,\mu)$ for each value of μ from their respective boundary conditions. This embarrassingly parallel process makes neutron transport methods effectively parallelizable.

Now, we would like to discretize the angular and spatial variables of Eqs. (4.17), (4.18), and (4.15). The *discrete ordinate method* (SN) approximates the equations as follows:

$$\mu_n\frac{d\psi_n^{(l)}}{dx} + \Sigma_t(x)\psi_n^{(l)}(x) = \frac{1}{2}\left[\Sigma_s(x) + \nu\Sigma_f(x)\right]\phi^{(l-1)}(x) + Q_n(x), \tag{4.19}$$

$$\phi^{(l)}(x) = \sum_{n=1}^{N}\psi_n^{(l)}(x)w_n\,, \tag{4.20}$$

$$\begin{cases} \psi_n(0) = \psi_{L,n}, & \mu_n > 0, \\ \psi_n(X) = \psi_{R,n}, & \mu_n < 0, \end{cases} \tag{4.21}$$

where $\psi_n(x) = \psi(x, \mu_n)$ and Gauss-Legendre quadrature set of order N are used for μ_n and w_n. The left boundary condition (and also the right) can be simply interpreted as $\psi_{L,n} = \psi_L(\mu_n)$ or can be more accurately defined as

$$\psi_{L,n} = \frac{\int_{\mu_n - w_n/2}^{\mu_n + w_n/2} \mu \psi_L(\mu) d\mu}{\mu_n w_n}, \tag{4.22}$$

which preserves the total neutrons entering the system within incident cosine angle in w_n about μ_n.

The next task is to discretize the spatial dependence of the system. Let us consider the same system with piece-wise constant cross-sections within non-uniform spatial meshes of size Δx_j described in Eq. (3.23). We note that the spatial grids are identically set up like the box-scheme FDM for neutron diffusion discussed in Section 3.2. Now, we define the *cell-edge* and *cell-average* quantities (denoted by subscripts $_{j\pm1/2}$ and $_j$, respectively):

$$\begin{cases} \psi_{n,j\pm1/2} = \psi_n(x_{j\pm1/2}), \qquad \psi_{n,j} = \dfrac{1}{\Delta x_j} \displaystyle\int_{x_{j-1/2}}^{x_{j+1/2}} \psi_n(x) dx, \\ \phi_j = \displaystyle\sum_{n=1}^{N} \psi_{n,j} w_n, \qquad Q_{n,j} = \dfrac{1}{\Delta x_j} \displaystyle\int_{x_{j-1/2}}^{x_{j+1/2}} Q_n(x) dx, \qquad j = 1, 2, \dots, J. \end{cases} \tag{4.23}$$

The spatial grid of ψ (and also ϕ) is identical to the one illustrated in Figure 3.1. We note that subscript j and n respectively indicate spatial and angular indexes. By operating Eqs. (4.19) and (4.20) with $\Delta x_j^{-1} \int_{x_{j-1/2}}^{x_{j+1/2}} (\cdot) dx$, we obtain

$$\frac{\mu_n}{\Delta x_j} \left(\psi_{n,j+1/2}^{(l)} - \psi_{n,j-1/2}^{(l)} \right) + \Sigma_{t,j} \psi_{n,j}^{(l)} = \frac{1}{2} (\Sigma_{s,j} + \nu \Sigma_{f,j}) \phi_j^{(l-1)} + Q_{n,j}, \tag{4.24}$$

$$\phi_j^{(l)} = \sum_{n=1}^{N} \psi_{n,j}^{(l)} w_n, \tag{4.25}$$

$$\begin{cases} \psi_{n,1/2} = \psi_{L,n}, & \mu_n > 0, \\ \psi_{n,J+1/2} = \psi_{R,n}, & \mu_n < 0. \end{cases} \tag{4.26}$$

Finally, to be able to solve Eq. (4.24), we need an approximation that relates the cell-edge $\psi_{n,j\pm 1/2}$ with the cell-average $\psi_{n,j}$. One of the simplest spatial discretization is the second-order-accurate *Diamond-Difference* method (DD), which approximates

$$\psi_{n,j}^{(l)} = \frac{1}{2}\left(\psi_{n,j+1/2}^{(l)} + \psi_{n,j-1/2}^{(l)}\right). \tag{4.27}$$

It is evident that DD is essentially CN for spatial discretization.

Eqs. (4.24) and (4.27) (hereafter referred to as “SI with SN-DD”) can be solved by sweeping the solutions, for each angular index n, from one end to the other. For *forward transport sweep* ($\mu_n > 0$), we solve for $\psi_{n,j+1/2}$ and $\psi_{n,j}$ from the previously solved $\psi_{n,j-1/2}$ or the boundary condition $\psi_{n,1/2}$ per Eqs. (4.24) and (4.27). For *backward transport sweep* ($\mu_n < 0$), we solve for $\psi_{n,j-1/2}$ and $\psi_{n,j}$ from the previously solved $\psi_{n,j+1/2}$ or the boundary condition $\psi_{n,J+1/2}$ per Eqs. (4.24) and (4.27). Once we obtain all of the cell-average $\psi_{n,j}$, we can update the cell-average neutron flux iterate solution ϕ_j per Eq. (4.25) and proceed to the next iteration.

Another widely used spatial discretization method is the *Method of Characteristics*. In this method, instead of using Eq. (4.27), we *analytically* solve Eq. (4.19) for each spatial step j by assuming a constant RHS source:

$$\mu_n \frac{d\psi_n^{(l)}}{dx} + \Sigma_{t,j}\psi_n^{(l)}(x) = \hat{Q}_{n,j}^{(l-1)}, \tag{4.28}$$

where

$$\hat{Q}_{n,j}^{(l-1)} = \frac{1}{2}\left(\Sigma_{s,j} + \nu\Sigma_{f,j}\right)\phi_j^{(l-1)} + Q_{n,j}. \tag{4.29}$$

For forward transport sweep, we analytically solve for $\psi_{n,j+1/2}$ and $\psi_{n,j}$ from the given $\psi_{n,j-1/2}$. This gives

$$\psi_{n,j+1/2}^{(l)} = \left(\psi_{n,j-1/2}^{(l)} - \frac{\hat{Q}_{n,j}^{(l-1)}}{\Sigma_{t,j}}\right) e^{-\tau_{n,j}} + \frac{\hat{Q}_{n,j}^{(l-1)}}{\Sigma_{t,j}}, \tag{4.30}$$

$$\psi_{n,j}^{(l)} = \frac{\hat{Q}_{n,j}^{(l-1)}}{\Sigma_{t,j}} - \frac{1}{\tau_{n,j}}\left(\psi_{n,j+1/2}^{(l)} - \psi_{n,j-1/2}^{(l)}\right) e^{-\tau_{n,j}}, \tag{4.31}$$

where the non-dimensional quantity $\tau_{n,j}$ is the number of optical thickness traveled—with respect to the flight direction μ_n—per spatial mesh Δx_j:

$$\tau_{n,j} = \frac{\Sigma_t \Delta x_j}{\mu_n}. \tag{4.32}$$

Eqs. (4.30) and (4.31) are hereafter referred to as “SI with MOC”.

4.3 Transport Methods for BE and MBTD

In the previous section, we describe standard transport methods SI with SN-DD [Eqs. (4.24) and (4.27)] and SI with MOC [Eqs. (4.30) and (4.31)] for solving the steady-state neutron transport Eq. (4.7). In this section, we will apply the time-stepping methods to the time-dependent neutron transport Eq. (4.5) such that we end up with a problem in the form of the steady-state Eq. (4.7). Therefore, we can utilize the standard steady-state transport methods described in the previous section to solve the discretized time-dependent neutron transport.

Applying the time-stepping method BE to the time-dependent neutron transport Eq. (4.5), we obtain the following:

$$\left[\boldsymbol{T_k}(x,\mu) + \frac{1}{v\Delta t_k}\right]\psi_k^{(l)}(x,\mu) = \left[\boldsymbol{S_k}(x) + \frac{\boldsymbol{F_{eff,k}}(x)}{2}\right]\phi_k^{(l-1)}(x) + Q_{eff,k}(x,\mu), \tag{4.33}$$

where the effective independent source is

$$\begin{aligned} Q_{eff,k}(x,\mu) = Q_k(x,\mu) + \frac{1}{v\Delta t_k}\psi_{k-1}(x,\mu) \\ + \frac{1}{2}\left[\sum_{k'=k-2}^{k-1} \xi_{k'\to k}\boldsymbol{F_{d,k'}}(x)\phi_{k'}(x) + \xi_{C\to k}\lambda C_{k-1}(x)\right]. \end{aligned} \tag{4.34}$$

The effective fission production operator $\boldsymbol{F_{eff}}$ is identical to the one for the neutron diffusion equation in Eq. (3.75), and the delayed source contribution factors $\xi_{k'\to k}$ and $\xi_{C\to k}$ are provided in Table 3.2. Once we solve Eq. (4.33) and obtain the neutron flux $\phi_k(x)$, we can calculate the precursor concentration solution $C_k(x)$ as follows:

$$C_k(x) = \xi_{C\to k}C_{k-1}(x) + \frac{1}{\lambda}\sum_{k'=k-2}^{k}\xi_{k'\to k}\boldsymbol{F_{d,k'}}(x)\phi_{k'}(x)\,. \tag{4.35}$$

We note that the source iteration method has been applied to Eq. (4.33), as shown by the iteration indexes $^{(l)}$. This method is hereafter referred to as BE-SI. By inspection, it is evident that BE-SI Eq. (4.33) is similar to the steady-state SI Eq. (4.16), except for the time absorption cross-section $1/v\Delta t_k$ augmenting the transport operator, the effective fission production operator, and the effective independent source. Therefore, we can directly apply the neutron transport methods SN-DD and MOC to discretize and solve BE-SI Eq. (4.33); the resulting methods are hereafter referred to as "BE-SI with SN-DD" and "BE-SI with MOC", respectively.

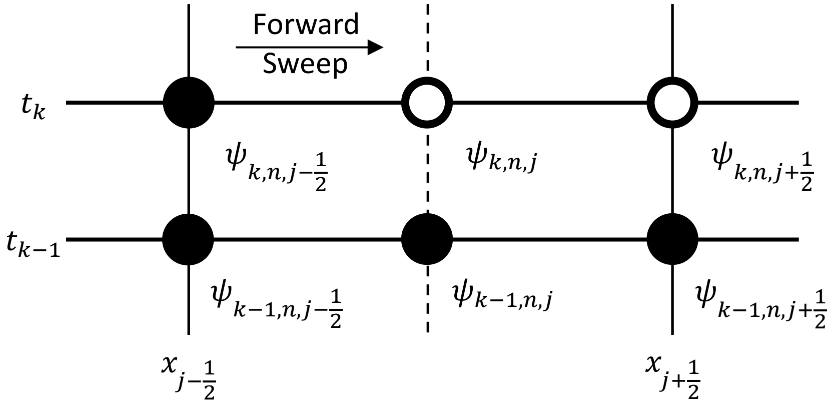

Forward
Sweep
t_k
t_{k-1}
$\psi_{k,n,j-\frac{1}{2}}$
$\psi_{k,n,j}$
$\psi_{k,n,j+\frac{1}{2}}$
$\psi_{k-1,n,j-\frac{1}{2}}$
$\psi_{k-1,n,j}$
$\psi_{k-1,n,j+\frac{1}{2}}$
$x_{j-\frac{1}{2}}$
$x_{j+\frac{1}{2}}$

Figure 4.1: Illustration of BE forward ($\mu_n > 0$) transport sweep

At each iteration, BE-SI (either with SN-DD or MOC) solves the discretized transport problem by means of forward and backward transport sweeps. A transport sweep is performed for each directional index n. Figure 4.1 illustrates how the forward transport sweep of BE is performed. The broken vertical line indicates the cell-average quantities. At each step of a forward sweep, we solve for the hollow nodes (cell-average $\psi_{k,n,j}$ and outgoing flux $\psi_{k,n,j+1/2}$) from the given solid nodes. We know the solutions at t_{k-1} ($\psi_{k-1,n,j-1/2}$, $\psi_{k-1,n,j}$, and

$\psi_{k-1,n,j+1/2}$) from the previous time-step solve or the initial condition, and we know the *incoming* flux solution at t_k ($\psi_{k,n,j-1/2}$) from the previous forward sweep solve or the boundary condition. At each forward sweep, SN-DD [in the form of Eqs. (4.24) and (4.27)] essentially solves a 2×2 matrix problem, while MOC computes the BE-SI version of Eqs. (4.30) and (4.31), which includes evaluation of exponential function $\exp\left[-\left(1+1/\eta_j\right)\tau_{n,j}\right]$ that depends on the material property $\Sigma_{t,k,j}$ and the numerical discretization parameters Δx_j, μ_n, and Δt_k.

Now, by applying MBTD to the time-dependent neutron transport Eq. (4.5), we obtain the following MBTD coupled balance equations:

$$\begin{cases} \boldsymbol{T_{k-1/2}}(x,\mu)\psi_{k-1/2}^{(l)}(x,\mu) + \dfrac{1}{v\Delta t_k}\psi_k^{(l)}(x,\mu) \\ \qquad = \left[\boldsymbol{S_{k-1/2}}(x) + \dfrac{\boldsymbol{F_{eff,k-1/2}}(x)}{2}\right]\phi_{k-1/2}^{(l-1)}(x) + \xi_{k\to k-1/2}\dfrac{\boldsymbol{F_{d,k}}(x)}{2}\phi_k^{(l-1)}(x) \\ \qquad\quad + Q_{eff,k-1/2}(x,\mu), \\ -\dfrac{2}{v\Delta t_k}\psi_{k-1/2}^{(l)}(x,\mu) + \left[\boldsymbol{T_k}(x,\mu) + \dfrac{2}{v\Delta t_k}\right]\psi_k^{(l)}(x,\mu) \\ \qquad = \xi_{k-1/2\to k}\dfrac{\boldsymbol{F_{d,k-1/2}}(x)}{2}\phi_{k-1/2}^{(l-1)}(x) + \left[\boldsymbol{S_k}(x) + \dfrac{\boldsymbol{F_{eff,k}}(x)}{2}\right]\phi_k^{(l-1)}(x) \\ \qquad\quad + Q_{eff,k}(x,\mu), \end{cases} \tag{4.36}$$

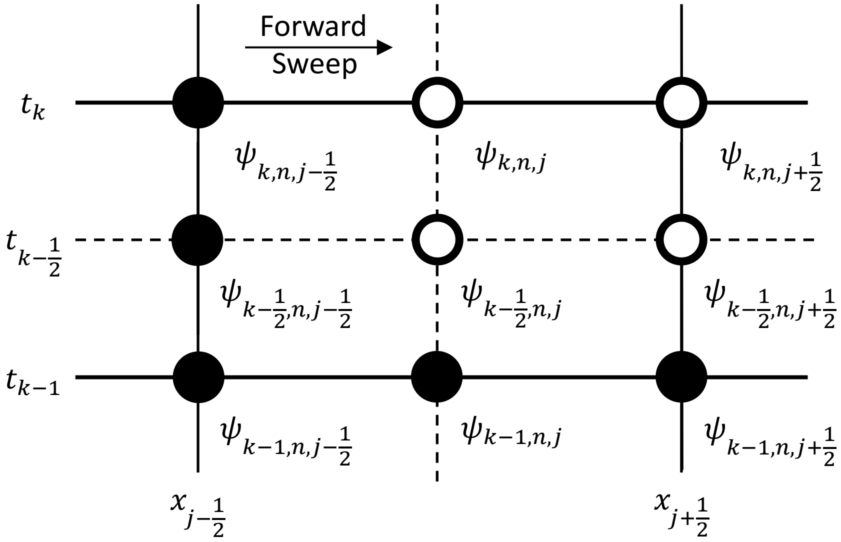
where the effective independent sources are

$$\begin{aligned} Q_{eff,k-1/2}(x,\mu) = Q_{k-1/2}(x,\mu) + \frac{1}{v\Delta t_k}\psi_{k-1}(x,\mu) \\ + \frac{1}{2}\left[\xi_{k\to k-1/2}\boldsymbol{F_{d,k-1}}(x)\phi_{k-1}(x) + \xi_{C\to k-1/2}\lambda C_{k-1}(x)\right], \end{aligned} \tag{4.37}$$

$$Q_{eff,k}(x,\mu) = Q_k(x,\mu) + \frac{1}{2}\left[\xi_{k-1\to k}\boldsymbol{F_{d,k-1}}(x)\phi_{k-1}(x) + \xi_{C\to k}\lambda C_{k-1}(x)\right]. \tag{4.38}$$

The effective fission production operator $\boldsymbol{F_{eff}}$ is similarly defined as Eq. (3.75), and the delayed source contribution factors $\xi_{k'\to}$ and $\xi_{C\to}$ are provided in Table 3.3 for MBTD standard delayed source formulation. Once we solve Eq. (4.33) and obtain the neutron fluxes $\phi_k(x)$ and $\phi_{k-1/2}(x)$, we can calculate the precursor concentration solution $C_k(x)$ as follows:

$$C_k(x) = \xi_{C\to k} C_{k-1}(x) + \frac{1}{\lambda}\sum_{i=-2}^{0} \xi_{k+i/2\to k} \boldsymbol{F}_{\boldsymbol{d},\boldsymbol{k+i/2}}(x)\phi_{k+i/2}(x). \tag{4.39}$$



Forward
Sweep
t_k
$t_{k-\frac{1}{2}}$
t_{k-1}
$\psi_{k,n,j-\frac{1}{2}}$
$\psi_{k,n,j}$
$\psi_{k,n,j+\frac{1}{2}}$
$\psi_{k-\frac{1}{2},n,j-\frac{1}{2}}$
$\psi_{k-\frac{1}{2},n,j}$
$\psi_{k-\frac{1}{2},n,j+\frac{1}{2}}$
$\psi_{k-1,n,j-\frac{1}{2}}$
$\psi_{k-1,n,j}$
$\psi_{k-1,n,j+\frac{1}{2}}$
$x_{j-\frac{1}{2}}$
$x_{j+\frac{1}{2}}$

Figure 4.2: Illustration of MBTD forward ($\mu_n > 0$) transport sweep

We note that a source iteration method has been applied to the MBTD coupled balance equations Eq. (4.36) as shown by the iteration index $^{(l)}$; this method is hereafter referred to as MBTD-SI. We wish to discretize MBTD-SI Eq. (4.36) such that at each iteration we can solve the discretized problem by means of forward and backward transport sweeps similar to that of BE-SI. Figure 4.2 illustrates how we plan to perform the forward transport sweep of MBTD. The vertical and horizontal broken lines respectively indicate the cell-average and the mid-point in time quantities. At each step of a forward sweep, we solve for the four hollow nodes ($\psi_{k-1/2,n,j}$, $\psi_{k-1/2,n,j+1/2}$, $\psi_{k,n,j}$, and $\psi_{k,n,j+1/2}$) from the given solid nodes. It is evident that this transport sweep procedure *simultaneously* sweeps and solves for the solutions at $t_{k-1/2}$ and t_k. This means it is essentially the Simultaneous Solve strategy, and we can see how it relates to the modified-GS (with respect to the standard GS) illustrated in Figure 2.6.

Before we discuss the adaptation of the neutron transport methods SN-DD and MOC for MBTD-SI in the next paragraph, we compare the memory complexity of MBTD-SI and BE-SI. In the Simultaneous Solve strategy for solving *neutron diffusion* Eq. (3.82), we double the number of neutron flux unknowns ϕ. In addition to the time-step solution ϕ_k, we need to store the mid-point solution $\phi_{k-1/2}$ as well, each of size $(J\times 1)$. This makes the memory complexity of MBTD twice that of BE. In the *neutron transport* problem, however, the memory complexity is dominated by the angular neutron flux ψ of the size $(J\times N)$. While it is not necessary to store all $(J\times N)$ of ψ during the source iteration of solving a steady-state problem, in a time-dependent problem we need to store the next time-step solution ψ_k because it will be used as an independent source term for the next time-step calculation. Therefore, both MBTD-SI and BE-SI need to store ψ_k. However, while it is true that MBTD-SI needs to store ϕ_k and $\phi_{k+1/2}$, it does not need to store $\psi_{k-1/2}$. Therefore, both MBTD-SI and BE-SI have similar memory complexities dominated by ψ_k of size $(J\times N)$.

We can directly derive an SN-DD formulation for MBTD-SI Eq. (4.36). This gives us a tridiagonal 4 × 4 matrix problem (instead of just 2 × 2 like the one for BE) to solve at each transport sweep step. While in practice it is not necessary to actually create and solve the tridiagonal 4 × 4 (or 2 × 2 for BE) matrix problem, this observation effectively compares the computational complexity of BE and MBTD transport sweeps with SN-DD.

Adapting MOC for MBTD, unfortunately, is not as straightforward as SN-DD. Reflecting on the main idea of MOC, we can attempt to formulate its adaptation for MBTD as follows. We would like to *analytically* solve Eq. (4.36) for each spatial step j by assuming constant RHS sources:

$$\begin{cases} \left(\mu_n\dfrac{d}{dx}+\Sigma_{t,k-1/2,j}\right)\psi_{k-1/2,n}^{(l)}(x)+\dfrac{1}{v\Delta t_k}\psi_{k,n}^{(l)}(x)=\hat{Q}_{k-1/2,n,j}^{(l-1)}, \\ -\dfrac{2}{v\Delta t_k}\psi_{k-1/2,n}^{(l)}(x)+\left(\mu_n\dfrac{d}{dx}+\Sigma_{t,k,j}+\dfrac{2}{v\Delta t_k}\right)\psi_{k,n}^{(l)}(x)=\hat{Q}_{k,n,j}^{(l-1)}, \end{cases} \tag{4.40}$$

where

$$\hat{Q}_{k-1/2,n,j}^{(l-1)}=\left(\boldsymbol{S_{k-1/2,j}}+\frac{\boldsymbol{F_{eff,k-1/2,j}}}{2}\right)\phi_{k-1/2,j}^{(l-1)}+\xi_{k\to k-1/2}\frac{\boldsymbol{F_{d,k,j}}}{2}\phi_{k,j}^{(l-1)}+Q_{eff,k-1/2,n,j}, \tag{4.41}$$

$$\hat{Q}_{k,n,j}^{(l-1)}=\xi_{k-1/2\to k}\frac{\boldsymbol{F_{d,k-1/2,j}}}{2}\phi_{k-1/2,j}^{(l-1)}+\left(\boldsymbol{S_{k,j}}+\frac{\boldsymbol{F_{eff,k,j}}}{2}\right)\phi_{k+1,j}^{(l-1)}+Q_{eff,k+1,n,j}. \tag{4.42}$$

Unlike the heterogeneous first-order linear ODE solved in MOC of BE [which is structurally equivalent to the one found in the steady-state MOC Eq. (4.28)], here for MBTD we have a system of heterogeneous first-order linear ODEs that needs to be solved at each step of the transport sweep. One can analytically solve Eq. (4.28) by means of a matrix method and find that the matrix of the system of ODEs has *complex conjugate eigenvalues*. It follows that in calculating the solutions ($\psi_{k-1/2,n,j}$, $\psi_{k-1/2,n,j+1/2}$, $\psi_{k,n,j}$, and $\psi_{k,n,j+1/2}$), we need to evaluate not only the exponential function $\exp\left[-\left(1+1/\eta_j\right)\tau_{n,j}\right]$—which is the only "non-trivial" function to be evaluated in BE—but also trigonometry functions $\sin\left(\tau_{n,j}/\eta_j\right)$ and $\cos\left(\tau_{n,j}/\eta_j\right)$. Later in Section 4.6.2, it is found that the evaluation of the trigonometry functions takes a large portion of the overall computational work, such that a considerable speedup can be obtained by storing the required values of the nontrivial function evaluations. Complete derivations of SN-DD and MOC for MBTD are presented in Appendix A.

4.4 Fourier Analysis

In the previous section, we formulated BE-SI and MBTD-SI with SN-DD and MOC, and compared their computational complexities per transport source iteration. Another factor that typically dominates the overall computational work of an iterative method is the convergence rate, which represents how quickly the iteration converges and gives a rough idea of how many iterations are needed to obtain the solutions with the desired accuracy. In this section, a Fourier analysis (FA) [48][12] is performed to characterize and compare the convergence rate of BE-SI and MBTD-SI.

Let us consider a simple non-fission infinite ($-\infty < x < \infty$) homogeneous medium problem with isotropic scattering. The equation for BE is

$$\begin{cases} \mu\dfrac{\partial\psi_k}{\partial x}+\left(\Sigma_t+\dfrac{1}{v\Delta t}\right)\psi_k(x,\mu)=\dfrac{1}{2}\Sigma_s\phi_k(x)+\dfrac{1}{v\Delta t}\psi_{k-1}(x,\mu), \\ \phi_k(x)=\displaystyle\int_{-1}^{1}\psi_k(x,\mu)d\mu\,. \end{cases} \tag{4.43}$$

BE-SI solves Eq. (4.43) as follows:

$$\begin{cases} \mu\frac{\partial\psi_k^{(l)}}{\partial x}+\left(\Sigma_t+\frac{1}{v\Delta t}\right)\psi_k^{(l)}(x,\mu)=\frac{1}{2}\Sigma_s\phi_k^{(l-1)}(x)+\frac{1}{v\Delta t}\psi_{k-1}(x,\mu), \\ \phi_k^{(l)}(x)=\int_{-1}^{1}\psi_k^{(l)}(x,\mu)d\mu\,, \end{cases} \tag{4.44}$$

The *iteration error equation* for BE-SI can be obtained by subtracting Eq. (4.44) from Eq. (4.43):

$$\begin{cases} \mu\frac{\partial f^{(l)}}{\partial x}+\left(\Sigma_t+\frac{1}{v\Delta t}\right)f^{(l)}(x,\mu)=\frac{1}{2}\Sigma_s F^{(l-1)}(x), \\ F^{(l)}(x)=\int_{-1}^{1}f^{(l)}(x,\mu)d\mu\,, \end{cases} \tag{4.45}$$

where the *iterate errors* at iteration number (l) are defined as

$$\begin{cases} f^{(l)}(x,\mu)=\psi_k(x,\mu)-\psi_k^{(l)}(x,\mu), \\ F^{(l)}(x)=\phi_k(x)-\phi_k^{(l)}(x). \end{cases} \tag{4.46}$$

Next, we introduce the typical *Fourier ansatz* [12][48] for neutron transport source iterations:

$$f^{(l)}(x,\mu)=\theta^{l-1}a(\mu)e^{i\omega\Sigma_t x}, \qquad F^{(l)}(x)=\theta^{l}Ae^{i\omega\Sigma_t x}. \tag{4.47}$$

Here A and $a(\mu)$ are Fourier coefficients associated with the Fourier mode ω, and θ is the eigenvalue of the iteration method, which acts as an error amplification factor for the Fourier mode ω. The ω dependency of A, $a(\mu)$, and θ in the Fourier ansatz Eq. (4.47) is not written for simplicity. It is evident that the iteration method converges if $|\theta|<1$ for all ω, such that the error $f^{(l)}(x,\mu)$ decreases as l increases (or as we keep iterating). The main objective of this Fourier analysis is to determine θ and the *spectral radius* ρ of the iteration method:

$$\rho=\max_{-\infty<\omega<\infty}|\theta(\omega)|\,. \tag{4.48}$$

By substituting the Fourier ansatz Eq. (4.47) into the iteration error Eq. (4.45), we obtain (for BE-SI):

$$a(\omega,\mu)=\frac{c}{2}\frac{1}{\left(1+\frac{1}{\eta}\right)+i\mu\omega}, \tag{4.49}$$

$$\theta_{\text{BE-SI}}(\omega)=\frac{c}{\omega}\arctan\left(\frac{\omega}{1+1/\eta}\right), \tag{4.50}$$

$$\rho_{\text{BE-SI}}=\lim_{\omega\to 0}|\theta_{\text{BE-SI}}(\omega)|=c\frac{1}{1+1/\eta}. \tag{4.51}$$

where we use the non-dimensional parameters $c=\Sigma_s/\Sigma_t$ and $\eta=v\Sigma_t\Delta t$. Before we discuss the significance of the Fourier analysis results of BE-SI [Eqs. (4.49)–(4.51)], the Fourier analysis of MBTD-SI is discussed next.

