The Advancement of Stable, Efficient and Parallel Acceleration Methods for the Neutron Transport Equation

Wesley Ford

université PARIS-SACLAY

NNT : 2019SACLX105

Thèse de doctorat

The Advancement of Stable, Efficient and Parallel Acceleration Methods for the Neutron Transport Equation

Thèse de doctorat de l'Université Paris-Saclay
préparée à l'Ecole Polytechnique

École doctorale n°573 Interfaces (INTERFACES)
Spécialité de doctorat : Ingénierie des systèmes complexes

Thèse présentée et soutenue à Saclay, le 08/11/2019, par

WESLEY FORD

Composition du Jury :

Grégoire Allaire Professeur, École Polytechnique	Président
Alain Hébert Professeur, Polytechnique Montréal	Rapporteur
Richard Sanchez Professeur, Seoul National University	Rapporteur
Patrick Ciarlet Professeur, ENSTA Paris	Examinateur
Olga Mula Maitre de Conference, Professeur assistant, Paris Dauphine University	Examinateur
Christophe Calvin Directeur de Recherche, CEA Saclay	Directeur de thèse
Emiliano Masiello Ingénieur de Recherche, CEA Saclay	Examinateur
François Févotte Ingénieur de Recherche, EDF	Examinateur

Abstract

In this paper a new library of non-linear techniques for accelerating the discrete-ordinates transport equation is proposed. Two new types of nonlinear acceleration methods called the Spatially Variant Rebalancing Method (SVRM) and the Response Matrix Acceleration (RMA), respectively, are introduced and investigated. The first method, SVRM, is based on the computation of the zeroth and first order spatial variation of the neutron balance equation. RMA, is a DP0 method that uses knowledge of the transport operator to form a consistent relationship. Two distinct variants of RMA, called Explicit-RMA (E-RMA) and Balance (B-RMA), respectively, are derived. The convergence properties of CMFD, SVRM and RMA are investigated for two different iteration schemes of the method of characteristics (MOC) transport operator for a 1D slab, using spectral and Fourier analysis. Based off the results of the 1D comparison, only RMA and CMFD were implemented in the library. The performance of RMA is compared to CMFD using the C5G7 and the ZPPR 3D benchmarks. Both parallel and sequential solving schemes are considered. Analysis of the results indicates that both variants of RMA have improved effectiveness and stability relative to CMFD, for optically diffusive materials. Moreover, RMA shows great improvement in stability and effectiveness when the geometry is spatially decomposed. To achieve optimal numerical performance, a combination of RMA and CMFD is suggested. Further investigation into the use and improvement of RMA is proposed. As well, many ideas for extending the features of the library are presented.

Résumé

Les méthodes déterministes sont couramment utilisées dans le domaine des neutroniques computationnels pour permettre aux chercheurs de mieux comprendre le comportement des neutrons dans un cœur de réacteur. Cependant, les méthodes déterministes telles que la méthode de transport coordonnée discrète, souffrent d'une convergence lente lorsqu'elles sont résolues de manière itérative. Afin de résoudre cette convergence lente, des méthodes connues sous le nom de méthodes d'accélération sont généralement appliquées à l'équation itérative du transport. Les méthodes d'accélération fonctionnent en isolant et en résolvant les eigenvectors convergents les plus lents de l'équation de transport, puis en utilisant la solution pour mettre à jour les inconnues de transport. Alors que les méthodes d'accélération publiées précédemment se sont avérées efficaces pour accélérer l'équation de transport lorsque l'épaisseur optique d'une région spatiale reste avec certaines limites ; les méthodes populaires telles que le CMFD peuvent devenir instables (ne convergeront pas vers une solution) lorsque ces limites sont instables. Ainsi, il y a un intérêt continu pour le développement de nouvelles méthodes d'accélération avec une stabilité et une efficacité améliorées.

Dans ce document, une nouvelle bibliothèque de techniques non linéaires pour accélérer l'équation de transport discrète est proposée. Deux nouveaux types de méthodes d'accélération non linéaire appelées méthode de rééquilibrage des variantes spatiales (SVRM) et d'accélération de la matrice de réponse (RMA), respectivement, sont introduits et étudiés. La première méthode, SVRM, est basée sur le calcul de la variation spatiale zéro et premier ordre de l'équation de l'équilibre neutronique. Lorsque la variation spatiale de premier ordre du flux n'est pas fournie par l'équation de transport, la variation spatiale de premier ordre est approximative à l'aide d'une représentation parabolique, du flux d'interface et du flux moyen. RMA

est une méthode DP0 qui utilise la connaissance de l'opérateur de transport pour former une relation cohérente. À partir de l'équation de transport, RMA définit le flux angulaire comme étant proportionnel au courant partiel, et les moments angulaires d'ordre plus élevé de la source comme étant proportionnels à la source isotrope. Deux variantes distinctes de RMA, appelées Explicit-RMA (E-RMA) et Balance (B-RMA), respectivement, sont dérivées. Les propriétés de convergence de CMFD, SVRM et RMA sont étudiées pour deux schémas d'itération différents de l'opérateur de transport de la méthode des caractéristiques (MOC) pour une dalle 1D, en utilisant l'analyse spectrale et fourier. D'après les résultats de la comparaison 1D, SVRM s'est avéré moins efficace que RMA et de souffrir d'instabilité lorsque l'épaisseur optique a été augmentée au-delà d'un seuil de 10mfp. Il a été démontré que l'instabilité de SVRM résultait de l'utilisation de la représentation parabolique pour approximer la répartition spatiale du flux. Ainsi, seuls rMA et CMFD ont été mis en œuvre dans la bibliothèque pour l'analyse comparative numérique.

La performance de La RMA a été comparée à la CMFD à l'aide des repères C5G7 et ZPPR 3D. L'analyse des résultats a indiqué que les deux variantes de RMA ont amélioré l'efficacité et la stabilité par rapport au CMFD, quand la géométrie est optiquement diffusive. En outre, RMA montre une grande amélioration de la stabilité et l'efficacité lorsque la géométrie est spatialement décomposée pour le calcul parallèle. Cependant, le coût de calcul de la résolution de RMA a été sensiblement augmenté par rapport à la résolution CMFD. Pour obtenir des performances numériques optimales, une combinaison de RMA et CMFD a été testée et a montré qu'elle offre des améliorations significatives de performance par rapport à l'utilisation de RMA ou cmFD séparément. Les deux synchrones et asynchrones itératifs schémas de résolution parallèle s'ils sont comparés à l'aide d'un test de mise à l'échelle forte. Le schéma asynchrone a été trouvé pour offrir la plus grande accélération dans le temps lorsque le nombre de noyaux a été augmenté vers 1000.

Sur la base des résultats recueillis tout au long de la thèse, les auteurs concluent que rMA devrait être sérieusement considéré comme une technique d'accélération complémentaire pour CMFD. Une enquête plus approfondie sur l'utilisation et l'amélioration de la RMA est proposée. Plus précisément, les auteurs suggèrent que le futur chercheur cherche à améliorer l'efficacité de la RMA en utilisant une relation non linéaire pour tenir compte de la dépendance de la proportionnalité entre le flux angulaire et le courant partiel sur la source. En raison de l'efficacité de RMA à résoudre des géométries optiquement épaisses, l'étude de RMA en tant que solveur pour les géométries à pleine échelle de noyau est également proposée. De plus, de nombreuses idées pour étendre les caractéristiques de la bibliothèque sont présentées.

Acknowledgements

I would like to thank my encadrent, Emiliano Masiello, for his guidance and support, especially when it came to understanding the mathematical framework in which this project was conducted. More importantly, I would like to thank him for shouldering the massive burden of editing my writing. I also wish to thank my PhD director Christophe Calvin for supporting this project and for his valuable advice. François Févotte and Bruno Lathuilière, also deserve my thanks for assisting with the editing of my writing and for managing the collaboration with EDF. As well, I would like to thank Richard Sanchez and Alain Hébert for agreeing to be the reviewers of my manuscript. I would also like to thank my office mate, Thomas Freiman for supporting me professionally and personally, as well as for his help with all things administrative. Although it was a shared burden, I would particularly like to thank Margaux Faucher for her free French tutoring services, and for her positive reinforcement. To the PhD students of the SERMA, I would like to say thank you very much for being such an inclusive group and making this country feel like home. Finally, I would like to thank the Commissariat a l'Energie Atomique (CEA) and Electricite de France (EDF) for support and funding this project.

Contents

Chapter 1

Introduction

As public concern for climate change continues to mount, many governments and industries are looking for ways to significantly reduce their greenhouse emissions. Notably, on the 22nd of April, 2016, 174 countries and the European Union, signed on to the Paris agreement, which pledged at minimum to keep the global temperature rise of this century below 2 degrees Celsius above pre-industrial levels, with the more ambitious goal of limiting the temperature increase to 1.5 degrees Celsius, [35]. In order to meet these goals, a combination of measures aimed at reducing emissions and carbon capture are required. The optimal composition of the combination of measures is a contentious debate among both the scientific community and the greater public. However, in every scenario that the intergovernmental panel on climate change considered to realize these goals, an increase of nuclear capacity between 98-501% by the year 2050 was required, [17]. The IPCC is not alone in this assessment, other prominent energy sector scientific groups, such as the IAEA have reached similar conclusions. While there are some who disagree with the conclusions of these studies, what is undeniable is that nuclear technology has the opportunity to be an essential part of the fight against climate change.

The fundamental purpose of a nuclear reactor is to convert the energy released from fission into electricity. Fission occurs when a heavy nucleus has been spatially deformed, to the point that the internal, repulsive, electrostatic forces overwhelm the strong forces holding the nucleons together. This causes the atom to split into two daughter atoms, releasing neutrons and photons. The initial deformation can be caused by the random movements of the nucleons within the nucleus, in which case the reaction is referred to as spontaneous fission. In a nuclear reactor, fission is predominately induced by heavy nuclei absorbing a neutron. The neutrons produced from fission typically have a kinetic energy of ~ 2 MeV, []. One of the results of quantum mechanics is that, slow moving neutrons typically have a higher probability of being absorbed by a nucleus than fast moving neutrons, [5]. In a thermal reactor, the fast neutrons produced by fission are slowed down by a series of scattering collisions. If the neutron is not lost (captured or has escaped the reactor) it will eventually reach thermal equilibrium with its surroundings, in which case it is referred to as a thermal neutron. In particular, the typical life cycle of a neutron within a Light Water Reactor (LWR) can be summarized to be as follows. Starting from the neutron's birth as a result of fission, the neutron travels from the nuclear fuel to the light water coolant. There, the neutron's energy is moderated by a series of scattering collisions, until the neutron has been thermalized. The neutron then returns to the fuel, causing a fission reaction. At any point during this cycle, the neutron can be lost due to radiative capture,

or as a result of the neutron escaping the geometry.

The positive feedback mechanism, by which neutrons produced as a result of fission, go on to induce fission, producing more neutrons is called the nuclear chain reaction. The neutron population will exponentially expand or decay, depending on the ratio that neutrons produced by fission will cause a fission reaction relative to those that are lost. If the ratio is greater, or less than one, then the neutron population is said to be supercritical and subcritical, respectively, [25]. If the ratio is exactly one than the neutron population is said to be critical. The lifetime of a neutron produced directly by fission, called a prompt neutron, is usually on the order of 10^{-4} seconds. Thus, if the neutron population were solely made up of prompt neutrons, it would vary far too quickly for it to be controlled. Fortunately, the daughter atoms also emit neutrons as they decay to a more stable energy state. These delayed neutrons appear milliseconds to minutes after the original fission event. As long as the prompt portion of the neutron population is not able to achieve criticality by itself, a state which is called prompt critical, then the rate of change of the neutron population is limited by the delayed neutron population. This phenomenon is what allows us to control the neutron population, making nuclear power possible. Ensuring that the neutron population does not reach a prompt critical state is one of the most important safety constraints of nuclear power.

When fission occurs, $\sim$ 200 MeV of the starting atom's binding energy is converted into the kinetic energy of the fission products, [5]. In particular, the kinetic energy of the fission fragments, prompt neutrons and gamma rays respectively make up $\sim 85\%$, $\sim 2.5\%$ and $\sim 3.5\%$ of the the released potential energy. The remainder of the binding energy appears as the kinetic energy of the delayed neutrons and gamma rays emitted from the decay of the fission fragments. This kinetic energy is rapidly transformed into thermal energy as the fission products collide with the surrounding atoms. The thermal energy is then carried away by the coolant where it is later used to drive a turbine and generate electricity. The exact sequence of steps that leads to the transformation of the thermal energy to electricity is dependent on the thermodynamic cycle employed, which is in turn dependent on the reactor core chosen. For instance, a boiling water reactor transforms the water coolant from a liquid to a gaseous state within the core and therefore has no need for the steam generator required by a pressure water reactor. Currently operating nuclear reactors have a thermodynamic efficiency of roughly 30%. Some new nuclear designs are predicted to have thermodynamic efficiencies around 50%, [20], this represents almost a doubling in the thermal efficiency. Such improvements to the thermal efficiency are achieved by increasing the temperature and pressure of the coolant exiting the core and by using new materials to construct the core. However, implementing these large changes to the reactor design significantly affects the neutron behaviour with in the core. Thus, it is vital to be able to better predict the neutron behavior in scenarios beyond our current scope of knowledge, in order to improve the performance of nuclear power.

Both experiments and computer models are used to predict neutron behavior within a reactor. Experiments provide the most reliable information about the behavior of the neutron population and thus are an important part of developing nuclear technology. However, experiments are very expensive to perform, and not all cases can be studied experimentally. This is why computer modeling is of growing importance for the development of nuclear technology. To computationally model neutron behavior within a

reactor, a specialized form of the Boltzmann equation, referred to as the neutron transport equation, is used. At present, there are two categories of techniques for solving the neutron transport equation, Monte Carlo and deterministic. Monte Carlo seeks to solve the continuous neutron transport equation by randomly sampling the solution space until a distribution with the desired statistical uncertainty is achieved. In contrast, deterministic methods seek to directly solve a discretized form of the neutron transport equation, [25]. Since Monte Carlo methods solve the continuous form of the neutron transport equation, they can achieve more accurate results than deterministic methods. However, the computational cost of solving the solution distribution with a high degree of precision, is generally far greater with Monte Carlo methods than with deterministic methods. Thus, deterministic methods are more commonly used when a result is required in a short time frame. In this manuscript, we will only consider deterministic methods.

The greatest number of discontinuities in the properties of the geometry and subsequently the flux, are encountered along the energy and spatial domain. As a result, the majority of the computational time is spent solving the energy or the spatial domain. During this project, we have chosen to focus on problems when the spatial domain dominates the time to convergence. The advancement of spatial discretization techniques for the transport equation is a very active field of research. Recent gains in parallel computing power have led more researchers to turn to easily parallelizable spatial discretization schemes, which more accurately reproduce the continuous neutron transport equation. The most prominent among these techniques is the Method Of Characteristics (MOC), due to its ability to work with and preserve complex geometries and its parallel scalability. However, the computational cost of solving MOC alone can be prohibitive, when the problem being solve has a strong global spatial dependence. This is usually the case when the transport regime is dominated by neutron scattering, [3] or is weakly absorbing.

To alleviate this issue, acceleration methods are employed in an attempt to isolate and solve for the effects that arise from global spatial dependence. Three types of acceleration methods exist, relaxation, linear, and nonlinear methods, [3]. While relaxation methods are simple to implement, they are relatively ineffective at speeding up convergence. In contrast, linear methods are typically quite effective at speeding up convergence but have high memory requirements since they must use the same energy and spatial discretization as the transport operator. Nonlinear methods are typically able to provide effective speed up and use a separate, coarser mesh than the transport operator to reduce their memory consumption. However, nonlinear methods, such as the Coarse Mesh Finite Difference method (CMFD), can be ineffective for some cases or lead to an unstable system of equations that will never converge, [14]. For CMFD, one source of its instability comes from its spatial representation of the neutron flux being inconsistent with the transport operator.

For these reasons, the investigation of new, stable acceleration methods with more consistent performance is an active field of research. However, progress in this field is slowed by the fact that most acceleration method implementations are tied to the structure of the user's code. This lack of generality leads to a variety of problems. First, methods implemented in this way inherit the limitations of the user's code, making them less capable. Second, dependencies on the programs structure and capability limitations, can cause the implementation of the method to be incompatible with other codes, greatly limiting the potential for collaborative advancement and adoption. Third, the computational

performance of an acceleration method depends greatly on its implementation. Therefore, when comparing the performance of different acceleration methods implemented in different codes, it is difficult to accurately assess whether observed variations are due to the implementation or the properties of the acceleration methods.

It was the objective of this PhD to create and analyze a library of new, stable, efficient, parallelizable and nonlinear acceleration methods. The library has been designed to be used either purely as a library or as a toolkit (the parts of the library code are compiled with the user's code. This allows for great flexibility and optimization depending on the user's needs. In order to allow the user to easily select a variety of matrix inversion techniques and GPGPU acceleration, the PETSC library, [10], is used by the acceleration library.

Two new types of acceleration methods, called the Spatially Variant Rebalancing Method (SVRM) and the Response Matrix Acceleration method (RMA), respectively, are proposed. Both methods have been designed to complement CMFD by offering improved stability and performance in cases where CMFD is ineffective. SVRM attempts to achieve this goal by using a higher-order spatial representation of the flux than what is used by CMFD. In contrast, RMA uses knowledge of the transport operator in order to ensure that a consistent relationship with the transport operator is achieved. As well, SVRM seeks to constrain the zeroth and first order spatial moments of the balance equation, whereas RMA is a nonlinear DP0 method. Both methods are compared to CMFD using Fourier and spectral radius analysis. Due to its superior performance relative to SVRM, several variants of RMA, were implemented in the library. The code IDT, developed as a solver for APOLLO3 by the Laboratoire de Transport Stochastique et Deterministe (LTSD) at CEA Saclay, was chosen as the neutron transport solver that the acceleration library would be linked to. This work was carried out at CEA Saclay in the DEN/D2MS/SERMA/LTSD laboratory, in partnership with Electricite de France (EDF).

This manuscript has been organized into three parts. In the first part, the techniques and assumptions used to numerically model neutron behavior within a reactor will be discussed. Some of the underlying physics will also be discussed in this section to justify the decisions and the assumptions being made. The second part of this manuscript contains the derivation and theoretical analysis of the acceleration method developed during this PhD. The final part is devoted to detailing the capabilities of the created library, along with a large scale, parallel, numerical benchmark of the library, for all the contained acceleration methods.

Chapter 2

Background

In this chapter we will briefly discuss the background theory which will be referenced throughout the manuscript. The contents of this chapter are as follows, starting from the Boltzmann neutron transport equation, the Method Of Characteristics (MOC) transport equation is derived. Next, the application of this transport equation for various iteration schemes and the resulting convergence properties are discussed. Finally, the purpose, classification and current limitations of acceleration methods is described.

2.1 Problem position

In the field of neutronics, there are many different reactor core behaviors that researchers are interested in investigating. Depending on the type of behavior that is to be investigated, different types of calculations are used. For instance, should one want to determine the depletion of the fuel within the reactor, then it is necessary to analyze the nonlinear relationship between the neutron population and the medium composition, over a large time scale. Such time dependent simulations are prohibitively expensive to compute unless large simplifications are made. However, researchers are most commonly interested in analyzing the behavior of the neutron population near steady state, since this is the most common case in a power reactor. Moreover, the lifetime of a neutron within a reactor is many orders of magnitude smaller than the time scale at which the composition and properties of the transited medium changes. Thus, the nonlinear dependence of the medium on the neutron population can be removed and the system can accurately be assumed to be independent of time for a time scale of the prompt neutron lifetime. Using these assumptions, the neutron population is far more computationally efficient to determine than with a transient calculation. Moreover, these time independent calculations are important for determining whether the neutron population is sub critical, or super critical and to what degree, [30]. These types of calculations are called criticality calculations. In this manuscript, only criticality calculations will be considered.

Before beginning to analyze the neutron population within a reactor, there are some observations about neutron behavior that can be made, to greatly simplify the problem. The neutron number density within an operating reactor is typically between the orders of magnitude of $10^{13}/cm^3$ and $10^{15}/cm^3$, meaning that the population is statistically significant, [25]. This allows us to treat the neutron population as a continuum, whose behavior is determined on the macroscopic scale. When a neutron encounters another neutron the probability of an interaction occurring is on a similar order of magnitude

to the probability of a neutron interacting with a nucleus. However, the typical atomic number density, $\approx 10^{23} atoms/cm^3$, is several orders of magnitude larger than that of the neutron population within a reactor. This disparity allows us to neglect neutron-neutron collisions since they will occur on such a minute scale compared to neutron-atom collisions. Although neutrons are composed of electrically charged quarks, the overall charge of the particle is zero. Moreover, even though neutrons possess a magnetic dipole, their sensitivity to external electrical magnetic fields is so weak that it is negligible within a nuclear reactor. This leads to neutrons only interacting with the nucleus of an atom and having nearly straight trajectories between collisions. Since the distance between successive neutron collisions is large compared to the size of a neutron, neutrons can be viewed as point particles that are only able to interact with one nucleus at a time.

2.2 Cross sections

We are now ready to discuss the neutron-nuclei interactions that determine neutron behavior within a nuclear core. To define the probability of such reactions occurring, it is necessary to introduce the concept of cross sections. Consider a collimated beam of neutrons of intensity I (I: the number of neutrons per unit time and surface) perpendicularly striking a uniform target of width ds, composed of a single isotope. The microscopic, partial cross section of the isotope, is defined by the following relation, [25],

$$dR(E) = M\sigma_x(E)I(E)ds. \tag{2.1a}$$

$$I(E) = N(E)u_R(E). \tag{2.1b}$$

The unknowns of Eqs. (2.1a) and (2.1b) are defined as follows:

- E: represents the kinetic energy of the incoming neutrons in the reference frame of the laboratory.
- $dR(E)$: represents the number of interactions occurring per time and unit surface and is called the surfacic reaction rate, [21].
- $N(E), M$: represent the neutron number density and the nuclei number density, respectively.
- $\sigma_x(E)$: represents the microscopic, partial, cross section for interaction x, such as absorption, fission ... etc.
- $u_R(E)$: is the relative speed of the neutron to the target atom, corresponding to the neutron energy E.

The microscopic cross section, has the units of surface area, commonly expressed in barns (b), $b = 10^{-24} cm^2$, in order to satisfy Eq. (2.1a). To understand the concept of cross sections using a more physical interpretation, we assume that the nuclei are uniformly, randomly distributed throughout the medium. In this context, the microscopic cross section can be viewed as the effective cross sectional area per nucleus, as seen by the neutron. This observation is where the name cross section comes from, [25].

The total microscopic cross section is defined as the summation of the microscopic cross section over all possible interactions,

$$\sigma(E) = \sum_{x=1}^{X} \sigma_x(E),$$

where $\sigma(E)$ and X, are the total microscopic cross section and the total number of independent interactions, respectively.

In general, M is dependent on the temperature and pressure of the medium. The temperature also determines the speed distribution of the nuclides in the medium. If we were considering the time dependent case, it would be necessary to consider that as the neutrons interact with the isotopes, the composition, temperature and pressure of the medium would vary with time. This would add several dimensions to the problem that would have to be solved. However, since only the time independent case is being considered, in which there is local thermodynamic equilibrium, it is not necessary to track these extra dimensions. This results in massive reduction in the amount of computation required to achieve a solution and is one of the primary benefits of performing this type of calculation. For deterministic codes, the effect of the speed distribution of the nuclides relative to the medium is included in the cross section through a process known as Doppler broadening. Thus, $u_R(E)$ in Eq. (2.1b), becomes $u(E)$, the speed of the neutron in the reference frame of the laboratory. Except for incoming neutrons with very low energies, the directional dependence of material properties within a reactor are insignificant. Thus, for the rest of the manuscript, all media will be assumed to be isotropic.

Combining the nuclei density and the total microscopic cross section into a single variable gives us the total macroscopic cross section, $\Sigma(E)$,

$$\Sigma(E) = M\sigma(E). \tag{2.2}$$

The total macroscopic cross section represents the probability of any neutron-nuclei interaction occurring, per path length in the medium. For mediums made of multiple isotopes, it is common practice to simply sum the total macroscopic cross sections of all the isotopes to get the total cross section, [25],

$$\Sigma(E) = \sum_{j=1}^{J} \Sigma^j(E),$$

where j is the isotope index and J is the total number of isotopes. However, this formulation neglects the different binding effects that will occur within a multi-isotope material. These effects are significant when the neutron kinetic energy is similar or less than the binding energy of the molecule, $\approx 1eV$. To alleviate this inaccuracy, cross sections of materials that have strong binding effects and are prevalent in nuclear reactors, such as water, are measured and stored.

2.3 Secondary neutron distributions

To finish defining the effects of neutron-nuclei interactions, the incident neutron properties need to be related to the angular and energy probability distributions of the outgoing neutrons, for each reaction. The normalized probability density of an incident neutron with energy E', traveling in direction $\hat{\Omega}'$, scattering to E and $\hat{\Omega}$, is defined by $p(E' \rightarrow E, \hat{\Omega}' \cdot \hat{\Omega})$. Note that as a consequence of having assumed that the medium is isotropic, $p(E' \rightarrow E, \hat{\Omega}' \cdot \hat{\Omega})$ is only dependent on the cosine of the deviation angle, $\hat{\Omega}' \cdot \hat{\Omega}$. Due to the similarity of their mathematical representation, we classify every (n, xn) reaction, where

x is the secondary neutron multiplicity of the interaction, to be a scattering interaction, [25]. It is common practice to simplify the notation by defining the composite differential scattering cross section, [25],

$$\Sigma_{sc}(E' \to E, \hat{\Omega}' \cdot \hat{\Omega}) = \Sigma_{el}(E', \hat{\Omega}' \cdot \hat{\Omega}) + \Sigma_{inel}(E' \to E, \hat{\Omega}' \cdot \hat{\Omega}) \tag{2.3}$$
$$+ \sum_{j=1}^{J} \sum_{x=2}^{X} \sum_{i=1}^{x} \Sigma^{j}_{(n,xn)}(E') p^{j}_{x}(E' \to E, \hat{\Omega}' \cdot \hat{\Omega}),$$

where $\Sigma_{sc}(E' \to E, \hat{\Omega}' \cdot \hat{\Omega})$, $\Sigma_{x_{el}}(E', \hat{\Omega}' \cdot \hat{\Omega})$, $\Sigma_{x_{inel}}(E' \to E, \hat{\Omega}' \cdot \hat{\Omega})$ and $\Sigma^{j}_{(n,xn)}(E')$ are the composite differential scattering cross section, the elastic scattering cross section, the inelastic scattering cross section and the (n,xn) interaction cross section, respectively. X is the maximum neutron multiplicity of the scattering interactions.

The fission reaction is treated differently than scattering due to the limited dependence between the properties of the incoming neutron and the secondary neutrons. This is a result of the incoming neutron energy and momentum being relatively small compared to the absolute sum of momentum and energy of the secondary particles, particularly in a thermal reactor. Subsequently, it is common to assume that the angular distribution of the secondary neutrons is isotropic, [25]. As well, the secondary neutron multiplicity $\nu(E')$ and energy distribution $\chi(E)$, are assumed to only depend on the incoming and outgoing energy, respectively. Thus, the average number of neutrons produced by fission, per unit path length $p^{j}(E, E')$, with a kinetic energy E and in the direction $\hat{\Omega}$, from an incident neutron with energy E', traveling in direction $\hat{\Omega}'$ is given by,

$$p^{j}(E, E') = \frac{\chi^{j}(E)\nu^{j}(E')\Sigma^{j}_{f}(E')}{4\pi},$$

where $\Sigma^{j}_{f}(E')$ is the macro cross section of the fission reaction for isotope j and $\nu^{j}(E')$ is the neutron yield of the interaction. Neutrons produced almost immediately ($\sim 10^{-13}s$) after the fission event are called prompt neutrons, whereas those produced by the radioactive decay of the fission fragments on a time scale of milliseconds to minutes are referred to as delayed neutrons. Due to the time difference, the contribution of prompt and delayed neutrons must be accounted for separately when the system is not at steady state. However, since the system under analysis is at steady state, the contribution of prompt neutrons and delayed neutrons, are both contained within $p^{j}(E, E')$, [25]. Contributions from other sources, such as photofission, are insignificant for most power reactors. Therefore, we will neglect their contribution in our analysis.

2.4 Neutron transport equation

To begin deriving the equations that define the neutron distribution within the core, we need to choose a spatial frame of reference to formulate the neutron transport equation. As is typically done in fluid dynamics, we will choose a Lagrangian control volume as our spatial frame of reference. This reference frame is set to follow the neutrons, as they move through the core in time. Thus, in order to map from the fixed spatial reference of the problem being solved, to the Lagrangian spatial reference frame, the spatial coordinates must be parameterized in time. In contrast, E and $\hat{\Omega}$ are not parameterized with respect to time since there are no external forces affecting the neutron trajectory. Using the

Lagrangian control volume as our spatial frame of reference, the time dependent particle balance equation, [25], is defined as,

$$\frac{d}{dt}N(t,\vec{r}(t),E,\hat{\Omega})+C(t,\vec{r}(t),E,\hat{\Omega})=Q_{sc}(t,\vec{r}(t),E,\hat{\Omega})+Q_f(t,\vec{r}(t),E,\hat{\Omega}), \tag{2.4a}$$

where

$$C(t,\vec{r}(t),E,\hat{\Omega})=\Sigma(\vec{r},E)N(t,\vec{r}(t),E,\hat{\Omega})u(E), \tag{2.4b}$$

$$Q_{sc}(t,\vec{r}(t),E,\hat{\Omega})=\int_{4\pi}d^2\hat{\Omega}'\int_0^\infty dE'\;\Sigma_{sc}(\vec{r},E'\rightarrow E,\hat{\Omega}'\cdot\hat{\Omega})N(t,\vec{r}(t),E',\hat{\Omega}')u(E'), \tag{2.4c}$$

and

$$Q_f(t,\vec{r}(t),E,\hat{\Omega})=\sum_{j=0}^{J}\left[\frac{\chi^j(\vec{r},E)}{4\pi}\int_0^\infty dE'\;\tilde{\Sigma}_f^j(\vec{r},E')\int_{4\pi}d^2\hat{\Omega}'\;N(t,\vec{r}(t),E',\hat{\Omega}')u(E')\right]. \tag{2.4d}$$

The variables of Eqs. (2.4a) to (2.4d) are defined as follows

- t: represents the time.
- $\vec{r}$: represents the spatial position vector, respectively
- $\tilde{\Sigma}_f^j(\vec{r},E)$: is simply a substitution for $\nu^j(E)\Sigma_f^j(\vec{r},E)$.
- $C(t,\vec{r}(t),E,\hat{\Omega})$: represent the interaction rate per volume, at the $(t,\vec{r}(t),E,\hat{\Omega})$ point of the phase space.
- $Q_{sc}(t,\vec{r}(t),E,\hat{\Omega})$, and $Q_f(t,\vec{r}(t),E,\hat{\Omega})$: represent the rate of secondary neutron production per volume from scattering and fission, respectively, at the $(t,\vec{r}(t),E,\hat{\Omega})$ point of the phase space.

Using the chain rule, the time derivative is expanded to get,

$$\frac{d}{dt}=\left[\frac{\partial}{\partial t}+\vec{\nabla}\cdot\hat{\Omega}u(E)\right], \tag{2.5a}$$

where

$$\hat{\Omega}u(E)=\frac{\partial\vec{r}(t)}{\partial t}. \tag{2.5b}$$

Since $\hat{\Omega}$ is an independent variable Eq. (2.5a) can be simplified to get,

$$\frac{d}{dt}=\left[\frac{\partial}{\partial t}+u(E)\hat{\Omega}\cdot\vec{\nabla}\right]. \tag{2.6}$$

Substituting Eq. (2.6) into Eq. (2.4a), gives us,

$$\begin{aligned}&\left[\frac{\partial}{\partial t}+u(E)\hat{\Omega}\cdot\vec{\nabla}\right]N(t,\vec{r}(t),E,\hat{\Omega})+C(t,\vec{r}(t),E,\hat{\Omega})\\&=Q_{sc}(t,\vec{r}(t),E,\hat{\Omega})+Q_f(t,\vec{r}(t),E,\hat{\Omega}).\end{aligned} \tag{2.7}$$

At this point, we wish to remove the time dependence by applying our assumption that the neutron population is near steady state. Since the neutron population is not actually

at steady state, the equality between the two sides of Eq. (2.7) must be synthetically preserved. Two common approaches for doing this are the k and α eigenvalue methods. The k eigenvalue method achieves balance by dividing the fission source by the k eigenvalue. In the case of the α eigenvalue method, the partial time derivative is replaced by the α eigenvalue. The k eigenvalue method is commonly preferred for many reasons. First, for sub critical systems with voided regions, the α eigenvalue term can produce a negative value when combined with the total cross section. This can cause instability for some numeric solving techniques, [25]. Second, the α eigenvalue directly biases the neutron energy distribution by absorbing and adding thermal neutrons in super and sub critical systems, respectively. In a thermal reactor, this bias artificially pushes the critically towards 1. While the k eigenvalue also biases the neutron energy distribution by dividing, or multiplying fast fission neutrons, it does so to a lesser extent. Third, the iterative schemes that can be developed from the k eigenvalue method have more desirable properties than those of the α eigenvalue method. As this is a common point of confusion, it is important to note that unlike the k eigenvalue method, the deterministic version of the α is not equivalent to that of Monte Carlo. For the Monte Carlo implementation of the α eigenvalue method, the neutron population is uniformly combed, [16]. Thus, the Monte Carlo implementation does not suffer from the same population biasing that the deterministic implementation does and is equivalent to doing a dynamic criticality calculation. For the remainder of this manuscript, we will only analyze the k eigenvalue method. Applying the k eigenvalue method to Eq. (2.7) gives us,

$$\begin{aligned}&\left[u(E)\hat{\Omega}\cdot\vec{\nabla}\right]N(\vec{r},E,\hat{\Omega})+C(\vec{r},E,\hat{\Omega})\\&=Q_{sc}(\vec{r},E,\hat{\Omega})+\frac{1}{k}Q_f(\vec{r},E,\hat{\Omega}).\end{aligned}\tag{2.8}$$

By simplifying Eq. (2.8) we get,

$$\left[\hat{\Omega}\cdot\vec{\nabla}+\Sigma(\vec{r},E)\right]\psi(\vec{r},E,\hat{\Omega})=Q_{sc}(\vec{r},E,\hat{\Omega})+\frac{1}{k}Q_f(\vec{r},E,\hat{\Omega}),\tag{2.9a}$$

where

$$\psi(\vec{r},E,\hat{\Omega})=\hat{\Omega}u(E)N(\vec{r},E,\hat{\Omega}),\tag{2.9b}$$

and

$$\phi(\vec{r},E)=\int_{4\pi}d^2\hat{\Omega}\;\psi(\vec{r},E,\hat{\Omega}).\tag{2.9c}$$

where $\psi(\vec{r},E,\hat{\Omega})$, and $\phi(\vec{r},E)$: are the angular flux and the scalar flux, respectively. Substituting the definitions of $\psi(\vec{r},E,\hat{\Omega})$, and $\phi(\vec{r},E)$ into $Q_{sc}(t,\vec{r},E,\hat{\Omega})$, and $Q_f(t,\vec{r},E,\hat{\Omega})$ we get,

$$\begin{aligned}Q_{sc}(\vec{r},E,\hat{\Omega})&=\int_{4\pi}d^2\hat{\Omega}'\int_0^\infty dE'\;\Sigma_{sc}(\vec{r},E'\rightarrow E,\hat{\Omega}'\cdot\hat{\Omega})\psi(\vec{r},E',\hat{\Omega}'),\\Q_f(\vec{r},E,\hat{\Omega})&=\sum_{j=0}^{J}\frac{\chi^j(\vec{r},E)}{4\pi}\int_0^\infty dE'\;\tilde{\Sigma}_f^j(\vec{r},E')\phi(\vec{r},E').\end{aligned}$$

2.5 Multi-group approximation

For the Boltzmann equation, Eq. (2.9a), to be numerically solvable it is necessary to discretize its phase space. Starting with the energy domain, we turn it into a finite domain by setting a maximum energy value (E_{max}). The typical value of E_{max} for reactor simulations is 20MeV since the number of neutrons produced by fission with energies greater than this limit are insignificant. As well, most neutron cross-section libraries, such as the Evaluated Nuclear Data File (ENDF), [11], do not contain tabulated data from experiments beyond this limit. Next, the continuous domain is broken up into discrete groups $g \in [1, G]$, where the energy range of group g is defined by $E \in [E_g, E_{g-1}]$, [25]. Using this indexing scheme, the integral over the energy domain can be rewritten as follows,

$$\int_0^{E_{max}} dE = \sum_{g=1}^{G} \int_{E_g}^{E_{g-1}} dE. \tag{2.10}$$

For the sake of concision, the notation of the above equation is simplified to,

$$\int_0^{E_{max}} dE = \sum_{g=1}^{G} \int_{g} dE. \tag{2.11}$$

By substituting the operator defined by Eq. (2.11) into Eq. (2.9a), we get,

$$\left[\hat{\Omega} \cdot \vec{\nabla} + \Sigma_g(\vec{r}, \hat{\Omega})\right] \psi_g(\vec{r}, \hat{\Omega}) = Q_g(\vec{r}, \hat{\Omega}), \tag{2.12a}$$

where

$$Q_g(\vec{r}, \hat{\Omega}) = \sum_{j=0}^{J} \frac{\chi_g^j(\vec{r})}{4\pi} \sum_{g'=1}^{G} \tilde{\Sigma}_{f,g'}^j(\vec{r}) \phi_{g'}(\vec{r}) + \int_{4\pi} d^2\hat{\Omega}' \sum_{g'=1}^{G} \Sigma_{sc,g,g'}(\vec{r}, \hat{\Omega}' \to \hat{\Omega}) \psi_{g'}(\vec{r}, \hat{\Omega}'). \tag{2.12b}$$

The group scalar flux and the group angular flux are defined as:

$$\phi_g(\vec{r}) = \int_{4\pi} d^2\hat{\Omega}\ \psi_g(\vec{r}, \hat{\Omega}),$$

and

$$\psi_g(\vec{r}, \hat{\Omega}) = \int_{g} dE\ \psi(\vec{r}, E, \hat{\Omega}),$$

respectively. The energy condensed cross sections, $\Sigma_g(\vec{r}, \hat{\Omega}), \tilde{\Sigma}_{f,g'}^j(\vec{r}), \Sigma_{sc,g,g'}(\vec{r}, \hat{\Omega}' \to \hat{\Omega})$ and $\chi^j(\vec{r})$ are defined as:

$$\Sigma_g(\vec{r}, \hat{\Omega}) = \frac{\int_g dE\ \Sigma(\vec{r}, E)\psi(\vec{r}, E, \hat{\Omega})}{\psi_g(\vec{r}, \hat{\Omega})}, \tag{2.13a}$$

$$\tilde{\Sigma}_{f,g}^j(\vec{r}) = \frac{\int_g dE\ \tilde{\Sigma}_f^j(\vec{r}, E)\phi(\vec{r}, E)}{\phi_g(\vec{r})}, \tag{2.13b}$$

$$\Sigma_{sc,g,g'}(\vec{r}, \hat{\Omega}' \to \hat{\Omega}) = \frac{\int_g dE \int_{g'} dE'\ \Sigma_{sc}(\vec{r}, E' \to E, \hat{\Omega}' \cdot \hat{\Omega})\psi(\vec{r}, E', \hat{\Omega}')}{\psi_g(\vec{r}, \hat{\Omega})}, \tag{2.13c}$$

and

$$\chi_g^j(\vec{r}) = \int_g dE\ \chi^j(\vec{r}, E). \tag{2.13d}$$

The multi-group formulation of Eq. (2.12a), has two undesirable properties. The first problem is that, $\Sigma_g(\vec{r}, \hat{\Omega})$ and $\Sigma_{sc,g,g'}(\vec{r}, \hat{\Omega}' \to \hat{\Omega})$ are now dependent on $\hat{\Omega}$. This added dependency would lead to a large increase in memory use and would effectively eliminate the benefits of the isotropic medium assumption we made earlier. The second problem is the dependence of Eq. (2.12a) on the flux. In order to solve for the neutron flux, it is required to already know its energy distribution. Therefore, solving Eqs. (2.13a) through (2.13c) without making any added assumptions is not practical.

A common first step towards building a low order approximation is to assume that the flux is separable in energy, [25], such that,

$$\psi(\vec{r}, E, \hat{\Omega}) = f(E)\psi(\vec{r}, \hat{\Omega}), \tag{2.14a}$$

$$\phi(\vec{r}, E) = f(E)\phi(\vec{r}), \tag{2.14b}$$

where, $f(E)$ is the approximate energy distribution of the angular flux. $f(E)$ is provided by various libraries for different reactor types. It is important to note that by using $f(E)$, we are not accounting for any angular and spatial dependence of the flux weight. As an example, if we were to substitute Eq. (2.14a) into the total multigroup group cross section defined by Eq. (2.13a) we would get,

$$\Sigma_g(\vec{r}, \hat{\Omega}) = \frac{\psi(\vec{r}, \hat{\Omega}) \int_g dE\ \Sigma(\vec{r}, E) f(E)}{\psi(\vec{r}, \hat{\Omega}) \int_g dE\ f(E)}.$$

$\psi(\vec{r}, \hat{\Omega})$ can then be eliminated in the above equation to get,

$$\Sigma_g(\vec{r}) = \frac{\int_g dE\ \Sigma(\vec{r}, E) f(E)}{\int_g dE\ f(E)}.$$

However, using the approximation defined by Eqs. (2.14a) and (2.14b) alone is not sufficient since it does not account for coupling between the spatial domain and the energy domain. To resolve this, a self-shielding calculation must be run. It is the responsibility of this calculation to define the spatial dependence of the multigroup cross sections due to the effects of self-shielding. Self-shielding effects is a blanket term for the effects arising from cross section resonance and beam hardening, [25]. Since the energy profile is of primary concern for self-shielding calculations, usually a fine energy discretization, or a continuous energy domain, is used along with a coarse approximation of the spatial properties of the problem of interest. Substituting the corrected multigroup cross sections into Eq. (2.9a) leads us to the multi-group transport equation,

$$[\hat{\Omega} \cdot \vec{\nabla} + \Sigma_g(\vec{r})]\psi_g(\vec{r}, \hat{\Omega}) = Q_g(\vec{r}, \hat{\Omega}), \tag{2.15a}$$

where

$$Q_g(\vec{r}, \hat{\Omega}) = \sum_{j=0}^{J} \frac{\chi_g^j(\vec{r})}{4\pi} \sum_{g'=1}^{G} \tilde{\Sigma}_{f,g'}^j(\vec{r})\phi_{g'}(\vec{r}) + \int_{4\pi} d^2\hat{\Omega}' \sum_{g'=1}^{G} \Sigma_{sc,g,g'}(\vec{r}, \hat{\Omega}' \cdot \hat{\Omega})\psi_{g'}(\vec{r}, \hat{\Omega}'). \tag{2.15b}$$

It is important to note that while the energy distributions of the neutron population are well understood for existing reactor designs under normal operation, this is not true for new designs, or anomalous cases. Thus, the accuracy of the multigroup approximation is particularly limited, when it comes to modelling new, or not well understood problems.

2.6 The P_N approximation of the scattering kernel

By assuming that the medium to be solved is isotropic, we were able to simplify the angular dependence of the scattering cross section to only depend on the scattering angle. However, the number of dimensions of the scattering cross section that must be stored can be further reduced by expanding the scattering cross section with Legendre polynomials along the angular space, [25]. For convenience we separate the scattering source from the rest of the Eq. (2.15b) to get,

$$Q_{sc,g}(\vec{r},\hat{\Omega}) = \int_{4\pi} d^2\hat{\Omega}' \sum_{g'=1}^{G} \Sigma_{sc,g,g'}(\vec{r},\mu)\psi_{g'}(\vec{r},\hat{\Omega}'),$$
$$\mu = \hat{\Omega}' \cdot \hat{\Omega}$$

where, $Q_{sc,g}(\vec{r},\hat{\Omega})$ represents the multigroup scattering source. Next, the Legendre expansion of the scattering cross section is taken along the angular space to get,

$$\Sigma_{sc,g,g'}(\vec{r},\mu) = \sum_{l=0}^{\infty}(2l+1)\Sigma_{sc,g,g',l}(\vec{r})P_l(\mu) \tag{2.16a}$$

$$Q_{sc,g}(\vec{r},\hat{\Omega}) = \sum_{l=0}^{\infty}(2l+1)\sum_{g'=1}^{G}\Sigma_{sc,g,g',l}(\vec{r})\int_{4\pi} d^2\hat{\Omega}'\ P_l(\mu)\psi_{g'}(\vec{r},\hat{\Omega}'), \tag{2.16b}$$

and

$$\Sigma_{sc,g,g',l}(\vec{r}) = \int_{-1}^{1}\frac{d\mu}{2}\Sigma_{sc,g,g'}(\vec{r},\mu)P_l(\mu) \tag{2.16c}$$

In the equation above, $P_l(\mu)$ is the Legendre polynomial of degree l, defined in Appendix A.1. At this point, it is common to take advantage of Legendre addition theorem in order to simplify the representation of the scattering source, [25]. Legendre addition theorem states that,

$$P_l(\mu) = \frac{1}{2l+1}\sum_{k=-l}^{l} Y_{l,k}(\hat{\Omega}')Y_{l,k}(\hat{\Omega}).$$

Legendre addition theorem can be viewed as a generalization of the more commonly known trigonometric identity,

$$cos(\omega'-\omega) = cos(\omega')cos(\omega) + sin(\omega)sin(\omega').$$

Applying the Legendre addition theorem to Eq. (2.16b) and simplifying gives us,

$$Q_{sc,g}(\vec{r},\hat{\Omega}) = \sum_{l=0}^{\infty}\sum_{k=-l}^{l} Y_{l,k}(\hat{\Omega})\sum_{g'=1}^{G}\Sigma_{sc,g,g',l}(\vec{r})\phi_{g',l,k}(\vec{r}),$$

where

$$\phi_{g',l,k}(\vec{r}) = \int_{4\pi} d^2\hat{\Omega}' \; Y_{l,k}(\hat{\Omega}')\psi_{g'}(\vec{r},\hat{\Omega}').$$

For the system to be numerically solvable, the spherical harmonic expansion is truncated, such that $l \in [0, L]$,

$$Q_{sc,g}(\vec{r},\hat{\Omega}) \approx \sum_{l=0}^{L}\sum_{k=-l}^{l} Y_{l,k}(\hat{\Omega}) \sum_{g'=1}^{G} \Sigma_{sc,g,g',l}(\vec{r})\phi_{g',l,k}(\vec{r}). \tag{2.17}$$

Finally, substituting Eq. (2.17) into Eq. (2.15b) gives us,

$$Q_g(\vec{r},\hat{\Omega}) = \sum_{j=0}^{J} \frac{\chi_g^j(\vec{r})}{4\pi} \sum_{g'=1}^{G} \tilde{\Sigma}^j_{f,g'}(\vec{r})\phi_{g'}(\vec{r}) + \sum_{l=0}^{L}\sum_{k=-l}^{l} Y_{l,k}(\hat{\Omega}) \sum_{g'=1}^{G} \Sigma_{sc,g,g',l}(\vec{r})\phi_{g',l,k}(\vec{r}). \tag{2.18}$$

2.7 Angular discretization

At present, there are two methods for discretizing the angular domain, the spherical harmonic approximation (P_N) and the discrete ordinates approximation (S_N). In both cases the N refers to the order of the approximate representation.

2.7.1 P_N Approximation

Since the scattering source has already been expanded as a series of spherical harmonic moments, it is logical to try and match the unknowns of Eq. (2.18) by expanding the angular flux as a series of spherical harmonics, [25],

$$\psi_g(\vec{r},\hat{\Omega}) = \sum_{l=0}^{\infty}\sum_{k=-l}^{l} Y_{l,k}(\hat{\Omega})\phi_{g,l,k}(\vec{r}), \tag{2.19a}$$

where

$$\phi_{g,l,k}(\vec{r}) = \int_{4\pi} d^2\hat{\Omega}' \; Y_{l,k}(\hat{\Omega}')\psi_g(\vec{r},\hat{\Omega}'). \tag{2.19b}$$

Substituting Eq. (2.19a) into Eq. (2.18) give us,

$$[\hat{\Omega}\cdot\vec{\nabla} + \Sigma_g(\vec{r})] \sum_{l=0}^{\infty}\sum_{k=-l}^{l} Y_{l,k}(\hat{\Omega})\phi_{g,l,k}(\vec{r}) = Q_g(\vec{r},\hat{\Omega}). \tag{2.20}$$

For Eq. (2.20) to be solvable, it must be transformed into a series of ordinary differential equations. This can be done by making use of the orthogonality of the spherical harmonics. By multiplying Eq. (2.20) by $Y_{l',k'}(\hat{\Omega})$ and integrating over the angular domain, we obtain,

$$\sum_{l=0}^{\infty}\sum_{k=-l}^{l} \vec{\nabla}\cdot\int_{4\pi} d^2\hat{\Omega} Y_{l',k'}(\hat{\Omega})Y_{l,k}(\hat{\Omega})\hat{\Omega}\phi_{g,l,k}(\vec{r}) + \Sigma_g(\vec{r})\phi_{g,l',k'} = Q_{g,l',k'}(\vec{r}). \tag{2.21}$$

As a result of the orthogonality condition, the summations over the angular moments have disappeared from all the terms of Eq. (2.21) except for the streaming term. While further

derivation of the P_N method is beyond the scope of this manuscript, it is important to observe that it is the streaming term that causes the coupling of the angular moments of the flux.

Just as the Fourier series expansion represents the fundamental modes of the solution over a string, spherical harmonics represent the fundamental modes of the solution over the surface of a sphere, [22]. This commonality of the two expansions leads to the P_N approximation possessing the same widely known advantages and disadvantages of a Fourier series expansion. Namely, both expansions can efficiently represent regular solutions that possess periodic symmetry over their domains but have great difficulty representing solutions that are discontinuous, or not periodic. In the case of neutron transport this means that the P_N approximation is efficient for representing solutions in optically thick, diffusive mediums. However, the P_N approximation is inefficient for representing solutions in optically thin mediums or an anisotropic angular flux distribution, [25]. Moreover, when a high order P_N approximation is used, it is necessary to store more angular moments than are needed by the scattering kernel, significantly increasing the memory requirements of the solver. The P_N approximation also has the computational disadvantage that each spherical harmonic affects the entire angular domain and subsequently, all the neighboring space. So not only does the transport matrix dimension increase quadratically with the number of angular moments, the matrix density also increases with the number of spatial dimensions. This increase in the degrees of freedom of the P_N transport equation means that the computational cost of solving it rapidly increases with the number of angular moments and spatial dimensions.

2.7.2 Discrete ordinates approximation

In order to overcome the computational drawbacks of the P_N approximation, the S_N method is often used. In particular, the S_N method is able to handle highly anisotropic angular flux distributions without requiring dramatic increases in memory. The S_N approximation works by transforming the angular domain from a continuous space to a set of discrete angles ($\hat{\Omega}_m$, $m \in [1, M]$) and then using a numerical quadrature defined by the weights (w_m, $m \in [1, M]$) to approximate the angular integrals, [25]. Substituting the discrete ordinates approximation, into Eq. (2.18) gives,

$$\left[\hat{\Omega}_m \cdot \vec{\nabla} + \Sigma_g(\vec{r})\right] \psi_{g,m}(\vec{r}) = Q_{g,m}(\vec{r}), \tag{2.22a}$$

$$Q_{g,m}(\vec{r}) = Q_g(\vec{r}, \hat{\Omega}_m), \tag{2.22b}$$

$$\phi_{g,l,k}(\vec{r}) = \sum_{m=1}^{M} w_m Y_{l,k}(\hat{\Omega}_m) \psi_{g,m}(\vec{r}), \tag{2.22c}$$

$$\psi_{g,m}(\vec{r}) = \psi_g(\vec{r}, \hat{\Omega}_m). \tag{2.22d}$$

Unlike the P_N approximation, the S_N approximation can efficiently represent solutions in optically thin mediums and angular discontinuities in the solution that occur along the boundaries of the problem. Each discrete angular flux of the S_N approximation is only dependent on the other directions by way of the scattering kernel, making the method more computationally efficient. However, because of this lack of dependence to the entire angular space, the transport equation with the S_N approximation becomes

inefficient to solve, in optically thick, diffusive mediums. As well, in mediums with low scattering ratios (c_l, where $c_l = \Sigma_{sc,g,g,l}/\Sigma_g$) and localized sources, the S_N approximation can exhibit ray effects, [25]. This is a result of large discontinuities in the flux occurring in between the set of discrete angles, for which the quadrature is unable to accurately approximate. This effect can be diminished by a variety of angularly smeared methods, or by simply increasing the order of the S_N approximation. Despite these disadvantages, the S_N approximation is the most widely used angular discretization technique for high fidelity simulations, due to its computational efficiency. Thus, it is the only technique which will be considered for the remainder of this manuscript.

2.8 Spatial discretization techniques

There are many techniques by which the spatial domain can be discretized. Some of the more well-known techniques include the diamond difference scheme, [25], and the step, linear, and bilinear nodal schemes and the finite element method. While these schemes are still widely used, the method of characteristics (MOC) has become one of the preferred techniques for accurately solving large scale problems. This is due to its ability to treat complex geometries, [29]. MOC is the only transport technique that we will describe in this manuscript. However, the nodal and diamond difference techniques will be analyzed, in relation to the new acceleration method techniques, in Section 5.

2.9 Method of characteristics

The method of characteristics allows any hyperbolic partial differential equation (PDE) to be transformed into a family of ordinary differential equations (ODE), [15]. This transformation works by finding curves, called characteristic curves, along which the PDE becomes an ODE. Starting from the boundaries of the problem's domain, the produced ordinary differential equation can be integrated along the PDE's characteristic curves and solved.

2.9.1 The characteristic transformation

Starting with the transport equation defined by Eq. (2.22a), one can observe that all three spatial dimensions are coupled as a result of the streaming term Eq. (2.22a) $\hat{\Omega}_m \cdot \vec{\nabla}$. However, the dot product can be taken advantage of by reorienting the coordinate system such that two of the dimensions are perpendicular to the direction of travel and subsequently disappear from the streaming term. In other words, when $\vec{r}$ changes along a fixed direction, $\hat{\Omega}$, the streaming term reduces to an ordinary differential, [8]. Thus, by parameterizing $\vec{r}$ in terms of the starting position on the boundary ($\vec{r}_{s'}$), we get

$$\vec{r} = \vec{r}_{s'} + b\hat{\Omega}_m. \tag{2.23}$$

Substituting Eq. (2.23) into Eq. (2.22a) gives us the ODE,

$$\left[\frac{d}{db} + \Sigma_g(\vec{r}_{s'}, b)\right] \psi_{g,m}(\vec{r}_{s'}, b) = Q_{g,m}(\vec{r}_{s'}, b). \tag{2.24}$$

Eq. (2.24) can now be solved using the integrating factor $e^{-\int_0^b db\ \Sigma_g(\vec{r}_{s'},b)}$ such that,

$$\psi_{g,m}(\vec{r}_{s'},b) = \psi_{g,m}(\vec{r}_{s'},0)e^{-\int_0^b db'\ \Sigma_g(\vec{r}_{s'},b')} + \int_0^b db'\ Q_{g,m}(\vec{r}_{s'},b')e^{-\int_{b'}^b db''\ \Sigma_g(\vec{r}_{s'},b'')}. \quad (2.25)$$

2.9.2 Computational spatial mesh

For deterministic methods it is common practise to break up the global domain into many homogeneous regions $r \in [1, R]$. Within each region a flat representation of the cross sections is employed. However, this flat representation of the cross sections can lead to grave inaccuracies in the results if it is not properly used. Recalling from Section 2.5 that the multigroup cross sections are generated using flux spectral weighting, a flat representation of a cross section within a region is only accurate if the flux energy spectrum is flat within the region as well. However, the flux is not close to being flat when the medium is strongly absorbing or exhibits significant resonance behaviour as is commonly the case for nuclear fuel, [19]. In order, to improve the accuracy of the flat cross section distribution in these medium, several regions are used to create a piece-wise constant distribution, as is shown by Fig. 2.1. However, this leads to a much more complicated geometry, which is not trivial to index.

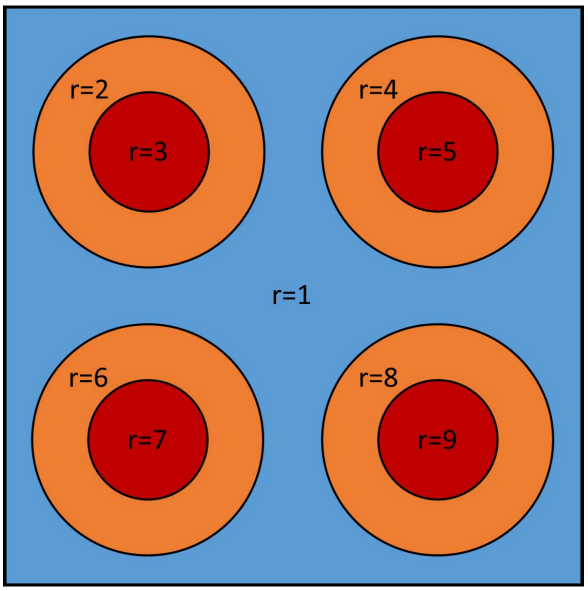

Figure 2.1 – The region indexing for typical pin cell geometry. Black lines are used for region interfaces and colour fill is used to distinguish different media.

Some programs, such as TDT, [27], use an unstructured mesh indexing scheme to map the spatial domain. The advantage of this approach is that the indexes of the map are directly associated with the discrete regions as is shown in Fig. 2.1. This allows for a lot of flexibility when it comes to the kind of geometries that can be represented. However, this unstructured approach does have a few drawbacks. First, there is not a reliable pattern between a region's index and those of its neighbours. Thus, a complex algorithm is required in order to generate the mapping. Second, the region indices are not always close to those of the adjacent regions, resulting in poor data locality.

To avoid these issues, an additional layer of spatial discretization, called the computational mesh, composed of polyhedron cells, $i \in [1, I]$, is introduced in IDT [28]. An example of the geometry shown in Fig. 2.1, with a computation mesh applied to it, is shown by Fig. 2.2. A clear pattern can be seen to exist for the cell indices. Moreover, the indices of neighbouring regions can be seen to be closer together, resulting in improved data locality. A disadvantage of this approach is that a Cartesian symmetry of the geometry is required in order for the mapping to be effective and an extra level of indexing is required. Moreover, redundancies in the indexing of the coolant can be seen to occur.

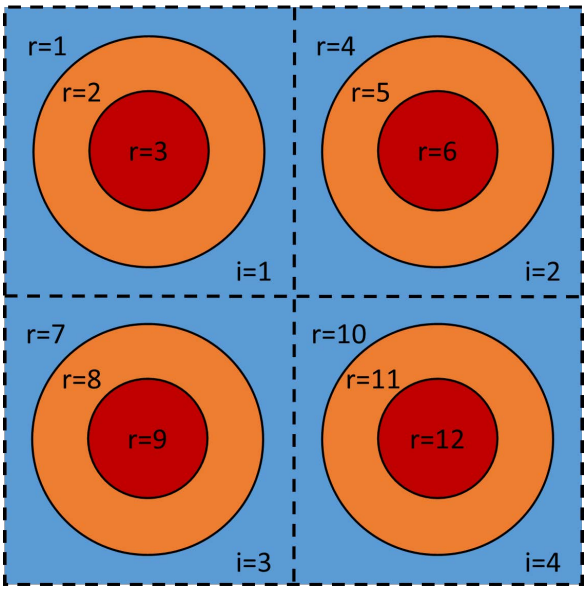

Figure 2.2 – The cell indexing for a typical pin cell geometry. Dotted lines are used for cell interfaces, solid lines are used for region interfaces and colour fill is used to distinguish different mediums.

For clarity, we will begin by defining the MOC equations for the case where each cell's medium is homogeneous, $i = r$. Then in Section 2.9.6 the short characteristic equations will be expanded for heterogeneous cells. By enforcing that the medium within each cell is homogeneous we get,

$$\begin{aligned}
\Sigma_g(\vec{r}) &= \Sigma_{g,i}, \\
\chi_g^j(\vec{r}) &= \chi_{g,i}^j, \\
\tilde{\Sigma}_{f,g}^j(\vec{r}) &= \tilde{\Sigma}_{f,g,i}^j, \\
\Sigma_{sc,g,g',l}(\vec{r}) &= \Sigma_{sc,g,g',l,i},
\end{aligned}$$

when

$$\vec{r} \in D_i,$$

where D_i is the spatial domain of cell i.

2.9.3 Step characteristics

One of the most commonly used spatial representations of the source within each cell is the constant (step) representation. This is primarily due to its simplicity and the fact that it guarantees flux positivity. By applying the step representation of the source, we get the spatial cell averaged quantity,

$$Q_{g,m,i} = \frac{\int_{D_i} d^3r \; Q_{g,m}(\vec{r})}{V_i}, \tag{2.27}$$

where V_i is the volume of cell i. Next, by applying Eq. (2.27) to Eq. (2.25), we get,

$$\psi_{g,m}(\vec{r}_{i,s'}, \tilde{b}) = \psi_{g,m}(\vec{r}_{i,s'}, 0)e^{-\Sigma_{g,i}\tilde{b}} + \frac{Q_{g,m,i}}{\Sigma_{g,i}}\left[1 - e^{-\Sigma_{g,i}\tilde{b}}\right], \tag{2.28a}$$

$$Q_{g,m,i} = \sum_{j=1}^{J} \frac{\chi^j_{g,i}}{4\pi} \sum_{g'=1}^{G} \tilde{\Sigma}^j_{f,g',i} \bar{\phi}_{g',0,0,i} + \sum_{l=0}^{L} \sum_{k=-l}^{l} Y_{l,k}(\hat{\Omega}_m) \sum_{g'=1}^{G} \Sigma_{sc,g,g',l,i} \bar{\phi}_{g',l,k,i}, \tag{2.28b}$$

and

$$\bar{\phi}_{g,l,k,i} = \sum_{m=1}^{M} w_m Y_{l,k}(\hat{\Omega}_m) \frac{\int_{\vec{r}_{i,s'} \in \Gamma_i} d^2 s_i \; \int_0^{l_{m,i}(\vec{r}_{i,s'})} d\tilde{b} \; \psi_{g,m}(\vec{r}_{i,s'}, \tilde{b})}{V_i}, \tag{2.28c}$$

where $\vec{r}_{i,s'} \in \Gamma_i$ is the starting position on the boundary of cell i, Γ_i is the vector space of the boundary of cell i, $\tilde{b} \in [0, l_{m,i}(\vec{r}_{i,s'})]$ is the distance from the cell boundary $\vec{r}_{i,s'}$ along direction m to the position of interest $\vec{r}$ and $l_{m,i}(\vec{r}_{i,s'})$ is the chord length. Solving the integral along the characteristic for Eq. (2.28c) we have,

$$\bar{\phi}_{g,l,k,i} = \sum_{m=1}^{M} \frac{w_m Y_{l,k}(\hat{\Omega}_m)}{V_i} \int_{\vec{r}_{s'} \in \Gamma_i} d^2 s_i \left\{ \frac{\psi_{g,m}(\vec{r}_{i,s'}, 0)}{\Sigma_{g,i}} \left[1 - e^{-\Sigma_{g,i} l_{m,i}(\vec{r}_{i,s'})}\right] \right. \tag{2.29}$$
$$\left. + \frac{Q_{g,m,i} l_{m,i}(\vec{r}_{i,s'})}{\Sigma_{g,i}} \left[1 - \frac{1 - e^{-\Sigma_{g,i} l_{m,i}(\vec{r}_{i,s'})}}{\Sigma_{g,i} l_{m,i}(\vec{r}_{i,s'})}\right] \right\}.$$

2.9.4 Discretization of the interface angular flux

The two common schemes for discretizing the angular flux along the boundaries of the cell are the long and short characteristics. It is important to note that typically the order of the polynomial representation is common to both the discretization of the source and the interface angular flux. Thus, if a piece-wise constant representation is used for the source, then a piece-wise constant representation is also used for the interface angular flux.

2.9.4.1 Long characteristics

For the long characteristic approach, each characteristic of Eq. (2.28a) is solved from problem boundary to problem boundary along a continuous trajectory. In the case of step MOC, the spatial dimensions of the flux that are perpendicular to the direction of travel are represented using a piecewise constant distribution. For brevity the long step

characteristics equation is commonly denoted by MoC. By imposing the MoC discretization scheme on Eq. (2.28a) and Eq. (2.29) and rewriting the angular flux in terms of its interface quantities we get,

$$\psi^{+}_{g,m,i,t} = \psi^{-}_{g,m,i,t} e^{-\Sigma_{g,i} l_{m,i,t}} + \frac{Q_{g,m,i}}{\Sigma_{g,i}} \left[1 - e^{-\Sigma_{g,i} l_{m,i,t}}\right], \tag{2.30a}$$

and

$$\bar{\phi}_{g,l,k,i} = \sum_{m=1}^{M} \frac{w_m Y_{l,k}(\hat{\Omega}_m)}{V_i} \sum_{t=1}^{T_{m,i}} w_t \left\{ \frac{\psi^{-}_{g,m,i,t}}{\Sigma_{g,i}} \left[1 - e^{-\Sigma_{g,i} l_{m,i,t}}\right] + \frac{Q_{g,m,i} l_{m,i,t}}{\Sigma_{g,i}} \left[1 - \frac{1 - e^{-\Sigma_{g,i} l_{m,i,t}}}{\Sigma_{g,i} l_{m,i,t}}\right] \right\},$$

where $t \in [1, T_{m,i}]$ are the indices of the discrete set of characteristics, along direction m, that intersect cell i and w_t is the geometric cross section of characteristic t. Moreover, $\psi^{-}_{g,m,i,t}$, and $\psi^{+}_{g,m,i,t}$ represent the incoming and outgoing interface angular flux within cell i and along characteristic t, respectively.

