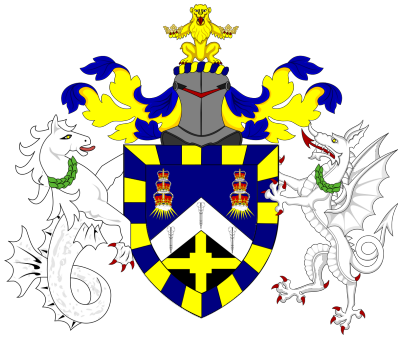
Reduced order models for the angular discretisation of the Boltzmann transport equation



Alexander Conway Hughes

Submitted in partial fulfillment of the requirements

of the Degree of Doctor of Philosophy

School of Engineering and Materials Science

Queen Mary University of London

January 2023

Statement of originality

I, Alexander Conway Hughes, confirm that the research included within this thesis is my own work or that where it has been carried out in collaboration with, or supported by others, that this is duly acknowledged below and my contribution indicated. Previously published material is also acknowledged below.

I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

Details of collaboration and publications:

Material from the author's previous academic publications is included in this thesis. These publications were produced during the course of this research project, in collaboration with the author's academic supervisor, Dr. Andrew Buchan.

Signature: Alexander Conway Hughes

Date: 28/01/23

Abstract

In order to predict the behaviour of nuclear reactors, it is necessary to produce high fidelity numerical solutions to the Boltzmann transport equation. However, this is computationally expensive due to the complexity of both the equation and the physical systems involved. The purpose of this work is to develop novel reduced order models for the angular discretisation of the Boltzmann transport equation, based on proper orthogonal decomposition (POD), in order to reduce the computational cost of producing such solutions. Previous implementations of angular POD created global basis functions spanning the sphere, and the entire problem domain in space. Chapter two presents the first innovation of this work, which is to partition the snapshots in angle, such that each POD basis function has compact support over a single octant. An algorithm for angular adaptivity using these basis functions is also presented. Chapter three presents a model which partitions both the angular and spatial dimensions, and generates independent angular basis sets for each partition. An implementation of angular adaptivity for this model is also described. Chapter four describes two solver acceleration methods involving the aforementioned models. The first is a simple multigrid solver for the reduced order models, and the second is an investigation into their ability to accelerate full order model solves. The methods presented in chapter two are shown to significantly improve solver stability and reduce error for a given number of basis functions, thus improving computational efficiency over previous methods. The methods in chapter three are demonstrated to reduce error compared to the methods in the previous chapter, thus increasing computational efficiency further. The solver acceleration methods in chapter four are shown to be effective in reducing solve times, though the benefits depend on the complexity and resolution of the full order models.

Acknowledgements

I would like to express my deepest gratitude to my supervisor Dr. Andrew Buchan for his vital support, guidance, and encouragement throughout my PhD studies. I truly appreciate the opportunity to work with him, and the invaluable lessons I have learned along the way.

Many thanks to my supervisor Dr. Edo Boek, and to Dr. Jens-Dominik Mueller for conducting my yearly reviews and providing useful feedback on my progress.

I would also like to extend my sincere thanks to the reviewers of this thesis for generously donating their time, and for the valuable feedback they provided.

I am thankful to the faculty and staff of the School of Engineering and Materials Science, and Queen Mary University of London as a whole, for providing me with the resources and support required to complete my research. This research utilised Queen Mary's Apocrita HPC facility, supported by QMUL Research-IT: http://doi.org/10.5281/zenodo.438045

Finally, I am immensely grateful to my family for their support, encouragement, and love. I am particularly appreciative of their willingness to entertain my endless questions as a child, which inspired my interest in science and technology. I couldn't have done this without you!

Contents

List of Figures

List of Tables

Abbreviations, Conventions and Variables

This section provides an explanation of conventions, a table of frequently used abbreviations, and several tables of variable definitions used throughout the text, for the convenience of the reader. Each abbreviation, convention and variable is also defined upon its first use in the text.

Conventions

Throughout this work, variables in brackets will denote continuous dependance, so that $\Psi(\vec{r}, \Omega)$ is continuous in space and angle, whereas $\Psi(\vec{r})$ is continuous in space but discretised in angle.

All cross-sections used in this work are macroscopic, with units of inverse length. They represent the expected number of interactions per second per unit flux.

Abbreviations

Abbreviation	Definition
(A)DPOD	(Adaptive) Discontinuous Proper Orthogonal Decomposition
(A)RDPOD	(Adaptive) Regional Discontinuous Proper Orthogonal Decomposition
BE	Boundary Element
BF	Basis Function
BR	Boundary Region
BTE	Boltzmann Transport Equation
DG	Discontinuous Galerkin
FDM	Finite Difference Method
FEM	Finite Element Method
FVM	Finite Volume Method
P_N	Spherical Harmonics
POD	Proper Orthogonal Decomposition
ROM	Reduced Order Model
S_N	Discrete Ordinates
SVD	Singular Value Decomposition

Cross-sections

Variable	Definition
Σ_a	Absorption cross-section
Σ_f	Fission cross-section
Σ_s	Scattering cross-section
Σ_t	Total interaction cross-section
Σ_{tr}	Transport cross-section