The coupled balance equations for MBTD are

$$\begin{cases}\left(\mu\frac{\partial}{\partial x}+\Sigma_t\right)\psi_{k-1/2}(x,\mu)+\frac{1}{v\Delta t}\psi_k(x,\mu)=\frac{1}{2}\left(\Sigma_s+v\Sigma_f\right)\phi_{k-1/2}(x)+\frac{1}{v\Delta t}\psi_{k-1}(x,\mu),\\ -\frac{2}{v\Delta t}\psi_{k-1/2}(x,\mu)+\left[\mu\frac{\partial}{\partial x}+\left(\Sigma_t+\frac{2}{v\Delta t}\right)\right]\psi_k(x,\mu)=\frac{1}{2}\left(\Sigma_s+v\Sigma_f\right)\phi_k(x),\\ \phi_{k-1/2}(x)=\int_{-1}^{1}\psi_{k-1/2}(x,\mu)d\mu, \qquad \phi_k(x)=\int_{-1}^{1}\psi_k(x,\mu)d\mu.\end{cases} \tag{4.52}$$

MBTD-SI solves Eq. (4.52) as follows:

$$\begin{cases}\left(\mu\frac{\partial}{\partial x}+\Sigma_t\right)\psi_{k-1/2}^{(l)}(x,\mu)+\frac{1}{v\Delta t}\psi_k^{(l)}(x,\mu)=\frac{1}{2}\left(\Sigma_s+v\Sigma_f\right)\phi_{k-1/2}^{(l-1)}(x)+\frac{1}{v\Delta t}\psi_{k-1}(x,\mu),\\ -\frac{2}{v\Delta t}\psi_{k-1/2}^{(l)}(x,\mu)+\left[\mu\frac{\partial}{\partial x}+\left(\Sigma_t+\frac{2}{v\Delta t}\right)\right]\psi_k^{(l)}(x,\mu)=\frac{1}{2}\left(\Sigma_s+v\Sigma_f\right)\phi_k^{(l-1)}(x),\\ \phi_{k-1/2}^{(l)}(x)=\int_{-1}^{1}\psi_{k-1/2}^{(l)}(x,\mu)d\mu, \qquad \phi_k^{(l)}(x)=\int_{-1}^{1}\psi_k^{(l)}(x,\mu)d\mu.\end{cases} \tag{4.53}$$

The *iteration error equation* for MBTD-SI can be obtained by subtracting Eq. (4.53) from Eq. (4.52):

$$\begin{cases} \mu\dfrac{\partial g^{(l)}}{\partial x}+\Sigma_t g^{(l)}(x,\mu)+\dfrac{1}{v\Delta t}f^{(l)}(x,\mu)=\dfrac{1}{2}\left(\Sigma_s+\nu\Sigma_f\right)G^{(l-1)}(x) \\ -\dfrac{2}{v\Delta t}g^{(l)}(x,\mu)+\mu\dfrac{\partial f^{(l)}}{\partial x}+\left(\Sigma_t+\dfrac{2}{v\Delta t}\right)f^{(l)}(x,\mu)=\dfrac{1}{2}\left(\Sigma_s+\nu\Sigma_f\right)F^{(l-1)}(x), \\ G^{(l)}(x)=\displaystyle\int_{-1}^{1}g^{(l)}(x,\mu)d\mu\,, \qquad F^{(l)}(x)=\int_{-1}^{1}f^{(l)}(x,\mu)d\mu\,. \end{cases} \tag{4.54}$$

where the iterate errors at iteration number (l) are defined as

$$\begin{cases} g^{(l)}(x,\mu)=\psi_{k-1/2}(x,\mu)-\psi_{k-1/2}^{(l)}(x,\mu), \qquad f^{(l)}(x,\mu)=\psi_k(x,\mu)-\psi_k^{(l)}(x,\mu), \\ G^{(l)}(x)=\phi_{k-1/2}(x)-\phi_{k-1/2}^{(l)}(x), \qquad F^{(l)}(x)=\phi_k(x)-\phi_k^{(l)}(x). \end{cases} \tag{4.55}$$

Referring to Eq. (4.47), we consider the following Fourier ansatz:

$$\begin{cases} g^{(l)}(x,\mu)=\theta^{l-1}b(\mu)e^{i\omega\Sigma_t x}, \qquad G^{(l)}(x)=\theta^{l}Be^{i\omega\Sigma_t x}, \\ f^{(l)}(x,\mu)=\theta^{l-1}a(\mu)e^{i\omega\Sigma_t x}, \qquad F^{(l)}(x)=\theta^{l}Ae^{i\omega\Sigma_t x}. \end{cases} \tag{4.56}$$

Again, for simplicity, the Fourier mode ω dependency of the eigenvalue θ and Fourier coefficients A, B, $a(\mu)$, and $b(\mu)$ is not written. By introducing the Fourier ansatz Eq. (4.56) into Eq. (4.54), we obtain

$$\begin{cases} b(\mu)=\dfrac{c}{2}\left[\dfrac{h_1(\omega,\mu)}{h_3(\omega,\mu)}B-\dfrac{1}{\eta}\dfrac{h_{12}(\omega,\mu)}{h_3(\omega,\mu)}A\right], \\ a(\mu)=\dfrac{c}{2}\left[\dfrac{2}{\eta}\dfrac{h_{12}(\omega,\mu)}{h_3(\omega,\mu)}B+\dfrac{h_2(\omega,\mu)}{h_3(\omega,\mu)}A\right], \end{cases} \tag{4.57}$$

and

$$\begin{cases} \theta B=I_1(\omega)B-\dfrac{1}{\eta}I_{12}(\omega)A, \\ \theta A=\dfrac{2}{\eta}I_{12}(\omega)B+I_2(\omega)A, \end{cases} \tag{4.58}$$

where

$$\begin{cases} h_1(\omega,\mu) = \dfrac{1}{1+i\omega\mu}, \qquad h_2(\omega,\mu) = \dfrac{1}{(1+2/\eta)+i\omega\mu}, \\ h_{12}(\omega,\mu) = h_1(\omega,\mu)h_2(\omega,\mu), \qquad h_3(\omega,\mu) = 1+\dfrac{2}{\eta^2}h_{12}(\omega,\mu), \end{cases} \tag{4.59}$$

$$\begin{cases} I_1(\omega) = \dfrac{c}{2}\displaystyle\int_{-1}^{1}\dfrac{h_1(\omega,\mu)}{h_3(\omega,\mu)}d\mu\,, \qquad I_2(\omega) = \dfrac{c}{2}\displaystyle\int_{-1}^{1}\dfrac{h_2(\omega,\mu)}{h_3(\omega,\mu)}d\mu\,, \\ I_{12}(\omega) = \dfrac{c}{2}\displaystyle\int_{-1}^{1}\dfrac{h_{12}(\omega,\mu)}{h_3(\omega,\mu)}d\mu\,. \end{cases} \tag{4.60}$$

Rearranging Eq. (4.58) in terms of the eigenvalue θ, we obtain:

$$\theta^2+\tilde{b}(\omega)\theta+\tilde{c}(\omega) = 0, \tag{4.61}$$

where,

$$\tilde{b}(\omega) = -[I_1(\omega)+I_2(\omega)], \qquad \tilde{c}(\omega) = I_1(\omega)I_2(\omega)+\frac{2}{\eta^2}I_{12}(\omega)^2. \tag{4.62}$$

Therefore, the eigenvalue of MBTD-SI is:

$$\theta_{\text{MBTD-SI}}(\omega) = \frac{-\tilde{b}(\omega)\pm\sqrt{\tilde{b}^2(\omega)-4\tilde{c}(\omega)}}{2}. \tag{4.63}$$

One can plot $|\theta_{\text{MBTD-SI}}(\omega)|$ as a function of ω (as shown in Figure 4.3) to find that the magnitude of the eigenvalue maximizes as $\omega \to 0$, similar to that of BE-SI in Eq. (4.50). This means the spectral radius can be determined as

$$\rho_{\text{MBTD-SI}} = \lim_{\omega\to 0}|\theta_{\text{MBTD-SI}}(\omega)| = \left|c\frac{\eta^2+\eta\pm i\eta}{\eta^2+2\eta+2}\right| = c\frac{1}{\sqrt{1+2/\eta+2/\eta^2}}, \tag{4.64}$$

where it turns out that the leading eigenvalues of MBTD-SI are a *pair of complex conjugates*.

The Fourier analysis results for BE-SI and MBTD-SI are summarized in Figure 4.3 and Figure 4.4. Figure 4.3 presents the eigenvalue $|\theta(\omega)|$ as a function of the Fourier mode ω at a certain value of c and η, while Figure 4.4 presents the spectral radius ρ as a function of η at the largest possible value of $c = 1.0$. The spectral radius of an iteration method represents the worst-case estimate of error reduction factor per iteration; the smaller ρ, the faster the convergence. It is found that convergence is guaranteed by both methods regardless of η (or Δt), because ρ is always smaller than 1 (note that $\rho \to c$ as $\eta \to \infty$). This verifies the applicability of the proposed method MBTD-SI. Furthermore, it is found that BE-SI and MBTD-SI have very similar spectral radii, and thus convergence rates. This means the number of iterations required by BE-SI and MBTD-SI to achieve convergence is about the same.

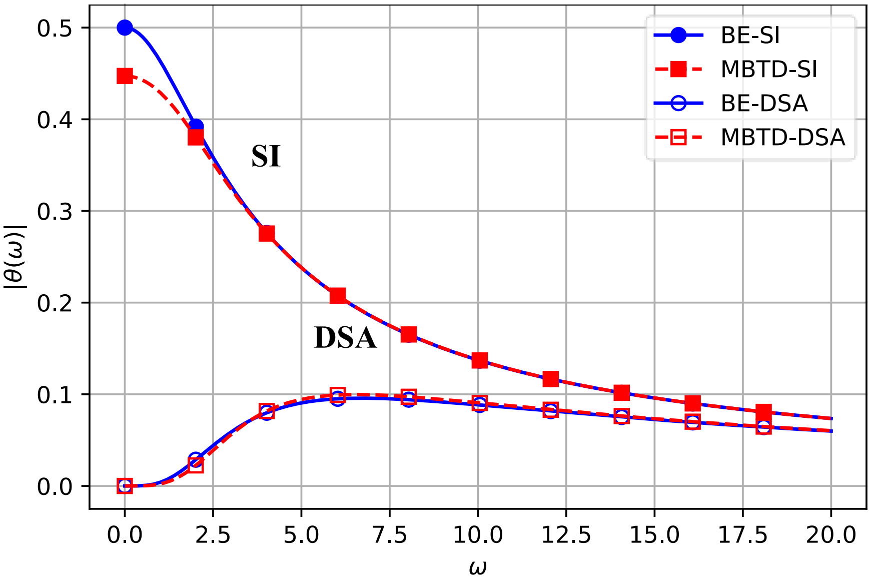
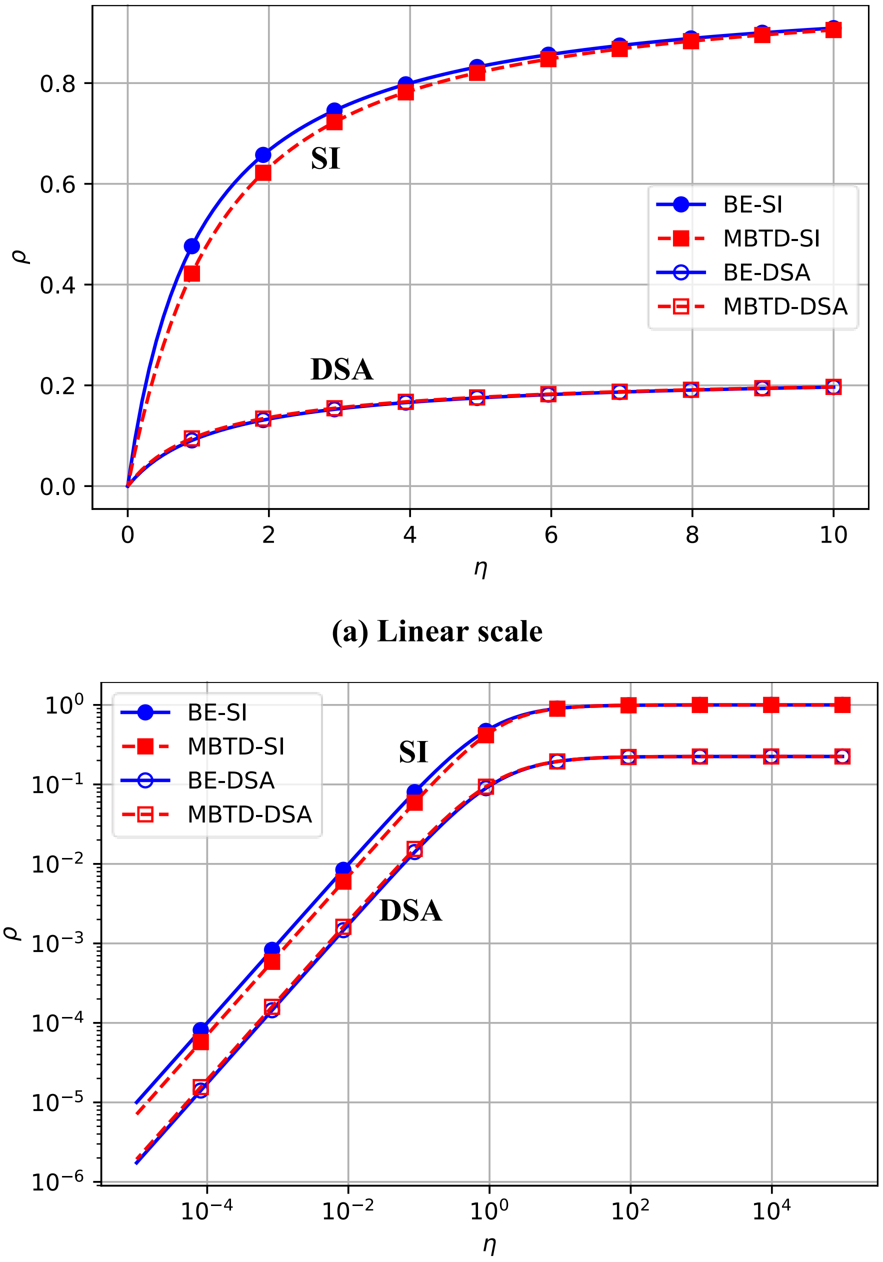


Figure 4.3: Eigenvalues of SI and DSA for $c = \eta = 1.0$

The Fourier analysis performed in this section assumes a system without fission. However, we can easily replace c with $k + (1 - k)c$ for systems with fission, where for the infinite medium problem $k = \nu\Sigma_f/\Sigma_a$. In a supercritical system ($k > 1$), the factor $k + (1 - k)c$ is larger than one. This means that a sufficiently small η (or Δt) is needed for convergence ($\rho <$ 1) in simulating a supercritical system, and this generally applies to both BE-SI and MBTD-SI.



(a) Linear scale

(b) Log scale

Figure 4.4: Theoretical spectral radius of SI and DSA for $c = 1$

4.5 Diffusion Synthetic Acceleration

Nuclear reactor simulation problems are typically characterized by a large dimension (relative to the optical thickness) and high scattering ratio c. This makes a huge challenge for BE-SI and MBTD-SI (or source iteration methods in general), whose spectral radius is proportional to c, as discussed in the previous section. To address this issue, a diffusion-based acceleration method is usually performed in-between the source iterations. In this chapter, we derive, and then Fourier-analyze, Diffusion Synthetic Accelerations (DSA) for BE-SI and MBTD-SI, which hereafter are referred to as BE-DSA and MBTD-DSA, respectively.

A theoretical justification on why a diffusion method can effectively accelerate the convergence of BE-SI and MBTD-SI can be obtained by observing the Fourier coefficients in Eqs. (4.49) and (4.57), respectively. In the previous section, we learned that the most persisting error modes (the largest $|\theta|$) correspond to $\omega \to 0$. By Taylor-expanding the coefficients $a(\omega, \mu)$ and $b(\omega, \mu)$ in Eqs. (4.49) and (4.57) and taking the limit $\omega \to 0$, one can find that the most slowly converging error modes of both methods are nearly isotropic or weakly dependent on μ. A quantity that is weakly dependent on direction, which in this case is the iteration error, can be very well captured and *eliminated* by the neutron diffusion approximation.

4.5.1 BE-DSA

Let us consider an adaptation of the more general BE problem in Eq. (4.33):

$$\begin{cases} \mu \dfrac{\partial \psi_k}{\partial x} + \left[\Sigma_{t,k}(x) + \dfrac{1}{v\Delta t_k}\right] \psi_k(x,\mu) \\ \qquad\qquad = \dfrac{1}{2}\left[\Sigma_{s,k}(x) + \boldsymbol{F_{eff,k}}(x)\right]\phi_k(x) + Q_{eff,k}(x,\mu), \\ \qquad\qquad \phi_k(x) = \displaystyle\int_{-1}^{1} \psi_k(x,\mu) d\mu\,, \\ \qquad \psi_k(0,\mu) = \psi_L(\mu), \qquad \mu \in [0,1], \\ \quad \psi_k(X,\mu) = \psi_k(X,-\mu), \qquad \mu \in [-1,0]. \end{cases} \tag{4.65}$$

Here we have a left-incident-flux [for vacuum $\psi_L(\mu) = 0$] and right-reflecting boundary condition. In BE-DSA, each iteration starts by normally calculating a source iterate solution of BE-SI, but here the iterate solution is considered to be an *intermediate solution*, which is denoted by superscript $^{(l-1/2)}$ as shown in the following:

$$\begin{cases} \mu \dfrac{\partial \psi_k^{(l-1/2)}}{\partial x} + \left[\Sigma_{t,k}(x) + \dfrac{1}{v\Delta t_k}\right] \psi_k^{(l-1/2)}(x,\mu) \\ \qquad = \dfrac{1}{2}\left[\Sigma_{s,k}(x) + \boldsymbol{F_{eff,k}}(x)\right]\phi_k^{(l-1)}(x) + Q_{eff,k}(x,\mu), \\ \phi_k^{(l-1/2)}(x) = \displaystyle\int_{-1}^{1} \psi_k^{(l-1/2)}(x,\mu)d\mu\,, \\ \psi_k^{(l-1/2)}(0,\mu) = \psi_L(\mu), \qquad \mu \in [0,1], \\ \psi_k^{(l-1/2)}(X,\mu) = \psi_k^{(l-1/2)}(X,-\mu), \qquad \mu \in [-1,0]. \end{cases} \tag{4.66}$$

By subtracting Eq. (4.66) from the "exact" Eq. (4.65), and introducing iteration errors

$$\begin{cases} f(x,\mu) = \psi_k(x,\mu) - \psi_k^{(l-1/2)}(x,\mu), \\ F_n(x) = \displaystyle\int_{-1}^{1} \mu^n f(x,\mu)d\mu\,, \end{cases} \tag{4.67}$$

we obtain the following exact error equation:

$$\begin{cases} \mu \dfrac{\partial f}{\partial x} + \left[\Sigma_{t,k}(x) + \dfrac{1}{v\Delta t_k}\right] f(x,\mu) \\ \qquad = \dfrac{1}{2}\left[\Sigma_{s,k}(x) + \boldsymbol{F_{eff,k}}(x)\right]\left[F_0(x) + \phi_k^{(l-1/2)}(x) - \phi_k^{(l-1)}(x)\right], \\ f(0,\mu) = 0, \qquad \mu \in [0,1], \\ f(X,\mu) = f(X,-\mu), \qquad \mu \in [-1,0]. \end{cases} \tag{4.68}$$

We note that there is no error in the left-incident-flux boundary.

By taking the zeroth angular moment, or $\int_{-1}^{1}(\cdot)d\mu$, of the interior equation in Eq. (4.68), we obtain:

$$\begin{aligned} &\frac{dF_1}{dx} + \left[\Sigma_{a,k}(x) - \boldsymbol{F_{eff,k}}(x) + \frac{1}{v\Delta t_k}\right] F_0(x) \\ &\qquad = \left[\Sigma_{s,k}(x) + \boldsymbol{F_{eff,k}}(x)\right]\left[\phi_k^{(l-1/2)}(x) - \phi_k^{(l-1)}(x)\right]; \end{aligned} \tag{4.69}$$

while by taking the first angular moment, or $\int_{-1}^{1}\mu(\cdot)d\mu$, we obtain:

$$\frac{dF_2}{dx}+\left[\Sigma_{t,k}(x)+\frac{1}{v\Delta t_k}\right]F_1(x)=0. \tag{4.70}$$

However, by using the *P1 approximation*,

$$f(x,\mu)=\frac{1}{2}F_0(x)+\frac{3}{2}\mu F_1(x)\ \rightarrow\ F_2\approx\frac{1}{3}F_0(x), \tag{4.71}$$

Eq. (4.70) gives

$$F_1(x)=-\widehat{D}(x)\frac{dF_0}{dx},\qquad \widehat{D}(x)=\frac{1}{3\Sigma_{t,k}(x)\left(1+\frac{1}{\eta_k}\right)}. \tag{4.72}$$

Finally, substituting Eq. (4.72) into Eq. (4.69) we obtain

$$\begin{aligned}-\frac{d}{dx}\widehat{D}(x)\frac{dF_0}{dx}+\left[\Sigma_{a,k}(x)-\boldsymbol{F_{eff,k}}(x)+\frac{1}{v\Delta t_k}\right]F_0(x)\\ =\left[\Sigma_{s,k}(x)+\boldsymbol{F_{eff,k}}(x)\right]\left[\phi_k^{(l-1/2)}(x)-\phi_k^{(l-1)}(x)\right].\end{aligned} \tag{4.73}$$

Now we need boundary conditions. Respectively operating the left and right boundary conditions in Eq. (4.68) with $\int_0^1\mu(\cdot)d\mu$ and $\int_{-1}^{0}\mu(\cdot)d\mu$, while considering the P1 approximation Eqs. (4.71) and and (4.72), we obtain:

$$\begin{cases}\frac{1}{4}F_0(0)-\frac{1}{2}\widehat{D}(x)\left.\frac{dF_0}{dx}\right|_{x=0}=0,\\ \left.\frac{dF_0}{dx}\right|_{x=X}=0.\end{cases} \tag{4.74}$$

This suggests that flux-incident and reflecting neutron transport boundary conditions respectively translate into vacuum and reflecting boundary conditions for the error diffusion equation.

Equation (4.73) is essentially a steady-state diffusion for F_0 which can be numerically solved using the box-scheme FDM discussed in Section 3.2. Once we obtain F_0, we can use it to improve, or make a correction to, the intermediate solutions per Eqs. (4.67) and (4.71). We then declare the new, accelerated, iterate solutions $\phi_k^{(l)}$:

$$\phi_k^{(l)}(x) = \phi_k^{(l-1/2)}(x) + F_0(x). \tag{4.75}$$

Additionally, when convergence is achieved, we can calculate $\psi_k^{(l)}$ to be used as the independent source in the next time-step calculation as follows:

$$\psi_k^{(l)}(x,\mu) = \psi_k^{(l-1/2)}(x,\mu) + \frac{1}{2}F_0(x) + \frac{3}{2}\mu F_1(x), \tag{4.76}$$

where the cell-average of the error current $F_1(x)$ can be approximated as:

$$F_{1,j} = -\widehat{D}_j \frac{F_{0,j+1/2} - F_{0,j-1/2}}{\Delta x_j}, \qquad F_{0,j+1/2} = \frac{\left(\widehat{D}_j/\Delta x_j\right)F_j + \left(\widehat{D}_{j+1}/\Delta x_{j+1}\right)F_{j+1}}{\widehat{D}_j/\Delta x_j + \widehat{D}_{j+1}/\Delta x_{j+1}}. \tag{4.77}$$

This is consistent with the left- and right-handed derivative approximations of the box-scheme FDM Eq. (3.25). However, to be able to perform Eq. (4.76), we need to store $\psi_k^{(l-1/2)}$; this means another angular flux storage of size $(J \times N)$ is needed, in addition to the one for ψ_{k-1}. Otherwise, if we do not want to store $\psi_k^{(l-1/2)}$, we need to perform an extra transport iteration, during which we directly update ψ_{k-1} into $\psi_k^{(l)}$.

Now we summarize how BE-DSA works. First, we perform a normal transport sweep of BE-SI to solve for the intermediate solution $\phi_k^{(l-1/2)}$ per Eq. (4.66), either by using SN-DD or MOC. Next, we solve the diffusion Eq. (4.73) [with diffusion coefficient in Eq. (4.72) and BCs in Eq. (4.74)] by using the box-scheme FDM to obtain the error F_0. Finally, we calculate the new iterate solution $\phi_k^{(l)}$ per Eq. (4.75).

A Fourier analysis of BE-DSA can be similarly performed like that of BE-SI in Section 4.4 by using the following Fourier ansatz:

$$\begin{cases} f^{(l-1/2)}(x,\mu) = \theta^l \hat{a}(\mu) e^{i\omega\Sigma_t x}, \qquad F^{(l-1/2)}(x) = \theta^l \hat{A} e^{i\omega\Sigma_t x}, \\ \qquad F^{(l)}(x) = \theta^l A e^{i\omega\Sigma_t x}, \qquad F_0(x) = \theta^l \gamma e^{i\omega\Sigma_t x}, \end{cases} \tag{4.78}$$

where the iteration solution error $F^{(l)}$, and the intermediate solution errors $f^{(l-1/2)}$ and $F^{(l-1/2)}$, are defined as

$$\begin{cases} F^{(l)}(x) = \phi_k(x) - \phi_k^{(l)}(x), \\ f^{(l-1/2)}(x,\mu) = \psi_k(x,\mu) - \psi_k^{(l-1/2)}(x,\mu), \\ F^{(l-1/2)}(x) = \phi_k(x) - \phi_k^{(l-1/2)}(x), \end{cases} \tag{4.79}$$

and F_0 is essentially an estimate of the actual intermediate error $F^{(l-1/2)}$ per P_1 approximation Eq. (4.73).

The resulting eigenvalue and spectral radius of BE-DSA are:

$$\theta_{\text{BE-DSA}}(\omega) = \theta_{\text{BE-SI}}(\omega) - \frac{c[1-\theta_{\text{BE-SI}}(\omega)]}{1 - c + \frac{1}{\eta} + \frac{\omega^2}{3(1+1/\eta)}}, \tag{4.80}$$

$$\rho_{\text{BE-DSA}} = \lim_{-\infty<\omega<\infty} |\theta_{\text{BE-DSA}}(\omega)|, \tag{4.81}$$

which are respectively presented in Figure 4.3 and Figure 4.4. It is evident that BE-DSA effectively eliminates the most persisting error mode of BE-SI, which occurs as $\omega \to 0$. The spectral radius of $\rho_{\text{BE-DSA}}$ can be obtained by searching for the maximum $|\theta_{\text{BE-DSA}}(\omega)|$; in this thesis work, SciPy's optimization function [47] is used to determine ω at which $|\theta_{\text{BE-DSA}}(\omega)|$ maximizes.