2.9.4.2 Step short characteristics

The step short characteristics approach uses a flat distribution to represent the interface angular flux, such that,

$$\psi^{\pm}_{g,m}(\vec{r}_{s'}) = \psi^{\pm}_{g,m,i,s} \quad \forall\ \vec{r}_{s'} \in \Gamma^{\pm}_{i,s}, \tag{2.31a}$$

$$\psi^{\pm}_{g,m,i,s} = \left|\hat{\Omega}_m \cdot \hat{n}_{i,s}\right| \int_{\Gamma^{\pm}_{i,s}} d^2 r_s\ \psi^{\pm}_{g,m}(\vec{r}_{s'}), \tag{2.31b}$$

$$\psi^{+}_{g,m,i,s} = \psi_{g,m,i,s} \quad \text{when } m \in M^{+}_{i,s}, \tag{2.31c}$$

$$\psi^{-}_{g,m,i,s'} = \psi_{g,m,i,s'} \quad \text{when } m \in M^{-}_{i,s'}, \tag{2.31d}$$

where $s \in [1, S]$ is the cell surface index, $\Gamma^{\pm}_{i,s}$ is the spatial domain of surface s of cell i and $\psi_{g,m,i,s}$ is the zeroth spatial moment of the interface angular flux. $\hat{n}_{i,s}$ is the unitary vector that defines the surface normal pointing outward from the cell. $M^{+}_{i,s}$ and $M^{-}_{i,s}$ are sets of m such that $\Omega_m \cdot \hat{n}_{i,s} > 0$ and $\Omega_m \cdot \hat{n}_{i,s} < 0$, respectively. By substituting Eq. (2.31a) for the interface angular flux of Eq. (2.28a) and Eq. (2.29), we get,

$$\psi^{+}_{g,m,i,s} = \sum_{s' \in S^{-}_{m,i,s}} \left|\hat{\Omega}_m \cdot \hat{n}_{i,s'}\right| \int_{\Gamma^{-}_{i,s'}} d^2 r_s \times \left\{ \psi^{-}_{g,m,i,s'} e^{-\Sigma_{g,i} l_{m,i,s}(\vec{r}_{s'})} + \frac{Q_{g,m,i}}{\Sigma_{g,i}} \left[1 - e^{-\Sigma_{g,i} l_{m,i,s}(\vec{r}_{s'})}\right] \right\}, \tag{2.32a}$$

and

$$\bar{\phi}_{g,l,k,i} = \sum_{m=1}^{M} \frac{w_m Y_{l,k}(\hat{\Omega}_m)}{V_i} \sum_{s' \in S^{-}_{m,i}} \left|\hat{\Omega}_m \cdot \hat{n}_{i,s'}\right| \int_{\Gamma^{-}_{i,s'}} d^2 r_s \times \left\{ \frac{\psi^{-}_{g,m,i,s'}}{\Sigma_{g,i}} \left[1 - e^{-\Sigma_{g,i} l_{m,i,s}(\vec{r}_{s'})}\right] + \frac{Q_{g,m,i} l_{m,i,s}(\vec{r}_{s'})}{\Sigma_{g,i}} \left[1 - \frac{1 - e^{-\Sigma_{g,i} l_{m,i,s}(\vec{r}_{s'})}}{\Sigma_{g,i} l_{m,i,s}(\vec{r}_{s'})}\right] \right\}, \tag{2.32b}$$

where, $S^{-}_{m,i}$ is the set of surfaces of cell i whose normal satisfies $\Omega_m \cdot \hat{n}_{i,s} < 0$ and $S^{-}_{m,i,s}$ is the set of surfaces of cell i whose incoming angular flux contributes to surface s along

direction m. Also, $l_{m,i,s}(\vec{r}_{s'})$ is the chord length along direction m from the position $\vec{r}_{s'}$ to surface s. Finally, Eqs.(2.32a) and (2.32b) are rewritten more concisely as,

$$\psi^{+}_{g,m,i,s} = \sum_{s' \in S^{-}_{m,i,s}} T_{g,m,i,s,s'} \psi^{-}_{g,m,i,s'} + E_{g,m,i,s} Q_{g,m,i}, \tag{2.33a}$$

and

$$\bar{\phi}_{g,l,k,i} = \sum_{m=1}^{M} w_m Y_{l,k}(\hat{\Omega}_m) \left[\sum_{s \in S^{-}_{m,i}} I_{g,m,i,s} \psi^{-}_{g,m,i,s} + C_{g,m,i} Q_{g,m,i} \right], \tag{2.33b}$$

where, $T_{g,m,i,s,s'}$, $E_{g,m,i,s}$, $I_{g,m,i,s}$ and $C_{g,m,i}$, are the MOC coefficients that define the transmission of the uncollided flux from surface s' to surface s, the escape of the collided flux to surface s, the contribution of the incident uncollided flux to the average flux and the contribution of the collided flux to the average flux, respectively.

2.9.5 Higher order short MOC

Before we begin, it is important to note that for the remainder of the text, bold double underlined font will be used to indicate a matrix and bold font will be used to indicate a vector. For higher order short characteristics, an orthogonal polynomial basis is used to represent the volumetric source and the interface angular flux, such that,

$$Q_{m,i}(\vec{r}) = \boldsymbol{P}_i(\vec{r}) \cdot \boldsymbol{Q}_{m,i}, \tag{2.34a}$$

$$\psi^{\pm}_{m,i,s}(\vec{r}_s) = \boldsymbol{P}_{i,s}(\vec{r}_s) \cdot \boldsymbol{\psi}^{\pm}_{m,i,s}. \tag{2.34b}$$

The moments in previous Eqs. are defined by the projections:

$$\boldsymbol{Q}_{m,i} = \int_{D_i} d^3\vec{r}\ \boldsymbol{P}_i(\vec{r}) Q_{m,i}(\vec{r}),$$

$$\boldsymbol{\psi}^{\pm}_{m,i,s} = \left| \hat{\Omega}_m \cdot \hat{n}_{i,s} \right| \int_{\Gamma_{i,s}} d^2\vec{r}_s\ \boldsymbol{P}_{i,s}(\vec{r}_s) \psi^{\pm}_{m,i,s}(\vec{r}_s).$$

where $\boldsymbol{P}_i(\vec{r})$ and $\boldsymbol{P}_{i,s}(\vec{r}_s)$ are vectors of the spatial basis functions in the volume and on the surface, respectively. $\boldsymbol{Q}_{m,i}$ and $\boldsymbol{\psi}^{\pm}_{m,i,s}$ are vectors of the spatial moments of the volumetric source and the interface angular flux, respectively. By substituting the expansions (2.34a) and (2.34b) into the integral transport equation (2.25), one obtains the discrete form of the integral transport equation,

$$\boldsymbol{\psi}^{+}_{m,i,s} = \sum_{s' \in S^{-}_{m,i,s}} \underline{\underline{\boldsymbol{T}}}_{m,i,s,s'} \boldsymbol{\psi}^{-}_{m,i,s'} + \underline{\underline{\boldsymbol{E}}}_{m,i,s} \boldsymbol{Q}_{m,i}, \tag{2.36a}$$

$$\bar{\boldsymbol{\phi}}_{l,k,i} = \sum_{m=1}^{M} w_m Y_{l,k}(\hat{\Omega}_m) \sum_{s' \in S^{-}_{m,i}} \underline{\underline{\boldsymbol{I}}}_{m,i,s'} \boldsymbol{\psi}^{-}_{m,i,s'} + \underline{\underline{\boldsymbol{C}}}_{m,i} \boldsymbol{Q}_{m,i}, \tag{2.36b}$$

$$\boldsymbol{Q}_{m,i} = \sum_{l=0}^{L} \sum_{k=-l}^{l} Y_{l,k}(\hat{\Omega}_m) \left[\Sigma_{sc,l,i} \boldsymbol{\phi}_{l,k,i} + \boldsymbol{Q}_{Ext,l,k,i} \right]. \tag{2.36c}$$

The first equation transmits the angular flux through the region to the outgoing surfaces, while the second equation takes into account the high-order interior spatial moments of the flux.

2.9.6 Short MOC applied to heterogeneous Cartesian cells

The high-order short MOC has been extended for Heterogeneous Cartesian Cells (HCC) in IDT. With reference to Fig. 2.3, the HCC is composed by an inner heterogeneous volume mesh and a boundary Cartesian mesh. When specialized to a HCC, the expansion of the source defined by Eq. (2.34a) becomes,

$$Q_{m,i,\alpha}(\vec{r}) = \boldsymbol{P}_{i,\alpha}(\vec{r}) \cdot \boldsymbol{Q}_{m,i,\alpha}, \tag{2.37}$$

where, α is the subregion index of cell i. Substituting Eq. (2.37) into Eqs. (2.36a) and (2.36b) and summing over all the contributing subregions β of cell i we get,

$$\boldsymbol{\psi}^{+}_{m,i,s} = \sum_{s' \in S^{-}_{m,i,s}} \underline{\underline{\boldsymbol{T}}}_{m,i,s,s'} \boldsymbol{\psi}^{-}_{m,i,s'} + \sum_{\beta \in i} \underline{\underline{\boldsymbol{E}}}_{m,i,s,\beta} \boldsymbol{Q}_{m,i,\beta}, \tag{2.38a}$$

$$\bar{\boldsymbol{\phi}}_{l,k,i,\alpha} = \sum_{m=1}^{M} w_m Y_{l,k}(\hat{\Omega}_m) \sum_{s' \in S^{-}_{m,i}} \underline{\underline{\boldsymbol{I}}}_{m,i,\alpha,s'} \boldsymbol{\psi}^{-}_{m,i,s'} + \sum_{\beta \in i} \underline{\underline{\boldsymbol{C}}}_{m,i,\alpha,\beta} \boldsymbol{Q}_{m,i,\beta}, \tag{2.38b}$$

where s is now the subsurface index of cell i.

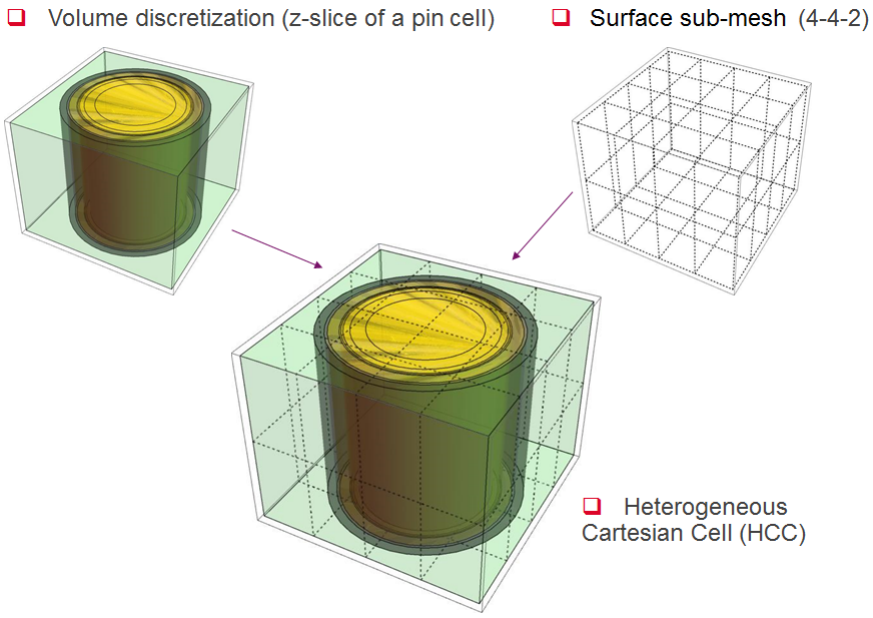

Figure 2.3 – A visualization of the modelling of a HCC in IDT.

2.9.7 Comparison of long and short characteristics

For long characteristics, Eq. (2.30a), the coefficients that define the contribution of the incoming angular flux and the source to the outgoing angular flux are unique to each cell and intersecting characteristic and are therefore often impractical to store in memory. However, for short characteristics, Eq. (2.33a), the coefficients that define the contribution of the incoming angular flux and the source to the outgoing angular flux are unique to each cell shape, medium and surface. Since there is often a high degree of geometric symmetry within a reactor, it is usually possible to store the short characteristics coefficients in

memory. By storing the coefficients in this way, the short characteristic approach is computationally less expensive than the long characteristic approach but requires more memory. Due to its computational performance and the time constraints of the PhD, we have chosen to only collect results for the short characteristic approach. Therefore, we will only refer to the short MOC past this point.

2.10 Boundary Conditions

In order to close the system of transport equations, boundary conditions need to be specified.

Reflective boundary conditions,

$$\psi^{-}_{g,m,i,s} = \psi^{+}_{g,m',i,s} \quad \forall\ (i,s) \in \Gamma_{Bnd}, \tag{2.39a}$$

$$\hat{\Omega}_m = \hat{\Omega}_{m'} - 2|\hat{n}_{i,s} \cdot \hat{\Omega}_{m'}|\hat{n}_{i,s}, \tag{2.39b}$$

can be used when there is reflective symmetry along the problem boundary Γ_{Bnd}. An illustration of reflective boundary conditions is shown by Fig. 2.4.

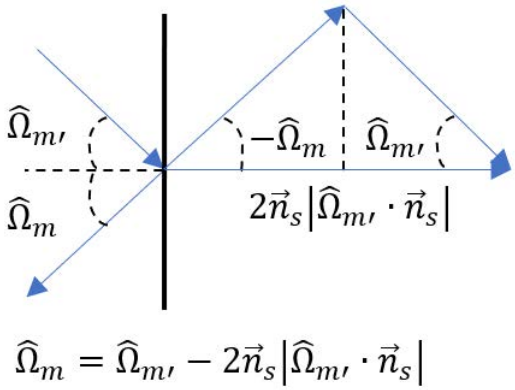

Figure 2.4 – A visualization of reflective boundary conditions.

Periodic boundary conditions,

$$\psi^{-}_{g,m,i,s} = \psi^{+}_{g,m,i',s'} \quad \forall\ (i',s') \in \Gamma_{Bnd}, \tag{2.40a}$$

$$\vec{r}_{i,s} = \vec{r}_{i',s'} + \vec{r}_{shift}, \tag{2.40b}$$

where $\vec{r}_{shift}$ is the periodic shift, are used when there is a periodic symmetry in the geometry. An illustration of periodic boundary conditions is shown by Fig. 2.5.

Vacuum boundary conditions,

$$\psi^{-}_{g,m,i,s} = 0 \quad \forall\ (i,s) \in \Gamma_{Bnd}, \tag{2.41}$$

are used to represent boundary upon which the incoming flux is insignificant.

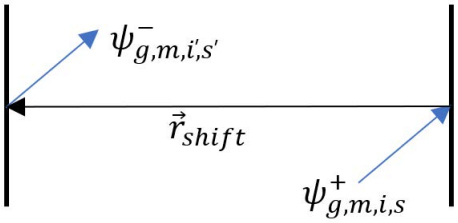

Figure 2.5 – A visualization of periodic boundary conditions.

Albedo boundary condition,

$$\psi^-_{g,m,i,s} = \sum_{g'=1}^{G} \sum_{m' \in M^+_{i,s}} \sum_{(i',s') \in \Gamma_{Bnd}} \alpha(E_{g'} \to E_g, \hat{\Omega}_{m'} \to \hat{\Omega}_m, \vec{r}_{i',s'} \to, \vec{r}_{i,s}) \psi^+_{g',m',i',s'}, \quad (2.42)$$

where $\alpha(E_{g'} \to E_g, \hat{\Omega}_{m'} \to \hat{\Omega}_m, \vec{r}_{i',s'} \to, \vec{r}_{i,s})$ is the albedo coefficient, is used for boundaries where the incoming flux can be approximated as a multiple of the outgoing flux.

2.11 Iteration schemes

The solution of the steady-state transport equation, which is a homogeneous problem, is the so called fundamental mode associated to its largest eigenvalue, k. In order to define the iterative scheme used to solve for the fundamental mode, we will first put the transport equation in the eigenvalue form. The transport equation can be expressed in matrix form such that,

$$k\boldsymbol{\Psi} = \underline{\underline{\boldsymbol{B}}}^{-1} \underline{\underline{\tilde{\boldsymbol{F}}}} \boldsymbol{\Psi}, \quad (2.43)$$

where $\underline{\underline{\boldsymbol{B}}}$, represents the combination of the streaming, the absorption and the scattering operators, $\boldsymbol{\Psi}$ is the vector of the angular flux and $\underline{\underline{\tilde{\boldsymbol{F}}}}$ is the fission operator. The vector $\boldsymbol{\Psi}$ contains the discrete coefficients relative to the spatial discretization, the multigroup discretization and the angular discretization for the angular flux. Next, $\underline{\underline{\tilde{\boldsymbol{F}}}}$ is applied to both sides of Eq. (2.43) and simplified to get,

$$k\boldsymbol{F} = \underline{\underline{\boldsymbol{A}}}\boldsymbol{F}, \quad (2.44a)$$

$$\underline{\underline{\boldsymbol{A}}} = \underline{\underline{\tilde{\boldsymbol{F}}}}\,\underline{\underline{\boldsymbol{B}}}^{-1}, \quad (2.44b)$$

and

$$\boldsymbol{F} = \underline{\underline{\tilde{\boldsymbol{F}}}}\boldsymbol{\Psi}, \quad (2.44c)$$

where, $\boldsymbol{F}$ is the fission source. Under very mild conditions $\underline{\underline{\boldsymbol{A}}}$ can be proven to have a dominant eigenvalue which is nondegenerate and positive, with a corresponding positive $\boldsymbol{F}$. The most common approach to solving the eigenvalue problem defined by Eq. (2.44a) is the power iteration method, [25].

$$\boldsymbol{F}^{(e+1)} = \frac{1}{k^{(e)}} \underline{\underline{\boldsymbol{A}}} \boldsymbol{F}^{(e)},$$

$$k^{(e+1)} = k^{(e)} \frac{\boldsymbol{W} \cdot \boldsymbol{F}^{(e+1)}}{\boldsymbol{W} \cdot \boldsymbol{F}^{(e)}},$$

where $\boldsymbol{W}$ represents a suitable weight vector and e is the power/outer iteration index. By using the power iteration method, Eq. (2.44a) is assumed to have a unique, maximum, positive and real eigenvalue with a positive corresponding $\boldsymbol{F}$. In IDT, $\boldsymbol{W} = \boldsymbol{F}^{(e)}$ such that,

$$k^{(e+1)} = k^{(e)} \frac{\boldsymbol{F}^{(e+1)} \cdot \boldsymbol{F}^{(e)}}{\boldsymbol{F}^{(e)} \cdot \boldsymbol{F}^{(e)}}.$$

In practise, Eq. (2.43) is not constructed by direct inversion and is instead computed using a multilevel iterative scheme. At the top level, a Gauss-Seidel iteration scheme over energy is used by IDT to update the group to group scattering source such that,

$$\left[\hat{\Omega} \cdot \nabla + \Sigma_g + \underline{\underline{\boldsymbol{H}}}_{g,g}\right] \boldsymbol{\Psi}_g^{(t+1)} = \boldsymbol{Q}_{Ext,g}^{(t+1)}, \tag{2.45a}$$

$$\boldsymbol{Q}_{Ext,g}^{(t+1)} = \sum_{g'<g} \underline{\underline{\boldsymbol{H}}}_{g,g'} \boldsymbol{\Psi}_{g'}^{(t+1)} + \sum_{g'>g} \underline{\underline{\boldsymbol{H}}}_{g,g'} \boldsymbol{\Psi}_{g'}^{(t)} + \frac{\boldsymbol{F}_g^{(e)}}{k^{(e)}}, \tag{2.45b}$$

where $\boldsymbol{F}_g$ is the contribution of the fission source to group g, $\underline{\underline{\boldsymbol{H}}}_{g,g}$ represents the self-scattering of group g and $t \in [1, T]$ is the thermal iteration index. The first two terms of $Q_{Ext,g}^{(t)}$, represent the down and up scattering to group g, respectively. Since the fission source is of primary interest, it is not common for the convergence of the group to group scattering to be enforced. In fact, it is typical for Eq. (2.45a) to be solved only once per group, per outer iteration.

For thermal reactors, fission almost exclusively occurs as the result of the absorption of thermal neutrons, [2]. Thus, the distribution of the flux among the thermal energy groups has a strong effect on $k^{(e+1)}$. As well, neutrons in the thermal groups are closer to being in thermal equilibrium with their surrounding materials. This means that the velocity of the target atom, relative to the medium, has a significant effect on the energy of the scattered neutron. As a result, thermal neutrons have the potential to gain energy as well as lose it. This leads to the neutron distribution of the thermal groups being more strongly dependent on the surrounding groups than those of fast neutrons, causing them to take more iterations to converge. To alleviate this issue, codes, such as IDT, solve the fast groups once and the thermal groups multiple times for each outer iteration. The purpose of these thermal iterations is to focus the computational effort on converging the thermal groups since they are slowest to converge and have the most significant effect on the convergence of the k-effective.

At the lower level, the within group scattering source is usually iteratively converged, with the iterations commonly being referred to as inner iterations or Source iterations (SI),

$$\boldsymbol{\Psi}_g^{(n+1)} = \underline{\underline{\boldsymbol{L}}}_g \boldsymbol{\Psi}_g^{(n)} + \boldsymbol{Q}_g, \tag{2.46a}$$

$$\boldsymbol{Q}_g = \left[\hat{\Omega} \cdot \nabla + \Sigma_g\right]^{-1} \boldsymbol{Q}_{Ext,g}^{(t)}, \tag{2.46b}$$

with,

$$\underline{\underline{\boldsymbol{L}}}_g = \left[\hat{\Omega} \cdot \nabla + \Sigma_g\right]^{-1} \underline{\underline{\boldsymbol{H}}}_{g,g} \tag{2.46c}$$

where n is the inner iteration index.

2.12 Standard inner iteration scheme for step MOSC

In the case of step MOSC, the standard scheme in IDT for solving Eq. (2.46a) can be defined as,

$$\psi^{+,(n+1)}_{g,m,i,s} = \sum_{s' \in S^{-}_{m,i,s}} T_{g,m,i,s,s'} \psi^{-,(n+1)}_{g,m,i,s'} + E_{g,m,i,s} Q^{(n)}_{g,m,i}, \tag{2.47a}$$

$$\bar{\phi}^{(n+1)}_{g,l,k,i} = \sum_{m=1}^{M} w_m Y_{l,k}(\hat{\Omega}_m) \left[\sum_{s' \in S^{-}_{m,i}} I_{g,m,i,s'} \psi^{-,(n+1)}_{g,m,i,s'} + C_{g,m,i} Q^{(n)}_{g,m,i} \right], \tag{2.47b}$$

$$Q^{(n+1)}_{g,m,i} = \sum_{l=0}^{L} \sum_{k=-l}^{l} Y_{l,k}(\hat{\Omega}_m) \left[\Sigma_{sc,l,i} \bar{\phi}^{(n+1)}_{g,l,k,i} + Q^{(t)}_{Ext,g,l,k,i} \right], \tag{2.47c}$$

where each interface incoming angular flux is updated by the upstream interface angular flux from the $(n+1)$ iteration. As mentioned in Section 2.7.2, one of the benefits of the S_N methods is the ease at which they can be computed in parallel. From Eq. (2.47a), each direction of the angular flux exiting the region, $\psi^{+,(n+1)}_{g,m,i,s}$, is only dependent on the same direction of the incoming angular flux, $\psi^{-,(n+1)}_{g,m,i,s'}$, and the previous iteration of the scattering source, $Q^{(n)}_{g,m,i}$. This property allows for each direction of the angular flux to be solved simultaneously in parallel.

Along the boundaries of the spatial domain, the incoming angular flux is computed using one of the boundary conditions described in Section 2.10 and the (n) iteration of the outgoing angular flux. By setting the incoming boundary angular flux to be from the previous iteration, the remaining angular flux can be solved from boundary to boundary in a single sweep as shown by Fig. 2.6. The advantage of this scheme is that the need for a direct inversion is avoided. As will be shown in Section 2.16.1, this advantage comes at the cost of a diminished convergence rate.

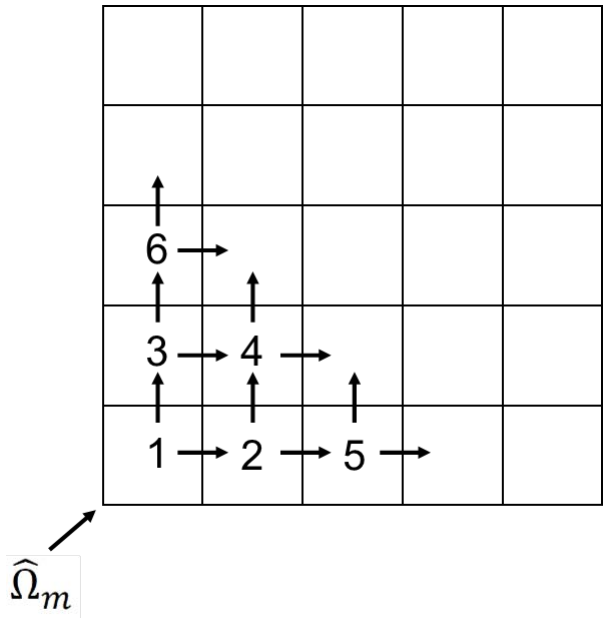

Figure 2.6 – A visualization of the standard S_N sweeping in IDT, for a 2D cartesian mesh. Numbers are used to indicate the order in which each cell will update its outgoing interface angular flux for direction m.

2.13 1D matrix form of the standard inner iteration scheme for step MOC operator

Before we begin to analyze the convergence properties of the source iterations of step MOC, we will first simplify our analysis by restricting ourselves to 1D slab geometries with isotropic scattering. Fig. 2.7 illustrates the indexing scheme employed by the authors for 1D geometries. Subscripts L and R denote the left and the right side of the cell.

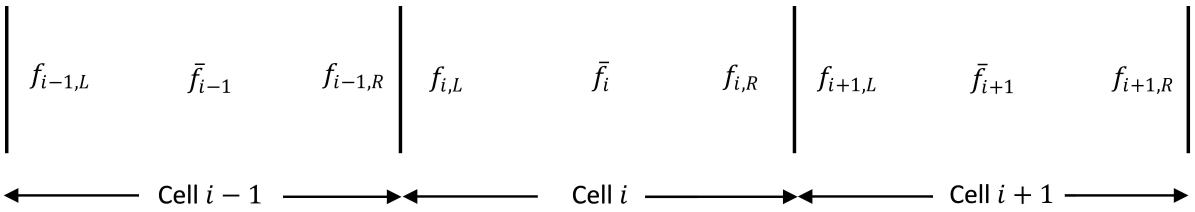

Figure 2.7 – An illustration of the indexing scheme for a 1D geometry, for an ambiguous variable f.

The MOC inner iteration scheme can conveniently be written in matrix form for a single cell i as,

$$-\underline{\underline{\boldsymbol{L}}}_{i,i-1}\,\boldsymbol{\Psi}_{i-1}^{(n+1)}+\boldsymbol{\Psi}_{i}^{(n+1)}-\underline{\underline{\boldsymbol{L}}}_{i,i+1}\,\boldsymbol{\Psi}_{i+1}^{(n+1)}=\underline{\underline{\boldsymbol{L}}}_{i,i}\boldsymbol{\Psi}_{i}^{(n)}+\boldsymbol{Q}_i, \tag{2.48}$$

The matrices and vectors of Eq. (2.48) are defined as,

$$\underline{\underline{\boldsymbol{L}}}_{i,i} = \begin{bmatrix} 0 & 0 & \boldsymbol{E}_i \\ 0 & 0 & \boldsymbol{E}_i \\ 0 & 0 & \Sigma_{sc,i} C_i \end{bmatrix}, \tag{2.49a}$$

$$\underline{\underline{\boldsymbol{L}}}_{i,i-1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \underline{\underline{\boldsymbol{T}}}_i & 0 \\ 0 & \boldsymbol{I}_i & 0 \end{bmatrix}, \tag{2.49b}$$

$$\underline{\underline{\boldsymbol{L}}}_{i,i+1} = \begin{bmatrix} \underline{\underline{\boldsymbol{T}}}_i & 0 & 0 \\ 0 & 0 & 0 \\ \boldsymbol{I}_i & 0 & 0 \end{bmatrix}, \tag{2.49c}$$

$$\boldsymbol{\Psi}_i = \begin{bmatrix} \boldsymbol{\Psi}^+_{i,L} \\ \boldsymbol{\Psi}^+_{i,R} \\ \bar{\phi}_i \end{bmatrix}, \tag{2.49d}$$

$$\boldsymbol{Q}_i = \begin{bmatrix} \tilde{\boldsymbol{E}}_i Q_{Ext,i} \\ \tilde{\boldsymbol{E}}_i Q_{Ext,i} \\ C_i Q_{Ext,i} \end{bmatrix}, \tag{2.49e}$$

where,

- $\underline{\underline{\boldsymbol{T}}}_i$ is $M/2 \times M/2$ matrix that represents the contribution of the surface to surface transmission of uncollided flux to the interface angular flux.
- $\boldsymbol{E}_{i,L/R}$ and $\tilde{\boldsymbol{E}}_{i,L/R}$ are $M/2 \times 1$ vectors that represent the contribution of the escape of the collided to the interface angular flux, from within group scattering and from all other collisions, respectively.
- $\boldsymbol{I}_i$ is $1 \times M/2$ vector that represents the contribution of the incident uncollided flux to the average flux.
- $\boldsymbol{\Psi}_{i,L/R}$ and $\bar{\phi}_i$, are the outgoing angular flux on the left/right surface and the average scalar of flux of cell i, respectively.
- $Q_{Ext,i}$, represents the combination of the sources from fission and scattering from external groups.

Furthermore, the matrices and vectors of Eqs. (2.49a) to (2.49e) are defined as,

$$\underline{\underline{\boldsymbol{T}}}_i = \begin{bmatrix} T_{1,i} & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & T_{M/2,i} \end{bmatrix},$$

$$\boldsymbol{E}_i = \begin{bmatrix} \Sigma_{sc,i} E_{1,i} \\ \vdots \\ \Sigma_{sc,i} E_{M/2,i} \end{bmatrix},$$

$$\tilde{\boldsymbol{E}}_i = \begin{bmatrix} E_{1,i} \\ \vdots \\ E_{M/2,i} \end{bmatrix},$$

$$\boldsymbol{I}_i = \begin{bmatrix} I_{1,i} & \dots & I_{M/2,} \end{bmatrix},$$

$$T_{m,i} = e^{-\Sigma_i l_{m,i}},$$

$$E_{m,i} = \frac{[1 - T_{m,i}]}{\Sigma_i},$$

$$I_{m,i} = \frac{w_m E_{m,i}}{l_{m,i}},$$

$$C_i = \sum_{m=1}^{M} \frac{I_{m,i}}{\Sigma_i},$$

where $l_{m,i}$ is the chord length of cell i along direction m.

Finally, the standard MOC transport iteration scheme for the global problem can be defined as

$$\boldsymbol{\Psi}^{(n+1)} = \underline{\underline{\boldsymbol{L}}} \boldsymbol{\Psi}^{(n)} + \boldsymbol{Q}, \tag{2.50a}$$

$$\underline{\underline{\boldsymbol{L}}} = (\underline{\underline{\hat{\boldsymbol{I}}}} - \underline{\underline{\boldsymbol{L}}}_l - \underline{\underline{\boldsymbol{L}}}_u)^{-1} \underline{\underline{\boldsymbol{L}}}_d, \tag{2.50b}$$

$$\boldsymbol{Q} = (\underline{\underline{\hat{\boldsymbol{I}}}} - \underline{\underline{\boldsymbol{L}}}_l - \underline{\underline{\boldsymbol{L}}}_u)^{-1} \boldsymbol{Q}_{Ext}, \tag{2.50c}$$

where the $\underline{\underline{\boldsymbol{L}}}_d$ is the block-diagonal matrix composed by the matrices $\underline{\underline{\boldsymbol{L}}}_{i,i}$, while $\underline{\underline{\boldsymbol{L}}}_l$ and $\underline{\underline{\boldsymbol{L}}}_u$ respectively, are the lower and upper triangular matrices composed by the block matrices $\underline{\underline{\boldsymbol{L}}}_{i,i-1}$ and $\underline{\underline{\boldsymbol{L}}}_{i,i+1}$.

2.14 Convergence analysis of the transport operator

From Eq. (2.50a), it is easily discerned that the final solution of the transport operator, $\boldsymbol{\Psi}$, must satisfy the following relationship.

$$\boldsymbol{\Psi} = \underline{\underline{\boldsymbol{L}}} \boldsymbol{\Psi} + \boldsymbol{Q}. \tag{2.51}$$

By studying the change in the iterative error, $\partial \boldsymbol{\Psi}^{(n+1)} = \boldsymbol{\Psi} - \boldsymbol{\Psi}^{(n+1)}$, the rate at which $\boldsymbol{\Psi}^{(n+1)}$ converges to $\boldsymbol{\Psi}$ can be determined. Since the transport operator is linear, the difference of the current iteration of the transport solution and the final solution can be determined to be,

$$\partial \boldsymbol{\Psi}^{(n+1)} = \underline{\underline{\boldsymbol{L}}} \partial \boldsymbol{\Psi}^{(n)}. \tag{2.52}$$

Applying Eq. (2.52) recursively from the first iteration, [4], we get,

$$\partial\boldsymbol{\Psi}^{(n+1)} = \underline{\underline{\boldsymbol{L}}}^{n+1}\partial\boldsymbol{\Psi}^{(0)}. \tag{2.53}$$

Since we are only interested in the magnitude of the rate of convergence and not the direction, we take the L_∞ norm of Eq. (2.53) to get,

$$||\partial\boldsymbol{\Psi}^{(n+1)}|| = ||\underline{\underline{\boldsymbol{L}}}^{n+1}\partial\boldsymbol{\Psi}^{(0)}||.$$

By assuming that $\underline{\underline{\boldsymbol{L}}}$ is diagonizable, the following substitution can be made,

$$||\partial\boldsymbol{\Psi}^{(n+1)}|| = ||\underline{\underline{\boldsymbol{V}}}\,\underline{\underline{\boldsymbol{\Lambda}}}^{n+1}\underline{\underline{\boldsymbol{V}}}^{-1}\partial\boldsymbol{\Psi}^{(0)}||, \tag{2.54}$$

where the columns of $\underline{\underline{\boldsymbol{V}}}$ are the eigenvectors of $\underline{\underline{\boldsymbol{L}}}$ and $\underline{\underline{\boldsymbol{\Lambda}}}$ is a diagonal matrix such that $\underline{\underline{\boldsymbol{\Lambda}}}_{i,i}$ equals the eigenvalue corresponding to column i of $\underline{\underline{\boldsymbol{V}}}$. As $n \to \infty$, the eigenvector corresponding to the spectral radius of $\underline{\underline{\boldsymbol{L}}}$, $\rho = max\{|\lambda_1|, ..., |\lambda_N|\}$, will dominate $\partial\boldsymbol{\Psi}$ such that,

$$||\partial\boldsymbol{\Psi}^{(n+1)}|| = \rho^{(n+1)}||\partial\boldsymbol{\Psi}^{(0)}||, \quad \text{when } n \to \infty. \tag{2.55}$$

Thus, when $\boldsymbol{\Psi}^{(0)}$ is far from convergence, the rate of convergence is determined by the spectral radius. Moreover, in order for $\boldsymbol{\Psi} - \boldsymbol{\Psi}^{(n+1)}$ to approach to 0 when $n \to \infty$, ρ must be less than 1. For the remainder of this manuscript we will refer to the process of determining the spectral radius of an iterative operator as **spectral radius analysis**.

As stated in the introduction, Section 1, we have chosen to focus on problems when the spatial domain dominates the time to convergence. Although spectral radius analysis, is useful for assessing the rate of convergence and stability of an operator, it does not provide insight into the spatial dependence of the spectral radius. In order to assess the spatial dependence of the spectral radius, Fourier analysis must be used.

2.15 Fourier analysis for linear operators

Fourier analysis works by taking the Fourier expansion of $\partial\boldsymbol{\Psi}_i$ over the spatial domain, [4], such that,

$$\partial\boldsymbol{\Psi}_i^{(n)} = \int_{-\infty}^{\infty} d\zeta\ \underline{\underline{\boldsymbol{F}}}_i(\zeta)\partial\boldsymbol{\Psi}^{(n)}(\zeta), \tag{2.56a}$$

$$\underline{\underline{\boldsymbol{F}}}_i(\zeta) = \begin{bmatrix} e^{j\frac{2\pi\zeta}{\Delta}\left[x_i - \frac{\Delta}{2}\right]}\hat{\underline{\underline{\boldsymbol{I}}}}_{\frac{M}{2}\times\frac{M}{2}} & 0 & 0 \\ 0 & e^{j\frac{2\pi\zeta}{\Delta}\left[x_i + \frac{\Delta}{2}\right]}\hat{\underline{\underline{\boldsymbol{I}}}}_{\frac{M}{2}\times\frac{M}{2}} & 0 \\ 0 & 0 & e^{j\frac{2\pi\zeta}{\Delta}x_i} \end{bmatrix}. \tag{2.56b}$$

where $\partial\boldsymbol{\Psi}(\zeta)$ is a vector of amplitudes for the periodic functions, $\hat{\boldsymbol{I}}$ is the identity matrix, x_i is the midpoint of region i and ζ is the spatial frequency. Fourier expansion used in Eq. (2.56a) assumes that the solution is periodic in space which means that Fourier analysis can only be applied to a system of equations that is invariant under translation. For the neutron transport equation this means that the geometry must be composed of a repeated, potentially heterogeneous, cells. This means that the matrices representing the contributions of cells $i-1$, i and $i+1$ to cell i are fixed such that,

$$\underline{\underline{\boldsymbol{L}}}_{i,i-1} = \underline{\underline{\boldsymbol{L}}}_L, \qquad \underline{\underline{\boldsymbol{L}}}_{i,i} = \underline{\underline{\boldsymbol{L}}}_C, \qquad \underline{\underline{\boldsymbol{L}}}_{i,i+1} = \underline{\underline{\boldsymbol{L}}}_R,$$

Substituting Eq. (2.56a) into Eq. (2.48) gives us,

$$\int_{-\infty}^{\infty} d\zeta \ \left[-\underline{\underline{\boldsymbol{L}}}_L \ \underline{\underline{\boldsymbol{F}}}_{i-1}(\zeta)\partial\boldsymbol{\Psi}^{(n+1)}(\zeta) + \underline{\underline{\boldsymbol{F}}}_i(\zeta)\partial\boldsymbol{\Psi}^{(n)}(\zeta) - \underline{\underline{\boldsymbol{L}}}_R \ \underline{\underline{\boldsymbol{F}}}_{i+1}(\zeta)\partial\boldsymbol{\Psi}^{(n+1)}(\zeta)\right] \tag{2.57}$$
$$= \int_{-\infty}^{\infty} d\zeta \ \underline{\underline{\boldsymbol{L}}}_C \ \underline{\underline{\boldsymbol{F}}}_i(\zeta)\partial\boldsymbol{\Psi}^{(n)}(\zeta)$$

Due to the orthogonality of the periodic functions, each spatial frequency of Eq. (2.57) can be analyzed separately such that

$$\partial\boldsymbol{\Psi}^{(n+1)}(\zeta) = \underline{\underline{\boldsymbol{L}}}(\zeta) \ \partial\boldsymbol{\Psi}^{(n)}(\zeta). \tag{2.58}$$
$$\underline{\underline{\boldsymbol{L}}}(\zeta) = \left[-\underline{\underline{\boldsymbol{L}}}_L \ \underline{\underline{\boldsymbol{F}}}_{i-1}(\zeta) + \underline{\underline{\boldsymbol{F}}}_i(\zeta) - \underline{\underline{\boldsymbol{L}}}_R \ \underline{\underline{\boldsymbol{F}}}_{i+1}(\zeta)\right]^{-1} \underline{\underline{\boldsymbol{L}}}_C \ \underline{\underline{\boldsymbol{F}}}_i(\zeta)$$

By analyzing the spectral radius of $\underline{\underline{\boldsymbol{L}}}(\zeta)$, the dependence of the spectral radius on the spatial frequency can be discerned.

2.16 Test cases for investigating convergence

Throughout our convergence analysis of the source iterations, the investigated geometry is a 1D homogeneous slab with isotropic scattering. As well, a S_8 angular distribution is used to represent the angular flux. Using the homogeneous geometry, three types of tests are performed for analyzing the convergence of the unaccelerated and accelerated source iteration scheme (SI) of MOC. For the remainder of this section we will outline what these tests are and briefly describe why they are interesting. A more detailed justification for why these tests are important is provided alongside the results in Section 2.16.1. This is done so that the meaning of the justifications can be more clearly interpreted from the results.

Infinite slab test: Perform a Fourier analysis of SI when the optical thickness and the scattering ratio of the medium are varied. The main interest of this test is to observe how the convergence behavior of the iterative scheme depends on the spatial frequency of the Fourier transformed solution. Studying this behavior is important for gaining insight into what parts of the solution are the slowest to converge. Moreover, by comparing the dependence of the spectral radius on the spatial frequency for SI alone and SI accelerated, important information can be discerned. Namely, the effectiveness at which the acceleration method is able to eliminate the spectral radius of certain spatial frequency. As well, this comparison is useful for checking if the acceleration method amplifies the spectral radius of any of the spatial frequencies.

Finite slab test: Perform a spectral radius analysis of a finite domain when the optical thickness and the scattering ratio are varied. Analyzing this behavior for a finite domain is necessary since important differences in the behavior occur as a result of the boundary incoming boundary flux being from the previous iteration. These effects will be discussed in more detail in Section 2.16.2. Moreover, the finite geometry is a more realistic representation of the numerical problems that we want to understand. As well, the standard scheme is compared to an iterative scheme where the incoming interface angular flux is from the previous iteration. This second iterative scheme, called the Gauss-Jacobi (GJ) scheme, is defined in detail in Section 2.16.3. By setting the incoming interface angular

flux to be from the previous iteration, the GJ scheme allows for each spatial region to be solved simultaneously in parallel. In contrast, the standard iteration scheme must be solved sequentially along the spatial domain. Thus, the standard and GJ schemes represent the two extremes of spatial parallelization. By comparing the spectral radius of the two different iterative schemes, the effectiveness at which the spatial domain of the problem can be decomposed for parallel simulation can be discerned. It is important to note that whenever the type of SI scheme is not explicitly specified, then the standard scheme is the scheme being referred to.

Spatial discretization test: Perform a spectral radius analysis of standard SI scheme for a finite domain when the optical thickness of each region is fixed to 0.1 mfp and the scattering ratio and the number of cells are varied. By varying the size of the mesh, the trend of how the spectral radius of SI changes as a more realistic sized problem is approached becomes clear. Moreover, the spectral radius results can also be interpreted as the effectiveness of solving a single parallel subdomain as the size of the subdomain is increased.

2.16.1 Convergence analysis of MOC

In the following sections the results of the spectral and Fourier analysis of the MOC transport operator are presented. As well, each section is dedicated to explaining an aspect of the three tests outlined in Section 2.16. The first section is dedicated to comparing the results of Fourier and spectral radius analysis. In the next section, the differences between analyzing infinite and finite problems are discussed. In the third section, a Gauss-Jacobi variant of MOC is introduced and compared to the standard sweeping scheme. The last section is dedicated to the numerical validation of the analytical results.

2.16.1.1 Comparison of Fourier and spectral radius analysis

In order to underline the differences and the commonalities between the Fourier and spectral radius analysis, a comparison of the two approaches is performed. The problem is modeled according to the specifications of the infinite slab test. As stated in Section 2.15, Fourier analysis can only be applied to geometries composed of an infinitely repeating pattern of cells. Thus, the problem we choose to investigate, was a finite homogeneous slab with periodic boundary conditions, Eq. (2.40a), where the incoming angular boundary flux is from the current, $(n+1)$, iteration,

$$\psi^{-,(n+1)}_{m,i,s} = \psi^{+,(n+1)}_{m,i',s'} \quad \forall\ (i',s') \in \Gamma_{Bnd}.$$

A comparison of the spectral radius obtained from the spectral and Fourier analysis, when optical thickness and scattering ratio are varied, is shown in Fig. 2.8. For all cases, the spectral radius of the two different approaches are equivalent. This is due to the dominance of the spectral radius corresponding to the zeroth spatial frequency, as is shown by Fig. 2.9. When the spatial frequency is set to zero, it is trivial to determine that the wave functions of the Fourier transform reduce to 1 such that Eq. (2.58) becomes,

$$\partial\boldsymbol{\Psi}^{(n+1)}(0) = \underline{\underline{\boldsymbol{L}}}_{i,i'}(0)\partial\boldsymbol{\Psi}^{(n)}(0) = \underline{\underline{\boldsymbol{L}}}_{i,i'}\partial\boldsymbol{\Psi}^{(n)}(0),$$

which is equivalent to Eq. (2.52). Therefore, when the spatial frequency is set to zero, the Fourier transformed transport operator is equivalent to the transport operator of the repeated geometry analyzed by spectral radius analysis. Also, the spectral radius of both approaches is simply the scattering ratio. This is caused by the incoming angular flux being from the same iteration as the outgoing angular flux, resulting in the angular flux only being solved in terms of the scalar flux. By eliminating the angular flux from the solution and choosing the repeated medium to be made up of only one cell, the following relation is obtained,

$$\phi^{(n+1)} = c\phi^{(n)} + Q_{Ext},$$

where c is the scattering ratio. From the above equation it readily apparent that the spectral radius of the chosen problem is the scattering ratio.

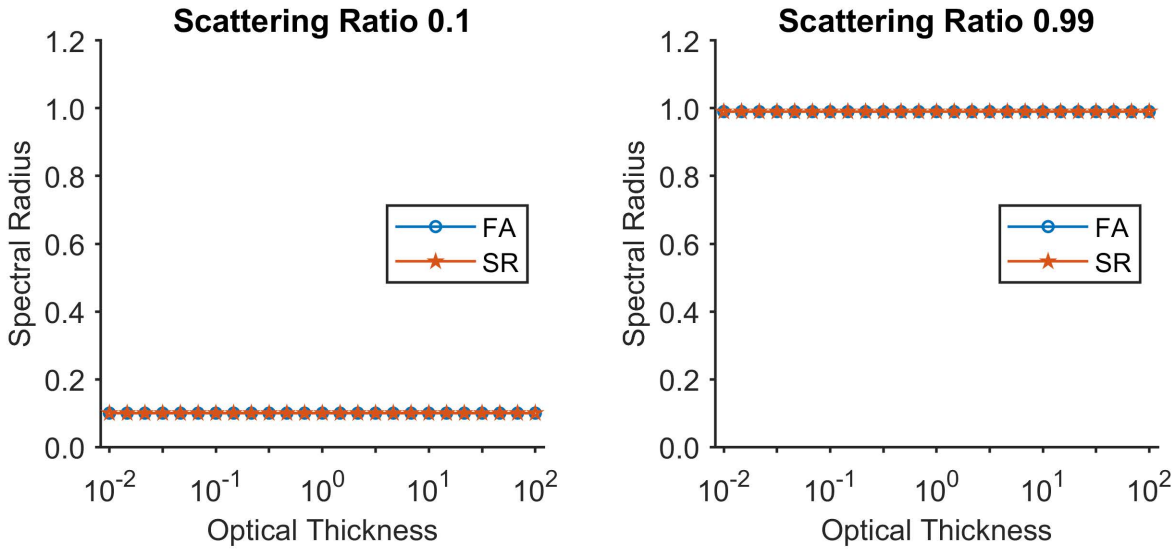

Figure 2.8 – Infinite slab test: a comparison of the Fourier (FA) and spectral radius (SR) analysis of MOC for an infinite slab.

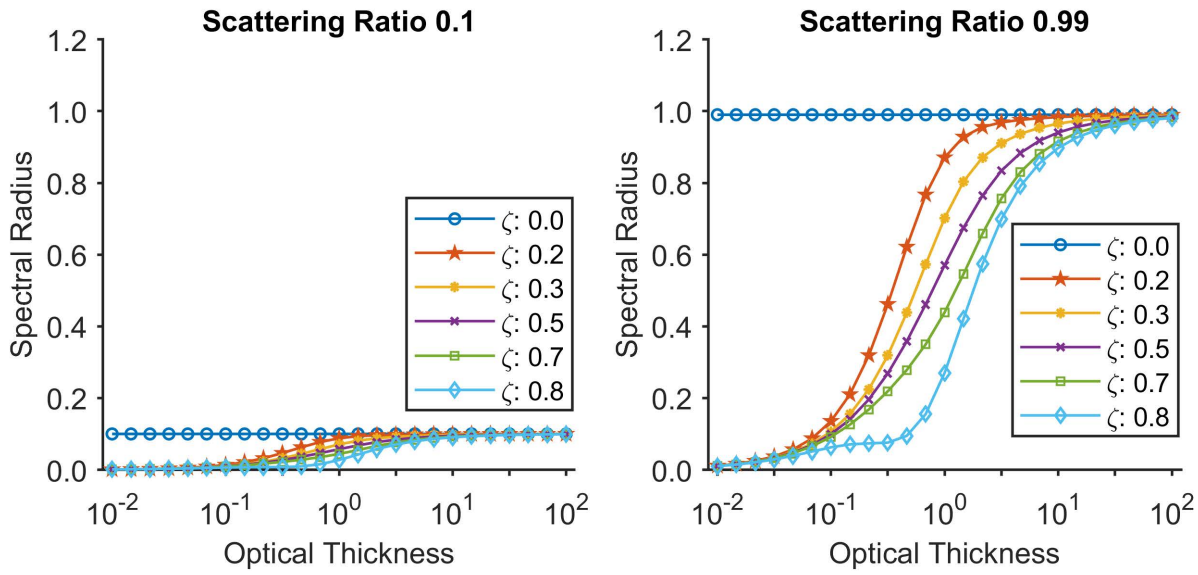

Figure 2.9 – Infinite slab test: a comparison of the spectral radius of several different spatial frequencies, in units of Hz, of the Fourier transformed solution of MOC for an infinite slab.

2.16.2 Comparison of finite and infinite domains

The importance of why it is necessary to investigate finite geometries, is explained in this section by comparing finite and infinite domains. The finite problem was created according to the specifications of the finite slab test case with the exception that only the standard iteration scheme is investigated. For an infinite domain, the Fourier analysis acts as an upper bound for the spectral radial analysis. The Fourier analysis has the general disadvantage that it can only analyze infinite geometries. For such problems the rate of convergence is dominated by the low spatial frequencies of the solution. In other words, the propagation of changes in the solution across the spatial domain. However, when the boundary incoming angular flux comes from the previous iteration and the reactor geometry is optically thin from boundary to boundary, the spectral radius is dominated by the transmission of the boundary angular flux, [32]. Since the infinite domain required by Fourier analysis is not a valid representation of such a case, these boundary contributions to the spectral radius cannot be observed by Fourier analysis. A comparison of the spectral radius of MOC is shown by Fig. 2.10 for an infinite geometry and a finite geometry with the boundary conditions,

$$\psi^{-,(n+1)}_{m,i,s} = \psi^{+,(n)}_{m,i',s'} \quad \forall\ (i',s') \in \Gamma_{Bnd}.$$

For the finite geometry, the spectral radii are close to 1 when the medium is optically thin. This is due to the spectral radius being dominated by the eigenvalue corresponding to the transmission of the previous iteration of the incoming angular flux when the medium is optically thin, [32]. This can be readily observed in the case when a beam of neutrons transverses a slab with the optical thickness $\Sigma_{g,i} l_{m,i,s}(\vec{r}_{s'}) \approx \epsilon/|\mu_m| = \epsilon_m$, where μ_m is

the cosine of the angular direction. By applying Eq. (2.47a) to this case, the following relation is achieved,

$$\psi^{+,(n+1)}_{m,i,s} = \sum_{s' \in S^-_{m,i,s}} T_{m,i,s,s'} \psi^{-,(n+1)}_{m,i,s'} + c_i \epsilon_m \phi^{(n)}_i,$$

$$\bar{\phi}^{(n+1)}_i = \sum_{m=1}^{M} w_m \left[\sum_{s' \in S^-_{m,i}} I_{m,i,s'} \psi^{-,(n+1)}_{m,i,s'} + c_i \epsilon_m \phi^{(n)}_i \right],$$

$$T_{m,i,s,s'} = \left| \hat{\Omega}_m \cdot \hat{n}_{i,s'} \right| \int_{\Gamma^-_{i,s'}} d^2 r_s \; e^{-\epsilon_m},$$

$$I_{m,i,s'} = \frac{1}{A_{i,s}} \left| \hat{\Omega}_m \cdot \hat{n}_{i,s'} \right| \int_{\Gamma^-_{i,s'}} d^2 r_s \; \frac{1 - e^{-\epsilon_m}}{\epsilon_m}.$$

It is important to note that the external source has been dropped from the above equation since they do not affect the convergence. The above equation shows that the spectral radius corresponds to the exponential term. A more detailed analysis of the convergence behavior for this type of transport iteration scheme has been done by Sanchez [32].

Conversely, when the medium is optically thick, the spectral radius is dominated by the scattering ratio as a result of the unconverged scattering source, [8]. This can be observed when the MOC equations are applied to the case when there is a beam traversing a slab, with isotropic scattering and with $e^{-\Sigma_{g,i} l_{m,i,s}(\vec{r}_{s'})} \approx \epsilon_m$,

$$\psi^{+,(n+1)}_{m,i,s} = \sum_{s' \in S^-_{m,i,s}} \epsilon_m + c_i \bar{\phi}^{(n)}_i.$$

Thus, MOC converges slowly for optically diffusive problems. In fact, as previously mentioned in Section 2.7.2, this convergence problem exists for all S_N transport schemes. Since the scattering ratio can be close to 1 in the moderator of a reactor, this convergence behavior is a significant performance limitation for the SI scheme.

Throughout the optical thickness and scattering ratio domain, the spectral radius of the finite problem is greater than that of the infinite problem. From Fig. 2.11, a rescaled version of Fig. 2.10, the spectral radius of the finite geometry is closer to 1 than the infinite geometry, even when the scattering ratio is set to 0.99. Thus, not only are finite geometries a more realistic representation of real-world problems, they are also more difficult to converge. This is why, finite geometries will be the focus of our analysis in future sections.

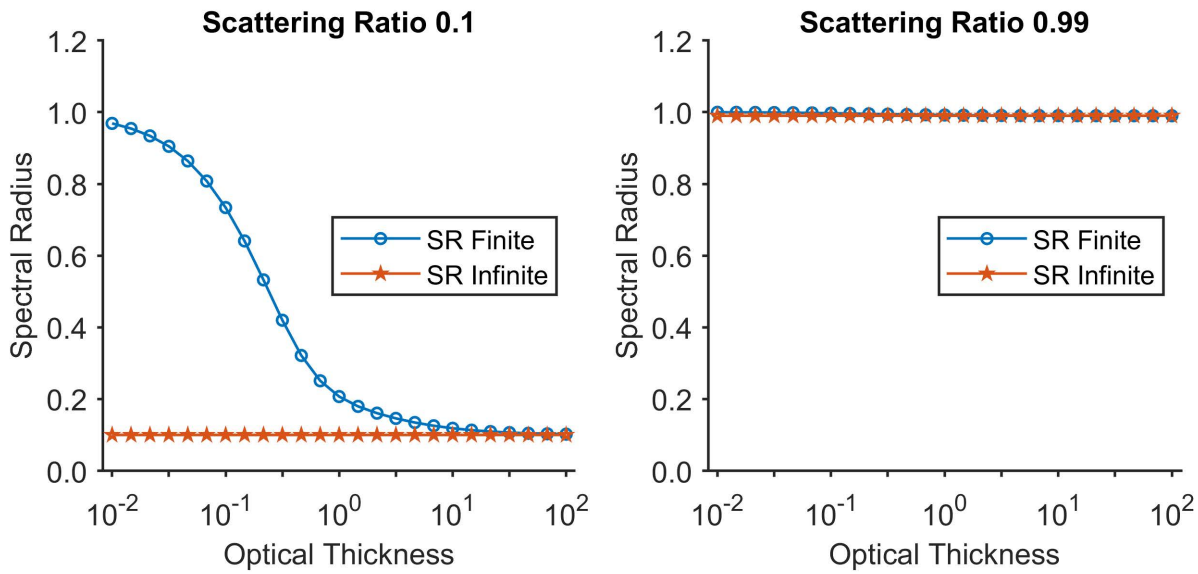

Figure 2.10 – Finite slab test versus infinite slab test: a comparison of the spectral radius (SR) of MOC for a finite (finite slab test) and infinite (infinite slab test) geometry.

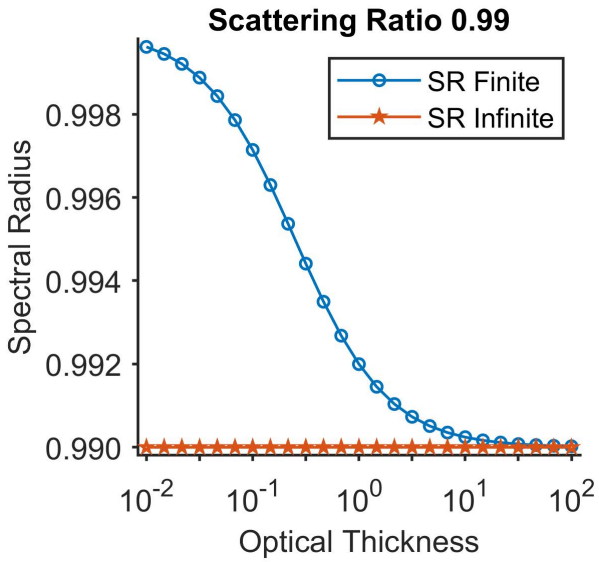

Figure 2.11 – A rescaled version of Fig. 2.10 to highlight the difference between the spectral radius (SR) of the finite (finite slab test) and infinite (infinite slab test) geometries.

2.16.3 Analysis of spatially decomposed geometries for parallel simulation

Parallelization of the angular domain is often not sufficient for computing large spatial meshes in an acceptable time frame. This problem is the result of too few directions relative to the number of spatial cells and the poor data locality involved in solving a single direction across a vast spatial mesh. In order to improve the opportunity for parallelism and the locality of the data, it is common to break up the spatial domain into subdomains that can be solved simultaneously. For highly spatially discretized problems, spatial decomposition is essential for achieving acceptable runtimes, as has been shown for the diamond difference transport equations by Zerr and Azmy, [40]. As a geometry is spatially decomposed into an increasing number of subdomains, the optical thickness of the subdomains decreases. As was shown in Section 2.16.2 for MOC, a decreasing optical thickness leads to a larger spectral radius. Since it is typically more desirable to analyze the convergence behaviour of the most difficult to converge cases, we have chosen to focus on the asymptotic case where each subdomain contains only 1 cell. A similar type of analysis has been carried out for the diamond difference transport equations by Dmitriy Anistratov and Yousry Azmy, [6]. In the case that each cell is solved simultaneously, SMOC take on the following form,

$$\psi^{+,(n+1)}_{g,m,i,s} = \sum_{s' \in S^{-}_{m,i,s}} T_{g,m,i,s,s'} \psi^{-,(n)}_{g,m,i,s'} + E_{g,m,i,s} Q^{(n)}_{g,m,i}, \tag{2.59a}$$

$$\bar{\phi}^{(n+1)}_{g,l,k,i} = \sum_{m=1}^{M} w_m Y_{l,k}(\hat{\Omega}_m) \left\{ \sum_{s \in S^{+}_{m,i}} I_{g,m,i,s} \psi^{-,(n)}_{g,m,i,s} + C_{g,m,i} Q^{(n)}_{g,m,i} \right\}, \tag{2.59b}$$

$$Q^{(n+1)}_{g,m,i} = \sum_{l=0}^{L} \sum_{k=-l}^{l} Y_{l,k}(\hat{\Omega}_m) \left[\Sigma_{sc,g,l,i} \bar{\phi}^{(n+1)}_{g,l,k,i} + Q^{(t)}_{Ext,g,l,k,i} \right], \tag{2.59c}$$

where all the interface angular fluxes are simultaneously updated at each iteration. Thus, the GJ scheme that was first introduced in Section 2.16, is defined by Eqs. (2.59a) to (2.59c). By comparing the standard scheme, Eqs. (2.47a) to (2.47c), and the GJ scheme, we see that they are distinguished by the iteration index of the interface angular flux that appears on the right-hand side of the system of equations.

In order to analyze the different convergence behaviour between the GJ and standard iteration scheme, we perform the finite slab test. The purpose of this test is to analyze the influence of the optical thickness and the scattering ratio on the spectral radius for the two different iteration schemes. The results of this analysis are shown in Fig. 2.12. For all cases, the spectral radius of the GJ operator is larger than that of the standard iteration scheme. For the case when the scattering ratio is set to 0.99, the difference between the two iterative schemes can be more clearly seen from the rescaled version of Fig. 2.12, i.e. Fig. 2.13.

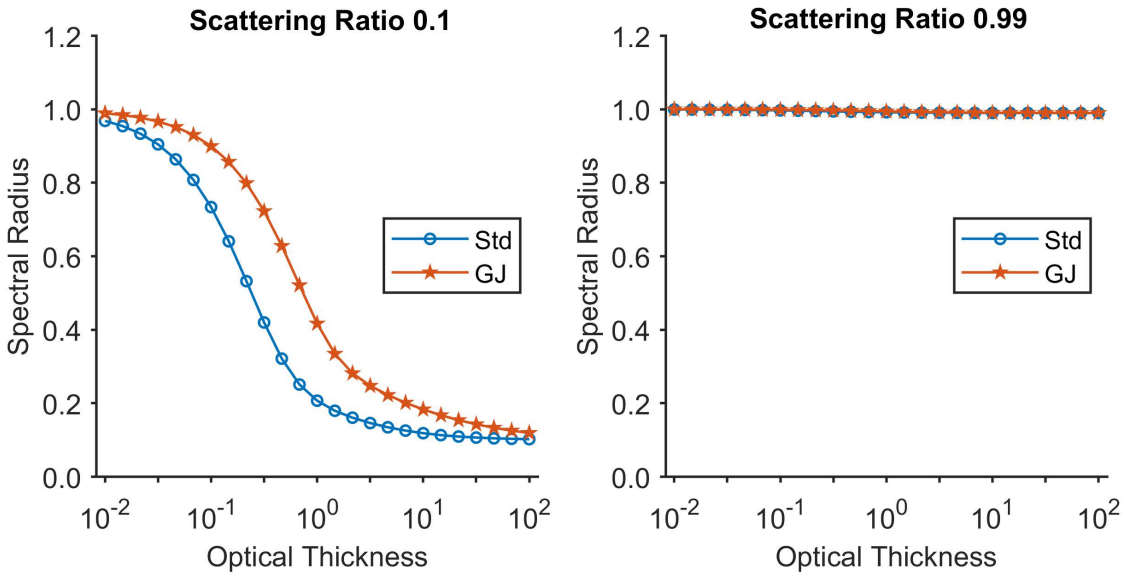

Figure 2.12 – Finite slab test: the spectral radius analysis of the GJ and standard (Std) iteration schemes of MOC for a finite slab.

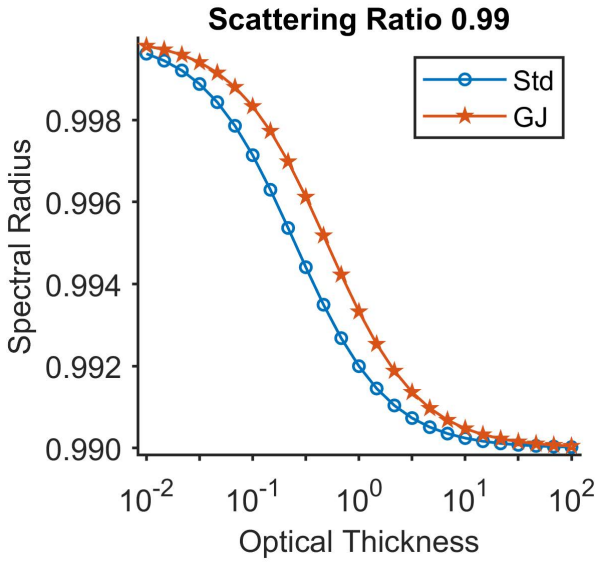

Figure 2.13 – A rescaled version of Fig. 2.12 to highlight the difference between the GJ and standard (Std) iteration schemes for a finite slab.

Next, we perform the spatial discretization test for the standard and the GJ scheme to get the results shown in Fig. 2.14. By comparing the results of the spectral radius analysis, a stark difference between the GJ and the standard iteration scheme occurs when the number of cells in the subdomain increases. As well, the spectral radius of the standard iteration scheme approaches the scattering ratio as the number of cells increases. This is expected by our previous observations since even when each cell is individually optically thin, the optical thickness of the subdomain is still increasing. In contrast, the spectral radius of the GJ remains the same since each interface incoming angular flux of cell i is from the previous iteration ($\psi^{-,(n)}_{m,i,s}$), resulting in the spectral radius being dependent on

the optical thickness of the cell instead of the subdomain.