Dimensions

Variable	Definition
E	Energy
$\vec{r}$	Position
t	Time
V	Volume
Ω	Direction

Fluxes

Variable	Definition
$\Psi(\vec{r}, \Omega)$	Space-angle continuous angular flux
$\Psi(\vec{r})$	Spatially continuous angular flux
ψ	Fully discretised angular flux
$\Phi(\vec{r})$	Spatially continuous scalar flux
ϕ	Fully discretised scalar flux
$\alpha(\vec{r})$	Spatially continuous reduced-order angular flux
α	Fully discretised reduced-order angular flux

Matrices

Variable	Definition
A	Operator matrix
S	Snapshot matrix
U	Basis functions
$\mathcal{U}$	Reduced order basis functions
Σ	Singular values

Numbers

Variable	Definition
k_{eff}	Effective neutron multiplication factor
N_a	Full order angular basis functions
N_e	Elements in spatial mesh
N_n	Nodes in spatial mesh
N_p	Variations of problem used for training model
N_q	Angular coefficients within octant q
N_r	Full order snapshots associated with the region r
N_s	Full order spatial basis functions
n_a	Reduced order angular basis functions
n_h	Snapshots used to generate reduced order model
n_{qe}	Basis functions in octant q element e
n_{qr}	Basis functions in octant q region r
n_r	Spatial partitions for RDPOD
$n_{rx} \times n_{ry}$	Columns $\times$ rows of spatial partitions
n_0	Basis functions in coarse multigrid stage
n_1	Basis functions in fine multigrid stage
n_+	Basis functions to add per adaptive stage
n_-	Basis functions to remove per adaptive stage
ν	Average neutrons per fission
v	Particle velocity

Other

Variable	Definition
$D(\vec{r}, E, t)$	Diffusion coefficient
$\mathcal{G}(\Omega)$	Full order angular basis functions
$\vec{J}$	Current vector
$\hat{n}$	Unit vector normal to element boundary
$\mathcal{N}(\vec{r})$	Full order spatial basis functions
$q_{ex}(\vec{r}, \Omega, E, t)$	External source
$q_s(\vec{r}, \Omega' \rightarrow \Omega)$	Scattering source
$Q(\vec{r}, E, t)$	Scalar source
V_e	Element volume
Z	Space-angle phase-space partition
Γ_e	Element boundary
$\xi(E)$	Fission neutron energy distribution

Chapter 1

Introduction

This chapter aims to provide an introduction to this project. It begins by explaining the motivation and objectives of the investigation, including a brief description of the relevant physics and solution methods, as well as the issues with existing methods which this work intends to address. A summary of the information contained in each chapter is then provided.

1.1 Motivation & Objectives

Nuclear fission is a radioactive decay process in which a large 'parent' nucleus splits into two or more smaller 'daughter' nuclei, releasing some of the nuclear binding energy of the parent nucleus in the process. The amount of energy released in each of these events is relatively large on the scale of atoms, but insignificant on a macroscopic scale, approximately 200MeV or 3×10^{-11}J for the fission of ^{235}U [5]. Therefore, many quadrillions of fission interactions are required to generate useful amounts of power. While some level of radioactive decay occurs naturally, it is a relatively slow process in the relevant cases. However, many decay paths release neutrons with significant kinetic energy, which are capable of interacting with other parent nuclei and inducing fission. This induced fission releases more neutrons, which can cause more fission in turn. This cascading process by which fission begets fission is known as a nuclear chain reaction. The purpose of a nuclear reactor is to sustain a controlled nuclear chain reaction in order to generate heat, which is used to spin a turbine and produce electrical power.

An important measure of this process is the effective multiplication factor, k_{eff}. This represents the average number of neutrons released by each neutron in the system. If $k_{eff} < 1$, then each generation of neutrons will be smaller than the previous generation. This is known as a subcritical reaction, and will result in the chain reaction dying out. If $k_{eff} = 1$, then each existing neutron produces exactly one additional neutron on average, and the reaction will sustain itself at its current rate. This is known as a critical reaction, the desired state of a nuclear reactor at operating power. If $k_{eff} > 1$, the reaction is in a supercritical state. This means that each generation of neutrons will produce more neutrons, and the reaction will increase in power exponentially. This situation is likely to result in catastrophic damage to the reactor, and has the potential to cause widespread devastation should the reactor's containment be breached [6]. Since most fission events are caused by interactions between neutrons and the nuclear fuel, the value of k_{eff} is dependent on the distribution of neutrons within the reactor in both position and momentum. It is therefore vital to accurately predict this distribution in various circumstances, in order to ensure that the reactor in question can maintain a critical chain reaction without the possibility of supercriticality.

The neutron distribution within a reactor is described by the Boltzmann transport equation (BTE), a linear partial differential equation which describes the statistical behaviour of systems of particles [7]. Rather than tracking the positions and momenta of individual particles, it considers the probability distribution of particles within a system. Solving the equation therefore provides the expected distribution of particles within a particular system over time, in both position and momentum space. Properties such as the rate of fission, nuclear burn-up and radiation dose rates can then be calculated from such solutions. A more detailed theoretical discussion of the BTE is presented in section 2.1.

While the BTE can be solved analytically in some cases [8], this is not possible in general. As a result, numerical methods are frequently used to produce approximate solutions. This class of methods works through the process of discretisation - reducing the degrees of freedom of a system from infinite to finite. This results in a solution which is described by a finite number of unknown positions, momenta and times, and can therefore be solved computationally. Outside of these points, interpolation is typically

used to produce an approximation to the continuous solution. Each type of dimension must be discretised differently, and a variety of methods exist for each. Some common methods of discretising the angular dimensions are presented in section 2.3, and methods for the spatial dimensions are presented in section 2.4.