4.5.2 MBTD-DSA

Next, we formulate MBTD-DSA by following the derivation of BE-DSA in the previous section. The following MBTD coupled balance equations are considered:

$$
\begin{cases}
\left[\mu\dfrac{\partial}{\partial x}+\Sigma_{t,k-1/2}(x)\right]\psi_{k-1/2}^{(l-1/2)}(x,\mu)+\dfrac{1}{v\Delta t_k}\psi_k^{(l-1/2)}(x,\mu) \\
\qquad =\dfrac{1}{2}\left[\Sigma_{s,k-1/2}(x)+\boldsymbol{F}_{\boldsymbol{eff},k-1/2}(x)\right]\phi_{k-1/2}^{(l-1)}(x)+\xi_{k\to k-1/2}\dfrac{\boldsymbol{F}_{\boldsymbol{d},\boldsymbol{k}}(x)}{2}\phi_k^{(l-1)}(x) \\
\qquad\quad +Q_{eff,k-1/2}(x,\mu), \\
-\dfrac{2}{v\Delta t_k}\psi_{k-1/2}^{(l-1/2)}(x,\mu)+\left\{\mu\dfrac{\partial}{\partial x}+\left[\Sigma_{t,k}(x)+\dfrac{2}{v\Delta t_k}\right]\right\}\psi_k^{(l-1/2)}(x,\mu) \\
\qquad =\xi_{k-1/2\to k}\dfrac{\boldsymbol{F}_{\boldsymbol{d},k-1/2}(x)}{2}\phi_{k-1/2}^{(l-1)}(x)+\dfrac{1}{2}\left[\Sigma_{s,k}(x)+\boldsymbol{F}_{\boldsymbol{eff},\boldsymbol{k}}(x)\right]\phi_k^{(l-1)}(x) \\
\qquad\quad +Q_{eff,k+1}(x,\mu), \\
\phi_{k-1/2}^{(l-1/2)}(x)=\displaystyle\int_{-1}^{1}\psi_{k-1/2}^{(l-1/2)}(x,\mu)d\mu\,, \qquad \phi_k^{(l-1/2)}(x)=\displaystyle\int_{-1}^{1}\psi_k^{(l-1/2)}(x,\mu)d\mu\,, \\
\psi_k^{(l-1/2)}(0,\mu)=\psi_{k-1/2}^{(l-1/2)}(0,\mu)=\psi_L(\mu), \qquad \mu\in[0,1], \\
\psi_k^{(l-1/2)}(X,\mu)=\psi_k^{(l-1/2)}(X,-\mu), \qquad \psi_{k-1/2}^{(l-1/2)}(X,\mu)=\psi_{k-1/2}^{(l-1/2)}(X,-\mu),\ \ \mu\in[-1,0].
\end{cases}
\tag{4.82}
$$

We note that the MBTD-DSA iteration indexing, which considers the intermediate solutions, is already applied in Eq. (4.82).

Similar to that of BE-DSA, by subtracting Eq. (4.82) from its associated "exact" equation, and introducing iteration errors

$$
\begin{cases}
g(x,\mu)=\psi_{k-1/2}(x,\mu)-\psi_{k-1/2}^{(l-1/2)}(x,\mu), \qquad f(x,\mu)=\psi_k(x,\mu)-\psi_k^{(l-1/2)}(x,\mu), \\
G_n(x)=\displaystyle\int_{-1}^{1}\mu^n g(x,\mu)d\mu\,, \qquad F_n(x)=\displaystyle\int_{-1}^{1}\mu^n f(x,\mu)d\mu\,,
\end{cases}
\tag{4.83}
$$

we obtain the coupled error equations. Then, by taking the zeroth and the first angular moments of the error equations, applying the P1 approximation to both $f(x,\mu)$ and $g(x,\mu)$, and appropriately translating the neutron transport boundary conditions into the error diffusion equations, we obtain the coupled diffusion equations for the errors $F_0(x)$ and $G_0(x)$:

$$\begin{cases}\begin{aligned}&\left\{-\frac{d}{dx}\widetilde{D}(x)\frac{dG_0}{dx}+\left[\Sigma_{a,k-1/2}(x)-\boldsymbol{F}_{\boldsymbol{eff},k-1/2}(x)\right]G_0(x)\right\}\\&\qquad+\left\{\frac{d}{dx}D^*(x)\frac{dF_0}{dx}+\left[\frac{1}{v\Delta t_k}-\xi_{k\to k-1/2}\boldsymbol{F}_{\boldsymbol{d},\boldsymbol{k}}(x)\right]F_0(x)\right\}\\&=\left[\Sigma_{s,k-1/2}(x)+\boldsymbol{F}_{\boldsymbol{eff},\boldsymbol{k-1/2}}(x)\right]\left[\phi_{k-1/2}^{(l-1/2)}(x)-\phi_{k-1/2}^{(l-1)}(x)\right]\\&\qquad+\xi_{k\to k-1/2}\boldsymbol{F}_{\boldsymbol{d},\boldsymbol{k}}(x)\left[\phi_k^{(l-1/2)}(x)-\phi_k^{(l-1)}(x)\right],\end{aligned}\\[2ex]\begin{aligned}&\left\{-2\frac{d}{dx}D^*(x)\frac{dG_0}{dx}-\left[\frac{2}{v\Delta t_k}+\xi_{k-1/2\to k}\boldsymbol{F}_{\boldsymbol{d},\boldsymbol{k-1/2}}(x)\right]G_0(x)\right\}\\&\qquad+\left\{-\frac{d}{dx}\widehat{D}\frac{dF_0}{dx}+\left[\Sigma_{a,k}(x)-\boldsymbol{F}_{\boldsymbol{eff},\boldsymbol{k}}(x)+\frac{2}{v\Delta t_k}\right]F_0(x)\right\}\\&=\xi_{k-1/2\to k}\boldsymbol{F}_{\boldsymbol{d},\boldsymbol{k-1/2}}(x)\left[\phi_{k-1/2}^{(l-1/2)}(x)-\phi_{k-1/2}^{(l-1)}(x)\right]\\&\qquad+\left[\Sigma_{s,k}(x)+\boldsymbol{F}_{\boldsymbol{eff},\boldsymbol{k}}(x)\right]\left[\phi_k^{(l-1/2)}(x)-\phi_k^{(l-1)}(x)\right],\end{aligned}\end{cases}\tag{4.84}$$

with boundary conditions

$$\begin{cases}\frac{1}{4}F_0(0)+\frac{1}{2}F_1(0)=0, & \frac{1}{4}G_0(0)+\frac{1}{2}G_1(0)=0,\\[2ex] F_1(X)=0, & G_1(X)=0.\end{cases}\tag{4.85}$$

Here the error “currents” and the three diffusion equations are respectively defined as

$$\begin{cases}G_1(x)=-\widetilde{D}(x)\frac{dG_0}{dx}+D^*(x)\frac{dF_0}{dx},\\[2ex] F_1(x)=-2D^*(x)\frac{dG_0}{dx}-\widehat{D}(x)\frac{dF_0}{dx},\end{cases}\tag{4.86}$$

$$\begin{cases}D^*(x)=\dfrac{1}{\eta_k(x)+2+\dfrac{2}{\eta_{k-1/2}(x)}}\dfrac{1}{3\Sigma_t(x,t_{k-1/2})},\\[2ex] \widetilde{D}(x)=[\eta_k(x)+2]D^*(x), \qquad \widehat{D}(x)=\eta_{k-1/2}(x)D^*(x).\end{cases}\tag{4.87}$$

Equation (4.84) is similar to the MBTD coupled balance equations for neutron diffusion in Eq. (3.82); both of the equations are a four-block matrix problem. However, different from the MBTD coupled balance equations for neutron diffusion [Eq. (3.82)], where only its diagonal blocks are tridiagonal (corresponding to the typical steady-state diffusion operator), in the error equations of MBTD-DSA [Eq. (4.84)] all of the four blocks are tridiagonal. Nevertheless, we can still apply the box-scheme FDM discussed in Section 3.2 to numerically discretize and solve for the errors G_0 and F_0 in Eq. (4.84), since each of the block is essentially a steady state diffusion operator. It is worth mentioning that the diffusion operators of the diagonal blocks have left-vacuum and right-reflecting BCs, while the diffusion operators of the off-diagonal blocks have left- and right-reflecting BCs. This is true because at the boundaries, the D^* terms in the off-diagonal blocks of Eq. (4.84) collapse into the BCs Eq. (4.85) per Eq. (4.86).

Once we obtain G_0 and F_0, we can calculate the new iterate solutions:

$$\begin{cases} \phi_{k-1/2}^{(l)}(x) = \phi_{k-1/2}^{(l-1/2)}(x) + G_0(x), \\ \phi_k^{(l)}(x) = \phi_k^{(l-1/2)}(x) + F_0(x). \end{cases} \tag{4.88}$$

When convergence is achieved, we can calculate $\psi_k^{(l)}(x,\mu)$ per Eq. (4.76), but with

$$F_{1,j} = -2D_j^* \frac{G_{0,j+1/2} - G_{0,j-1/2}}{\Delta x_j} - \widehat{D}_j \frac{F_{0,j+1/2} - F_{0,j-1/2}}{\Delta x_j}, \tag{4.89}$$

to be used as the independent source in the next time-step calculation [note that we do not need to calculate and store $\psi_{k-1/2}^{(l)}(x,\mu)$].

Now, we summarize how MBTD-DSA works. First, we perform a normal MBTD-SI transport sweep to solve for the intermediate solutions $\phi_{k-1/2}^{(l-1/2)}$ and $\phi_k^{(l-1/2)}$ per Eq. (4.82), either by using the MBTD adaptation of SN-DD or MOC. Next, we solve the four-block diffusion Eq. (4.84) [with diffusion coefficients in Eq. (4.87) and BCs in Eq. (4.85)] by using the box-scheme FDM to obtain the errors G_0 and F_0. Finally, we calculate the new iterate solutions $\phi_{k-1/2}^{(l)}$ and $\phi_k^{(l)}$ per Eq. (4.88).

A Fourier analysis of MBTD-DSA can be similarly performed like that of MBTD-SI in Section 4.4 by using the following Fourier ansatz:

$$\begin{cases} G^{(l)}(x) = \theta^l B e^{i\omega\Sigma_t x}, & F^{(l)}(x) = \theta^l A e^{i\omega\Sigma_t x}, \\ g^{(l-1/2)}(x,\mu) = \theta^{l-1}\hat{b}(\mu) e^{i\omega\Sigma_t x}, & f^{(l-1/2)}(x,\mu) = \theta^{l-1}\hat{a}(\mu) e^{i\omega\Sigma_t x}, \\ G^{(l-1/2)}(x) = \theta^{l-1}\hat{B} e^{i\omega\Sigma_t x}, & F^{(l-1/2)}(x) = \theta^{l-1}\hat{A} e^{i\omega\Sigma_t x}, \\ G_0(x) = \theta^{l-1}\gamma_G e^{i\omega\Sigma_t x}, & F_0(x) = \theta^{l-1}\gamma_F e^{i\omega\Sigma_t x}. \end{cases} \tag{4.90}$$

Here the iteration solution errors $G^{(l)}$ and $F^{(l)}$, and the intermediate solution errors $g^{(l-1/2)}$, $f^{(l-1/2)}$, $G^{(l-1/2)}$, and $F^{(l-1/2)}$, are defined as

$$\begin{cases} G^{(l)}(x) = \phi_{k-1/2}(x) - \phi_{k-1/2}^{(l)}(x), \quad F^{(l)}(x) = \phi_k(x) - \phi_k^{(l)}(x), \\ g^{(l-1/2)}(x,\mu) = \psi_{k-1/2}(x,\mu) - \psi_{k-1/2}^{(l-1/2)}(x,\mu), \\ f^{(l-1/2)}(x,\mu) = \psi_k(x,\mu) - \psi_k^{(l-1/2)}(x,\mu), \\ G^{(l-1/2)}(x) = \phi_{k-1/2}(x) - \phi_{k-1/2}^{(l-1/2)}(x), \quad F^{(l-1/2)}(x) = \phi_k(x) - \phi_k^{(l-1/2)}(x), \end{cases} \tag{4.91}$$

and G_0 and F_0 are essentially estimates of the actual intermediate errors $G^{(l-1/2)}$ and $F^{(l-1/2)}$ per P_1 approximation Eq. (4.84). Details of the Fourier analysis of MBTD-DSA can be found in Appendix B.

The resulting eigenvalue and spectral radius of MBTD-DSA are respectively presented in Figure 4.3 and Figure 4.4. It is evident that MBTD-DSA effectively eliminates the most persisting error mode of MBTD-SI, which occurs as $\omega \rightarrow 0$. More importantly, it is shown that the spectral radius of MBTD-DSA is similar to that of BE-DSA. This means that both have similar convergence rates and require similar numbers of transport iterations to achieve convergence.

4.6 Numerical Results

Two numerical experiments are performed. First, in Section 4.6.1, we verify the theoretical convergence rates obtained from the Fourier Analysis of the four iterative methods: (1) BE-SI, (2) MBTD-SI, (3) BE-DSA, and (4) MBTD-DSA. Second, in Section 4.6.2, the same numerical experiment performed in Section 3.8.3 is considered to compare the relative efficiency of BE and MBTD with transport methods SN-DD and MOC.

4.6.1 Convergence Rate of SI and DSA

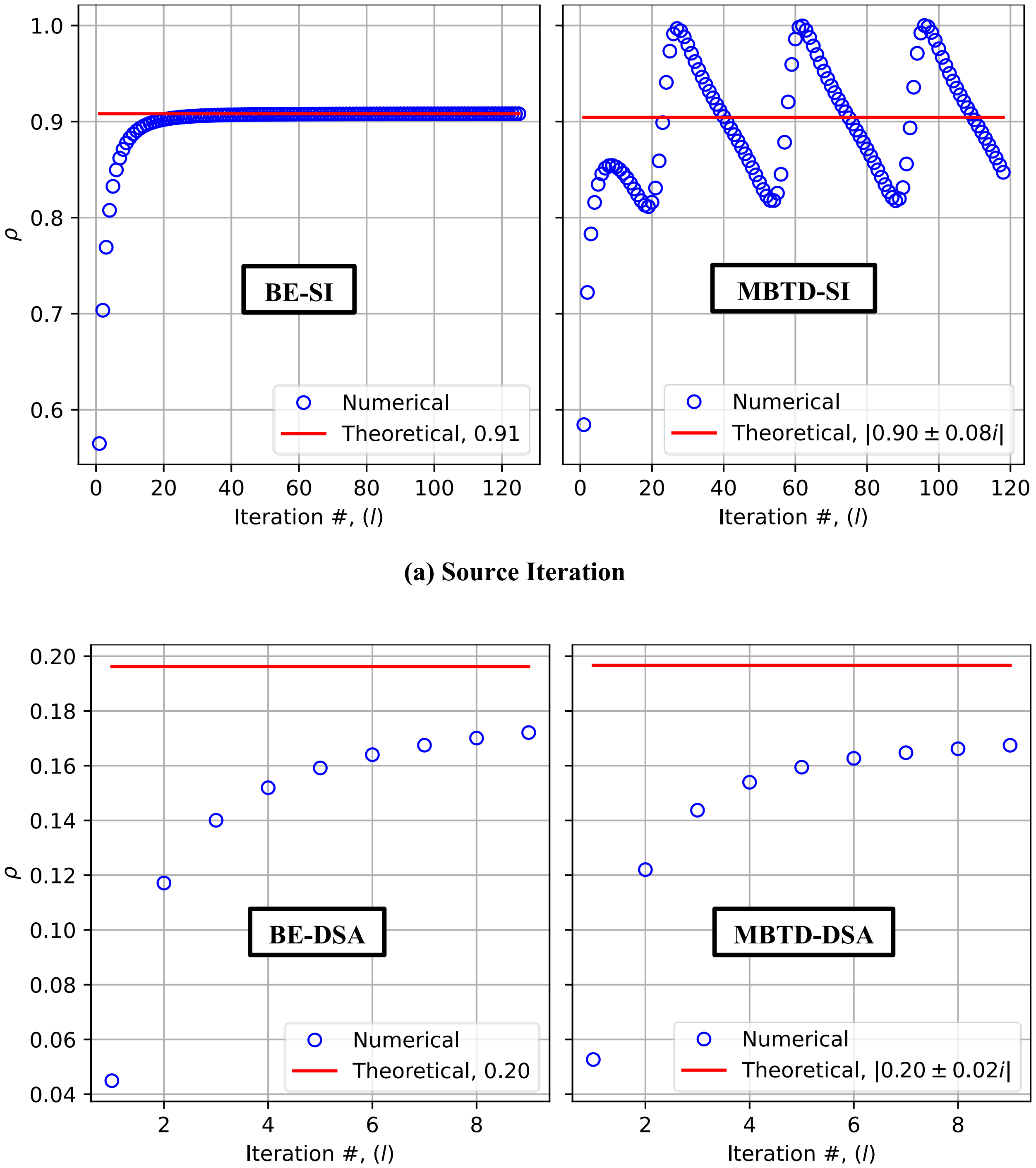
The spectral radius of an iterative method can be numerically estimated by taking the ratio of "residuals" of two consecutive iterations:

$$\rho_{\text{numerical}}^{(l)} = \frac{\left\|\phi^{(l+1)} - \phi^{(l)}\right\|_2}{\left\|\phi^{(l)} - \phi^{(l-1)}\right\|_2}. \tag{4.92}$$

As we keep iterating and l increases, $\rho_{\text{numerical}}^{(l)}$ converges to the actual spectral radius of the method. Therefore, in this experiment, we would like to verify whether the ratio of the residuals—or the numerical spectral radius $\rho_{\text{numerical}}^{(l)}$ defined in Eq. (4.92)—of the four iteration methods (BE-SI, MBTD-SI, BE-DSA, and MBTD-DSA) converge to their respective theoretical spectral radii obtained from the Fourier Analysis in Sections 4.4 and 4.5.

The kinetic problem KP1 described in Section 3.7 is simulated using the four iteration methods with a certain value of Δt (or η). Figure 4.5 shows the resulting numerical spectral radii $\rho_{\text{numerical}}^{(l)}$ for the four methods which are compared to their respective theoretical values per Eqs. (4.51), (4.64), and (4.81).

It is found that the numerical spectral radii of BE-SI and BE-DSA (top-left and bottom-left in Figure 4.5) converge to their respective theoretical values. However, this is not the case for the MBTD methods. The numerical spectral radius of MBTD-SI (top-right in Figure 4.5) does not converge to the expected theoretical value. Yet, it is observed that upon "convergence", the numerical spectral radius keeps changing periodically around the theoretical value. This anomaly is due to its complex conjugate leading eigenvalues, something that typically does not appear in standard iteration methods. In MBTD-DSA (bottom-right in Figure 4.5), however, such oscillations in the numerical spectral radius are not evident. This is because MBTD-DSA has a high convergence rate (low spectral radius ρ) so that convergence is achieved only in a few iterations and the iteration is terminated before oscillations in the numerical spectral radius start to appear. Further investigation is needed to better characterize the significance of the complex conjugate leading eigenvalues in MBTD-SI and MBTD-DSA.



(a) Source Iteration

(b) Diffusion Synthetic Acceleration

Figure 4.5: Numerical spectral radius for problem KP1 with $\eta = 10.0$

Strictly speaking, such oscillating numerical spectral radius or ratio of the residuals, which is evident in MBTD methods, seems problematic. However, the methods still, *in average,* converge with the same rate as the theoretical convergence rates. The oscillations of the "instantaneous" convergence rate are just an inherent feature of MBTD methods.

It is worth adding that the resulting numerical spectral radii converge to slightly below the theoretical values. This is because the numerical experiments simulate a very large—yet finite—system, while the theoretical spectral radius is based on the Fourier Analysis of an infinite medium system. The faster rate of convergence observed for the finite system is likely due to leakage, which is not accounted for in the infinite-medium Fourier analysis.

4.6.2 Efficiency

The same numerical experiment performed in Section 3.8.3 is considered, except that now we use the time-dependent neutron transport methods discussed in this chapter for simulating the large 1D-slab models of the four test problems shown in Figure 3.2. We would like to compare the relative efficiency of BE and MBTD for the following combinations of transport methods: (1) DSA with SN-DD and (2) DSA with MOC. The pure, unaccelerated, SI methods are not considered because they require extremely small Δt (relative to the simulation time interval $2t_s$) to obtain a reasonably low spectral radius ρ and thus a reasonable convergence rate.

All of the BE methods use the linear delayed fission production approximation (Section 3.6.1), while all of the MBTD methods use the standard MBTD delayed source formulation (Section 3.6.2). It is worthwhile to emphasize that we adapt the Simultaneous Solve strategy for solving the MBTD coupled balance equations. The number of directional quadrature N is set to 8, and the spatial-mesh size Δx is taken to be one tenth a mean free path. The convergence criterion for the transport iterations is set as follows:

$$\max\frac{\left|\phi^{(l+1)}-\phi^{(l)}\right|}{\left|\phi^{(l)}\right|}\leq\left(1-\rho_{\text{numerical}}^{(l)}\right)\epsilon_{tol}, \tag{4.93}$$

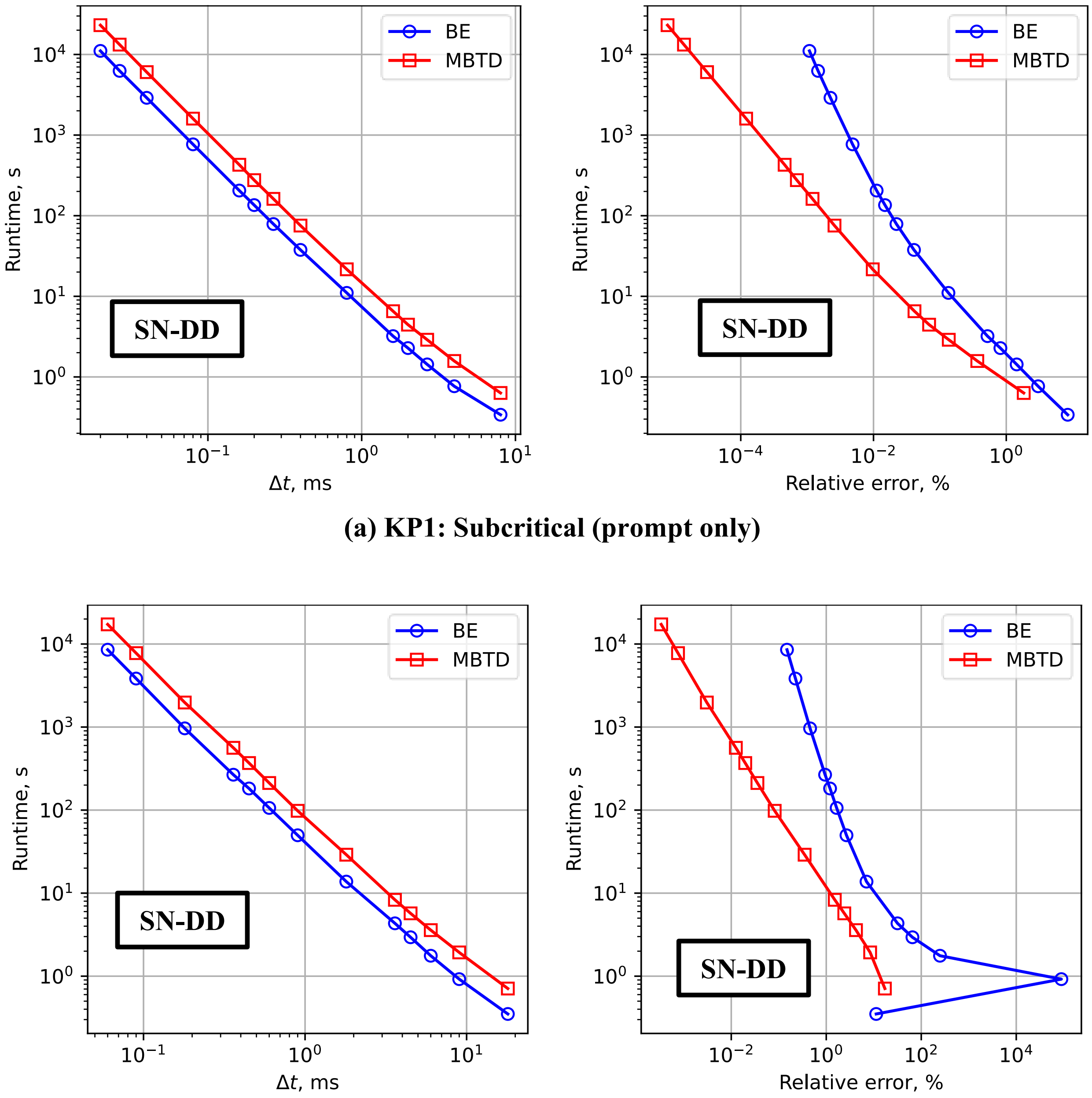
where the desired error tolerance $\epsilon_{tol}=10^{-5}$ is multiplied by one minus the current iterate spectral radius estimate $\rho_{numerical}^{(l)}$ defined in Eq. (4.92). Finally, SciPy's sparse ILU-preconditioned GMRES linear solver is used to solve the diffusion problems of the error corrections F_0 and G_0 in the DSA methods [Eqs. (4.73) and (4.84)] with a relative error tolerance of 10^{-7}.

The simulations are performed with different time-step sizes Δt, and the resulting relative errors and the simulation runtimes are shown in Figure 4.6 and Figure 4.7 for DSA with SN-DD and MOC, respectively. We note that the figures are identically set up as the ones for neutron diffusion methods in Figure 3.8 (we refer to Section 3.8.3 for a discussion on the significance of the axes of the plots). It is found that MBTD-DSA with SN-DD and MOC are respectively about 2.5 and 3 times computationally more expensive than those of BE-DSA given the same value of Δt, but the second-order accuracy of the MBTD methods make them more efficient than BE methods for a wide range of achieved accuracies. Nevertheless, the efficiency margins are not as large as the one observed in the time-dependent neutron diffusion problems in Chapter 3, where MBTD is only two times computationally more expensive than BE. In other words, the benefit of using MBTD over BE is the largest in neutron diffusion simulations, followed by neutron transport simulations with SN-DD, and then with MOC.

It is worth mentioning that there is a trend anomaly in the first two relative error data of BE in simulating KP4 [figure part (b) in Figure 4.6 and figure part (d) in Figure 4.7]. This is because BE suffers from spurious oscillations for an exponentially growing system if Δt is not sufficiently small (as shown in Figure 3.6).

As briefly mentioned in Section 4.3, we obtain a considerable speedup for MBTD with MOC by storing the values of the non-trivial function evaluations, which require memory of size $(J \times N)$. The speedup is demonstrated in figure part (a) of Figure 4.7. It is found that MBTD with MOC and the additional storage is about as efficient as MBTD with SN-DD.

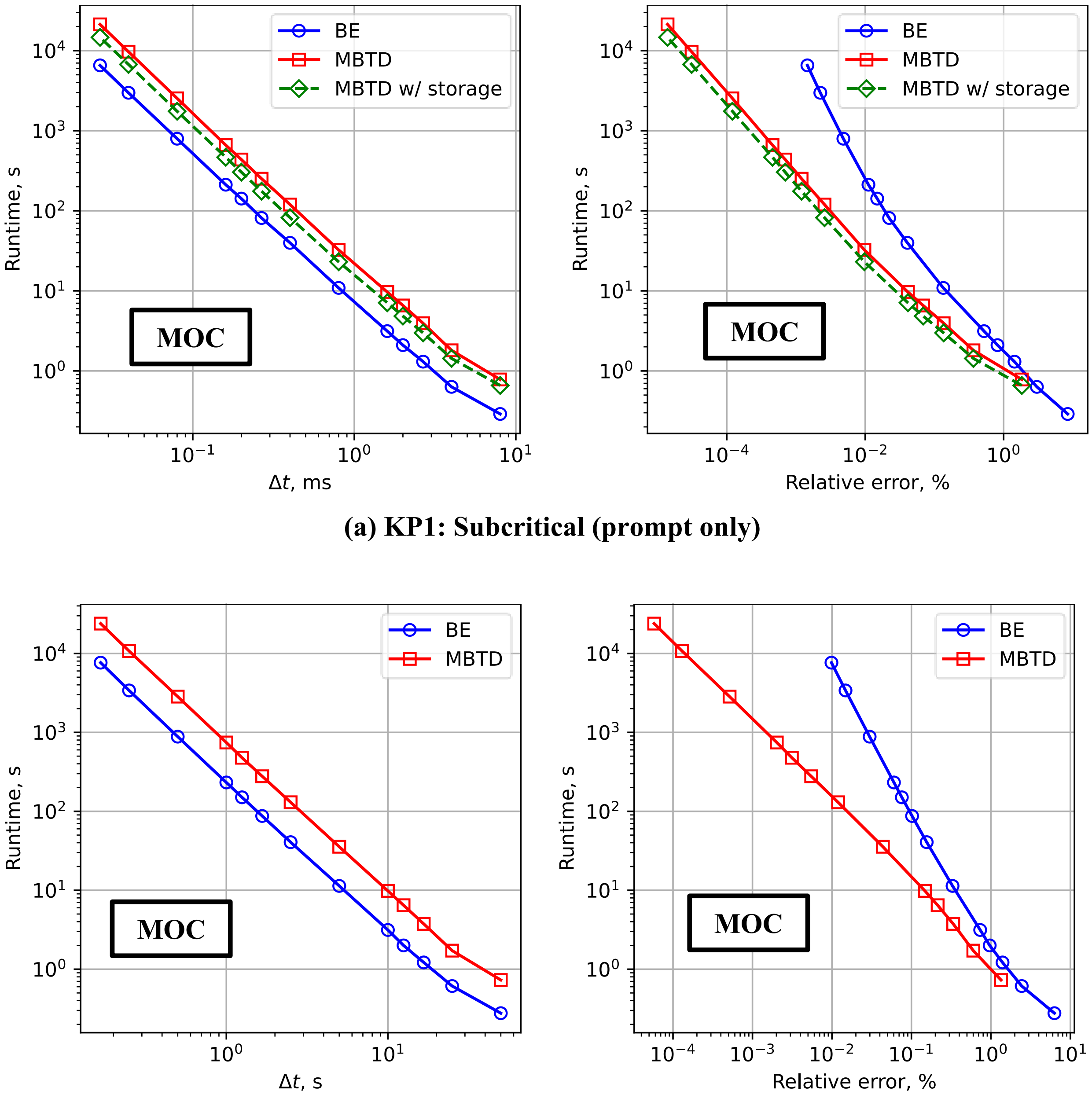
It is found that MBTD (either -DSA or -SI) with SN-DD does not converge in the stiff problems KP2 and KP3, except if the time step is sufficiently small; the ratio of the residuals oscillates around unity as we keep iterating. This reminds us of the stability issue encountered in solving the stiff problems KP2 and KP3 with CN in Section 3.8.1. There seems to be a connection with the issue that we have now since DD is essentially CN applied to spatial discretization. However, this issue does not occur for BE (either -DSA or -SI) with SN-DD. This indicates that the combination of MBTD and DD yields transport iteration methods that conditionally converge. A Fourier analysis of the discretized systems should reveal and characterize this conditional (for MBTD with SN-DD) an unconditional (for BE with SN-DD) convergence.