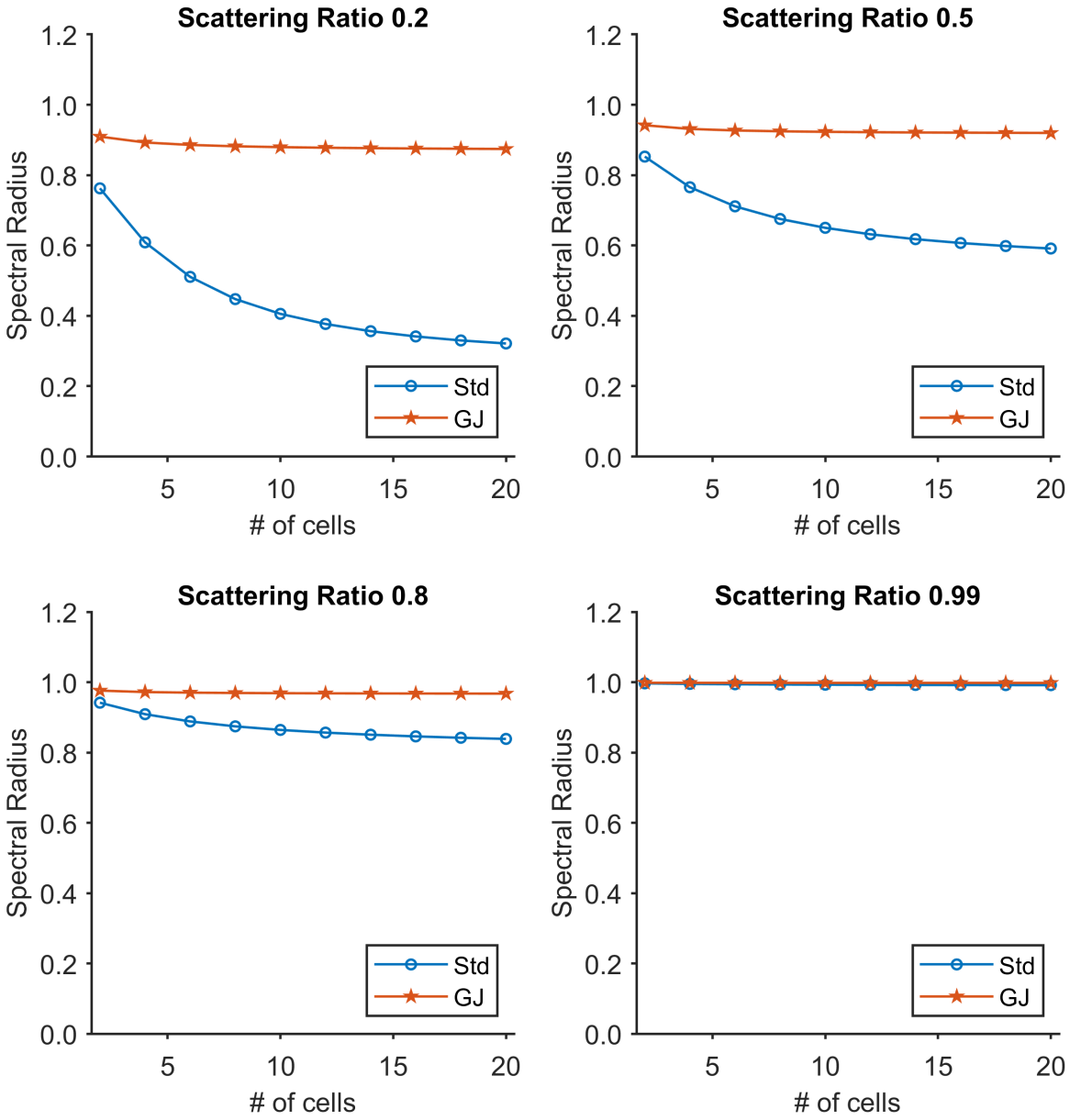

Figure 2.14 – Spatial discretization test: The spectral radius analysis of the GJ and standard (Std) iteration schemes of MOC for an increasing number of cells. The optical thickness of each cell has been fixed to a value of 0.1 mfp.

2.16.4 Numerical validation of the spectral radius analysis

In order to validate the convergence analysis, a comparison of the spectral radius obtained through our analytical formulation and those obtained from a numerical simulation, in this case IDT, is performed. However, the spectral radius of the transport operator is estimated using the data available from a numerical simulation. Starting with Eq. (2.55),

$$||\partial\mathbf{\Psi}^{(n+1)}|| = \rho^{(n+1)}||\partial\mathbf{\Psi}^{(0)}||, \quad \text{when } n \to \infty.$$

the approximation is made such that when n is sufficiently large, the spectral radius can be estimated using,

$$\rho \approx \left[\frac{||\mathbf{\Psi} - \mathbf{\Psi}^{(n+1)}||}{||\mathbf{\Psi} - \mathbf{\Psi}^{(0)}||} \right]^{1/(n+1)}. \tag{2.60}$$

It is important to note that Eq. (2.60) differs from the more commonly used approximation,

$$\rho \approx \frac{||\mathbf{\Psi}^{(n+1)} - \mathbf{\Psi}^{(n)}||}{||\mathbf{\Psi}^{(n)} - \mathbf{\Psi}^{(n-1)}||}. \tag{2.61}$$

Eq. (2.61) does not require knowledge of the final solution and can therefore easily used for any problem. However, for simple problems, where the final solution is easily obtained, we have chosen to use Eq. (2.60) since it avoids the numerical precision problems of Eq. (2.61) when the solution converges slowly.

Using Eq. (2.60), the results of the spectral radius analysis of the numerical simulation were compared to the analytical results for the finite slab test. The results of this comparison are shown in Fig. 2.15. The numerical results are almost exactly equivalent to those of the spectral analysis, for all cases, validating the analysis we have done up until this point. An exact match between the numerical and analytical results is not expected due to the limitations of numerical precision.

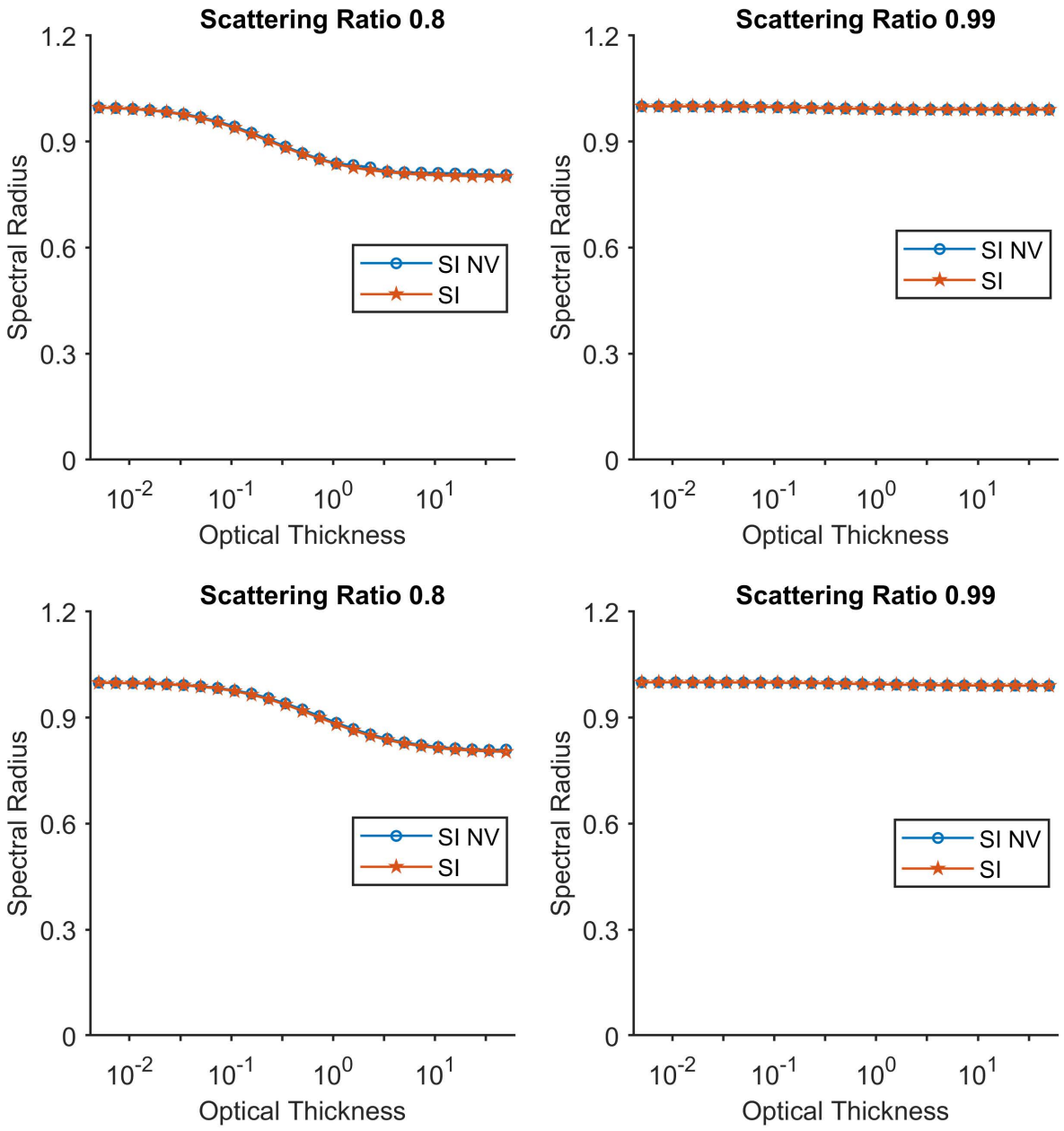

Figure 2.15 – Finite slab test: the numerical validation of the spectral radius analysis for a finite slab. NV is used to denote the numerical validation.

2.17 False convergence

During a numerical simulation it is important to qualify the distance of the current solution to the true solution, so that the program can determine when it can stop. However, since $\boldsymbol{\Psi}$ is not typically known, it is common to calculate the L_∞ norm of the change in the transport operator's unknowns, in order to qualify the state of convergence. By subtracting $\underline{\underline{\boldsymbol{L}}}\partial\boldsymbol{\Psi}^{(n+1)}$ from both sides of Eq. (2.52), and by rearranging we get,

$$\partial\boldsymbol{\Psi}^{(n+1)} = \left[\underline{\hat{\boldsymbol{I}}} - \underline{\underline{\boldsymbol{L}}}\right]^{-1} \underline{\underline{\boldsymbol{L}}} \left[\boldsymbol{\Psi}^{(n+1)} - \boldsymbol{\Psi}^{(n)}\right], \tag{2.62}$$

where $\underline{\hat{\boldsymbol{I}}}$ is the identity matrix. It is important to note that $\left[\underline{\hat{\boldsymbol{I}}} - \underline{\underline{\boldsymbol{L}}}\right]$ has been assumed to be invertible, in the above formulation. The difference between the $(n+1)$ iteration of the transport solution and the final solution linearly depends on the difference between the $(n+1)$ and the (n) iteration. However, the commonly made assumption that $\partial\boldsymbol{\Psi}^{(n+1)}$ will be converged within a similar tolerance as $\boldsymbol{\Psi}^{(n+1)} - \boldsymbol{\Psi}^{(n)}$ is not always accurate. By applying the approximation defined by Eq. (2.60) to Eq. (2.62), the following approximation is created,

$$||\partial\boldsymbol{\Psi}^{(n+1)}|| \approx \frac{\rho}{1-\rho}||\boldsymbol{\Psi}^{(n+1)} - \boldsymbol{\Psi}^{(n)}||. \tag{2.63}$$

As $\rho \to 1$, the ratio between $||\partial\boldsymbol{\Psi}^{(n+1)}||$ and $||\boldsymbol{\Psi}^{(n+1)} - \boldsymbol{\Psi}^{(n)}||$ rapidly increases. Thus, when the solution converges slowly, $||\partial\boldsymbol{\Psi}^{(n+1)}||$ will converge within a much larger tolerance than $||\boldsymbol{\Psi}^{(n+1)} - \boldsymbol{\Psi}^{(n)}||$, causing the final result of the simulation to not achieve the desired precision. This phenomenon is referred to as false convergence. Therefore, not only does a large spectral radius mean that the iterative scheme will be computationally expensive to solve, it also means that precision of the final result is not guaranteed for the set convergence criteria. Since many problems of interest can suffer from these two effects, there is a lot of interest in creating methods that will accelerate the convergence of the source iterations.

2.18 Acceleration methods

As discussed in Section 2.14, the rate of convergence of a linear iterative scheme is determined by the spectral radius of the iterative operator. Moreover, the form of the iterative operator is determined by the discretization scheme and the iteration indexing of the unknowns. Thus, in order to have an acceptable rate of convergence, one must be careful when choosing which discretized transport operator to apply and which iteration scheme to use for solving the discretized transport operator. As an example, in the Section 2.16.2 the coefficient associated to $\bar{\phi}^{(n)}_{g,l,k,i}$ was shown to approach a value of 1 when the geometry was optically diffusive for the step MOSC equation. An obvious remedy to this problem would be to include $\bar{\phi}_{g,l,k,i}$ in the $(n+1)$ iteration. However, this would lead to a denser system of equations that is more computationally expensive to apply and less parallelizable. Instead, operators referred to as acceleration methods, are applied to iterative schemes to isolate and converge the terms with the greatest impact on convergence. The typical accelerated iterative scheme is shown in Fig. 2.16 below. The $(n+1/2)$ iteration index has been introduced to denote the current state of the solution after the application of the source iteration.

Figure 2.16 – The typical accelerated iterative scheme.

2.18.1 Comparison of acceleration method types

There are two main types of acceleration methods, linear methods and nonlinear methods, [25]. Since the spectral radius of a linear method is independent of the solution, its performance and stability can be exactly determined. In contrast, the stability and performance of nonlinear methods are typically only validated when the solution is near convergence. This certainty is one of the primary advantages of linear methods.

However, linear methods are required to use the same mesh as transport, which entails high computational and memory requirements. As well, by using a separate mesh from the transport operator, nonlinear methods can isolate and solve for the more troublesome to converge, low spatial frequencies, [3]. Nonlinear methods do require the transport operator to cumulate data necessary for construction of the nonlinear operator and solution. This cumulation increases the computational cost of each transport iteration and typically requires modifications to the transport algorithm.

Another drawback of linear methods is that they require knowledge of the transport operator in order to ensure stability. Nonlinear operators instead, rely on coefficients, created from the current iteration of the solution, to ensure the existence of the fixed point, [4]. By construction, the solution of the acceleration equation must reproduce the transport solution projected on to the coarse space once convergence has been reached. This requirement is commonly met by ensuring that the acceleration operator is immediately satisfied by the final solution and by ensuring that the prolongation operator, is a pseudo inverse of the projection from the transport to acceleration solution space, for all iterations of the transport solution. This enforced consistency allows balance-based nonlinear methods to work with any conservative numerical transport scheme. For these reasons we have chosen to create and implement nonlinear acceleration methods in our library.

2.18.2 Strategy for creating nonlinear methods

As mentioned in Section 2.7.1, the P_N approximation is efficient for representing the flux in optically thick, diffusive media. Thus, a common starting point for building a nonlinear acceleration method for S_N transport schemes is to project the integral transport equations over a P_0 angular space. To do this, $\int_{4\pi} d^2\hat{\Omega}\ Y_{0,0}(\hat{\Omega}) \int_{D_i} d^3r$ is applied to Eq.

(2.12a) to get what is commonly referred to as the balance equation,

$$\sum_{s \in S_i} A_{i,s} J_{g,i,s} + [\Sigma_{g,i} - \Sigma_{sc,g,0,i}] V_i \bar{\phi}_{g,0,0,i} = Q_{Ext,g,i} V_i, \tag{2.64a}$$

$$Q_{Ext,g,i} = \int_{4\pi} d^2\hat{\Omega}\ Y_{0,0}(\hat{\Omega}) \int_{D_i} d^3r\ \frac{Q_{Ext,g}(\hat{\Omega}, \vec{r})}{V_i}, \tag{2.64b}$$

and

$$J_{g,i,s} = \int_{4\pi} d^2\hat{\Omega}\ Y_{0,0}(\hat{\Omega}) \int_{\Gamma_{i,s}} d^2r_s\ \hat{\Omega} \cdot \hat{n}_{i,s} \frac{\psi_g(\hat{\Omega}, \vec{r})}{A_{i,s}}, \tag{2.64c}$$

where $J_{g,i,s}$ is the interface current in the direction of the outward normal and $A_{i,s}$ is the surface area. The balance equation is only a function of the interface currents and the scalar flux. However, this is still too many unknowns to be constrained by the balance equation alone. Thus, in order to close Eq. (2.64a) it is necessary to define the relationship between the scalar flux and the surface current. There is no equation that exactly defines the relationship between the current and the scalar flux, without bringing us back to the transport operator. Thus, an approximation of the relationship between the flux and current that accurately represents the relationship defined by the transport operator must be used instead. The more consistent the chosen relation is with that of the transport operator for the case being solved, the closer the solution of the acceleration operator will be to that of the transport operator. However, as the number of degrees of freedom of the chosen relation increases, so does the size of the operator that must be inverted. For these reasons, the choice of the scalar flux and surface current relationship is one of the most important aspects affecting the performance of an acceleration method.

2.18.3 Coarse mesh

As previously mentioned, one of the advantages of nonlinear methods is that they can operate on a coarser mesh than the transport operator. The coarse angular domain is fixed by the definition of the acceleration method. However, the level of coarsening of the spatial and energy domains can be set in order to optimize the performance of the acceleration method. The angular moments of the flux and the partial currents for a coarse group ($\tilde{g}$), a coarse cell ($\tilde{i}$) and a coarse surface ($\tilde{s}$) are defined as:

$$\phi_{\tilde{g},l,k,\tilde{i}} = \frac{\sum_{g \in \tilde{g}} \sum_{i \in \tilde{i}} \phi_{g,l,k,i} V_i}{V_{\tilde{i}}},$$

$$J^{\pm}_{\tilde{g},\tilde{i},\tilde{s}} = \frac{\sum_{g \in \tilde{g}} \sum_{i \in \tilde{i}} \sum_{s \in \tilde{s}} J^{\pm}_{g,i,s} A_{i,s}}{A_{\tilde{i},\tilde{s}}},$$

where

$$V_{\tilde{i}} = \sum_{i \in \tilde{i}} V_i,$$

and

$$A_{\tilde{i},\tilde{s}} = \sum_{i \in \tilde{i}} \sum_{s \in \tilde{s}} A_{i,s}.$$

Likewise, the cross sections of the coarse mesh can be defined as:

$$\Sigma_{\tilde{g},\tilde{i}} = \frac{\sum_{g\in\tilde{g}}\sum_{i\in\tilde{i}}\Sigma_{g,i}\phi_{g,0,0,i}V_i}{\phi_{\tilde{g},0,0,\tilde{i}}V_{\tilde{i}}}, \tag{2.65a}$$

$$\tilde{\Sigma}^j_{f,\tilde{g},\tilde{i}} = \frac{\sum_{g\in\tilde{g}}\sum_{i\in\tilde{i}}\tilde{\Sigma}^j_{f,g,i}\phi_{g,0,0,i}V_i}{\phi_{\tilde{g},0,0,\tilde{i}}V_{\tilde{i}}}, \tag{2.65b}$$

$$\Sigma_{sc,\tilde{g},\tilde{g}',0,\tilde{i}} = \frac{\sum_{g\in\tilde{g}}\sum_{g'\in\tilde{g}'}\sum_{i\in\tilde{i}}\Sigma_{sc,g,g',0,i}\phi_{g',0,0,i}V_i}{\phi_{\tilde{g}',0,0,\tilde{i}}V_{\tilde{i}}}, \tag{2.65c}$$

and

$$\chi^j_{\tilde{g},\tilde{i}} = \frac{\sum_{i\in\tilde{i}}V_i\sum_{g\in\tilde{g}}\chi^j_{g,i}\sum_{g'=1}^{G}\tilde{\Sigma}^j_{f,g',i}\phi_{g',0,0,i}}{\sum_{i\in\tilde{i}}V_i\sum_{g'=1}^{G}\tilde{\Sigma}^j_{f,g',i}\phi_{g',0,0,i}}. \tag{2.65d}$$

By using a coarse mesh, nonlinear methods can substantially reduce the degrees of freedom. Moreover, a coarse mesh allows the acceleration method to converge the eigenvectors associated with the propagation of changes across the global energy and spatial domain.

2.18.4 Diffusion approximation

Previously, in Section 2.16.1.1 we noted that the spectral radius of the MOC operator for a geometry that was approaching infinity corresponded to the zeroth spatial frequency and was set by the scattering ratio. This means that MOC has difficulty converging eigenmodes with weak spatial dependence for problems with large scattering ratios. Moreover, Adams and Larsen have shown with a more rigorous derivation that the spectral radius of the transport source iteration has weak angular dependence as well, [4]. Thus, in order to effectively accelerate the source iteration of the transport equation for big geometries with a large scattering ratio, the closure relations of the acceleration method should be most accurate when the spatial and angular dependence are weak. The diffusion approximation satisfies these criteria very well. As a result, it is has been used as a closure relation for many acceleration methods such as the Diffusive Synthetic Acceleration (DSA) scheme and CMFD. An example produced by Adams and Larsen of how effectively the diffusion approximation reduces the spectral radius of the source iteration is shown in Fig. 2.17. The spectral radius corresponding to the zeroth spatial frequency has been eliminated.

Another benefit of the diffusion approximation is that it defines the surface current as a linear function of the scalar flux, allowing for the surface current to be eliminated from the solution. This reduction in dimension makes these acceleration operators efficient to invert. However, the diffusion approximation is not accurate for all cases and can lead to poor acceleration or even instability in some cases. In order to remove the instability caused by directly relating the current to the scalar flux, one can instead relate the scalar flux to the partial surface currents, which are defined by

$$J^{\pm}_{g,i,s} = \sum_{m\in M^{\pm}_{i,s}} w_m \left|\hat{\Omega}_m\cdot\hat{n}_s\right| \psi^{\pm}_{g,m,i,s}. \tag{2.66}$$

By relating the scalar flux to the partial currents, acceleration methods (such as P-CMFD) are typically able to achieve improved stability, as has been proven by both numerical

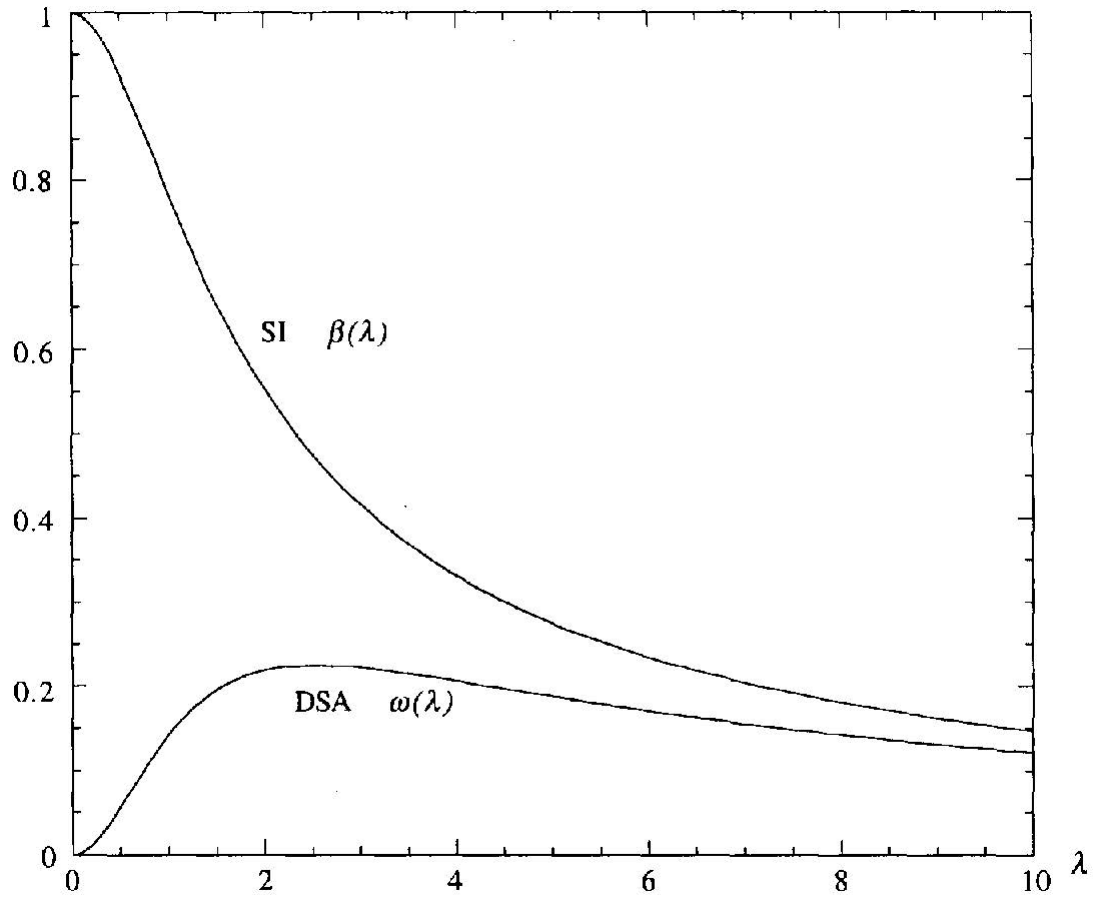

Figure 2.17 – A comparison, produced by Adams and Larsen, of the spectral radius of SI ($\beta(\lambda)$) and DSA ($\omega(\lambda)$) for the continuous transport equation solved over a 1D slab when c=1, where λ is the spatial frequency, [4]

benchmarks and Fourier analysis, [39].

Another problem with the diffusion approximation is that it introduces spatial derivatives into the acceleration equations, requiring knowledge of the spatial distribution of the flux. The accuracy at which the acceleration operator preserves the spatial distribution of the transport solution strongly determines the accuracy and thus the effectiveness, of the acceleration operator. Conversely, the more the dimension of the acceleration solution is reduced, the more numerically efficient the operator is to solve. The simplest way to ensure the accuracy of the spatial representation of the coarse mesh is to use a similar discretization as the transport operator. To improve the effectiveness of CMFD, the spatial discretization of the coarse mesh should typically be chosen to be as fine as can be stably solved. However, a coarse energy discretization can be employed by CMFD without greatly diminishing its effectiveness [26]. However, within each discrete spatial region, the flux is assumed to be constant everywhere, with discontinuous values at the surfaces. While this approximation offers a large dimension reduction from transport, it can be an inaccurate representation of the spatial distribution of the transport flux.

2.18.5 Pre-existing acceleration methods

Due to their popularity, the Coarse Mesh Rebalance method (CMR), [13], and CMFD were chosen as the reference acceleration methods that we would compare our new methods against. The form of the scalar Flux and Current Closure Relationship (FCCR) for these methods comes from diffusion theory, [12],

$$J_g(\vec{r}) = -\frac{1}{3\Sigma_{tr,g}(\vec{r})}\nabla\phi_g(\vec{r}), \tag{2.67a}$$

$$\Sigma_{tr,g}(\vec{r}) = \Sigma_g(\vec{r}) - \Sigma_{sc,1,g}(\vec{r}), \tag{2.67b}$$

where $\Sigma_{tr,g}(\vec{r})$ is the transport cross section. However, it is important to keep in mind that these methods are ad hoc solutions, meaning that intuition rather than theory is used to define the corrective nonlinear coefficients.

Starting from Eq. (2.64a), **CMR** closes the system of equations by using the following FCCR,

$$J^+_{g,i,s} = \tilde{D}_{g,i,s}\bar{\phi}_{g,0,0,i}, \tag{2.68}$$

where $\tilde{D}_{g,i,s}$ is the nonlinear coefficient that ensures Eq. (2.68) is satisfied by the transport solution. The simplicity of its implementation and its relative effectiveness at the time of its creation has resulted in CMR being a commonly implemented method. However, in recent years the popularity of CMR has waned due to the comparatively better performance offered by CMFD, [41].

Like CMR, the starting point of CMFD is Eq. (2.64a). However, there are many different variants of CMFD, with each variant being distinguished by its FCCR. For consistency, we chose to implement in the library the same formulations of CMFD that exist in IDT, namely, MCNH (named after the creators Moon, Cho, Noh and Hong), [37], pCMFD (Partial current CMFD), [39], and AAF (Aragones and Ahnert Formulation), [7].

The **MCNH** variant, distinguished by the FCCR,

$$J_{g,i,s} = -D_{g,i,s}[\phi_{g,i,s} - \bar{\phi}_{g,0,0,i}] + \tilde{D}_{g,i,s}[\phi_{g,i,s} + \bar{\phi}_{g,0,0,i}], \tag{2.69a}$$

$$\phi_{g,i,s} = \sum_{m \in M^-_{i,s}} w_m \psi^-_{g,m,i,s} + \sum_{m \in M^+_{i,s}} w_m \psi^+_{g,m,i,s}, \tag{2.69b}$$

is one of the most performant variants of CMFD. $D_{g,i,s}$ is defined by,

$$D_{g,i,s} = \frac{1}{3\Sigma_{tr,g}\Delta_{i,s}},$$

where $\Delta_{i,s}$ is the distance between the barycenter of cell i and surface s. By enforcing the continuity of $\phi_{g,i,s}$ and $J_{g,i,s}$ at the interface,

$$J_{g,i',s'} = J_{g,i,s},$$
$$\phi_{g,i',s'} = \phi_{g,i,s},$$

where i' and s' are the adjacent cell and surface index, for Eq. (2.69a) we get,

$$\begin{aligned} &- D_{g,i',s'}[\bar{\phi}_{g,0,0,i'} - \phi_{g,i,s}] + \tilde{D}_{g,i',s'}[\bar{\phi}_{g,0,0,i'} + \phi_{g,i',s'}] \\ &= -D_{g,i,s}[\phi_{g,i,s} - \bar{\phi}_{g,0,0,i}] + \tilde{D}_{g,i,s}[\phi_{g,i,s} + \bar{\phi}_{g,0,0,i}], \end{aligned} \tag{2.70}$$

Solving Eq. (2.70) for $\phi_{g,i',s'}$ leads to,

$$\phi_{g,i,s} = \frac{[D_{g,i',s'} - \tilde{D}_{g,i',s'}]\bar{\phi}_{g,0,0,i'} + [D_{g,i,s} + \tilde{D}_{g,i,s}]\bar{\phi}_{g,0,0,i}}{[D_{g,i',s'} + \tilde{D}_{g,i',s'} + D_{g,i,s} - \tilde{D}_{g,i,s}]} \tag{2.71}$$

Substituting Eq. (2.71) into Eq. (2.69a) eliminates $\phi_{g,i,s}$ as an unknown of MCNH and gives us,

$$\begin{aligned} J_{g,i,s} = &\left\{ \frac{[-D_{g,i,s} + \tilde{D}_{g,i,s}][D_{g,i,s} + \tilde{D}_{g,i,s}]}{[D_{g,i',s'} + \tilde{D}_{g,i',s'} + D_{g,i,s} - \tilde{D}_{g,i,s}]} + [D_{g,i,s} + \tilde{D}_{g,i,s}] \right\} \bar{\phi}_{g,0,0,i} \\ &+ \frac{[-D_{g,i,s} + \tilde{D}_{g,i,s}][D_{g,i',s'} - \tilde{D}_{g,i',s'}]}{[D_{g,i',s'} + \tilde{D}_{g,i',s'} + D_{g,i,s} - \tilde{D}_{g,i,s}]} \bar{\phi}_{g,0,0,i'} \end{aligned}$$

Although MCNH is a popular choice due to its performance, it also suffers from instability in optically diffusive mediums.

pCMFD is commonly used for its improved stability. The FCCR of pCMFD is defined by ,

$$J^+_{g,i,s} = -\hat{D}_{g,i,i',s,s'}[\bar{\phi}_{g,0,0,i'} - \bar{\phi}_{g,0,0,i}] + \tilde{D}_{g,i,s}\bar{\phi}_{g,0,0,i},$$
$$\hat{D}_{g,i,i',s,s'} = \frac{D_{g,i,s}D_{g,i',s'}}{D_{g,i,s} + D_{g,i',s'}},$$

where i' and s' are the adjacent cell and surface index, respectively.

AAF is the final variant of CMFD implemented in the library. The FCCR of **AAF** is defined by,

$$J_{g,i,s} = -D_{g,i,s}[\phi_{g,i,s} - \bar{\phi}_{g,0,0,i}] + \tilde{D}_{g,i,s}\phi_{g,i,s}.$$

AAF is one of the older variants. Thus, like CMR it is of interest primarily for the purpose of analyzing the progression of newer methods.

2.19 Spectral radius and Fourier analysis of nonlinear operators

In order to examine the convergence of nonlinear methods such as CMR and CMFD, we begin with a generic nonlinear iterative scheme defined by

$$\boldsymbol{\Psi}^{(n+1)} = \boldsymbol{S}(\boldsymbol{\Psi}^{(n)}) \tag{2.72}$$

where $\boldsymbol{S}(\boldsymbol{\Psi}^{(n)})$ is a nonlinear operator. At the final solution $(\boldsymbol{\Psi})$, the following relation must hold.

$$\boldsymbol{\Psi} = \boldsymbol{S}(\boldsymbol{\Psi}).$$

By setting $\boldsymbol{\Psi}^{(n)}$ to be very close to $\boldsymbol{\Psi}$ and by performing a Taylor series expansion of $\boldsymbol{S}(\boldsymbol{\Psi}^{(n)})$ at the point $\boldsymbol{\Psi}$, we get,

$$\boldsymbol{S}(\boldsymbol{\Psi}^{(n)}) = \boldsymbol{S}(\boldsymbol{\Psi}) + \frac{d\boldsymbol{S}(\boldsymbol{\Psi}^{(n)})}{d\boldsymbol{\Psi}^{(n)}}\bigg|_{\boldsymbol{\Psi}^{(n)}=\boldsymbol{\Psi}} (\boldsymbol{\Psi}^{(n)} - \boldsymbol{\Psi}) + o(\boldsymbol{\Psi}^{(n)} - \boldsymbol{\Psi})^2, \tag{2.73}$$

where $\boldsymbol{S}(\boldsymbol{\Psi}^{(n)})$ is assumed to be differentiable, and $o(\boldsymbol{\Psi}^{(n)} - \boldsymbol{\Psi})^2$ is the truncation error of the expansion. By applying the expansion defined by Eq. (2.73) to Eq. (2.72) we get,

$$\boldsymbol{\Psi}^{(n+1)} = \boldsymbol{S}(\boldsymbol{\Psi}) + \frac{d\boldsymbol{S}(\boldsymbol{\Psi}^{(n)})}{d\boldsymbol{\Psi}^{(n)}}\bigg|_{\boldsymbol{\Psi}^{(n)}=\boldsymbol{\Psi}} (\boldsymbol{\Psi}^{(n)} - \boldsymbol{\Psi}).$$

Since $\boldsymbol{\Psi} = \boldsymbol{S}(\boldsymbol{\Psi})$, we obtain,

$$(\boldsymbol{\Psi}^{(n+1)} - \boldsymbol{\Psi}) = \frac{d\boldsymbol{S}(\boldsymbol{\Psi}^{(n)})}{d\boldsymbol{\Psi}^{(n)}}\bigg|_{\boldsymbol{\Psi}^{(n)}=\boldsymbol{\Psi}} (\boldsymbol{\Psi}^{(n)} - \boldsymbol{\Psi}).$$

Substituting in the error term $\epsilon^{(n+1)} = (\boldsymbol{\Psi}^{(n+1)} - \boldsymbol{\Psi})$, the Jacobian is defined as,

$$\epsilon^{(n+1)} = \underline{\underline{\tilde{\boldsymbol{J}}}}(\boldsymbol{\Psi})\epsilon^{(n)}, \tag{2.74a}$$

$$\underline{\underline{\tilde{\boldsymbol{J}}}}(\boldsymbol{\Psi}) = \frac{d\boldsymbol{S}(\boldsymbol{\Psi}^{(n)})}{d\boldsymbol{\Psi}^{(n)}}\bigg|_{\boldsymbol{\Psi}^{(n)}=\boldsymbol{\Psi}}. \tag{2.74b}$$

In the above methodology we have linearized $\boldsymbol{S}(\boldsymbol{\Psi}^{(n)})$ around the fixed point. Thus, the Fourier analysis can proceed exactly the same ways as in Section 2.15. However, in general $\boldsymbol{S}(\boldsymbol{\Psi}^{(n)})$ is a nonlinear operator, which means that the convergence behaviour of the solution depends on the current iteration of the solution. Thus, in order to ensure the stability of a nonlinear operator, one would have to analyze the spectral radius across the possible solution space. As mentioned in Section 2.18 this is one of the primary drawbacks of nonlinear operators.

In order to analyze the spectral radius, when the current iteration of the solution is far from convergence, or when the final solution is not known, the following approximation is commonly used,

$$\begin{aligned}(\boldsymbol{\Psi}^{(n+1)} - \boldsymbol{\Psi}^{(n)}) &= \underline{\underline{\tilde{\boldsymbol{J}}}}(\boldsymbol{\Psi}^{(n)})(\boldsymbol{\Psi}^{(n)} - \boldsymbol{\Psi}^{(n-1)}),\\ \rho(\underline{\underline{\tilde{\boldsymbol{J}}}}(\boldsymbol{\Psi}^{(n)})) &\approx \frac{||\boldsymbol{\Psi}^{(n+1)} - \boldsymbol{\Psi}^{(n)}||}{||\boldsymbol{\Psi}^{(n)} - \boldsymbol{\Psi}^{(n-1)}||}.\end{aligned} \tag{2.75}$$

It is important to note that Eq. (2.75) is a generalization of Eq. (2.61) for nonlinear operators.

2.20 Convergence analysis of MCNH and CMR

In this section, we apply the three tests outlined in Section 2.16 to analyze the convergence behavior of CMFD and CMR. For some acceleration methods, such as CMFD, the coarsening of the spatial domain can also improve the stability of the method, [26]. Thus, in order to examine the convergence behavior of the acceleration methods in the most difficult to converge scenarios, we have chosen for the coarse mesh to be the same as the fine mesh.

To begin, the results of the Fourier analysis of CMR using the infinite slab test, are shown in Fig. 2.18. When the scattering ration is 0.1 or when the cell optical thickness is less than $\sim$ 0.2 mfp, the spectral radius of CMR is less than SI. This is a result of CMR reducing the eigenvalue of the eigenmode corresponding to the zeroth frequency to zero. However, when the optical thickness is greater than 0.2 and the scattering ratio is 0.99 the spectral radius of CMR is larger than SI, and in fact larger than 1. This instability comes from the eigenvalues of the higher spatial frequencies increasing as a result of the application of CMR.

A Fourier analysis of the MCNH variant of CMFD using the infinite slab test is shown by Fig. 2.19. Like CMR, when the scattering ration is 0.1 or when the cell optical thickness is less than $\sim$ 1 mfp, the spectral radius of MCNH is less than SI. However, MCNH also displays instability when the scattering ratio is 0.99 and the optical thickness is larger than 1 mfp. In the case of MCNH, this instability is brought about by its use of a step constant approximation of the flux inside a cell. When the medium is optically thick, large spatial variations in the flux occur which cannot be accurately represented using a step constant approximation. This leads to the stimulation of the eigenmodes associated to high spatial frequencies of the flux. Moreover, when the scattering ratio is high, the scalar flux has a large impact on the spectral radius, as explained in Section 2.16.1.1. A direct comparison of the spectral radius of MCNH and CMR can be seen in Fig. 2.20. The spectral radius of MCNH is less than that of CMR everywhere that MCNH is stable.

Next, the spectral radius analysis of the finite slab test is applied to MCNH and CMR. the results of this analysis are shown by Fig. 2.21. When the optical thickness of the medium is less than 1 or the scatter ratio is 0.1, MCNH has the same spectral radius as CMR or less. However, when the scattering ratio is 0.99 and the optical thickness is greater than 1, MCNH rapidly becomes unstable. In order to investigate geometries that are spatially decomposed for parallel computation, Fig. 2.21 also shows the results of the Gauss-Jacobi (GJ) scheme, described in Section 2.16. In the case when the GJ transport operator is used and the medium has a scattering ratio of 0.99, the spectral radius of MCNH is the same or larger than that of CMR.

Finally, the results of the spatial discretization test are shown by Fig. 2.22 for SI, MCNH and CMR. The spectral radius of CMR reaches a minimum at around 6 cells, after which CMR rapidly becomes unstable. The spectral radius of MCNH decreases as the number of cells increases and is less than the spectral radius of SI for all cases.

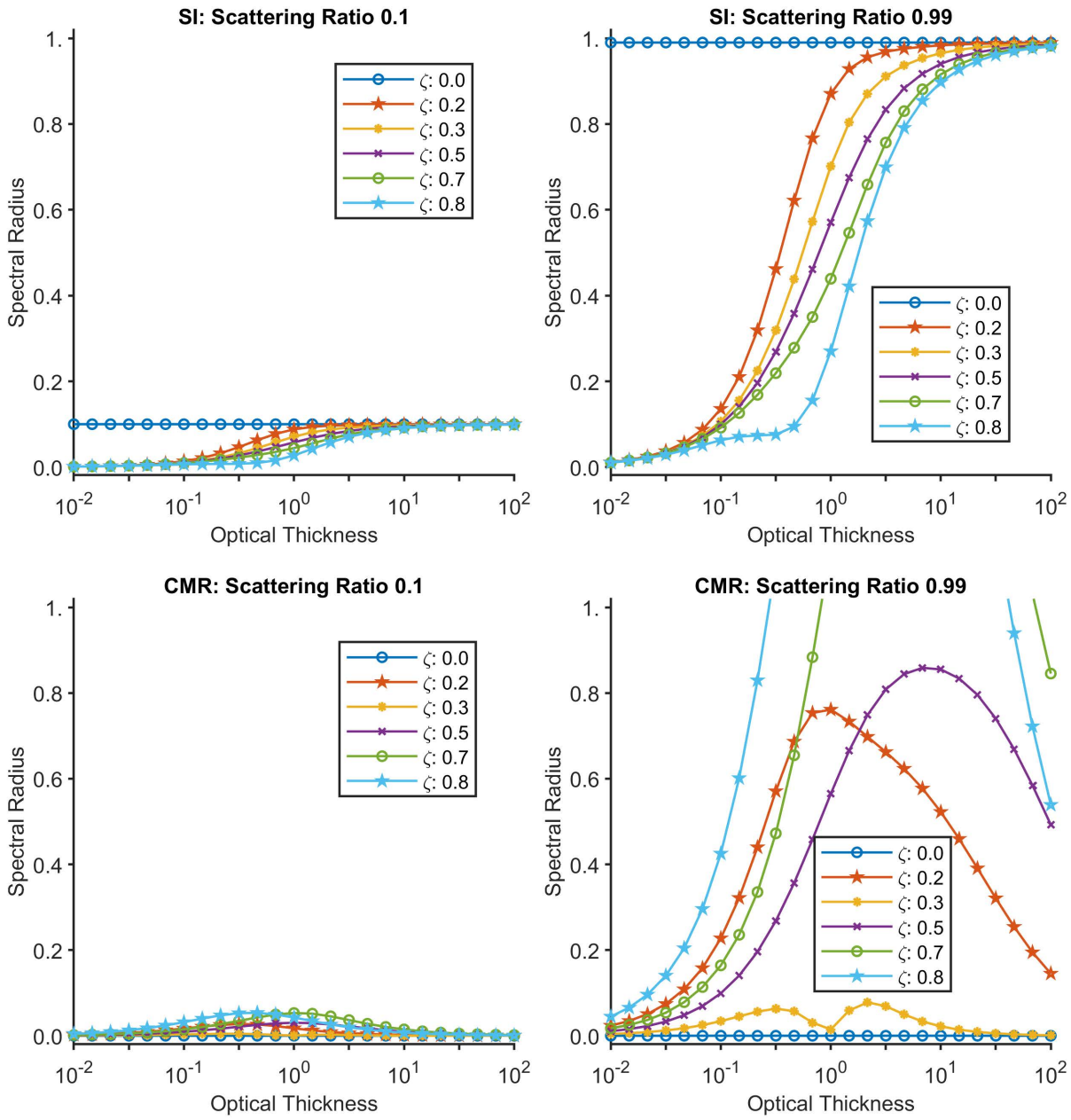

Figure 2.18 – Infinite slab test: a comparison of the spectral radius for several different spatial frequencies, in units of Hz, of the Fourier transformed unknowns of SI and accelerated by CMR, for an infinite slab .

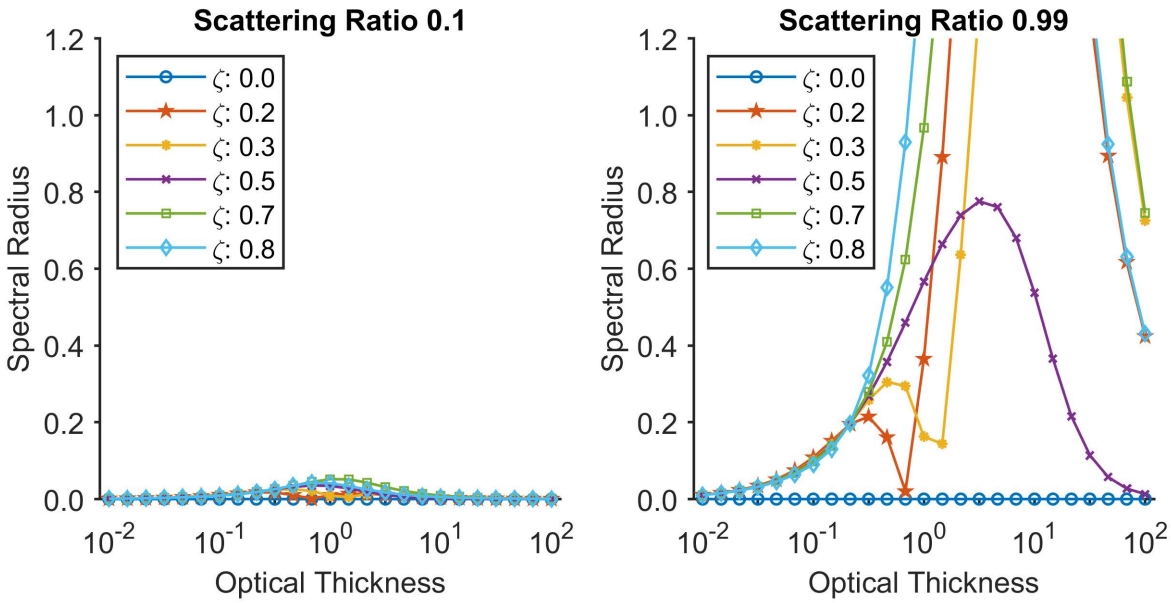

Figure 2.19 – Infinite slab test: a comparison of the spectral radius of several different spatial frequencies, in units of Hz, of the Fourier transformed unknowns of MOC accelerated by MCNH for an infinite slab.

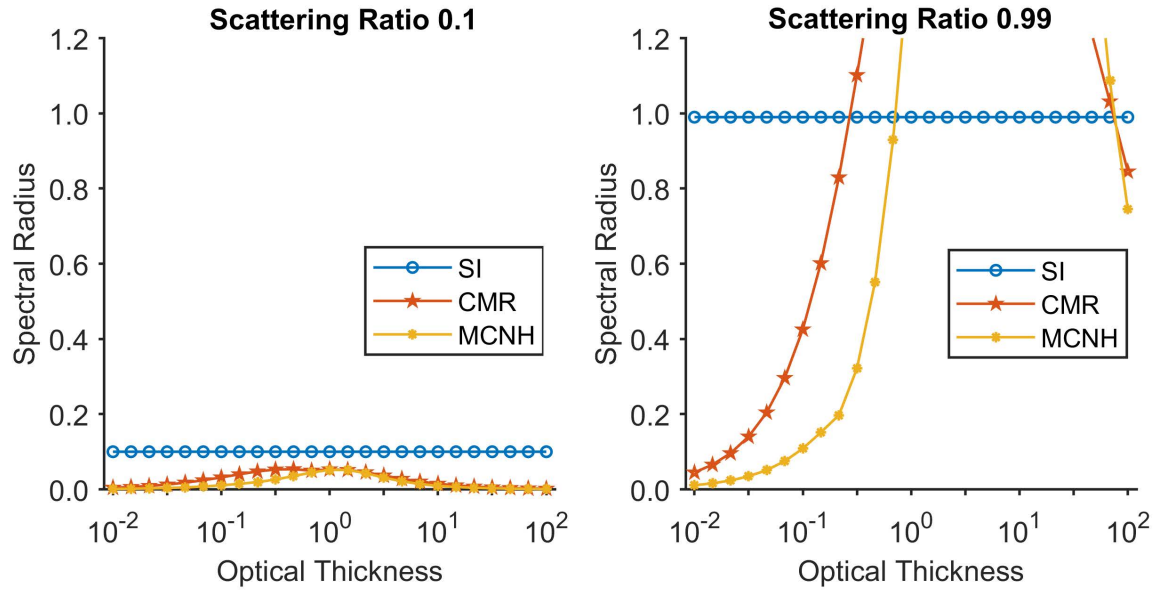

Figure 2.20 – Infinite slab test: a comparison of the spectral radius of SI, MOC accelerated by CMR and MOC accelerated by MCNH for an infinite slab.

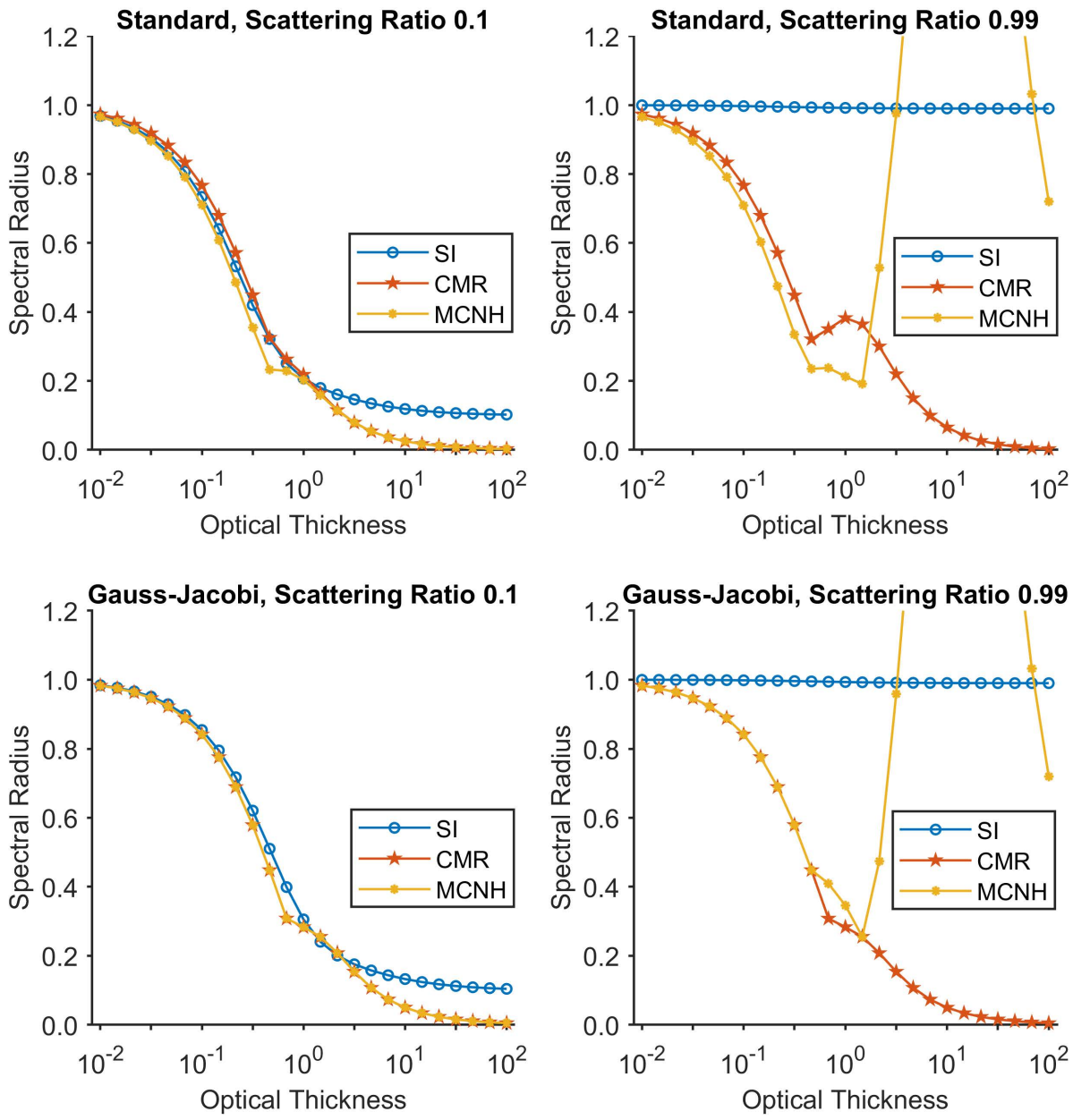

Figure 2.21 – finite slab test: a comparison of the spectral radius analysis of SI, MOC accelerated by CMR and MOC accelerated by MCNH for a finite slab.

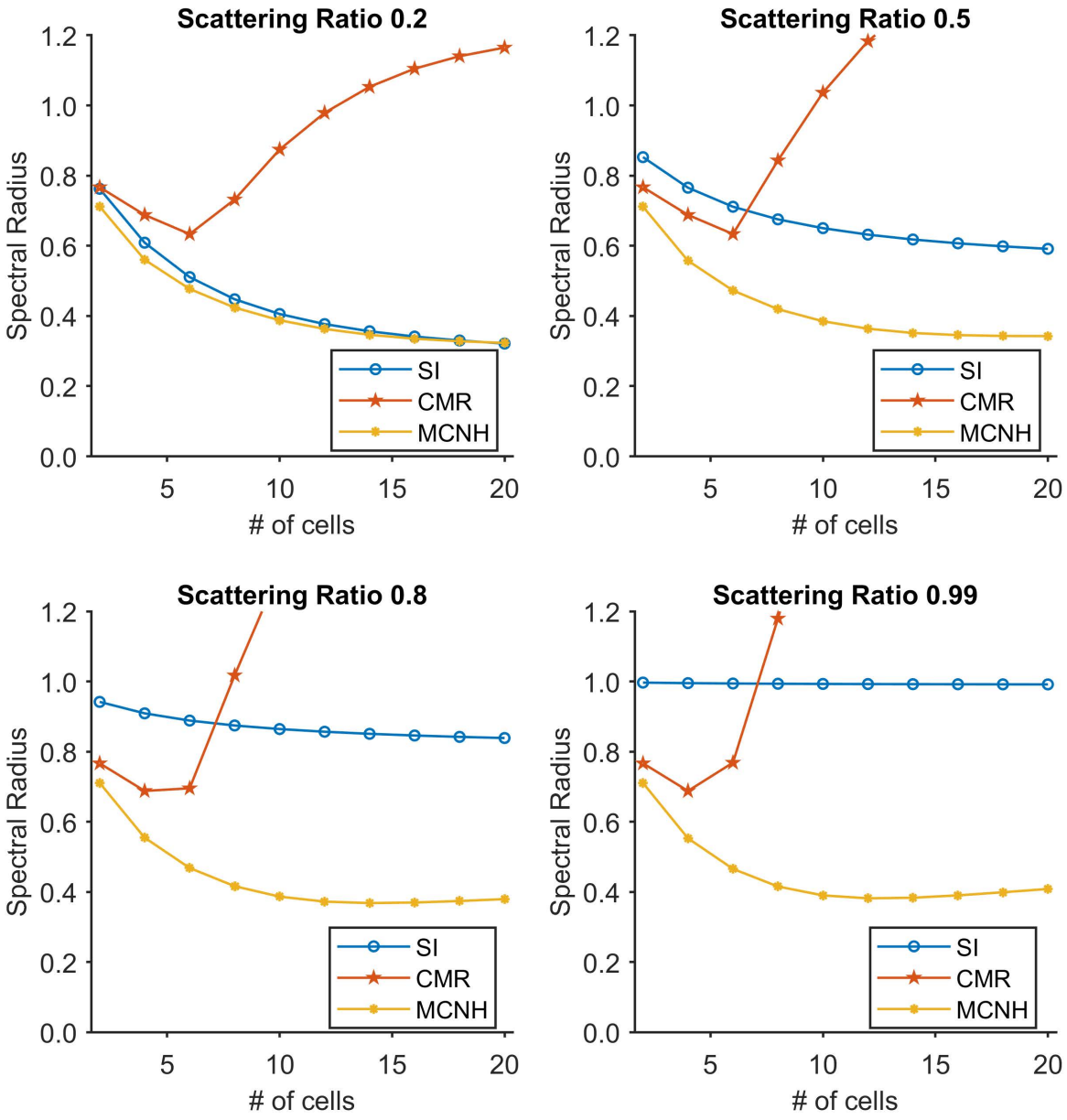

Figure 2.22 – spatial discretization test: A comparison of the spectral radius analysis of SI, MOC accelerated by CMR and MOC accelerated by MCNH for an increasing number of cells. The optical thickness of each cell has been fixed to a value of 0.1 mfp.

Chapter 3

New acceleration methods

3.1 Spatially variant rebalance method

As was explained in Section 2.20, the step constant spatial representation of flux used by CMFD results in instability for optically thick mediums. Moreover, CMFD is limited to such low order spatial representations of the flux due to a lack of degrees of freedom. Thus, the primary goal for developing the Spatially Variant Rebalance Method (SVRM), was to increase the degrees of freedom so that a higher order representation of the spatial distribution of the flux could be used. In order to catch these high order spatial variations, SVRM solves for the first order spatial projection of the balance equation along with the more commonly used zero order projection.

3.1.1 Derivation of fundamental equations

For simplicity, we will first derive the SVRM equations in 1D and then expand them to 3D. Fig. 3.1 illustrates the indexing scheme that will be used for 1D geometries. Subscripts L and R denote the left and the right side of the cell.

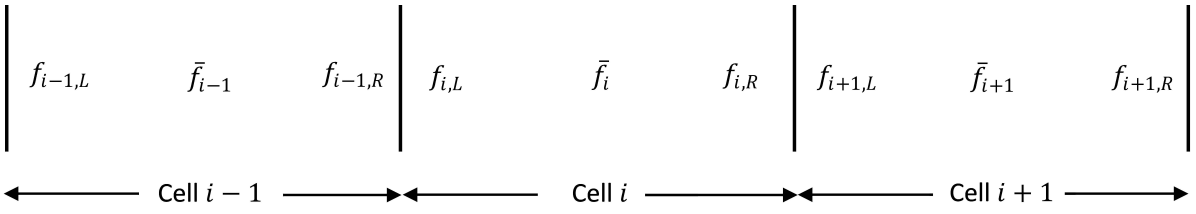

Figure 3.1 – An illustration of the indexing scheme for a 1D geometry, for a variable f.

.

The first equation of SVRM is simply the neutron balance equation with isotropic scattering,

$$\frac{dJ}{dx} + \Sigma_r(x)\phi(x) = Q_{Ext}(x), \tag{3.1}$$

where $\Sigma_r(x) = \Sigma_t(x) - \Sigma_{s0}(x)$ is the removal cross section. Integrating Eq. (3.1) over a discrete portion of space $x \in [-\Delta/2, \Delta/2]$ leads to,

$$J(\Delta/2) - J(-\Delta/2) + \int_{-\Delta/2}^{\Delta/2} \Sigma_r(x)\phi(x)\ dx = \int_{-\Delta/2}^{\Delta/2} Q_{Ext}(x)\ dx. \tag{3.2}$$

To obtain the second equation that characterizes the acceleration method, we introduce a first order spatial projection of the balance equation. We apply $\int_{-\Delta/2}^{\Delta/2} x\ dx$ to Eq. (3.1) to obtain,

$$\int_{-\Delta/2}^{\Delta/2} \frac{dJ(x)}{dx}\ xdx + \int_{-\Delta/2}^{\Delta/2} \Sigma_r(x)\phi(x)\ xdx = \int_{-\Delta/2}^{\Delta/2} Q_{Ext}(x)\ xdx. \tag{3.3}$$

Using the integration by parts technique, the integral of the first term of Eq. (3.3) becomes,

$$\int_{-\Delta/2}^{\Delta/2} \frac{dJ(x)}{dx}\ xdx = \int_{-\Delta/2}^{\Delta/2} \frac{d}{dx}[J(x)x]\ dx - \int_{-\Delta/2}^{\Delta/2} J(x)\ dx. \tag{3.4}$$

By simplifying and substituting Eq. (3.4) into Eq. (3.3), we get

$$\frac{J(\Delta/2)+J(-\Delta/2)}{2}\Delta - \int_{-\Delta/2}^{\Delta/2} J(x)\ dx + \int_{-\Delta/2}^{\Delta/2} \Sigma_r(x)\phi(x)\ xdx = \int_{-\Delta/2}^{\Delta/2} Q_{Ext}(x)\ xdx. \tag{3.5}$$

Eq. (3.5) represents the "balance" for the first-order spatial variation. By solving for Eq. (3.5), we are attempting to correct the higher order spatial moments of the flux.

3.1.2 Need for spatial representation

To make the system of equations numerically solvable, Eqs. (3.2) and (3.5) must be discretized. However, by closely examining the third and fourth term of Eq. (3.5) a complication can be discerned. In order to solve the first order expansion, knowledge of the spatial distribution of the scalar flux and fixed source within a region is required. Moreover, Eq. (3.5) becomes non-trivial when the flux and the source have a spatial variation within the discrete region. It is important to note that even when step approximation for the source is applied, MOC still considers the contributions of these high-order spatial variations of the flux. For instance, the n_{th} order moment of the scalar flux in a homogeneous slab can be explicitly computed using step MOC to be

$$\begin{aligned}
\int_{-\Delta/2}^{\Delta/2} dx\ x^n\phi(x) = \int_{-\Delta/2}^{\Delta/2} dx\ x^n \int_0^1 d\mu\ \Bigg\{ & \\
& \psi_L(\mu)e^{-\Sigma_t(x+\Delta/2)/|\mu|} + \frac{\Sigma_{sc,0}\bar{\phi}+Q_{Ext}}{\Sigma_t}\left[1-e^{-\Sigma_t(x+\Delta/2)/|\mu|}\right] \\
& + \psi_R(-\mu)e^{-\Sigma_t(\Delta/2-x)/|\mu|} + \frac{\Sigma_{sc,0}\bar{\phi}+Q_{Ext}}{\Sigma_t}\left[1-e^{-\Sigma_t(\Delta/2-x)/|\mu|}\right]\Bigg\}, \\
\bar{\phi} = \frac{\int_{-\Delta/2}^{\Delta/2} dx\ \phi(x)}{\Delta}, & \\
Q_{Ext} = \frac{\int_{-\Delta/2}^{\Delta/2} dx\ Q_{Ext}(x)}{\Delta}, &
\end{aligned}$$

where $\psi_L(\mu)$ and $\psi_R(-\mu)$ are the left and right boundary angular fluxes, respectively. It is important to note that for step MOC, only the uncollided flux is allowed to vary within the cell. By definition, the collided flux is represented using a flat distribution. Unfortunately, the first order spatial moment, required to satisfy Eq. (3.5), is not provided by the standard step MOC. Because of this lack of knowledge of the spatial moments, Eq. (3.5) must be artificially satisfied.

3.1.3 Flux, source and current spatial representation

SVRM attempts to artificially satisfy Eq. (3.5) by taking profit of the valuable information contained in the interface quantities, in order to form a parabolic representation of the scalar flux and the fixed source. In particular, MOC provides the average scalar flux and fixed source $(\bar{\phi}, Q_{Ext})$, and the left and right interface scalar flux and fixed source $(\phi_L, \phi_R, Q_{Ext,L}, Q_{Ext,R})$.

We begin by letting the function $f(x)$ $x \in [-\Delta/2, \Delta/2]$ represent the scalar flux or the fixed source. The generic function $f(x)$ is then approximated by the parabolic function $\widetilde{f}(x)$, as depicted in Fig. 3.2,

$$\widetilde{f}(x) = \bar{f} + [f_R - f_L]\frac{x}{\Delta} + 4\left[\frac{f_R + f_L}{2} - \bar{f}\right]\left[3\frac{x^2}{\Delta^2} - 1\right], \tag{3.6}$$

where $(\bar{f}, f_L, f_R)$, are the average of $f(x)$, and the point-wise values on the left and right boundaries, respectively. The expansion (3.6) is constructed such that the following three conditions are met,

$$\begin{aligned} \widetilde{f}(-\Delta/2) &= f_L, \\ \int_{-\Delta/2}^{\Delta/2} \widetilde{f}(x)dx &= \Delta\bar{f}, \\ \widetilde{f}(\Delta/2) &= f_R. \end{aligned}$$

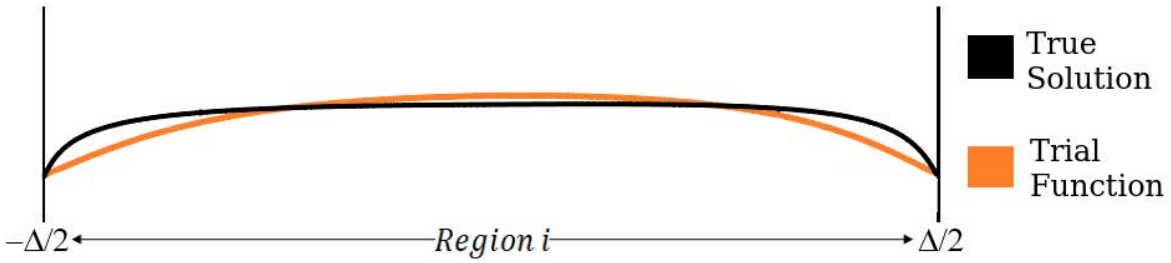

Figure 3.2 – The spatial distribution of the scalar flux and fixed source within each spatial region, assumed by SVRM

Rewriting Eqs. (3.2) and (3.5) using the parabolic trial function for the scalar flux integral (third term of Eq. (3.5)) and the fixed source integral (fourth term of Eq. (3.5)), we obtain

$$J_R - J_L + \tau_r\bar{\phi} = Q_{Ext}\Delta \tag{3.7}$$

for Eq. (3.2) and

$$\Delta\left[\frac{J_R + J_L}{2} - \bar{J}\right] + \widetilde{\delta}\tau_r\left[\phi_R - \phi_L\right] = \Delta\widetilde{\delta}\left[Q_{Ext,R} - Q_{Ext,L}\right] \tag{3.8}$$

for Eq. (3.5). Here, the symbols are

— $\phi_{L/R}$, indicating the interface scalar flux on the left/right side,

— $J_{L/R}$ and $\bar{J}$, respectively indicating the interface current on the left/right side and the average current,
— $Q_{Ext,L/R}$, indicating the interface fixed source on the left/right side,
— $\tau_r = [\Sigma_t - \Sigma_{s0}]\,\Delta$, indicating the removal optical thickness,
— $\widetilde{\delta}$, the modified spatial parameter that accounts for the first order variation, in particular

$$\widetilde{\delta} = \frac{\Delta}{12}.$$

By introducing a partition of the slab with the region index $i = 1, \ldots, I$, where I is the number of regions, and specializing Eqs. (3.7) and (3.8) in each spatial mesh, the acceleration method has 6 unknowns per mesh cell i,

$$[\phi_{i,L}, \phi_{i,R}, \bar{\phi}_i, J_{i,L}, J_{i,R}, \bar{J}_i],$$

and three given values for the source, i.e. $[Q_{Ext,i,L}, Q_{Ext,i,R}, Q_{Ext,i}]$. It is important to note that the transport equation constrains the scalar flux and the current to be continuous at each surface. Subsequently, we have also constrained the SVRM to respect continuity at the interface such that,

$$\begin{aligned} \phi_{i,L} &= \phi_{i-1,R} \text{ and } \phi_{i,R} = \phi_{i+1,L}, \\ J_{i,L} &= J_{i-1,R} \text{ and } J_{i,R} = J_{i+1,L}. \end{aligned}$$

The source is not constrained to be continuous at the region interfaces. Moreover, if the transport equation contains higher order spatial variations of the source, these variations can be approximated by the source of SVRM. However, when a step source is employed by the transport solver, $Q_{Ext,i,L} = Q_{Ext,i,R} = Q_{Ext,i}$.