Numerical models of the BTE can require significant computational resources due to the scale of the problems involved and the high dimensionality of the equations which must be solved. Scale is an issue because accurately resolving a problem requires sufficient spatial and angular resolution, which can be computationally expensive for problems with large or complex domains. For example, reactor cores are constructed from tens of thousands of fuel and control rods, and therefore require enormous spatial meshes and high angular resolution to solve. The issue of dimensionality arises because the BTE is a seven-dimensional equation - the distribution of particles has up to three spatial coordinates, three momentum-space coordinates, and evolves over time [9]. Each additional dimension multiplies the degrees of freedom of the problem, increasing both the computation time and memory requirements. It is sometimes possible to simplify the equation in order to reduce the number of relevant dimensions. For example, if only the steady-state solution to a particular system is required, the time dimension can be neglected, or if the problem is invariant in one spatial axis, that axis can effectively be ignored. However, given the complexity of the systems involved, this is often insufficient, even when it is possible.

Over the years, significant research effort has been dedicated to efficiently solving the BTE using numerical methods. In the last decade, reduced order models (ROMs) have gained popularity for this purpose, as they are often capable of reducing problem sizes by multiple orders of magnitude, resulting in a similar scale of reductions to solve times. This is achieved by projecting the governing equations onto a space which is capable of representing the solution accurately with fewer degrees of freedom, typically because it is optimised for the particular problem under consideration. ROMs can be broadly divided into two categories - *a priori* ROMs attempt to use information about the problem to create optimised basis functions before the problem has been solved, whereas *a posteriori* ROMs allow solutions from a conventional model to be used in the creation of a new model with comparable accuracy but reduced computational complexity. The motivation

behind this project was to develop novel *a posteriori* ROMs to reduce the computational cost of solving the BTE in the context of neutron transport calculations.

A wide variety of methods for generating reduced basis sets have been proposed, which are discussed further in section 2.6.3. This work will focus on the method of proper orthogonal decomposition (POD) in angle, which makes use of the singular value decomposition (SVD) to generate basis functions which are guaranteed to be optimal representations of the known solutions in the Frobenius norm [10]. This method has been shown to drastically reduce the number of basis functions required to reach a given level of error [11]. However, it introduced instability which caused the solver to converge more slowly in some situations, counteracting the improvements in computational efficiency from the use of a smaller basis. The primary objectives of this project were to eliminate the solver instability of angular POD in order to realise its potential as a reduced order modelling method, and to further improve its computational efficiency if possible.

1.2 Modelling Methods

A wide variety of discretisation, solution and reduced order modelling methods have been applied to the BTE, which are discussed in more detail in chapter 2. However, reduced order modelling methods are both easier to produce and more generally useful when they rely on common techniques rather than being built upon more exotic methods. Therefore, this project will exclusively utilise the integro-differential form of the BTE derived in section 2.1.1, the spatial dimensions of both the full and reduced order models will be discretised using the finite element method described in 2.4.3 with uniform square grids and discontinuous Galerkin basis functions, and the discrete ordinates method presented in 2.3.1 will be utilised to discretise the angular dimensions. These techniques are widely used in the field of radiation transport, which maximises the utility of the resultant models. Additionally, they are relatively simple both conceptually and mathematically, which makes the models easier to understand and develop. Furthermore, the full order model described is able to accurately resolve any problem given sufficient resolution, including both purely advective and highly scattering problems, as well as problems which combine the two. This is a significant advantage, as other formulations can struggle in some situations. For example, the diffusion approximation is unable to

accurately resolve fluxes which are highly anisotropic, the even parity formulation has difficulty dealing with void regions, and the P_N discretisation encounters problems in the vicinity of physical discontinuities, all of which are of interest in reactor physics.

A sweep-based solver utilising first-order upwinding will be used to produce solutions for both the full and reduced order models, as described in section 2.5.1. The full order model sweeps each ray individually in one direction. For the reduced order models, the basis functions are heavily coupled, and so all angular coefficients associated with a particular element are solved for at once. In addition, standard angular POD basis functions span the sphere, and are thus capable of transmitting flux in any direction. Every basis function must therefore be swept in all four directions each iteration in order to guarantee that information is propagated correctly through the domain. By contrast, the reduced order models presented later in this thesis span a subset of the sphere, and it should therefore be possible to sweep each a single time per iteration. This would be a significant optimisation, improving the computational efficiency of the models by up to a factor of four in two dimensions and eight in three dimensions. However, the focus of this research is not on developing the most efficient solver technology to resolve the various angular models, and doing so would require substantial development. Solver optimisation is therefore left for future work to consider.

1.3 Test Problems

In order to investigate the efficacy of the models produced in the course of this research, three benchmark problems which are commonly used in the field of radiation transport have been examined. These problems were selected to provide a fair assessment of the capabilities of the models in a wide range of material regimes, including highly scattering materials, highly absorptive materials and voids. A diagram and brief description of each problem is presented here, along with an explanation of their utility. The discretisations and quantitative material properties used are provided in later chapters, as in some cases these varied between chapters.