(a) KP1: Subcritical (prompt only)

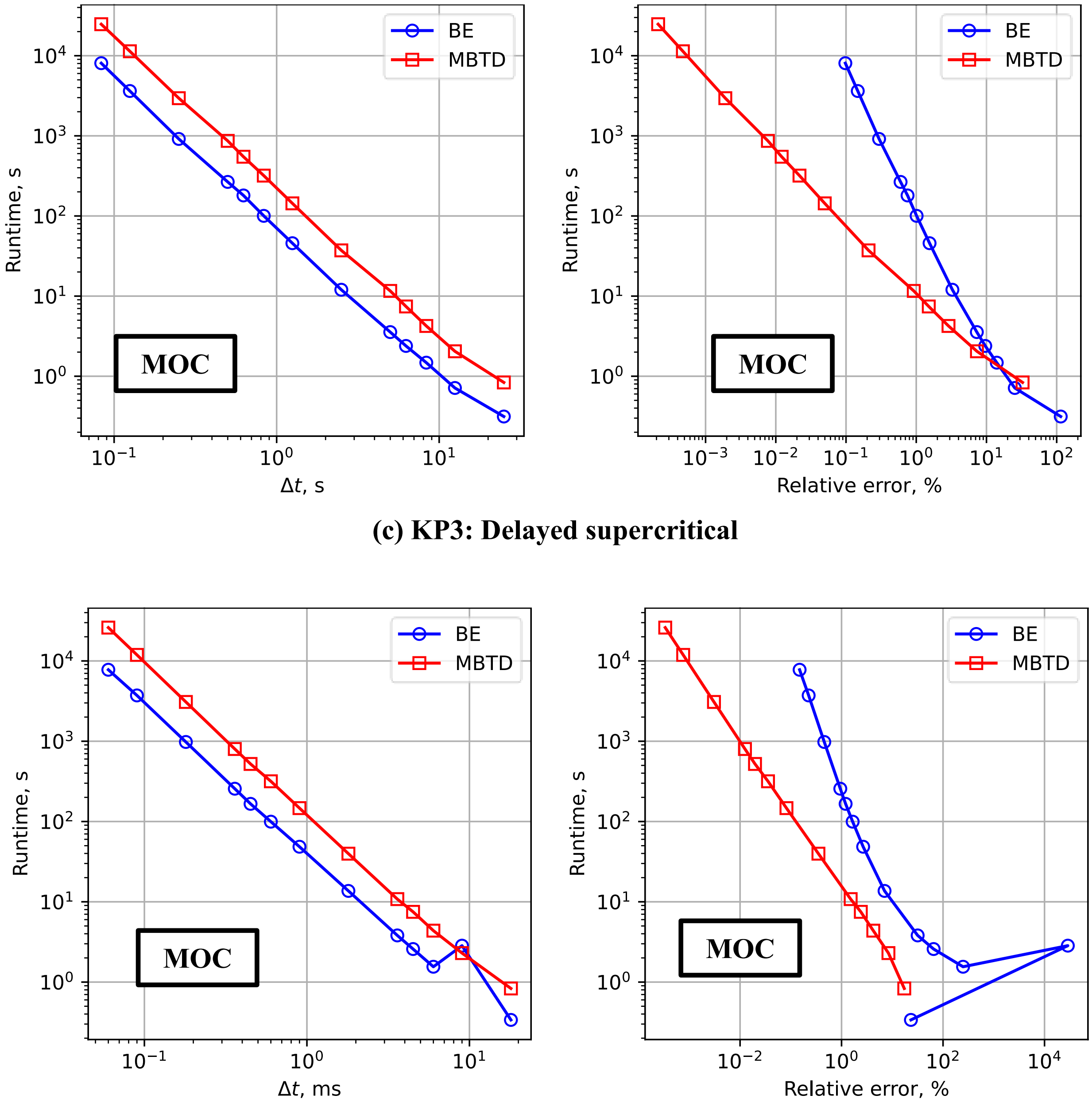
(b) KP4: Prompt supercritical

Figure 4.6: Runtime and efficiency of BE and MBTD with SN-DD



(a) KP1: Subcritical (prompt only)

(b) KP2: Subcritical



(c) KP3: Delayed supercritical

(d) KP4: Prompt supercritical

Figure 4.7: Runtime and efficiency of BE and MBTD with MOC

4.7 Summary

In this chapter, we investigated the application of MBTD to time-dependent neutron transport problems. The MBTD formulations for Source Iteration with SN-DD and MOC neutron transport methods are derived in Section 4.3. Distinct from with BE-SI or standard SI for steady-state problems, MBTD-SI (with Simultaneous Solve strategy) performs two transport sweeps simultaneously (as shown in Figure 4.2) instead of just one (Figure 4.1): (1) for SN-DD, this means MBTD-SI needs to solve a tridiagonal 4×4 matrix instead of 2×2 at each transport sweep step, (2) while for MOC, this means MBTD-SI needs to "analytically" solve a system of two (instead of one) heterogeneous first-order linear ODEs, which has complex conjugate eigenvalues requiring evaluation of not only an exponential function but also trigonometry functions. Another important observation is that the memory complexity of MBTD transport methods is similar to that of BE methods. MBTD methods need to store both ϕ_k and $\phi_{k+1/2}$, while BE methods only need to store ϕ_k. However, while both BE and MBTD methods need to store the memory-dominating $\psi_{k,n,j}$, MBTD does not need to store $\psi_{k+1/2,n,j}$.

A Fourier analysis is performed in Section 4.4 to characterize the convergence rate of BE-SI and MBTD-SI, and the results are shown in Figure 4.3 and Figure 4.4. It is found that both methods have similar spectral radii and thus convergence rates, regardless of η. Furthermore, the most persisting Fourier error modes for both methods is the flat mode, which is associated with a nearly isotropic angular dependence indicating a strong incentive of diffusion acceleration.

Diffusion Synthetic Acceleration for MBTD is derived in Section 4.5. Unlike BE-DSA or the standard DSA for steady-state problems, the error correction diffusion problem of MBTD-DSA is a four-block (instead of one) of typical diffusion tridiagonal matrices. A Fourier analysis is performed for BE-DSA and MBTD-DSA, and it is found that both methods have a similar spectral radius and thus convergence rate, regardless of η, as shown in Figure 4.4.

Figure 4.5 verifies the theoretical convergence rates of the SI and DSA methods obtained by the Fourier analysis in the previous sections. Furthermore, it is found that MBTD methods "suffer" from an oscillating ratio of residuals upon convergence because of their complex conjugate leading eigenvalues. Figure 4.6 and Figure 4.7 respectively present the relative efficiency of DSA methods with SN-DD and MOC for the four kinetic test problems. It is found that MBTD-DSA with SN-DD and MOC is respectively about 2.5 and 3 times computationally more expensive than BE-DSA given the same value of Δt, but the second-order accuracy of the MBTD method makes it a more efficient method compared to BE for reasonably accurate simulations. A caution, however, needs to be given in simulating stiff problems (like KP2 and KP3) using MBTD with SN-DD, due to its conditional transport iteration convergence.

CHAPTER 5

MBTD Application to a Multi-Physics Problem

This chapter presents the first attempt to apply the robust second-order MBTD method to a non-linear multi-physics problem. In Section 5.1, a multi-physics tight-coupling method for MBTD is derived. A simple multi-physics problem of coupled neutronics/thermal-hydraulics, which is typically encountered in reactor simulations, is developed and numerically solved in Section 5.2.

5.1 Multi-Physics Tight-Coupling for MBTD

Let us consider a generic time-dependent 2-physics problem:

$$\frac{\partial u_i}{\partial t} = f_i[t, u_1(t), u_2(t)]\,, \qquad i = 1, 2\,, \tag{5.1}$$

where $f_i[t, u_1(t), u_2(t)]$ is the *single-physics operator* of physics i. By directly applying BE to this multi-physics problem, we obtain for each i

$$\frac{u_i(t_k) - u_i(t_{k-1})}{\Delta t_k} = f_i[t_k, u_1(t_k), u_2(t_k)]. \tag{5.2}$$

By following the MBTD formulation for non-linear problems discussed in Chapter 2, we obtain for each i

$$\begin{cases} \dfrac{u_i(t_k) - u_i(t_{k-1})}{\Delta t_k} = f_i\left[t_{k-1/2}, u_1\left(t_{k-1/2}\right), u_2\left(t_{k-1/2}\right)\right], \\ \dfrac{u_i(t_k) - u_i\left(t_{k-1/2}\right)}{\Delta t_k/2} = f_i[t_k, u_1(t_k), u_2(t_k)]. \end{cases} \tag{5.3}$$

Equation (5.2) can be solved iteratively as follows:

$$\begin{cases} \dfrac{u_1^{(l)}(t_k) - u_1(t_{k-1})}{\Delta t_k} = f_1\left[t_k, u_1^{(l)}(t_k), u_2^{(l-1)}(t_k)\right], \\ \dfrac{u_2^{(l)}(t_k) - u_2(t_{k-1})}{\Delta t_k} = f_2\left[t_k, u_1^{(l)}(t_k), u_2^{(l)}(t_k)\right], \end{cases} \qquad (5.4)$$

where one physics is solved at a time by lagging the other physical variable or using the updated solution if available. The iteration method of Eq. (5.4) is also known as the multi-physics *tight-coupling* or *Piccard Iteration*. Furthermore, a *staggered* (Gauss-Seidel-style) approach is applied in Eq. (5.4), where new solutions are used once they are available. A multi-physics MBTD tight-coupling method for this problem would be:

$$\begin{cases} \begin{cases} \dfrac{u_1^{(l)}(t_k) - u_1(t_{k-1})}{\Delta t_k} = f_1\left[t_{k-1/2}, u_1^{(l)}(t_{k-1/2}), u_2^{(l-1)}(t_{k-1/2})\right], \\ \dfrac{u_1^{(l)}(t_k) - u_1^{(l)}(t_{k-1/2})}{\Delta t_k/2} = f_1\left[t_k, u_1^{(l)}(t_k), u_2^{(l-1)}(t_k)\right]. \end{cases} \\ \begin{cases} \dfrac{u_2^{(l)}(t_k) - u_2(t_{k-1})}{\Delta t_k} = f_2\left[t_{k-1/2}, u_1^{(l)}(t_{k-1/2}), u_2^{(l)}(t_{k-1/2})\right], \\ \dfrac{u_2^{(l)}(t_k) - u_2^{(l)}(t_{k-1/2})}{\Delta t_k/2} = f_2\left[t_k, u_1^{(l)}(t_k), u_2^{(l)}(t_k)\right]. \end{cases} \end{cases} \qquad (5.5)$$

5.2 Multi-physics Test Problem and Results

We adopt the simple 1D multi-physics test problem proposed in [49]. This problem was designed to capture the behavior of a k-eigenvalue neutronics/thermal-hydraulics system calculation of typical circular fuel pin cells inscribed within square water coolant channels by a one-dimensional model. The parameters and variables are treated as constant values at a given axial height. Furthermore, mono-energetic neutron diffusion was assumed. This steady-state multi-physics test problem is then modified into a transient problem.

A time-dependent neutron diffusion equation without delayed neutrons is considered:

$$\frac{1}{v}\frac{\partial \phi}{\partial t} - \frac{\partial}{\partial x}D(T)\frac{\partial \phi}{\partial x} + \Sigma_a(T)\phi(x,t) = \xi \nu \Sigma_f(T)\phi(x,t), \qquad t > 0, \qquad (5.6)$$

with vacuum boundary conditions and a spatial domain $x \in [0, X]$ spanning the fuel channel length $X = 360$ cm. The diffusion coefficient is defined as $D(T) = 1/3\Sigma_t(T)$, the neutron speed v and fission multiplication ν are respectively set to be 2.2×10^5 cm/s and 2.43, and the parameter ξ is a variable to alter the criticality of the system. The neutron flux distribution $\phi(x,t)$ is proportional to the linear heat generation rate in the fuel: $q(x,t) = \kappa_f \Sigma_f(T)\phi(x,t)$. The neutron cross-sections are functions of the fuel and coolant (water) temperatures $T(x,t) = [T_f(x,t) \quad T_w(x,t)]$, which are modeled by linear interpolations based on the data showed in Table 5.1:

$$\Sigma_*(T_f, T_w) = C_x + C_{x,f} T_f + C_{x,w} T_w, \qquad \Sigma_* = \Sigma_t, \Sigma_s, \nu\Sigma_f. \tag{5.7}$$

Table 5.1: Cross-sections at reference temperatures [49]

T_f [K]	T_w [K]	$\Sigma_t(T)$ [/cm]	$\Sigma_s(T)$ [/cm]	$\nu\Sigma_f(T)$[/cm]
565	565	0.655302	0.632765	0.0283063
1565	565	0.653976	0.631252	0.0277754
565	605	0.61046	0.589171	0.0265561

The time-dependent heat equations for the fuel and the coolant are

$$\rho_f A_f c_{p,f}(T_f) \frac{\partial T_f}{\partial t}(x,t) + 2\pi R_f h\left[T_f(x,t) - T_w(x,t)\right] = q(x,t), \tag{5.8}$$

$$\rho_w(T_w) A_w c_{p,w}(T_w) \frac{\partial T_w}{\partial t}(x,t) + \dot{m} c_{p,w}(T_w) \frac{\partial T_w}{\partial x}(x,t) = 2\pi R_f h\left[T_f(x,t) - T_w(x,t)\right], \tag{5.9}$$

with a boundary (inlet temperature) condition of $T_w(0,t) = T_{w,in} = 565$ K. The cross-sectional areas of the fuel and coolant are respectively $A_f = \pi R_f^2$ and $A_w = (d^2 - A_f)$, with $R_f = 0.5$ cm and $d = 1.3$ cm. The mass flow rate $\dot{m}$, heat transfer coefficient h, and energy deposited per fission κ_f are respectively set to be 0.3 kg/s, 0.2 W/cm^2-K, and 191.4 MeV. The Python package `thermo` [50] is used to generate the density ρ and specific heat capacity c_p of the fuel (UO_2) and coolant (water at 15.5 MPa) as a function of the respective temperature.

The steady-state solutions—$\phi_{SS}(x)$, $T_{f,SS}(x)$, and $T_{w,SS}(x)$ shown in Figure 5.1—at a linear power rate of $P = 200$ W/cm is used as the initial condition. This is obtained by solving the steady-state k-eigenvalue multi-physics problem of Eqs. (5.6)–(5.9), with the neutron flux normalized to the power rate P, and searching for the value of ξ that gives $k \approx 1$ (or ξ_{SS}). The neutron diffusion Eq. (5.6) and the water coolant heat transfer Eq. (5.9) are spatially discretized using the box-scheme FDM (Section 3.2) and the *upwind* method [6], respectively. With the *Relaxed* Picard Iteration method (similar to the one studied in [49]), relaxation parameter $\omega = 0.5$, $J = 1800$ equally sized spatial meshes, multi-physics coupling relative error tolerance $\epsilon_r = 10^{-5}$ (for ϕ, T_f, and T_w), and SciPy's sparse ILU-preconditioned GMRES and non-linear solver MINPACK [47] (for respectively solving the linear problem in the k-eigenvalue neutron diffusion and the non-linear heat transfer problems, both with a relative error tolerance of $\epsilon_r \times 10^{-2}$), we found $\xi_{SS} = 0.8097$, which corresponds to $k = 1.00001$.

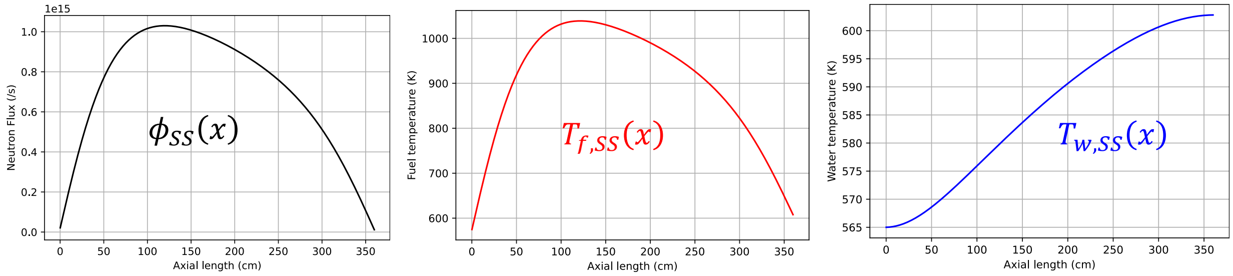


Figure 5.1: Initial conditions of the multi-physics problem

In the test problem, the transient is induced by instantaneously dropping the inlet coolant temperature $T_{w,in}$ from 565 to 515 K at $t = 0$. The transient is simulated up to 1 s. This temperature perturbation gradually inserts positive reactivity (due to T_w negative feedback) from $x = 0$ to X, which is later counteracted by a negative reactivity insertion due to the fuel temperature rise (T_f negative feedback). The reactivity change in time is shown in the secondary y-axis of Figure 5.2. This time-dependent multi-physics problem is solved with the staggered multi-physics tight-coupling methods of BE and MBTD [Eqs. (5.4) and (5.5), respectively] with $K \in [20, 1000]$ uniform time steps. A staggered multi-physics tight-coupling CN solution with $K = 2000$ is used as the reference solution. The simulation results are presented in Figure 5.2 and Figure 5.3.

Figure 5.2 shows the average linear power of the reference, MBTD, and BE solutions with several different values of K. The first and the second arguments in the parentheses next to the plot label in the legend respectively denote K and the time required to generate the solution. It is found that with $K = 100$, MBTD manages to produce a highly accurate result (red broken line with square marker). On the other hand, with the same $K = 100$, the BE solution (purple broken line with upside-down triangle marker) is far from accurate. If we refine the time step to

$K = 600$, the solution of BE (blue broken line with circle marker) improves, but it is still considerably less accurate than that of MBTD with $K = 100$, not to mention that the runtime already exceeds the runtime of the MBTD. If we further refine the time step to $K = 1000$, the solution of BE (broken green line with triangle marker) slightly improves, but it is still less accurate than the MBTD solution with $K = 100$, while the runtime is more than twice longer.

Figure part (a) of Figure 5.3 shows that the runtime of MBTD to generate solutions is about three times longer than that of BE for the same Δt. Three error metrics are used for comparing the efficiency of MBTD and BE: (1) the total power $\int P(t)dt$, (2) the peak power, and (3) the time of peak power. These are respectively presented in figure (b), (c), and (d) of Figure 5.3. It is found that MBTD is generally more efficient than BE for reasonably accurate simulations (relative error smaller than ~10%).

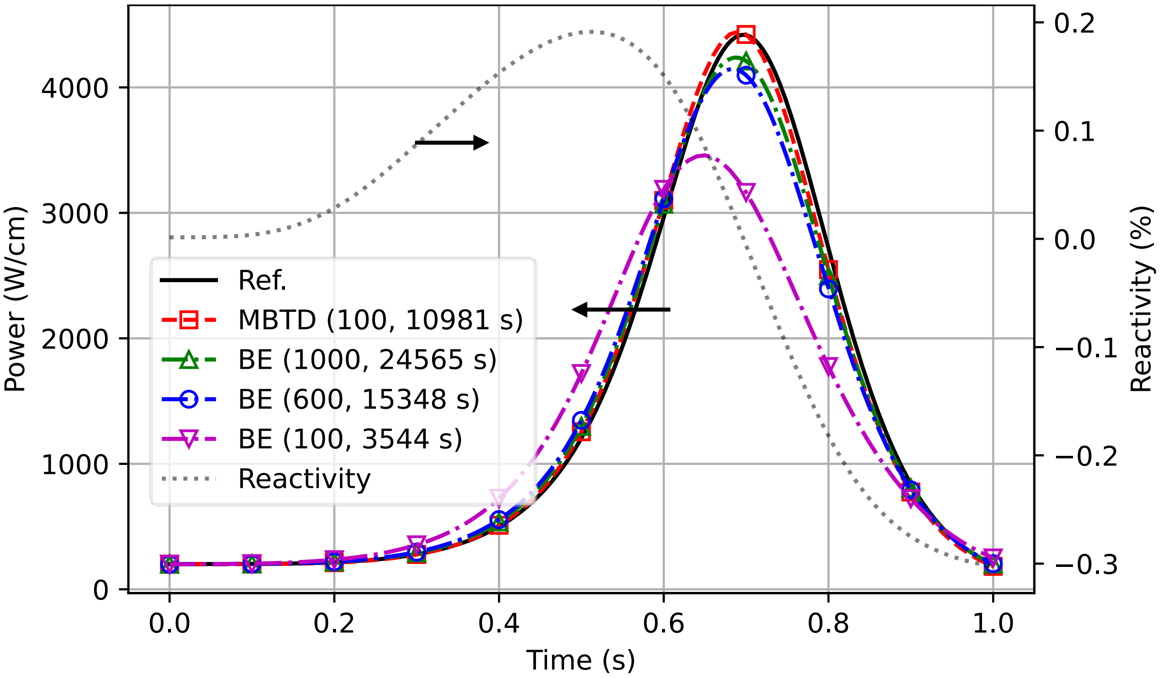
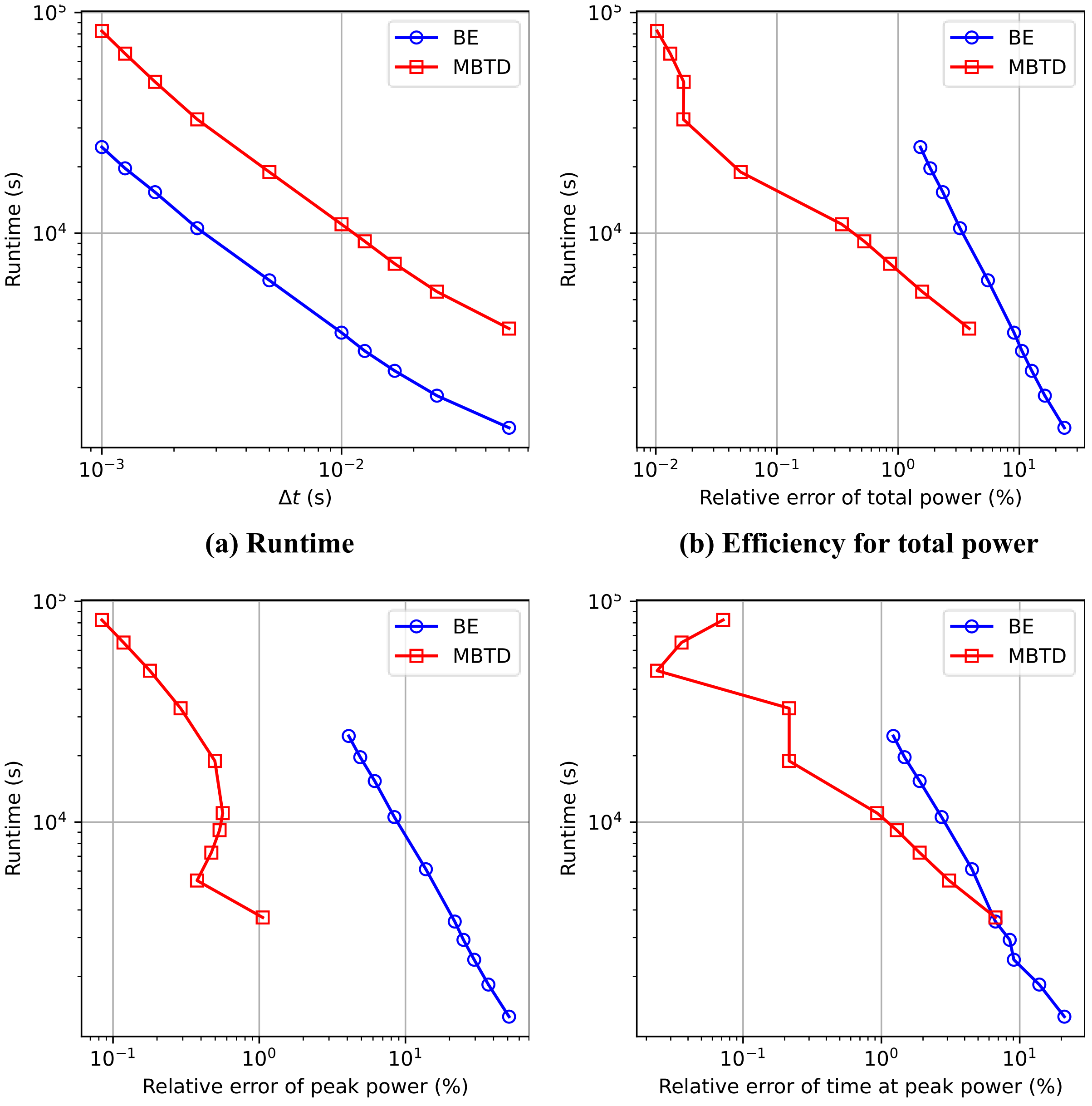


Figure 5.2: Simulation results of the multi-physics problem



(a) Runtime

(b) Efficiency for total power

(c) Efficiency for peak power

(d) Efficiency for time at peak power

Figure 5.3: Runtime and efficiency of BE and MBTD for the multi-physics problem

CHAPTER 6

α-Eigenvalue Neutron Transport Iteration Method

At this point, we shift gear into the second part of this thesis, to investigate the effectiveness of the α-weighted multigroup constants ($\boldsymbol{\alpha}$-MGXS). In this chapter, we discuss several methods for solving the α-eigenvalue neutron transport equation. In Section 6.1, the k- and α-eigenvalue neutron transport problems are presented, followed by a review of their respective iteration methods in Section 6.2. A simple infinite medium kinetic problem is considered; however, it serves as general guidance for applications in more practical problems. Implementations of a selected α-eigenvalue iteration method to the open-source Monte Carlo code OpenMC [51] are discussed in Section 6.3. These implementations are then tested against several benchmark problems in Section 6.4 for verification.