Next, we substitute the relation

$$\begin{aligned} J_{i,R} &= J^+_{i,R} - J^+_{i+1,L}, \text{ and} \\ J_{i,L} &= J^+_{i-1,R} - J^+_{i,L}, \end{aligned}$$

into Eqs. (3.7) and (3.8)to get,

$$-J^+_{i+1,L} + \left[J^+_{i,R} + J^+_{i,L}\right] - J^+_{i-1,R} + \tau_{i,r}\bar{\phi}_i = Q_{Ext,i}\Delta_i, \tag{3.9a}$$

$$\Delta_i \left[\frac{J^+_{i,R} - J^+_{i,L} + J^+_{i-1,R} - J^+_{i+1,L}}{2} - \bar{J}_i\right] + \widetilde{\delta}_i \tau_{i,r}\left[\phi_{i,R} - \phi_{i,L}\right] = \Delta_i \widetilde{\delta}_i (Q_{Ext,i,R} - Q_{Ext,i,L}). \tag{3.9b}$$

Thus, the acceleration equations, Eqs. (3.9a) (3.9b), are now in terms of $J^+_{i,L}$, $J^+_{i,R}$, $\phi_{i,L}$, $\phi_{i,R}$, $\bar{\phi}_i$ and $\bar{J}_i$.

3.1.4 Closure equations

After discretizing the acceleration equations (3.9a) and (3.9b), the method has more unknowns than constraints. Thus, we introduce closure relations to constrain our additional unknowns.

3.1.4.1 Interface scalar flux closure relation

We assume the surface flux is proportional to the sum of the partial surface currents, as is commonly the case for CMR type accelerations, thus

$$\phi_{i,R} = \tilde{\beta}_{i,R}\left[J^+_{i,R} + J^+_{i+1,L}\right], \tag{3.10a}$$

$$\phi_{i,L} = \tilde{\beta}_{i,L}\left[J^+_{i,L} + J^+_{i-1,R}\right]. \tag{3.10b}$$

Here, the coefficients $\tilde{\beta}_{i,L/R}$ are transport computed parameters. The underlying assumption of Eqs. (3.10a) and (3.10b) is that the distribution of the angular flux within each angular hemisphere is fixed at the interface. This assertion can be seen to be true from the definition of the interface flux, Eq. (2.69b),

$$\begin{aligned}\phi_{i,s} &= \sum_{m\in M} w_m \psi_{m,i,s}, \\ &= \frac{\sum_{m\in M} w_m \psi_{m,i,s}}{\sum_{m\in M} w_m |\mu_m| \psi_{m,i,s}} \sum_{m\in M} w_m |\mu_m| \psi_{m,i,s},\end{aligned}$$

by enforcing the fixed angular shape function,

$$\begin{aligned}&\frac{\psi_{m,i,s}}{\sum_{m\in M}\psi_{m,i,s}} = \frac{\psi^{(n+1/2)}_{m,i,s}}{\sum_{m\in M}\psi^{(n+1/2)}_{m,i,s}} \\ &\phi_{i,s} = \tilde{\beta}_{i,s}\left[J^+_{i,s} + J^-_{i,s}\right], \\ &\mu_m = \hat{\Omega}_m \cdot \hat{x}, \\ &\tilde{\beta}_{i,s} = \frac{\sum_{m\in M} w_m \psi^{(n+1/2)}_{m,i,s}}{\sum_{m\in M} w_m |\mu_m| \psi^{(n+1/2)}_{m,i,s}},\end{aligned}$$

where s can be L or R. This assumption is accurate when the angular shape function of the angular flux is close to convergence. Thus, it is left up to the transport operator to converge the angular shape function of the angular flux.

3.1.4.2 The average scalar flux closure relation

Next, we assume that $\bar{\phi}_i$ is proportional to the sum of the outgoing partial currents,

$$\bar{\phi}_i = \tilde{\beta}_i\left[J^+_{i,R} + J^+_{i,L}\right]. \tag{3.11}$$

Closure relation (3.11) uses the same underlying assumption made by Eqs. (3.10a) and (3.10b) except that we are now applying it over the entire volume of the region. Note that all nonlinear coefficients, $\tilde{\beta}_{i,R}$, $\tilde{\beta}_{i,L}$, and $\tilde{\beta}_i$, are denoted by a tilde and are set such that the equations that introduce them are satisfied.

3.1.4.3 The average current closure relation

Finally, we link the average current of the region to the boundary spatial currents as follows

$$\bar{J}_i = \tilde{\alpha}_{i-1,R} J^+_{i-1,R} - \tilde{\alpha}_{i+1,L} J^+_{i+1,L}. \tag{3.12}$$

The $\tilde{\alpha}_{i-1,R}$ and $\tilde{\alpha}_{i+1,L}$ coefficients in Eq. (3.12) are derived by an explicit use of the MOC transmission equation. Since the source is constant within the cell, the average current depends only on the contribution of the incoming angular flux, thus

$$\bar{J}_i = \int_0^1 d\mu\ \mu|\mu|\ [\psi_L(\mu) - \psi_R(-\mu)]\frac{1 - e^{-\tau_i/|\mu|}}{\tau_i}, \tag{3.13}$$

Next, we modify relation (3.13) to be,

$$\bar{J}_i = \int_0^1 d\mu\ \mu|\mu| \left[\psi_L(\mu)\frac{J^+_{i+1,L}}{J^+_{i+1,L}} - \psi_R(-\mu)\frac{J^+_{i-1,R}}{J^+_{i-1,R}}\right] \frac{1 - e^{-\tau_i/|\mu|}}{\tau_i}. \tag{3.14}$$

By comparing Eq. (3.12) and Eq. (3.14) we obtain the definition for the coefficients $\tilde{\alpha}_{i-1,R}$ and $\tilde{\alpha}_{i+1,L}$ to be,

$$\tilde{\alpha}_{i-1,R} = \frac{\int_0^1 d\mu\ \mu|\mu| \left[1 - e^{-\tau_i/|\mu|}\right] \psi_R(-\mu)}{\tau_i J^+_{i-1,R}}, \text{ and}$$
$$\tilde{\alpha}_{i+1,L} = \frac{\int_0^1 d\mu\ \mu|\mu| \left[1 - e^{-\tau_i/|\mu|}\right] \psi_L(\mu)}{\tau_i J^+_{i+1,L}},$$

where the integrals are performed with the S_N quadrature formula.

3.1.5 Consistency with the transport equation and reduction of the spatial order of the acceleration

If the transport numerical scheme is conservative, the balance equation, Eq. (3.9a), will be satisfied by the final transport solution. However, because the parabolic representation of the flux and of the current is an approximation, the first-order spatial variation of the balance equation, Eq. (3.9b), will not necessarily satisfied by the transport solution. Therefore, we need to use a nonlinear parameter $\tilde{\eta}$ to ensure that the Eq. (3.9b) is satisfied by the converged solution. Additionally, the current or surface flux terms of Eq. (3.9b), could go to zero when the flux and current are flat. Thus, we need to ensure that the relation we choose to represent the parameter $\tilde{\eta}$ will not result in an undefined value for these cases. We start by substituting the previously defined closure equations into Eq. (3.9b) to get,

$$\begin{aligned}
&\Delta_i \left\{ \frac{[J^+_{i,R} - J^+_{i,L}] + [J^+_{i-1,R} - J^+_{i+1,L}]}{2} - \tilde{\alpha}_{i-1,R} J^+_{i-1,R} + \tilde{\alpha}_{i+1,L} J^+_{i+1,L} \right\} \\
&+ \widetilde{\delta}_i \tau_{i,r} \left\{ \tilde{\beta}_{i,R}[J^+_{i,R} + J^+_{i+1,L}] - \tilde{\beta}_{i,L}[J^+_{i,L} + J^+_{i-1,R}] \right\} \\
&= \Delta_i \widetilde{\delta}_i [Q_{Ext,i,R} - Q_{Ext,i,L}].
\end{aligned} \tag{3.15}$$

It is important to note that for step MOC $[Q_{Ext,i,R} - Q_{Ext,i,L}] = 0$. Since the non-linear coefficients together with the partial currents are all positive quantities, we can rearrange Eq. (3.15) into positive-signed terms P_i and negative-signed terms N_i, obtaining

$$P_i = N_i, \tag{3.16}$$

where

$$P_i = \left[\frac{\Delta_i}{2} + \widetilde{\delta}_i \tau_{i,r} \tilde{\beta}_{i,R}\right] J^+_{i,R} + \frac{\Delta_i}{2} J^+_{i-1,R} + \tilde{\alpha}_{i+1,L} J^+_{i+1,L} + \widetilde{\delta}_i \tau_{i,r} \tilde{\beta}_{i,R} J^+_{i+1,L} + \Delta_i \widetilde{\delta}_i Q_{Ext,i,L}$$

and

$$N_i = \left[\frac{\Delta_i}{2} + \widetilde{\delta}_i \tau_{i,r} \tilde{\beta}_{i,L}\right] J^+_{i,L} + \frac{\Delta_i}{2} J^+_{i+1,L} + \tilde{\alpha}_{i-1,R} J^+_{i-1,R} + \widetilde{\delta}_i \tau_{i,r} \tilde{\beta}_{i,L} J^+_{i-1,R} + \Delta_i \widetilde{\delta}_i Q_{Ext,i,R}.$$

Next we ensure that Eq. (3.16) is satisfied by the transport equation by introducing the parameter $\tilde{\eta}$ such that,

$$\begin{aligned} \tilde{\eta}_i P_i &= [1 - \tilde{\eta}_i] N_i, \\ \tilde{\eta}_i &= \frac{N_i}{P_i + N_i}. \end{aligned}$$

Thus Eq. (3.15) is rewritten as

$$\begin{aligned} &\tilde{\eta}_i \left[\frac{\Delta_i}{2} + \widetilde{\delta}_i \tau_{i,r} \tilde{\beta}_{i,R}\right] J^+_{i,R} \\ &+ \left\{\tilde{\eta}_i \frac{\Delta_i}{2} - [1 - \tilde{\eta}_i] \widetilde{\delta}_i \tau_{i,r} \tilde{\beta}_{i,L} - [1 - \tilde{\eta}_i] \tilde{\alpha}_{i-1,R}\right\} J^+_{i-1,R} \\ &- [1 - \tilde{\eta}_i] \left[\frac{\Delta_i}{2} + \widetilde{\delta}_i \tau_{i,r} \tilde{\beta}_{i,L}\right] J^+_{i,L} \\ &+ \left\{\tilde{\eta}_i \tilde{\alpha}_{i+1,L} + \tilde{\eta}_i \widetilde{\delta}_i \tau_{i,r} \tilde{\beta}_{i,R} - [1 - \tilde{\eta}_i] \frac{\Delta_i}{2}\right\} J^+_{i+1,L} \\ &= \tilde{\eta}_i \Delta_i \widetilde{\delta}_i Q_{Ext,i,R} - [1 - \tilde{\eta}_i] \Delta_i \widetilde{\delta}_i Q_{Ext,i,L}. \end{aligned} \tag{3.17}$$

3.1.6 Boundary conditions

The last constraints are the boundary conditions. Since the transport operator does not invert the boundary solution with each iteration, there is an erroneous current on the boundary surfaces until the final solution is reached. Since the final transport solution must satisfy the boundary conditions, the SVRM method makes use of the "true" boundary conditions instead of using a non-linear albedo coefficient as is commonly done in CMR and CMFD. This is a way of achieving additional acceleration on the boundary. The boundary conditions of SVRM are as follows:
the **reflective** boundary condition

$$J^-_{1/I,L/R} = J^+_{1/I,L/R},$$

the **translation** boundary condition

$$J^-_{1/I,L/R} = J^+_{I/1,R/L},$$

and the **vacuum** boundary condition

$$J^-_{1/I,L/R} = 0.$$

The boundary conditions of SVRM can be expressed in a general form as,

$$J^-_{1,L} = v_{L\leftarrow L} J^+_{1,L} + v_{L\leftarrow R} J^+_{I,R} \quad \text{for } i = 1,$$

$$J^-_{I,R} = v_{R\leftarrow L} J^+_{1,L} + v_{R\leftarrow R} J^+_{I,R} \quad \text{for } i = I.$$

Here, the currents $J^-_{1,L}$ and $J^-_{I,R}$ indicate the incoming boundary current on the left and on the right sides respectively. The albedo coefficients $v_{L\leftarrow L}$, $v_{L\leftarrow R}$, $v_{R\leftarrow L}$, and $v_{R\leftarrow R}$ are defined by Table (3.1).

	Reflective	Periodic	Vacuum
$v_{L\leftarrow L} =$	1	0	0
$v_{L\leftarrow R} =$	0	1	0
$v_{R\leftarrow L} =$	0	1	0
$v_{R\leftarrow R} =$	1	0	0

Table 3.1 – Values of the albedo coefficients at the left and right boundaries.

3.1.7 Solving SVRM

The SVRM equations (3.9a) and (3.9b) can be written in matrix form as

$$(\underline{\underline{\tilde{\boldsymbol{A}}}} + \underline{\underline{\tilde{\boldsymbol{B}}}})\ \boldsymbol{J} = \boldsymbol{Q_{Ac}}, \tag{3.18}$$

Where $\boldsymbol{J}$ is the acceleration solution, which is comprised of the outgoing partial current. $\underline{\underline{\tilde{\boldsymbol{A}}}}$ accounts for the terms of Eqs. (3.9a) and (3.9b) that represent the contributions of the internal regions and $\underline{\underline{\tilde{\boldsymbol{B}}}}$ accounts for the contributions of the boundary conditions. Finally, $\boldsymbol{Q}_{Ac}$ is the acceleration source, which is made up of the constant terms of Eqs. (3.9a) and (3.9b) and the external boundary source.

$\underline{\underline{\tilde{\boldsymbol{A}}}}$, can be explicitly written as a block matrix of the form,

$$\underline{\underline{\tilde{\boldsymbol{A}}}} = \begin{bmatrix} \underline{\underline{\tilde{\boldsymbol{A}}}}_{1,1} & \underline{\underline{\tilde{\boldsymbol{A}}}}_{1,2} & 0 & \dots & 0 \\ \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i-1} & \underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i} & \underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i+1} & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ 0 & \dots & 0 & \underline{\underline{\tilde{\boldsymbol{A}}}}_{I,I-1} & \underline{\underline{\tilde{\boldsymbol{A}}}}_{I,I} \end{bmatrix}.$$

More specifically for node i, the equations take a 2x2 matrix form of the type

$$\underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i-1}\ \boldsymbol{J}_{i-1} + \underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i}\ \boldsymbol{J}_i + \underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i+1}\ \boldsymbol{J}_{i+1} = \boldsymbol{Q}_{Ac,i} \quad \text{when } i \neq 1 \text{ and } i \neq I, \tag{3.19}$$

where

$$\boldsymbol{J}_i = \begin{bmatrix} J^+_{i,L} \\ J^+_{i,R} \end{bmatrix}.$$

By substituting the closure relations, Eqs. (3.10a)-(3.12), and Eq. (3.17) into Eq.(3.19), we get an expression for the matrices $\underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i-1}$, $\underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i}$, $\underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i+1}$ and $\boldsymbol{Q}_{Ac,i}$ for $i = 1, \dots, I$,

$$\underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i-1} = \begin{bmatrix} 0 & -1 \\ 0 & \tilde{\eta}_i \frac{\Delta_i}{2} - (1-\tilde{\eta}_i)(\tilde{\delta}_i \tau_{i,r} \tilde{\beta}_{i,L} + \tilde{\alpha}_{i-1,R}) \end{bmatrix},$$

$$\underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i} = \begin{bmatrix} 1+\tilde{\beta}_i\tau_{i,r} & 1+\tilde{\beta}_i\tau_{i,r} \\ [-(1-\tilde{\eta}_i)(\frac{\Delta_i}{2}+\widetilde{\delta}_i\tau_{i,r}\tilde{\beta}_{i,L})] & [\tilde{\eta}_i(\frac{\Delta_i}{2}+\widetilde{\delta}_i\tau_{i,r}\tilde{\beta}_{i,R})] \end{bmatrix},$$

$$\underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i+1} = \begin{bmatrix} -1 & 0 \\ \tilde{\eta}_i(\widetilde{\delta}_i\tau_{i,r}\tilde{\beta}_{i,R}+\tilde{\alpha}_{i+1,L})-(1-\tilde{\eta}_i)\frac{\Delta_i}{2} & 0 \end{bmatrix},$$

and finally,

$$\boldsymbol{Q}_{Ac,i} = \begin{bmatrix} \Delta_i Q_{Ext,i} \\ \Delta_i\widetilde{\delta}_i(\tilde{\eta}Q_{Ext,i,R}-(1-\tilde{\eta})Q_{Ext,i,L}) \end{bmatrix}.$$

$\underline{\underline{\tilde{\boldsymbol{B}}}}$, can be explicitly written as a block matrix of the form,

$$\underline{\underline{\tilde{\boldsymbol{B}}}} = \begin{bmatrix} \underline{\underline{\tilde{\boldsymbol{A}}}}_{1,0}\underline{\underline{\tilde{\boldsymbol{B}}}}_{L,1} & 0 & \dots & 0 & \underline{\underline{\tilde{\boldsymbol{A}}}}_{1,0}\underline{\underline{\tilde{\boldsymbol{B}}}}_{L,I} \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 \\ \underline{\underline{\tilde{\boldsymbol{A}}}}_{I,I+1}\underline{\underline{\tilde{\boldsymbol{B}}}}_{R,1} & 0 & \dots & 0 & \underline{\underline{\tilde{\boldsymbol{A}}}}_{I,I+1}\underline{\underline{\tilde{\boldsymbol{B}}}}_{R,I} \end{bmatrix},$$

where

$$\underline{\underline{\tilde{\boldsymbol{B}}}}_{L,1} = \begin{bmatrix} v_{L\leftarrow L} & 0 \\ 0 & 0 \end{bmatrix},$$

$$\underline{\underline{\tilde{\boldsymbol{B}}}}_{L,I} = \begin{bmatrix} 0 & v_{L\leftarrow R} \\ 0 & 0 \end{bmatrix},$$

$$\underline{\underline{\tilde{\boldsymbol{B}}}}_{R,1} = \begin{bmatrix} 0 & 0 \\ v_{R\leftarrow L} & 0 \end{bmatrix},$$

$$\underline{\underline{\tilde{\boldsymbol{B}}}}_{R,I} = \begin{bmatrix} 0 & 0 \\ 0 & v_{R\leftarrow R} \end{bmatrix},$$

Applying the accelerated transport iteration index scheme to Eq. (3.18) we get,

$$\left[\underline{\underline{\tilde{\boldsymbol{A}}}}^{(n+1/2)} + \underline{\underline{\tilde{\boldsymbol{B}}}}^{(n+1/2)}\right]\boldsymbol{J}^{(n+1)} = \boldsymbol{Q}_{Ac}^{(n+1/2)}. \tag{3.20}$$

3.1.8 Prolongation operator

In order to accurately update the transport solution using the SVRM partial current, we must preserve the assumptions of the closure equations. Since the angular flux distribution within each partial current was fixed by Eqs. (3.10a) and (3.10b), the updated angular flux must be proportional to the partial current. The prolongation operator uses the values $(J^+_{i,L}, J^+_{i,R})$ to correct both the interface angular flux and the scalar flux. As is common for a rebalancing method, SVRM reconstructs the scalar flux using Eq. (3.11) with the SVRM-computed partial currents,

$$\bar{\phi}_i^{(n+1)} = \tilde{\beta}_i[J^+_{i,R} + J^+_{i,L}]^{(n+1)}. \tag{3.21}$$

The second update is for the interface angular flux,

$$\psi^{+,(n+1)}_{i,L/R,m} = \frac{J^{+,(n+1)}_{i,L/R}}{J^{+,(n+1/2)}_{i,L/R}}\psi^{+,(n+1/2)}_{1,L/R,m}, \quad m \in S^+_{i,L/R}. \tag{3.22}$$

It is important to note that the interior interface angular flux is only updated using Eq. (3.21) when the transport sweep is done using a GJ iterative scheme. Eqs. (3.22) and (3.22) can be cast in a matrix formalism such that,

$$\boldsymbol{\Psi}^{(n+1)} = \underline{\underline{\tilde{\boldsymbol{P}}}}\ \boldsymbol{J}^{(n+1)},$$

where $\underline{\underline{\tilde{\boldsymbol{P}}}}$ is a rectangular diagonal-by-block matrix composed of matrix $\underline{\underline{\tilde{\boldsymbol{P}}}}_i$ containing the coefficients of Eqs. (3.21), (3.22) for each region i,

$$\underline{\underline{\tilde{\boldsymbol{P}}}}_i = \begin{bmatrix} \frac{\boldsymbol{\Psi}_{i,L}^{+,(n+1/2)}}{J_{i,L}^{+,(n+1/2)}} & 0 \\ 0 & \frac{\boldsymbol{\Psi}_{i,R}^{+,(n+1/2)}}{J_{i,R}^{+,(n+1/2)}} \\ \tilde{\beta}_i & \tilde{\beta}_i \end{bmatrix},$$

$$\boldsymbol{\Psi}_{i,L/R}^{+,(n+1/2)} = \begin{bmatrix} \vdots \\ \psi_{i,L/R,m}^{+,(n+1/2)} \\ \vdots \end{bmatrix}, m \in M_{L/R}^{+}.$$

3.1.9 Expansion for a multidimensional cartesian geometry

There are 6 partial currents that need to be solved for a 3D Cartesian cell. However, SVRM would only have the zeroth and 3 first-order spatial projections ($\int_{-\Delta_x/2}^{\Delta_x/2} x dx$, $\int_{-\Delta_y/2}^{\Delta_y/2} y dy$ and $\int_{-\Delta_z/2}^{\Delta_z/2} z dz$) of the balance equation to close them. Thus, for multidimensional cases, SVRM requires additional closure equations. To do this we consider that the partial current is a combination of the contribution of scalar flux and a correction factor. Thus, for a 3D Cartesian cell, the following closure relations are added to the system of equations,

$$\begin{aligned}
J_{i,R}^{+} &= \tilde{\beta}_{i,x}\bar{\phi}_i + \tilde{c}_x, \\
J_{i,L}^{+} &= \tilde{\beta}_{i,x}\bar{\phi}_i - \tilde{c}_x, \\
J_{i,T}^{+} &= \tilde{\beta}_{i,y}\bar{\phi}_i + \tilde{c}_y, \\
J_{i,B}^{+} &= \tilde{\beta}_{i,y}\bar{\phi}_i - \tilde{c}_y, \\
J_{i,U}^{+} &= \tilde{\beta}_{i,z}\bar{\phi}_i + \tilde{c}_z, \\
J_{i,D}^{+} &= \tilde{\beta}_{i,z}\bar{\phi}_i - \tilde{c}_z, \\
\tilde{\beta}_{i,x} &= \frac{J_{i,R}^{+} + J_{i,L}^{+}}{2\bar{\phi}_i}, \\
\tilde{\beta}_{i,y} &= \frac{J_{i,T}^{+} + J_{i,B}^{+}}{2\bar{\phi}_i}, \\
\tilde{\beta}_{i,z} &= \frac{J_{i,U}^{+} + J_{i,D}^{+}}{2\bar{\phi}_i}, \\
\bar{J}_{x/y/z,i} &= \sum_{s' \in S_i} \tilde{I}_{x/y/z,i,s'} J_{i,s'}^{-}, \\
\tilde{I}_{x/y/z,i,s'} &= \frac{\sum_{m \in M_{s'}^{-}} \vec{n}_{x/y/z} \cdot \Omega_m w_m I_{m,i,s'} \psi_{m,i,s'}^{-}}{J_{i,s'}^{-}},
\end{aligned}$$

where T, B, U, D refer to the top, bottom, up and down surfaces of the cell. $\tilde{c}_{x/y/z}$ is the corrective unknown along the $x/y/z$ axis. S_i is the set of all the surfaces of cell i. It is important to note that in comparison to comparison to Eq. (3.11) $\tilde{\beta}_{i,x} = 1/2\tilde{\beta}_i$. By adding these closure relations to those defined in Section 3.1.4, the new set of unknowns solved by SVRM is composed of the scalar flux and the corrective terms. For example, by using the new unknowns for a 1D geometry, Eq. (3.19) becomes,

$$\underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i-1}\ \boldsymbol{\Phi}_{i-1} + \underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i}\ \boldsymbol{\Phi}_i + \underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i+1}\ \boldsymbol{\Phi}_{i+1} = \boldsymbol{Q}_{Ac,i} \quad \text{when } i \neq 1 \text{ and } i \neq I, \tag{3.23}$$

where

$$\boldsymbol{\Phi}_i = \begin{bmatrix} \bar{\phi}_i \\ \tilde{c}_x \end{bmatrix},$$

$$\underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i-1} = \begin{bmatrix} -\tilde{\beta}_{i-1,x} & -1 \\ \tilde{\beta}_{i-1,x}\left[\tilde{\eta}_i\frac{\Delta_i}{2} - (1-\tilde{\eta}_i)(\widetilde{\delta}_i\tau_{i,r}\tilde{\beta}_{i,L} + \tilde{\alpha}_{i-1,R})\right] & \tilde{\eta}_i\frac{\Delta_i}{2} - (1-\tilde{\eta}_i)(\widetilde{\delta}_i\tau_{i,r}\tilde{\beta}_{i,L} + \tilde{\alpha}_{i-1,R}) \end{bmatrix},$$

$$\underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i} = \begin{bmatrix} 2(\tilde{\beta}_{i,x} + \tau_{i,r}) & 0 \\ \tilde{\beta}_{i,x}\left\{-\frac{\Delta_i}{2} + \Delta_i\tilde{\eta}_i + \widetilde{\delta}_i\tau_{i,r}\left[-(1-\tilde{\eta}_i)\tilde{\beta}_{i,L} + \tilde{\eta}_i\tilde{\beta}_{i,R}\right]\right\} & \frac{\Delta_i}{2} + \widetilde{\delta}_i\tau_{i,r}\left[(1-\tilde{\eta}_i)\tilde{\beta}_{i,L} + \tilde{\eta}_i\tilde{\beta}_{i,R}\right] \end{bmatrix},$$

$$\underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i+1} = \begin{bmatrix} -\tilde{\beta}_{i+1,x} & 1 \\ \tilde{\beta}_{i+1,x}\left[\tilde{\eta}_i(\widetilde{\delta}_i\tau_{i,r}\tilde{\beta}_{i,R} + \tilde{\alpha}_{i+1,L}) - (1-\tilde{\eta}_i)\frac{\Delta_i}{2}\right] & -\tilde{\eta}_i(\widetilde{\delta}_i\tau_{i,r}\tilde{\beta}_{i,R} + \tilde{\alpha}_{i+1,L}) + (1-\tilde{\eta}_i)\frac{\Delta_i}{2} \end{bmatrix},$$

and finally,

$$\boldsymbol{Q}_{Ac,i} = \begin{bmatrix} \Delta_i Q_{Ext,i} \\ \Delta_i\widetilde{\delta}_i(\tilde{\eta}Q_{Ext,i,R} - (1-\tilde{\eta})Q_{Ext,i,L}) \end{bmatrix}.$$

3.1.10 Convergence analysis

For the sake of consistency, we have chosen to analyze SVRM using the three test cases introduced in Section 2.16. An explicit derivation of the Jacobian of SVRM is presented in Appendix B.3.3 as an example.

Following the protocol of the infinite slab test, the results of the Fourier analysis of MOC accelerated by SVRM, is shown in Fig. 3.3. Like CMR and the MCNH variant of CMFD, the dominant eigenmode of SVRM belongs to the highest spatial frequency, whereas the zeroth spatial frequency has an eigenvalue of zero. As shown by Fig. 3.4, the spectral radius of SVRM is less than that of MCNH for much of the domain when the scattering ratio is 0.99. As well, SVRM is stable for a larger portion of the studied domain than MCNH. This result is due to the fact that SVRM does not stimulate the high spatial frequencies to the degree that MCNH does. This likely is a consequence of the parabolic spatial representation used in SVRM being more accurate than that of MCNH. However, when the scattering ratio is 0.1, the spectral radius of MCNH is less or equal to that of SVRM.

Next, the spectral radius analysis specified by the finite slab test, applied to SI and MOC accelerated by MCNH and SVRM, is shown by Fig. 3.5. When the scattering ratio is 0.1 or the optical thickness is less than ~ 2 mfp, MCNH's spectral radius is equal to or less than that of SVRM for the standard and Gauss-Jacobi transport iteration schemes.

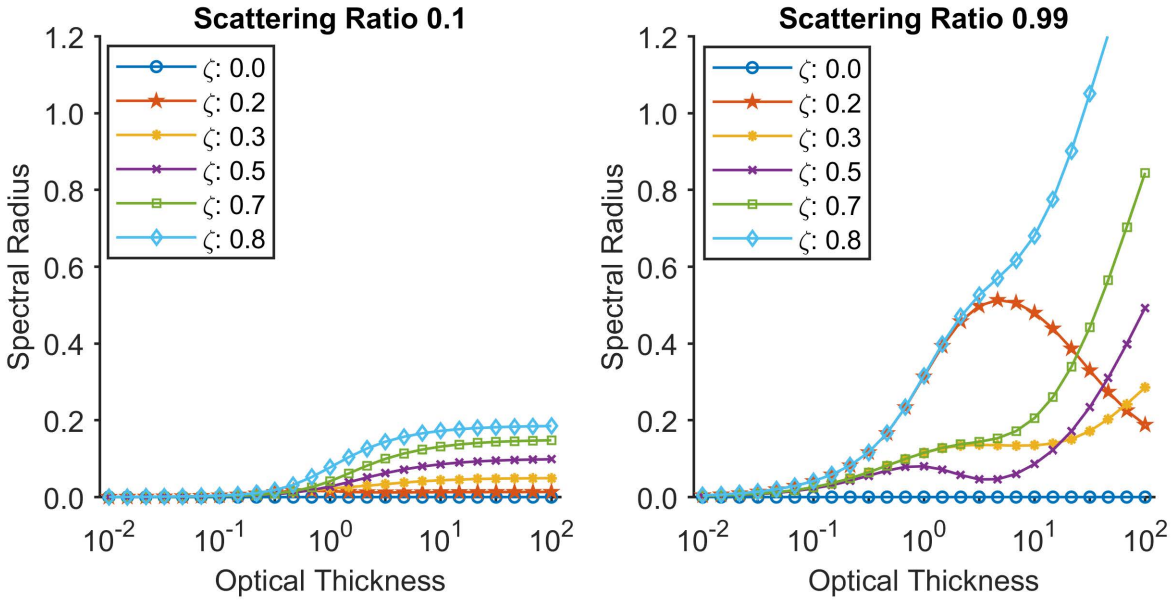

Figure 3.3 – Infinite slab test: a comparison of the spectral radius of several different spatial frequencies, in units of Hz, of the Fourier transformed unknowns of MOC accelerated by SVRM for an infinite slab.

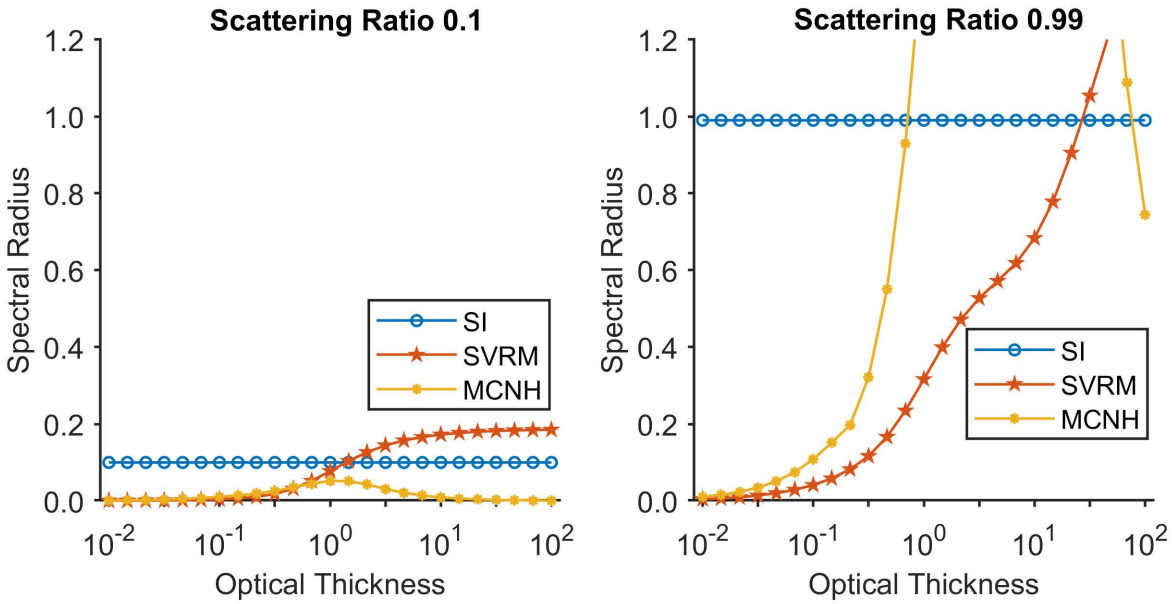

Figure 3.4 – Infinite slab test: a comparison of the spectral radius of SI, MOC accelerated by MCNH and MOC accelerated by SVRM for an infinite slab.

However, When the scattering ratio is 0.99 and the optical thickness is greater than 2 mfp, SVRM performs significantly better than MCNH. When the Gauss-Jacobi transport iteration scheme is used, the spectral radius of SVRM is the same or less than that of MCNH for all of the explored domain. This suggests that SVRM is able to more effectively accelerate spatially decomposed domains due to its higher order representation of the spatial variations of the flux.

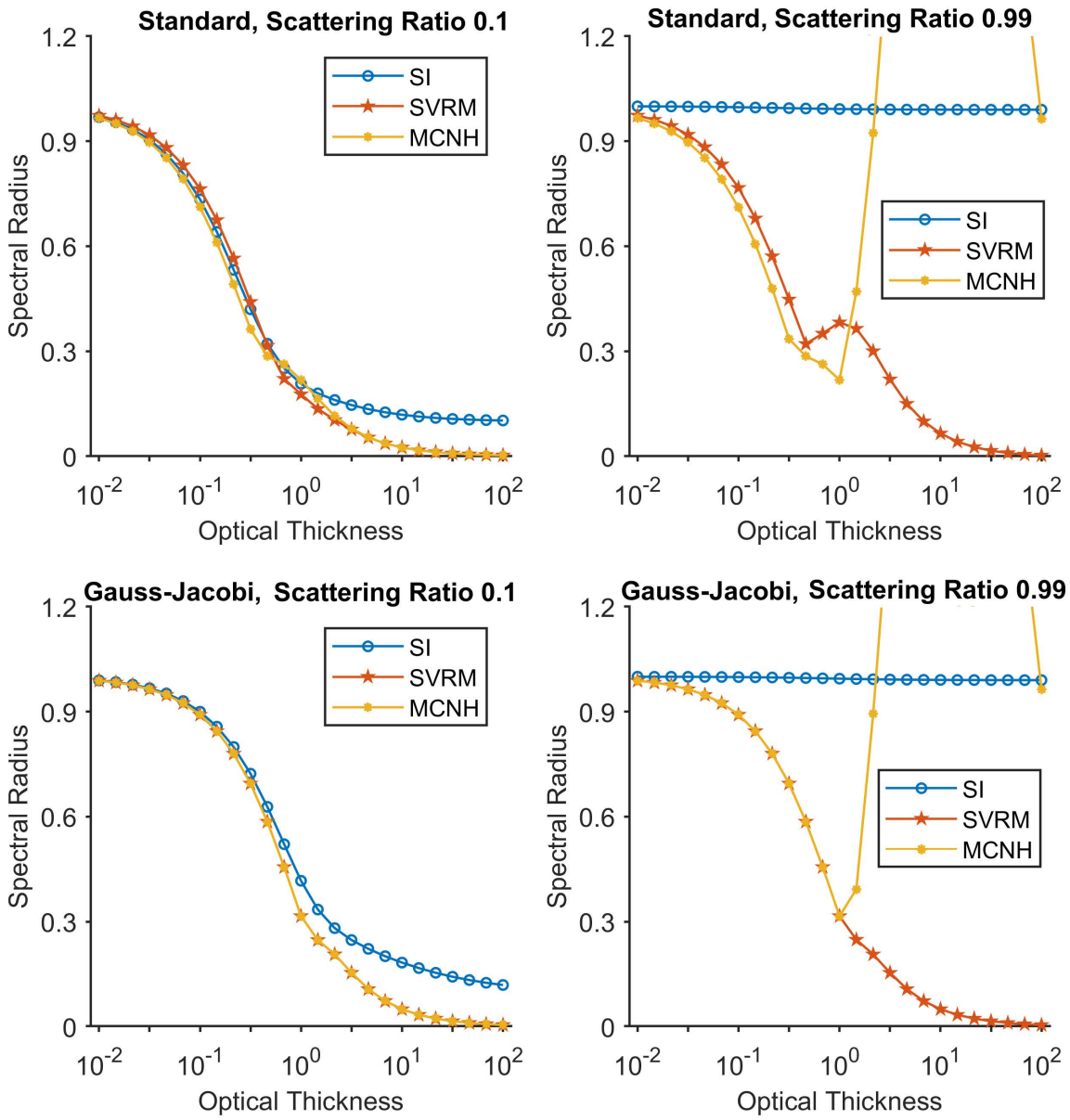

Figure 3.5 – Finite slab test: a comparison of the spectral radius analysis of SI, MOC accelerated by MCNH and MOC accelerated by SVRM for a finite slab.

Finally, the spectral radius of SI and MOC accelerated by MCNH and SVRM are compared using the spatial discretization test. The results of the comparison are shown by Figs. 3.6 and 3.7, the optical thickness of each cell has been fixed to 0.1 mfp and 10 mfp, respectively. By analyzing Fig. 3.6, the spectral radius of MCNH is less than that of SVRM everywhere. In fact, SVRM performs worse than SI when the scattering ratio is

0.2. For both SVRM and MCNH, the spectral radius decreases as the number of cells increases.

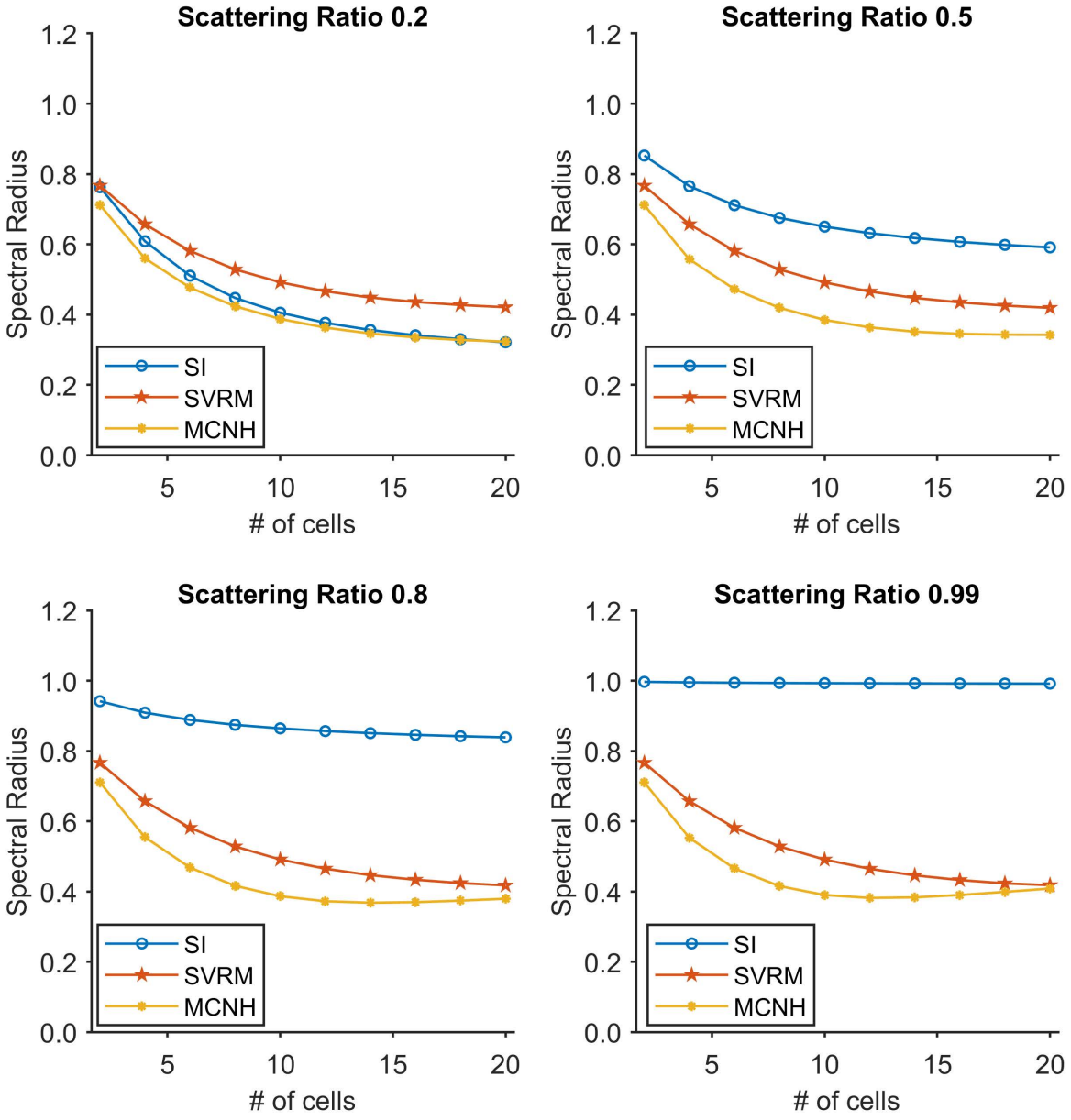

Figure 3.6 – Spatial discretization test: a comparison of the spectral radius analysis of SI, MOC accelerated by MCNH and MOC accelerated by SVRM for an increasing number of cells. The optical thickness of each cell has been fixed to a value of 0.1 mfp.

However, for all the cases shown by Fig. 3.7, the spectral radius of SVRM is less than that of MCNH. When the scattering ratio is set to a value of 0.99, both SVRM and MCNH are unstable when the cell count is greater than 2. For all other cases, the spectral radii of the two methods is relatively flat and less than that of transport.

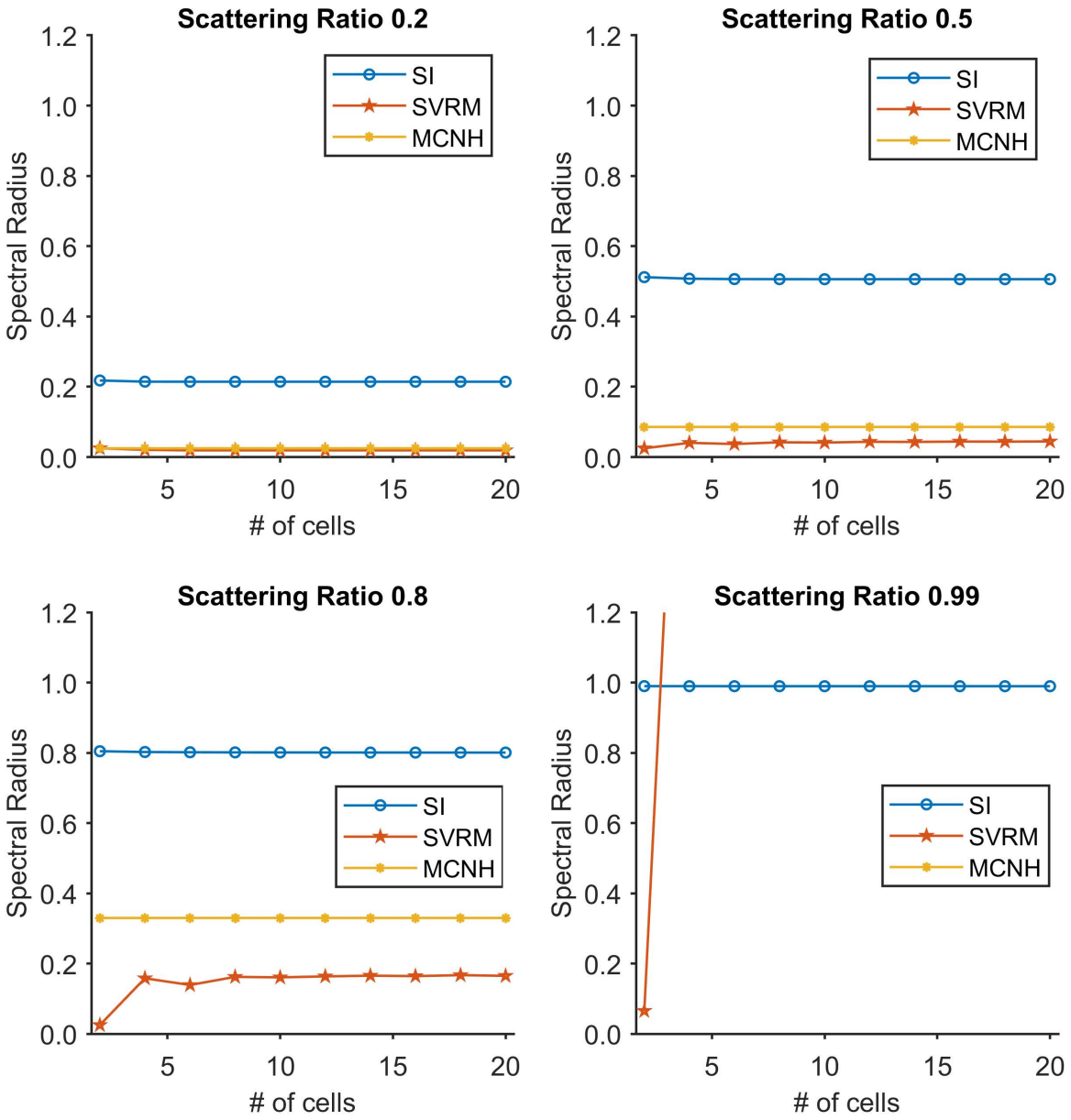

Figure 3.7 – Spatial discretization test: a comparison of the spectral radius analysis of SI, MOC accelerated by MCNH and MOC accelerated by SVRM for an increasing number of cells. The optical thickness of each cell has been fixed to a value of 10 mfp.

3.1.11 Advantages and disadvantages of SVRM

SVRM has improved convergence properties compared to MCNH for some of the cases studied in Section 3.1.10 and [18]. In particular, Fig. 3.4 shows that for an infinite slab when the scattering ratio is set to 0.99 MCNH is unstable when the optical thickness exceeds $\sim$ 0.8 mfp whereas SVRM is stable until $\sim$ 30 mfp. Also, SVRM accounts for high order variations of the flux that are not typically accounted for by other acceleration methods. Moreover, in the case that the medium is approaching a vacuum, SVRM requires no modifications in order to work, whereas diffusion coefficients of CMFD approach a singularity.

However, SVRM still suffers from instability when the scattering ratio is set to 0.99 and the optical thickness is increased beyond $\sim$ 30 mfp for an infinite slab. This instability was shown by Fig. 3.3 to be the result of the stimulation of the eigenmodes associated with the high spatial frequencies. This stimulation is the result of an inconsistency between the parabolic spatial representation used by SVRM and the spatial representation of the step MOSC operator. As was noted in Section 3.1.2, step MOSC uses a step representation for the scattering source and a exponential variation for the uncollided flux. Since SVRM does not know which transport operator it is accelerating, SVRM uses a parabolic representation for both the scattering source and the uncollided flux. As the optical thickness increases, difference between the transport representation of the scattering source and the uncollided flux grows, increasing the discrepancy of the parabolic representation used by SVRM. Moreover, when the scattering ratio is increased, the discrepancy between the step and parabolic representation is amplified, as can be seen in Fig. 3.3. For the 1D C5G7 benchmark, [23], SVRM was not able to converge to a result. As well, closure equation (3.11) of SVRM becomes ill conditioned when the outgoing partial currents go to zero. Finally, SVRM cannot be easily expanded for an arbitrary cell shape and requires additional closure equations for multidimensional problems.

For the aforementioned reasons, we chose to discontinue the development of SVRM and instead try a different approach. In order to correct the instability of SVRM, we decided to use nonlinear coefficients derived from the transport equation to close the system. This way, a consistent representation of the spatial distribution of the flux would be ensured. As well, we decided that the number of acceleration equations should expand with the number of interfaces to ensure that the method would be independent of cell shape and would work for unstructured meshes. Following this methodology, a new acceleration method called the Response Matrix Acceleration method (RMA) was conceived.

3.2 Response matrix acceleration method

The main goals of RMA were to create a nonlinear DP0 method that was stable for every case. Using the increased degrees of freedom offered by RMA, we attempt to account for the high-order moment contributions to the scalar flux. As well, we have chosen the unknowns of RMA to be composed of the interface partial current. For clarity, we will begin by deriving the equations of RMA for the step MOSC operator, and then expand the RMA equations for a higher-order transport operator. Focusing on a single energy

group with the step MOSC equations for a problem with a fixed external source we get

$$\psi^{+}_{m,i,s} = \sum_{s' \in S^{-}_{m,i,s}} T_{m,i,s,s'} \psi^{-}_{m,i,s'} + E_{m,i,s} Q_{m,i}, \tag{3.24a}$$

$$\bar{\phi}_{l,k,i} = \sum_{m=1}^{M} w_m Y_{l,k}(\hat{\Omega}_m) \left[\sum_{s' \in S^{-}_{m,i}} I_{m,i,s'} \psi^{-}_{m,i,s'} + C_{m,i} Q_{m,i} \right], \tag{3.24b}$$

$$Q_{m,i} = \sum_{l=0}^{L} \sum_{k=-l}^{l} Y_{l,k}(\hat{\Omega}_m) Q_{l,k,i}. \tag{3.24c}$$

In order to solve for the interface partial currents, the angular shape function of the surface flux is assumed to be fixed within each angular hemisphere, such that

$$\begin{aligned} \psi^{\pm}_{m,i,s} &= \tilde{h}^{\pm}_{m,i,s} J^{\pm}_{i,s}, \\ \tilde{h}^{\pm}_{m,i,s} &= \psi^{\pm}_{m,i,s} / J^{\pm}_{i,s} \end{aligned} \tag{3.25a}$$

where $\tilde{h}^{\pm}_{m,i,s}$ is the nonlinear coefficient set such that Eq. (3.25a) is satisfied. This is the fundamental assumption upon which RMA is built. Therefore, it has a significant influence on the performance of the method as will be seen in Section 3.2.13. Substituting Eq. (3.25a) into Eq. (3.24a) gives us,

$$\tilde{h}^{+}_{m,i,s} J^{+}_{i,s} = \sum_{s' \in S^{-}_{m,i,s}} T_{m,i,s,s'} \tilde{h}^{-}_{m,i,s'} J^{-}_{i,s'} + E_{m,i,s} Q_{m,i}. \tag{3.26}$$

By applying $\sum_{m \in M^{\pm}_{i,s}} w_m \left| \hat{\Omega}_m \cdot \hat{n}_{i,s} \right|$ to Eq. (3.26) and by expanding $Q_{m,i}$ in terms of $Y_{l,k}(\hat{\Omega}_m)$ we get,

$$\begin{aligned} J^{+}_{i,s} = &\left[\sum_{m \in M^{+}_{s}} w_m \left| \hat{\Omega}_m \cdot \hat{n}_{i,s} \right| \sum_{s' \in S^{-}_{m,i,s}} T_{m,i,s,s'} \tilde{h}^{-}_{m,i,s'} \right] J^{-}_{i,s'} \\ &+ \sum_{l=0}^{L} \sum_{k=-l}^{l} Q_{l,k,i} \sum_{m \in M^{+}_{s}} w_m \left| \hat{\Omega}_m \cdot \hat{n}_{i,s} \right| Y_{l,k}(\hat{\Omega}_m) E_{m,i,s}, \end{aligned}$$

where $M^{\pm}_{s}$ are the set of outgoing and incoming directions that intersect surface s. We simplify the notation by introducing the nonlinear coefficient $\tilde{T}_{i,s,s'}$ and the linear coefficient $E_{l,k,i,s}$ to get,

$$J^{+}_{i,s} = \sum_{s' \in S_{i,s}} \tilde{T}_{i,s,s'} J^{-}_{i,s'} + \sum_{l=0}^{L} \sum_{k=-l}^{l} E_{l,k,i,s} Q_{l,k,i}, \tag{3.27a}$$

$$\begin{aligned} \tilde{T}_{i,s,s'} &= \sum_{m \in M^{+}_{s} \cap M^{-}_{s'}} w_m \left| \hat{\Omega}_m \cdot \hat{n}_{i,s} \right| T_{m,i,s,s'} \tilde{h}^{-}_{m,i,s'}, \\ &= \frac{\sum_{m \in M^{+}_{s} \cap M^{-}_{s'}} w_m \left| \hat{\Omega}_m \cdot \hat{n}_{i,s} \right| T_{m,i,s,s'} \psi^{-}_{m,i,s'}}{J^{-}_{i,s'}}, \end{aligned} \tag{3.27b}$$

$$E_{l,k,i,s} = \sum_{m \in M^{+}_{s}} w_m \left| \hat{\Omega}_m \cdot \hat{n}_{i,s} \right| Y_{l,k}(\hat{\Omega}_m) E_{m,i,s}, \tag{3.27c}$$

where $S_{i,s}$ is the set of surfaces of cell i whose incoming angular flux contributes to the outgoing angular flux of surface s. Note that the RMA transmission coefficient is directly computed by a partial current weighting of the original transport transmission coefficients.

We begin to close the system of equations by enforcing that the partial current is continuous at the interfaces such that,

$$J^{\pm}_{i,s} = J^{\mp}_{i',s'},$$

where i' is the index of the cell adjacent to cell i at the interface s and s' is the surface of cell i' such that $(i,s) \cap (i',s') \neq 0$. The next step towards forming a closed system of equations is to rewrite $Q_{l,k,i}$ in terms of the partial currents. First, we define the nonlinear relation,

$$\tilde{E}_{i,s} = \sum_{l=0}^{L} \sum_{k=-l}^{l} E_{l,k,i,s} Q_{l,k,i} / Q_{0,0,i}. \tag{3.28}$$

The nonlinear coefficient of Eq. (3.28) is the source-weighted escape probability of the original transport equation. It is important to notice that the nonlinear coefficient takes into account the contribution of the anisotropy of the source to the outgoing partial current. This attribute is of fundamental importance when applying RMA to problems where the anisotropy of the scattering is pronounced. Substituting Eq. (3.28) and the definition of the source from Eq. (3.24c), into Eq. (3.27a) gives us the transmission equation of RMA,

$$J^{+}_{i,s} = \sum_{s' \in S_{i,s}} \tilde{T}_{i,s,s'} J^{-}_{i,s'} + \tilde{E}_{i,s} \left[\Sigma_{sc,0,i} \bar{\phi}_{0,0,i} + Q_{Ext,0,0,i} \right]. \tag{3.29}$$

The next step is to define the relationship between the partial current and $\bar{\phi}_{0,0,i}$. There are two distinct approaches for accomplishing this that will lead us to the Explicit RMA (E-RMA) and Balance RMA (B-RMA) variants of the RMA method.

3.2.1 Closure relation for explicit RMA

We start from the step MOSC relation for the average scalar flux, as defined by Eq. (2.33b),

$$\bar{\phi}_{0,0,i} = \sum_{m=1}^{M} w_m \left[\sum_{s' \in S^{-}_{m,i}} I_{m,i,s'} \psi^{-}_{m,i,s'} + C_{m,i} Q_{m,i} \right].$$

Again, expanding $Q_{m,i}$ in terms of $Y_{l,k}(\hat{\Omega}_m)$ and substituting Eq. (3.25a) into the above equation gives us,

$$\bar{\phi}_{0,0,i} = \sum_{s' \in S_i} \tilde{I}_{i,s'} J^-_{i,s'} + \sum_{l=0}^{L} \sum_{k=-l}^{l} C_{l,k,i} Q_{l,k,i}, \tag{3.30a}$$

$$\begin{aligned} \tilde{I}_{i,s'} &= \sum_{m \in M^-_{s'}} w_m I_{m,i,s'} h^-_{m,i,s'}, \\ &= \frac{\sum_{m \in M^-_{s'}} w_m I_{m,i,s'} \psi^-_{m,i,s'}}{J^-_{i,s'}}, \end{aligned} \tag{3.30b}$$

$$C_{l,k,i} = \sum_{m=1}^{M} w_m Y_{l,k}(\hat{\Omega}_m) C_{m,i}, \tag{3.30c}$$

where S_i is the set of all the surfaces of cell i. Next, we define the nonlinear relation,

$$\tilde{C}_i = \sum_{l=0}^{L} \sum_{k=-l}^{l} C_{l,k,i} Q_{l,k,i} / Q_{0,0,i}. \tag{3.31}$$

and substitute it, along with the definition of the source from Eq. (3.24a), into Eq. (3.30a) to get,

$$\bar{\phi}_{0,0,i} = \sum_{s' \in S_i} \tilde{I}_{i,s'} J^-_{i,s'} + \tilde{C}_i \left[\Sigma_{sc,0,i} \bar{\phi}_{0,0,i} + Q_{Ext,0,0,i} \right].$$

Finally, we combine like terms and solve $\bar{\phi}_{0,0,i}$ to be,

$$\bar{\phi}_{0,0,i} = \frac{1}{1 - \Sigma_{sc,0,i} \tilde{C}_i} \left[\sum_{s' \in S_i} \tilde{I}_{i,s'} J^-_{i,s} + \tilde{C}_i Q_{Ext,0,0,i} \right], \tag{3.32}$$

giving us the defining closure relationship of E-RMA.

It is important to note that the methodology presented in this section could be extended to the higher order angular moments of the flux. As defined by Eq. (2.33b), each angular moment of the flux depends on the incoming interface angular flux and all the angular moments of the flux. By using Eq. (3.25a) to solve each angular moment in terms of only the incoming partial current, the linear coefficients $E_{l,k,i,s}$ and $C_{l,k,i}$ could be used by RMA instead of $\tilde{E}_{i,s}$ and $\tilde{C}_i$. This way, the approximation made by Eqs. (3.28) and (3.31) would be avoided and the escape and collision coefficient would only have to be computed once. However, the system of equations that defines the angular moments of the flux as a function of the partial currents would be dimensioned by the number of angular moments and potentially very expensive to store in memory.

3.2.2 Closure relation for balance RMA

The second approach solves for the scalar flux using the balance equation, Eq. (2.64a), such that,

$$\bar{\phi}_{0,0,i} = \frac{\sum_{s \in S_i} A_{i,s} \left[J^-_{i,s} - J^+_{i,s} \right] + Q_{Ext,0,0,i} V_i}{[\Sigma_i - \Sigma_{sc,0,i}] V_i}, \tag{3.33}$$

where Σ_i and V_i are the total cross section and the volume of cell i, respectively. This approach is attractive since it requires the computation of less nonlinear coefficients and it is simpler to implement than E-RMA. However, the scalar flux is now dependent on both the incoming and outgoing partial currents. As will be shown in Section 3.2.4, this results in the B-RMA operator possessing twice the density relative to the E-RMA operator making it much more computationally expensive to solve.

3.2.3 Boundary conditions

As was done for SVRM, RMA uses the "true" boundary conditions. Since the unknowns of RMA are the outgoing partial currents, it is easy to implement the physical boundary conditions, defined in Section 2.10, instead of the non-linear albedo condition used by CMR or CMFD. The boundary conditions of RMA are as follows:
the **reflective** boundary condition

$$J^-_{i,s} = J^+_{i,s},$$

the **translation** boundary condition

$$\begin{aligned} J^-_{i,s} &= J^+_{i',s'}, \\ \vec{r}_{i,s} &= \vec{r}_{i',s'} + \vec{r}_{shift}, \end{aligned}$$

and the **vacuum** boundary condition

$$J^-_{i,s} = 0.$$

The boundary conditions of RMA can be expressed in a general form as,

$$J^-_{i,s} = v_{s\leftarrow s} J^+_{i,s} + v_{s\leftarrow s'} J^+_{i',s'},$$

The albedo coefficients $v_{s\leftarrow s}$ and $v_{s\leftarrow s'}$ are defined by Table (3.2).

	Reflective	Periodic	Vacuum
$v_{s\leftarrow s} =$	1	0	0
$v_{s\leftarrow s'} =$	0	1	0

Table 3.2 – Values of the albedo coefficients at the domain boundaries.

3.2.4 Solving RMA

By substituting Eq. (3.32) into (3.29), one obtains the following **E-RMA transmission equation**

$$J^+_{i,s} = \sum_{s'\in S_{i,s}} \hat{T}_{i,s,s'} J^-_{i,s'} + \Sigma_{sc,0,i} \tilde{E}_{i,s} \tilde{C}_i Q_{Ext,0,0,i}. \tag{3.34}$$

where the modified transmission coefficient, i.e. $\hat{T}_{i,s,s'}$, is defined by

$$\hat{T}_{i,s,s'} = \tilde{T}_{i,s,s'} + \Sigma_{sc,0,i} \tilde{E}_{i,s} \tilde{I}_{i,s'}. \tag{3.35}$$

Otherwise, if Eq. (3.33) is used in (3.29) to express the scalar flux, one obtains the **B-RMA transmission equation**

$$J^+_{i,s} = \sum_{s'\in S_{i,s}} \hat{T}_{i,s,s'} J^-_{i,s'} + \sum_{s'\in S_{i,s}} \hat{E}_{i,s,s'} J^+_{i,s'} + \frac{\Sigma_i \tilde{E}_{i,s} Q_{Ext,0,0,i}}{\Sigma_i - \Sigma_{sc,0,i}}. \tag{3.36}$$

where the modified transmission and escape coefficients, i.e. $\hat{T}_{i,s,s'}$ and $\hat{E}_{i,s,s'}$, respectively, are

$$\hat{T}_{i,s,s'} = \tilde{T}_{i,s,s'} + \hat{E}_{i,s,s'},$$

and

$$\hat{E}_{i,s,s'} = \frac{\Sigma_{sc,0,i}\tilde{E}_{i,s}A_{i,s'}}{V_i(\Sigma_i - \Sigma_{sc,0,i})}.$$

As Eq. (3.36) shows, the number of dependent terms is doubled for the B-RMA transmission equation with respect to the E-RMA transmission equation. On the other hand, the B-RMA coefficients are much simpler to compute with respect to the E-RMA variant. Also, the B-RMA assures that the neutron balance is automatically satisfied, as a result of Eq. (3.33).

By putting the region-wise equation of RMA, Eq. (3.34) or Eq. (3.36), into a global matrix form, we get

$$\boldsymbol{J}^+ = \underline{\tilde{\boldsymbol{A}}}\boldsymbol{J}^+ + \tilde{\boldsymbol{Q}}_{Ext,0,0,i}. \tag{3.37}$$

where $\boldsymbol{J}^+ = \{J^+_{i,s}\}_{i\in[1,I]}$ is the global region-wise outgoing partial current vector. The global matrix $\underline{\tilde{\boldsymbol{A}}}$ is composed of the combination of the global transmission matrix ($\underline{\tilde{\boldsymbol{T}}}$), representing the contribution of the interior surfaces, and the homogeneous boundary condition matrix $\underline{\tilde{\boldsymbol{B}}}$, representing the contributions of the boundary surfaces, such that

$$\underline{\tilde{\boldsymbol{A}}} = \underline{\tilde{\boldsymbol{T}}} + \underline{\tilde{\boldsymbol{B}}}.$$

It is important to note that $\underline{\tilde{\boldsymbol{B}}}$ is explicitly solved, not iteratively like the standard MOC scheme. The modified external source takes into account the contribution from the external source and the fixed incoming boundary current $\hat{\boldsymbol{J}}_{in}$ such that,

$$\tilde{\boldsymbol{Q}}_{Ext,0,0,i} = \underline{\underline{\boldsymbol{E}}}Q_{Ext,0,0,i} + \hat{\boldsymbol{J}}_{in}. \tag{3.38}$$

The iterative scheme for solving RMA is as follows,

$$\boldsymbol{J}^{+,(n+1)} = \underline{\tilde{\boldsymbol{A}}}^{(n+1/2)}\boldsymbol{J}^{+,(n+1)} + \tilde{\boldsymbol{Q}}_{Ext,0,0,i}.$$

As previously stated, one of the attributes of step MOSC is that it is guaranteed to have a positive solution. Since RMA is a nonlinear coarsening of step MOSC, it is logical to assume that it would retain this attribute. However, since RMA is nonlinear it is difficult to prove that it is positive definite. Instead, we justify our assumption is also justified by the fact that the trace of $\underline{\hat{\boldsymbol{I}}} - \underline{\underline{\tilde{\boldsymbol{A}}}}^{(n+1/2)}$ is guaranteed to be positive for physical cross section values (as is shown in Section 3.2.10) suggesting that most of the eigenvalues are positive. Moreover, if one were to consider a symmetric polyhedron, with isotropic scattering and a symmetric transport solution, then $\underline{\hat{\boldsymbol{I}}} - \underline{\underline{\tilde{\boldsymbol{A}}}}^{(n+1/2)}$ would be symmetric. We use these observations to assume that $\underline{\hat{\boldsymbol{I}}} - \underline{\underline{\tilde{\boldsymbol{A}}}}^{(n+1/2)}$ will be positive definite. We take advantage of this assumption by choosing to use the Bi-Stabilized Conjugate Gradient Method (BiStabCG), [31], which is typically an efficient technique for positive definite matrices. It is important to note that BiStabCG was chosen instead of the conjugate gradient method in order to ensure convergence even when our assumption of the matrix being positive definite is not true. Empirical results to support the above supposition are presented later in Section 5.