1.3.1 The Dog-Leg Duct Problem

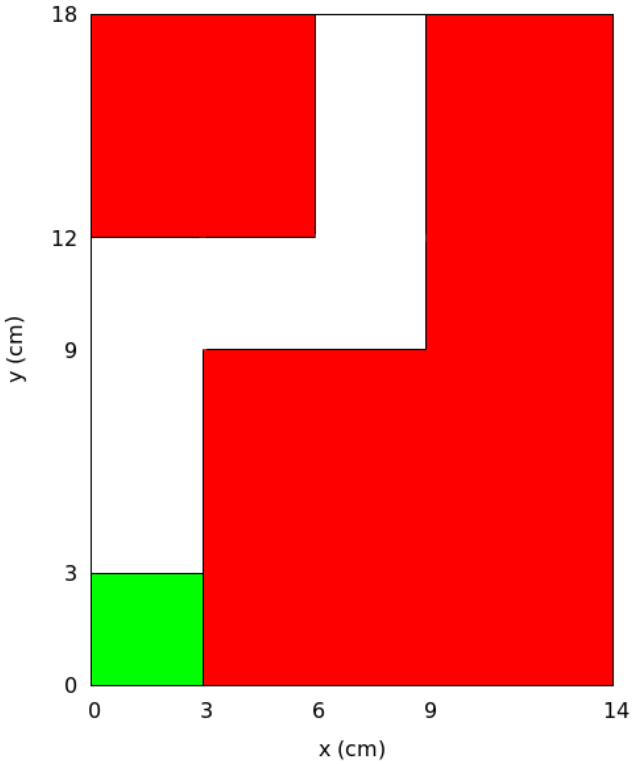


Figure 1.1: Schematic of the dog-leg duct problem. Green is a source, white is the duct and red is a highly absorbing material.

The first test problem used in this work is an advective dog-leg duct problem [12]. Figure 1.1 shows a schematic of the domain. The green region is an isotropic source, the white region is the duct, and the red regions are heavy absorbers. Vacuum boundary conditions are applied to the top and right boundaries, and reflective boundary conditions to the bottom and left boundaries. The domain does not contain any scattering materials, and so tests the abilities of the models in a purely advective regime, such as those found in radiation shielding. This is a particularly difficult class of problem for the S_N discretisation to solve, since the lack of scattering makes ray effects more prominent.

1.3.2 The Watanabe-Maynard Problem

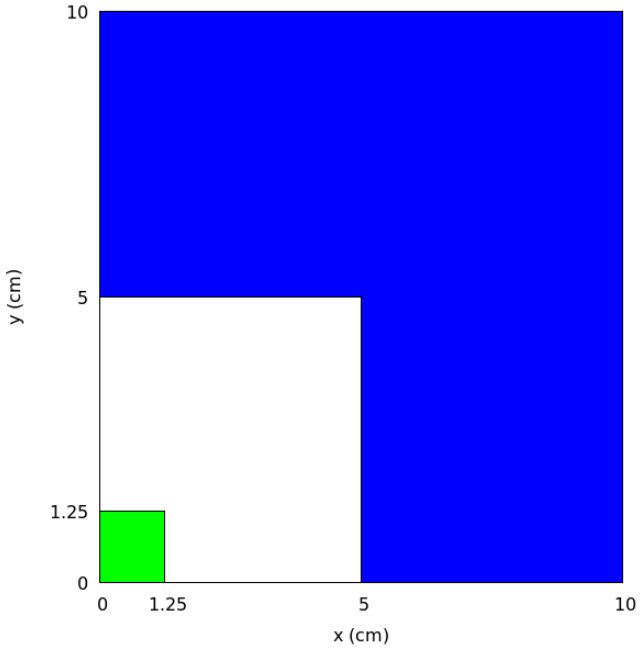


Figure 1.2: Schematic of the Watanabe-Maynard problem. Green is the source, white is a void and blue is a highly scattering material.

The second test problem used in this work is the Watanabe-Maynard problem, proposed by Watanabe and Maynard in 1986 [13]. A schematic is shown in figure 1.2. The green region is an isotropic square source, the white region is a void, and the blue region is a highly scattering material. Vacuum boundary conditions are applied to the top and right boundaries, and reflective boundary conditions to the bottom and left boundaries. The domain contains both void and highly scattering regions, which test different aspects of the models. The void region tests their ability to resolve fluxes in purely advective regions with no absorption. Some models struggle to deal with voids, particularly those based on second order formulations of the BTE, as discussed in section 2.1.4. The scattering region tests the ability of the models to resolve fluxes in highly scattering media, such as neutron moderators. These tests are relevant because scattering and near-void regions are both present in nuclear reactors. This is also a relatively difficult problem for the models to solve as ray effects are a significant problem in void regions.

1.3.3 The Checkerboard Problem

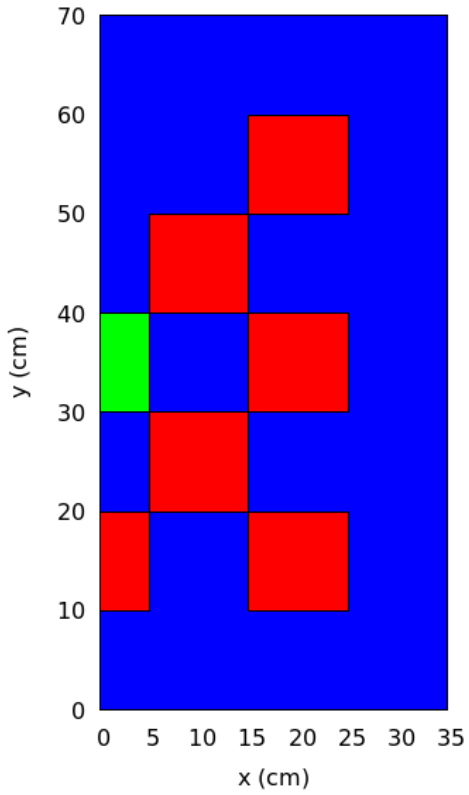


Figure 1.3: Schematic of the checkerboard problem. The green region is the source, the blue regions are scattering and the red regions are highly absorptive.