6.1 Eigenvalue Neutron Transport Equations

An infinite medium neutron transport problem with autonomous (time-independent) cross-sections is considered:

$$\begin{cases} \dfrac{1}{v}\dfrac{\partial\phi}{\partial t} + \Sigma_t(E)\phi(E,t) = \displaystyle\int_0^\infty \nu_s\Sigma_s(E'\to E)\phi(E',t)dE' \\ + \displaystyle\int_0^\infty \chi_p(E'\to E)\nu_p\Sigma_f(E')\phi(E',t)dE' + \sum_{i=1}^{I}\sum_{j=1}^{J}\chi_{d,i,j}(E)\lambda_{i,j}C_{i,j}(t)\,, \qquad t>0, \\ \qquad\qquad \dfrac{dC_{i,j}}{dt} + \lambda_{i,j}C_{i,j}(t) = \displaystyle\int_0^\infty \nu_{d,i,j}\Sigma_{f,i}(E')\phi(E',t)dE'\,, \end{cases} \tag{6.1}$$

with initial conditions $\phi(E,0)=\phi_{\text{init}}(E)$ and $C_{i,j}(0)=C_{\text{init},i,j}$, where $i=1,2,\dots,I$ and $j=$

$1, 2, \dots, J$ respectively indicate isotope and delayed neutron precursor group index. Scattering multiplicity ν_s and isotope-dependent delayed precursor group data [$\lambda_{i,j}$, $\nu_{d,i,j}(E)$, $\chi_{d,i,j}(E)$] are considered. Furthermore, it is worth to acknowledge the isotopic precursor group concentration $C_{i,j}$, especially if $\lambda_{i,j} \neq \lambda_{i'\neq i,j}$ which indicates that $C_{i,j}$ and $C_{i'\neq i,j}$ are essentially different species.

The k-eigenvalue neutron transport equation of Eq. (6.1) is the following:

$$\begin{aligned}\Sigma_t(E)\phi_k(E) &= \int_0^\infty \nu_s \Sigma_s(E' \to E)\phi_k(E')dE' \\ &+ \frac{1}{k}\int_0^\infty \sum_i^I \left[\chi_{p,i}(E' \to E)\nu_{p,i}(E') + \sum_{j=1}^J \chi_{d,i,j}(E)\nu_{d,i,j}(E')\right]\Sigma_{f,i}(E')\,\phi_k(E')dE',\end{aligned} \tag{6.2}$$

where $\phi_k(E)$ is the eigenfunction corresponding to the eigenvalue k. In k-eigenvalue calculation, we are typically interested in obtaining the *fundamental mode* which corresponds to the largest eigenvalue k. The k-eigenvalue Eq. (6.2) essentially forces a system that is originally unsteady [or time-dependent per Eq. (6.1)] into becoming steady, by scaling the fission neutron production with the factor k. Therefore, Eqs. (6.1) and (6.2) are essentially two different physical problems, unless if $k = 1$.

Meanwhile, the corresponding α-eigenvalue neutron transport equation of Eq. (6.1) is as follows:

$$\left\{\begin{aligned} &\left[\Sigma_t(E) + \frac{\alpha}{v}\right]\phi_\alpha(E) = \int_0^\infty \nu_s\Sigma_s(E' \to E)\phi_\alpha(E')dE' \\ &+ \int_0^\infty \chi_p(E' \to E)\nu_p\Sigma_f(E')\phi_\alpha(E')dE' + \sum_{i=1}^I\sum_{j=1}^J \chi_{d,i,j}(E)\lambda_{i,j}C_{\alpha,i,j}, \\ &\alpha C_{\alpha,i,j} + \lambda_{i,j}C_{\alpha,i,j} = \int_0^\infty \nu_{d,i,j}\Sigma_{f,i}(E')\phi_\alpha(E')dE', \end{aligned}\right. \tag{6.3}$$

which is derived by assuming solutions in the form of

$$\phi(E,t) = \phi_\alpha(E)e^{\alpha t}, \qquad C_{i,j}(t) = C_{\alpha,i,j}e^{\alpha t}, \tag{6.4}$$

where $\phi_\alpha(E)$ and $C_{\alpha,i,j}$ are the corresponding eigenfunctions of the eigenvalue α. We note that

Eq. (6.3) is essentially exact because it is derived by directly substituting Eq. (6.4) into Eq. (6.1), without introducing any approximation. It is customary to substitute the precursor group equation in Eq. (6.3) into the neutron equation, making possible a direct comparison with the k-eigenvalue Eq. (6.2):

$$\left[\Sigma_t(E)+\frac{\alpha}{v}\right]\phi_\alpha(E)=\int_0^\infty \nu_s\Sigma_s(E'\to E)\phi_\alpha(E')dE' \\ +\int_0^\infty\sum_{i=1}^{I}\left\{\chi_{p,i}(E'\to E)\nu_{p,i}(E')+\sum_{j=1}^{J}\chi_{d,i,j}(E)\left[\frac{\lambda_{i,j}}{\alpha+\lambda_{i,j}}\right]\nu_{d,i,j}(E')\right\}\Sigma_{f,i}(E')\,\phi_\alpha(E')dE'\,. \tag{6.5}$$

In an α-eigenvalue calculation, we are typically interested in obtaining the *fundamental mode*, which corresponds to the fundamental, or the algebraically largest (the right-most), eigenvalue. If the fundamental eigenvalue is very large, such that $\alpha \gg \lambda_{i,j}$ (e.g., in a prompt supercritical system), the simpler *prompt* α-eigenvalue equation is preferred instead:

$$\left[\Sigma_t(E)+\frac{\alpha}{v}\right]\phi_\alpha(E)=\int_0^\infty \nu_s\Sigma_s(E'\to E)\phi_\alpha(E')dE' \\ +\int_0^\infty \chi_p(E'\to E)\nu_p\Sigma_f(E')\phi_\alpha(E')dE'\,. \tag{6.6}$$

In a delayed supercritical or subcritical system, however, the fundamental mode corresponds to the slowest delayed mode, which is comparable to the smallest $|\lambda_{i,j}|$. This means Eq. (6.6) is not appropriate for calculating the fundamental mode of subcritical and delayed supercritical systems. Nevertheless, if we are looking for the *prompt mode of a deeply subcritical system*, which corresponds to the most negative or the left-most eigenvalue such that $|\alpha| \ll \lambda_{i,j}$, it is justified to use the prompt α-eigenvalue Eq. (6.6).

6.2 α-Eigenvalue Iteration Methods

Let us first consider iteration methods for solving the k-eigenvalue Eq. (6.2). Equation (6.2) can be represented in the following operator terms:

$$\boldsymbol{T}\phi_k(E)=\boldsymbol{S}\phi_k(E)+\frac{1}{k}\boldsymbol{F}\phi_k(E), \tag{6.7}$$

where

$$\begin{cases} \boldsymbol{T}\phi_k(E) = \Sigma_t(E)\phi_k(E), \qquad \boldsymbol{S}\phi_k(E) = \int_0^\infty \nu_s\Sigma_s(E' \to E)\phi_k(E')dE', \\ \boldsymbol{F}\phi_k(E) = \int_0^\infty \sum_i^I \left[\chi_{p,i}(E' \to E)\nu_{p,i}(E') + \sum_{j=1}^J \chi_{d,i,j}(E)\nu_{d,i,j}(E')\right]\Sigma_{f,i}(E')\,\phi_k(E')dE'. \end{cases} \tag{6.8}$$

Two k-eigenvalue iteration methods are considered—(1) power iteration (hereafter is referred to as k-Iteration), and (2) source iteration (hereafter is referred to as k-SI)—which respectively solve Eq. (6.7) as the following:

$$(\boldsymbol{T}-\boldsymbol{S})\phi_k^{(l)}(E) = \frac{1}{k^{(l-1)}}\boldsymbol{F}\phi_k^{(l-1)}(E), \tag{6.9}$$

$$\boldsymbol{T}\phi_k^{(l)}(E) = \left(\boldsymbol{S} + \frac{1}{k^{(l-1)}}\boldsymbol{F}\right)\phi_k^{(l-1)}(E). \tag{6.10}$$

Both methods update the eigenvalue k as follows:

$$k^{(l)} = \frac{\langle \boldsymbol{F}\phi_k^{(l)}(E)\rangle}{\langle(\boldsymbol{T}-\boldsymbol{S})\phi_k^{(l)}(E)\rangle}, \tag{6.11}$$

where $\langle\cdot\rangle = \int_0^\infty (\cdot)dE$. Most, if not all, Monte Carlo transport codes adapt the k-Iteration method [Eq. (6.9)]. On the other hand, deterministic transport codes typically adapt the k-SI method [Eq. (6.10)].

Now, let us discuss the α-eigenvalue iteration methods. Equations (6.5) can be represented in the following operator terms:

$$\left(\boldsymbol{T} + \frac{\alpha}{v}\right)\phi_\alpha(E) = \left[\boldsymbol{S} + \sum_{i=1}^I \left(\boldsymbol{F}_{\boldsymbol{p,i}} + \sum_{j=1}^J \frac{\lambda_{i,j}}{\alpha+\lambda_{i,j}}\boldsymbol{F}_{\boldsymbol{d,i,j}}\right)\right]\phi_\alpha(E), \tag{6.12}$$

or

$$\left(\boldsymbol{T}+\frac{\alpha}{v}\right)\phi_\alpha(E)=(\boldsymbol{S}+\boldsymbol{F_\alpha})\phi_\alpha(E), \tag{6.13}$$

where

$$\begin{cases} \boldsymbol{F_{p,i}}\phi_\alpha(E)=\int_0^\infty \chi_{p,i}(E'\to E)\nu_{p,i}(E')\Sigma_{f,i}(E')\phi_\alpha(E')dE', \\ \boldsymbol{F_{d,i,j}}\phi_\alpha(E)=\int_0^\infty \chi_{d,i,j}(E)\nu_{d,i,j}(E')\Sigma_{f,i}(E')\phi_\alpha(E')dE', \\ \boldsymbol{F}=\sum_{i=1}^{I}\left[\boldsymbol{F_{p,i}}+\sum_{j=1}^{J}\boldsymbol{F_{d,i,j}}\right], \qquad \boldsymbol{F_\alpha}=\sum_{i=1}^{I}\left(\boldsymbol{F_{p,i}}+\sum_{j=1}^{J}\frac{\lambda_{i,j}}{\alpha+\lambda_{i,j}}\boldsymbol{F_{d,i,j}}\right). \end{cases} \tag{6.14}$$

We define $\boldsymbol{F_\alpha}$ as the *time-corrected fission production operator*. Similarly, the prompt α-eigenvalue problem Eq. (6.6) becomes

$$\left(\boldsymbol{T}+\frac{\alpha}{v}\right)\phi_\alpha(E)=(\boldsymbol{S}+\boldsymbol{F_p})\phi_\alpha(E). \tag{6.15}$$

There are several known methods for solving the α-eigenvalue neutron transport equation. Some are iterative, and some are not. Five α-eigenvalue iteration methods are considered: (1) Traditional α-k Iteration [31][32], (2) Relaxed α-k Iteration [35][37], (3) α-Iteration-F [33], (4) α-Iteration-T [36], and (5) α-Iteration-F&T [38].

6.2.1 Traditional α-k Iteration

The first of the kind is the Traditional α-k Iteration, made popular by Hill [31] and Brockway et al. [32] and designed to solve the prompt α-eigenvalue Eq. (6.15). In the Traditional α-k Iteration, a parameter k is introduced to the α-eigenvalue as the following:

$$\left(\boldsymbol{T}+\frac{\alpha^{(l-1)}}{v}\right)\phi_\alpha^{(l)}(E)=\left(\boldsymbol{S}+\frac{1}{k^{(l)}}\boldsymbol{F_p}\right)\phi_\alpha^{(l)}(E). \tag{6.16}$$

Given a fixed value of $\alpha^{(l-1)}$, we essentially have a k-eigenvalue method with an additional time absorption $\alpha^{(l-1)}/v$. This k-eigenvalue method is then solved (either with k-Iteration or k-SI) to obtain $\phi_\alpha^{(l)}$ and $k^{(l)}$. After that, the α eigenvalue can be updated as the following:

$$\alpha^{(l)} = \frac{\langle (\boldsymbol{S} - \boldsymbol{T} + \boldsymbol{F_p})\phi_\alpha^{(l)}(E)\rangle}{\langle \frac{1}{v}\phi_\alpha^{(l)}(E)\rangle}. \tag{6.17}$$

As we keep iterating, the parameter k converges to unity, while eigenvalue α converges to the desired prompt α eigenvalue. This method is considerably more expensive than k-eigenvalue methods, since a k-eigenvalue problem is solved in each iteration.

6.2.2 Relaxed α-k Iteration

Zoia et al. relax [35] the α-k Iteration method by allowing an α eigenvalue estimate update at each k-eigenvalue iterate. The Relaxed α-k Iteration for solving the prompt α-eigenvalue problem is the following:

$$\left(\boldsymbol{T} + \frac{\alpha^{(l-1)}}{v} - \boldsymbol{S}\right)\phi_\alpha^{(l)}(E) = \frac{1}{k^{(l-1)}}\boldsymbol{F_p}\phi_\alpha^{(l-1)}(E), \tag{6.18}$$

where the k “eigenvalue” is updated as follows:

$$k^{(l)} = \frac{\langle \boldsymbol{F_p}\phi_\alpha^{(l)}(E)\rangle}{\langle \left(\boldsymbol{T} + \frac{\alpha^{(l-1)}}{v} - \boldsymbol{S}\right)\phi_\alpha^{(l)}(E)\rangle}. \tag{6.19}$$

Hereafter, any mention of α-k Iteration refers to Relaxed α-k Iteration.

A simple linear update procedure is used in [35] for updating the prompt α eigenvalue. Later, Kia et al. [39] introduce an improved prompt α eigenvalue update procedure for the Relaxed α-k Iteration; yet, it is basically equivalent to Eq. (6.17).

Later, Zoia et al. generalize [37] the α-k Iteration by considering the contribution of the delayed fission neutron production, or in other words adapting the method for solving the actual α-eigenvalue problem Eq. (6.13) with delayed neutrons:

$$\left(\boldsymbol{T} + \frac{\alpha^{(l-1)}}{v} - \boldsymbol{S}\right)\phi_\alpha^{(l)}(E) = \frac{1}{k^{(l-1)}}\boldsymbol{F}_{\boldsymbol{\alpha}^{(l-1)}}\phi_\alpha^{(l-1)}(E), \tag{6.20}$$

where

$$\boldsymbol{F}_{\boldsymbol{\alpha}^{(l-1)}} = \sum_{i=1}^{I}\left(\boldsymbol{F}_{\boldsymbol{p},\boldsymbol{i}} + \sum_{j=1}^{J}\frac{\lambda_{i,j}}{\alpha^{(l-1)} + \lambda_{i,j}}\boldsymbol{F}_{\boldsymbol{d},\boldsymbol{i},\boldsymbol{j}}\right), \tag{6.22}$$

and the k and α eigenvalues are respectively updated as follows:

$$k^{(l)} = \frac{\langle \boldsymbol{F}_{\boldsymbol{\alpha}^{(l-1)}}\phi_\alpha^{(l)}(E)\rangle}{\langle\left(\boldsymbol{T} + \frac{\alpha^{(l-1)}}{v} - \boldsymbol{S}\right)\phi_\alpha^{(l)}(E)\rangle}, \tag{6.24}$$

$$\alpha^{(l)} = \frac{\langle(\boldsymbol{S} - \boldsymbol{T} + \boldsymbol{F}_{\boldsymbol{\alpha}^{(l)}})\phi_\alpha^{(l)}(E)\rangle}{\langle\frac{1}{v}\phi_\alpha^{(l)}(E)\rangle}, \tag{6.25}$$

or per Eq. (6.22):

$$\alpha^{(l)} = \frac{\langle(\boldsymbol{S} - \boldsymbol{T} + \boldsymbol{F}_{\boldsymbol{p}})\phi_\alpha^{(l)}(E)\rangle + \sum_{i=1}^{I}\sum_{j=1}^{J}\left[\frac{\lambda_{i,j}}{\alpha^{(l)} + \lambda_{i,j}}\langle\boldsymbol{F}_{\boldsymbol{d},\boldsymbol{i},\boldsymbol{j}}\phi_\alpha^{(l)}(E)\rangle\right]}{\langle\frac{1}{v}\phi_\alpha^{(l)}(E)\rangle}. \tag{6.26}$$

Equation (6.26) is a non-linear problem with $(I \times J) + 1$ eigenvalue $\alpha^{(l+1)}$ solutions. Section 6.3.3 will discuss how the eigenvalue α is updated in a Monte Carlo simulation per Eq. (6.26).

The other three iterative methods—introduced by Yamamoto et al. [33], Shim et al. [36], and Josey et al. [38]—are categorized as α-Iteration methods, in which the α-eigenvalue problem is iteratively solved without introducing the artificial k eigenvalue. These methods—hereafter referred to as α-Iteration-F [33], α-Iteration-T [36], and α-Iteration-F&T [38]—differ in which terms in the equation are lagged in their respective iteration schemes. These are discussed in the next subsections. The prompt α-eigenvalue will be used in the discussion; nevertheless, generalizations to include the delayed neutrons can be similarly made like the one in [37] or in this subsection.

6.2.3 α-Iteration-F

The α-Iteration-F introduced by Yamamoto et al. [33] lags the fission production term as follows:

$$\left(\boldsymbol{T} + \frac{\alpha^{(l-1)}}{v} - \boldsymbol{S}\right)\phi_\alpha^{(l)}(E) = \boldsymbol{F_p}\phi_\alpha^{(l-1)}(E), \tag{6.27}$$

where the α eigenvalue is equivalently updated per Eq. (6.17).

By closer observation, it is evident that α-Iteration-F [Eq. (6.27)] is similar to the Relaxed α-k Iteration [Eq. (6.18)]. The only difference between the methods is the artificial constant k introduced in the α-k Iteration. However, this artificial constant k is not essential in the overall iteration scheme of α-k Iteration; it only acts as a normalization factor. Therefore, Relaxed α-k Iteration is equivalent to α-Iteration-F, except that Relaxed α-k Iteration calculates an estimate of k at each iteration. Nevertheless, since k converges to unity, having an estimate of k at each iteration is useful as it provides additional information for convergence monitoring.

It has been well observed [52] that instability may occur in simulating a very subcritical system with the α-k Iteration (or α-Iteration-F). This instability is attributed to the large negative value of α, leading to an overall negative value of the augmented total cross-section during the iteration: $\boldsymbol{T} + \alpha^{(l-1)}/v < 0$. This instability issue can be eliminated by lagging the time absorption (or production) term, as adapted in the next two methods.

6.2.4 α-Iteration-T

The α-Iteration-T introduced by Shim et al. [36] lag the time absorption/production term:

$$(\boldsymbol{T} - \boldsymbol{S} - \boldsymbol{F_p})\phi_\alpha^{(l)}(E) = -\frac{\alpha^{(l-1)}}{v}\phi_\alpha^{(l-1)}(E). \tag{6.28}$$

This method is essentially a power iteration of the α-eigenvalue problem. It is evident that, for *finite medium* transport problems, α-Iteration-T requires storing the angular flux from the previous iteration. Furthermore, in a supercritical system, α-Iteration-T needs to solve a supercritical fixed source problem, which has no physical solution, at each of its iterations. This makes α-Iteration-T problematic for simulating a supercritical system.

6.2.5 *α-Iteration-F&T*

The α-Iteration-F&T introduced by Josey et al. [38] lag the fission production and the time absorption/production terms:

$$(\boldsymbol{T}-\boldsymbol{S})\phi_\alpha^{(l)}(E)=\left(\boldsymbol{F_p}-\frac{\alpha^{(l-1)}}{v}\right)\phi_\alpha^{(l-1)}(E). \qquad (6.29)$$

Similar to α-Iteration-T, α-Iteration-F&T was invented to avoid the potential instability in simulating a deeply subcritical system. However, different from α-Iteration-T, α-Iteration-F&T works for a supercritical system because it also lags the fission production term.

The α-k Iteration and the three α-Iteration methods are often referred to as α-static iteration methods. These methods are well known for their Monte Carlo implementations; however, they are essentially iteration schemes applicable for the deterministic method as well. It is worth to emphasize that specific implementations or techniques—such as the time absorption (or delta production) reaction [32][24] and the particle weight correction [33]—do not necessarily introduce new iteration methods.

Other notable methods for solving the α-eigenvalue problem include the Transition Rate Matrix Method (TRMM) [30] and the Domain Mode Decomposition (DMD) [22]. TRMM discretizes the system and directly solves the discretized α-eigenvalue problem. DMD calculates the α-modes corresponding to an approximate transport operator constructed via domain mode decomposition of generated time-dependent neutron transport solutions. As described earlier in this section, in most applications of the α-eigenvalue neutron transport equation, obtaining the prompt or the fundamental α-mode is of main interest. In such applications, the iterative α-static methods are more favourable than TRMM and DMD: TRMM needs to finely discretize the system and calculates all of the corresponding α-modes just to accurately obtain the desired prompt/fundamental α-mode, while DMD needs to accurately solve the time-dependent neutron transport equation for a sufficiently long time so that the prompt/fundamental α-mode is contained in the approximate transport operator.

6.3 Monte Carlo Implementation and OpenMC

The generalized Relaxed α-k Iteration described in Eq. (6.20) is implemented into the open-source Monte Carlo code OpenMC [51] by slightly modifying the existing k-eigenvalue simulation routine. The implementation is set up to converge to the fundamental α-mode, which corresponds to the algebraically largest eigenvalue. As mentioned in the previous section, the Relaxed α-k Iteration may suffer from instability upon simulating the prompt α-mode of a deeply subcritical system because of the large negative eigenvalue $\alpha \ll -\max(\lambda_{i,j})$. However, this issue does not arise when we are solving for the fundamental α-mode which corresponds to $\alpha > -\min(\lambda_{i,j})$. Following the discussion of the previous sections, the equations presented in this section are those of infinite medium problems. Yet, the implementation in OpenMC applies to general neutron transport problems.

By comparing the Relaxed α-k Iteration Eq. (6.20) and the k-Iteration method Eq. (6.9), which is the long-established standard Monte Carlo method for solving k-eigenvalue problems, it is found that besides the usual streaming and reaction terms, there are two α-eigenvalue-specific terms that must be simulated as well: (1) the time absorption/source, and (2) the time-corrected delayed neutron emission.

6.3.1 *Time Absorption/Source*

Time absorption/source is pertinent to the time cross-section term α/v in Eq. (6.20). This term can be physically treated as a time absorption reaction ($\alpha > 0$) or a time delta source production ($\alpha < 0$) that competes with the actual neutron reactions. This is by far the most used technique in treating the time absorption/source term [32][24][35][36][38].

Another approach is to apply the particle weight correction technique [33][25], in which anytime a particle moves a distance d, its weight is corrected as

$$w_{\text{new}} = w_{\text{old}} \exp\left(-\frac{\alpha}{v} d\right), \tag{6.30}$$

$$\bar{w} = \frac{1}{d}\int_0^d w_{\text{old}} \exp\left(-\frac{\alpha}{v} s\right) ds\,. \tag{6.31}$$

The corrected weight w_{new} is the neutron weight at the end of the track, while the average weight along the track $\bar{w}$ is used for any track-length estimator tally. This approach is physically meaningful: during a Monte Carlo simulation, neutrons advance in time when they move from one point to another, and their potential contribution (weight) to the change of the neutron population continuously changes during the travel.

The particle weight correction approach is implemented to OpenMC. With this implementation, care must be taken in tally-scoring with a mesh filter since the particle weight continuously changes along the particle flight track.

6.3.2 Time-corrected Delayed Neutron Emission

This is pertinent to the fission production term $\nu_{d,i,j}\Sigma_{f,i}$ in Eq. (6.20). The delayed neutron production of each precursor group, $\nu_{d,i,j}$, is corrected with a factor $\lambda_{i,j}/\left(\alpha+\lambda_{i,j}\right)$ [37][38][25], and the time-corrected total fission production $\nu_{\alpha,i}$ replaces the one normally used in standard k-eigenvalue simulation as follows:

$$\nu_i(E)=\nu_{p,i}(E)+\sum_{j=1}^{J}\nu_{d,i,j}(E) \quad \Rightarrow \quad \nu_{\alpha,i}(E)=\nu_{p,i}(E)+\sum_{j=1}^{J}\frac{\lambda_{i,j}}{\alpha+\lambda_{i,j}}\nu_{d,i,j}(E)\,. \qquad (6.32)$$

This correction is performed during fission neutron sampling and k eigenvalue tally scoring. The correction is physically meaningful: if $|\alpha|\gg\lambda_{i,j}$, then prompt fission neutrons are mostly sampled, suppressing the average contribution of delayed neutrons. However, if $|\alpha|$ is comparable to $\lambda_{i,j}$, then the delayed neutrons' average contribution and sampling probabilities are appropriately scaled up or down. In particular, when searching for the fundamental eigenvalue of a very subcritical system, which is slightly greater than $-\min\left(\lambda_{i,j}\right)$, the probability of sampling delayed neutrons from the slowest precursor group is scaled up to a large extent. This significantly shifts the fission neutron spectrum to the delayed neutron spectrum $\chi_{d,i,j}(E)$.

It is important to note that rescaling should be performed during tally-scoring of fission production with an analog estimator. In addition to rescaling due to the division by k, delayed neutron emission rescaling should also be performed, so that the proper fission production is scored. This is typically the case when tallying fission production with a post-collision energy filter. The following prompt and delayed fission neutron correction factor should be multiplied to the original analog estimator score:

$$f_p = k, \qquad f_{d,i,j} = k\frac{\alpha + \lambda_{i,j}}{\lambda_{i,j}}. \tag{6.33}$$

6.3.3 Relaxed *α-k Iteration as Monte Carlo Simulation*

In a Monte Carlo implementation, the Relaxed α-k Iteration [Eqs. (6.20)–(6.26)] is translated into the following:

$$\begin{aligned}\left[\frac{\alpha^{(l-1)}}{v} + \Sigma_t(E)\right]\phi_\alpha^{(l)}(E) &= \int_0^\infty \nu_s\Sigma_s(E' \to E)\phi_\alpha^{(l)}(E')dE' \\ &+ \frac{1}{k^{(l-2)}}\int_0^\infty \sum_{i=1}^{I}\left[\begin{array}{c}\chi_{p,i}(E' \to E)\nu_{p,i}(E') \\ +\displaystyle\sum_{j=1}^{J}\chi_{d,i,j}(E)\frac{\lambda_{i,j}}{\alpha^{(l-2)} + \lambda_{i,j}}\nu_{d,i,j}(E')\end{array}\right]\Sigma_{f,i}(E')\,\phi_\alpha^{(l-1)}(E')dE',\end{aligned} \tag{6.34}$$

$$k^{(l)} = \left[\int_0^\infty \sum_{i=1}^{I}\left[\nu_{p,i}(E) + \sum_{j=1}^{J}\frac{\lambda_{i,j}}{\alpha^{(l-1)} + \lambda_{i,j}}\nu_{d,i,j}(E)\right]\Sigma_{f,i}(E)\,\phi_\alpha^{(l)}(E)dE\right], \tag{6.35}$$

$$\begin{aligned}\left(\alpha^{(l)} - \alpha^{(l-1)}\right)\left[\int_0^\infty \frac{1}{v}\phi_\alpha^{(l)}(E)dE\right] &= \left[\int_0^\infty \nu_p(E)\Sigma_f(E)\phi_\alpha^{(l)}(E)dE\right] + \\ &+ \sum_{i=1}^{I}\sum_{j=1}^{J}\frac{\lambda_{i,j}}{\alpha^{(l)} + \lambda_{i,j}}\left[\int_0^\infty \nu_{d,i,j}(E)\Sigma_{f,i}(E)\phi_\alpha^{(l)}(E)dE\right] - 1,\end{aligned} \tag{6.36}$$

$$\frac{1}{k^{(l-1)}}\int_0^\infty \sum_{i=1}^{I}\left[\nu_{p,i}(E) + \sum_{j=1}^{J}\frac{\lambda_{i,j}}{\alpha^{(l-1)} + \lambda_{i,j}}\nu_{d,i,j}(E)\right]\Sigma_{f,i}(E)\,\phi_\alpha^{(l)}(E)dE = 1. \tag{6.37}$$

Equation (6.34) describes how the α-k Monte Carlo simulation works. The lagged fission term indicates that sampled fission neutrons are stored for the next iteration/generation fixed-source, which in turn is normalized, as shown in Eq. (6.37). The standard k eigenvalue update procedure is shown in Eq. (6.35). The α eigenvalue update procedure shown in Eq. (6.36) is derived by integrating Eq. (6.34) over the whole phase space using the new eigenfunction estimate; then, the absorption and leakage integrals are replaced by using the normalization condition and the previous eigenvalue iterate definition. Equations (6.34), (6.35), and (6.37) are readily implemented in OpenMC once the time correction mechanisms previously discussed are enabled.