3.2.5 Prolongation operator

After the RMA operator has been solved, $\psi^{-}_{m,i,s'}, \psi^{+}_{m,i,s}$ are corrected according to Eq. (3.25a) in order to maintain consistency,

$$\psi^{\pm,(n+1)}_{m,i,s} = \tilde{h}^{\pm,(n+1/2)}_{m,i,s} J^{\pm,(n+1)}_{i,s}, \quad \text{when } m \in M^{\pm}_{i,s}.$$

However, the choice about how to update the higher order angular moments of the scalar flux is less straight forward. For simplicity we have chosen to update $\bar{\phi}_{l,k,i}$ by using the classic rebalancing relation,

$$\bar{\phi}^{(n+1)}_{l,k,i} = \bar{\phi}^{(n+1/2)}_{l,k,i} \frac{\bar{\phi}^{(n+1)}_{0,0,i}}{\bar{\phi}^{(n+1/2)}_{0,0,i}}. \tag{3.39}$$

3.2.6 RMA applied to a vacuum region

Inside a region approaching a vacuum, the collided flux goes to zero, as explained in Section 2.16.2. Thus, the contributions of the collided flux to the outgoing angular flux and the scalar flux also go to zero. This scenario is often difficult to approach using methods built off the diffusion coefficient $\frac{1}{3\Sigma_g(\vec{r})}$, since as the cross section goes to zero, a singularity is created. If the external source within the vacuum region is set to zero, then the E-RMA equation (3.34) becomes,

$$J^{+}_{i,s} = \sum_{s' \in S_{i,s}} \tilde{T}_{i,s,s'} J^{-}_{i,s'},$$
$$\bar{\phi}_{0,0,i} = \sum_{s \in S_i} \tilde{I}_{i,s} J^{-}_{i,s}.$$

Thus, E-RMA does not require any modifications in order to handle a vacuum region.

However, B-RMA does not have the same robustness as E-RMA. According to Eq. (3.33), a singularity maybe created as result of numerical precision as the medium approaches a vacuum. Therefore, when the optical thickness of a region approaches the numerical precision limit of the program, E-RMA should be used. However, it is important to note that analytically Eq. (3.33) does not become a singularity as the medium approaches a vacuum.

3.2.7 Comments about the non-linearity

The RMA coefficients have two different sources of nonlinearity, even when RMA shares the same spatial discretization as transport. The transmission coefficient of RMA, Eq. (3.27b), and the incident coefficient of E-RMA, Eq. (3.30b), are non-linear because they are computed with an interface-angular-flux weighting. This reduction takes into account the anisotropy of the angular flux among the region.

The second source of non-linearity is contained in the escape coefficient, Eq. (3.28), of RMA and the collision coefficient of E-RMA, Eq. (3.31). This nonlinearity accounts for the anisotropy of the scattering source and the fixed source. With regards to the escape coefficient, RMA is projecting the contributions of the anisotropic source to the outgoing

angular flux on to $Q_{0,0,i}$ using Eq. (3.28). Moreover, E-RMA projects the contributions of the anisotropic source to the scalar flux onto $Q_{0,0,i}$ using Eq. (3.31). In this way, RMA accounts for the contributions of the anisotropic components of the source without incurring the computational cost necessary to solve them separately. This is an important quality of RMA, since methods such as CMFD do not account for these contributions. As will be seen later in Section 5, this attribute allows RMA to better maintain its effectiveness than CMFD for problems with highly anisotropic scattering.

In the following sections, two other sources of non-linearity for RMA will be introduced. The first is associated to the spatial homogenization of the transport mesh. When RMA is applied on a coarse mesh, its coefficients are obtained by a non-linear spatial weighting of the contribution of the fine regions to the coarse regions. The second source of nonlinearity comes from the application of RMA to high-order spatial discretization schemes. In particular, when the source and the flux are expanded on local basis functions, the RMA coefficients collapse the high order spatial moments onto the zero-order spatial moment.

3.2.8 Extension of RMA to heterogeneous cells: homogenization effect

By applying assumption (3.25a) to the transport equation for heterogeneous cells, Eq. (2.38a), one obtains the transmission over the coarse node i,

$$J^{+}_{i,s} = \sum_{s' \in S^{-}_{i,s}} \tilde{T}_{i,s,s'} J^{-}_{i,s'} + \hat{E}_{i,s} \left[\Sigma_{sc,0,i} \bar{\phi}_{0,0,i} + Q_{Ext,0,0,i} \right], \tag{3.40}$$

which is formally identical to Eq. (3.29), but with the new definition for the non-linear escape coefficient,

$$\hat{E}_{i,s} = \frac{\sum_{m \in M^{+}_{s}} w_m \left| \hat{\Omega}_m \cdot \hat{n}_{i,s} \right| \sum_{l=0}^{L} \sum_{k=-l}^{l} \sum_{\alpha \in i} V_\alpha Y_{l,k}(\hat{\Omega}_m) E_{l,k,i,\alpha,s} Q_{l,k,i,\alpha}}{\sum_{\alpha \in i} V_\alpha Q_{0,0,i,\alpha}}, \tag{3.41}$$

where V_α is the volume of subregion α. In Eq. (3.40), the cross section $\Sigma_{sc,0,i}$ is the flux-weighted homogenized cross section of the node i. Moreover, $Q_{Ext,0,0,i}$ and $\bar{\phi}_{0,0,i}$ are the homogeneous volume-weighted external source and flux, respectively.

For the E-RMA, the closure equation for the scalar flux, Eq. (3.30a) remains formally the same except for the new collision coefficient,

$$\tilde{C}_i = \frac{\sum_{m=1}^{M^{-}} w_m \sum_{l=0}^{L} \sum_{k=-l}^{l} Y_{l,k}(\hat{\Omega}_m) \sum_{\alpha \in i} V_\alpha C_{l,k,i,\alpha} Q_{l,k,i,\alpha}}{\sum_{\alpha \in i} V_\alpha Q_{0,0,i,\alpha}}. \tag{3.42}$$

The coefficients (3.41) and (3.42) take into account the heterogeneous and the non-uniform distribution of neutron collisions among the heterogeneous region i.

3.2.9 Extension of RMA to high-order spatial moments of the transport operator

When RMA is applied to a transport operator with high-order spatial moments, the fundamental closure equation (3.25a) becomes,

$$\boldsymbol{\psi}^{\pm}_{m,i,s} = \tilde{\boldsymbol{h}}^{\pm}_{m,i,s} J^{\pm}_{i,s}, \tag{3.43}$$

where $\boldsymbol{\psi}^{\pm}_{m,i,s}$ and $\tilde{\boldsymbol{h}}^{\pm}_{m,i,s}$ are vectors of the spatial moments of $\psi^{\pm}_{m,i,s}$ and $\tilde{h}^{\pm}_{m,i,s}$, respectively. By applying $\sum_{m\in M^{\pm}_{i,s}} w_m \left|\hat{\Omega}_m \cdot \hat{n}_{i,s}\right|$ and Eq. (3.43) to Eq. (2.34b) we get,

$$\begin{aligned} &\boldsymbol{R}_s \cdot \tilde{\boldsymbol{h}}^{\pm}_{m,i,s} = 1, \\ &\boldsymbol{R}_s = \frac{1}{A_{i,s}} \int_{\Gamma_{i,s}} d^2\, r_s \boldsymbol{P}_s(\vec{r}_s) \sum_{m\in M^{+}_{s}} w_m |\hat{\Omega}_m \cdot \vec{n}_{i,s}|\cdot, \end{aligned} \tag{3.44}$$

to the high-order transmission equation (2.36a), one obtains the non-linear transmission equation of RMA,

$$J^{+}_{i,s} = \sum_{s'\in S^{-}_{i,s}} \hat{T}_{i,s,s'} J^{-}_{i,s'} + \hat{E}_{i,s}\left[\Sigma_{sc,0,i}\bar{\phi}_{0,0,i} + Q_{Ext,0,0,i}\right], \tag{3.45}$$

which is formally identical to Eq. (3.29), but with the new definition for the non-linear coefficients,

$$\begin{aligned} \hat{T}_{i,s,s'} &= \boldsymbol{R}_{s,s'} \underline{\underline{\boldsymbol{T}}}_{m,i,s,s'} \tilde{\boldsymbol{h}}^{-}_{m,i,s'}, \\ &= \boldsymbol{R}_{s,s'} \frac{\underline{\underline{\boldsymbol{T}}}_{m,i,s,s'} \boldsymbol{\psi}^{-}_{m,i,s'}}{J^{-}_{i,s'}}, \end{aligned} \tag{3.46a}$$

$$\begin{aligned} \hat{E}_{i,s} &= \boldsymbol{R}_s \frac{\underline{\underline{\boldsymbol{E}}}_{m,i,s} \boldsymbol{Q}_{m,i}}{Q_{0,0,i}}, \\ \boldsymbol{R}_{s,s'} &= \frac{1}{A_{i,s}} \int_{\Gamma_{i,s}} d^2\, r_s \boldsymbol{P}_s(\vec{r}_s) \cdot \sum_{m\in M^{+}_{s}\cap M^{-}_{s'}} w_m \left|\hat{\Omega}_m \cdot \hat{n}_{i,s}\right| \cdot . \end{aligned} \tag{3.46b}$$

By applying the projection operator defined by,

$$\boldsymbol{R}_i = \frac{1}{V_i} \int_{D_i} d^3\, r \boldsymbol{P}(\vec{r}) \sum_{m\in M} w_m \cdot$$

to the high-order collision equation (3.24b), one obtains the collision non-linear equation for E-RMA,

$$\bar{\phi}_{0,0,i} = \sum_{s\in S^{-}_{i}} \frac{\hat{I}_{i,s}}{1-\Sigma_{sc,0,i}\hat{C}_i} J^{-}_{i,s} + \frac{\hat{C}_i}{1-\Sigma_{sc,0,i}\hat{C}_i} Q_{Ext,0,0,i}, \tag{3.47}$$

which has the same form as Eq. (3.32), but with the new definition for the non-linear coefficients,

$$\begin{aligned} \hat{I}_{i,s'} &= \tilde{\boldsymbol{R}}_{i,s} \underline{\underline{\boldsymbol{I}}}_{m,i,s'} \boldsymbol{h}^{-}_{m,i,s'}, \\ &= \tilde{\boldsymbol{R}}_{i,s} \frac{\underline{\underline{\boldsymbol{I}}}_{m,i,s'} \boldsymbol{\psi}^{-}_{m,i,s'}}{J^{-}_{i,s'}}, \end{aligned} \tag{3.48a}$$

$$\hat{C}_i = \boldsymbol{R}_i \frac{\underline{\underline{\boldsymbol{C}}}_{m,i} \boldsymbol{Q}_{m,i}}{Q_{0,0,i}}. \tag{3.48b}$$

$$\tilde{\boldsymbol{R}}_{i,s} = \frac{1}{V_i} \int_{D_i} d^3\, r \boldsymbol{P}(\vec{r}) \cdot \sum_{m\in M^{-}_{s'}} w_m \cdot . \tag{3.48c}$$

For B-RMA, no modifications are required for computing the scalar flux since the balance equation used by B-RMA does not depend on the spatial basis functions of the transport

operator.

Note that Eqs. (3.46a), (3.46b), (3.48a) and (3.48b) work for non orthogonal basis functions. In the case that the basis functions $\boldsymbol{P}_s(\vec{r}_s)$ and $\boldsymbol{P}(\vec{r})$ are orthogonal on the surfaces and in the volume, respectively, then the spatial integrals inside Eqs. (3.46a),(3.46b),(3.48a) and (3.48b) become

$$\frac{1}{A_{i,s}} \int_{\Gamma_{i,s}} d^2 \, r_s \boldsymbol{P}_s(\vec{r}_s) = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{3.49}$$

$$\frac{1}{V_i} \int_{D_i} d^3 r \; \boldsymbol{P}(\vec{r}) = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{3.50}$$

and the RMA coefficients are obtained by collapsing the transport matrices on the zero-order spatial moment.

For clarity, the explanation of how RMA is expanded for heterogeneous cells and for the high order moments of the transport operator was separated in the previous two sections. However, it is important to note that both expansions can be used together. In fact, many of the cases that will be investigated in Section 5 are composed of heterogeneous cells and are solved with the linear short characteristic transport operator.

3.2.10 Positivity of RMA

RMA guarantees a positive solution if the transport scheme ensures positive results. It is indeed evident that the coefficients of E-RMA and B-RMA, $\tilde{T}$, $\tilde{I}$, $\tilde{E}$ and $\tilde{C}$, are defined by the ratio and the linear combination of positive quantities. Thus, the resulting RMA coefficients are positive. Moreover, if the transport scheme respects the theoretical limit in a vacuum and in optically thick region, as is the case for step MOC and MOSC, the global matrix of RMA ($\underline{\underline{\hat{\boldsymbol{I}}}} - \underline{\underline{\tilde{\boldsymbol{A}}}}$) is diagonally dominant because the elements of $\underline{\underline{\boldsymbol{A}}}$ are less than 1. In particular,

$$(1 - \tilde{A}_{i,s,s}) \geq \sum_{s' \neq s} \tilde{A}_{i,s,s'},$$

where $\tilde{A}_{i,s,s'}$ is an element of $\underline{\underline{\boldsymbol{A}}}$ and is explicitly defined by Eq. (3.35) when s' does not belong to the boundary of the domain.

3.2.11 Comparison with the DP0 method

RMA has a lot in common with the DP0 method, which was investigated by Sanchez and Chetaine, [33], and later expanded to DPN by Santandrea and Sanchez, [34]. Both methods build their coefficients from the transport coefficients and solve for the outgoing partial current. As a result, the equations of RMA and DP0 have very similar forms, however there are some important differences. DP0 is a linear acceleration method and

therefore must approximate the angular flux from the partial current such that,

$$\psi^{\pm}_{m,i,s} \approx \chi^{\pm}_{m,i,s} J^{\pm}_{i,s}, \tag{3.51}$$

where

$$\chi^{\pm}_{m,i,s} = \begin{cases} 1 & \text{if } \hat{\Omega}_m \in M^{+}_{i,s}, \\ 0 & \text{otherwise.} \end{cases} \tag{3.52}$$

As with all linear methods, this approximation is remedied by using additive correction,

$$\psi^{\pm,(n+1)}_{m,i,s} = \psi^{\pm,(n+1/2)}_{m,i,s} + \chi^{\pm}_{m,i,s} \left[J^{\pm,(n+1)}_{i,s} - J^{\pm,(n+1/2)}_{i,s} \right]$$

In contrast, the fundamental assumption of RMA, $\psi^{\pm}_{m,i,s} = \tilde{h}^{\pm}_{m,i,s} J^{\pm}_{i,s}$, does not approximate the angular distribution of the angular flux. Moreover, $\chi^{\pm}_{m,i,s}$ is a constant characteristic function while $\tilde{h}^{\pm}_{m,i,s}$ is a transport computed shape function of the anisotropy of the interface flux.

DP0 shares the same spatial and energy mesh as the transport equation, as is required for a linear synthetic acceleration method. Instead, RMA can make use of spatial homogenization and energy condensation to form a separate coarser mesh. Thus, RMA can reduce its memory imprint relative to DP0. This is especially important when RMA is applied to 3D, full core simulations. However, RMA does have the drawback that its coefficients have to be recomputed with each iteration of the transport operator.

3.2.12 Cumulation of nonlinear coefficients

One of the notable attributes of RMA compared to CMFD is that its nonlinear coefficients are formulated using the transport coefficients. This results in RMA requiring the cumulation of data not typically provided by the transport operator. For instance, both RMA variants require the transport operator to cumulate the numerators of $\tilde{T}_{i,s,s'}$ and $\tilde{E}_{i,s}$, i.e. Eqs. (3.27b) and (3.27c), on top of the more commonly required cumulation of the partial current. Moreover, E-RMA requires the transport operator to cumulate the numerators of $\tilde{I}_{i,s'}$ and $\tilde{C}_i$, i.e. Eqs. (3.30b) and (3.31) as well. This means that in order to use RMA, the transport solver has to be modified to cumulate the required data. As well, this additional cumulation results in a much larger computational overhead for generating the RMA method compared to CMFD, as will be seen in Section 5.

3.2.13 Convergence analysis

We begin by analyzing RMA using the same three tests presented in Section 2.16 and finish with a parameterized heterogeneous case. The RMA equations in matrix form for 1D geometries are detailed in Appendices B.4 to B.4.2. It is important to note that due to the similarity of E-RMA and B-RMA, they often share the same spectral radius, causing the results of E-RMA to be hidden behind those of B-RMA. However, if the reader looks closely, they will notice that the star marker of E-RMA can still be observed behind the B-RMA results.

3.2.13.1 Homogeneous geometry

With regards to the results of the infinite slab test shown in Fig. 3.8, the spectral radius of MOC accelerated by E-RMA and B-RMA is very close to zero for all frequencies and media compositions. This result is expected, since the only assumption made by RMA is that the angular distribution of the interface flux is fixed within each hemisphere. This assumption is true when the distribution of the angular flux of the current iteration is the same as the final solution. For all three test cases the final distribution of the angular flux is flat. Since the scattering source is isotropic, only the contribution from the previous iteration of the incoming angular flux can cause the outgoing angular flux to have a non-flat distribution. However, for the standard SI scheme, only the boundary angular flux is from the previous iteration, the contribution of which goes to zero for an infinite domain. A more thorough analysis of this phenomenon is discussed in Section 2.16.1.1. Thus, for the infinite slab test, RMA is essentially solving the transport operator.

In order to prove our assertion that the only assumption made by RMA is the fixed angular shape function, will modify the problem to use a S2 angular distribution. In 1D, this means that each partial current is composed of a single angular flux, making the fixed angular shape function assumption irrelevant. Thus, if our assertion is correct, the spectral radius of RMA should go to zero. This can be seen to be true for both the standard and Gauss-Jacobi iteration scheme for an arbitrary finite domain in Fig. 3.9.

Returning to a S8 angular distribution, the spectral radius of E-RMA and B-RMA is less than that of SI and the MCNH variant of CMFD, everywhere, in Fig. 3.10. The largest improvement in the spectral radius of RMA compared to MCNH occurs when the scattering ratio is set to 0.99 and the optical thickness is greater than 0.8 mfp. As noted in Section 2.20, this rapid divergence in the spectral radius occurs as the result of MCNH stimulating the high spatial frequencies of the transport solution. In particular, MCNH's use of a step constant approximation for the spatial distribution of the flux stimulates these eigenmodes. In contrast, RMA does not stimulate these eigenmodes since it the contributions of the higher order spatial moments are account for by its formulation. The spectral radius of B-RMA lies directly on top of that of E-RMA. As explained before this is due to the similarity of E-RMA and B-RMA. This trend occurs for all the results presented in this section. In order to avoid redundancy no more comments will be made about it.

The spectral radius analysis of MCNH, E-RMA and B-RMA, for the finite slab test is shown by Fig. 3.11. Since the domain is finite, the incoming boundary angular flux influences the spectral radius for the standard iteration scheme. Which is why the spectral radius of RMA has the same exponential behaviour discussed in Section 2.16.2. The GJ iteration scheme, exhibits the same behavior except that the spectral radius is dominated by the transmission of the interface angular flux instead of the boundary angular flux. This is why the spectral radius of GJ iteration scheme accelerated by RMA is larger than the standard scheme accelerated by RMA, everywhere. When the medium is optically diffusive, the contribution of the scattering source is the leading order term for determining the convergence of SI. However, this contribution is nullified with the application of RMA. Throughout the figure, the spectral radius of RMA is equal to or less than that of MCNH. Moreover, it is less than 1 throughout the figure, meaning that the iterative scheme is stable.

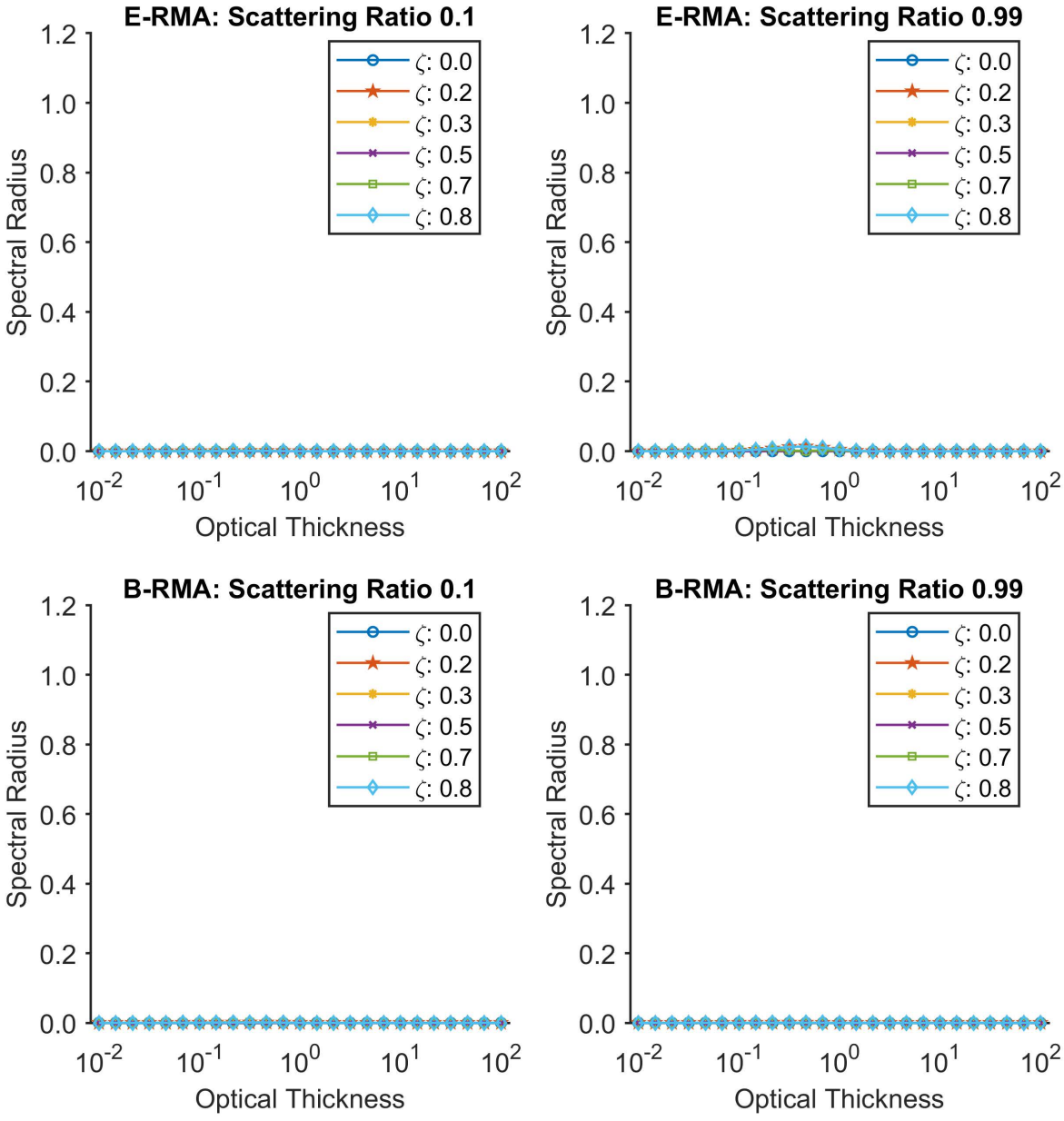

Figure 3.8 – Infinite slab test: a comparison of the spectral radius of several different spatial frequencies, in units of Hz, of the Fourier transformed solution of MOC accelerated by E-RMA and MOC accelerated by B-RMA for a infinite slab.

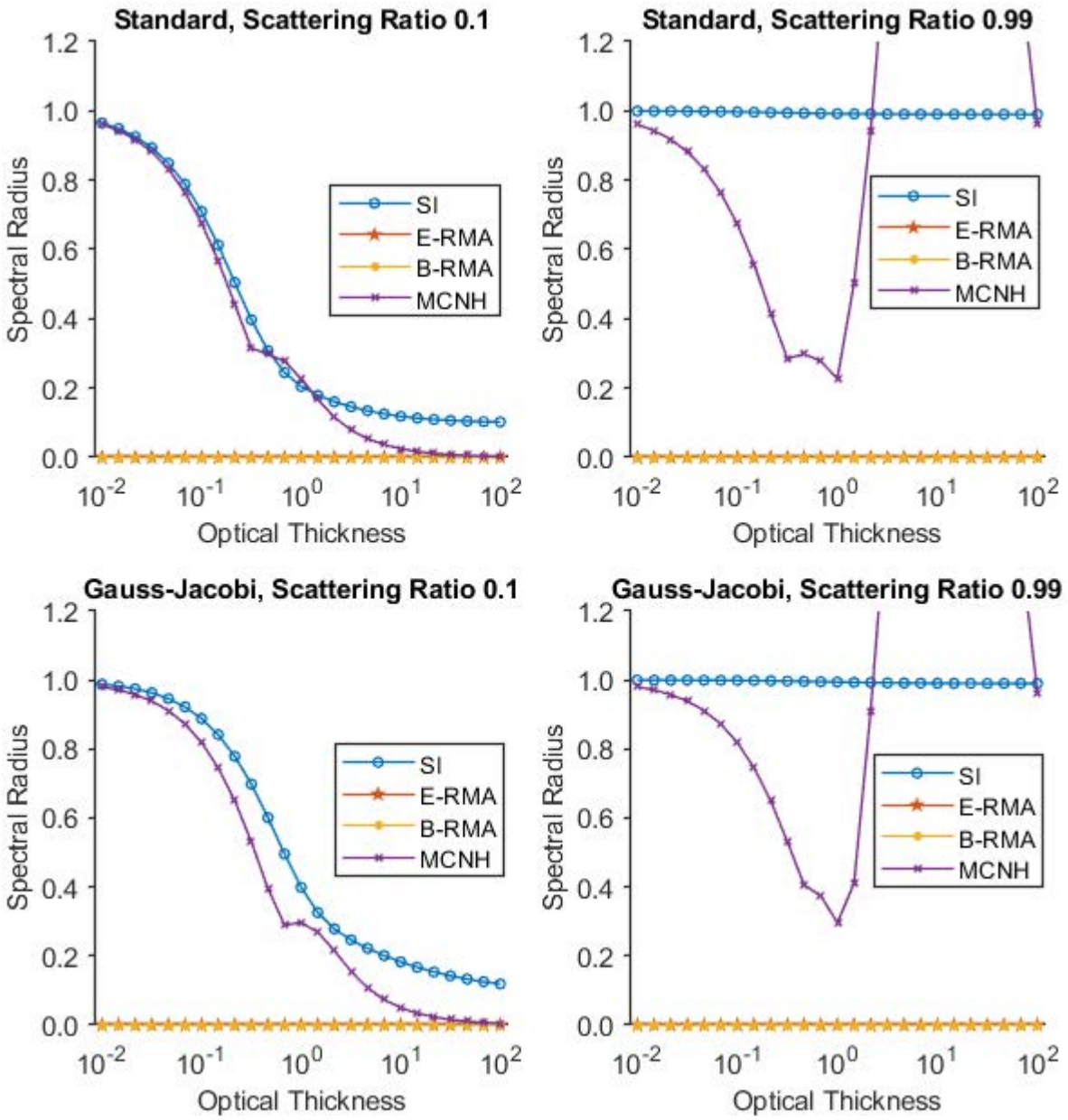

Figure 3.9 – Finite slab test: a comparison of the spectral radius analysis of SI, MOC accelerated by MCNH, MOC accelerated by E-RMA and MOC accelerated by B-RMA for a finite slab. A S_2 angular quadrature is used. The spectral radii of E-RMA and B-RMA completely overlap for all media compositions.

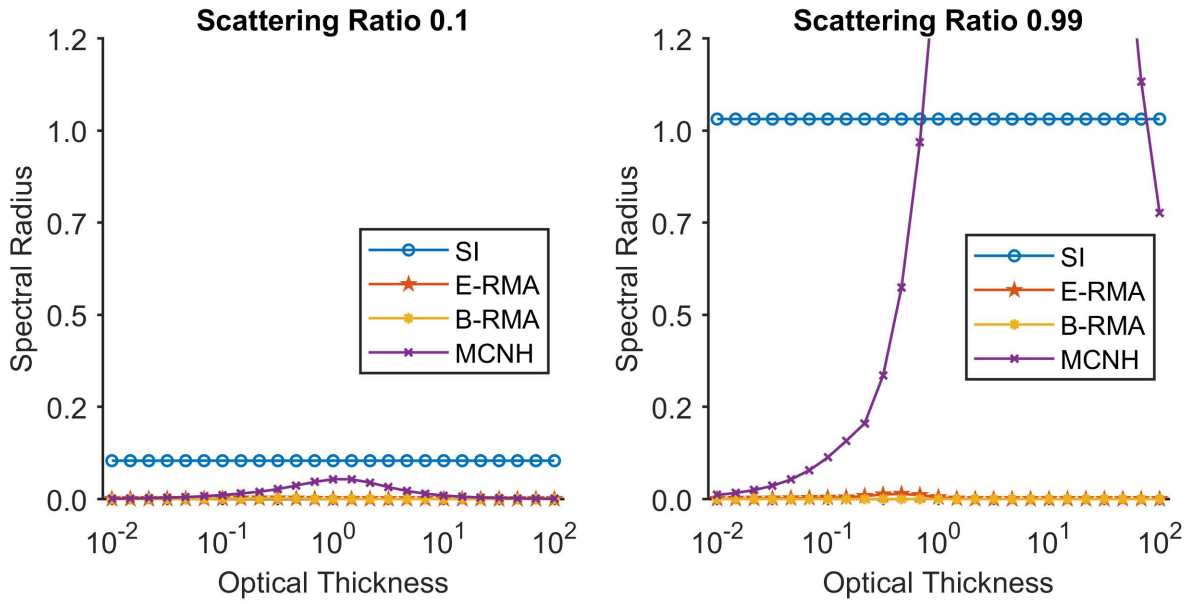

Figure 3.10 – Infinite slab test: a comparison of the spectral radius of MOC accelerated by MCNH, MOC accelerated by E-RMA and MOC accelerated by B-RMA for a infinite slab. The spectral radii of E-RMA and B-RMA completely overlap for all media compositions.

Next, the spectral radius of SI and MOC accelerated by E-RMA, B-RMA and MCNH are compared for an increasing number of cells according to the spatial discretization test. As is shown by Fig. 3.12, the spectral radius of RMA is less than that of MCNH everywhere. Moreover, the spectral radius of RMA goes towards zero as the number of cells increases, validating the Fourier analysis. Moreover, the spectral radius of both variants of RMA can be observed to be less than that of MCNH, everywhere.

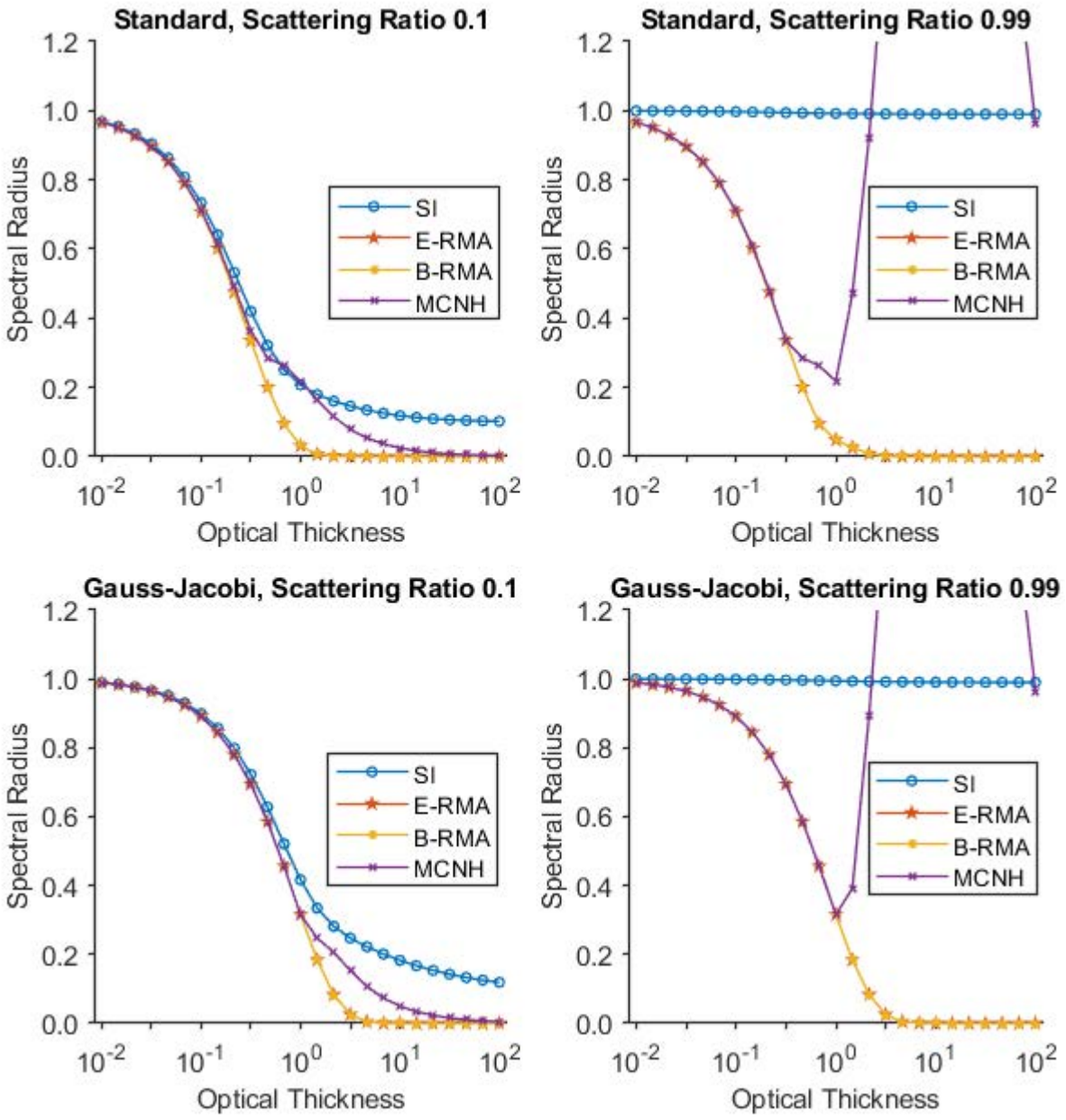

Figure 3.11 – Finite slab test: a comparison of the spectral radius analysis of SI, MOC accelerated by MCNH, MOC accelerated by E-RMA and MOC accelerated by B-RMA for a finite slab. The spectral radii of E-RMA and B-RMA completely overlap for all media compositions.

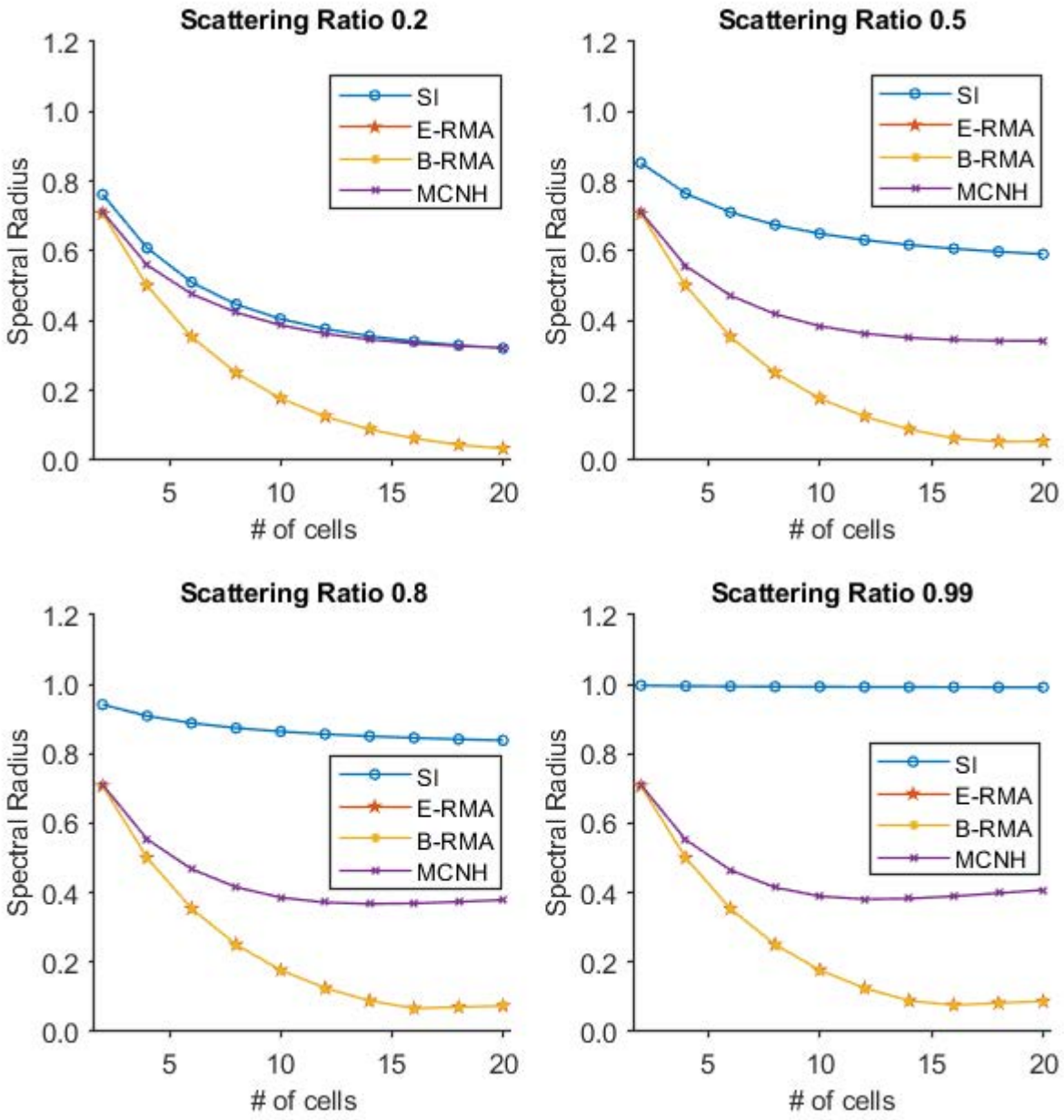

Figure 3.12 – Spatial discretization test: a comparison of the spectral radius analysis of SI, MOC accelerated by MCNH, MOC accelerated by E-RMA and MOC accelerated by B-RMA for an increasing number of cells. The optical thickness of each cell has been fixed to a value of 0.1 mfp. The spectral radii of E-RMA and B-RMA completely overlap for all media compositions.

3.2.13.2 Heterogeneous geometry

As explained in Section 2.18, the accuracy of the closure equations relating the surface current to the scalar flux significantly affects the performance and stability of the method. The most dramatic changes in the flux and current profile occur at boundaries with large material discontinuities. Consequently, the accuracy of the scalar flux and surface current relationship is of greatest significance at these highly discontinuous boundaries. Thus, in order to assess the accuracy of the scalar flux and surface current relationship used by RMA, we will study the behavior of RMA at boundaries with large material discontinuities. This meant that it was necessary to conduct a spectral radius analysis of RMA for heterogeneous geometries. The geometry was chosen to be composed of two distinct, repeating mediums. In order to study a wide range of discontinuities, a parameterization of the optical thickness and the scattering ratio was chosen, as was first done by Masiello for analyzing the stability of CMFD, [26]. The optical thickness and the scattering cross-section were fixed for one medium and varied for the other according to the following 1-parameter equations,

$$\tau_2(P) = \frac{(1-c_1)\tau_1}{P} + Pc_1\tau_1, \tag{3.53a}$$

$$c_2(P) = \frac{Pc_1}{\frac{(1-c_1)}{P} + Pc_1}, \tag{3.53b}$$

$$P > 0,$$

where τ_1 and τ_2 are the optical thickness of the fixed and parameterized mediums, respectively, and P is the parameter to be varied. Visualization summarizing the relation between P and τ_2 and c_2 are shown by Figs. 3.13 and 3.14, respectively. As dictated by Eq. (3.53a), when $c_1 < 1$, material 2 becomes infinitely absorbing, relative to material 1, as P goes to zero. In contrast, material 2 becomes an infinitely diffusive material, relative to medium 1, as P increases to the infinite. For the extreme cases ofc_1 we have,

$$\begin{aligned}
&\tau_2(P) = \frac{\tau_1}{P}, \quad \text{when } c_1 = 0 \\
&c_2(P) = 0, \text{and} \\
&\tau_2(P) = Pc_1\tau_1, \quad \text{when } c_1 = 1 \\
&c_2(P) = 1.
\end{aligned}$$

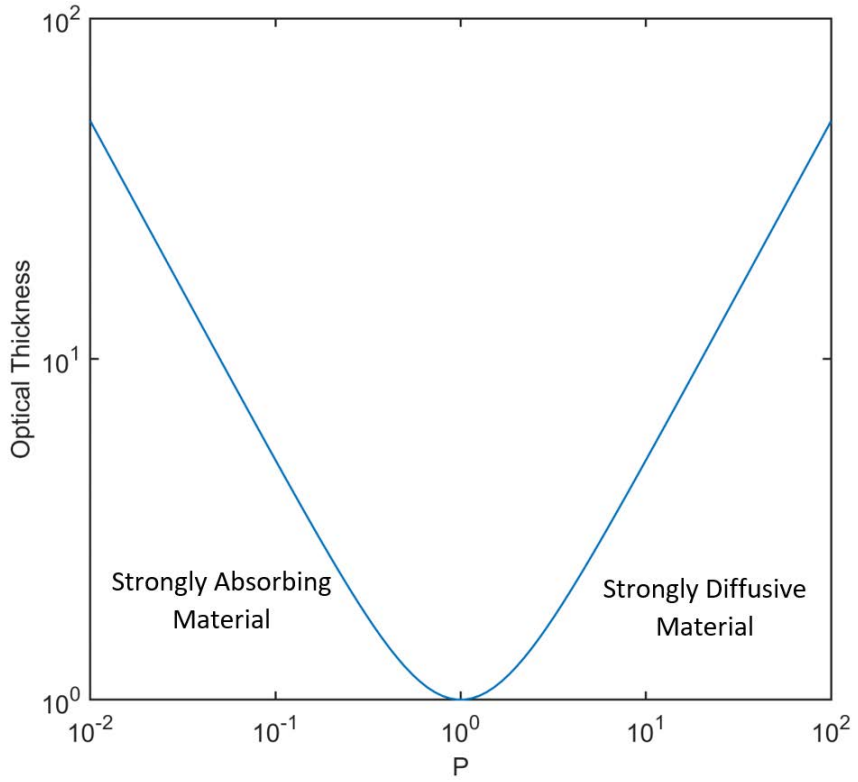

Figure 3.13 – The variation of τ_2 with P, according to Eq. (3.53a), when $tau_1 = 1$ and $c_1 = 0.5$

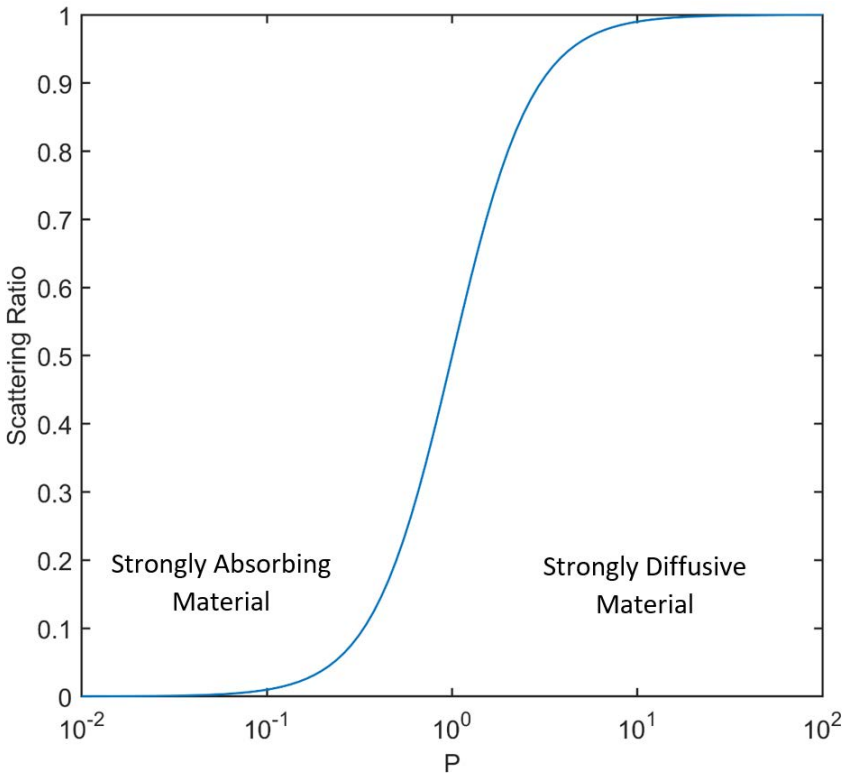

Figure 3.14 – The variation of c_2 with P, according to Eq. (3.53b), when $tau_1 = 1$ and $c_1 = 0.5$

The spectral radius of the parameterized problem was investigated for the case that the optical thickness of the first zone was set to 0.1, 1.0 and 10 mfp. The results of the analysis for SI, MCNH, E-RMA, and B-RMA is shown by Figs. 3.15, 3.16 and 3.17. For all the figures, the spectral radius of RMA is equal to or less than that of MCNH. When the optical thickness of the first zone is fixed to a value of 0.1, the spectral radius of RMA separates from that of MCNH when P is sufficiently small or large. The maximum value of the spectral radius of RMA remains relatively constant as the scattering ratio is varied. However, the position and width of the peak can be seen to change.

When $\tau_1 = 1$, the spectral radius of RMA is less than that of MCNH for all investigated scattering ratios. Moreover, the spectral radius of MCNH, increases rapidly when P reaches a threshold value. The value of P at which this divergence occurs, is dependent on the scattering ratio of the fixed medium. The maximum spectral radius of RMA occurs when the scattering ratio and P are set to 0.99 and 0.1 respectively.

Finally, when $\tau_1 = 10$, the spectral radius of RMA is very close to zero for all cases. Significant gap in the spectral radius of RMA and MCNH occurs when the scattering ratio is set to 0.8 or 0.99. Moreover, MCNH becomes unstable when the scattering ratio of the fixed medium is set to 0.99 and P exceeds a value of ~ 0.2.

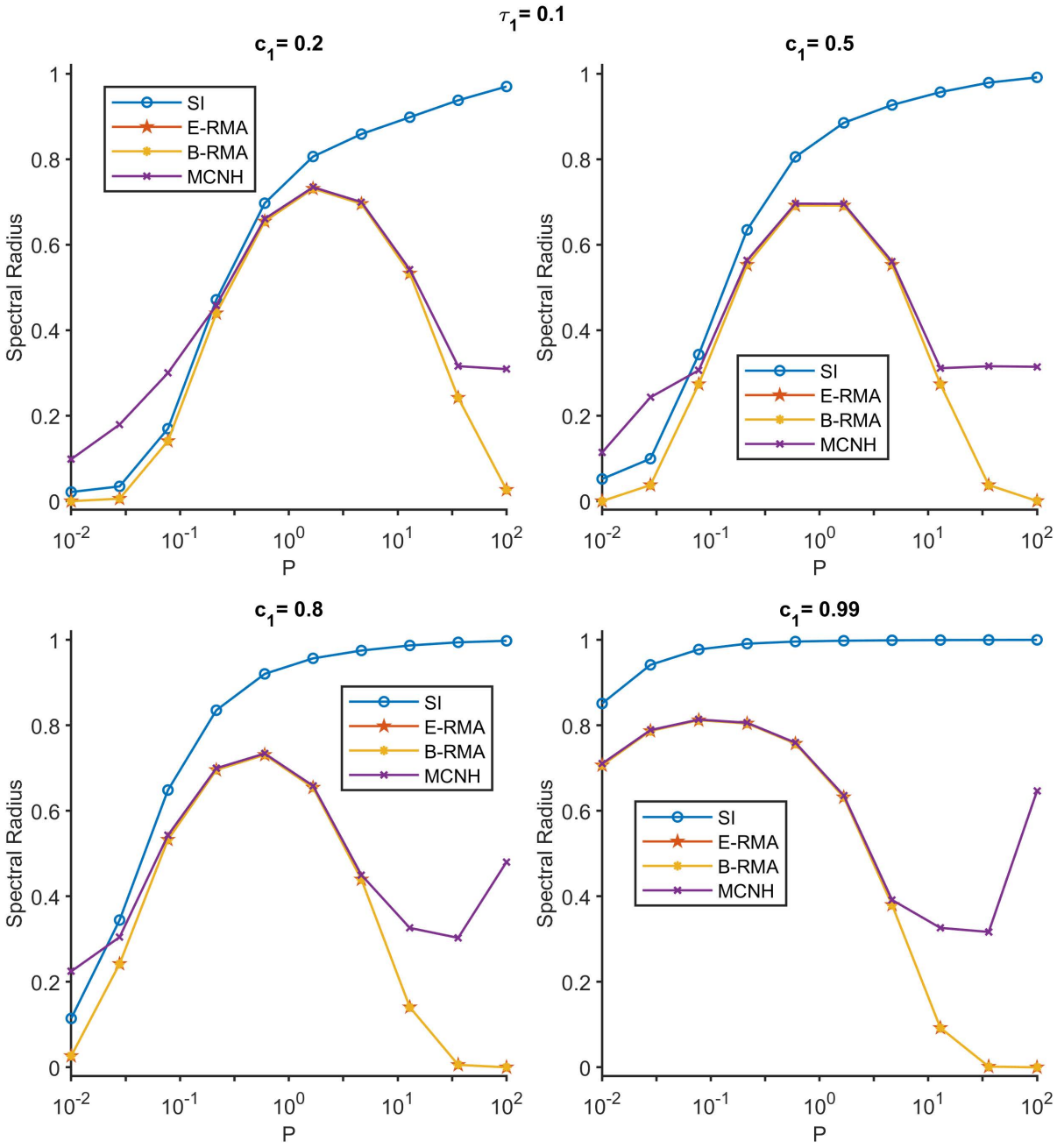

Figure 3.15 – A comparison of the spectral radius analysis of SI, MOC accelerated by MCNH, MOC accelerated by E-RMA and MOC accelerated by B-RMA for two repeating zones. Each subplot corresponding to a distinct scattering of the first zone. $\tau_1 = 0.1$ mfp for all the subplots. The optical thickness and scattering ratio of the second cell are set by Eqs. (3.53a) and (3.53a), respectively. The spectral radii of E-RMA and B-RMA completely overlap for all media compositions.

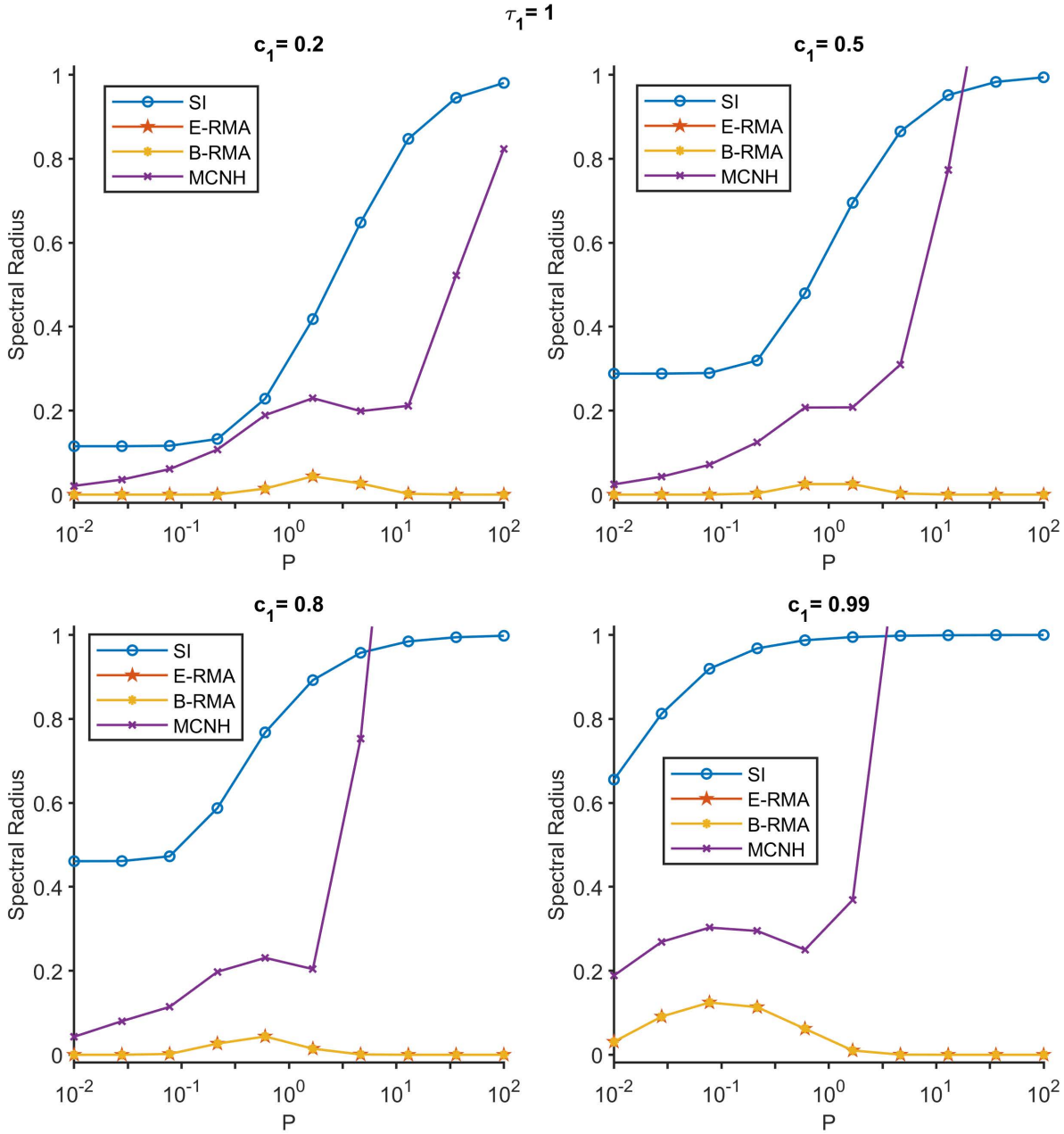

Figure 3.16 – A comparison of the spectral radius analysis of SI, MOC accelerated by MCNH, MOC accelerated by E-RMA and MOC accelerated by B-RMA for two repeating cells. Each subplot corresponding to a distinct scattering of the first zone. $\tau_1 = 1$ mfp for all the subplots. The optical thickness and scattering ratio of the second cell are set by Eqs. (3.53a) and (3.53a), respectively. The spectral radii of E-RMA and B-RMA completely overlap for all media compositions.

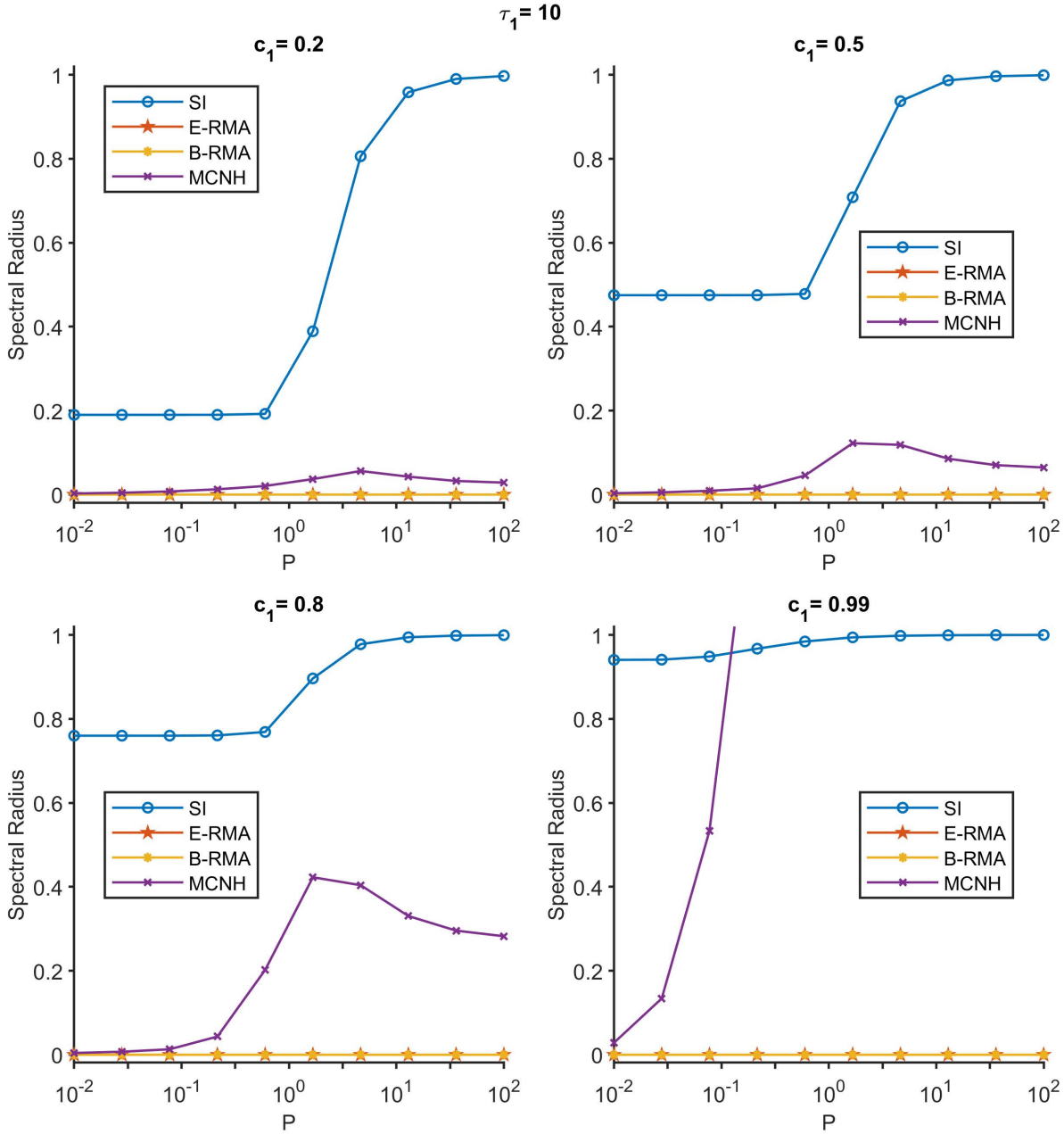

Figure 3.17 – A comparison of the spectral radius analysis of SI, MOC accelerated by MCNH, MOC accelerated by E-RMA and MOC accelerated by B-RMA for two repeating cells. Each subplot corresponding to a distinct scattering of the first zone. $\tau_1 = 10$ mfp for all the subplots. The optical thickness and scattering ratio of the second cell are set by Eqs. (3.53a) and (3.53a), respectively. The spectral radii of E-RMA and B-RMA completely overlap for all media compositions.

3.2.13.3 The multigroup hybrid method: alternating CMFD with RMA

RMA was shown in the previous section to have improved effectiveness compared to the MCNH variant of CMFD for all the cases analyzed. However, RMA is also more computationally expensive compared to MCNH due to its increased degrees of freedom and its need for additional data cumulation. Moreover, when the optical thickness of each cell is optically thin, the spectral radius of MCNH and RMA are nearly equivalent, as was shown by Figs. 3.10 and 3.11. Thus, we concluded that the ratio between the computational cost of applying the acceleration and its effectiveness could be optimized, by using RMA to solve the flux for energy groups when $\max_{i \in I}(\tau_{g,i}) > 1$ and MCNH otherwise. The limit $\max_{i \in I}(\tau_{g,i}) > 1$ was specifically chosen since Figs. 3.10 and 3.11 show that MCNH rapidly became unstable in this regime. We will refer to the resulting combination of the RMA and MCNH methods as the hybrid method. It is important to note that the hybrid acceleration scheme can be used to accelerate both the inner and the outer iterations as will be demonstrated later in the results section.

Since the geometry becomes more optically thick as the energy decreases, the hybrid method typically calls for MCNH to solve the fast energy groups and RMA to call the thermal energy groups. As a point of future work, it would be interesting to alternate between using CMFD and RMA for different spatial regions within a single energy group solve. Specifically, using the CMFD equations for cells with $\tau_{g,i} \leq 1$ and the RMA equations for cells with $\tau_{g,i} > 1$. This would likely result in a method that is less expensive to compute, since RMA would only be used for distinct energy and region pairs that are optically thick rather than for all regions of an energy that has at least one optically thick region.

Chapter 4

Acceleration library

In order to allow for the cross-development of new acceleration methods and to expedite the implementation of such methods in a user's code, we decided to create a library of acceleration methods. The library had to meet several criteria in order to be a viable solution. First, the library needed to be able to interface with any transport solver. Second, the library needed to be able to work with unstructured meshes. Third, the library needed to have access to high quality matrix solving routines. Fourth, the library had to be able to work efficiently in an HPC environment and utilize modern parallel paradigms such as GPGPU acceleration. Finally, the library must be able to work for an arbitrary iteration scheme. Beginning with a brief overview of the structure of the library, we will go through each one of these points in the following sections.

4.1 Library Structure

A simplified visualization of the structure of the library is shown below by Fig. 4.1. The data container and the manager classes are the only parts of the library that are directly linked to the user's code. Moreover, all the work related to generating and managing the library data, as well as solving the acceleration operator and updating the transport solution is done inside the library. Thus, only three lines of code are typically needed to apply the library within the user's code. This approach was chosen to minimize the interface between the user's code and the library, allowing the user to easily modify where and how the library is applied. However, there is the caveat that the user must provide routines to the library that either fetch the raw data required by the library to build the acceleration method or directly generate the required data. An example of a simple fetch routine for the scalar flux is shown by Algorithm 1.

Algorithm 1: Fetches the scalar flux.

input : g is the energy group index of the transport mesh,
i is the cell index of the transport mesh.
output: ϕ is the scalar flux of the transport mesh.
Function `GetFlux(`g, i`):`
  return $\phi[g,i]$

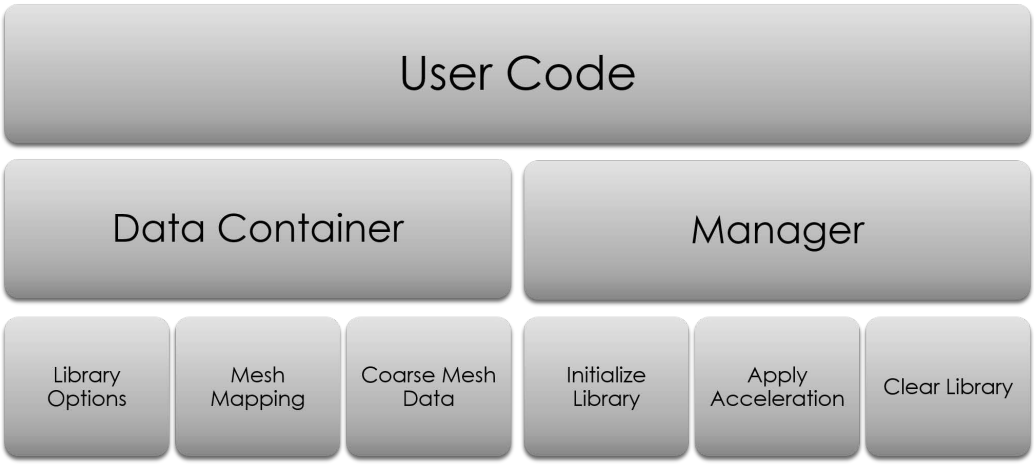

Figure 4.1 – A simplified visualization of the structure of the library.

4.1.1 Storage container

All the data required by the library is contained in one object. Thus, without changing any of the library functional calls, multiple acceleration methods and schemes can be used simply by modifying te data object or passing a different data object. The data contained by the storage class can be grouped into six categories, as illustrated by Fig. 4.2. The first partition, "Library Options", is responsible for defining which acceleration methods and solving schemes will be used when the library is applied. The second partition, "Cumulated Data", contains the cumulated data of the coarse (acceleration) mesh, discussed in Section 2.18.3. This data is used to generate the acceleration operator, unknowns and source. The third partition, "Fine-Coarse Mapping", stores the mapping between the fine (transport) mesh and the coarse mesh. These mappings are required for creating the cumulated data and for correcting the transport unknowns with the acceleration solution. The fourth partition, "Coarse Intra-Mapping", partition defines the mapping between adjacent cells of the coarse mesh. This information is needed by the library in order to construct the acceleration operator. when the global domain has been broken down into subdomains for parallel simulation, the fifth partition, "Subdomain Mapping", is responsible for storing the mapping between subdomains. The final partition, "PETSc Data", contains the PETSc data, [10]. This includes the acceleration matrix and all of the data required by PETSc to solve the acceleration matrix.