The third test problem used in this work is a checkerboard problem first proposed by Brunner in 2002 [14]. A schematic is shown in Figure 1.3. The green region is an isotropic source, the blue regions are purely scattering, and the red regions are highly absorptive. The full problem presented in [14] is symmetrical about a vertical line through the centre. The right half of the problem has therefore been solved with reflective boundary conditions on the left boundary and vacuum conditions applied to all other boundaries. This is an equivalent problem, but with half of the spatial complexity, making it faster to solve. The checkerboard problem has highly absorptive regions which create large discontinuities in the flux. These discontinuities are difficult for some models to deal with, such as those based on the P_N angular discretisation, as discussed in section 2.3.2. The problem also contains both scattering and heavy absorption, whereas the other test problems contain one or the other. This is an important combination of material properties to test, since both are common inside nuclear reactors.

1.4 Chapter Contents

Chapter 2 provides a broader and more detailed introduction to the background and theory of this research. It begins with a discussion of the Boltzmann transport equation, including its derivation and commonly utilised formulations. Various methods for discretising and solving the BTE are then described, along with some popular reduced order modelling techniques. The history and importance of these topics are explained, and the literature around reduced order modelling is reviewed, both for the neutron transport equation and more generally.

The previous angular POD method described in [11] exhibited significant solver instability which reduced its computational efficiency, and so the first objective was to fix this instability. To that end, chapter 3 introduces a new model known as discontinuous POD (DPOD), which partitions the angular domain and generates independent angular POD functions for each partition. The resulting angular basis functions span only a subset of the sphere, in contrast to previous angular POD methods which generated basis functions spanning the entire sphere. It is demonstrated that DPOD is capable of creating stable basis sets which produce efficient solutions to multi-dimensional transport problems, eliminating the aforementioned solver instability of the existing angular POD method. Chapter 3 also introduces a method of angular adaptivity which relies on the fact that DPOD produces hierarchical basis functions with compact angular support. An article on the DPOD method with angular adaptivity has been published in the *International Journal for Numerical Methods in Engineering* (*IJNME*) [15].

Chapter 4 presents a highly effective modification to DPOD which improves its ability to reduce the dimensionality of problems, and therefore reduces error for a given solve time. Known here as regional discontinuous POD (RDPOD), the new method partitions the spatial domain as well as the angular domain, then forms and applies independent sets of angular basis functions for each partition of the space-angle phase-space. That is, the method forms separate angular POD functions over distinct regions of space, rather than using a single basis set per octant for all regions of space as in the previous method. In addition, chapter 4 presents an implementation of angular adaptivity using the RDPOD basis functions. This method provided further improvements to computational efficiency compared to DPOD, and has also been published in the *IJNME* [16].

Chapter 5 discusses two solver acceleration techniques which utilise the reduced order models presented in earlier chapters. The first of these is the use of low-order POD solutions to accelerate the solution of higher-order POD solutions, which is a simple multigrid method. This method allows solutions to be formed on coarse angular meshes, which can then be used as initial solutions for fine angular meshes, reducing the number of iterations required to converge the higher resolution solution. This allows relatively large and accurate reduced order models to be solved in a fraction of the time that would be required otherwise, by sequentially generating approximations of increasing accuracy. The second acceleration technique is the use of the aforementioned angular POD methods to generate an initial solution to a problem, which is then projected onto the full order discrete ordinate model and solved again, increasing the solution's accuracy compared to the reduced order model solution. This allows a solution with comparable accuracy to the full order model to be produced in fewer full order iterations, and likely less time overall, depending on the speed and accuracy of the reduced order model. These two investigations provide interesting insights into solver acceleration technologies involving ROMs, and could potentially be developed into an additional paper after further development and data gathering.

Finally, chapter 6 summarises the information presented in previous chapters. It discusses the effectiveness of the new methods, and draws conclusions based on the results presented. Potential improvements and new ideas related to the contents of this thesis are also discussed.

Chapter 2

Numerical Models of Neutron Transport

This chapter contains a detailed discussion of the theory behind this thesis, including the Boltzmann transport equation (BTE), numerical methods, and reduced order modelling. It describes the history and importance of these subjects, along with the current state of the art. Many of the methods used to discretise and solve the BTE are presented, including those used in this work. A review of the literature surrounding reduced order modelling is also included, both for neutron transport and other applications.