Equation (6.36) makes a non-linear problem with $(I \times J) + 1$ eigenvalue $\alpha^{(l)}$ solutions, where $\alpha_1^{(l)} > \alpha_2^{(l)} > \cdots > \alpha_{(I \times J)+1}^{(l)}$ and the fundamental eigenvalue satisfies $\alpha_1^{(l)} > -\min(\lambda_{i,j})$. This is where one can effectively control which α-mode to simulate. In this thesis work, Newton-Raphson (NR) algorithm is used to obtain the fundamental mode $\alpha_1^{(l)}$. However, if $\alpha^{(l)} < -\min(\lambda_{i,j})$ is obtained during an NR iteration, which indicates that NR tries to jump over the fundamental root $\alpha_1^{(l)}$, then the iterate solution is changed into $\alpha^{(l)} = 0.5\left[\alpha^{(l-1)} - \min(\lambda_{i,j})\right]$, giving a flavour of the bisection algorithm to the update strategy for subcritical systems.

The implementations presented here are introduced on top of OpenMC repository commit `8fcd74b14bba3b553ac8fc904100e787a69e87ed`. Concerning the user interface, to activate the time correction capabilities and run the static fundamental α-eigenvalue simulation, the Python API attribute `Settings.alpha_mode` is set to `True` while running in `eigenvalue` mode. In addition, when running in `multi-group` energy mode, the multigroup speeds must be listed into Python API attribute `Settings.mg_speeds`.

6.4 Verification Results

In all continuous energy calculations (including those in Chapter 7), OpenMC Official Data Library ENDF/B-VII.1 is used, along with the provided thermal scattering $S(\alpha,\beta)$ data for ^{1}H in H_2O, at temperature 294 K. Two sets of static problems are set up to verify the static fundamental α-eigenvalue method implementation in OpenMC. The significance of the time-corrected spectrum is also investigated by comparing the calculated fundamental k- and α-eigenfunctions.

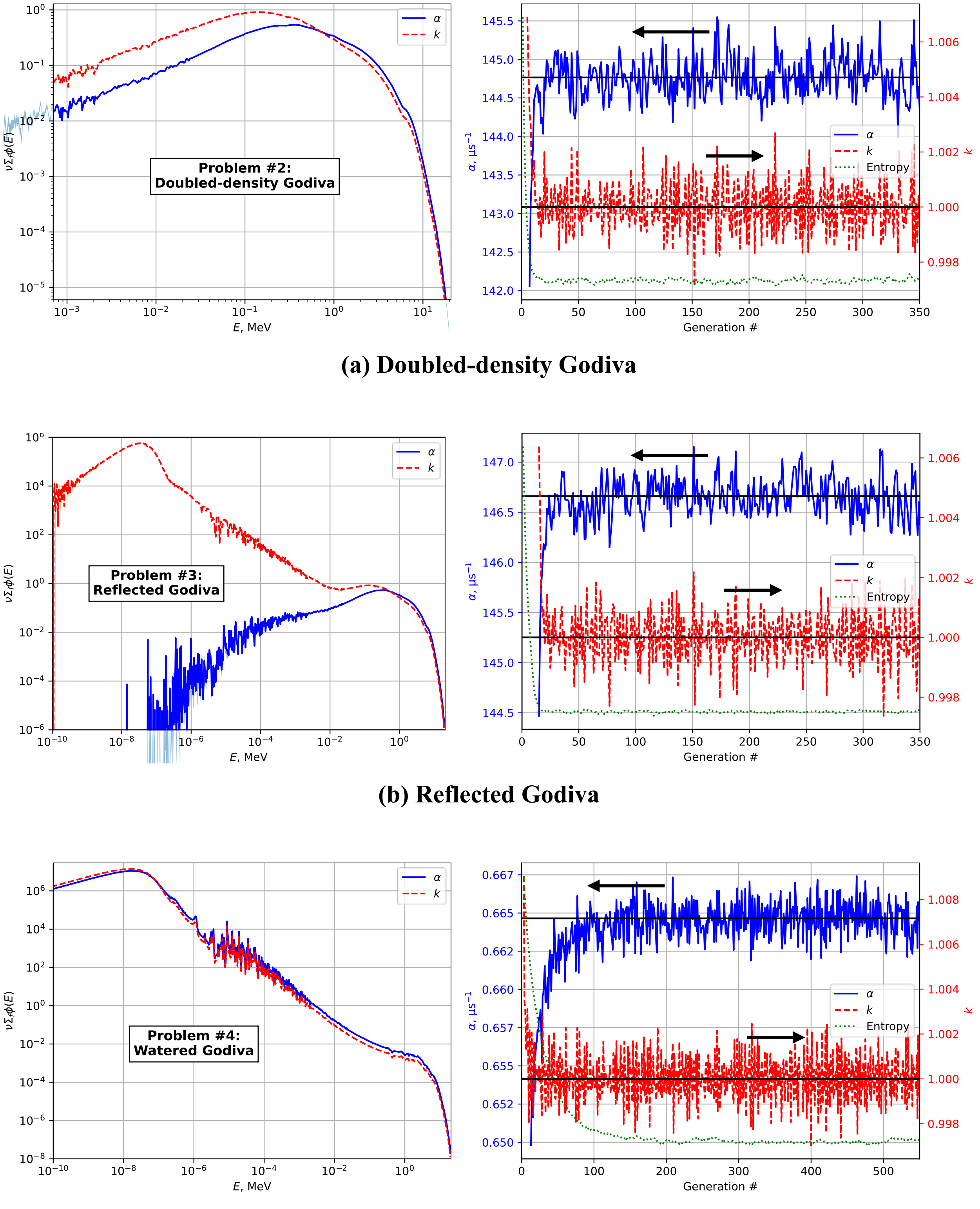
6.4.1 Cullen's Godiva Problems

Cullen's Godiva problems [24] have been used as a benchmark for static prompt α-eigenvalue Monte Carlo calculation. This problem set consists of the original and modified versions of the Godiva homogeneous spherical Highly Enriched Uranium (HEU) reactor, which primarily tests the time absorption/source capability described in Section 6.3.1. The prompt supercritical problems (Problem #2, #3, and #4) are considered in this work. Problems #1 (slightly subcritical) and #5 (very subcritical) are not considered because the referred benchmark solutions calculate the prompt instead of the fundamental mode of the subcritical systems. Problem #2 (bare fast system) consists of the bare HEU (0.937695 at.% ^{235}U, 0.052053 at.% ^{238}U, 0.010252 at.% ^{234}U) sphere with a density of 37.4796 g/cc and a radius of 8.7407 cm. Problem #3 (reflected fast system) is similar to Problem #2, except for the 30 cm radius-thick 1 g/cc light water (2.0 at.% ^{1}H, 1.0 at.% ^{16}O) covering the HEU sphere. Problem #4 (bare thermal system) is similar to Problem #2, except for the light water blended into the HEU sphere giving an atomic fraction composition of 0.937695 at.% ^{235}U, 0.052053 at.% ^{238}U, 0.010252 at.% ^{234}U, 200.0 at.% ^{1}H, and 100.0 at.% ^{16}O normalized to a density of 18.7398 g/cc.

The problems are run with 10^6 particles per generation and 300 active generations; the number of passive generations is set to 50 for Problems #2 and #3, and it is set to 250 for Problem #4. The resulting fundamental k and α eigenvalues are compared to those obtained by Cullen et al. and Zoia et al. [24][35], as shown in Table 6.1. Zoia et al. use ENDF/B-VII.0 at room temperature, which may contribute to the slight discrepancy. The resulting normalized fission neutron productions $\nu\Sigma_f\phi(E)$ associated with the fundamental k- and α-eigenfunction are shown in Figure 6.1; 1,000 equal-lethargy energy bins in interval 10^{-4} eV to 20 MeV and an additional bin encompassing interval 0 to 10^{-4} eV are used.

Table 6.1: Fundamental k- and α-eigenvalues of Cullen's Godiva problems

	Problem	#2	#3	#4
k	OpenMC	1.59458	1.67047	1.80426
	Tripoli-4 [35]	1.58455	1.66782	1.79282
	TART [24]	1.582	1.661	1.771
α, μs^{-1}	OpenMC	144.766	146.659	0.665
	Tripoli-4 [35]	144.9	146.9	0.671
	TART [24]	144.7	146.6	0.653



(a) Doubled-density Godiva

(b) Reflected Godiva

(c) Watered Godiva

Figure 6.1: Fission productions and eigenvalues convergences of Cullen's Godiva problems

The convergence of the eigenvalue α is shown in the right figures of Figure 6.1. It is also shown that the eigenvalue k converges to unity and the Shannon entropy (of $1\ \text{cm}^3$ cubes covering the system) converges to a constant. The calculated fundamental α-eigenvalues are reasonably close to those found by Cullen et al. and Zoia et al. The resulting normalized fission neutron productions shown in the left figures of Figure 6.1 agree with those of Cullen et al. It is shown that the k-eigenfunction overestimates the thermal neutron distributions; this is because the k-eigenvalue simulation disregards the time absorption (α/v) effect, which is stronger for the slow thermal neutrons. The criticality k of Problem #4 is larger than the others, but its α eigenvalue is considerably smaller; this is because the neutrons driving the chain reaction move much more slowly in the bare thermal system of Problem #4, leading to slower population growth or smaller α, regardless of super-prompt criticality. This indicates that in thermal systems, α and the time absorption effect are generally smaller and less pronounced than in fast systems. Problem #3 emphasizes a striking time absorption effect in a reflected fast system; the markedly different spectrum is evident because the static k-eigenvalue simulation does not account for the significant time absorption suffered by neutrons moderated in the reflector and reflected back into the core.

6.4.2 Two-Group Infinite Media

A two-group, infinite medium problem is considered:

$$\begin{aligned}\alpha\frac{1}{v_g}\phi_g+\Sigma_{t,g}\phi_g&=\sum_{g'=1}^{2}\Sigma_{s,g'\to g}\phi_{g'}\\&\quad+\sum_{g'=1}^{2}\chi_{p,g}(1-\beta)\nu_f\Sigma_{f,g'}\phi_{g'}+\sum_{j=1}^{2}\chi_{d,j\to g}\lambda_jC_j\,,\qquad g=1,2,\end{aligned}\tag{6.38}$$

$$\alpha C_j+\lambda_jC_j=\sum_{g'=1}^{2}\beta_j\nu_f\Sigma_{f,g'}\phi_{g'}\,,\qquad j=1,2.\tag{6.39}$$

The physical parameters used are those suggested by Zoia et al. [37]: $v_1=10$, $v_2=5$, $\Sigma_{s,1\to1}=1/2$, $\Sigma_{s,1\to2}=1/2$, $\Sigma_{s,2\to1}=0$, $\Sigma_{s,2\to2}=1$, $\Sigma_{f,1}=0$, $\Sigma_{f,2}=1$, $\Sigma_{t,1}=2$, $\Sigma_{t,2}=3$, $\beta_1=\delta/4$, $\beta_2=\delta/8$, $\chi_{p,1}=1$, $\chi_{p,2}=0$, $\chi_{d,1\to1}=3/4$, $\chi_{d,1\to2}=1/4$, $\chi_{d,2\to1}=1/2$, $\chi_{d,2\to2}=1/2$, $\lambda_1=$

0.5, and $\lambda_2 = 0.07$. Given ν_f and δ, the eigenvalue problem can be deterministically solved; the resulting dominant eigenvalue α and the associated neutron flux eigenvector $[\phi_1, \phi_2]$ serve as the benchmark solutions. Criticality ($\alpha = 0$) is achieved when $\nu_f = 24/5$ and $\delta = 1$. Two sets of problems are considered: (1) varying ν_f about $24/5$ with fixed $\delta = 1$, and (2) varying δ about 1 with fixed $\nu_f = 24/5$. These subcritical, critical, and delayed supercritical problems primarily test the time-corrected delayed neutron emission capability described in Section 6.3.2.

The two-group infinite-medium problems are solved with the static fundamental α-eigenvalue implementation in OpenMC with 10^6 particles per generation and 50 passive and 300 active generations. The Python library `numpy.linalg.eig` is used to directly solve the eigenvalue problems and obtain the reference solutions. The resulting fundamental α-eigenvalues and the thermal-to-fast ratios of the associated neutron flux eigenvectors, ϕ_2/ϕ_1, are shown and compared in Figure 6.2. Excellent agreement is found between the OpenMC results and the reference solutions.

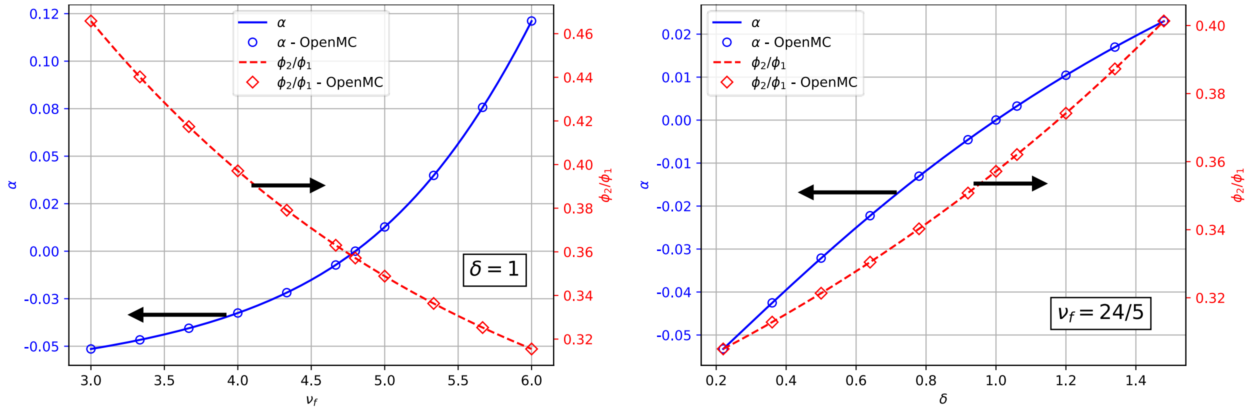


Figure 6.2: Fundamental eigenvalues and the thermal-to-fast flux ratios of 2G infinite media

CHAPTER 7

α-Weighted Multigroup Constants (α-MGXS)

In this chapter, we investigate the relative effectiveness of the $\boldsymbol{\alpha}$-MGXS and $\boldsymbol{k}$-MGXS. Multigroup constants that preserve the underlying time-dependent physics of the neutron transport are derived in Section 7.1. A simple infinite medium kinetic problem is considered; however, it serves as general guidance for applications in more practical problems. The typically-used k-weighted ($\boldsymbol{k}$-MGXS) and the proposed α-weighted multigroup constants ($\boldsymbol{\alpha}$-MGXS) are discussed in Section 7.2. Section 7.3 presents numerical results of four static and kinetic problems, which are devised to emphasize the significant differences of the fundamental k- and α-eigenfunctions and to compare the effectiveness of the $\boldsymbol{\alpha}$-MGXS and $\boldsymbol{k}$-MGXS.

7.1 Multigroup Constants for Time-Dependent Problem

Any time-stepping method used to solve the time-dependent problem of Eq. (6.1) involves operating the equation with $\Delta t^{-1}\int_{t_{n-1}}^{t_n}(\cdot)dt$:

$$\begin{cases} \dfrac{1}{v\Delta t}[\phi_n(E)-\phi_{n-1}(E)]+\Sigma_t(E)\bar{\phi}_n(E)=\displaystyle\int_0^\infty \nu_s\Sigma_s(E'\to E)\bar{\phi}_n(E')dE' \\ \quad +\displaystyle\int_0^\infty \chi_p(E'\to E)\nu_p\Sigma_f(E')\bar{\phi}_n(E')dE'+\sum_{i=1}^{I}\sum_{j=1}^{J}\chi_{d,i,j}(E)\lambda_{i,j}\bar{C}_{n,i,j}\,, \\ \quad \dfrac{1}{\Delta t}\left(C_{n,i,j}-C_{n-1,i,j}\right)+\lambda_{i,j}\bar{C}_{n,i,j}=\displaystyle\int_0^\infty \nu_{d,i,j}\Sigma_{f,i}(E')\bar{\phi}_n(E')dE'\,, \end{cases} \tag{7.1}$$

where the initial condition becomes $\phi_0(E)=\phi_{\text{init}}(E)$ and $C_{0,i,j}=C_{\text{init},i,j}$. The time-step and time-average solutions are defined as follows:

$$\begin{cases} \phi_n(E) = \phi(E, t_n), \qquad \bar{\phi}_n(E) = \dfrac{1}{\Delta t}\displaystyle\int_{t_n}^{t_{n+1}} \phi(E, t)\, dt, \\ \quad C_{n,i,j} = C_{i,j}(t_n), \qquad \bar{C}_{n,i,j} = \dfrac{1}{\Delta t}\displaystyle\int_{t_n}^{t_{n+1}} C_{i,j}(t)\, dt. \end{cases} \tag{7.2}$$

Everything is treated exactly up to this point. Now, by introducing the multigroup approximation, the equation reduces to

$$\begin{cases} \dfrac{1}{\Delta t}\left(\dfrac{1}{v_{n,g}}\phi_{n,g} - \dfrac{1}{v_{n-1,g}}\phi_{n-1,g}\right) + \Sigma_{t,g}\bar{\phi}_{n,g} \\ \quad = \displaystyle\sum_{g'=1}^{G} (\nu_s\Sigma_s)_{g'\to g}\bar{\phi}_{n,g'} + \sum_{g'=1}^{G} \left(\chi_p \nu_p \Sigma_f\right)_{g'\to g}\bar{\phi}_{n,g'} + \sum_{j=1}^{J} (\chi_d\lambda)_{j\to g}\bar{C}_{n,j}\,, \\ \quad \dfrac{1}{\Delta t}\left(C_{n,j} - C_{n-1,j}\right) + \lambda_j\bar{C}_{n,j} = \displaystyle\sum_{g'=1}^{G} \left(\nu_d\Sigma_f\right)_{g'\to j}\bar{\phi}_{n,g'}\,, \end{cases} \tag{7.3}$$

$$\phi_{0,g} = \int_{E_g} \phi_{\text{init}}(E)dE\,, \qquad C_{0,j} = \sum_{i=1}^{I} C_{\text{init},i,j}\,. \tag{7.4}$$

$$\begin{cases} \bar{\phi}_{n,g} = \displaystyle\int_{E_g} \bar{\phi}_n(E)dE\,, \qquad \bar{C}_{n,j} = \sum_i \bar{C}_{n,i,j}\,, \\ \phi_{n,g} = \displaystyle\int_{E_g} \phi_n(E)dE\,, \qquad C_{n,j} = \sum_i C_{n,i,j}\,. \end{cases} \tag{7.5}$$

The next step would be introducing a time-stepping method (e.g., BE, CN, etc.) to approximate the time-average quantities so that we can numerically solve the problem. However, our objective here is to derive multigroup constants that preserves the physical properties of the time-dependent system as much as possible.

Two kinds of homogenization occur in the transition from Eq. (7.1) to Eq. (7.3): (1) continuous energy E into discrete energy group g, and (2) isotopic precursor group (i, j) into effective precursor group j, as shown in Eq. (7.5). It is desirable for the multigroup constants (energy and effective precursor groups) in Eq. (7.3) to satisfy Eq. (7.5) and to preserve all the physical terms in Eq. (7.1). This suggests

$$\frac{1}{v_{n,g}} = \frac{\left[\int_{E_g} \frac{1}{v} \phi_n(E) dE\right]}{\left[\int_{E_g} \phi_n(E) dE\right]}, \tag{7.6}$$

$$\Sigma_{t,g} = \frac{\left[\int_{E_g} \Sigma_t(E) \bar{\phi}_n(E) dE\right]}{\left[\int_{E_g} \bar{\phi}_n(E) dE\right]}, \tag{7.7}$$

$$(\nu_s \Sigma_s)_{g' \to g} = \frac{\left[\int_{E_g} \int_{E_{g'}} \nu_s \Sigma_s(E' \to E) \bar{\phi}_n(E') dE' \, dE\right]}{\left[\int_{E_{g'}} \bar{\phi}_n(E') dE'\right]}, \tag{7.8}$$

$$\left(\chi_p \nu_p \Sigma_f\right)_{g' \to g} = \frac{\left[\int_{E_g} \int_{E_{g'}} \chi_p(E' \to E) \nu \Sigma_f(E') \bar{\phi}_n(E') dE' \, dE\right]}{\left[\int_{E_{g'}} \bar{\phi}_n(E') dE'\right]}, \tag{7.9}$$

$$\left(\nu_d \Sigma_f\right)_{g' \to j} = \frac{\left[\sum_i \int_{E_g} \nu_{d,i,j} \Sigma_{f,i}(E') \bar{\phi}_n(E') dE'\right]}{\left[\int_{E_{g'}} \bar{\phi}_n(E') dE'\right]}, \tag{7.10}$$

$$\lambda_j = \frac{\left[\sum_i \int_0^\infty \nu_{d,i,j} \Sigma_{f,i}(E') \bar{\phi}_n(E') dE'\right]}{\sum_i \frac{1}{\lambda_{i,j}} \left[\int_0^\infty \nu_{d,i,j} \Sigma_{f,i}(E') \bar{\phi}_n(E') dE'\right]}, \tag{7.11}$$

$$(\chi_d \lambda)_{j \to g} = \frac{\left[\int_{E_g} \chi_{d,j}(E) \int_0^\infty \nu_{d,j} \Sigma_f(E') \bar{\phi}_n(E') dE' \, dE\right]}{\left[\int_0^\infty \nu_{d,j} \Sigma_f(E') \bar{\phi}_n(E') dE'\right]} \lambda_j. \tag{7.12}$$

Eqs. (7.6)–(7.10) follow the typical flux-weighted multigroup constant definition. The effective precursor group decay constant λ_j is defined as shown in Eq. (7.11) to satisfy the effective precursor group definition in Eq. (7.5). Given Eqs. (7.5) and (7.11), the effective delayed neutron emission is defined as shown in Eq. (7.12) to preserve the group-wise total delayed neutron emission. Furthermore, it follows that the definitions of Eqs. (7.11) and (7.12) simultaneously implies

$$\begin{cases} \lambda_j = \dfrac{\sum_i \Delta \bar{C}_{n,i,j}}{\sum_i \dfrac{1}{\lambda_{i,j}} \Delta \bar{C}_{n,i,j}}, \qquad (\chi_d \lambda)_{j \to g} = \dfrac{\sum_i \int_{E_g} \chi_{d,i,j}(E) dE \, \Delta \bar{C}_{n,i,j}}{\sum_i \dfrac{1}{\lambda_{i,j}} \Delta \bar{C}_{n,i,j}}, \\ \Delta C_{\kappa,i,j} = C_{n,i,j} - C_{n-1,i,j} \,. \end{cases} \tag{7.13}$$

The multigroup constants defined in Eqs. (7.6)–(7.12) are relevant for any time-stepping method. Most Monte Carlo codes are capable of tallying physical quantities expressed by the square-bracketed terms in Eqs. (7.6)–(7.12). However, a tally filter of pre- and post-collision energy and of a delayed precursor group, as well as a nuclide-wise tally scoring mechanism, are needed to compute the desired quantities. These capabilities are offered in OpenMC as tally filter `EnergyFilter`, `EnergyoutFilter`, and `DelayedGroupFilter`, and tally attribute `nuclides`.

7.2 *α*-MGXS

Based on Eqs. (7.6)–(7.12), it is evident that to calculate multigroup constants that effectively preserve the underlying time-dependent physics of the system during the calculation time step n, estimates of $\phi_{n-1}(E)$, $\bar{\phi}_n(E)$ and $\phi_n(E)$ are needed as the weighting spectra. The typical approach $\boldsymbol{k}$-MGXS [20][21] assumes the following:

$$\phi_{n-1}(E) \approx \phi_n(E) \approx \bar{\phi}_n(E) \approx \phi_k(E), \tag{7.14}$$

where $\phi_k(E)$ is the fundamental eigenfunction of the k-eigenvalue neutron transport Eq. (6.2). However, per discussion in Section 6.1, the assumption of Eq. (7.14) is justified only if $k \approx 1$. This is because Eqs. (6.1) and (6.2), and thus Eq. (7.1), are essentially two different physical problems if $k \neq 1$ (in other words, we solve a wrong problem).

An alternative approach is to assume the weighting spectra in Eqs. (7.6)–(7.12) to be the following:

$$\phi_{n-1}(E) \approx \phi_n(E) \approx \bar{\phi}_n(E) \approx \phi_\alpha(E), \tag{7.15}$$

where $\phi_\alpha(E)$ is the fundamental eigenfunction of the α-eigenvalue neutron transport Eq. (6.5). Similar to the $\boldsymbol{k}$-MGXS approximation Eq. (7.14), the $\boldsymbol{\alpha}$-MGXS approximation Eq. (7.15) is justified if $\alpha \approx 0$ (or $k \approx 1$). In particular, for a critical system ($\alpha = 0$ and $k = 1$), Eqs. (6.1), (6.2), and (6.5) are equivalent, so that the energy dependence of $\phi_k(E)$, $\phi_\alpha(E)$, and $\phi(E,t)$ are identical. However, unlike the k-eigenvalue Eq. (6.2) (which is the basis of $\boldsymbol{k}$-MGXS), the α-eigenvalue Eq. (6.5) is still effectively equivalent to Eq. (6.1) given the factorization Eq. (6.4), even if $k \neq 1$ (in other words, we still solve the right problem).

The $\boldsymbol{\alpha}$-MGXS approximation of Eq. (7.15) is based on the fact that the fundamental α-eigenfunction $\phi_\alpha(E)$ represents the asymptotic shape of the actual flux $\phi(E,t)$. Eventually, the energy dependence (not the amplitude) of $\phi(E,t)$ converges to $\phi_\alpha(E)$. In particular, if $\phi_{n-1}(E) = \phi_\alpha(E)$, then the actual time-dependent neutron flux $\phi(E,t)$ already reaches the asymptotic shape and evolves only in magnitude. It follows that the closer the initial condition is to the fundamental α-eigenfunction—i.e., $\phi_{n-1}(E) \approx \phi_\alpha(E)$—the better the $\boldsymbol{\alpha}$-MGXS should work. This physical feature is not offered by the $\boldsymbol{k}$-MGXS approximation Eq. (7.14). The α-eigenvalue iteration method implementation in OpenMC (discussed in Chapter 6) is used to solve Eq. (6.5) and generate the fundamental α-eigenfunction weighting spectrum $\phi_\alpha(E)$ for calculating the multigroup constants in Eqs. (7.6)–(7.12).

7.3 Numerical Results

7.3.1 Static Problems

Four problems are considered: prompt supercritical and subcritical homogeneous fast and thermal infinite medium systems. The doubled-density and watered HEU material in Cullen's Godiva Problems #2 and #4 are used for the prompt supercritical fast and thermal systems, respectively. Boron-10 with atomic fractions of 10 at.% and 0.5 at.% are respectively mixed into the HEU materials to make the subcritical fast and thermal systems.

Table 7.1: Fundamental k- and α-eigenvalues of the static infinite medium problems

Problem	Fast system		Thermal system	
	Supercritical	Subcritical	Supercritical	Subcritical
k	2.26138	0.42924	1.82603	0.48846
α (s^{-1})	223.852×10^6	-0.013082	0.683×10^6	-0.013082

The problems are run with 10^6 particles per generation and 300 active generations; the number of passive generations is respectively set to 50 and 250 for the fast and thermal systems. The resulting fundamental k and α eigenvalues are shown in Table 7.1. The resulting normalized neutron fluxes, presented in a semi-log plot of $E\phi(E)$, associated with the fundamental k- (broken red line) and α-eigenfunction (solid blue line) are shown in Figure 7.1; 1,000 equal-lethargy energy bins in interval 10^{-4} eV to 20 MeV and an additional bin encompassing interval 0 to 10^{-4} eV are used for the thermal systems (the threshold 10^{-4} eV is replaced with 100 eV for the fast systems).