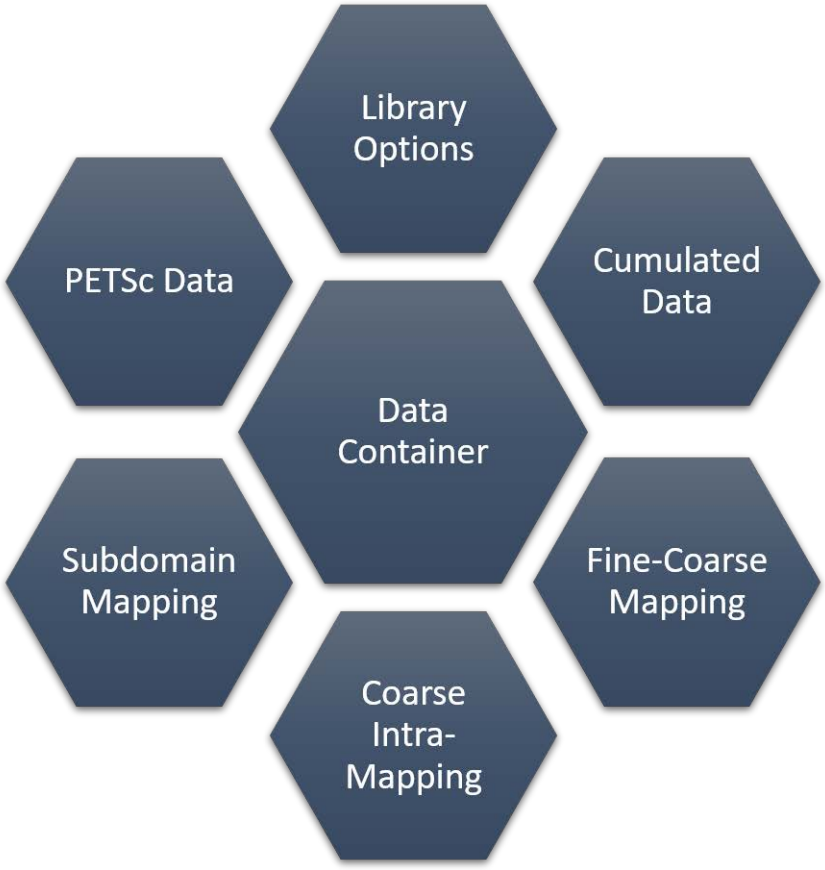

Figure 4.2 – A visualization of the library's data container.

4.1.2 Primary routines

The three primary routines that the user can access through the manager class are the **Initialization**, **Application**, and **Deletion** of the library. The initialization routine is responsible for creating all the data that normally remains constant throughout the usage of the library. This encompasses all the data contained by the storage class except for the cumulated data. The routine in charge of applying the acceleration library has three main tasks. First, it calls for any library data that is missing or needs to be updated to be generated. Nonlinear operators are dependent on the current solution transport solution. Thus, the cumulated data on the coarse mesh must always be regenerated during this step. Should the user want, even the data generated during the initialization of the library can be modified here. Next, the acceleration operator is solved. Finally, the unknowns of transport are corrected using the acceleration solution. The deletion routine is responsible for clearing all the data contained by the library storage class.

4.2 Library features

4.2.1 Contained acceleration methods

As previously discussed, RMA, CMR, and the MCNH, pCMFD and AAF variants of CMFD have all been implemented in the library. All acceleration methods share a common interface with the rest of the library and have access to the same set of homogenization routines. Moreover, to create an acceleration method in the library, the user must fill in a standard set of functions contained within a template class. These functions are responsible for defining what fine data must be homogenized, how to create the acceleration operator, unknowns, and source, and how to update the transport unknowns. This object-oriented approach was chosen not only to avoid redundant implementations, but also to ensure that all acceleration methods are capable of working with any combination of the library's options.

4.2.2 User interface

Since the library contains only nonlinear methods, it can accelerate any conservative transport operator. In order to take full advantage of this property, the library needs to be able to interface with any transport code. For this reason, the library has been designed to be used either as a true library or as a toolkit. It is left up to the user the level of abstraction that they will use the library with.

For rapid development and to maximize the compatibility of their code with different versions of the library, the library should be used purely like a library. In this case the library should only be accessed through the manager module. Using this type of implementation, the user has the option of completing the interface by simply defining a series of mapping functions that point to the data required by the library. The drawback of this approach is that the library routines responsible for homogenizing the user's data may assume an iteration order that is inconsistent with the actual ordering of the data. For most routines, the time wasted due to this poor data locality would be negligible compared to the cost of applying the acceleration. However, the cumulation of the fission and scattering source can incur a significant cost if the data locality is poor due to their

high dimensionality, as can be inferred from Eqs. (2.65c) and (2.65d).

To achieve better performance the user is encouraged to define their own homogenization routines for performance-critical parts of the code. This mixed approach allows the user to optimize the implementation time relative to the performance of the library. The user can also use the library purely as a toolkit and pick and choose the functions that they would like to use in their own code. However, this would require the longest implementation time and would greatly limit the compatibility of the user's code with different versions of the library.

4.2.3 Mapping

One of the noteworthy features of the library is its ability to work with meshes that are unstructured in energy and space. This flexibility makes it easier for the user to optimize the properties of the coarse mesh for the chosen acceleration method. As an example, the user can attempt to stabilize CMFD for their problem by ensuring that the optical thickness of each cell of the coarse mesh is less than 1 mfp, [14]. To implement this feature, the mapping between the fine and coarse mesh and the intra-mapping of the coarse mesh, had to possess certain capabilities. First, the fine to coarse mesh mapping had to allow for any contiguous block of cells on the fine mesh to be mapped to a cell on the coarse mesh. An example of this mapping is shown by Fig. 4.3.

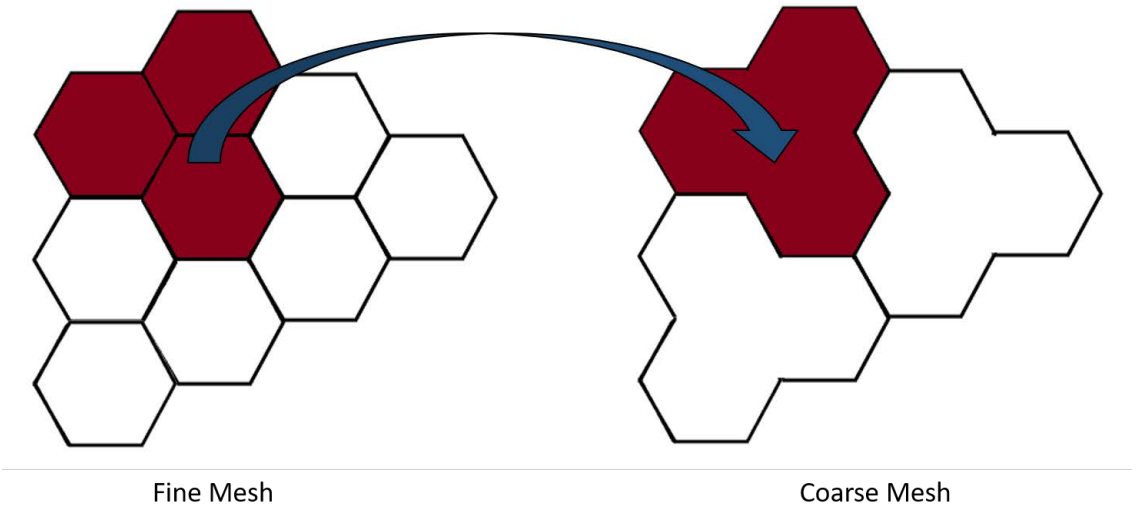

Figure 4.3 – A visualization of the mapping between the transport and acceleration mesh. A blue arrow is used to indicate the mapping from the fine mesh to the coarse mesh.

Second, each cell of the coarse mesh is responsible for knowing the cells adjacent to it in both energy and space. Moreover, each cell is responsible for knowing if it is adjacent to the boundary of the problem. An example of the mapping between cells that are adjacent in space and energy is shown by Fig. 4.4. Since the cross section can differ greatly with energy, the ability to vary the spatial mesh with energy is an important capability for optimizing the coarse mesh for all energies. However, this flexibility comes with the memory and computational burden of storing and following the map. To resolve this, the library contains optimization flags that allows the user to switch off features, such

as the ability to have an energy dependent mesh, in order to improve the computational performance of the library.

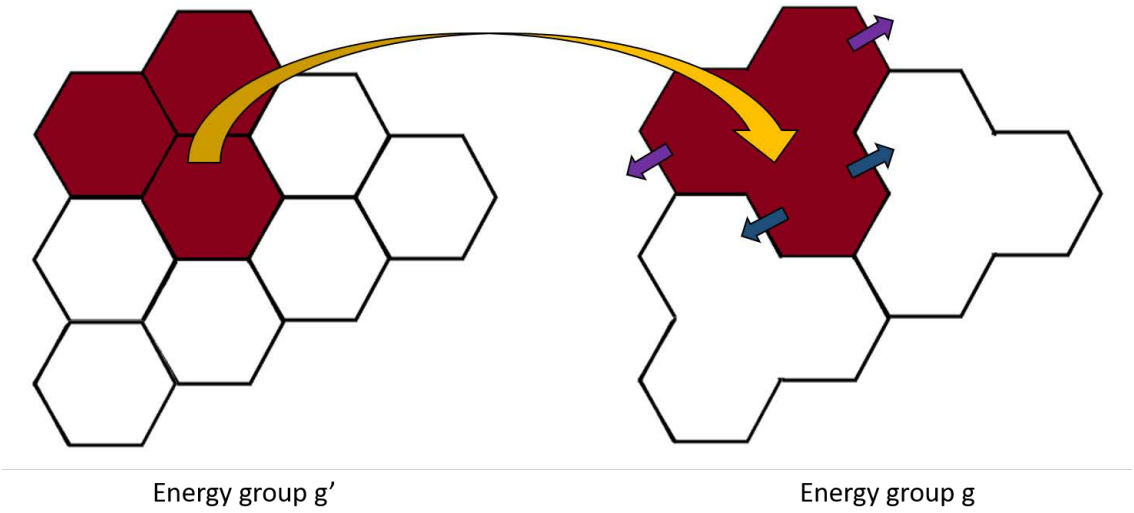

Figure 4.4 – A visualization of the mapping between adjacent cells of the coarse mesh. A yellow arrow is used to indicate the mapping between coarse energy groups. Blue arrows are used to indicate interfaces between coarse cells in space and purple arrows are used to define surfaces along geometric boundaries.

4.2.4 PETSc

Rather than developing matrix solvers within the library, we chose to use a library of matrix solving techniques. This was done to focus the development of the library on the interface and the acceleration methods. Moreover, by choosing to use a library of matrix solving techniques, we ensured that we would have access to a variety of high-quality methods for experimentation. PETSc was chosen as the library of matrix solving techniques that would be linked to our library. This decision was made due to the variety of techniques offered by PETSc, its simple and flexible implementation style, and its support of both the C++ and FORTRAN programming languages. Some of the other beneficial features of PETSc are its support for General Purpose programming on Graphics Processing Units (GPGPU), its built-in error checker and profiler, and its active community support and development.

Currently, the library only calls the PETSc library for solving linear systems of equations. Moreover, the operator, unknowns and fixed terms are built in the library, with out the assembly routines of PETSc, as shown by Fig. 4.5. The advantage of this approach is that eventually the acceleration library can call its own matrix inversion techniques or those of a different matrix solving library. However, to make better use of the capabilities of PETSc, a more integrated approach could be investigated in future. For instance, PETSc has the capability to solve nonlinear problems and therefore could be implemented to solve the accelerated transport problem. As well, PETSc can automatically redistribute and coarsen the vectors and matrices built within it, which would be interesting for balance loading purposes.

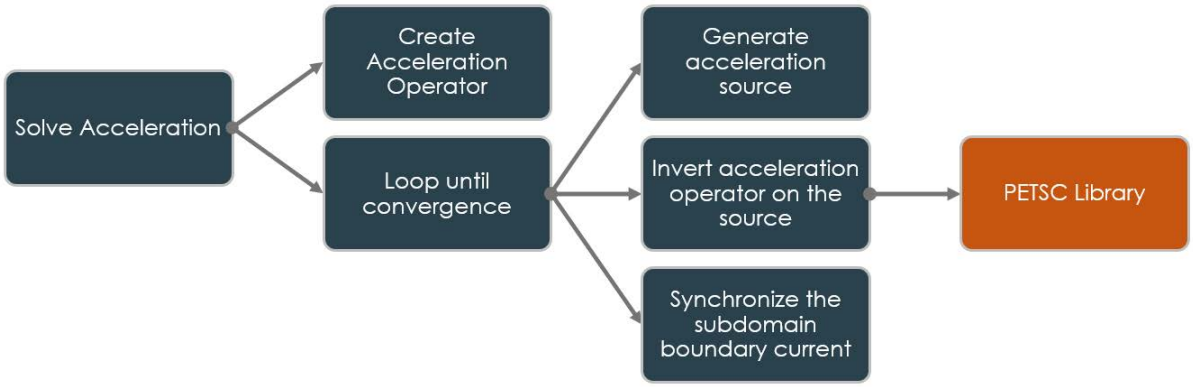

Figure 4.5 – A visualization of the interface between the acceleration library and PETSc.

4.2.5 Parallelism

Processors on the same node have access to the same RAM and thus communicate directly through the shared memory. In contrast, communications between processors on different nodes must be routed through much slower (lower bandwidth and higher latency) mediums, making them computationally more expensive. When solving a system of equations, one can try to reduce the number of inter-node communications by performing several intra-node communications per inter-node communication. This multilevel parallel communication approach has been implemented in the library using MPI protocol. The library allows the user to choose at which iteration level of the solving scheme, the inter-node or intra-node communications take place. For instance, the user may choose to apply the inter-node communication at the outer iteration level and the intra-node communication at the inner iteration level. This way the user would minimize the amount of time spent waiting for the inter-node communications at the cost of performing more inner iterations.

All MPI communications within the library are done asynchronously. Asynchronous communication means that instead of the communication occurring in a single step (synchronous communication), the call to communicate and barrier at which the communication must finish (synchronization point) are separated. Moreover, the MPI communication calls and barriers of the library are functionalized and made accessible to the user. This allows the user to place library routines and their own code in between the MPI communication call and barrier in order to mask the cost of the communication. GPGPU parallelism by means of CUDA or OpenCL are available for use within the matrix solving routines of PETSc. Thus, while the library itself only contains MPI commands, we are still able to take advantage of heterogeneous parallelism for the most computationally expensive routines of the acceleration library.

4.2.6 Solving schemes

Because of the flexible way that the library has been written, the type of acceleration method can be adjusted throughout the simulation. This allows different acceleration schemes to be called depending on the energy group, the subdomain composition, or even the iteration number, as we will later demonstrate in the results.

The library also allows for the acceleration energy and spatial mesh to be modified at runtime. This allows the user to solve a coarser version of the transport mesh for stability and computational performance benefits. As well, the user can choose to solve a subdomain of the original transport mesh, i.e. a select number of energy groups or spatial cells. This allows the user to focus the computational efforts on solving the parts of the domain that they expect to be the most difficult to converge parts of the domain.

Two different parallel solving schemes are available within the library, called the synchronous and asynchronous parallel schemes respectively. The synchronous scheme directly solves the global matrix for each coarse energy group in parallel using the PETSc library, as is shown by Algorythm 2. By directly solving the global problem in parallel, the number of iterations needed to converge the multigroup acceleration equation does not depend on the number processors. As well, in order to do such a parallel solve, the library has the restriction that the number of subdomains must match the number of processors. Moreover, in order to avoid unnecessary communication, the library also requires that the acceleration subdomain must encompass the same space as that of the transport operator. This restriction is illustrated by Fig. 4.6.

Algorithm 2: Synchronous multigroup parallel solving scheme.

Result: Solves for $\tilde{\boldsymbol{\Psi}}$ and k
$lgUnConv$=true
while $lgUnConv$ **do**
  for g=[1,G] **do**
    `CreateSource(`$\boldsymbol{Q}_{Acc,g}$,$\tilde{\boldsymbol{\Psi}}_g$`)`
    `ParallelSolve(`$\underline{\underline{\boldsymbol{A}}}$,$\tilde{\boldsymbol{\Psi}}_g$,$\boldsymbol{Q}_{Acc,g}$`)`
    `CheckConvergenceOfFissionSource(`$lgUnConv$,$\tilde{\boldsymbol{\Psi}}$`)`
  end
end

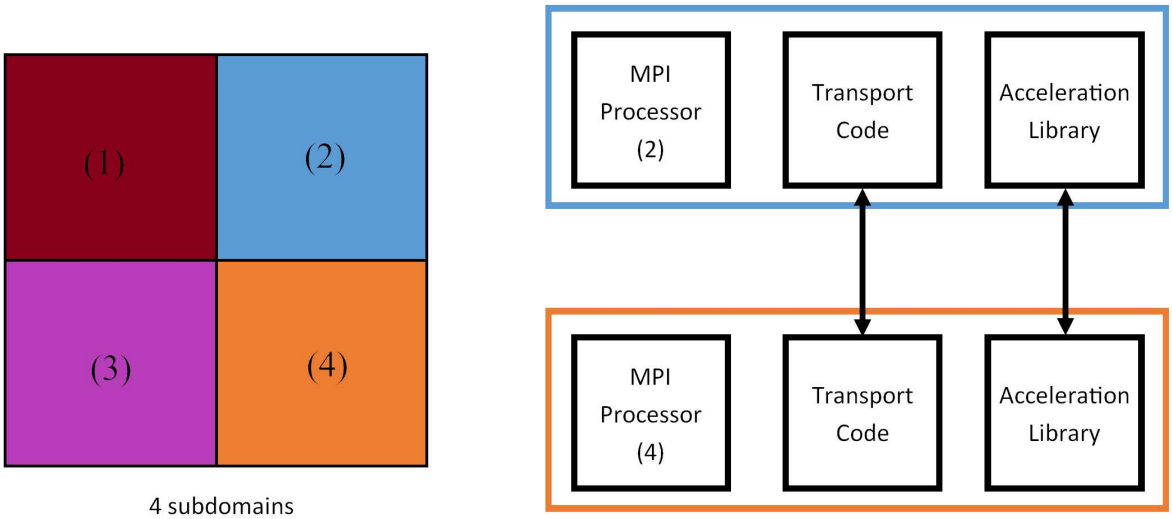

Figure 4.6 – A visualization of the required parallel implementation of the acceleration library in the user's code. Black arrows are used to indicate MPI communications.

The asynchronous parallel solving scheme seeks to coarsen the parallelism by only communicating the subdomain interface partial current once per multigroup iteration of the acceleration operator. As well, the user can place library routines and their own code in between the MPI communication calls of the solving routine in order to mask the cost of the communication. For example, when accelerating the outer iterations, by default the library solves each coarse group (g) separately. This allows the library to place the routines for solving $g' \neq g$ in between the MPI communication of group g, to hide the cost of the communication. The algorythm by which it accomplishes this is defined by Algorithm 3. The advantage of the asynchronous scheme compared to the synchronous scheme is that it typically requires less MPI communications in order to converge to a solution. This is important for improving the parallel efficiency of the library when cost of the MPI communication is significant compared to the cost of solving the acceleration operator. The disadvantage of the asynchronous scheme, compared to the synchronous scheme, is that the number of multigroup acceleration iterations are typically increased due to the spatial decomposition of the global problem.

Algorithm 3: Asynchronous multigroup parallel solving scheme.

Result: Solves for $\tilde{\boldsymbol{\Psi}}$ and k
$lgUnConv$=true
$lgUnConMPI$=true
while $lgUnConv$ **do**
  while $lgUnConMPI$ **do**
    for g=[1,G] **do**
      `EndMPICommunication(`$\tilde{\boldsymbol{\Psi}}_g$`)`
      `CreateSource(`$\boldsymbol{Q}_{Acc,g}$,$\tilde{\boldsymbol{\Psi}}_g$`)`
      `SequentialSolve(`$\underline{\underline{\boldsymbol{A}}}$,$\tilde{\boldsymbol{\Psi}}_g$,$\boldsymbol{Q}_{Acc,g}$`)`
      `BeginMPICommunication(`$\tilde{\boldsymbol{\Psi}}_g$`)`
    end
    `CheckConvergenceOfInterfaceFlux(`$lgUnConMPI$,$\tilde{\boldsymbol{\Psi}}$`)`
  end
  `CheckConvergenceOfFissionSource(`$lgUnConv$,$\tilde{\boldsymbol{\Psi}}$`)`
end

4.3 Library application

An overview of the iteration scheme for accelerating a transport solver, using the library, is shown by Fig. 4.7. In the following sections, the creation of the acceleration operator and the application of the prolongation operator will be discussed. The details of applying the transport operator and solving the acceleration operator have already been thoroughly discussed in Sections 2.11 and 4.2.6, respectively.

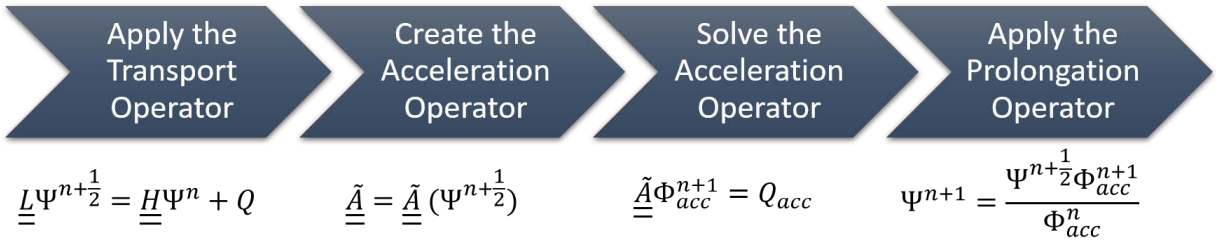

Figure 4.7 – A visualization of the iteration scheme for accelerating a transport solver using the library.

4.3.1 Acceleration operator generation

In order to create the acceleration operator, one must first create the coarse mesh by homogenizing the fine mesh data. A generic homogenization routine is shown by Algorithm 4. In the case of coarse partial current, **fineData** would be the fine partial current, **weight** would contain the surface area of the fine mesh, C would be composed of the coarse surface indices and F_c would contain the fine surface indices that lie on the coarse surface c.

Algorithm 4: A generic homogenization routine.

input : fineData contains the fine mesh data,
weight contains the weight function data,
C is the coarse index space,
F_c is the fine index space mapped to by the coarse index c.
output: coarseData contains the coarse mesh data.
Function `Homogenize(coarseData,fineData,weight,`C`,`F_c`):`

```
coarseData =0
for c ∈ C do
    sumWeight=0
    for f ∈ F_c do
        coarseData[c]=coarseData[c]+fineData[f]*weight[f]
        sumWeight=sumWeight+weight[f]
    end
    coarseData[c]=coarseData[c]/sumWeight
end
return
```

Next, the nonlinear transmission ($\tilde{T}_{i,s,s'}$), escape ($\tilde{E}_{i,s}$), incident ($\tilde{I}_{i,s'}$) and collision ($\tilde{C}_i$) coefficients defined by Eqs. (3.46a), (3.46b), (3.48a) and (3.48b) respectively, must be generated. The notation used by these algorythms is shown by Fig. 4.8. In addition, p is used to denote the spatial moment index. It is important to note that the spatial moments are defined by the orthonormal vectors $\boldsymbol{P}_{i,s}(\vec{r}_s)$. The library cumulates these nonlinear coefficients using Algorithms 5 to 8.

Finally, the acceleration operator is formed by substituting the homogenized data and the

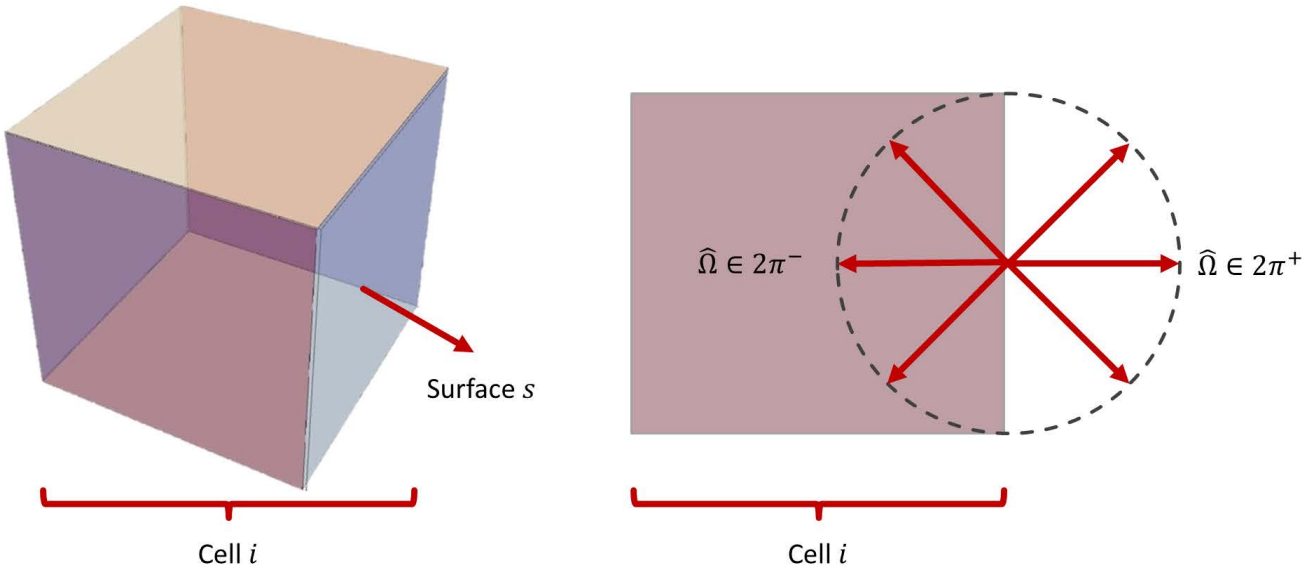

Figure 4.8 – An illustration of the notation used by the transport operator.

Algorithm 5: RMA: transmission coefficient generation.

input : $w[m,i,s]=w_m \left|\hat{\Omega}_m \cdot \hat{n}_{i,s}\right|$,
$T[m,i,p',s,s']=\frac{1}{A_{i,s}} \int_{\Gamma_{i,s}} d^2 r_s \boldsymbol{P}_s(\vec{r}_s) \cdot \boldsymbol{T}_{m,i,p',s,s'}$.

parameter: where p' is the spatial moment index,
$\boldsymbol{P}_{i,s}(\vec{r}_s)$ is an orthonormal vector of the spatial basis functions on the surface,
$\boldsymbol{T}_{m,i,p',s,s'}$ is a vector containing the contribution of the transmission of spatial moment p' on surface s', to all the spatial moments on surface s.

output : $\tilde{T}[i,s,s']$ contains the nonlinear transmission coefficients.

```
for i ∈ I do
    for s ∈ S_i do
        for s' ∈ S_{i,s} do
            for m ∈ M_s^+ ∩ M_{s'}^- do
                for p' ∈ P do
                    T̃[i,s,s']=w[m,i,s]*T[m,i,p',s,s']*ψ^-[m,i,p',s']/J^-[i,s]
                end
            end
        end
    end
end
```

Algorithm 6: RMA: escape coefficient generation.

input : $w[m,i,s]=w_m \left|\hat{\Omega}_m \cdot \hat{n}_{i,s}\right|$,
$E[m,i,\alpha,p',s]=\frac{1}{A_{i,s}} \int_{\Gamma_{i,s}} d^2 r_s \boldsymbol{P}_s(\vec{r}_s) \cdot \boldsymbol{E}_{m,i,p',\alpha,s}$.

parameter: where p' is the spatial moment index, α is the subregion index
$\boldsymbol{P}_{i,s}(\vec{r}_s)$ is an orthonormal vector of the spatial basis functions on the surface,
$\boldsymbol{E}_{m,i,p',\alpha,s}$ is a vector containing the contribution of the escape of spatial moment p' from subregion α, to all the spatial moments on surface s.

output : $\tilde{E}$ contains the nonlinear escape coefficients.

```
for i ∈ I do
    for s ∈ S_i do
        sumQ=0
        for α ∈ i do
            for m ∈ M_s^+ ∩ M_s'^- do
                for p' ∈ P do
                    Ẽ[i,s]=Ẽ[i,s]+w[m,i,s]*V[i,α]*E[m,i,α,p',s]*Q[m,i,α,p']
                end
            end
            sumQ=sumQ+V[i,α]*Q[0,0,i,α]
        end
        Ẽ[i,s]=Ẽ[i,s]/sumQ
    end
end
```

Algorithm 7: RMA: incident coefficient generation.

input : $I[m,i,\alpha,p',s']=\int_{D_{i,\alpha}} d^3r \boldsymbol{P}(\vec{r}) \cdot \boldsymbol{I}_{m,i,\alpha,p',s'}$.

parameter: where p' is the spatial moment index, α is the subregion index

$\boldsymbol{P}_i(\vec{r}_i)$ is an orthonormal vector of the spatial basis functions over the volume,

$\boldsymbol{I}_{m,i,\alpha,p',s'}$ is a vector containing the contribution of the incident flux of spatial moment p' from surface s', to all the spatial moments of subregion α.

output : $\tilde{I}$ contains the nonlinear incident coefficients.

```
for i ∈ I do
    sumV=0
    for α ∈ i do
        for s' ∈ S_i do
            for m ∈ M⁻_{s'} do
                for p' ∈ P do
                    Ĩ[i,s'] =w[m]*I[m,i,α,p',s']hm,i,p',s'/J⁻[i,s]
                end
            end
        end
        sumV=sumV+V[i,α]
    end
    Ĩ[i,s']=Ĩ[i,s']/sumV
end
```

Algorithm 8: RMA: collision coefficient generation.

input : $C[m,i,\alpha,p']=\int_{D_{i,\alpha}} d^3r \boldsymbol{P}(\vec{r}) \cdot \boldsymbol{C}_{m,i,\alpha,p'}$

where, p' is the spatial moment index, α is the subregion index.

parameter: where p' is the spatial moment index, α is the subregion index

$\boldsymbol{P}_i(\vec{r}_i)$ is an orthonormal vector of the spatial basis functions over the volume,

$\boldsymbol{C}_{m,i,\alpha,p'}$ is a vector containing the contribution of the collision of spatial moment p' from subregion α, to all the spatial moments of subregion α.

output : $\tilde{C}$ contains the nonlinear collision coefficients.

```
for i ∈ I do
    sumQ=0
    for α ∈ i do
        for m ∈ M do
            for p' ∈ P do
                C̃[i]=w[m]*C[m,i,α,p']*Q[m,i,α,p']
            end
        end
        sumQ=sumQ+V[i,α]*Q[0,0,i,α]
    end
    C̃[i]=C̃[i]/sumQ
end
```

nonlinear coefficients into the acceleration method's equations. As an example, the RMA operator for 1D geometries is defined in Appendix B.4.

4.3.2 Prolongation

After solving the acceleration equation, the fine mesh data must be updated. CMFD and RMA update the fine scalar flux and the fine angular flux using rebalancing relations. The rebalancing of the fine scalar flux and the fine angular flux, using arbitrary weight functions, is defined by Algorithms 9 and 10, respectively. When RMA is used, the weight function for the scalar flux is the coarse scalar flux and the weight function for the angular flux is the coarse outgoing partial current. When CMFD is used, the coarse scalar flux is used as the weight function for the fine scalar flux and the fine angular flux.

Algorithm 9: Generic scalar flux prolongation routine.

input : ϕ is the scalar flux of the fine mesh,
$\mathsf{weight}^{(n+1)}$ is the $(n+1)$ iteration of coarse mesh weight,
C is the coarse index space,
F_c is the fine index space mapped to by the coarse index c.

output: ϕ is the scalar flux of the fine mesh.

Function `FluxProlongation(`ϕ,$\mathsf{weight}^{(n+1)}$,$\mathsf{weight}^{(n+1/2)}$,$C$,$F_c$`):`

```
for c ∈ C do
    for f ∈ F_c do
        φ[f]=φ[f]*weight^(n+1)[c]/weight^(n+1/2)[c]
    end
end
return
```

Algorithm 10: Generic angular flux prolongation routine.

input : ψ is the scalar flux of the fine mesh,
$\mathsf{weight}^{(n+1)}$ is the $(n+1)$ iteration of coarse mesh weight,
C is the coarse index space,
F_c is the fine index space mapped to by the coarse index c.

output: ψ is the angular flux of the fine mesh.

Function `AngularFluxProlongation(`ψ,$\mathsf{weight}^{(n+1)}$,$\mathsf{weight}^{(n+1/2)}$,$C$,$F_c$`):`

```
for c ∈ C do
    for f ∈ F_c do
        ψ[f]=ψ[f]*weight^(n+1)[c]/weight^(n+1/2)[c]
    end
end
return
```

4.4 Parallelism within IDT

In order to spatially decompose a geometry for parallel simulation, IDT splits up the geometry along the Cartesian axes. An example of how IDT splits up a geometry into 3 parts along the x, y and z axis is shown by Fig. 4.9. For the remainder of the manuscript we will use the notation (# of divisions along the x axis)-(# of divisions along the y axis)-(# of divisions along the z axis), to specify the spatial decomposition of the geometry.

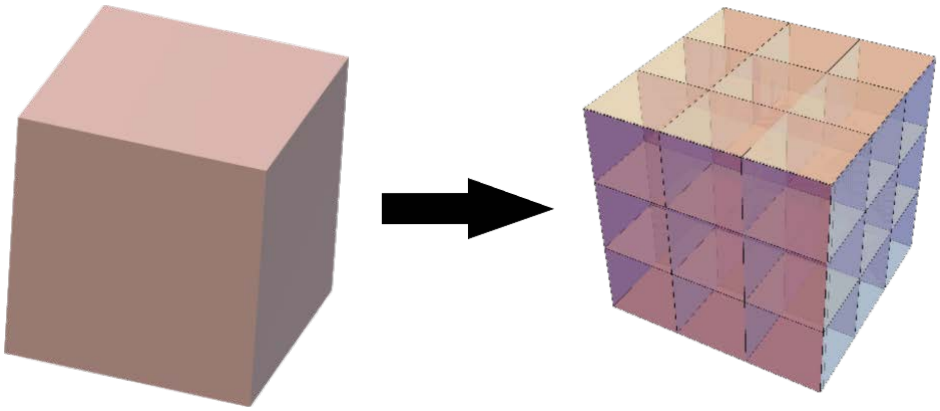

Figure 4.9 – A visualization of the spatial decomposition strategy employed by IDT.

IDT only synchronizes the subdomain boundary angular flux once per outer iteration, after the k and the fission source have been updated. Fig. 4.10 illustrates of the parallel multigroup procedure in IDT. In order to be consistent, the library also only synchronizes the subdomain boundary flux when it is accelerating the outer iterations. During the inner iterations, each subdomain is converged independently without any need for parallel communication. This solving scheme allows IDT to increase the amount of computation per MPI communication. As a result of this scheme, the continuity of the flux at subdomain boundaries is lost during the inner iterations. To resolve this, IDT requires that the subdomain boundary flux is converged within a given limit.

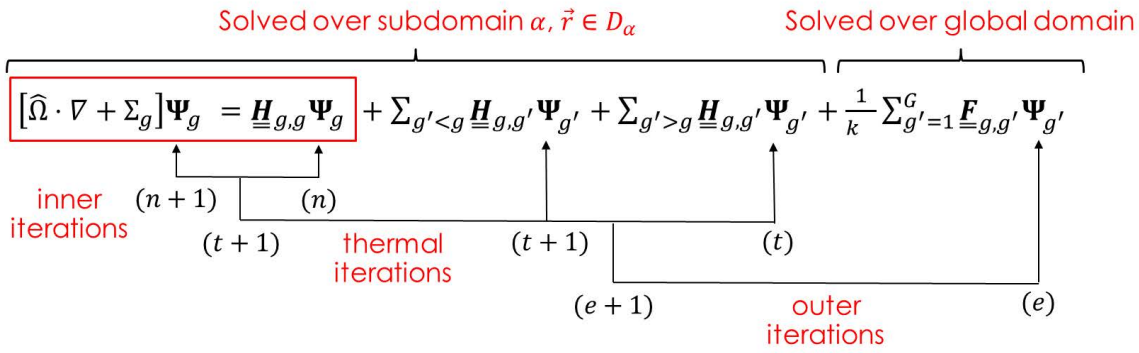

Figure 4.10 – An illustration of the parallel multigroup iteration scheme implemented in IDT.

Chapter 5

Numerical benchmark results

In this section the numerical performance of RMA and the MCNH variant of CMFD as well as the performance of the library will be assessed. In order to avoid instability and variance in the performance of the matrix solver, only the direct solver in PETSc was used to generate the benchmark results. Throughout this section NC (Not Converged) will be used to denote when the simulation failed to reach convergence. For the sake of consistency, all cases were run on the Broadwell partition of the Cobalt computing cluster of the CCRT, [1]. Each node of the Broadwell partition has 128 GB memory and is made up of 28 processors clocked at a frequency of 2.4 GHz. Moreover, the Intel parallel studio XE, compilers and MKL library were used to compile the code.

The remainder of this section is broken up into 2 subsections corresponding to the C5 MOX fuel assembly with a 7-group cross section library (C5G7), and Zero Power Physics Reactor (ZPPR) benchmarks. Starting with the C5G7 benchmark, a comparison of the performance of different solving techniques and preconditioners available within PETSc is done. This is followed by a performance comparison of RMA and MCNH when the number of discrete angles is varied. Next, the strong parallel scaling of the library is tested for the MCNH and RMA methods. Finally, for the ZPPR benchmark, the performance of RMA is tested when the degree of the Legendre polynomial representation of the scattering source is varied.

5.1 C5G7

In the following subsection, the performance results of simulating the 3D rodded-B configuration of the C5G7 MOX benchmark, [36], are presented. This benchmark was chosen in 2001 to test the ability of deterministic transport codes to model heterogeneous, whole core geometries without spatial homogenization. The geometry of the C5G7 benchmark is a miniature light water reactor with sixteen fuel assemblies, surrounded by a water reflector. In particular, the fuel assemblies are made up of eight uranium oxide assemblies and eight mixed oxide assemblies. The geometry has quarter-core radial symmetry as well as reflective symmetry across the x-y plane. Thus, only 1/8th of the modeled reactor is actually meshed and solved by IDT. A visualization of the C5G7 benchmark is shown in Fig. 5.1. The shaded sections of the bottom diagram of Fig. 5.1 depicts the depth of the control rods into the geometry along the A-A and B-B planes. The precise location of the control rods is shown by Fig. 5.2. A more detailed visualization of the fuel assemblies in the C5G7 benchmark is shown by Fig. 5.3. The convergence criteria used to solve

the C5G7 benchmark is listed in Table 5.1. The convergence tolerance of Table 5.1 was selected such that it satisfied the requirements of the C5G7 benchmark, 1.0×10^{-4} for the $k - effective$ and the $L\infty$ of the self scattering source.

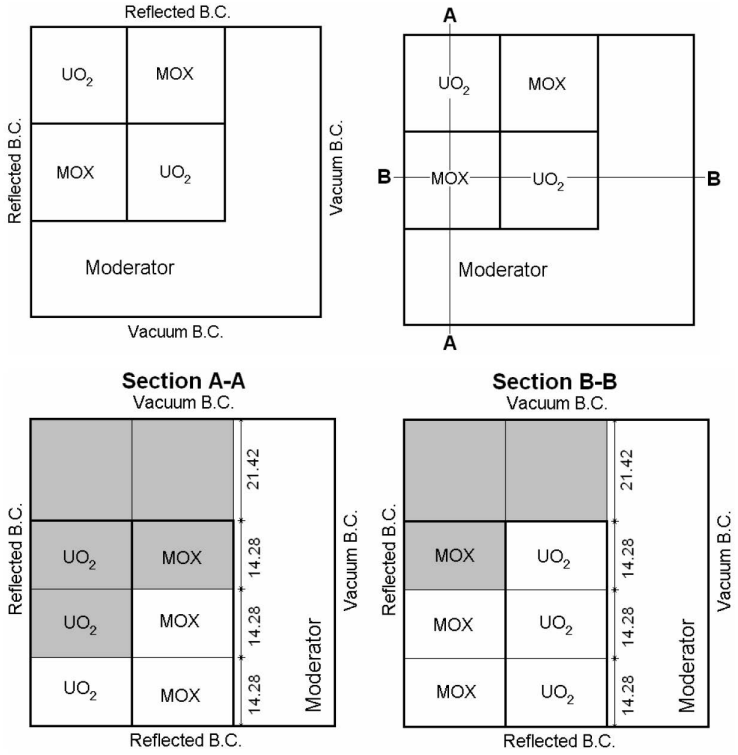

Figure 5.1 – A visualization of the C5G7 benchmark, provided by [36]

Convergence Criteria	k-effective	Fission Source	Subdomain Interface Flux	Self-Scattering Source
Precision	5.0×10^{-5}	5.0×10^{-4}	1.0×10^{-3}	3.0×10^{-5}

Table 5.1 – The convergence criteria used to solve the C5G7 benchmark.

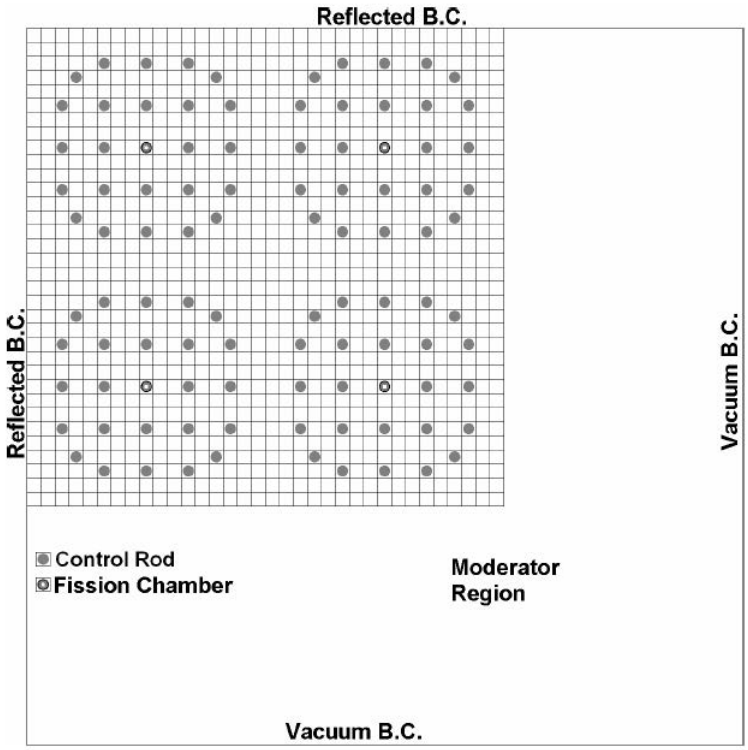

Figure 5.2 – A visualization of the control rod and fission chamber locations in the C5G7 benchmark, provided by [36]

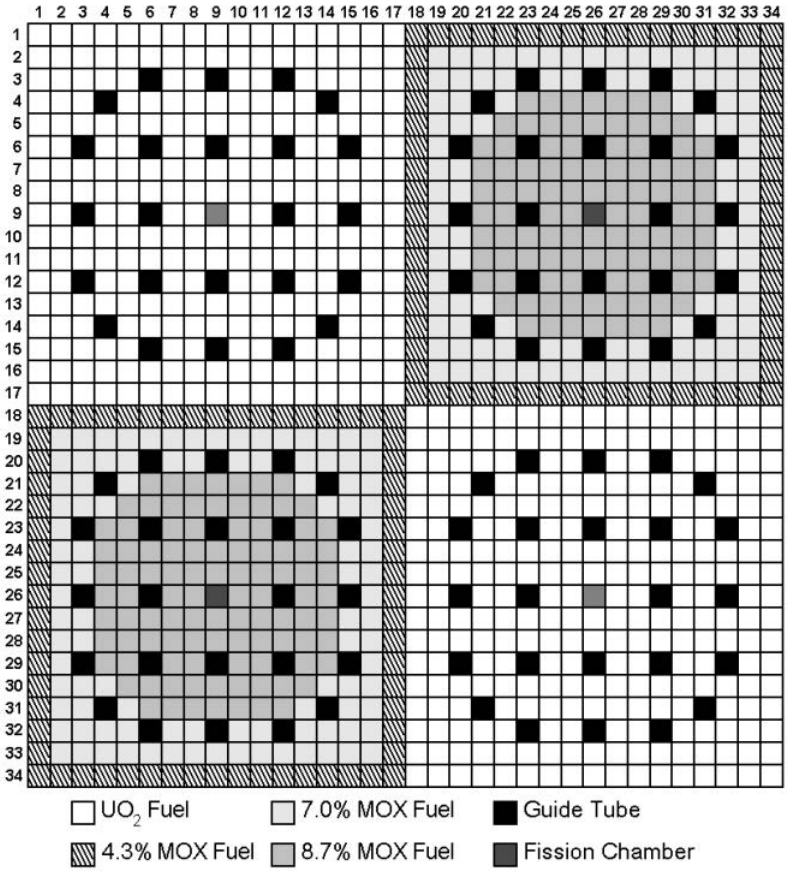

Figure 5.3 – A visualization of the fuel assemblies of the C5G7 benchmark, provided by [36].

5.1.1 Matrix solving techniques comparison

In order to find a suitable matrix solving technique and preconditioner, some empirical measurements were necessary. The C5G7 geometry was decomposed into 27 subdomains by dividing the geometry into 3 parts along the x, y and z axes (3-3-3), using the methodology explained in Section 4.4. This was done so that the number of cores would stay within the 28 core per node limit of the CCRT, ensuring that inter-node communication would not affect the results of the benchmark.

We begin the benchmark by comparing the Richardson method, Generalized Minimal Residual Method (GMRES), Stabilized BiConjugate Gradient method (BCGS) and Improved Stabilized BiConjugate Gradient method (IBCGS) when no preconditioner is applied. The results of this comparison are shown for the MCNH variant of CMFD, E-RMA and B-RMA in tables 5.2, 5.3, 5.4, respectively. BCGS requires the least amount of time and the least number of iterations to solve the MCNH and RMA operators. The next most performant technique is IBCGS, however in the case of MCNH this technique takes almost twice the time and nine times the number of iterations to solve the operator on average. GMRES is the most computationally expensive operator and in the case of E-RMA and B-RMA it fails to converge within the time limit. GMRES requires the matrix to be positive definite in order to guarantee convergence, thus the very slow convergence of the acceleration operators using GMRES is likely the result of the nonlinear operators not being positive definite throughout the simulation. The Richardson scheme fails to converge within the desired precision for both MCNH and B-RMA. In order to converge, the Richardson scheme requires the matrix to contain only positive eigenvalues. This property is not guaranteed by MCNH or RMA and in fact was found to not be true even for the simple cases analyzed during the spectral radius analysis.

Matrix Solver Type	Avg. Time to Solve Operator (ms)	Avg. # of Iter. to Solve Operator
RICHARDSON	NC	NC
GMRES	9220	1481
BCGS	7.98	144
IBCGS	15.6	1190

Table 5.2 – A comparison of various matrix solvers for solving the MCNH operator

Matrix Solver Type	Avg. Time to Solve Operator (ms)	Avg. # of Iter. to Solve Operator
RICHARDSON	99.2	367.7
GMRES	NC	NC
BCGS	53.5	65.6
IBCGS	60.9	65.7

Table 5.3 – A comparison of various matrix solvers for solving the E-RMA operator

Matrix Solver Type	Avg. Time to Solve Operator (ms)	Avg. # of Iter. to Solve Operator
RICHARDSON	NC	NC
GMRES	NC	NC
BCGS	81.8	67.2
IBCGS	88.9	67.2

Table 5.4 – A comparison of various matrix solvers for solving the B-RMA operator

Fixing the solving scheme to be BCGS, we compare the performance when no preconditioner is used to when the Jacobi, block Jacobi, additive Schwartz and Geometric Algebraic MultiGrid (GAMG) preconditioners are used. The results of this comparison are shown for MCNH, E-RMA and B-RMA in tables 5.5, 5.6 and 5.7, respectively. The block Jacobi preconditioner is the most effective at reducing the average solve time of the MCNH, E-RMA, and B-RMA operators. Moreover, by comparing the average operator construction time, the cost of creating the block Jacobi preconditioner can be seen to be negligible. E-RMA and B-RMA exceed the time constraints when the GAMG preconditioner is applied. However, GAMG is the most effective preconditioner for reducing the number of iterations to solve the MCNH operator. Thus, with future numerical optimization of the GAMG preconditioner, it may become the superior choice for MCNH and RMA.

Preconditioner Type	Time to Construct Operator (ms)	Time to Solve Operator (ms)	Avg. # of Iter. to Solve Operator
None	5.44	35.4	144
Jacobi	5.93	34.7	138
Block Jacobi	5.51	25.9	58.8
Additive Schwartz	23.9	30.8	43.9
Geometric Algebraic Multigrid	350	141	19.05

Table 5.5 – A comparison of various preconditioners for solving the MCNH operator

Preconditioner Type	Time to Construct Operator (ms)	Time to Solve Operator (ms)	Avg. # of Iter. to Solve Operator
None	4.26	53.5	65.6
Jacobi	5.12	55.8	65.6
Block Jacobi	4.0	43.53	27.8
Additive Schwartz	20.1	54.1	22.6
Geometric Algebraic Multigrid	NC	NC	NC

Table 5.6 – A comparison of various preconditioners for solving the E-RMA operator

Preconditioner Type	Time to Construct Operator (ms)	Time to Solve Operator (ms)	Avg. # of Iter. to Solve Operator
None	4.65	81.8	67.2
Jacobi	5.96	80.0	64.3
Block Jacobi	4.40	70.2	26.8
Additive Schwartz	0	0	0
Geometric Algebraic Multigrid	NC	NC	NC

Table 5.7 – A comparison of various preconditioners for solving the B-RMA operator

5.1.2 Performance scaling with discrete ordinates

In this section, the effectiveness of the MCNH variant of CMFD, E-RMA, B-RMA, and the hybrid method are analyzed with an increasing number of directions. For all cases, the Chebyshev Legendre quadrature was used to set the discrete directions, [38]. No acceleration was applied to the outer iterations of the benchmark. This choice was made to isolate the effect the S_N quadrature has on the convergence of the accelerated inner iterations. Moreover, this test provided insight into the effectiveness and numerical performance of the acceleration method when they are applied to the inner iterations.

The step and linear characteristics are investigated for two types of convergence schemes. The first scheme dictates that the self-scattering sources (inner iterations) are only fully converged once the fission sources (outer iterations) have been converged. We will refer to this scheme as the standard convergence scheme. The second scheme requires that the inner iterations are fully converged for each outer iteration. We will refer to this scheme as the fully converged scheme. Since the acceleration methods are nonlinear, their convergence behaviour depends on the current iteration of the transport solution. Thus, the second scheme is able to more rigorously asses the stability of the acceleration methods by forcing the acceleration to converge the inner iterations at multiple starting points in the solution space. For each of the four combination of cases, the number of inner iterations, execution time, as well as the iteration and time speedup, relative to transport, are shown by Figs. 5.4 to 5.10.

Starting with Fig. 5.4, the total number of inner iterations necessary to converge the benchmark is nearly independent of the number of directions for the hybrid method and both RMA variants. Indicating that the effectiveness of RMA is not greatly affected by the number of angular directions. The only anomaly occurs when the number of directions is set to 32 (S_4 angular quadrature) for the linear MOSC operator. For this case, the number of iterations of the hybrid method and RMA spikes up. This spike in the number iterations occurs as a result of the k of the S_4 simulation being discrepant compared to the other angular quadratures. This discrepancy is likely the result of the raytracing phenomenon due to the low number of directions being employed by the S_4 angular quadrature. For all cases shown in Fig. 5.4, E-RMA and B-RMA require almost exactly the same number of inner iterations to converge. This result is consistent with the spectral radius analysis presented in Section 3.2.13. With regards to MCNH, the number of inner iterations decays towards a limit for the step MOSC operator. As well, MCNH is unstable when either the linear MOSC operator is used or when a S_4 angular quadrature

is used with the step MOSC operator. This instability occurs when MCNH is applied to inner iterations of the 7th energy group, when the geometry is most optically diffusive. This instability with increasing optical diffusivity is consistent with the spectral radius results discussed in Section 2.20.

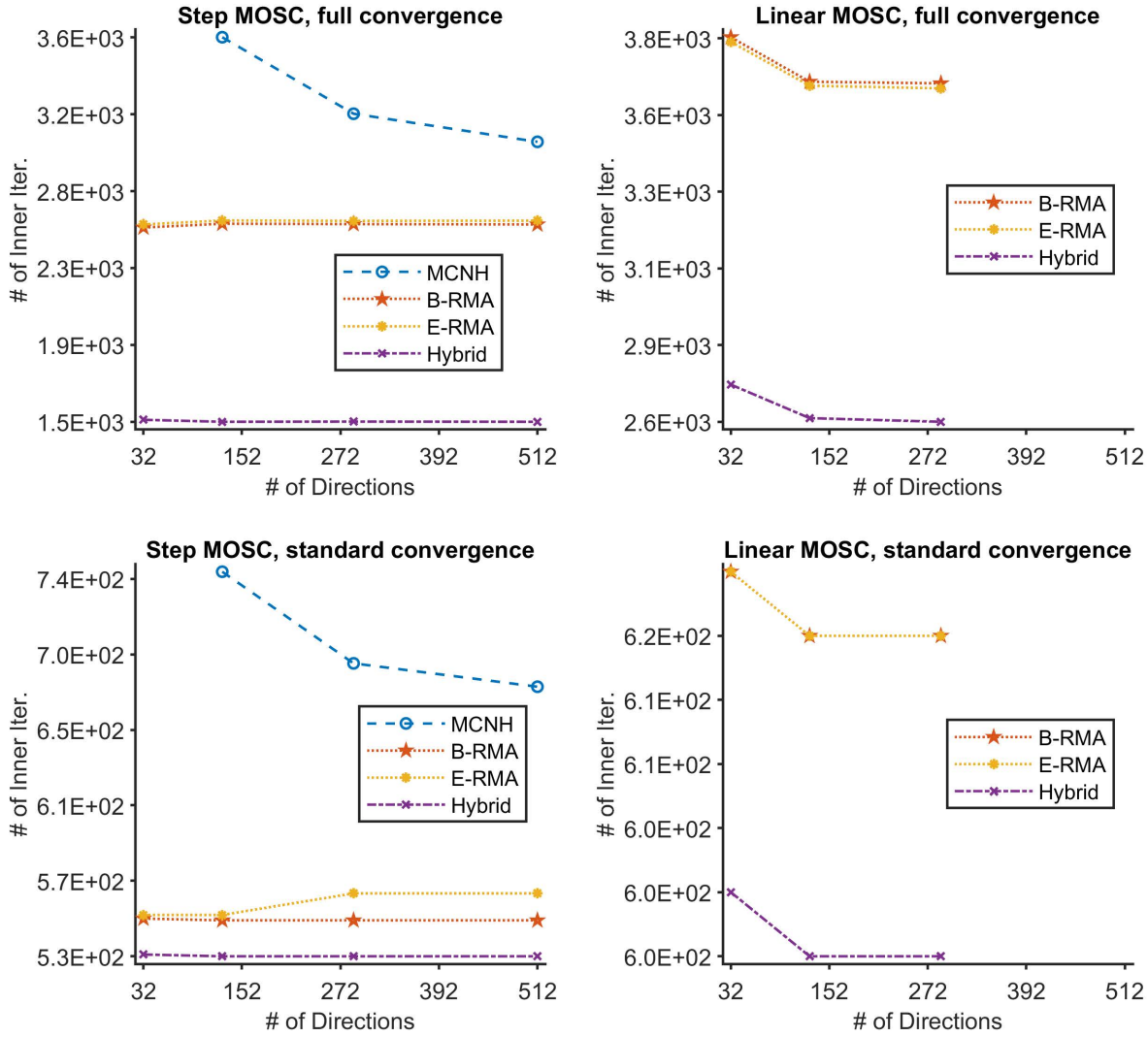

Figure 5.4 – Number of inner iterations versus the number of angular directions, when the inner iterations are accelerated by MCNH, E-RMA, B-RMA and the hybrid method (B-RMA&MCNH).

A comparison of the iteration speed up relative to SI is shown by Fig. 5.5. When MCNH is stable, it is less effective than RMA which is in turn less effective than the hybrid method. However, the relative gap between the effectiveness of RMA and the hybrid method is diminished when the standard convergence scheme is employed. Experimentally, this has been found to be due to the effectiveness of RMA increasing as the solution moves closer to convergence. This convergence behavior comes from RMA's assumption that the angular distribution of the flux is close to convergence. Thus, the closer the solution actually is to convergence, the more negligible this assumption becomes, and the

more effective RMA becomes. The standard convergence scheme only requires RMA to converge the inner iterations once the outer iterations are converged. Thus, the starting solution is closer to convergence, when RMA is required to converge the inner iterations for the standard convergence scheme, than when the fully converged scheme is used. This explains our previous observation that the relative gap between the effectiveness of RMA and the hybrid method is diminished when the standard convergence scheme is employed. As noted previously, we have chosen to accelerate only the inner iterations during this angular scaling comparison. However, it is important to note that much greater iteration speedups can be achieved when both the outer and the inner iterations are accelerated, as is shown by Fig. 5.6. A maximum inner iteration speedup of 81 is achieved for step MOSC, when the hybrid method is used for the outer and inner iterations and a maximum speedup of 70 is achieved for linear MOSC, when MCNH is used for the outer iterations and the hybrid method is used for the inner iterations. This represents a 5 fold increase in the maximum inner iteration speedup.

When linear MOSC is used, the effectiveness of E-RMA and B-RMA is reduced relative to when step MOSC is used. This result can be explained by the fact that RMA fixes the distribution of the spatial moments of the flux. When step MOSC is used, no approximation is made since only the zeroth spatial moment exists. However, this is not true when linear MOSC is used. Thus, it is logical to assume that the approximation of having a fixed distribution of spatial moments has a negative impact on the effectiveness of the acceleration method. However, one of the advantages of linear MOC is that it can use a coarser mesh than step MOC for similar accuracy. During this comparison the same mesh was used for step and linear MOC, however if a coarser mesh had been used for linear MOC then we would expect the effectiveness of RMA to improve for the linear MOC operator. Finally, it is important to note that the speedup values when the fully converged criteria is applied, are much lower than those achieved using the standard convergence criteria. The fully converged inner iteration requirement causes the accelerated solver to wastefully converge many inner iterations before the source from external groups has been converged.

With regards to Fig. 5.7, the total execution time can be observed to linearly depend on the number of directions. The hybrid method consistently finishes faster than all other methods. This result is expected since as was shown by Fig. 5.4, the hybrid method requires less iterations to solve the benchmark than any other method. When MCNH is stable, it consistently requires less time to converge than either RMA variant, despite requiring more inner iterations to converge. Moreover, B-RMA consistently converges faster than E-RMA, despite having the same iteration speed up and a denser matrix to solve. These results are due to the substantial computational cost associated with generating the RMA coefficients, as is illustrated by Fig. 5.8. Part of this computational cost will likely be eliminated by future optimization of the RMA algorithms. However, the coefficients of RMA are inherently more costly to compute than those of MCNH. Moreover, there is a large computational overhead for solving the RMA operator relative to MCNH, as shown by Fig. 5.9. As expected, the relative cost of solving the acceleration methods tends to decrease as the number of directions is increased.

A comparison of the execution time speed up relative to SI is shown by Fig. 5.10. The speedup of both variants of RMA is consistently larger for the linear MOSC operator than for the step MOSC operator, despite the contrary being true for the iteration speed

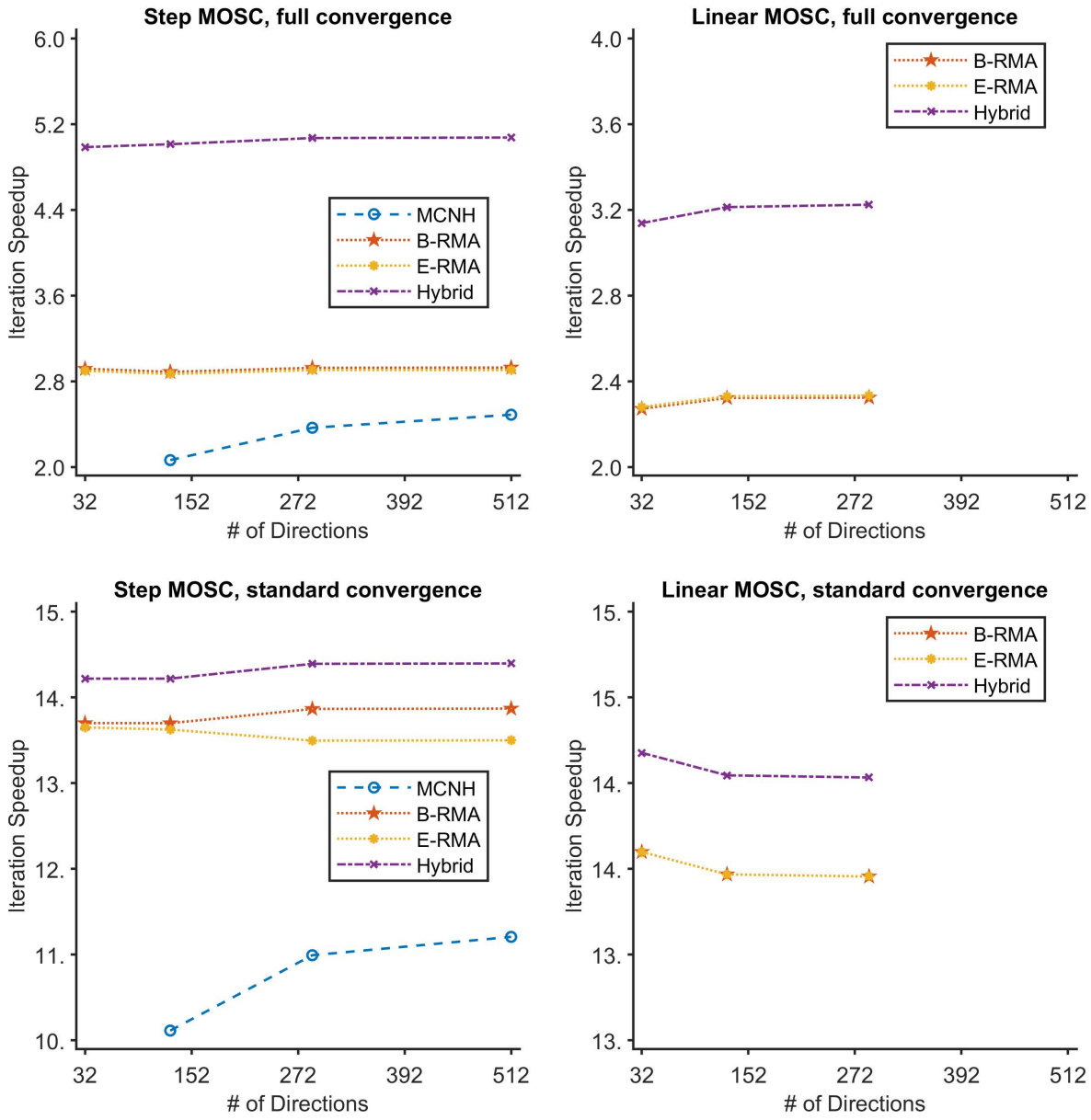

Figure 5.5 – Inner iteration speedup versus the number of angular directions, when the inner iterations are accelerated by MCNH, E-RMA, B-RMA and the hybrid method (B-RMA&MCNH).

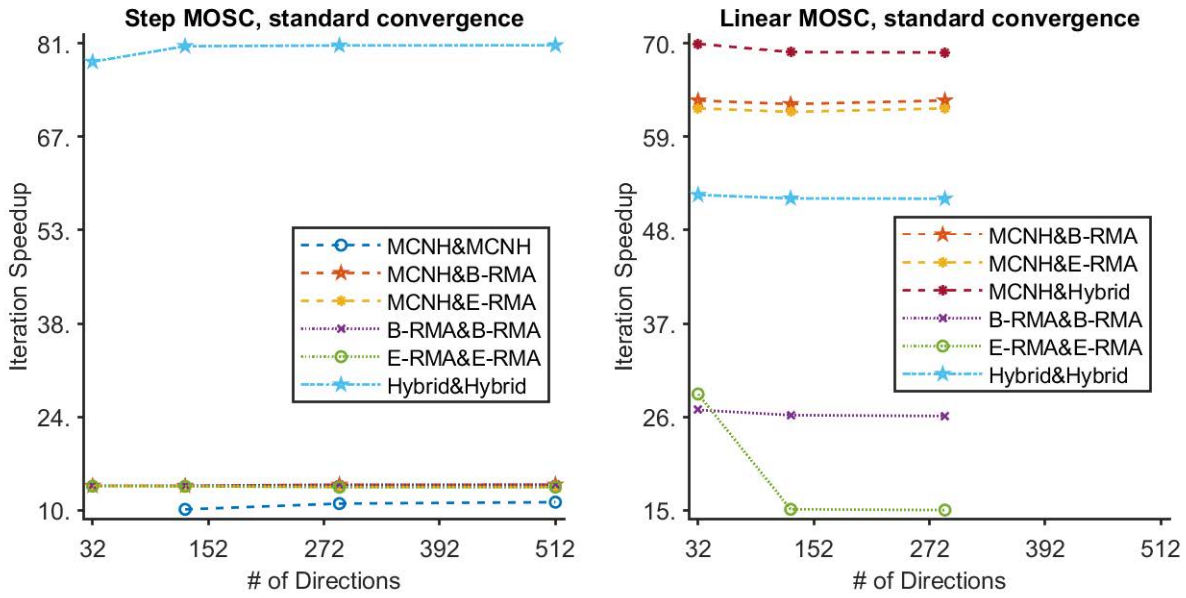

Figure 5.6 – Inner iteration speedup versus the number of angular directions, when the outer and the inner iterations are accelerated by MCNH, E-RMA, B-RMA and the hybrid method (B-RMA&MCNH).

up. This result is due to the linear MOSC operator being substantially more expensive to compute than the step MOSC operator. Thus, the cost of RMA relative to the transport operator is greatly diminished for the linear MOSC operator, causing the time speed up to approach the iteration speed up. When the fully converged criteria is applied, the time speed up of RMA can be less than 1 when the number of directions drops below a threshold. In these cases, the computational cost of applying RMA, relative to the transport operator, is too high. The hybrid method consistently offers the best speedup in time. Again, it is important to note that significantly improved speedups in time can be achieved when both the outer and the inner iterations are accelerated, as is shown by Fig. 5.11. The maximum speedup in time when the outer and the inner iterations are accelerated is 4 times that when only the inner iterations are accelerated. Due to the additional cost of computing the outer acceleration, the ratio between when the outer and inner iterations are accelerated and when only the inner iterations are accelerated is diminished when the speedup in time is considered compared to the speedup in inner iterations.