2.1 The Boltzmann Transport Equation

This project is primarily concerned with finding solutions to the BTE, derived by Ludwig Boltzmann in 1872 [7]. The equation is extremely difficult to solve analytically for any but the simplest geometries. In the 1930s, the study of radiation transport in stellar atmospheres led to analytical solutions for semi-infinite one-dimensional geometries, and further research since then has expanded the range of analytically solvable problems, but such solutions are still limited to idealised and simplified configurations [17]. At the time of its discovery, computers were not sufficiently advanced to be a viable means to solve the equation either. However, over the following century, computing power and modelling techniques improved to the point where solving the BTE on complex domains using numerical methods became a viable option. As the BTE governs the dynamics of

systems of particles, numerical models capable of solving it play an essential role in the analysis of many systems across a wide range of fields in science and engineering. For example, it is used to model airflow and aerodynamic properties [18–21], the evolution of various cosmological systems [22–25], and meteorological data [26–28]. The BTE has also been used in the process of design optimisation [29, 30], and to calculate material properties by modelling the movement of force carriers. For example, [31] modelled the transport of phonons in order to determine the thermal conductivity of various materials, and [32] calculated the electrical conductivity of a fluid by modelling electron transport.

2.1.1 Derivation

The BTE can be derived in its integro-differential form by considering the expected number of particles $N(\vec{r}, \Omega, E, t)dV d\Omega dE$ in an infinitesimal volume dV about the position $\vec{r}$, travelling within the set of infinitesimal angles $d\Omega$ about Ω, with energy in the range dE about E, at time t. Conservation requires that the change in N is equal to the number of particles entering or emitted within N, minus the losses from streaming and collisions. By considering each of these terms separately, and noting that the angular flux $\Psi(\vec{r}, \Omega, E, t) = vN(\vec{r}, \Omega, E, t)$, where v is the particle velocity, the BTE can be obtained in integro-differential form.

$$\frac{1}{v}\frac{\partial}{\partial t}\Psi(\vec{r}, \Omega, E, t) + \Omega \cdot \vec{\nabla}\Psi(\vec{r}, \Omega, E, t) + \Sigma_t(\vec{r}, E)\Psi(\vec{r}, \Omega, E, t) = \int_{\Omega'} \Sigma_s(\vec{r}, \Omega' \to \Omega)\Psi(\vec{r}, \Omega', E, t)d\Omega' + \int_{\Omega'}\int_{E'} \nu\chi(E)\Sigma_f(\vec{r}, E')\Psi(\vec{r}, \Omega', E', t)d\Omega' dE' + q_{ex}(\vec{r}, \Omega, E, t), \quad (2.1)$$

where the terms on the left hand side account for time dependence, particle streaming, and particles removed due to interactions, respectively. The terms on the right hand side account for particles added due to scattering, fission and external sources, respectively. $\Sigma_s(\vec{r}, \Omega' \to \Omega)$ represents the macroscopic scattering cross section from the initial angle Ω' to the final angle Ω, and the integration over Ω' therefore gives the amount of flux scattering into Ω from all angles. $\Sigma_f(\vec{r}, E')$ is the macroscopic fission cross section for the initial energy E', ν is the average number of neutrons released per fission, and $\chi(E)dE$ is the probability that one of these neutron will have an energy within dE about E. The integration over all initial energies E' and initial angles Ω' therefore gives the

expected flux distribution produced by fission events. $\Sigma_t(\vec{r}, E)$ is the total interaction cross section. In this project, fission sources will be neglected in order to simplify the gathering of results and the presentation of each problem, but they are relatively easy to implement. A full derivation of equation 2.1 is given in [17].

2.1.2 The Diffusion Approximation

By integrating equation 2.1 over all angles, one may remove the angular dependence of the equations and obtain an equation which involves only the scalar flux,

$$\frac{1}{v}\frac{\partial}{\partial t}\Phi(\vec{r}, E, t) + \vec{\nabla} \cdot \vec{J}(\vec{r}, E, t) + \Sigma_t(\vec{r}, E)\Phi(\vec{r}, E, t) = Q(\vec{r}, E, t), \tag{2.2}$$

where $\vec{J}(\vec{r}, E, t) = \int d\Omega\Omega\Psi(\vec{r}, \Omega, E, t)$ defines the current vector, which gives the net amount of neutrons flowing through a unit area in a unit of time, and $Q(\vec{r}, E, t) = \int d\Omega q(\vec{r}, \Omega, E, t)$ defines the scalar source.

Fick's first law gives the current vector of a purely diffusive system as,

$$\vec{J}(\vec{r}, E, t) = -D(\vec{r}, E, t)\nabla\Phi(\vec{r}, E, t), \tag{2.3}$$

where D is the diffusion coefficient. In a weakly absorbing medium, $D \approx (3\Sigma_{tr})^{-1}$, where Σ_{tr} is the macroscopic transport cross section. By substituting equation 2.3 into equation 2.2, one may obtain the diffusion approximation to the BTE,

$$\frac{1}{v}\frac{\partial}{\partial t}\Phi(\vec{r}, E, t) - \vec{\nabla} \cdot D(\vec{r}, E, t)\nabla\Phi(\vec{r}, E, t) + \Sigma_t(\vec{r}, E)\Phi(\vec{r}, E, t) = Q(\vec{r}, E, t). \tag{2.4}$$

This approximation is equivalent to assuming that the flux distribution is close to isotropic throughout the domain. While this is a reasonable assumption in specific circumstances, such as highly scattering media, it is a poor approximation to general angular flux distributions. However, in circumstances where it is applicable, the diffusion approximation is a highly computationally efficient method of solving the BTE, as integrating out the angular dimension reduces the degrees of freedom of the system.