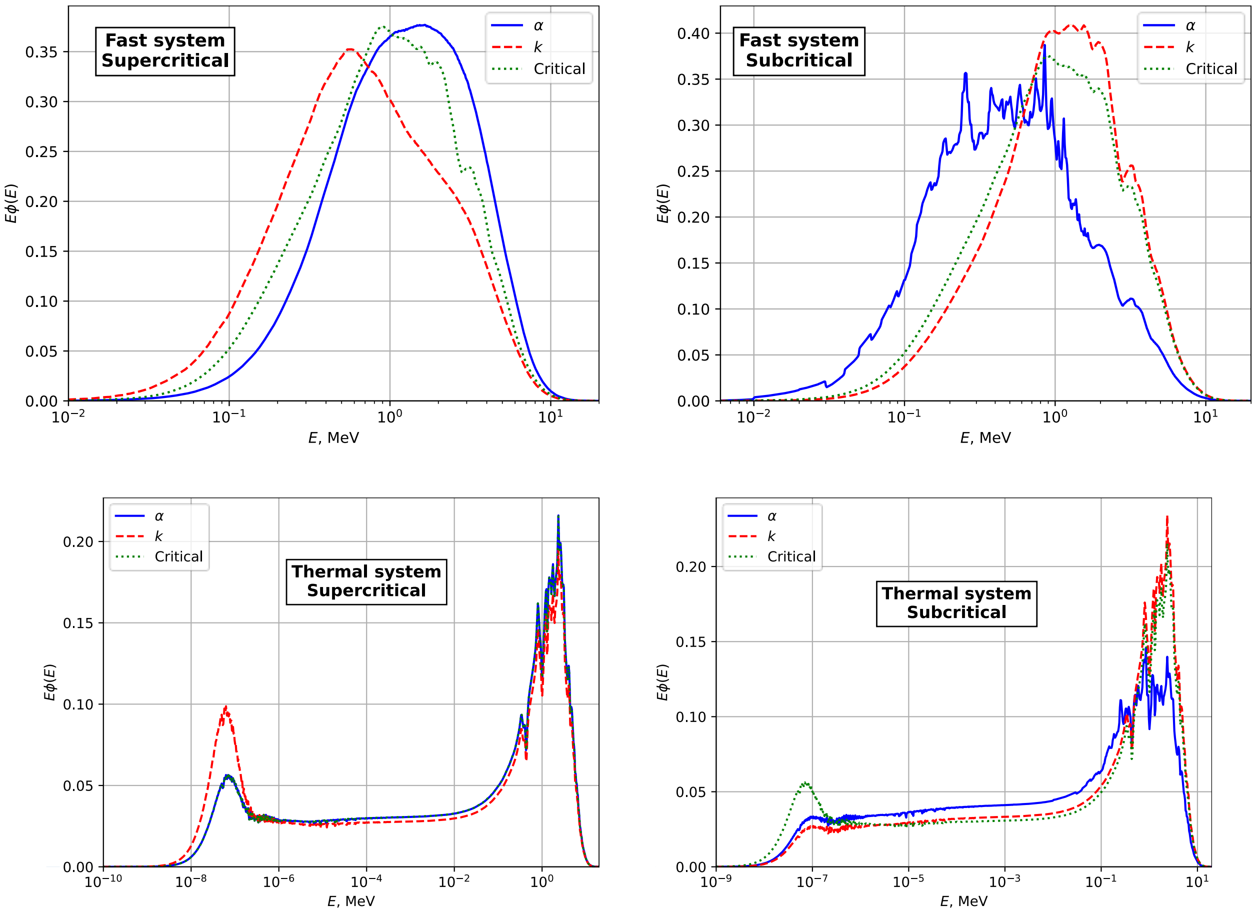


Figure 7.1: Fundamental k- and α-eigenfunctions for the four static problems

We emphasize that a semi-log plot of $E\phi(E)$, like those shown in Figure 7.1, effectively visualizes the relative weight of $\phi(E)$. A ratio of any two visual areas under a semi-log plot of $E\phi(E)$ is equivalent to the integral ratio of the corresponding $\phi(E)$; that is, if A_1 and A_2 are visual areas respectively spanning energy range $[E_1,\ E_0]$ and $[E_2,\ E_1]$, then

$$\frac{A_1}{A_2} = \frac{\int_{\log E_1}^{\log E_0} E\phi(E)d(\log E)}{\int_{\log E_2}^{\log E_1} E\phi(E)d(\log E)} = \frac{\int_{E_1}^{E_0} \phi(E)dE}{\int_{E_2}^{E_1} \phi(E)dE}. \tag{7.16}$$

Table 7.1 shows that the calculated fundamental α-eigenvalues for the subcritical problems are slightly greater than $-\min(\lambda_{i,j}) = -0.013336\ \mathrm{s}^{-1}$ (λ_1 of ^{235}U). It is important to emphasize how the resulting fundamental k- and α-eigenfunctions differ, because they are used as the multigroup constant weighting spectra in the kinetic problems discussed in the next section. It is shown that in the supercritical problems (figures on the left in Figure 7.1), the fundamental k-eigenfunctions overestimate the distribution of the slower neutrons. In the subcritical problems (figures on the right in Figure 7.1), despite the small magnitude of α (and thus negligible delta time source effect), significant differences are evident in the fundamental k- and α-eigenfunctions; this is primarily because the delayed fission spectra dominate the prompt. The relatively jagged delayed fission spectra and the high probability of scattering with ^{10}B give raise to the strongly jagged structure of the subcritical fast system spectrum, as shown on the top-right figure of Figure 7.1.

7.3.2 Kinetic Problems

Four kinetic problems are considered. The four static problems presented in the previous subsection are used as the perturbed system in the respective kinetic problems. The initial condition is chosen to be the critical configurations obtained by adjusting the Boron-10 content in the fast and thermal systems. Therefore, these kinetic problems simulate absorber injection and removal (at $t = 0$) to the initially-critical fast and thermal systems. In all of the problems, the flux shape or the energy distribution converges from the initial critical spectrum (dotted green in Figure 7.1) to the fundamental α-eigenfunction (solid blue in Figure 7.1) in time, and it may have nothing to do with the fundamental k-eigenfunction (broken red in Figure 7.1) at all. Table 7.2 shows the material composition of the initial critical and perturbed systems.

The kinetic problems are solved by adapting the $\boldsymbol{\alpha}$-MGXS and $\boldsymbol{k}$-MGXS, which respectively use the fundamental k- and α-eigenfunction as the weighting spectrum in calculating the multigroup constants per Eqs. (7.6)–(7.12). The thermal systems are solved with energy groups $G = 1, 2, 4, 8, 16$, and the fast systems are solved with $G = 1, 3, 5, 9, 17$. The group structures of the thermal systems are those of OpenMC's `CASMO-X` group structures. The group structures of the fast systems are similar to the one used in the static problems: $G - 1$ equal-lethargy energy bins in interval 100 eV to 20 MeV and an additional bin encompassing interval 0 to 100 eV. Finally, the second-order accurate time-stepping method CN is used with a thousand uniform time steps placed between the reference solutions logarithmically spaced time points to minimize time discretization error in solving the kinetic problems. TRMM [30] with $G = 1{,}001$ energy groups (identically structured like those of the static problem plots shown in Figure 7.1) are used as the reference solutions. The metric being compared to the reference solution is the total neutron flux $\phi(t) = \int_0^\infty \phi(E, t)dE$ at logarithmically spaced time points.

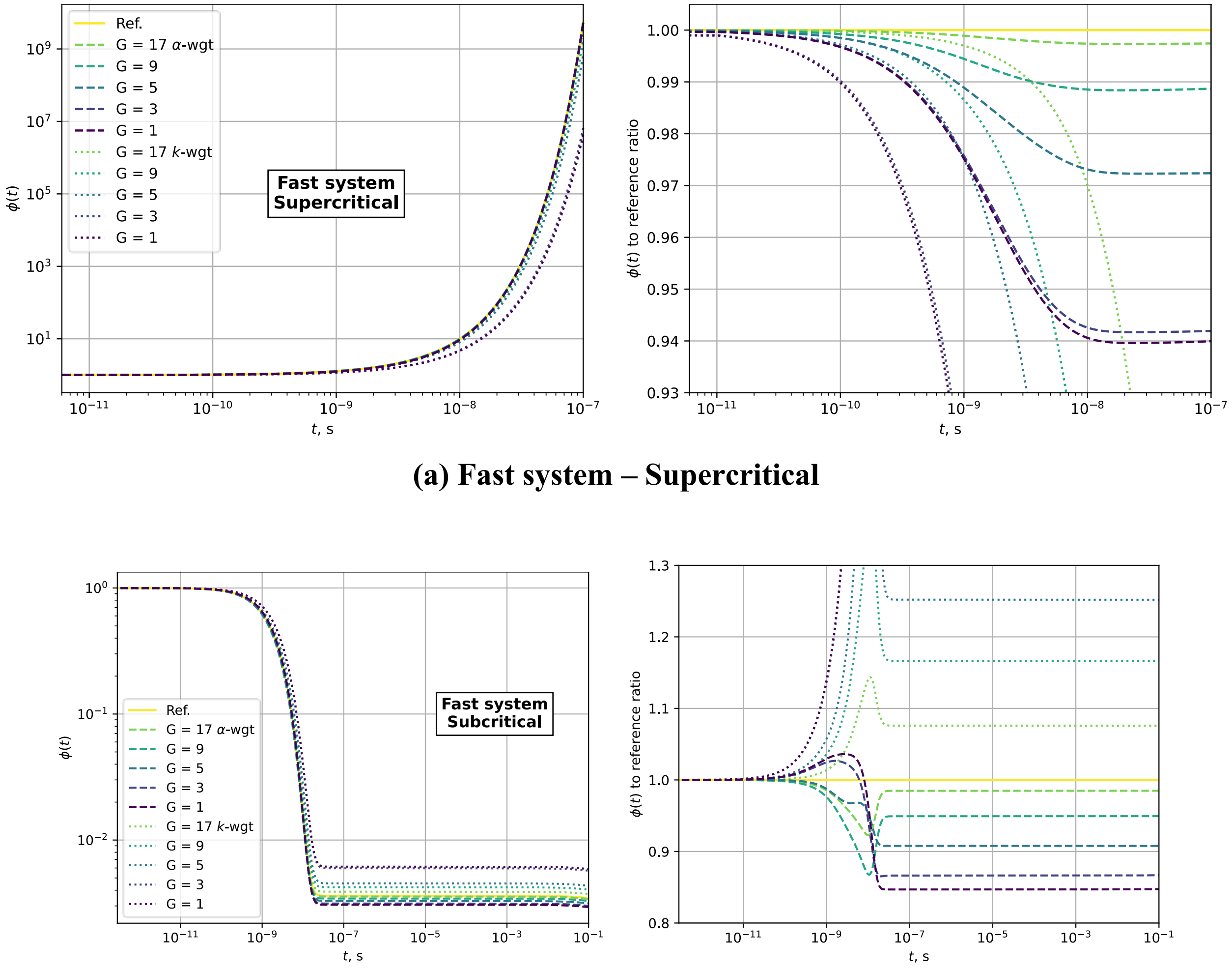
Table 7.2: Material composition for the kinetic problems

Atomic fraction, at.%		^{10}B	^{1}H	^{16}O	^{234}U	^{235}U	^{238}U
Fast system (37.4796 g/cc)	Critical	2.7092	0.0	0.0	0.010252	0.937695	0.052053
	Supercritical	0.0					
	Subcritical	10.0					
Thermal system (18.7398 g/cc)	Critical	0.14761	200.0	100.0			
	Supercritical	0.0					
	Subcritical	0.5					

Figure 7.2 shows the resulting total fluxes $\phi(t)$ (left figures) and the ratios of $\phi(t)$ to the reference (right figures). It is found that the solutions calculated using $\boldsymbol{\alpha}$-MGXS are generally more accurate than those using $\boldsymbol{k}$-MGXS. $\boldsymbol{\alpha}$-MGXS preserves the asymptotic behavior of all the four problems (flat lines at longer times in the right figures in Figure 7.2), regardless of G. Meanwhile, $\boldsymbol{k}$-MGXS only correctly captures the asymptotic behavior of the subcritical problems, which is mainly governed by the precursor group decay. The better performance of the $\boldsymbol{\alpha}$-MGXS is mainly because the initial condition critical spectra (dotted green in Figure 7.1) are close to the fundamental α-eigenfunction (solid blue in Figure 7.1). The fundamental k-eigenfunction spectra (broken red in Figure 7.1) may have nothing to do with the actual neutron population throughout time, since the time-dependent neutron energy distribution shape "moves" from the initial critical spectrum to the fundamental α-eigenfunction.

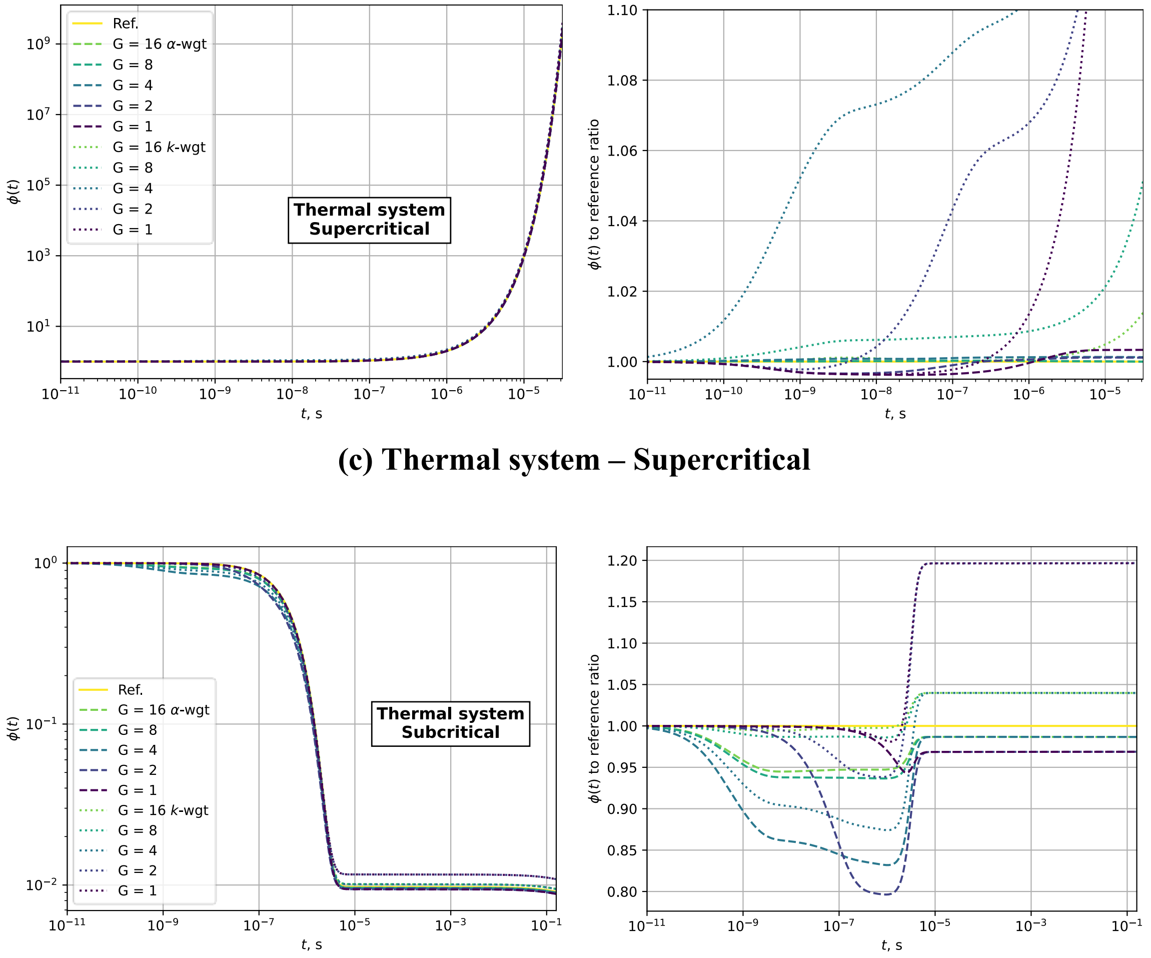
The supercritical thermal system is a special case: the initial critical spectrum is almost identical to the fundamental α-eigenfunction, as shown in the bottom-left figure of Figure 7.1.

This which leads to an exceptionally accurate solution throughout time, regardless of G. The two spectra are nearly identical because the only difference between the perturbed system and the initial system is the amount of Boron-10, for which the thermal absorption cross-section is proportional to $1/v$. This $1/v$ absorption in the initial critical system is effectively replaced by the time absorption α/v in the static calculation of the perturbed supercritical system. Another special case is the subcritical thermal system. In this case, the k-eigenfunction is the better match to the initial critical spectrum, as shown in the bottom-right figure of Figure 7.1. This enables the k-weighted approach to produce more accurate solutions at early times; nevertheless, the α-weighted approach is still more accurate at late times.



(a) Fast system – Supercritical

(b) Fast system – Subcritical



(c) Thermal system – Supercritical

(d) Thermal system – Subcritical

Figure 7.2: Total neutron flux in time (left) and its ratio to the reference solution (right)

CHAPTER 8

Conclusions and Future Work

8.1 MBTD

Multiple Balance Time-Discretization (MBTD) is a robust second-order accurate time-discretization method, an alternative to the highly reliable Backward Euler (BE). This thesis work investigates the applications of MBTD for time-dependent neutron diffusion (Chapter 3) and neutron transport (Chapter 4) problems, including a first attempt to solve a typical time-dependent multi-physics problem of coupled neutronics/thermal-hydraulics (Chapter 5). The main objective of the investigation is to assess the relative effectiveness of MBTD compared to BE, thus deciding whether MBTD is a worthy higher-order alternative robust method and a more efficient method than BE, taking into account (1) with the same computational resource, higher accuracy can be achieved, and/or (2) to achieve a certain accuracy, less computational resources are required. Simple, yet representative, mono-energetic 1-D slab problems with isotropic scattering are considered. Despite their simplicity, these problems yield interesting and informative results, and their numerical analysis and results reveal strong incentives for application to more practical problems.

The basics of MBTD were discussed in Chapter 2, including its formulation for linear and non-linear problems, its mathematical relation to other time-stepping methods, its second-order accuracy and robustness, and the significance and computational consequences of the MBTD coupled balance equations. Three strategies for solving the coupled balance equations were compared: (1) Substitution, (2) Iterative Solve, and (3) Simultaneous Solve. It was found that Substitution is the most efficient strategy, but only if numerically handling the squared operator is practical. Iterative Solve is attractive because it is the least invasive strategy, as we can use an existing steady-state solver as a black-box and iteratively solve the system; however, modification of the methods (e.g., limiting the number of inner iterations per outer iteration of the overall iteration scheme) is needed to make MBTD with Iterative Solve competitive to BE. Simultaneous Solve is by far the optimum strategy, due to its easier implementation and reasonably good computational performance.

Adaptations of the standard and the traditional delayed neutron source approximations (applicable for both neutron diffusion and transport) for MBTD were derived. It is found that the standard MBTD delayed source formulation is the most accurate method that maintains second-order accuracy.

Adaptations of the neutron transport Source Iteration (SI) and Diffusion Synthetic Acceleration (DSA) for MBTD were also derived. Fourier analysis revealed that the spectral radii, and thus convergence rates, of the derived MBTD transport iteration methods are similar to those of BE.

The computational complexity of MBTD, relative to that of BE, was assessed by comparing the problems resulted from their discretization methods. For neutron diffusion Finite Difference Method (FDM), MBTD yields a four-block matrix problem, where the diagonal blocks are of typical steady-state neutron diffusion operator, and the off-diagonals are diagonal matrices. In neutron transport, Discrete Ordinates with Diamond-Difference (SN-DD) yields a 4×4 (instead of just 2×2) tridiagonal matrix problem to be solved at each step of the transport sweep; while for MOC, MBTD needs to analytically solve a system of two (instead of one) heterogeneous first-order ODEs having complex conjugate matrix eigenvalues.

Four infinite-medium kinetic test problems were devised: (1) a source-driven subcritical system without delayed neutrons, (2) a source-driven subcritical system, (3) a delayed supercritical system, and (4) a prompt supercritical system. These four analytical test problems were used as benchmarks to verify accuracy and stability, and to assess the efficiency of the MBTD methods for neutron diffusion and neutron transport. It was found that—relative to BE—MBTD is about (1) two times computationally more expensive for neutron diffusion with FDM, (2) 2.5 times more expensive for neutron transport DSA with SN-DD, and (3) 3 times more expensive for neutron transport DSA with MOC, given the same value of Δt. However, thanks to its second-order of accuracy, we found that MBTD is generally more efficient than BE, for reasonably accurate simulations.

Similar trends were observed in the MBTD application for the multi-physics problem, which is a time-dependent problem adaptation of the simple, yet representative, 1D multi-physics steady-state problem proposed in [49]. By adapting the staggered multi-physics tight-coupling, we found that MBTD is about three times computationally more expensive than BE given the same value of Δt, but its second-order accuracy again makes MBTD more efficient than BE for reasonably accurate simulations (relative error smaller than $\sim 10\%$).

Future work includes testing the MBTD methods for solving more practical problems in which we have a heterogeneous multidimensional problem with multigroup cross-sections and anisotropic scattering. The preferred Simultaneous Solve strategy, which essentially solves the four-block operator problem, scales well with the increasing problem complexity, as illustrated in Figure 2.5: if the physical problem becomes more complex, the diagonal blocks individually follow the increasing complexity of the steady-state operator, while the off-diagonal blocks remain as diagonal operators. Furthermore, will be necessary to consider integrating MBTD into the more advanced neutron diffusion and transport methods, such as nodal diffusion methods [8] and quasi-static time-stepping methods [4][5], which are typically used in solving the more challenging practical problems.

In Chapter 4, it was revealed that the combinations of MBTD and SN-DD yield conditionally converging methods. This warrants a Fourier analysis of the discretized systems to better characterize the convergence rate of the methods. Another suggested convergence analysis work is to study the effect of the *coarse* diffusion acceleration typically done in practice, in which the error diffusion problem has a coarser spatial mesh compared to the one used in the main neutron transport problem.

A multi-physics tight-coupling method was used in Chapter 5 to test the MBTD application for solving multi-physics problems. Another widely used multi-physics coupling technique is the *loose-coupling* (also known as *operator-splitting*). This is similar to tight-coupling, except that no iteration is performed. The lagged terms are replaced by the solution from the previous time step. However, it is well understood that traditional operator-splitting introduces a first-order inaccuracy, which severely hinders the potential of MBTD. Therefore, to maximize the efficiency of MBTD in the multi-physics loose coupling, an operator-splitting technique that allows higher-order accuracy is needed [7].

8.2 $\boldsymbol{\alpha}$-MGXS

Deterministic methods remain as essential tools in reactor transient simulations. Since the deterministic methods are only as good as their multigroup constants, appropriate multigroup constants that preserve the time-dependent physics of the problem need to be calculated. The effectiveness of the α-Weighted Multigroup Constants ($\boldsymbol{\alpha}$-MGXS) offers advantageous physical characteristics over the traditionally used k-Weighted Multigroup Constants ($\boldsymbol{k}$-MGXS) for time-dependent neutron transport simulation because of their ability to preserve the asymptotic behavior of the system.

A review of α-eigenvalue iteration methods is presented in Chapter 7. The Relaxed α-k Iteration method is implemented into OpenMC and verified with Cullen's Godiva problems and two-region infinite medium problems, respectively testing the time absorption/source and time-corrected delayed fission emission capabilities. This new implementation is used to generate the fundamental α-eigenfunction for multigroup constant calculations.

Results from kinetic problems simulating absorber injection and removal to the initially-critical infinite-medium fast and thermal systems show that the $\boldsymbol{\alpha}$-MGXS generally produces more accurate solutions than the $\boldsymbol{k}$-MGXS. These results emphasize that, as a multigroup constant weighting spectrum, the fundamental α-eigenfunction offers physical characteristics that make it advantageous (in producing accurate solutions) over the typically used fundamental k-eigenfunction. The next work on $\boldsymbol{\alpha}$-MGXS is to investigate its effectiveness in simulating more practical heterogeneous reactor problems.

APPENDICES

APPENDIX A

SN-DD and MOC for MBTD

Derivations of SN-DD and MOC for BE are presented first, followed by those for MBTD. Simplified but representative equations are used as the starting points of the derivation. Please refer to Section 4.3 for a full context of the methods.

At each step of BE transport sweep, we would like to solve the following equation:

$$|\mu|\frac{d\psi}{dx}+\left(\Sigma_t+\frac{1}{v\Delta t}\right)\psi(x)=\hat{Q}, \qquad x\in[0,\Delta x], \tag{A.1}$$

to obtain the *cell-average* and the *outgoing cell-edge* solutions:

$$\bar{\psi}=\frac{1}{\Delta x}\int_0^{\Delta x}\psi(x)dx\,, \qquad \psi^+=\psi(\Delta x). \tag{A.2}$$

from the given *incoming cell-edge* solution:

$$\psi(0)=\psi^-. \tag{A.3}$$

SN-DD operates Eq. (A.1) with $\Delta x^{-1}\int_0^{\Delta x}(\cdot)dx$ and introduces the *diamond-difference auxiliary equation*:

$$\frac{1}{\tau}(\psi^+-\psi^-)+\left(1+\frac{1}{\eta}\right)\bar{\psi}=\frac{\hat{Q}}{\Sigma_t}, \tag{A.4}$$

$$\bar{\psi}=(\psi^++\psi^-), \tag{A.5}$$

Where the non-dimensional parameters are defined as

$$\tau = \frac{\Sigma_t \Delta x}{|\mu|}, \qquad \eta = v\Sigma_t \Delta t. \tag{A.6}$$

We note that Eqs. (A. 4) and (A. 5) make a 2×2 matrix problem with the unknown vector $[\bar{\psi} \quad \psi^+]^T$.

MOC *analytically* solves the heterogeneous linear ODE [Eq. (A. 1)]:

$$\psi(x) = \left(\psi^- - \frac{\hat{Q}}{\zeta\Sigma_t}\right) e^{\left(-\zeta\frac{\Sigma_t x}{|\mu|}\right)} + \frac{\hat{Q}}{\zeta\Sigma_t}, \tag{A.7}$$

where

$$\zeta = 1 + \frac{1}{\eta}. \tag{A.8}$$

This gives

$$\psi^+ = \left(\psi^- - \frac{\hat{Q}}{\zeta\Sigma_t}\right) e^{-\zeta\tau} + \frac{\hat{Q}}{\zeta\Sigma_t}, \tag{A.9}$$

$$\bar{\psi} = \frac{\hat{Q}}{\zeta\Sigma_t} - \frac{1}{\tau}(\psi^+ - \psi^-). \tag{A.10}$$

Now let us derive SN-DD and MOC for MBTD. At each step of MBTD transport sweep, instead of Eq. (A. 1), we need to solve the following coupled equations:

$$\begin{cases} |\mu|\dfrac{d\psi_1}{dx} + \Sigma_t\psi_1(x) + \dfrac{1}{v\Delta t}\psi_2(x) = \hat{Q}_1, \\ -\dfrac{2}{v\Delta t}\psi_1(x) + |\mu|\dfrac{d\psi_2}{dx} + \left(\Sigma_t + \dfrac{2}{v\Delta t}\right)\psi_2(x) = \hat{Q}_2, \end{cases} \tag{A.11}$$

to obtain the cell-average ($\bar{\psi}_1$ and $\bar{\psi}_2$) and the outgoing cell-edge solutions (ψ_1^+, ψ_2^+), from the given incoming cell-edge solution:

$$\psi_1(0) = \psi_1^-, \qquad \psi_2(0) = \psi_2^-. \tag{A.12}$$

We note that $\psi_1(x)$ and $\psi_2(x)$ corresponds to the *time-average* (or *mid-point*) and the *next time-step* solutions, respectively.

SN-DD operates Eq. (A.11) with $\Delta x^{-1}\int_0^{\Delta x}(\cdot)dx$ and introduces the diamond-difference auxiliary equations:

$$\begin{cases} \frac{1}{\tau}(\psi_1^+ - \psi_1^-) + \bar{\psi}_1 + \frac{1}{\eta}\bar{\psi}_2 = \frac{\hat{Q}_1}{\Sigma_t}, \\ -\frac{2}{\eta}\bar{\psi}_1 + \frac{1}{\tau}(\psi_2^+ - \psi_2^-) + \left(1+\frac{2}{\eta}\right)\bar{\psi}_2 = \frac{\hat{Q}_2}{\Sigma_t}, \end{cases} \tag{A.13}$$

$$\bar{\psi}_1 = (\psi_1^+ + \psi_1^-), \qquad \bar{\psi}_2 = (\psi_2^+ + \psi_2^-), \tag{A.14}$$

where the non-dimensional parameters η and τ are identical to those in Eq. (A.6). We note that Eqs. (A.13) and (A.14) make a 4×4 tridiagonal matrix problem with the unknown vector $[\bar{\psi}_1 \quad \psi_1^+ \quad \bar{\psi}_2 \quad \psi_2^+]^T$.