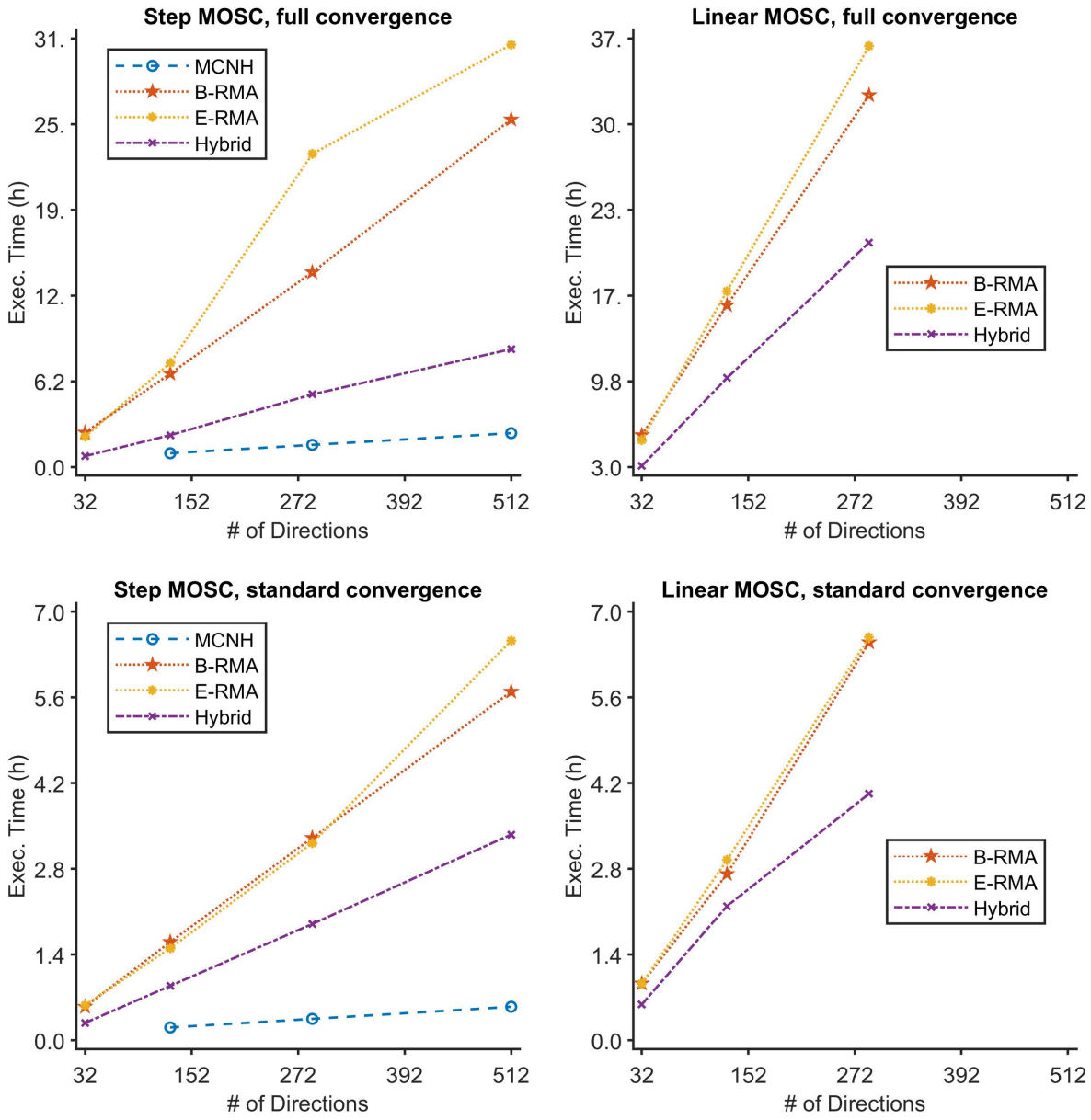

Figure 5.7 – Execution time versus the number of angular directions, when the inner iterations are accelerated by MCNH, E-RMA, B-RMA and the hybrid method (B-RMA&MCNH).

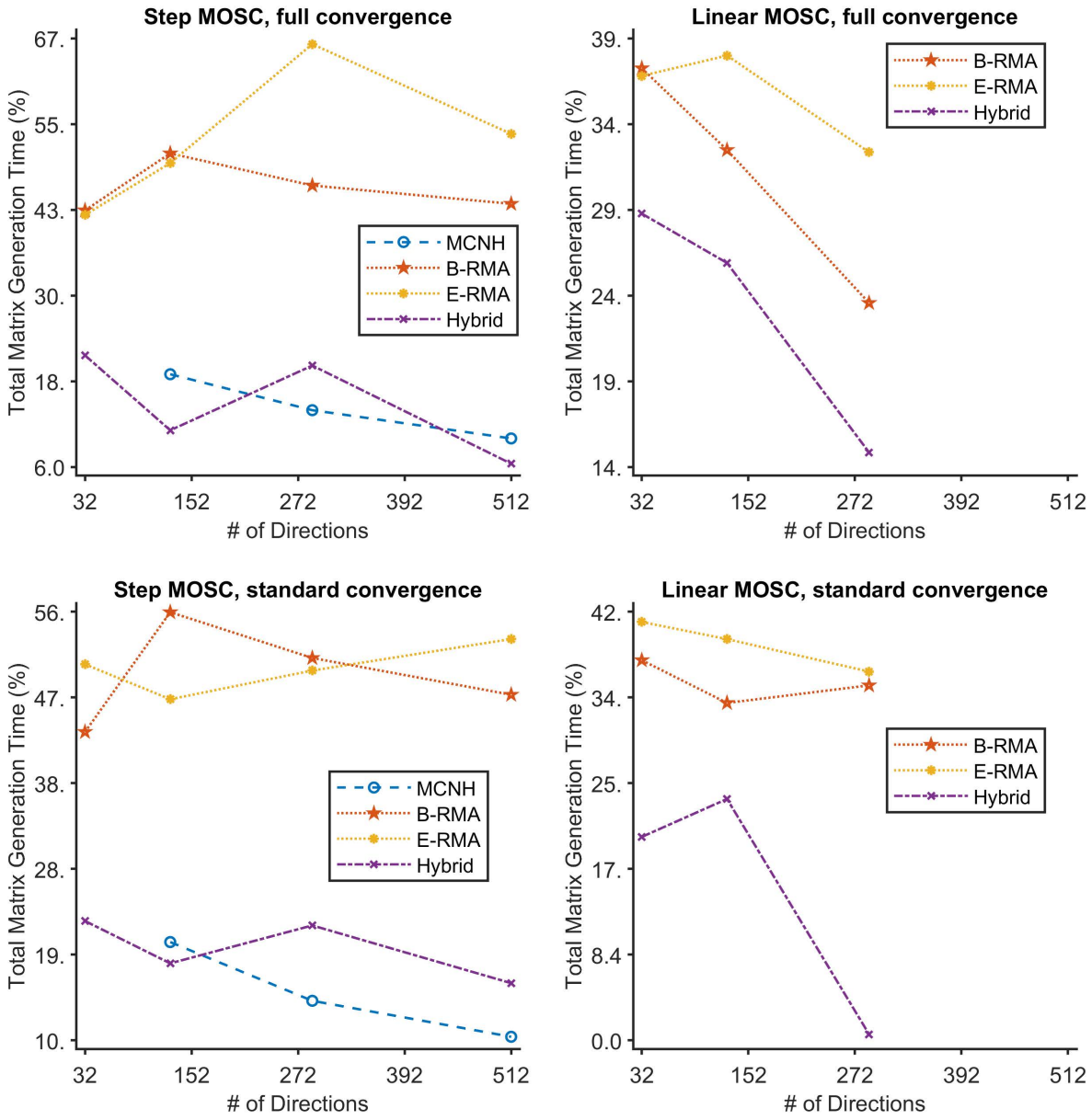

Figure 5.8 – Time to generate the acceleration matrix, as a percentage of the total execution time, versus the number of angular directions, when the inner iterations are accelerated by MCNH, E-RMA, B-RMA and the hybrid method (B-RMA&MCNH).

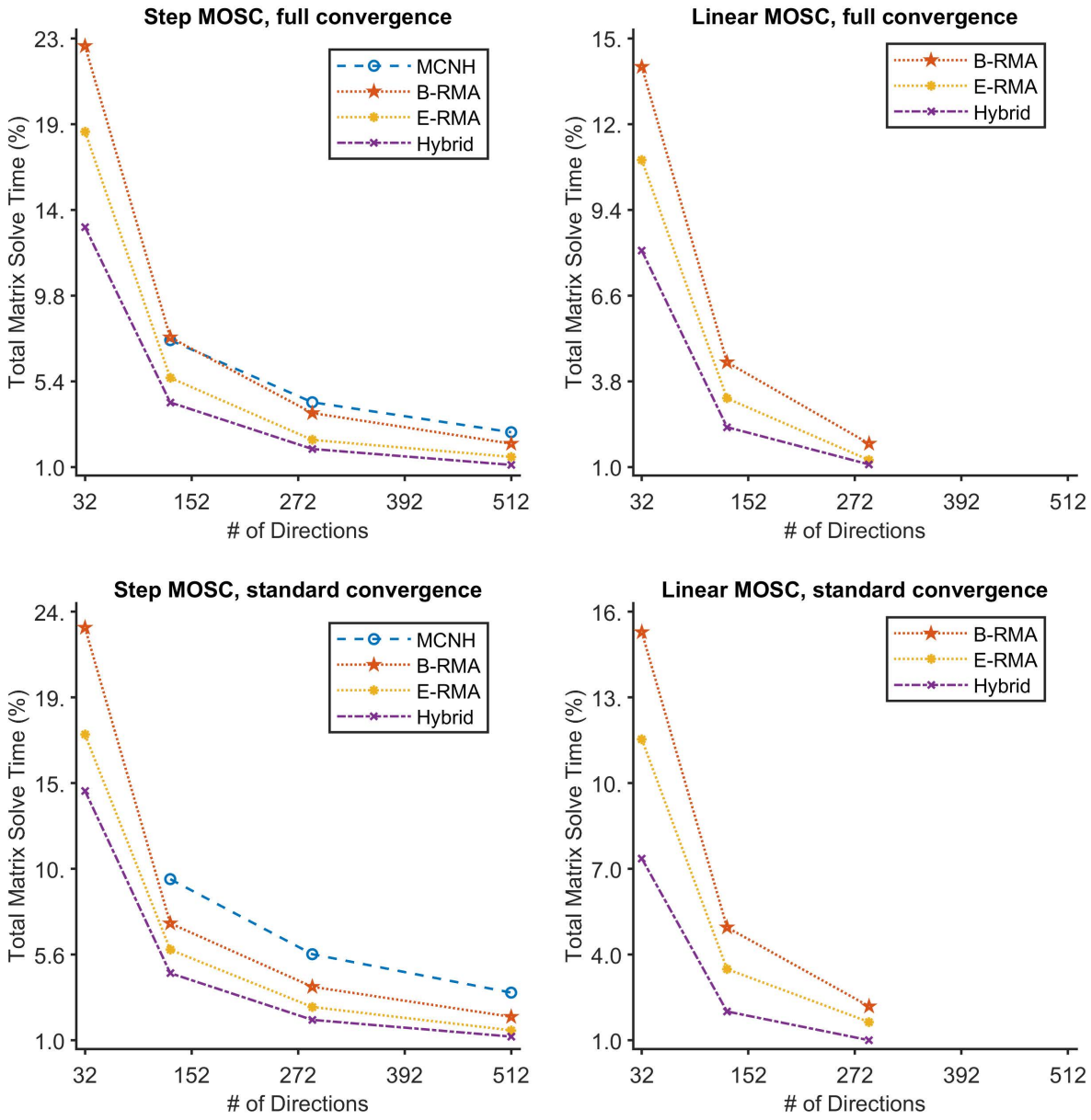

Figure 5.9 – Time to solve the acceleration matrix versus the number of angular directions, when the inner iterations are accelerated by MCNH, E-RMA, B-RMA and the hybrid method (B-RMA&MCNH).

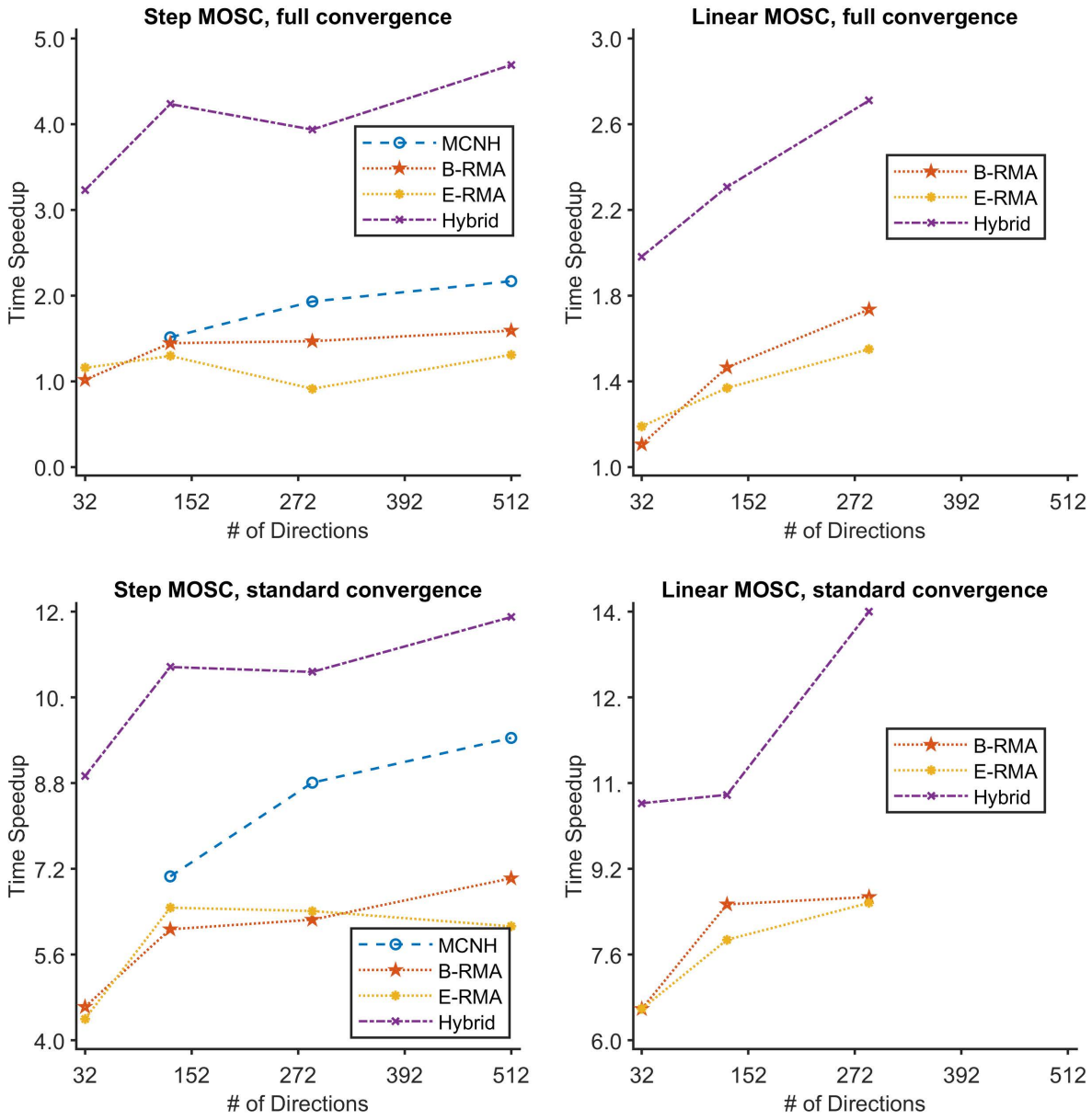

Figure 5.10 – Speedup in time versus the number of angular directions, when the inner iterations are accelerated by MCNH, E-RMA, B-RMA and the hybrid method (B-RMA&MCNH).

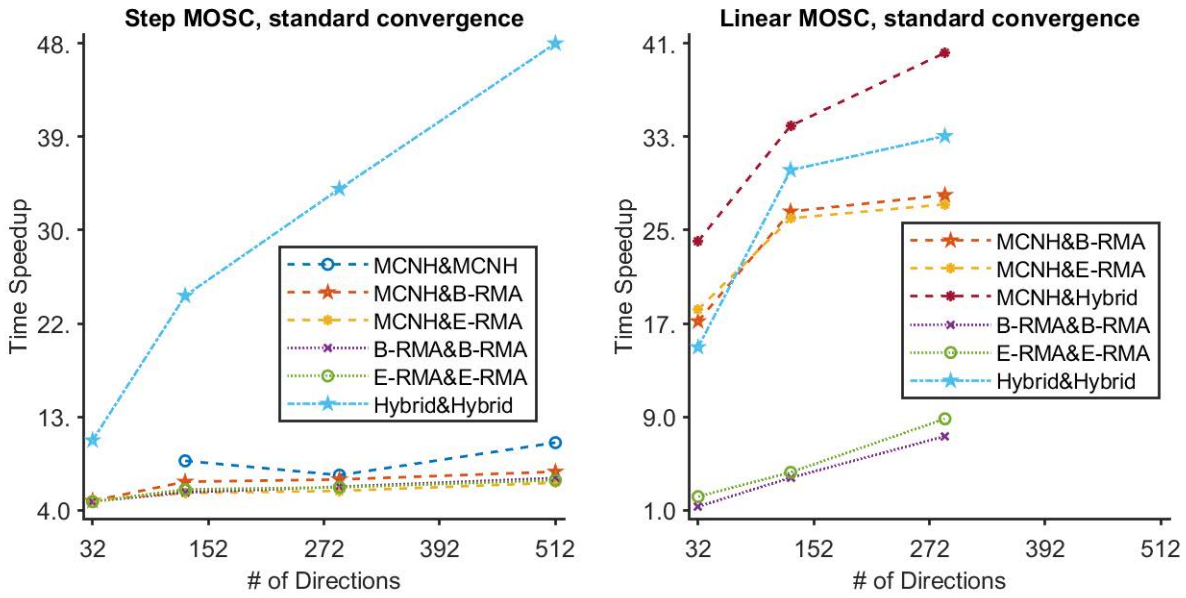

Figure 5.11 – Speedup in time versus the number of angular directions, when the outer and inner iterations are accelerated by MCNH, E-RMA, B-RMA and the hybrid method (B-RMA&MCNH).

5.1.3 Parallel scaling

In this section we will analyze the strong scaling parallel performance of RMA and MCNH as the C5G7 geometry is spatially decomposed into an increasing number of subdomains. Moreover, both the synchronous and asynchronous parallel solving schemes in the library will be tested and compared tested. For all of the test cases, an S_8 angular quadrature was used. When MCNH was applied to the inner iterations, it was found to be unstable for all cases except the sequential. Thus, only RMA was used as an inner solver for these tests. However, both MCNH and RMA were used to accelerate the outer iterations. As explained in Section 4.2.6, the synchronous parallel solver built into PETSc requires that the number of subdomains matches the number of processors. Thus, when we refer to an increasing number of processors, in the following discussion, we are also referring to an increasing number of subdomains. As a final point of notation, since both the outer and inner iterations are being accelerated, the combination of the two schemes will be identified with the combined name (name of outer acceleration method)&(name of inner acceleration method), in the figures.

The results have been partitioned into three parts that describe how the number of outer iterations, total execution time and average execution time, respectively, vary with the number of subdomains for MCNH and RMA. For the second and third parts, the speed up of the parallel cases relative to the sequential case are presented. Finally, the strong scaling of the library and the matrix solver in PETSc are presented in the third part.

Using the above methodology, a comparison of the number of outer iterations required to solve the C5G7 benchmark when the MCNH variant of CMFD, E-RMA and B-RMA are used to accelerate the outer iterations, is shown by Fig. 5.12. When the number of

subdomains is less than 216, RMA requires more outer iterations than MCNH in order to converge to a solution. The fact that RMA is relatively ineffective compared to MCNH for converging the outer iterations is due to the inconsistency of the fixed angular distribution assumption, when the transport solution is far from convergence. The outer iterations must converge the k-effective and the fission source in addition to the self-scattering source, whereas the inner iterations fix the contributions from external energy groups and only converge the self-scattering source. Thus, the solution to the multigroup problem is inherently farther from convergence than the solution for the inner iterations. As stated earlier, the further the solution is from convergence the less effective RMA becomes. This is why RMA is less effective at converging the outer iterations than MCNH despite being more effective at converging the inner iterations.

From Fig. 5.12, the number of outer iterations required when MCNH is used to accelerate them can be seen to increase with the number of processors. This result mirrors the spectral radius analysis in Section 2.20 where the threshold optical thickness at which MCNH becomes unstable decreased as the geometry becomes more spatially decomposed. In contrast, E-RMA and B-RMA require nearly the same number of outer iterations to converge when 8 and 729 subdomains are used. As was shown in Section 3.2.13, the stability of RMA is not affected when the geometry is spatially decomposed. However, the effectiveness of RMA varies greatly as the number of subdomains changes. When solving the inner iterations, the incoming angular flux on the boundary of the subdomain is fixed. This allows for the inner iterations of each subdomain to be converged in parallel without any MPI communications. However, when the subdomain boundary occurs along a large material discontinuity, this fixed angular assumption is very inaccurate. Since RMA's effectiveness is determined by how converged the angular distribution of the flux is, the placement of the subdomain boundaries has an important effect on its performance. In the case when 27 subdomains are used, the boundary between the fuel and the coolant is overlapped by subdomain boundaries. This is why the number of outer iterations reaches a maximum for this spatial decomposition of the C5G7 benchmark, when RMA is applied to the outer iterations. The fact that the different RMA variants require slightly different amounts of iterations to converge is likely due to numerical precision limitations.

The number of outer iterations required when the asynchronous iteration scheme is applied is consistently less than the equivalent synchronous calculation. The asynchronous scheme allows for the tolerance at which the subdomain boundary interface flux is converged to differ from that of the interior interface flux, unlike the synchronous scheme. As stated in Table 5.1, the tolerance at which IDT enforced the subdomain boundary flux to be converged was 1.0^{-3} where as the scattering source, which is dependent on the interior interface flux, is converged within a tolerance of 3.0^{-5}. Thus, the synchronous scheme is effectively over converging the interface boundary flux, resulting in more outer iterations being required. Moreover, the behaviour of the number of outer iterations for the asynchronous iteration scheme, more closely resembles that of the ideal behaviour than the synchronous iteration scheme, for all acceleration methods.

A comparison of the total execution time for the different acceleration methods is shown by Fig. 5.13. When only a few processors are used, MCNH is far better at reducing the execution time than RMA. This is expected since MCNH has already been shown to have an improved effectiveness compared to RMA for accelerating the outer iterations.

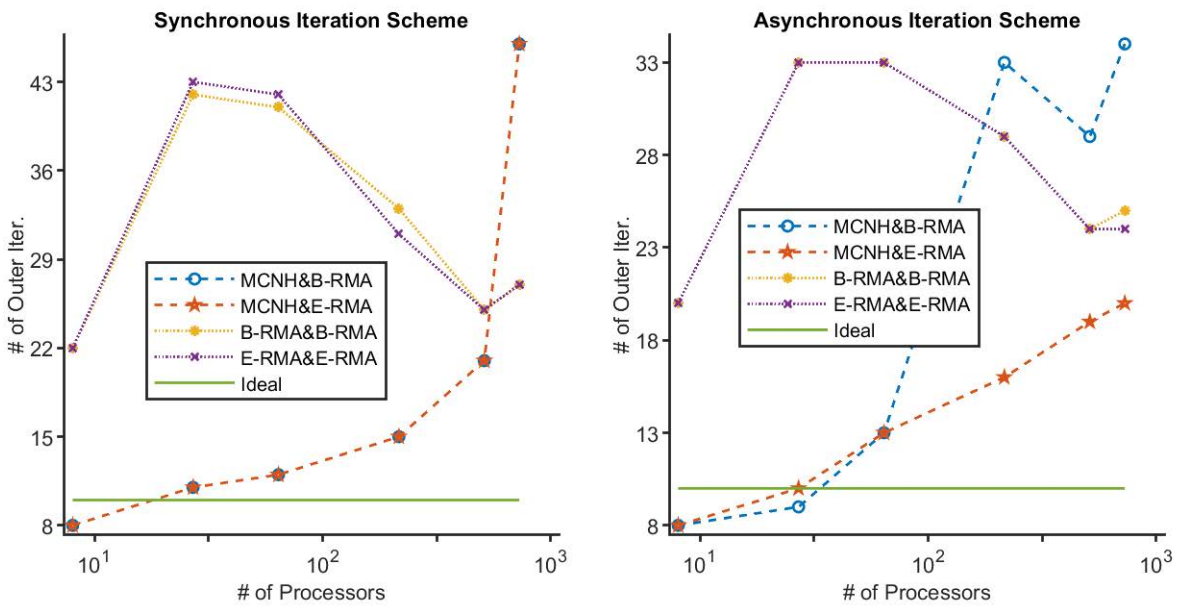

Figure 5.12 – Number of outer iterations as a function of the number of processors (subdomains) for the C5G7 benchmark. Notation for labeling the different acceleration schemes: (name of outer acceleration method)&(name of inner acceleration method).

As well, solving for the RMA operator is computationally more expensive than solving for the MCNH operator, as was shown in Section 5.1.2. However, as the number of processors increases, the execution time of RMA approaches that of MCNH. The primary reason for the convergence of the execution times is the improved parallel scaling of RMA. This can be seen to be true when the speedup in time of the parallel cases relative to the sequential case for MCNH and RMA are compared in Fig. 5.14. This improved parallel scaling is the result of three factors. The first and most important reason is that RMA is computationally more expensive to solve than MCNH. This means that the inter-node communication cost relative to the cost of solving the operator is much lower for RMA than it is for MCNH. The second reason is that the average number of iterations needed to solve the acceleration operator is less for RMA than for MCNH, for all parallel cases, as can be seen from Fig. 5.15. Since the matrices are being solved in parallel, this means that RMA requires a lower number of inter-node communications than MCNH. Third, RMA does not require any parallel communication in order to form its operator, whereas MCNH must synchronize the boundary flux and current in order for the operator to conserve the balance equation. Thus, the relative time to generate the acceleration matrix increases more rapidly for MCNH than for RMA.

As expected from the results of Fig. 5.12 and shown by Fig. 5.13, the asynchronous iteration scheme requires less time to finish executing than the synchronous iteration scheme. Moreover, Fig.5.14 displays that the time speedup of MCNH and RMA more closely follow the ideal behaviour when the asynchronous scheme is employed. In contrast, the time speedup plateaus and then declines as the number of processors increases for the synchronous scheme. The improved parallel performance of the asynchronous iteration scheme comes from its ability to allow the MPI processors to continue to compute while they are waiting for an MPI communication to complete. Thus, when an asynchronous

scheme is used the processors spend far less time idling at MPI communication barriers.

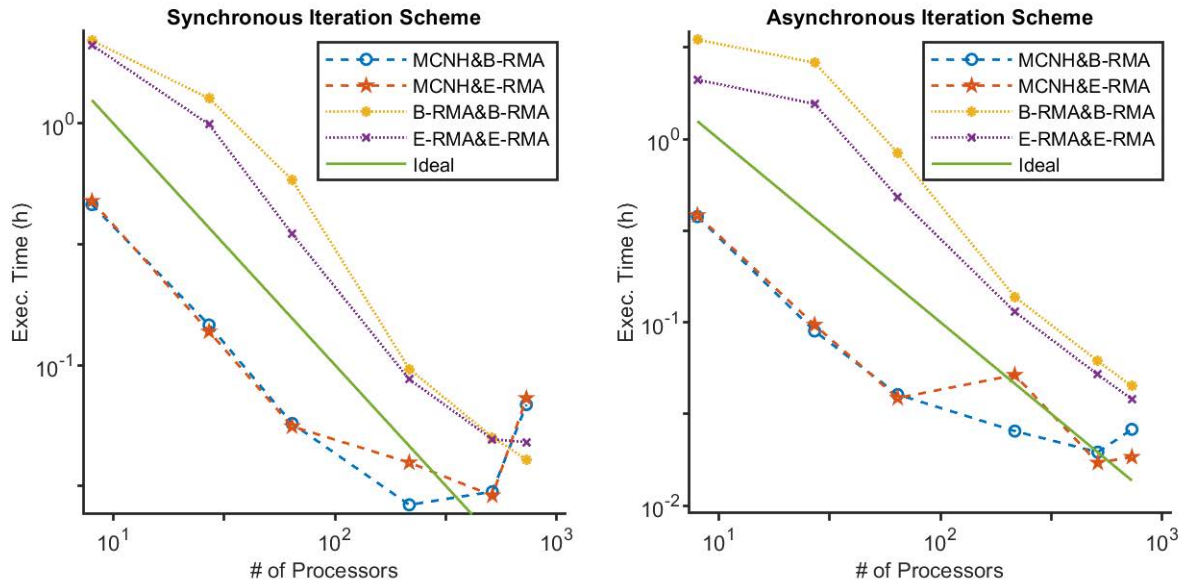

Figure 5.13 – Execution time as a function of the number of processors (subdomains) for the C5G7 benchmark. Notation for labeling the different acceleration schemes: (name of outer acceleration method)&(name of inner acceleration method).

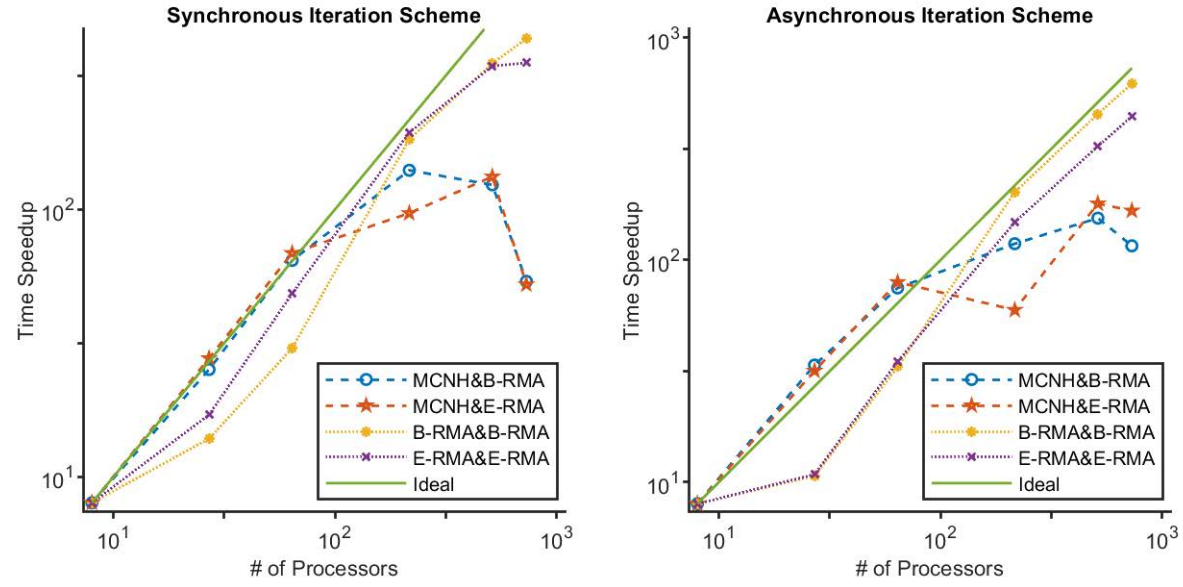

Figure 5.14 – Speedup in execution time as a function of the number of processors (subdomains) for the C5G7 benchmark. Notation for labeling the different acceleration schemes: (name of outer acceleration method)&(name of inner acceleration method).

A comparison of the Strong Scaling Efficiency (SSE) of the execution time when MCNH, E-RMA, and B-RMA are applied is shown by Fig. 5.16. When the number of processors

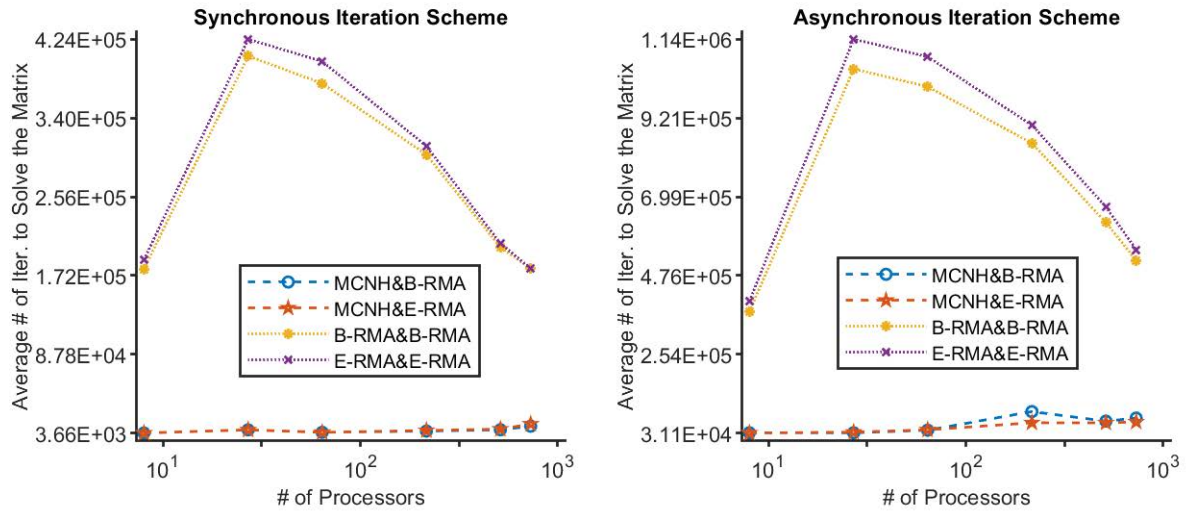

Figure 5.15 – Average number of iterations to solve the acceleration matrix as a function of the number of processors (subdomains) for the C5G7 benchmark. Notation for labeling the different acceleration schemes: (name of outer acceleration method)&(name of inner acceleration method).

is between 1 and 64, the SSE when RMA is used, rapidly decreases compared to when MCNH is used. This is a result of the number of outer iterations significantly increasing due to the spatial decomposition. However, once the number of processors is increased beyond 64 processors, the SSE of the execution time when RMA is applied, is consistently better than when MCNH is applied. This is due to the improved computational scalability of RMA relative to MCNH. It is important to remember that the SSE of the execution time is determined by the strong scaling of IDT as well as the library. As seen in Fig. 5.17, the percentage of time spent in library tends to increase as the number of processors increases. This means that the strong scaling within the transport solver of IDT is generally better than that of the acceleration methods within the library. This is a result of IDT placing the responsibility for the convergence of the global domain on the acceleration method rather than the transport solver. Where the transport solver solves each subdomain independently, with a focus on converging the scattering source, the outer acceleration solver must consider the global problem in order to converge the k and subsequently the fission source. In order to solve the global problem many MPI communications are required in order to transmit the interface flux on the subdomain boundaries to the adjacent subdomains. This is particularly true when MCNH is used as an acceleration matrix, due to its poor scalability compared to RMA. By comparing Fig. 5.17 to Fig. 5.18, the time to solve the acceleration matrix can be seen to dominate the cost of applying the library when RMA is used as an acceleration method. Thus, the parallel scalability of RMA is primarily dependent on the parallel matrix solver within PETSc.

In the previous discussion we analyzed how the total time spent in each section of the library varied with increasing parallelism. Variations in the total time spent are the combination of the changes in the number of iterations as well as the cost of each iteration. While the change in the number of iterations has already been described, the change in

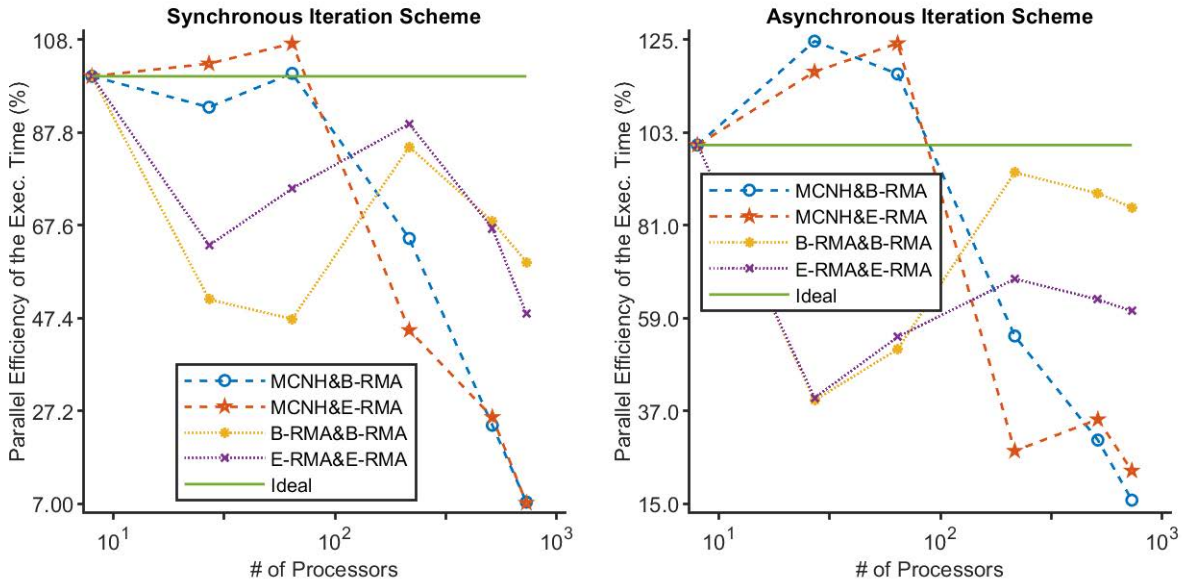

Figure 5.16 – Strong scaling efficiency of the execution time as a function of the number of processors (subdomains) for the C5G7 benchmark. Notation for labeling the different acceleration schemes: (name of outer acceleration method)&(name of inner acceleration method).

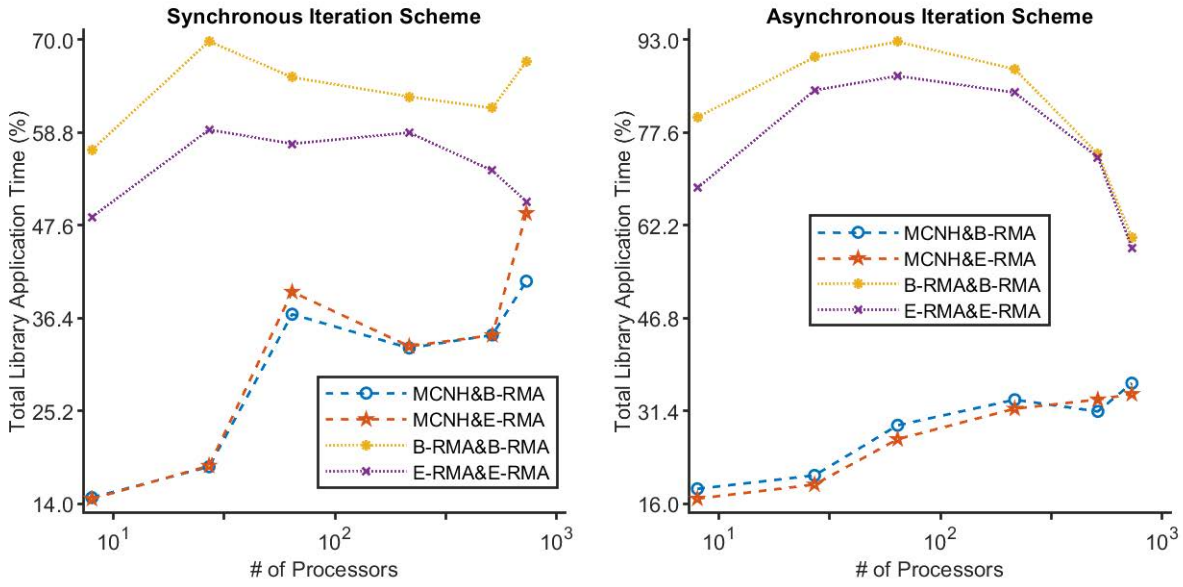

Figure 5.17 – Total time for applying the acceleration library, as a percentage of the execution time, versus the number of processors (subdomains) for the C5G7 benchmark. Notation for labeling the different acceleration schemes: (name of outer acceleration method)&(name of inner acceleration method).

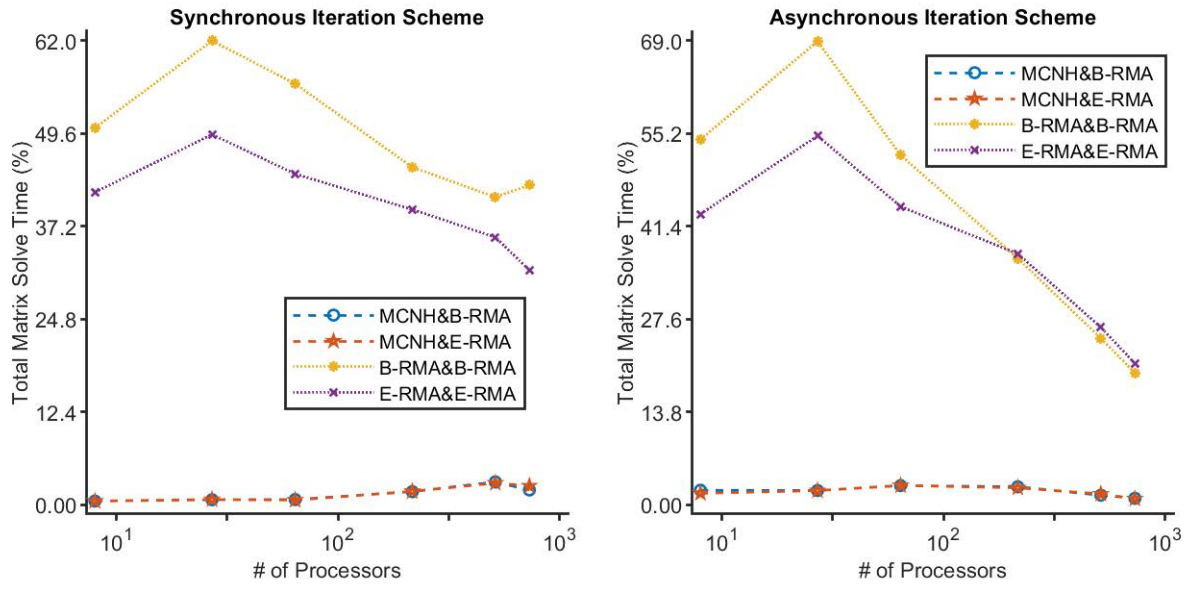

Figure 5.18 – Total time for solving the acceleration matrix, as a percentage of the execution time, versus the number of processors (subdomains) for the C5G7 benchmark. Notation for labeling the different acceleration schemes: (name of outer acceleration method)&(name of inner acceleration method).

the computational cost per iteration has not. The average cost of applying the library when RMA and MCNH are used is shown by Fig. 5.19. When MCNH is used the minimum average cost of applying the library occurs when the geometry has been spatially decomposed into 512 subdomains. Any further decomposition of the geometry for MCNH leads to an increase in inter-node communication which outweighs the benefits of computing a smaller subdomain. However, when RMA is applied to the outer iterations, the application time continues to decrease as the geometry is decomposed into a greater number of subdomains.

The speedup of the average cost of applying the library for the parallel cases relative to the sequential cases is shown by Fig. 5.20. Per outer iteration, E-RMA and B-RMA scale $\sim 10 \rightarrow 4$ times better than MCNH when the number of processors is set to 729. In comparison to the speedup for the total execution time, as was shown in Fig. 5.14, the average speed up of RMA per outer iteration is greatly improved. Thus, it is clear that RMA is primarily being held back by its reduced effectiveness as an acceleration method for the outer iterations, compared to MCNH. Ideas on how future researchers could remedy the effectiveness of RMA are presented in Section 6.4.2. As noted before, the asynchronous scheme can be observed to scale significantly better than the synchronous scheme as the number of processors is increased.

As was shown by Fig. 5.18, it is the parallel performance of PETSc that primarily determines the parallel performance of the library. The speedup of the average time required to solve the acceleration matrix is shown by Fig. 5.21. By comparing Fig. 5.21 to Fig. 5.20, the scalability of the matrix solver in PETSc is better than the rest of the library. When the synchronous scheme is used, the diminished scalability of the library, compared

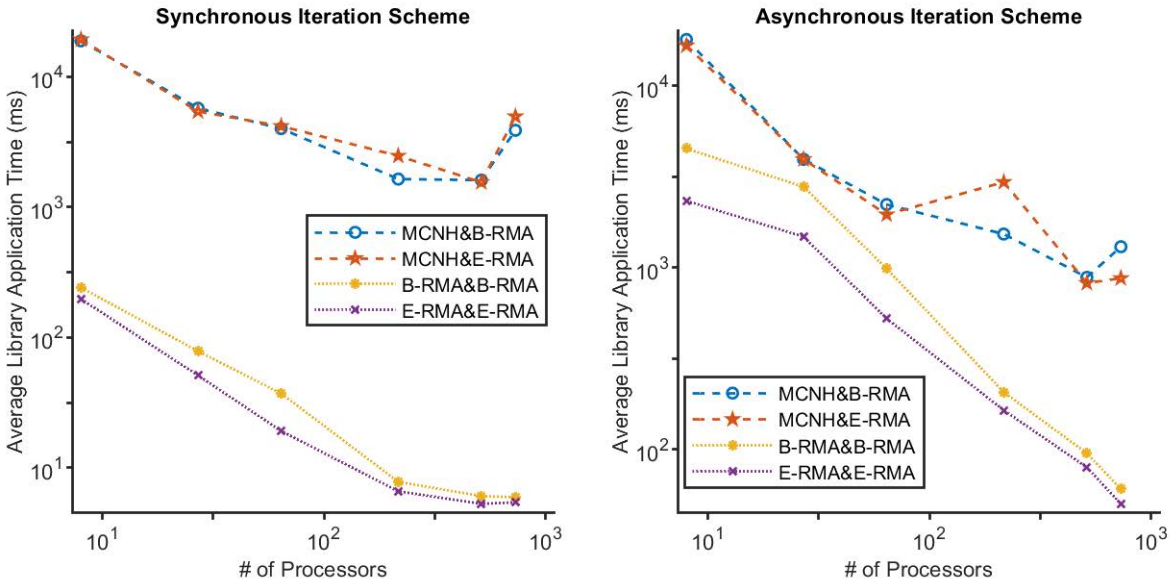

Figure 5.19 – Average time required to apply the acceleration library as a function of the number of processors (subdomains) for the C5G7 benchmark. Notation for labeling the different acceleration schemes: (name of outer acceleration method)&(name of inner acceleration method).

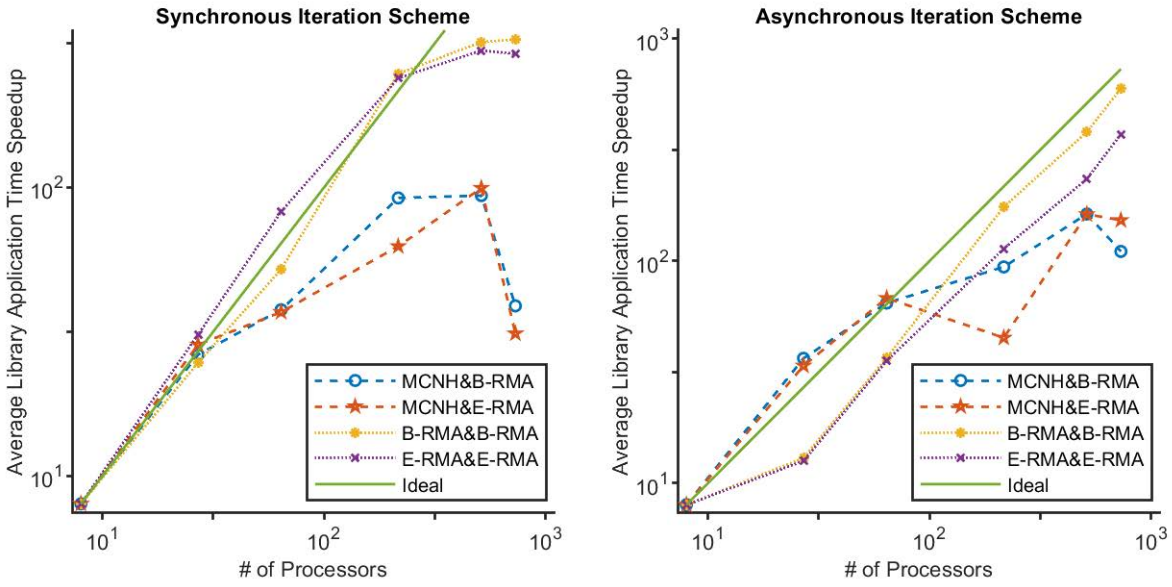

Figure 5.20 – Parallel speedup of the average time to apply the library as a function of the number of processors (subdomains) for the C5G7 benchmark. Notation for labeling the different acceleration schemes: (name of outer acceleration method)&(name of inner acceleration method).

to the matrix solver in PETSc, is primarily due to the formation of the parallel matrix and solver objects in PETSc. This is particularly true in the case when RMA is used, since no parallel communications, outside of the PETSc routines, are needed. With future releases of PETSc and optimization of its usage in the library, it is likely that this difference in scalability will be eliminated. When the asynchronous scheme is used, all the MPI communications occur outside of the PETSc matrix solving routine, in the library, resulting in the PETSc solver routines having nearly perfect scalability.

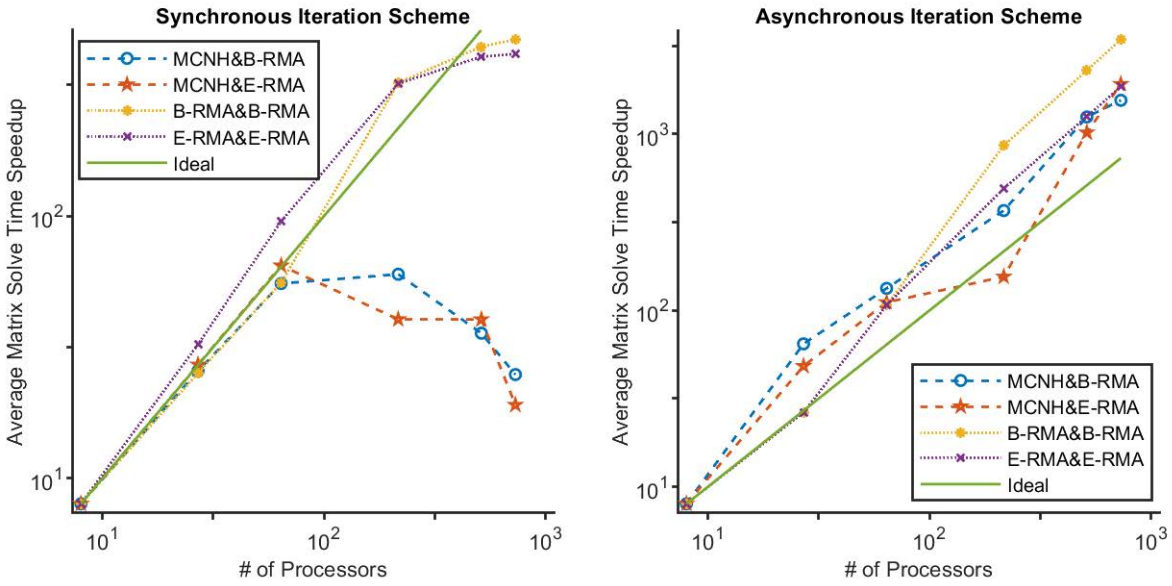

Figure 5.21 – Parallel speedup of the average time to solve the acceleration matrix as a function of the number of processors (subdomains) for the C5G7 benchmark. Notation for labeling the different acceleration schemes: (name of outer acceleration method)&(name of inner acceleration method).

The strong scaling efficiency of the average time required to solve the acceleration matrix is shown by Fig. 5.22. The SSE of the asynchronous scheme greatly exceeds 100%, since no MPI communications are occurring within the matrix solver and the cost of solving the matrix of each subdomain decreases quadratically with the number of subdomains. When the synchronous scheme is employed, the SSE of E-RMA and B-RMA is greatly improved compared to MCNH. However, even the SSE of RMA drops significantly when the number of processors is increased beyond 216. Each matrix solve only takes a few milliseconds, when the number of processors is increased beyond 216. As well there are many inter-node communications occurring per matrix solve. Thus, the ratio of the computational cost of solving the matrix relative to the summed cost of the inter-node communications is quite low when the synchronous scheme is used. Such a low ratio is more commonly referred to as fine grain parallelism. In order to coarsen the parallelism without increasing size of the task, the number of communications per matrix solve must be reduced. As was discussed in Section 5.1.1, matrix preconditioners such as the geometric algebraic multigrid method are able to greatly reduce the number of iterations required to solve a matrix. Since the number of communications is linked to the number of iterations, the scalability of PETSc matrix solver routines could be improved with the application of multigrid precondition-

ers. The use of such preconditioners for solving neutron transport problems has already been shown to have merit by Azmy, [9].

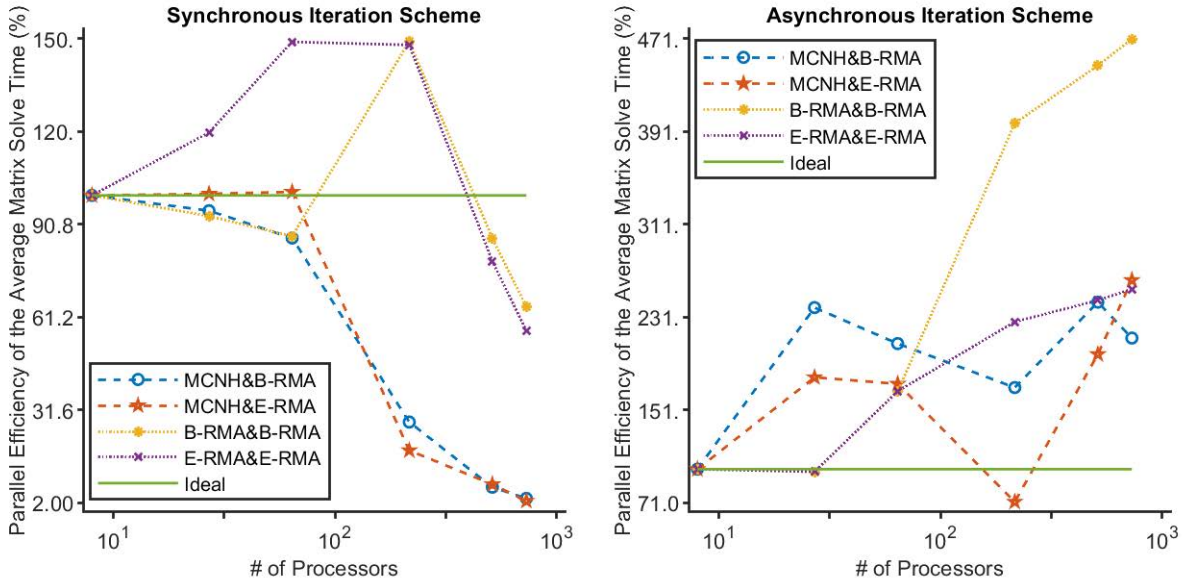

Figure 5.22 – Strong scaling efficiency for the average acceleration matrix solve time as a function of the number of processors (subdomains) for the C5G7 benchmark. Notation for labeling the different acceleration schemes: (name of outer acceleration method)&(name of inner acceleration method).

5.1.4 Energy and spatial coarsening

As discussed in Section 2.18.3, the ability to solve a coarse mesh is an important attribute of nonlinear acceleration methods. In this section the performance of the MCNH variant of CMFD applied to step MOSC for the C5G7 benchmark is compared for a coarse mesh of varying size. For this comparison, the spatial domain of the C5G7 problem was decomposed into 8 subdomains for parallel computation. We will denote the number of coarse groups as $\tilde{g}$ and the number of fine spatial cells per coarse spatial cell as p. When p was set greater than 2, MCNH failed to converge the k-effective and the interface flux. Thus, only results for when $p = 1$, Table 5.8, and $p = 2$, Table 5.9, are shown. Moreover, for many coarse mesh configurations, MCNH was not able to converge the interface flux within a precision of 1×10^{-3}. However, it was still able to converge the k-effective within a precision of 5×10^{-5}. Thus, the number of outer iterations required to converge the k-effective and the interface flux are presented separately, in Tables 5.8 and 5.9, so that the effect of coarsening on the convergence of both these properties can be observed.

The number of outer iterations needed to converge the k-effective and the interface flux, reaches a minimum when the coarse energy discretization is the same as the fine energy discretization and p is set to 1 or 2. As well, the average amount of time required to solve the multigroup acceleration operator reaches a maximum when the coarse mesh is equivalent to the fine mesh. As was previously discussed, only a small fraction of the

execution time is spent solving the MCNH multigroup acceleration operator. This means that reducing the number of outer iterations is the most important factor for reducing the execution time. Thus, when the coarse energy discretization is the same as the fine energy discretization and p is set to 1 or 2, the execution time reaches its lowest values of 0.290h and 0.296h, respectively. It is important to note that the discrepancy in the previous two execution times is due to random variation in the computational performance and not is due to a difference in the effectiveness.

When $\tilde{g} \in [2, 6]$, MCNH is unable to converge the interface flux within the given precision. However, some interesting trends can still be observed for the convergence of the k-effective. First, the number of outer iterations required to converge the k-effective decreases when p is increased from 1 to 2. By increasing the spatial coarsening, MCNH is isolating the lower spatial frequencies. In other words, the coarse spatial mesh is more effective at converging the global spatial contributions. The k-effective is a parameter over all space and energy, which is why the cases when $p = 2$ are more effective at converging the k-effective. Second, the average time per multigroup acceleration solve reaches a minimum when $\tilde{g} = 2$ and $p = 2$. Normally the coarsest mesh, in this case $\tilde{g} = 1$ and $p = 2$ should compute in the least amount of time. However, this is not true for this benchmark because the convergence criteria is not the same when $\tilde{g} \in [2, 6]$ compared to when $\tilde{g} = 1$ and $\tilde{g} = 7$. When we just focus on the cases when $\tilde{g} \in [2, 6]$, the expected trend of the average computational cost increasing with the size of the coarse mesh can be clearly discerned.

$\tilde{g}$	# of Outer Iter. to Converge the k-effective	# of Outer Iter. to Converge the Interface Flux	Avg. Time Per Multigroup Acceleration Solve (s)
1	10	14	1.34
2	22	NC	1.16
3	33	NC	1.68
4	24	NC	2.07
5	30	NC	2.33
6	30	NC	2.39
7	9	10	6.00

Table 5.8 – A comparison of how the performance of MCNH varies with the number of coarse energy groups when p=1, for the C5G7 benchmark

$\tilde{g}$	# of Outer Iter. to Converge the k-effective	# of Outer Iter. to Converge the Interface Flux	Avg. Time Per Multigroup Acceleration Solve (s)
1	15	17	0.636
2	7	NC	0.344
3	15	NC	0.398
4	25	NC	0.646
5	16	NC	0.821
6	17	NC	0.715
7	8	10	5.84

Table 5.9 – A comparison of how the performance of MCNH varies with the number of coarse energy groups when p=2, for the C5G7 benchmark

5.2 Zero Power Physics Reactor

In this section we will analyze the 3D model of the Zero Power Physics Reactor (ZPPR) [24]. The ZPPR was used to investigate a variety of core designs. Moreover, the benchmarks built off it are important for assessing the accuracy of neutron transport codes. An illustration of the third lattice configuration of the ZPPR is shown by Fig. 5.23. In order to study the performance of RMA for homogeneous cells we chose to use the 16-group, homogeneous fuel pin, cross section library. The convergence criteria used to solve the ZPPR benchmark is listed in Table 5.10.

Convergence Criteria	k-effective	Fission Source	Self-Scattering Source
Precision	1.0×10^{-5}	1.0×10^{-4}	1.0×10^{-5}

Table 5.10 – The convergence criteria used to solve the ZPPR benchmark.

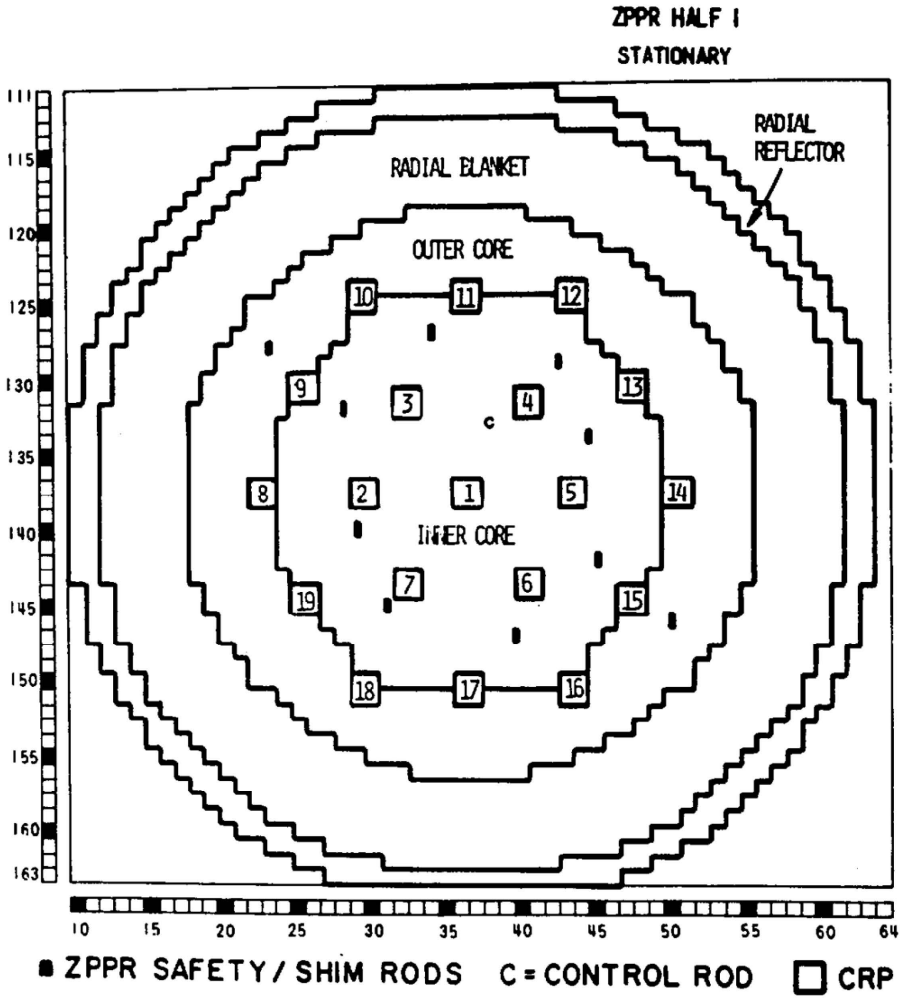

Figure 5.23 – An illustration of the ZPPR geometry, provided by [24].

5.2.1 Anisotropic Scattering Source

Until this point, we have only investigated RMA when an isotropic scattering source is used. In this section, the performance of RMA is compared when a P_0 and P_1 representation of the scattering source is used for the ZPPR benchmark. In this way, the effects on the performance of RMA when the order of the angular representation is increased can be discerned. As well, RMA is investigated for both the step and linear MOSC transport schemes. MCNH was not able to converge for any of the source representations or transport schemes so it was left out of the comparison. We chose to only accelerate the inner iterations of the benchmark in order to isolate the effect the scattering source representation has on the convergence of the inner iterations. As well, each simulation has been run sequentially. For all cases, a S_{16} angular distribution was used to discretize the angular domain.

The results of the comparison when the scattering source is isotropic are shown in Table 5.11. For the step MOSC transport scheme, E-RMA and B-RMA speedup the number of iterations required by a factor of 22.5 times. However, the runtime for both methods is longer than that of transport alone. As explained by Table 5.12, this due to the large amount of time taken to generate the RMA operators relative to the execution time. The code responsible for the generation of the operators is under development, with a focus on generality and transparency rather than optimization. This lack of optimization will be addressed in future and should lead to significant improvement of the time speedup when RMA is applied. However, our current goal is the reduction of inner iterations. In contrast to the generation time, the inversion of the acceleration operator takes up only a few percent of the total elapsed time.

By comparing the data from the step and linear MOSC transport schemes, the number of iterations required to converge is almost constant when it is accelerated by E-RMA or B-RMA. In contrast the number of iterations required by the unaccelerated transport operator increases significantly. As a result, the iteration speedup of E-RMA and B-RMA increases from a value of 22.5 for step MOSC to a value of 28.4 for linear MOSC. Due to the increased computational cost of linear MOSC relative to step MOSC, the speedup in time of both variants of RMA increases to a value of 1.03.

Acceleration Type	MOSC Type	# of Outer Iter.	# of Inner Iter.	Exec. Time (h)	Speedup in Iter.	Speedup in Time
None	Step	114	41050	7.50	1.00	1.00
E-RMA	Step	105	1822	27.2	22.5	0.276
B-RMA	Step	105	1822	27.5	22.5	0.273
None	Linear	125	51686	67.7	1.00	1.00
E-RMA	Linear	107	1819	65.8	28.4	1.03
B-RMA	Linear	107	1819	65.8	28.4	1.03

Table 5.11 – A comparison of the effectiveness of E-RMA and B-RMA for accelerating the inner iterations of the ZPPR benchmark when an isotropic representation of the scattering source is used.

The results of the comparison when the scattering source has a P_1 representation are shown in Table 5.13. Unfortunately, none of the simulations with the linear MOSC

Acceleration Type	MOSC Type	Time to Solve Operator (h)	Time to Solve Operator (%)	Time to Generate Operator (h)	Time to Generate Operator (%)
E-RMA	Step	2.17	7.97	22.7	83.3
B-RMA	Step	3.20	11.6	21.9	79.7
E-RMA	Linear	1.82	2.76	61.6	93.6
B-RMA	Linear	3.69	5.61	59.8	90.8

Table 5.12 – A comparison of the matrix solver performance of E-RMA and B-RMA when they are accelerating the inner iterations of the ZPPR benchmark when an isotropic representation of the scattering source is used.

transport scheme and the anisotropic representation of the source managed to converge within the 72h time limit imposed by the CCRT. By comparing the results of the P_1 and P_0 representation of the scattering source for the step MOSC transport scheme, the P_1 representation requires significantly less iterations to converge than the isotropic representation. However, the speedup in the number of iterations required when E-RMA and B-RMA are applied stays nearly constant with a factor of 22.6 times. Thus, the effectiveness of RMA is not affected by higher order representations of the scattering source for the ZPPR benchmark. This result is expected since, as was discussed in Section 3.2, RMA preserves the contributions of the higher order moments of the scattering source.