2.1.3 The Integral Formulation

The integral form of the BTE can be derived starting from equation 2.1 by noting that the streaming operator $\Omega \cdot \vec{\nabla}$ is the directional derivative along the direction of neutron travel. The equation can therefore be rewritten in terms of the distance along the direction of neutron travel, u. However, in order to derive the integral transport equation it is necessary to look back along the line of neutron travel, and so the equation is formulated in terms of $R = -u$, the distance from $\vec{r}$ back along the path of the neutron in the direction Ω to the point $\vec{r} - R(\vec{r}, \Omega)\Omega$ where neutrons enter the domain. This gives,

$$-\frac{d}{dR}\Psi(\vec{r} - R\Omega, \Omega) + \Sigma_t(\vec{r} - R\Omega)\Psi(\vec{r} - R\Omega, \Omega) = q(\vec{r} - R\Omega, \Omega) \tag{2.5}$$

The derivative of R can be removed by multiplying both sides of the equation with the integrating factor,

$$exp\bigg(- \int_0^R \Sigma_t(\vec{r} - R'\Omega)dR' \bigg), \tag{2.6}$$

where R' is a dummy variable which is integrated between 0 and R. When differentiated with respect to R, equation 2.6 gives:

$$\frac{d}{dR}exp\bigg(- \int_0^R \Sigma_t(\vec{r} - R'\Omega)dR' \bigg) = -\Sigma_t(\vec{r} - R\Omega)exp\bigg(- \int_0^R \Sigma_t(\vec{r} - R'\Omega)dR' \bigg). \tag{2.7}$$

Hence, multiplying equation 2.5 by this integrating factor gives,

$$\begin{aligned} - \frac{d}{dR}\Psi(\vec{r} - R\Omega, \Omega)exp\bigg(- \int_0^R \Sigma_t(\vec{r} - R'\Omega)dR' \bigg) \\ = q(\vec{r} - R\Omega, \Omega)exp\bigg(- \int_0^R \Sigma_t(\vec{r} - R'\Omega)dR' \bigg). \end{aligned} \tag{2.8}$$

Integrating along the path of neutron travel from 0 to R results in an equation for Ψ,

$$\begin{aligned} \Psi(\vec{r}, \Omega) = \int_0^R dR' q(\vec{r} - R'\Omega, \Omega)exp\bigg(- \int_0^R \Sigma_t(\vec{r} - R'\Omega)dR' \bigg) \\ + \Psi(\vec{r} - R\Omega, \Omega)exp\bigg(- \int_0^R \Sigma_t(\vec{r} - R'\Omega)dR' \bigg). \end{aligned} \tag{2.9}$$

By analogy to the transmission of light, the optical path τ between $\vec{r}$ and $\vec{r} - R'\Omega$ is expressed as,

$$\tau(\vec{r}, \vec{r} - R'\Omega) = \int_0^{R'} \Sigma_t(\vec{r} - R''\Omega) dR'', \tag{2.10}$$

where R'' is again a dummy variable used for integration. Inserting this expression into equation 2.9 gives the integral form of the BTE for angular flux as,

$$\Psi(\vec{r}, \Omega) = \int_0^R dR' q(\vec{r} - R'\Omega, \Omega) exp\Big(- \tau(\vec{r}, \vec{r} - R'\Omega)\Big) + \Psi(\vec{r} - R\Omega, \Omega) exp\Big(- \tau(\vec{r}, \vec{r} - R\Omega)\Big). \tag{2.11}$$

If R goes to infinity for all $\vec{r}$ and Ω, the second term in equation 2.11 goes to zero. As isotropic scattering is assumed, the emission density is independent of angle and $q(\vec{r} - R'\Omega, \Omega)$ can be replaced by $Q(\vec{r} - R'\Omega)$, which depends only on the scalar flux. The equation can therefore be integrated over all of Ω to give the scalar flux integral equation,

$$\Phi(\vec{r}) = \int d\Omega \int_0^\infty dR' Q(\vec{r} - R'\Omega) exp\Big(- \tau(\vec{r}, \vec{r} - R'\Omega)\Big). \tag{2.12}$$

This may be expressed as a volume integral by substituting $R' = |\vec{r} - \vec{r}\,'|$ and converting to spherical coordinates with $\vec{r}$ as the origin, which gives $dV' = 4\pi d\Omega R'^2 dR'$ and $d\Omega = \frac{sin\theta}{4\pi} dw d\theta$, resulting in the equation,

$$\Phi(\vec{r}) = \int dV' \frac{Q(\vec{r}\,') exp\Big(- \tau(\vec{r}, \vec{r}\,')\Big)}{4\pi |\vec{r} - \vec{r}\,'|^2}. \tag{2.13}$$

Note that while the volume integral extends over all of space, this form of the BTE also applies to finite problems provided that they have vacuum boundary conditions [17].

This form of the BTE enables solutions of arbitrary angular precision to be calculated, depending on the accuracy with which the terms arising from the angular integration are evaluated [17]. However, in this formulation all spatial points are coupled together, and as such the computational complexity of finding solutions scales poorly with increasing spatial resolution. This makes the integral transport method ideal for problems which are spatially simple but have complex angular distributions, such as periodic flux distributions within infinite reactor lattices. However, it is much less practical for

complex, high-resolution spatial meshes.

2.1.4 Second Order Formulations

Second order formulations of the BTE have also been proposed [33, 34], including the even parity [35] and self-adjoint angular flux [36] forms, which both transform the first-order initial value problem into a self-adjoint second-order boundary value problem. A significant advantage of these formulations is that they produce symmetric positive-definite matrix systems, enabling the use of efficient numerical techniques such as the conjugate gradient method [37].