MOC analytically solves the coupled heterogeneous linear ODEs [Eq. (A.11)]. We start by re-casting Eqs. (A.11) to a matrix form:

$$\frac{|\mu|}{\Sigma_t}\frac{d\boldsymbol{\psi}}{dx} + \boldsymbol{A}\boldsymbol{\psi}(x) = \frac{\widehat{\boldsymbol{Q}}}{\Sigma_t}, \tag{A.15}$$

with initial condition

$$\boldsymbol{\psi}(0) = \boldsymbol{\psi}^- = \begin{bmatrix}\psi_1^- \\ \psi_2^-\end{bmatrix}, \tag{A.16}$$

where

$$\boldsymbol{\psi}(x) = \begin{bmatrix}\psi_1(x) \\ \psi_2(x)\end{bmatrix}, \qquad \widehat{\boldsymbol{Q}} = \begin{bmatrix}\hat{Q}_1 \\ \hat{Q}_2\end{bmatrix}, \qquad \boldsymbol{A} = \begin{bmatrix}1 & 1/\eta \\ -2/\eta & 1+2/\eta\end{bmatrix}, \tag{A.17}$$

and

$$\bar{\boldsymbol{\psi}} = \begin{bmatrix}\bar{\psi}_1 \\ \bar{\psi}_2\end{bmatrix} = \frac{1}{\Delta x}\int_0^{\Delta x}\boldsymbol{\psi}(x)dx, \qquad \boldsymbol{\psi}^+ = \begin{bmatrix}\psi_1^+ \\ \psi_2^+\end{bmatrix} = \boldsymbol{\psi}(\Delta x). \tag{A.18}$$

The general solution is

$$\boldsymbol{\psi}(x) = \boldsymbol{\psi_h}(x) + \boldsymbol{\psi_p}, \tag{A.19}$$

where the particular solution $\boldsymbol{\psi_p}$ can be straightforwardly obtained:

$$\boldsymbol{\psi_p} = \boldsymbol{A^{-1}}\frac{\widehat{\boldsymbol{Q}}}{\Sigma_t}, \tag{A.20}$$

$$\boldsymbol{A^{-1}} = \left(1 + \frac{2}{\eta} + \frac{2}{\eta^2}\right)^{-1} \begin{bmatrix} 1 + 2/\eta & -1/\eta \\ 2/\eta & 1 \end{bmatrix}. \tag{A.21}$$

The homogeneous solution can be determined by solving

$$\frac{d\boldsymbol{\psi_h}}{dx} + \frac{\Sigma_t}{|\mu|}\boldsymbol{A}\boldsymbol{\psi_h}(x) = 0. \tag{A.22}$$

We need to determine the eigenvalues of the system of ODEs [Eqs. (A.22)]; it is found that it has complex conjugate eigenvalues:

$$\lambda_{\pm} = \lambda_{Re} \pm i\lambda_{Im}, \tag{A.23}$$

$$\lambda_{Re} = \zeta\frac{\Sigma_t}{|\mu|}, \qquad \lambda_{Im} = \frac{1}{\eta}\frac{\Sigma_t}{|\mu|}, \tag{A.24}$$

where ζ is identically defined in Eq. .

Let us consider a solution that corresponds to the eigenvalue λ_+. The associated eigenvector can be found to be

$$\boldsymbol{V}_+ = \begin{bmatrix} 1 \\ 1 + i \end{bmatrix}. \tag{A.25}$$

Thus, the corresponding solution is

$$\boldsymbol{\psi}_+(x) = e^{-\lambda_+}\boldsymbol{V}_+ = e^{-\lambda_{Re}x}\{[\boldsymbol{V_1}(x) + i\boldsymbol{V_2}(x)]\}, \tag{A.26}$$

where

$$\boldsymbol{V_1}(x) = \begin{bmatrix} \cos(\lambda_{Im}x) \\ \cos(\lambda_{Im}x) + \sin(\lambda_{Im}x) \end{bmatrix}, \qquad \boldsymbol{V_2}(x) = \begin{bmatrix} -\sin(\lambda_{Im}x) \\ \cos(\lambda_{Im}x) - \sin(\lambda_{Im}x) \end{bmatrix}. \tag{A.27}$$

These suggest that the homogeneous solution has the following form:

$$\boldsymbol{\psi_h}(x) = [C_1\boldsymbol{V_1}(x) + C_2\boldsymbol{V_2}(x)]e^{-\lambda_{Re}x}. \tag{A.28}$$

The constants C_1 and C_2 can be determined from the initial condition:

$$\boldsymbol{\psi}^- = [C_1\boldsymbol{V_1}(0) + C_2\boldsymbol{V_2}(0)] + \boldsymbol{\psi_p}. \tag{A.29}$$

Therefore,

$$\boldsymbol{\psi}^+ = [C_1\boldsymbol{V_1}(\Delta x) + C_2\boldsymbol{V_2}(\Delta x)]e^{-\zeta\tau} + \boldsymbol{\psi_p}, \tag{A.30}$$

$$\overline{\boldsymbol{\psi}} = \boldsymbol{\psi_p} - \frac{1}{\tau}\boldsymbol{A}^{-1}(\boldsymbol{\psi}^+ - \boldsymbol{\psi}^-). \tag{A.31}$$

APPENDIX B

Fourier Analysis of MBTD-DSA

Details of the algebra in the Fourier analysis of MBTD-DSA (described in Section 4.5) is presented here. Similar to the Fourier analysis of MBTD-SI in Section 4.4, a simple non-fission infinite ($-\infty < x < \infty$) homogeneous medium problem with isotropic scattering is considered.

The iteration error equations for MBTD-DSA can be obtained by referring to the iteration scheme shown in Eqs. (4.82), (4.84), and (4.88):

$$\begin{cases} \left(\mu\dfrac{\partial}{\partial x}+\Sigma_t\right)g^{(l-1/2)}(x,\mu)+\dfrac{1}{v\Delta t}f^{(l-1/2)}(x,\mu)=\dfrac{1}{2}\Sigma_s G^{(l-1)}(x), \\ -\dfrac{2}{v\Delta t}g^{(l-1/2)}(x,\mu)+\left[\mu\dfrac{\partial}{\partial x}+\left(\Sigma_t+\dfrac{2}{v\Delta t}\right)\right]f^{(l-1/2)}(x,\mu)=\dfrac{1}{2}\Sigma_s F^{(l-1)}(x) \end{cases} \tag{B.1}$$

$$\begin{cases} G^{(l-1/2)}(x)=\displaystyle\int_{-1}^{1}g^{(l-1/2)}(x,\mu)d\mu\,, \\ F^{(l-1/2)}(x)=\displaystyle\int_{-1}^{1}f^{(l-1/2)}(x,\mu)d\mu\,, \end{cases} \tag{B.2}$$

$$\begin{cases} \left[-\widetilde{D}\dfrac{d^2}{dx^2}+\Sigma_a\right]G_0(x)+\left[D^*\dfrac{d^2}{dx^2}+\dfrac{1}{v\Delta t}\right]F_0(x) \\ \qquad\qquad =\Sigma_s\left[G^{(l-1/2)}(x)-G^{(l-1)}(x)\right], \\ -2\left[D^*\dfrac{d^2}{dx^2}+\dfrac{1}{v\Delta t}\right]G_0(x)+\left[-\widehat{D}\dfrac{d^2}{dx^2}+\left(\Sigma_a+\dfrac{2}{v\Delta t}\right)\right]F_0(x) \\ \qquad\qquad =\Sigma_s\left[F^{(l-1/2)}(x)-F^{(l-1)}(x)\right], \end{cases} \tag{B.3}$$

$$\begin{cases} G^{(l)}(x) = G^{(l-1/2)}(x) + G_0(x), \\ F^{(l)}(x) = F^{(l-1/2)}(x) + F_0(x), \end{cases} \tag{B.4}$$

where

$$D^* = \frac{1}{\eta + 2 + \frac{2}{\eta}} \frac{1}{3\Sigma_t}, \qquad \widetilde{D} = (\eta + 2)D^*, \qquad \widehat{D}(x) = \eta D^*. \tag{B.5}$$

Here $g^{(l-1/2)}$, $f^{(l-1/2)}$, $G^{(l-1/2)}$, and $F^{(l-1/2)}$, are the intermediate solution errors; G_0 and F_0 are the correction terms; $G^{(l)}$ and $F^{(l)}$ are the iteration solution errors.

As mentioned in Section 4.5, we consider the following Fourier ansatz:

$$\begin{cases} G^{(l)}(x) = \theta^l B e^{i\omega\Sigma_t x}, & F^{(l)}(x) = \theta^l A e^{i\omega\Sigma_t x}, \\ g^{(l-1/2)}(x,\mu) = \theta^{l-1}\hat{b}(\mu) e^{i\omega\Sigma_t x}, & f^{(l-1/2)}(x,\mu) = \theta^{l-1}\hat{a}(\mu) e^{i\omega\Sigma_t x}, \\ G^{(l-1/2)}(x) = \theta^{l-1}\hat{B} e^{i\omega\Sigma_t x}, & F^{(l-1/2)}(x) = \theta^{l-1}\hat{A} e^{i\omega\Sigma_t x}, \\ G_0(x) = \theta^{l-1}\gamma_G e^{i\omega\Sigma_t x}, & F_0(x) = \theta^{l-1}\gamma_F e^{i\omega\Sigma_t x}. \end{cases} \tag{B.6}$$

By introducing the Fourier ansatz into the the iteration error equations, we obtain

$$\begin{cases} \hat{b}(\mu) = \frac{c}{2}\left[\frac{h_1(\omega,\mu)}{h_3(\omega,\mu)} B - \frac{1}{\eta}\frac{h_{12}(\omega,\mu)}{h_3(\omega,\mu)} A\right], \\ \hat{a}(\mu) = \frac{c}{2}\left[\frac{2}{\eta}\frac{h_{12}(\omega,\mu)}{h_3(\omega,\mu)} B + \frac{h_2(\omega,\mu)}{h_3(\omega,\mu)} A\right], \end{cases} \tag{B.7}$$

$$\begin{cases} \hat{B} = \int_{-1}^{1} \hat{b}(\mu) d\mu = I_1(\omega) B - \frac{1}{\eta} I_{12}(\omega) A, \\ \hat{A} = \int_{-1}^{1} \hat{a}(\mu) d\mu = \frac{2}{\eta} I_{12}(\omega) B + I_2(\omega) A, \end{cases} \tag{B.8}$$

$$\begin{cases} C_1(\omega)\gamma_G + C_{12}(\omega)\gamma_F = c\big(\hat{B} - B\big), \\ -2C_{12}(\omega)\gamma_G + C_2(\omega)\gamma_F = c\big(\hat{A} - A\big), \end{cases} \tag{B.9}$$

$$\begin{cases}\theta B = \hat{B} + \gamma_G, \\ \theta A = \hat{A} + \gamma_F,\end{cases} \tag{B.10}$$

where

$$\begin{cases} h_1(\omega,\mu) = \dfrac{1}{1+i\omega\mu}, \qquad h_2(\omega,\mu) = \dfrac{1}{(1+2/\eta)+i\omega\mu}, \\ h_{12}(\omega,\mu) = h_1(\omega,\mu)h_2(\omega,\mu), \qquad h_3(\omega,\mu) = 1 + \dfrac{2}{\eta^2}h_{12}(\omega,\mu), \end{cases} \tag{B.11}$$

$$\begin{cases} I_1(\omega) = \dfrac{c}{2}\displaystyle\int_{-1}^{1}\dfrac{h_1(\omega,\mu)}{h_3(\omega,\mu)}d\mu\,, \qquad I_2(\omega) = \dfrac{c}{2}\displaystyle\int_{-1}^{1}\dfrac{h_2(\omega,\mu)}{h_3(\omega,\mu)}d\mu\,, \\ I_{12}(\omega) = \dfrac{c}{2}\displaystyle\int_{-1}^{1}\dfrac{h_{12}(\omega,\mu)}{h_3(\omega,\mu)}d\mu\,, \end{cases} \tag{B.12}$$

$$\begin{cases} C_1(\omega) = \tilde{C} + (1-c), \qquad C_2(\omega) = \hat{C} + \left(1 - c + \dfrac{2}{\eta}\right), \qquad C_{12}(\omega) = -C^* + \dfrac{1}{\eta}, \\ \tilde{C} = (\eta+2)C^*, \qquad \hat{C}(x) = \eta C^*, \qquad C^* = \dfrac{\omega^2}{3\left(\eta + 2 + \dfrac{2}{\eta}\right)}. \end{cases} \tag{B.13}$$

Introducing Eq. (B.8) into Eq. (B.9), we obtain:

$$\begin{cases}\gamma_G = H_1(\omega)B - H_2(\omega)A, \\ \gamma_F = H_3(\omega)B + H_4(\omega)A,\end{cases} \tag{B.14}$$

where

$$H_1(\omega) = c\,\frac{I_1(\omega) - 1 - \dfrac{2}{\eta}\dfrac{C_2(\omega)}{C_3(\omega)}I_{12}(\omega)}{\hat{H}_1(\omega)}, \tag{B.15}$$

$$H_2(\omega) = c\,\frac{\dfrac{1}{\eta}I_{12}(\omega) + \dfrac{C_2(\omega)}{C_3(\omega)}[I_2(\omega) - 1]}{\hat{H}_1(\omega)}, \tag{B.16}$$

$$H_3(\omega) = c\frac{\frac{2}{\eta}I_{12}(\omega) + \frac{C_4(\omega)}{C_1(\omega)}[I_1(\omega) - 1]}{\widehat{H}_2(\omega)}, \tag{B.17}$$

$$H_4(\omega) = c\frac{I_2(\omega) - 1 - \frac{1}{\eta}\frac{C_4(\omega)}{C_1(\omega)}I_{12}(\omega)}{\widehat{H}_2(\omega)}, \tag{B.18}$$

$$\widehat{H}_1(\omega) = C_1(\omega) + \frac{C_2(\omega)C_4(\omega)}{C_3(\omega)}, \qquad \widehat{H}_2(\omega) = C_3(\omega) + \frac{C_2(\omega)C_4(\omega)}{C_1(\omega)}. \tag{B.19}$$

Finally, introducing Eqs. (B.14) and (B.8) into Eq. (B.10), and then rearranging in terms of the eigenvalue θ, we obtain:

$$\theta^2 + \tilde{b}(\omega)\theta + \tilde{c}(\omega) = 0, \tag{B.20}$$

where,

$$\tilde{b}(\omega) = -[I_2(\lambda,\eta) + H_3(\lambda,\eta,c) + I_1(\omega) + H_1(\omega)], \tag{B.21}$$

$$\tilde{c}(\omega) = [I_2(\omega) + H_3(\omega)][I_1(\omega) + H_1(\omega)] + \left[\frac{1}{\eta}I_{12}(\omega) + H_2(\omega)\right]\left[\frac{2}{\eta}I_{12}(\omega) + H_4(\omega)\right]. \tag{B.22}$$

Therefore, the eigenvalue of MBTD-DSA can be calculated as:

$$\theta_{\text{MBTD-DSA}}(\omega) = \frac{-\tilde{b}(\omega) \pm \sqrt{\tilde{b}^2(\omega) - 4\tilde{c}(\omega)}}{2}. \tag{B.23}$$

BIBLIOGRAPHY

[1] U.S. NRC, *Nuclear Reactors*, https://www.nrc.gov/reactors.html (10/18/2020).

[2] J. J. Duderstadt and L. J. Hamilton, *Nuclear Reactor Analysis*, Wiley New York, 1976.

[3] R. J. LeVeque, *Top Ten Reasons to Not Share Your Code (and why you should anyway)*, https://faculty.washington.edu/rjl/pubs/topten/topten.pdf (1/6/2021).

[4] A. Hsieh, G. Zhang, and W. S. Yang, "Consistent Transport Transient Solvers of the High-Fidelity Transport Code Proteus-MOC", *Nuclear Science and Engineering*, 2020. DOI: https://doi.org/10.1080/00295639.2020.1746619

[5] A. Zhu, *Transient Methods for Pin-Resolved Whole-Core Neutron Transport*, PhD Dissertation, University of Michigan, 2016.

[6] R. J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*, SIAM, 2007.

[7] S. MacNamara and G. Strang, "Operator Splitting", in R. Glowinski, S. J. Osher, and W. Yin (eds.), *Splitting Methods in Communication, Imaging, Science, and Engineering*, Springer, 2016.

[8] T. J. Downar, Y. Xu, and V. Seker, *PARCS v3.0 U.S. NRC Core Neutronics Simulator – Theory Manual*. RES/U.S. NRC, Rockville, Md, 2012.

[9] J. E. Morel and E. W. Larsen, "A Multiple Balance Approach for Differencing the S_N Equations", *Nuclear Science and Engineering*, **105**, pp. 1–15, 1990. DOI: https://doi.org/10.13182/NSE90-A19208

[10] J. J. Honrubia and J. E. Morel, "A Synthetically Accelerated Weighted Diamond Scheme for Charged-Particle Transport Calculations", *Nuclear Science and Engineering*, **104:2**, pp. 91-111, 1990. DOI: https://doi.org/10.13182/NSE90-A23707

[11] B. C. Yee and E. W. Larsen, "A New Multiple Balance Method for Spatially Differencing the S_N Equations," *Proc. M&C 2015, Nashville, USA*, April 19–23, 2015.

[12] E. W. Larsen, *Lecture Notes for NERS 590 Winter 2019: Numerical Transport Theory*.

[13] I. Variansyah, E. W. Larsen, and W. R. Martin, “A Robust Second-Order Multiple Balance Method for Time-Dependent Neutron Transport Simulations,” *Proc. PHYSOR 2020, Cambridge, United Kingdom*, March 29–April 2, 2020.

[14] D. E. Cullen, “TART 2016: An Overview of a Coupled Neutron Photon 3-D, Combinatorial Geometry Time Dependent Monte Carlo Transport Code”, *Technical Report: LLNL SM 704560*, 2016.

[15] V. Valtavirta, M. Hessan, and J. Leppänen, “Delayed Neutron Emission Model for Time Dependent Simulations with the Serpent 2 Monte Carlo Code First Results”, *Proc. PHYSOR 2016, Sun Valley, United States*, May 1–5, 2016.

[16] T. Trahan, “An Overview of the Monte Carlo Application ToolKit (MCATK)”, *Technical Report: LA UR 16 20076*, 2016.

[17] M. Faucher, D. Mancusi, and A. Zoia, “New Kinetic Simulation Capabilities for TRIPOLI-4®: Methods and Applications”, *Annals of Nuclear Energy*, **120**, pp. 74–88, 2018. DOI: https://doi.org/10.1016/j.anucene.2018.05.030

[18] N. Shaukat, M. Ryu, and H. J. Shim, “Dynamic Monte Carlo Transient Analysis for the Organization for Economic Co-operation and Development Nuclear Energy Agency (OECD/NEA) C5G7-TD Benchmark, *Nuclear Engineering and Technology*, **49**, pp. 920–927, 2017. DOI: https://doi.org/10.1016/j.net.2017.04.008

[19] Y. Nauchi, “Estimation of Time-Dependent Neutron Transport from Point Source based on Monte Carlo Power Iteration”, *Journal of Nuclear Science and Technology*, **56:11**, pp. 996–1005, 2019. DOI: https://doi.org/10.1080/00223131.2019.1633969

[20] W. Boyd, A. Nelson, P. K. Romano, S. Shaner, B. Forget, and K. Smith, “Multigroup Cross-Section Generation with the OpenMC Monte Carlo Particle Transport Code”, *Nuclear Technology*, **205:7**, pp. 928–944, 2019. DOI: https://doi.org/10.1080/00295450.2019.1571828

[21] J. Leppänen, M. Pusa, T. Viitanen, V. Valtavirta, and T. Kaltiaisenaho, “The Serpent Monte Carlo code: Status, Development and Applications in 2013”, *Annals of Nuclear Energy*, **82**, pp. 142–150, 2015. DOI: https://doi.org/10.1016/j.anucene.2014.08.024

[22] R. G. McClarren, “Calculating Time Eigenvalues of the Neutron Transport Equation with Dynamic Mode Decomposition”, *Nuclear Science and Engineering*, **193:8**, pp. 854–867, 2019. DOI: https://doi.org/10.1080/00295639.2018.1565014

[23] G. Velarde, C. Ahnert, and J. M. Aragonés, “Analysis of the Eigenvalue Equations in k, λ, γ, and α Applied to Some Fast- and Thermal-Neutron Systems”, *Nuclear Science and Engineering*, **66:3**, pp. 284–294, 1978. DOI: https://doi.org/10.13182/NSE78-A27213

[24] D. E. Cullen, C. J. Clouse, R. Procassini, and R. C. Little, “Static and Dynamic Criticality: Are They Different?”, *Technical Report: UCRL-TR-201506*, 2003.

[25] I. Variansyah, B. R. Betzler, and W. R. Martin, “α-Weighted Transition Rate Matrix Method”, *Proc. M&C 2019, Portland, USA*, August 25–29, 2019.

[26] V. Vitali, F. Chevallier, A. Jinaphanh, P. Blaise, and A. Zoia, “Spectral Analysis by Direct and Adjoint Monte Carlo Methods”, *Annals of Nuclear Energy*, 2019. DOI: https://doi.org/10.1016/j.anucene.2019.107033

[27] B. R. Betzler, *Calculating Alpha Eigenvalues and Eigenfunctions with a Markov Transition Rate Matrix Monte Carlo Method*, Ph.D. Dissertation, University of Michigan, 2014.

[28] B. R. Betzler, W. R. Martin, B. C. Kiedrowski, and F. B. Brown, “Calculating α Eigenvalues of One-Dimensional Media with Monte Carlo”, *Journal of Computational and Theoretical Transport*, **43:1–7**, pp. 38–49, 2014. DOI: https://doi.org/10.1080/00411450.2014.909851

[29] B. R. Betzler, B. C. Kiedrowski, F. B. Brown, and W. R. Martin, “Calculating Infinite-Medium α-Eigenvalue Spectra with Monte Carlo Using a Transition Rate Matrix Method”, *Nuclear Engineering and Design*, **295**, pp. 639–644, 2015. DOI: https://doi.org/10.1016/j.nucengdes.2015.07.052

[30] B. R. Betzler, B. C. Kiedrowski, W. R. Martin, and F. B. Brown, “Calculating Alpha Eigenvalues and Eigenfunctions with a Markov Transition Rate Matrix Monte Carlo Method”, *Nuclear Science and Engineering*, **192:2**, pp. 115–152, 2018. DOI: https://doi.org/10.1080/00295639.2018.1497397

[31] T. R. Hill, “Efficient Methods for Time-Absorption (α) Eigenvalue Calculations”, *Technical Report: LA-UR-82-3594*, 1983.

[32] D. Brockway, P. Soran, and P. Whalen, “Monte Carlo α Calculation”, *Technical Report: LA-UR-85-1224*, 1985.

[33] T. Yamamoto and Y. Miyoshi, “An Algorithm of α- and γ-Mode Eigenvalue Calculations by Monte Carlo Method”, *Proc. JAERI Conference*, 2003.

[34] T. Yamamoto, “Higher Order α Mode Eigenvalue Calculation by Monte Carlo Power Iteration”, *Nuclear Science and Technology*, **2**, pp. 826–835, 2011. DOI: https://doi.org/10.15669/PNST.2.826

[35] A. Zoia, E. Brun, and F. Malvagi, “Alpha Eigenvalue Calculations with TRIPOLI-4®”, *Annals of Nuclear Energy*, **63**, pp. 276–284, 2014. DOI: https://doi.org/10.1016/j.anucene.2013.07.018

[36] H. J. Shim, S. H. Jang, and S. M. Kang, “Monte Carlo Alpha Iteration Algorithm for a Subcritical System Analysis”, *Science and Technology of Nuclear Installations*, 2015. DOI: https://doi.org/10.1155/2015/859242

[37] A. Zoia, E. Brun, F. Damian, and F. Malvagi, “Monte Carlo Methods for Reactor Period Calculations”, *Annals of Nuclear Energy*, **75**, pp. 627–634, 2015. DOI: https://doi.org/10.1016/j.anucene.2014.09.014

[38] C. Josey and F. B. Brown, “A New Monte Carlo Alpha-Eigenvalue Estimator with Delayed Neutron”, *Transaction of American Nuclear Society*, **118**, pp. 903–906, 2018.

[39] H. A. Kia, M. Zangian, A. Minuchehr, and A. Zolfaghari, “An Improved Convergence Rate for the Prompt α Eigenvalue Calculation in α-k iteration methods”, *Annals of Nuclear Energy*, **118**, pp. 15–25, 2018. DOI: https://doi.org/10.1016/j.anucene.2018.03.039

[40] I. Variansyah, B. R. Betzler, and W. R. Martin, “Multigroup Constant Calculation with Static α-Eigenvalue Monte Carlo for Time-Dependent Neutron Transport Simulations”, *Nuclear Science and Engineering*, **194:11**, pp. 1025–1043, 2020. DOI: https://doi.org/10.1080/00295639.2020.1743578

[41] Jarrod D. Edwards, Jim E. Morel, Dana A. Knoll, “Nonlinear Variants of the TR-BDF2 Method for Thermal Radiative Diffusion”, *Journal of Computational Physics*, **230**, pp. 1198–1214, 2011. DOI: https://doi.org/10.1016/j.jcp.2010.10.035

[42] E. Magali, *Derivation of Multigroup Diffusion Coefficients That Preserve Physical Properties of the Transport Solution*, Master’s Thesis, University of Michigan, 2018.

[43] R. E. Alcouffe, “Diffusion Synthetic Acceleration Methods for the Diamond-Differenced Discrete-Ordinates Equations”, *Nuclear Science and Engineering*, **64:2**, pp. 344–355, 1977. DOI: https://doi.org/10.13182/NSE77-1

[44] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, 2003.

[45] K. O. Ott and R. J. Neuhold, *Introductory Nuclear Reactor Dynamics*, American Nuclear Society, 1985.

[46] G. Strang, *Linear Algebra and its Applications*, Harcourt, 1988.

[47] P. Virtanen, R. Gommers, T.E. Oliphant, et al., “SciPy 1.0: fundamental algorithms for scientific computing in Python”, *Nature Methods*, **17**, pp. 261–272, 2020. DOI: https://doi.org/10.1038/s41592-019-0686-2

[48] E. W. Larsen, “Unconditionally Stable Diffusion-Synthetic Acceleration Methods for the Slab Geometry Discrete Ordinates Equations. Part1: Theory,”, *Nuclear Science and Engineering*, **82**, pp. 47–63, 1982. DOI: https://doi.org/10.13182/NSE82-1

[49] A. Toth et al., “Analysis of Anderson Acceleration on a Simplified Neutronics/Thermal Hydraulics System,” *Proc. M&C 2015, Nashville, USA*, April 19–23, 2015.

[50] C. Bell, et al. (2016-2020). Thermo: Chemical properties component of Chemical Engineering Design Library (ChEDL) https://github.com/CalebBell/thermo.

[51] P. K. Romano, N. E. Horelik, B. R. Herman, A. G. Nelson, B. Forget, and K. Smith, “OpenMC: A State-of-the-art Monte Carlo Code for Research and Development”, *Annals of Nuclear Energy*, **82**, pp. 90–97, 2015. DOI: https://doi.org/10.1016/j.anucene.2014.07.048

[52] T. Ye, C. Chen, W. Sun, B. Zhang, and D. Tian, “Prompt Time Constants of a Reflected Reactor”, *Proceedings of Symposium of Nuclear Data 2006*, Tokai-mura, Japan, 2006.

[53] N. Terranova and A. Zoia, “Generalized Iterated Fission Probability for Monte Carlo Eigenvalue Calculations”, *Annals of Nuclear Energy*, **108**, 57–66, 2017. DOI: https://doi.org10.1016/j.anucene.2017.04.014

[54] A. Gerlach, *Adaptive Time Stepping for Transport Solution with the α-Eigenvalue*, Ph.D. Dissertation, University of Michigan, 2020.