As before, the runtime for step MOSC when accelerated by either variant of RMA is longer than that of transport alone. However, the speedup in time has almost doubled compared to when the isotropic representation of the scattering source is used. This is a result of the transport operator becoming more costly to compute due to the P_1 representation of the source. In contrast, the cost of solving the RMA operator can be seen in Table 5.14 to remain almost the same. This is because the constraints of RMA do not expand with representation of the scattering source. Instead, RMA collapses the contributions of the higher order components of the scattering source on to the zeroth moment.

Acceleration Type	MOSC Type	# of Outer Iter.	# of Inner Iter.	Exec. Time (h)	Speedup in Iter.	Speedup in Time
None	Step	107	38670	18.9	1.00	1.00
E-RMA	Step	98	1708	39.2	22.6	0.482
B-RMA	Step	98	1708	39.3	22.6	0.481

Table 5.13 – A comparison of the effectiveness of E-RMA and B-RMA for accelerating the inner iterations of the ZPPR benchmark when a P_1 representation of the scattering source is used.

Acceleration Type	MOSC Type	Time to Solve Operator (h)	Time to Solve Operator (%)	Time to Generate Operator (h)	Time to Generate Operator (%)
E-RMA	Step	1.91	4.88	35.0	89.4
B-RMA	Step	3.49	12.7	23.9	86.9

Table 5.14 – A comparison of the matrix solver performance of E-RMA and B-RMA when they are accelerating the inner iterations of the ZPPR benchmark when a P_1 representation of the scattering source is used.

Chapter 6

Conclusions and future work

6.1 Conclusions

In this manuscript a new library of acceleration methods for improving the convergence of the neutron transport equation in a parallel context, is introduced and analyzed. Moreover, two new types of nonlinear acceleration methods, SVRM and RMA, respectively, are proposed and investigated.

6.2 Spatially variant rebalance method

The Spatially Variant Rebalance Method (SVRM), represents a new class of rebalancing methods. By including linear spatial projections of the balance equation as constraints and by using a parabolic spatial representation of the flux, SVRM was designed to better preserve the high-order spatial variations of the flux. By means of Fourier and spectral radius analysis, SVRM was shown to have improved convergence properties compared to CMFD for cases with optically diffusive mediums. These improvements were demonstrated to be due to SVRM's reduction, relative to CMFD, of the eigenmodes associated with the high spatial frequencies. This reduction was considered to be the direct result of SVRM's improved spatial representation. However, SVRM was shown to still suffer from instability when the optical thickness was increased beyond a threshold ~ 30 mfp and the scattering ratio was 0.99. This instability was the result of an approximation made by SVRM when the first order spatial moment of the flux was not provided. However, such an approximation would not be needed when the first order spatial moment of the flux is provided, as is the case for linear MOSC. Thus, while SVRM's high-order representation of the flux allowed SVRM to improve upon its effectiveness and stability, it was not accurate enough to prevent instability. Also, the expansion of SVRM for an arbitrary cell shape and for multidimensional problems was demonstrated to not be trivial. For these reasons, we choose to discontinue the development of SVRM for the remainder of the PhD in order to focus on the development of RMA.

6.3 Response matrix acceleration method

Two variants of a new nonlinear DP0 method, called the Response Matrix Acceleration method (RMA), were proposed. By means of Fourier, spectral radius analysis and numer-

ical benchmarks, RMA was shown to be stable for any conservative transport scheme that ensures the positivity of the flux. Moreover, RMA was found to complement CMFD by offering improved performance in cases where CMFD is ineffective. RMA achieved these results by deriving its coefficients directly from the transport equation and by having an increased number of degrees of freedom compared to CMFD. A hybrid method was created in an attempt to optimize the computational expense of applying RMA relative to its improved effectiveness. This method worked by using RMA when $\max_{i \in I}(\tau_{g,i}) > 1$ and CMFD otherwise. Large performance improvements were demonstrated using the hybrid approach for accelerating the inner iterations of the C5G7 benchmark. When applied to the inner iterations of the C5G7 and ZPPR benchmark, RMA offered consistent performance when the S_N quadrature and the P_N representation of the scattering source were varied for the step MOSC and linear MOSC transport operators.

While RMA performed well as an acceleration technique for the inner iterations, it performed poorly as an outer iteration acceleration technique for the spatially decomposed C5G7 benchmark. This was theorized to be caused by RMA's lack of self consistency when the current transport solution is far from convergence. In contrast, CMFD worked very effectively as an acceleration technique for the outer iterations when RMA was used as the inner acceleration method. Therefore, in its current form, RMA should be primarily considered for inner acceleration. However, RMA had greatly improved parallel scalability compared to CMFD. When the C5G7 geometry was decomposed into 216 or more subdomains, the performance of RMA approached that of CMFD. Moreover, When the C5G7 geometry was decomposed into 1000 subdomains, RMA was able to converge but CMFD could not.

6.4 Future work

6.4.1 Spatially variant rebalance method

While SVRM in its current form has too many drawbacks to be considered a competitive option, such issues could be resolved with future work. As was done for RMA, the consistency of SVRM with the transport operator could be ensured if SVRM computed its coefficients from the transport coefficients. Additionally, SVRM would also have to use a separate spatial representation for the contribution of the collided and scattered flux, as is done by transport, in order to be consistent. Finally, piece-wise representations of the flux within the region should be considered to allow the number of equations to expand with the number of surfaces.

6.4.2 Response matrix acceleration method

While RMA is an intriguing option as it is currently formulated, there are still many possibilities for improving its performance that should be considered. One of the main disadvantages of RMA comes from its need to cumulate the transport data. This greatly increases the computational cost of the method and complicates its implementation in the user's code. To avoid this cumulation, one could assume that the RMA operator is cyclical and take advantage of geometric symmetries in order to determine the coefficients of RMA using commonly provided data. As was noted in Section 5.1.2, RMA is less

effective when the transport solution is far from convergence due to the inaccuracy of the fixed angular distribution. This draw back could be rectified by preconditioning the transport solution in space and energy before applying RMA. Another option to rectify this problem would be to apply a relaxation method to the interface angular flux before computing the coefficients of RMA and again after the application of RMA. In this way RMA would behave like a nonlinear multigrid method with the relaxation methods acting as the smoothers. One could also attempt to lessen the sensitivity of RMA to the angular flux distribution at the early stages of convergence by solving RMA on an optical thick coarse mesh and then progressively refining the mesh as the transport solution approaches convergence. Finally, a nonlinear solving scheme by which the contribution of the scalar flux to the coefficients of RMA is considered, would be of great interest for improving the effectiveness of RMA, particularly for the outer iterations

6.4.3 The library

Due to the time restrictions of the PhD, many features that are available in the library were not tested and left for future users to analyze. The ability to have unstructured meshes in energy, could allow for the stabilization of methods such as CMFD and the optimization of RMA. As well, the asynchronous nonlinear parallel solving scheme available in the library, or the multigrid preconditioner available in PETSc, could lead to greatly improved parallel strong scaling. Moreover, the computational cost of solving the acceleration could be greatly reduced by taking advantage of the GPGPU parallelism that exists in PETSc.

There were also many features that we did not have the time to implement. The most notable feature of these would be the ability to perform the nonlinear solve of the accelerated transport scheme within the library. By having the user provide a pointer to the function that either forms the transport operator or runs the transport sweep, the library could take advantage of the nonlinear solving techniques within PETSc for additional acceleration. Another, missing feature of great interest is the ability to use the extension of PETSc, Scalable Library for Eigenvalue Problem Computations (SLEPc), to solve the eigenvalue problem when converging the outer iterations. The expansion of the number of options for solving the eigenvalue problem would likely lead to a significant performance improvement.

Appendix A

Complete orthogonal sets

A.1 Legendre polynomials

Legendre polynomials, are a complete set of polynomials that satisfy the following orthogonality relationship,

$$\frac{1}{2}\int_{-1}^{1} d\mu \; P_l(\mu)P_{l'}(\mu) = \frac{\delta_{l,l'}}{2l+1},$$

where $\delta_{l,l'}$ is the Kronecker delta, [25]. The Legendre polynomials can be defined by the Rodrigues' formula,

$$P_l(\mu) = \frac{1}{2^l l!}\frac{d^l}{d\mu^l}(\mu^2-1)^l.$$

A.2 Associated Legendre polynomials

The associated Legendre polynomial, [25], is defined by the relation,

$$P_l^k(\mu) = (-1)^k(1-\mu^2)^{k/2}\frac{d^k}{d\mu^k}P_l(\mu).$$

Moreover, the associated Legendre polynomial satisfies the orthogonality relationship,

$$\frac{1}{2}\int_{-1}^{1} d\mu \; P_l^k(\mu)P_{l'}^{k'}(\mu) = \frac{\delta_{l,l'}\delta_{k,k'}}{2l+1}\frac{(l+k)!}{(l-k)!}.$$

A.3 Spherical harmonics

The normalized real spherical harmonics, [25], can be defined by the relation,

$$Y_{l,k}(\hat{\Omega}) = \begin{cases} (-1)^k\sqrt{2}\left[\frac{(2l+1)(l-|k|)!}{(l+|k|)!}\right]^{1/2} P_l^{|k|}(\mu)\sin(|k|\omega) & \text{when } k<0 \\ [2l+1]^{1/2}\,P_l^k(\mu) & \text{when } k=0\,, \\ (-1)^k\sqrt{2}\left[\frac{(2l+1)(l-k)!}{(l+k)!}\right]^{1/2} P_l^k(\mu)\cos(k\omega) & \text{when } k>0 \end{cases}$$

Moreover, the normalized real spherical harmonics must satisfy the orthogonality relationship,

$$\frac{1}{4\pi}\int_{4\pi} d^2\hat{\Omega} \; Y_{l,k}(\hat{\Omega})Y_{l',k'}(\hat{\Omega}) = \delta_{l,l'}\delta_{k,k'},$$

Appendix B

1D matrix form

In this section of the appendix, the matrix form of MOC, SVRM and RMA is defined for a 1D geometry with isotropic scattering. For the two acceleration methods, the definition of their discrete to moment operator and prolongation operator are included. The discrete to moment operator represents the cumulation of the required angular moments of the flux along the interface and volume, for the acceleration method. The 1D matrix form of SVRM's matrix operator and prolongation operator have already been defined in Sections 3.1.7 and 3.1.8 but they are included again here for completeness. The purpose of this section is to explicitly define the operators used by spectral radius analysis and the Fourier analysis. Namely, the operators of Eqs. (2.46a) and (3.18). Fig. B.1 illustrates the indexing scheme employed by the authors for 1D geometries. Subscripts L and R denote the left and the right side of the cell.

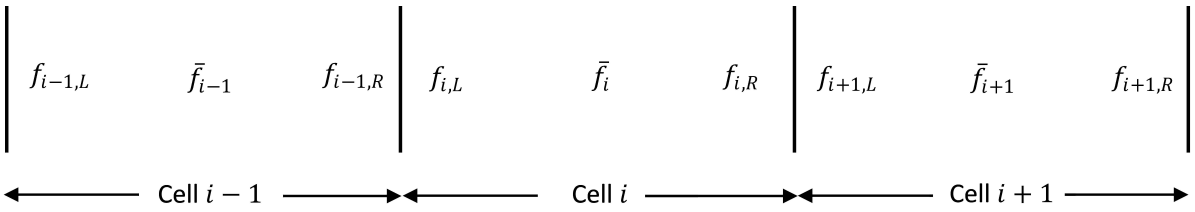

Figure B.1 – An illustration of the indexing scheme for a 1D geometry, for an ambiguous variable f.

B.1 MOC

The MOC inner iteration scheme can conveniently be written in matrix form as,

$$\boldsymbol{\Psi}_i - \underline{\underline{\boldsymbol{L}}}_{i,i-1}\ \boldsymbol{\Psi}_{i-1} - \underline{\underline{\boldsymbol{L}}}_{i,i+1}\ \boldsymbol{\Psi}_{i+1} = \underline{\underline{\boldsymbol{L}}}_{i,i}\boldsymbol{\Psi}_i + \boldsymbol{Q}_i, \tag{B.1}$$

The matrices and vectors of Eq. (B.1) are defined as,

$$\underline{\underline{\boldsymbol{L}}}_{i,i} = \begin{bmatrix} 0 & 0 & \boldsymbol{E}_i \\ 0 & 0 & \boldsymbol{E}_i \\ 0 & 0 & \Sigma_{sc,i} C_i \end{bmatrix}, \tag{B.2a}$$

$$\underline{\underline{\boldsymbol{L}}}_{i,i-1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \underline{\underline{\boldsymbol{T}}}_i & 0 \\ 0 & \boldsymbol{I}_i & 0 \end{bmatrix}, \tag{B.2b}$$

$$\underline{\underline{\boldsymbol{L}}}_{i,i+1} = \begin{bmatrix} \underline{\underline{\boldsymbol{T}}}_i & 0 & 0 \\ 0 & 0 & 0 \\ \boldsymbol{I}_i & 0 & 0 \end{bmatrix}, \tag{B.2c}$$

$$\boldsymbol{\Psi}_i = \begin{bmatrix} \boldsymbol{\Psi}^+_{i,L} \\ \boldsymbol{\Psi}^+_{i,R} \\ \bar{\phi}_i \end{bmatrix}, \tag{B.2d}$$

$$\boldsymbol{Q}_i = \begin{bmatrix} \tilde{\boldsymbol{E}}_i Q_{Ext,i} \\ \tilde{\boldsymbol{E}}_i Q_{Ext,i} \\ C_i Q_{Ext,i} \end{bmatrix}, \tag{B.2e}$$

where,

- $\underline{\underline{\boldsymbol{T}}}_i$ is $M/2 \times M/2$ matrix that represents the contribution of the surface to surface transmission of uncollided flux to the interface angular flux.
- M is the number of discrete directions set by the S_N quadrature.
- $\boldsymbol{E}_{i,L/R}$ and $\tilde{\boldsymbol{E}}_{i,L/R}$ are $M/2 \times 1$ vectors that represent the contribution of the escape of the collided to the interface angular flux, from within group scattering and from all other collisions, respectively.
- $\boldsymbol{I}_i$ is $1 \times M/2$ vector that represents the contribution of the incident uncollided flux to the average flux.
- $\boldsymbol{\Psi}_{i,L/R}$ and $\bar{\phi}_i$, are the outgoing angular flux on the left/right surface and the average scalar of flux of cell i, respectively.
- $Q_{Ext,i}$, represents the combination of the sources from fission and scattering from external groups.

Furthermore, the matrices and vectors of Eqs. (B.2a) to (B.2e) are defined as,

$$\underline{\underline{\boldsymbol{T}}}_i = \begin{bmatrix} T_{1,i} & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & T_{M/2,i} \end{bmatrix},$$

$$\boldsymbol{E}_i = \begin{bmatrix} \Sigma_{sc,i} E_{1,i} \\ \vdots \\ \Sigma_{sc,i} E_{M/2,i} \end{bmatrix},$$

$$\tilde{\boldsymbol{E}}_i = \begin{bmatrix} E_{1,i} \\ \vdots \\ E_{M/2,i} \end{bmatrix},$$

$$\boldsymbol{I}_i = \begin{bmatrix} I_{1,i} & \dots & I_{M/2,} \end{bmatrix},$$

$$T_{m,i} = e^{-\Sigma_i l_{m,i}},$$

$$E_{m,i} = \frac{[1 - T_{m,i}]}{\Sigma_i},$$

$$I_{m,i} = \frac{w_m E_{m,i}}{l_{m,i}},$$

$$C_i = \sum_{m=1}^{M} \frac{I_{m,i}}{\Sigma_i},$$

where $l_{m,i}$ is the chord length of cell i along direction m.

We can distinguish the Standard (Std) and GJ iteration schemes, by specifying Eq. (B.1) as

$$-\underline{\underline{\boldsymbol{L}}}_{i,i-1}\ \boldsymbol{\Psi}_{i-1}^{(n+1)} + \boldsymbol{\Psi}_i^{(n+1)} - \underline{\underline{\boldsymbol{L}}}_{i,i+1}\ \boldsymbol{\Psi}_{i+1}^{(n+1)} = \underline{\underline{\boldsymbol{L}}}_{i,i} \boldsymbol{\Psi}_i^{(n)} + \boldsymbol{Q}_i,$$

for the standard scheme and

$$\boldsymbol{\Psi}_i^{(n+1)} = \underline{\underline{\boldsymbol{L}}}_{i,i-1}\ \boldsymbol{\Psi}_{i-1}^{(n)} + \underline{\underline{\boldsymbol{L}}}_{i,i}\ \boldsymbol{\Psi}_i^{(n)} + \underline{\underline{\boldsymbol{L}}}_{i,i+1}\ \boldsymbol{\Psi}_{i+1}^{(n)} + \boldsymbol{Q}_i,$$

for the pure GJ scheme. Moreover, the standard MOC transport iteration scheme for the global problem can be defined as

$$\boldsymbol{\Psi}^{(n+1)} = \underline{\underline{\boldsymbol{L}}}_{Std} \boldsymbol{\Psi}^{(n)} + \boldsymbol{Q}_{Std},$$
$$\underline{\underline{\boldsymbol{L}}}_{Std} = (\underline{\underline{\hat{\boldsymbol{I}}}} - \underline{\underline{\boldsymbol{L}}}_l - \underline{\underline{\boldsymbol{L}}}_u)^{-1} \underline{\underline{\boldsymbol{L}}}_d,$$
$$\boldsymbol{Q}_{Std} = (\underline{\underline{\hat{\boldsymbol{I}}}} - \underline{\underline{\boldsymbol{L}}}_l - \underline{\underline{\boldsymbol{L}}}_u)^{-1}\ \boldsymbol{Q},$$

where the $\underline{\underline{\boldsymbol{L}}}_d$ is the block-diagonal matrix composed by the matrices $\underline{\underline{\boldsymbol{L}}}_{i,i}$, while $\underline{\underline{\boldsymbol{L}}}_l$ and $\underline{\underline{\boldsymbol{L}}}_u$ respectively, are the lower and upper triangular matrices composed by the block matrices $\underline{\underline{\boldsymbol{L}}}_{i,i-1}$ and $\underline{\underline{\boldsymbol{L}}}_{i,i+1}$. The global transport matrix for the GJ scheme can be written as,

$$\boldsymbol{\Psi}^{(n+1)} = \underline{\underline{\boldsymbol{L}}}_{GJ} \boldsymbol{\Psi}^{(n)} + \boldsymbol{Q},$$
$$\underline{\underline{\boldsymbol{L}}}\big|_{GJ} = \underline{\underline{\boldsymbol{L}}}_l + \underline{\underline{\boldsymbol{L}}}_d + \underline{\underline{\boldsymbol{L}}}_u.$$

B.2 Boundary conditions

The boundary conditions of SVRM and RMA can be expressed in a general form to be

$$J^-_{1,L} = v_{L\leftarrow L} J^+_{1,L} + v_{L\leftarrow R} J^+_{I,R} \quad \text{for } i = 1, \tag{B.3}$$

$$J^-_{I,R} = v_{R\leftarrow L} J^+_{1,L} + v_{R\leftarrow R} J^+_{I,R} \quad \text{for } i = I. \tag{B.4}$$

Here, the currents $J^-_{1,L}$ and $J^-_{I,R}$ indicate the incoming boundary current on the left and on the right sides respectively. The albedo coefficients $v_{L\leftarrow L}$, $v_{L\leftarrow R}$, $v_{R\leftarrow L}$, and $v_{R\leftarrow R}$ are defined by Table (B.1). $\tilde{\alpha}_{L/R}$ is defined as,

$$\tilde{\alpha}_{L/R} = \sum_{m\in M^-_{L/R}} |\Omega_m \cdot \hat{x}| \sum_{m'\in M^+_{L/R}} \alpha_{L/R}(\hat{\Omega}_{m'} \to \hat{\Omega}_m) \frac{\psi^+_{m',1/I,L/R}}{J^+_{1/I,L/R}},$$

where $\alpha_{L/R}(\hat{\Omega}_{m'} \to \hat{\Omega}_m)$ is the reflection coefficient at the left/right boundary.

	Reflective	Periodic	Vacuum	Albedo
$v_{L\leftarrow L} =$	1	0	0	$\tilde{\alpha}_L$
$v_{L\leftarrow R} =$	0	1	0	0
$v_{R\leftarrow L} =$	0	1	0	0
$v_{R\leftarrow R} =$	1	0	0	$\tilde{\alpha}_R$

Table B.1 – Values of the albedo coefficients at the left and right boundaries.

When the above boundary conditions are applied to an acceleration matrix, the contribution to the can be defined by the matrix $\underline{\underline{\tilde{\boldsymbol{B}}}}$,

$$\underline{\underline{\tilde{\boldsymbol{B}}}} = \begin{bmatrix} \underline{\underline{\tilde{\boldsymbol{A}}}}_{1,0}\underline{\underline{\tilde{\boldsymbol{B}}}}_{L,1} & 0 & \dots & 0 & \underline{\underline{\tilde{\boldsymbol{A}}}}_{1,0}\underline{\underline{\tilde{\boldsymbol{B}}}}_{L,I} \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 \\ \underline{\underline{\tilde{\boldsymbol{A}}}}_{I,I+1}\underline{\underline{\tilde{\boldsymbol{B}}}}_{R,1} & 0 & \dots & 0 & \underline{\underline{\tilde{\boldsymbol{A}}}}_{I,I+1}\underline{\underline{\tilde{\boldsymbol{B}}}}_{R,I} \end{bmatrix},$$

where

$$\underline{\underline{\tilde{\boldsymbol{B}}}}_{L,1} = \begin{bmatrix} v_{L\leftarrow L} & 0 \\ 0 & 0 \end{bmatrix},$$

$$\underline{\underline{\tilde{\boldsymbol{B}}}}_{L,I} = \begin{bmatrix} 0 & v_{L\leftarrow R} \\ 0 & 0 \end{bmatrix},$$

$$\underline{\underline{\tilde{\boldsymbol{B}}}}_{R,1} = \begin{bmatrix} 0 & 0 \\ v_{R\leftarrow L} & 0 \end{bmatrix},$$

$$\underline{\underline{\tilde{\boldsymbol{B}}}}_{R,I} = \begin{bmatrix} 0 & 0 \\ 0 & v_{R\leftarrow R} \end{bmatrix},$$

where $\underline{\underline{\tilde{\boldsymbol{B}}}}$ and $\underline{\underline{\tilde{\boldsymbol{A}}}}$ were first introduced by Eq. (3.18).

B.3 SVRM

The SVRM operator, $\underline{\underline{\tilde{\boldsymbol{A}}}}$, can be written as a block matrix of the form,

$$\underline{\underline{\tilde{\boldsymbol{A}}}} = \begin{bmatrix} \underline{\underline{\tilde{\boldsymbol{A}}}}_{1,1} & \underline{\underline{\tilde{\boldsymbol{A}}}}_{1,2} & 0 & \dots & 0 \\ \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i-1} & \underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i} & \underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i+1} & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ 0 & \dots & 0 & \underline{\underline{\tilde{\boldsymbol{A}}}}_{I,I-1} & \underline{\underline{\tilde{\boldsymbol{A}}}}_{I,I} \end{bmatrix}.$$

More specifically for node i, the equations take a 2x2 matrix form of the type

$$\underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i-1}\ \boldsymbol{J}_{i-1} + \underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i}\ \boldsymbol{J}_i + \underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i+1}\ \boldsymbol{J}_{i+1} = \boldsymbol{Q}_{Ac,i} \quad \text{when } i \neq 1 \text{ and } i \neq I, \tag{B.5}$$

where

$$\boldsymbol{J}_i = \begin{bmatrix} J^+_{i,L} \\ J^+_{i,R} \end{bmatrix}.$$

By substituting the closure relations, Eqs. (3.10a)-(3.12), and Eq. (3.17) into Eq.(B.5), we get an expression for the matrices $\underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i-1}$, $\underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i}$, $\underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i+1}$ and $\boldsymbol{Q}_{Ac,i}$ for $i = 1, \dots, I$,

$$\underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i-1} = \begin{bmatrix} 0 & -1 \\ 0 & \tilde{\eta}_i \frac{\Delta_i}{2} - (1-\tilde{\eta}_i)(\widetilde{\delta}_i \tau_{i,r} \tilde{\beta}_{i,L} + \tilde{\alpha}_{i-1,R}) \end{bmatrix},$$

$$\underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i} = \begin{bmatrix} 1 + \tilde{\beta}_i \tau_{i,r} & 1 + \tilde{\beta}_i \tau_{i,r} \\ [-(1-\tilde{\eta}_i)(\frac{\Delta_i}{2} + \widetilde{\delta}_i \tau_{i,r} \tilde{\beta}_{i,L})] & [\tilde{\eta}_i (\frac{\Delta_i}{2} + \widetilde{\delta}_i \tau_{i,r} \tilde{\beta}_{i,R})] \end{bmatrix},$$

$$\underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i+1} = \begin{bmatrix} -1 & 0 \\ \tilde{\eta}_i (\widetilde{\delta}_i \tau_{i,r} \tilde{\beta}_{i,R} + \tilde{\alpha}_{i+1,L}) - (1-\tilde{\eta}_i)\frac{\Delta_i}{2} & 0 \end{bmatrix},$$

and finally,

$$\boldsymbol{Q}_{Ac,i} = \begin{bmatrix} \Delta_i Q_{Ext,i} \\ \Delta_i \widetilde{\delta}_i (\tilde{\eta} Q_{Ext,i,R} - (1-\tilde{\eta}) Q_{Ext,i,L}) \end{bmatrix}.$$

The definitions of the of the terms used in $\underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i-1}$, $\underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i}$, $\underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i+1}$ and $\boldsymbol{Q}_{Ac,i}$ are contained in Section 3.1.

B.3.1 The discrete-to-moment operator

The acceleration space needs the transport angular moments. The mapping from the transport angular flux to the transport angular moments is given by the so-called discrete-to-moment operator. In the case of SVRM, the discrete-to-moment operator maps to both the volume and interface angular moments. The angular moments that define the

acceleration space can be defined as

$$
\begin{aligned}
J^+_{i,L} &= \sum_{m\in M^+_L} w_m|\mu_m|\psi^+_{m,i,L},\\
J^+_{i,R} &= \sum_{m\in M^+_R} w_m|\mu_m|\psi^+_{m,i,R},\\
\bar{J}_i &= \sum_{m\in M^+_R} |\mu_m| I_{m,i}\psi^+_{m,i-1,R} - \sum_{m\in M^+_L} |\mu_m| I_{m,i}\psi^+_{m,i+1,L},\\
\phi^+_{i,L} &= \sum_{m\in M^+_L} w_m\psi^+_{m,i,L},\\
\phi^+_{i,R} &= \sum_{m\in M^+_R} w_m\psi^+_{m,i,R},\\
\mu_m &= \hat{\Omega}_m\cdot\hat{x}
\end{aligned}
$$

where $\phi^+_{i,L/R}$ represents the outgoing partial interface flux and $M^+_{L/R}$ represents the set of outgoing directions at the left/right surface. By defining the vector

$$
\mathbf{\Phi}_i = \begin{bmatrix} J^+_{i,L}\\ J^+_{i,R}\\ J_i\\ \phi^+_{i,L}\\ \phi^+_{i,R}\\ \phi_i \end{bmatrix}
$$

we can write the discrete-to-moment operator as

$$
\mathbf{\Phi} = \underline{\underline{\boldsymbol{K}}}\ \mathbf{\Psi}. \tag{B.6}
$$

In the above equation, $\underline{\underline{\boldsymbol{K}}}$ is the block-rectangular matrix

$$
\underline{\underline{\boldsymbol{K}}} = \begin{bmatrix}
\underline{\underline{\boldsymbol{K}}}_{1,1} & \underline{\underline{\boldsymbol{K}}}_{1,2} & 0 & \dots & 0\\
0 & \ddots & \ddots & \ddots & \vdots\\
0 & \underline{\underline{\boldsymbol{K}}}_{i,i-1} & \underline{\underline{\boldsymbol{K}}}_{i,i} & \underline{\underline{\boldsymbol{K}}}_{i,i+1} & 0\\
\vdots & \ddots & \ddots & \ddots & 0\\
0 & \dots & 0 & \underline{\underline{\boldsymbol{K}}}_{I-1,I} & \underline{\underline{\boldsymbol{K}}}_{I,I}
\end{bmatrix},
$$

where the minors $\underline{\underline{\boldsymbol{K}}}_{i,i-1}$, $\underline{\underline{\boldsymbol{K}}}_{i,i}$ and $\underline{\underline{\boldsymbol{K}}}_{i,i+1}$ satisfy the equation

$$
\mathbf{\Phi}_i = \underline{\underline{\boldsymbol{K}}}_{i,i-1}\mathbf{\Psi}_{i-1} + \underline{\underline{\boldsymbol{K}}}_{i,i}\mathbf{\Psi}_i + \underline{\underline{\boldsymbol{K}}}_{i,i+1}\mathbf{\Psi}_{i+1},
$$

such that

$$\underline{\underline{\boldsymbol{K}}}_{i,i} = \begin{bmatrix} \underline{\underline{\boldsymbol{w}\boldsymbol{\mu}}} & 0 & 0 \\ 0 & \underline{\underline{\boldsymbol{w}\boldsymbol{\mu}}} & 0 \\ 0 & 0 & 0 \\ \boldsymbol{w} & 0 & 0 \\ 0 & \boldsymbol{w} & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$\underline{\underline{\boldsymbol{K}}}_{i,i-1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \underline{\underline{\boldsymbol{I}_i\boldsymbol{\mu}}} & 0 \\ 0 & \boldsymbol{w} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$\underline{\underline{\boldsymbol{K}}}_{i,i+1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \underline{\underline{\boldsymbol{I}_i\boldsymbol{\mu}}} & 0 \\ 0 & 0 & 0 \\ \boldsymbol{w} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$\underline{\underline{\boldsymbol{\mu}}} = \begin{bmatrix} |\mu_1| & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & |\mu_{M/2}| \end{bmatrix},$$

$$\boldsymbol{w} = \begin{bmatrix} w_1 & \dots & w_{M/2} \end{bmatrix}.$$

B.3.2 The prolongation operator

Eq. (3.22) can be cast in a matrix formalism,

$$\boldsymbol{\Psi}^{(n+1)} = \underline{\underline{\tilde{\boldsymbol{P}}}}\ \boldsymbol{J}^{(n+1)},$$

where $\underline{\underline{\tilde{\boldsymbol{P}}}}$ is a rectangular diagonal-by-block matrix composed of matrix $\underline{\underline{\tilde{\boldsymbol{P}}}}_i$ containing the coefficients of Eqs. (3.21), (3.22) for each region i,

$$\underline{\underline{\tilde{\boldsymbol{P}}}}_i = \begin{bmatrix} \frac{\boldsymbol{\Psi}^{+,(n+1/2)}_{i,L}}{J^{+,(n+1/2)}_{i,L}} & 0 \\ 0 & \frac{\boldsymbol{\Psi}^{+,(n+1/2)}_{i,R}}{J^{+,(n+1/2)}_{i,R}} \\ \tilde{\beta}_i & \tilde{\beta}_i \end{bmatrix}, \tag{B.7}$$

$$\boldsymbol{\Psi}^{+,(n+1/2)}_{i,L/R} = \begin{bmatrix} \vdots \\ \psi^{+,(n+1/2)}_{i,L/R,m} \\ \vdots \end{bmatrix}, m \in M^+_{L/R}.$$

B.3.3 Jacobian

This section has been dedicated to the tedious derivation of the partial derivatives required to formulate the Jacobian defined by Eq. (2.74b). Since $\frac{\partial \boldsymbol{\Phi}^{(n+1/2)}}{\partial \boldsymbol{\Psi}^{(n)}} = KT$ is the transport contribution to the Jacobian, the acceleration contribution, can be rewritten as

$$\frac{\partial \boldsymbol{\Psi}^{(n+1)}}{\partial \boldsymbol{\Phi}^{(n+1/2)}} = \underline{\underline{\tilde{\boldsymbol{P}}}} \, \frac{\partial \boldsymbol{J}^{(n+1)}}{\partial \boldsymbol{\Phi}^{(n+1/2)}} + \frac{\partial (\underline{\underline{\tilde{\boldsymbol{P}}}} \, \boldsymbol{J}^{(n+1)})}{\partial \boldsymbol{\Phi}^{(n+1/2)}}. \tag{B.8}$$

The matrix $\frac{\partial \boldsymbol{J}^{(n+1)}}{\partial \boldsymbol{\Phi}^{(n+1/2)}}$ is obtained by deriving of the SVRM equations, obtaining

$$\frac{\partial \boldsymbol{J}^{(n+1)}}{\partial \boldsymbol{\Phi}^{(n+1/2)}} = \underline{\underline{\tilde{\boldsymbol{A}}}}^{-1} \, \frac{\partial [\underline{\underline{\tilde{\boldsymbol{A}}}} \, \boldsymbol{J}^{(n+1)}]}{\partial \boldsymbol{\Phi}^{(n+1/2)}}. \tag{B.9}$$

We can explicitly compute the $\frac{\partial [\underline{\underline{\tilde{\boldsymbol{A}}}} \, \boldsymbol{J}^{(n+1)}]}{\partial \boldsymbol{\Phi}^{(n+1/2)}}$ matrix as a composition of partial derivatives

$$\frac{\partial [\underline{\underline{\tilde{\boldsymbol{A}}}} \, \boldsymbol{J}^{(n+1)}]}{\partial \boldsymbol{\Phi}^{(n+1/2)}} = \sum_{x \in (\tilde{\alpha}_{i-1,R}, \tilde{\alpha}_{i+1,L}, \tilde{\beta}_{i,R}, \tilde{\beta}_{i,L}, \tilde{\beta}_i, \tilde{\eta}_i, \tilde{\gamma}_R, \tilde{\gamma}_L)_{i=1,I}} \frac{\partial [\underline{\underline{\tilde{\boldsymbol{A}}}} \, \boldsymbol{J}^{(n+1)}]}{\partial x} \otimes \frac{\partial x}{\partial \boldsymbol{\Phi}^{(n+1/2)}}, \tag{B.10}$$

note that in Eq. (B.10) $\boldsymbol{J}^{(n+1)}$ is not derived. The non-zero derivatives that appears in (B.10) are

$$\frac{\partial \underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i+1}}{\partial \tilde{\alpha}_{i+1,L}} = \begin{bmatrix} 0 & 0 \\ \tilde{\eta}_i & 0 \end{bmatrix},$$

$$\frac{\partial \underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i-1}}{\partial \tilde{\alpha}_{i-1,R}} = \begin{bmatrix} 0 & 0 \\ 0 & -(1-\tilde{\eta}_i) \end{bmatrix},$$

$$\frac{\partial \underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i+1}}{\partial \tilde{\beta}_{i,R}} = \begin{bmatrix} 0 & 0 \\ \tilde{\eta}_i \widetilde{\delta}_i \tau_{i,r} & 0 \end{bmatrix},$$

$$\frac{\partial \underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i}}{\partial \tilde{\beta}_{i,R}} = \begin{bmatrix} 0 & 0 \\ 0 & \tilde{\eta}_i \widetilde{\delta}_i \tau_{i,r} \end{bmatrix},$$

$$\frac{\partial \underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i-1}}{\partial \tilde{\beta}_{i,L}} = \begin{bmatrix} 0 & 0 \\ 0 & -(1-\tilde{\eta}_i) \widetilde{\delta}_i \tau_{i,r} \end{bmatrix},$$

$$\frac{\partial \underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i}}{\partial \tilde{\beta}_{i,L}} = \begin{bmatrix} 0 & 0 \\ -(1-\tilde{\eta}_i) \widetilde{\delta}_i \tau_{i,r} & 0 \end{bmatrix},$$

$$\frac{\partial \underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i}}{\partial \tilde{\beta}_i} = \begin{bmatrix} \tau_{i,r} & \tau_{i,r} \\ 0 & 0 \end{bmatrix},$$

$$\frac{\partial \underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i}}{\partial \tilde{\eta}_i} = \begin{bmatrix} 0 & 0 \\ (\frac{\Delta_i}{2} + \widetilde{\delta}_i \tau_{i,r} \tilde{\beta}_{i,L}) & (\frac{\Delta_i}{2} + \widetilde{\delta}_i \tau_{i,r} \tilde{\beta}_{i,R}) \end{bmatrix},$$

$$\frac{\partial \underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i-1}}{\partial \tilde{\eta}_i} = \begin{bmatrix} 0 & 0 \\ 0 & \tilde{\alpha}_{i-1,R} + \frac{\Delta_i}{2} + \widetilde{\delta}_i \tau_{i,r} \tilde{\beta}_{i,L} \end{bmatrix} \text{ and}$$

$$\frac{\partial \underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i+1}}{\partial \tilde{\eta}_i} = \begin{bmatrix} 0 & 0 \\ \tilde{\alpha}_{i+1,L} + \frac{\Delta_i}{2} + \widetilde{\delta}_i \tau_{i,r} \tilde{\beta}_{i,R} & 0 \end{bmatrix},$$

In Eq. (B.10), the derivatives of the non-linear coefficients $(\tilde{\alpha}_{i-1,R}, \tilde{\alpha}_{i+1,L}, \tilde{\beta}_{i,R}, \tilde{\beta}_{i,L}, \tilde{\beta}_i, \tilde{\eta}_i, \tilde{\gamma}_R, \tilde{\gamma}_L)_{i=1,I}$ with respect to the angular flux moments $\mathbf{\Phi}_i^{(n+1/2)}$ are

$$\frac{\partial \tilde{\alpha}_{i-1,R}}{\partial \mathbf{\Phi}_{i-1}^{(n+1/2)}} = \frac{1}{J_{i-1,R}^{(n+1/2)}}[0,\ -\tilde{\alpha}_{i-1,R},\ 0,\ 0,\ 0,\ 0],$$

$$\frac{\partial \tilde{\alpha}_{i+1,L}}{\partial \mathbf{\Phi}_{i+1}^{(n+1/2)}} = \frac{1}{J_{i+1,L}^{(n+1/2)}}[-\tilde{\alpha}_{i+1,L},\ 0,\ 0,\ 0,\ 0,\ 0],$$

$$\frac{\partial \tilde{\beta}_{i,R}}{\partial \mathbf{\Phi}_{i}^{(n+1/2)}} = \frac{1}{(J_{i,R}^{+} + J_{i+1,L}^{+,})^{(n+1/2)}}[0,\ -\tilde{\beta}_{i,R},\ 0,\ 0,\ 1,\ 0],$$

$$\frac{\partial \tilde{\beta}_{i,R}}{\partial \mathbf{\Phi}_{i+1}^{(n+1/2)}} = \frac{1}{(J_{i,R}^{+} + J_{i+1,L}^{+,})^{(n+1/2)}}[-\tilde{\beta}_{i,R},\ 0,\ 0,\ 0,\ 0,\ 0],$$

$$\frac{\partial \tilde{\beta}_{i,L}}{\partial \mathbf{\Phi}_{i}^{(n+1/2)}} = \frac{1}{(J_{i,L}^{+} + J_{i-1,R}^{+,})^{(n+1/2)}}[-\tilde{\beta}_{i,L},\ 0,\ 0,\ 1,\ 0,\ 0],$$

$$\frac{\partial \tilde{\beta}_{i,L}}{\partial \mathbf{\Phi}_{i-1}^{(n+1/2)}} = \frac{1}{(J_{i,L}^{+} + J_{i-1,R}^{+,})^{(n+1/2)}}[0,\ -\tilde{\beta}_{i,L},\ 0,\ 0,\ 0,\ 0]$$

$$\frac{\partial \tilde{\beta}_{i}}{\partial \mathbf{\Phi}_{i}^{(n+1/2)}} = \frac{1}{(J_{R}^{+} + J_{L}^{+})^{(n+1/2)}}[-\tilde{\beta}_{i},\ -\tilde{\beta}_{i},\ 0,\ 0,\ 0,\ 0],$$

$$\frac{\partial \tilde{\eta}_{i}}{\partial \mathbf{\Phi}^{(n+1/2)}} = -\frac{1}{(P_i + N_i)}\left[(1-\tilde{\eta}_i)\frac{\partial P_i}{\partial \mathbf{\Phi}^{(n+1/2)}} + \tilde{\eta}_i \frac{\partial N_i}{\partial \mathbf{\Phi}^{(n+1/2)}}\right],$$

where the partial derivatives for P_i and N_i are

$$\frac{\partial P_i}{\partial \mathbf{\Phi}_{i}^{(n+1/2)}} = [0,\ (\frac{\Delta_i}{2} + \widetilde{\delta}_i \tau_{i,r} \tilde{\beta}_{i,R}),\ 0,\ 0,\ 0,\ 0],$$

$$\frac{\partial P_i}{\partial \mathbf{\Phi}_{i-1}^{(n+1/2)}} = [0,\ \frac{\Delta_i}{2},\ 0,\ 0,\ 0,\ 0],\ \text{and}$$

$$\frac{\partial P_i}{\partial \mathbf{\Phi}_{i+1}^{(n+1/2)}} = [\tilde{\alpha}_{i+1,L} + \widetilde{\delta}_i \tau_{i,r} \tilde{\beta}_{i,R},\ 0,\ 0,\ 0,\ 0,\ 0],$$

$$\frac{\partial N_i}{\partial \mathbf{\Phi}_{i}^{(n+1/2)}} = [(\frac{\Delta_i}{2} + \widetilde{\delta}_i \tau_{i,r} \tilde{\beta}_{i,L}),\ 0,\ 0,\ 0,\ 0,\ 0],$$

$$\frac{\partial N_i}{\partial \mathbf{\Phi}_{i-1}^{(n+1/2)}} = [0,\ \tilde{\alpha}_{i-1,R} + \widetilde{\delta}_i \tau_{i,r} \tilde{\beta}_{i,L},\ 0,\ 0,\ 0,\ 0]\ \text{and}$$

$$\frac{\partial N_i}{\partial \mathbf{\Phi}_{i+1}^{(n+1/2)}} = [\frac{\Delta_i}{2},\ 0,\ 0,\ 0,\ 0,\ 0].$$

Finally, for the reconstruction of the Jacobian, we must specify the second term of Eq. (B.8) which is characterized by the derivatives of the update operator $\underline{\underline{\tilde{\boldsymbol{P}}}}$, expressed in Eq. (B.7). This term is composed by the following non-zero partial derivatives

$$\frac{\partial \underline{\underline{\tilde{\boldsymbol{P}}}}_i}{\partial J_{i,L}^{+,(n+1/2)}} = \begin{bmatrix} -\frac{\psi_{L,2}^{(n+1/2)}}{(J_L^{+,(n+1/2)})^2} & 0 \\ 0 & 0 \\ \frac{\tilde{\beta}}{(J_L^{+}+J_R^{+})^{(n+1/2)}} & \frac{\tilde{\beta}}{(J_L^{+}+J_R^{+})^{(n+1/2)}} \end{bmatrix},$$

$$\frac{\partial \underline{\underline{\tilde{\boldsymbol{P}}}}_i}{\partial J_{i,R}^{+,(n+1/2)}} = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{\psi_{R,1}^{(n+1/2)}}{(J_R^{+,(n+1/2)})^2} \\ \frac{\tilde{\beta}}{(J_L^+ + J_R^+)^{(n+1/2)}} & \frac{\tilde{\beta}}{(J_L^+ + J_R^+)^{(n+1/2)}} \end{bmatrix},$$

$$\frac{\partial \underline{\underline{\tilde{\boldsymbol{P}}}}_i}{\partial \bar{\phi}_i^{(n+1/2)}} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{(J_L^+ + J_R^+)^{(n+1/2)}} & \frac{1}{(J_L^+ + J_R^+)^{(n+1/2)}} \end{bmatrix}. \tag{B.11}$$

The Jacobian matrix can be computed by reassembling by combining Eqs. (B.8), (B.6), (B.1) to get Eq. (2.74a). In particular, the derivatives contained by Eq. (B.8) are computed by combining Eqs. (B.9) to (B.11).

Note that the explicit computation of the Jacobian matrix allows the analysis of the convergence of the method far and close to the fixed point. Moreover, unlike Fourier analysis, the study of the spectral radius of the Jacobian allows us to analyze heterogeneous non-periodic geometries.

B.4 RMA

In 1D, the RMA operator, $\underline{\underline{\tilde{\boldsymbol{A}}}}$, can expressed as a block matrix composed of 2x2 matrices such that,

$$\underline{\underline{\tilde{\boldsymbol{A}}}} = \begin{bmatrix} \underline{\underline{\tilde{\boldsymbol{A}}}}_{1,1} & \underline{\underline{\tilde{\boldsymbol{A}}}}_{1,2} & 0 & \dots & 0 \\ \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i-1} & \underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i} & \underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i+1} & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ 0 & \dots & 0 & \underline{\underline{\tilde{\boldsymbol{A}}}}_{I,I-1} & \underline{\underline{\tilde{\boldsymbol{A}}}}_{I,I} \end{bmatrix},$$

where

$$\underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i-1}\ \boldsymbol{J}_{i-1} + \underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i}\ \boldsymbol{J}_i + \underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i+1}\ \boldsymbol{J}_{i+1} = \boldsymbol{Q}_{Ac,i} \quad \text{when } i \neq 1 \text{ and } i \neq I. \tag{B.12}$$

By substituting the explicit RMA closure relations into Eq. (B.12), we get the following expression for the 2x2 matrices $\underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i-1}$, $\underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i}$, $\underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i+1}$ and $\boldsymbol{Q}_{Ac,i}$,

$$\underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i-1} = \begin{bmatrix} 0 & -\Sigma_{sc,0,i}\tilde{E}_{i,L}\tilde{F}_{i,L}^- \\ 0 & -\tilde{T}_{i,R,R} - \Sigma_{sc,0,i}\tilde{E}_{i,R}\tilde{F}_{i,L}^- \end{bmatrix},$$

$$\underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i} = \begin{bmatrix} 1 - \Sigma_{sc,0,i}\tilde{E}_{i,L}\tilde{F}_{i,L}^+ & -\Sigma_{sc,0,i}\tilde{E}_{i,L}\tilde{F}_{i,R}^+ \\ -\Sigma_{sc,0,i}\tilde{E}_{i,R}\tilde{F}_{i,L}^+ & 1 - \Sigma_{sc,0,i}\tilde{E}_{i,R}\tilde{F}_{i,R}^+ \end{bmatrix},$$

$$\underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i+1} = \begin{bmatrix} -\tilde{T}_{i,L,L} - \Sigma_{sc,0,i}\tilde{E}_{i,L}\tilde{F}_{i,R}^- & 0 \\ -\Sigma_{sc,0,i}\tilde{E}_{i,R}\tilde{F}_{i,R}^- & 0 \end{bmatrix},$$

$$\boldsymbol{Q}_{Ac,i} = \begin{bmatrix} \tilde{E}_{i,L}[1 + \tilde{F}_i]Q_{Ext,0,0,i} \\ \tilde{E}_{i,R}[1 + \tilde{F}_i]Q_{Ext,0,0,i} \end{bmatrix},$$

where $\tilde{F}^{\pm}_{i,L/R}$ and $\tilde{F}_i$ are the contributions of $J^{\pm}_{i,L/R}$ and $Q_{Ext,0,0,i}$ to $\bar{\phi}_{0,0,i}$. For E-RMA,

$$\begin{aligned}
\tilde{F}^{-}_{i,L/R} &= \frac{\tilde{I}_{i,L/R}}{1-\Sigma_{sc,0,i}\tilde{C}_i}, \\
\tilde{F}^{+}_{i,L/R} &= 0, \\
\tilde{F}_i &= \frac{\tilde{C}_i}{1-\Sigma_{sc,0,i}\tilde{C}_i},
\end{aligned}$$

and for B-RMA

$$\begin{aligned}
\tilde{F}^{\pm}_{i,L/R} &= \mp\frac{A_{i,L/R}}{[\Sigma_i-\Sigma_{sc,0,i}]V_i}, \\
\tilde{F}_i &= [\Sigma_i-\Sigma_{sc,0,i}]^{-1}.
\end{aligned}$$

The definitions of the of the terms used in $\underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i-1}$, $\underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i}$, $\underline{\underline{\tilde{\boldsymbol{A}}}}_{i,i+1}$ and $\boldsymbol{Q}_{Ac,i}$ are contained in Section 3.2.

B.4.1 The discrete-to-moment operator and the prolongation operator

In the case of RMA, the discrete-to-moment operator maps only to the interface angular moments. The angular moments that define the acceleration space can be defined as

$$\begin{aligned}
J^{+}_{i,L} &= \sum_{m\in M^{+}_{L}} w_m|\mu_m|\psi_{i,L,m}, \\
J^{+}_{i,R} &= \sum_{m\in M^{+}_{R}} w_m|\mu_m|\psi_{i,R,m}, \\
\mu &= \hat{\Omega}_m\cdot\hat{x}
\end{aligned}$$

By defining the vector

$$\boldsymbol{\Phi}_i = \begin{bmatrix} J^{+}_{i,L} \\ J^{+}_{i,R} \end{bmatrix}$$

we can write the discrete-to-moment operator as

$$\boldsymbol{\Phi} = \underline{\underline{\boldsymbol{K}}}\ \boldsymbol{\Psi}.$$

In the above equation, $\underline{\underline{\boldsymbol{K}}}$ is the block-rectangular matrix

$$\underline{\underline{\boldsymbol{K}}} = \begin{bmatrix} \underline{\underline{\boldsymbol{K}}}_{1,1} & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \underline{\underline{\boldsymbol{K}}}_{I,I} \end{bmatrix},$$

where $\underline{\underline{\boldsymbol{K}}}_{i,i}$ satisfies the equation

$$\boldsymbol{\Phi}_i = \underline{\underline{\boldsymbol{K}}}_{i,i}\boldsymbol{\Psi}_i,$$

such that

$$\underline{\underline{\boldsymbol{K}}}_{i,i} = \begin{bmatrix} \boldsymbol{w}\underline{\underline{\boldsymbol{\mu}}} & 0 \\ 0 & \boldsymbol{w}\underline{\underline{\boldsymbol{\mu}}} \end{bmatrix},$$

$$\underline{\underline{\boldsymbol{\mu}}} = \begin{bmatrix} |\mu_1| & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & |\mu_{M/2}| \end{bmatrix},$$

$$\boldsymbol{w} = \begin{bmatrix} w_1 & \dots & w_{M/2} \end{bmatrix}.$$

B.4.2 The prolongation operator

Eqs. (3.39) and (3.39) can be cast in a matrix formalism,

$$\boldsymbol{\Psi}^{(n+1)} = \underline{\underline{\tilde{\boldsymbol{P}}}}\ \boldsymbol{J}^{(n+1)} + \boldsymbol{Q}_P,$$

where $\underline{\underline{\tilde{\boldsymbol{P}}}}$ is defined by the RMA closure relations. $\underline{\underline{\tilde{\boldsymbol{P}}}}$ can be written as a block matrix composed of $\underline{\underline{\tilde{\boldsymbol{P}}}}_{i,i-1}$, $\underline{\underline{\tilde{\boldsymbol{P}}}}_{i}$ and $\underline{\underline{\tilde{\boldsymbol{P}}}}_{i,i+1}$ for each region i such that,

$$\boldsymbol{\Psi}_i^{(n+1)} = \underline{\underline{\tilde{\boldsymbol{P}}}}_{i,i-1}\ \boldsymbol{J}_{i-1}^{(n+1)} + \underline{\underline{\tilde{\boldsymbol{P}}}}_{i,i}\ \boldsymbol{J}_{i}^{(n+1)} + \underline{\underline{\tilde{\boldsymbol{P}}}}_{i,i+1}\ \boldsymbol{J}_{i+1}^{(n+1)} + \boldsymbol{Q}_{P,i},$$

$$\underline{\underline{\tilde{\boldsymbol{P}}}}_{i,i} = \begin{bmatrix} \tilde{\boldsymbol{h}}_{i,L}^{+,(n+1/2)} & 0 \\ 0 & \tilde{\boldsymbol{h}}_{i,R}^{+,(n+1/2)} \\ 0 & 0 \end{bmatrix},$$

$$\underline{\underline{\tilde{\boldsymbol{P}}}}_{i,i-1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & \tilde{F}_{i,R}^{-} \end{bmatrix},$$

$$\underline{\underline{\tilde{\boldsymbol{P}}}}_{i,i+1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \tilde{F}_{i,L}^{-} & 0 \end{bmatrix},$$

$$\boldsymbol{Q}_{P,i} = \begin{bmatrix} 0 \\ 0 \\ \tilde{F}_i Q_{Ext,0,0,i} \end{bmatrix},$$

$$\tilde{\boldsymbol{h}}_{i,L/R}^{+,(n+1/2)} = \begin{bmatrix} \vdots \\ \tilde{h}_{m,i,L/R}^{\pm,(n+1/2)} \\ \vdots \end{bmatrix}, m \in M_{L/R}^{+}.$$

B.5 Fourier transform

The Fourier transformation operators can be expressed in matrix form as,

$$
\begin{aligned}
\partial\boldsymbol{\Psi}_i &= \int_{-\infty}^{\infty} d\zeta\ \underline{\underline{\boldsymbol{F}}}_i(\zeta)\boldsymbol{\Xi}(\zeta), \\
\underline{\underline{\boldsymbol{F}}}_i(\zeta) &= \begin{bmatrix} \underline{\underline{\boldsymbol{F}}}_{i,L}(\zeta) & 0 & 0 \\ 0 & \underline{\underline{\boldsymbol{F}}}_{i,R}(\zeta) & 0 \\ 0 & 0 & \underline{\underline{\boldsymbol{F}}}_{i,C}(\zeta) \end{bmatrix}, \\
\underline{\underline{\boldsymbol{F}}}_{i,L}(\zeta) &= \begin{bmatrix} e^{j\frac{2\pi\zeta}{\Delta}(x_i-\frac{\Delta}{2})} & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & e^{j\frac{2\pi\zeta}{\Delta}(x_i-\frac{\Delta}{2})} \end{bmatrix}, \\
\underline{\underline{\boldsymbol{F}}}_{i,R}(\zeta) &= \begin{bmatrix} e^{j\frac{2\pi\zeta}{\Delta}(x_i+\frac{\Delta}{2})} & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & e^{j\frac{2\pi\zeta}{\Delta}(x_i+\frac{\Delta}{2})} \end{bmatrix}, \\
\underline{\underline{\boldsymbol{F}}}_{i,C}(\zeta) &= \begin{bmatrix} e^{j\frac{2\pi\zeta}{\Delta}x_i} & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & e^{j\frac{2\pi\zeta}{\Delta}x_i} \end{bmatrix}, \\
\boldsymbol{\Psi}_i &= \begin{bmatrix} \boldsymbol{\Psi}^+_{i,L} \\ \boldsymbol{\Psi}^+_{i,R} \\ \bar{\phi}_i \end{bmatrix},
\end{aligned}
$$

where $\boldsymbol{\Xi}(\zeta)$ is a vector of the amplitudes, Δ is the width of the repeated cell and x_i is the midpoint of region i. $\underline{\underline{\boldsymbol{F}}}_{i,L/R}(\zeta)$ contain the periodic functions for the outgoing angular flux and $\underline{\underline{\boldsymbol{F}}}_{i,C}(\zeta)$ contains the periodic functions for the scalar flux.

Bibliography

[1] Description of the cobalt computing cluster. http://www-ccrt.cea.fr/fr/moyen_de_calcul/index.htm. Accessed: 2019-06-07.

[2] Fundamentals of power reactors module one science —& engineering fundamentals. Technical report, Atomic Energy Control Board of Canada, 1995.

[3] M. L. Adams and E. W. Larsen. Fast iterative methods for discrete-ordinates particle transport calculations. *Progress in Nuclear Energy*, 40(1):3–159, 2002.

[4] M. L. Adams and E. W. Larsen. Fast iterative methods for discrete-ordinates particle transport calculations. *Progress in Nuclear Energy*, 40(1):3 – 159, 2002.

[5] H. Andersen. Categorization, anomalies and the discovery of nuclear fission. *Studies in History and Philosophy of Modern Physics*, 27:463–492, 1996.

[6] D. Y. Anistratov and Y. Y. Azmy. Iterative stability analysis of spatial domain decomposition based on block jacobi algorithm for the diamond-difference scheme. *Journal of Computational Physics*, 297:462 – 479, 2015.

[7] J. Aragones and C. Ahnert. Linear discontinuous finite difference formulation for synthetic coarse-mesh few-group diffusion calculations. *Nucl. Sci. Eng.; (United States)*, 94:4, 12 1986.

[8] J. Askew. A characteristics formulation of the neutron transport equation in complicated geometries. *INIS*, 1972.

[9] Y. Azmy. A novel algorithm for solving the multidimensional neutron transport equation on massively parallel architectures. Technical report, North Carolina StateUniversity.

[10] S. Balay, S. Abhyankar, M. F. Adams, and et al. PETSc users manual. Technical Report ANL-95/11 - Revision 3.11, Argonne National Laboratory, 2019.

[11] D. A. Brown and et al. Endf/b-viii.0: The 8th major release of the nuclear reaction data library with cielo-project cross sections, new standards and thermal scattering data. Technical report, International Atomic Energy Agency, 2018.

[12] L. Cai. Condensation and homogenization of cross sections for the deterministic transport codes with monte carlo method : Application to the gen iv fast neutron reactors. Technical report, Université Paris Sud - Paris XI, 2014.

[13] G. R. Cefus and E. W. Larsen. Stability analysis of coarse-mesh rebalance. *Nuclear Science and Engineering*, 105(1):31–39, 1990.

[14] N. Cho and C. Park. A comparison of coarse mesh rebalance and coarse mesh finite difference accelerations for the neutron transport calculations. *Nuclear Mathematical and Computational Sciences: A Century in Review*, 2003.

[15] R. Courant and D. Hilbert. *Methods of Mathematical Physics*, volume 2. Wiley-InterScience, 1962.

[16] D. E. Cullen, C. J. Clouse, R. Procassini, and R. C. Little. Static and dynamic criticality: Are they different? Technical report, Lawrence Livermore National Laboratory, 11 2003.

[17] N. Ellis and et al. Global warming of 1.5 °c. Technical report, IPCC, 10 2018.

[18] W. Ford, E. Masiello, C. Calvin, F. Févotte, and B. Lathuilière. A spatially variant rebalancing method for discrete-ordinates transport equation. *Annals of Nuclear Energy*, 133:589 – 602, 11 2019.

[19] F. Froehner, D. C. Larson, and et al. Cross-section fluctuations and self-shielding effects in the unresolved resonance. Technical report, Nuclear Energy Agency of the OECD (NEA), 1995.

[20] L. L. K. H. and N.-D. Armando. Thermal-hydraulics program in support of canadian scwr concept development. *Journal of Nuclear Engineering and Radiation Science*, 4, Dec 2017.

[21] A. Hébert. *Applied Reactor Physics*. Presses internationales Polytechnique, 2009.

[22] E. Hobson. *The Theory of Spherical and Ellipsoidal Harmonics*. Chelsea Pub. Co., 1955.

[23] G. Lee and N. Cho. 2d/1d fusion method solutions of the three-dimensional transport oecd benchmark problem c5g7 mox. *Progress in Nuclear Energy*, 48:410–423, 07 2006.

[24] L. G. LeSage. An overview of the argonne national laboratory fast critical experiments. Technical report, Argonne National Laboratory, 04 2001.

[25] E. E. Lewis and W. F. M. Jr. *Computational Methods of Neutron Transport*. Wiley-InterScience, 1984.

[26] E. Masiello. Analytical stability analysis of coarse-mesh finite difference method. *International Conference on the Physics of Reactors 2008, PHYSOR 08*, 1, 01 2008.

[27] E. Masiello, R. Clemente, and S. Santandrea. High-order method of characteristics for 2-d unstructured meshes. 05 2009.

[28] E. Masiello, R. Sanchez, and I. Zmijarevic. New numerical solution with the method of short characteristics for 2-d heterogeneous cartesian cells in the apollo2 code: Numerical analysis and tests. *Nuclear Science and Engineering*, 161:257–278, 03 2009.

[29] T. Mazumdar and S. Degweker. Solution of neutron transport equation by method of characteristics. *Annals of Nuclear Energy*, 77:522–535, 03 2015.

[30] K. Okumura, Y. Oka, and Y. Ishiwatari. Nuclear reactor design, an advanced course in nuclear engineering. *Springer*, 2014.

[31] Y. Saad. *Iterative Methods for Sparse Linear Systems*. University of Minnesota, 2003.

[32] R. Sanchez. On the spectral analysis of iterative solutions of the discretized one-group transport equation. *Annals of Nuclear Energy*, 31:2059–2075, 11 2004.

[33] R. Sanchez and A. Chetaine. A synthetic acceleration for a two-dimensional characteristic method in unstructured meshes. *Nuclear Science and Engineering*, 136(1):122–139, 2000.

[34] S. Santandrea and R. Sanchez. Acceleration techniques for the characteristic method in unstructured meshes. *Annals of Nuclear Energy*, 29(4):323 – 352, 2002.

[35] D. Shepard. Closing paris agreement signing press release. Technical report, UNFCC, 04 2016.

[36] M. A. Smith, E. E. Lewis, and B.-C. Na. Benchmark on deterministic transport calculations without spatial homogenisation. Technical report, Organisation For Economic Co-operation And Development: Nuclear Energy Agency, 2005.

[37] K. Suk Moon, N. Zin Cho, J. Man Noh, and S. Hong. Acceleration of the analytic function expansion nodal method by two-factor two-node nonlinear iteration. *Nuclear Science and Engineering*, 132:194–202, 06 1999.

[38] W. Walters. Use of the chebyshev-legendre quadrature set in discrete-ordinate codes. 1 1987.

[39] S. Yuk and N. Z. Cho. Two-level convergence speedup schemes for p-cmfd acceleration in neutron transport calculation. *Nuclear Science and Engineering*, 188(1):1–14, 2017.

[40] R. Zerr and Y. Azmy. A parallel algorithm for solving the multidimensional within-group discrete ordinates equations with spatial domain decomposition. *International Conference on the Physics of Reactors 2010, PHYSOR 2010*, 1:476–491, 01 2010.

[41] N. Zin Cho and C. Park. A comparison of coarse mesh rebalance and coarse mesh finite difference accelerations for the neutron transport calculations. *Breakthrough of Nuclear Energy by Reactor Physics and Scientific Computing*, pages 16–26, 05 2019.

université PARIS-SACLAY
ÉCOLE DOCTORALE
INTERFACES
Approches Interdisciplinaires : Fondements, Applications et Innovation

Titre : Vers des méthodes d'accélération stables et efficaces en contextes parallèles

Mots clés : Equation de transport en coordonnées discrètes, accélération non linéaire, analyse de la stabilité, analyse du rayon spectral, CMFD

Résumé : Dans cet article, une nouvelle bibliothèque de techniques non linéaires permettant d'accélérer l'équation de transport en coordonnées discrètes est proposée. Deux nouveaux types de méthodes d'accélération non linéaire appelées méthode de rééquilibrage des variantes spatiales (SVRM) et accélération de matrice de réponse (RMA), respectivement, sont introduits et étudiés. La première méthode, SVRM, est basée sur le calcul de la variation spatiale de premier ordre de l'équation de la balance des neutrons. RMA, est une méthode DP0 qui utilise la connaissance de l'opérateur de transport pour former une relation cohérente. Deux variantes distinctes de RMA, appelées respectivement Explicit-RMA (E-RMA) et Balance (B-RMA), sont dérivées. Les propriétés de convergence de CMFD, SVRM et RMA sont étudiées pour deux schémas d'itérations différents de l'opérateur de transport de la méthode des caractéristiques (MOC) pour une dalle 1D, en utilisant une analyse spectrale et une analyse de Fourier. Sur la base des résultats de la comparaison 1D, seuls les outils RMA et CMFD ont été implémentés dans la bibliothèque. Les performances de RMA sont comparées à CMFD en utilisant les benchmarks 3D C5G7 et ZPPR. Les schémas de résolution parallèles et séquentiels sont considérés. L'analyse des résultats indique que les deux variantes de RMA ont une efficacité et une stabilité améliorées par rapport au CMFD, pour les matériaux à diffusion optique. De plus, le RMA montre une amélioration importante de la stabilité et de l'efficacité lorsque la géométrie est décomposée spatialement. Pour obtenir des performances numériques optimales, une combinaison de RMA et de CMFD est suggérée. Une enquête plus approfondie sur l'utilisation et l'amélioration de la RMA est proposée. De plus, de nombreuses idées pour étendre les fonctionnalités de la bibliothèque sont présentées.

Title : The Advancement of Stable, Efficient and Parallel Acceleration Methods for the Neutron Transport Equation

Keywords : Discrete-ordinates transport equation, non-linear acceleration, stability analysis, Spectral radius analysis, CMFD

Abstract : In this paper a new library of non-linear techniques for accelerating the discrete-ordinates transport equation is proposed. Two new types of nonlinear acceleration methods called the Spatially Variant Rebalancing Method (SVRM) and the Response Matrix Acceleration (RMA), respectively, are introduced and investigated. The first method, SVRM, is based on the computation of the zeroth and first order spatial variation of the neutron balance equation. RMA, is a DP0 method that uses knowledge of the transport operator to form a consistent relationship. Two distinct variants of RMA, called Explicit-RMA (E-RMA) and Balance (B-RMA), respectively, are derived. The convergence properties of CMFD, SVRM and RMA are investigated for two different iteration schemes of the method of characteristics (MOC) transport operator for a 1D slab, using spectral and Fourier analysis. Based off the results of the 1D comparison, only RMA and CMFD were implemented in the library. The performance of RMA is compared to CMFD using the C5G7 and the ZPPR 3D benchmarks. Both parallel and sequential solving schemes are considered. Analysis of the results indicates that both variants of RMA have improved effectiveness and stability relative to CMFD, for optically diffusive materials. Moreover, RMA shows great improvement in stability and effectiveness when the geometry is spatially decomposed. To achieve optimal numerical performance, a combination of RMA and CMFD is suggested. Further investigation into the use and improvement of RMA is proposed. As well, many ideas for extending the features of the library are presented.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France