The even parity form of the BTE may be derived from equation 2.1 [17]. The angular flux in a direction Ω is split into two components - the even parity flux,

$$\Psi^{+}(\vec{r},\Omega,E,t) = \frac{1}{2}[\Psi(\vec{r},\Omega,E,t) + \Psi(\vec{r},-\Omega,E,t)], \tag{2.14}$$

and the odd parity flux,

$$\Psi^{-}(\vec{r},\Omega,E,t) = \frac{1}{2}[\Psi(\vec{r},\Omega,E,t) - \Psi(\vec{r},-\Omega,E,t)]. \tag{2.15}$$

$-\Omega$ is substituted into equation 2.1, and the result is added to equation 2.1 to obtain an equation involving Ψ^{+} and Ψ^{-}. A second equation is obtained in the same way, but replacing addition with subtraction. The two equations can then be combined to eliminate Ψ^{-}, resulting in the second-order even parity form of the BTE:

$$-\Omega\cdot\vec{\nabla}\frac{1}{\Sigma_t(\vec{r},E)}\Omega\cdot\vec{\nabla}\Psi^{+}(\vec{r},\Omega,E,t)+\Sigma_t(\vec{r},E)\Psi^{+}(\vec{r},\Omega) = \Sigma_s(\vec{r},E)\Psi(\vec{r},\Omega,E,t)+Q(\vec{r},E,t) \tag{2.16}$$

This formulation has the advantage that the odd parity moments can be ignored when solving for the scalar flux, reducing the system's degrees of freedom. An issue with this formulation of the BTE is that the total absorption cross-section appears in a denominator, which causes the system to become ill-formed in low absorption regions, and undefined for voids.

The self-adjoint angular flux method presents an alternative to the even parity method [38]. It can be obtained by adding together the even and odd parity equations, resulting

in:

$$-\Omega \cdot \vec{\nabla} \frac{1}{\Sigma_t - Q_s} \Omega \cdot \vec{\nabla} \Psi + (\Sigma_t - Q_s)\Psi = Q_{ex} - \Omega \cdot \vec{\nabla} \frac{1}{\Sigma_t - Q_s} Q_{ex}. \tag{2.17}$$

Its main advantages over the even parity form are that the full angular flux is obtained, and it allows boundary conditions to be more easily implemented. However, it becomes ill-defined in low-absorption and void regions for the same reason as the even parity formulation, and requires the solution of twice as many angular flux coefficients [39].

2.2 Deterministic and Stochastic Numerical Methods

Numerical solvers can broadly be classified into *deterministic* and *stochastic* methods. Deterministic methods always produce the same output for a given input, whereas stochastic methods involve randomness and produce a different result each time [40]. There have also been attempts to combine the two into a hybrid method [41]. This project will focus exclusively on deterministic methods, but a brief discussion of stochastic methods is also provided for completeness.

2.2.1 Deterministic Methods

Deterministic numerical methods rely on discretising the continuous governing equations of a problem. As continuous equations have an infinite number of degrees of freedom, meaning that the solution must be specified at an infinite number of points, they can not be solved exactly using numerical methods. By converting continuous equations into discrete equations which are defined by a finite number of unknown coefficients, it becomes possible to find an approximate solution. Spatial, angular, energetic and temporal dimensions must be discretised separately, using techniques which are specific to each type of dimension. Methods of discretisation for the angular and spatial dimensions of the BTE are discussed in sections 2.3 and 2.4, respectively. The time and energy dimensions are neglected in this project, as the mono-energetic time-independent form of the BTE is used.

Once the problem has been fully discretised, the result is a matrix equation of the form $Ax = b$, where the solution vector $x \in \mathbb{R}^n$ must be found. Since direct methods of solving such systems have order $n^2 log(n)$ time complexity at best [42], and n can be on the order of billions or even trillions, iterative methods are typically used to find

approximate solutions. These methods involve producing an initial guess at the solution, and using an iterative procedure which will converge the approximate solution towards the exact solution. Typically this initial solution is a zero vector, though in some cases an approximate solution produced by other means may be used. This can be beneficial, as the closer the initial and final solutions are, the fewer solver iterations will be required to converge the solution to a given level of accuracy. Some specific examples of iterative methods are discussed in section 2.5.

2.2.2 Stochastic Methods

Stochastic methods rely on repeated random sampling to produce numerical solutions to a problem [43]. For this reason, they are often known as Monte Carlo methods, in reference to a famous casino of the same name. They work by simulating the lifetimes of individual particles, tracking their path through space and any interactions which occur until they are removed from the system. Typically a batch of particles, called a *generation*, is simulated with the same initial distribution. The information from each generation is used to refine the conditions of future generations in order to converge the simulation towards an accurate result. For example, the random distribution of new particles in each generation can be weighted by the distribution of emitted particles throughout the domain in previous generations, so that more particles from highly emitting regions will be tracked and vice versa. A major benefit of such methods is that the energy dimension can be treated as continuous, since each particle may have any value within the allowed range [44]. In addition, they are guaranteed to converge to the true solution given enough particles [45]. However, stochastic methods tend to be highly computationally expensive, and may require the simulation of an infeasibly large number of particles in order to converge, depending on the problem.

2.3 Full-order Angular Discretisation Methods for the Boltzmann Transport Equation

The angular dimensions of the BTE represent the direction components of the particles' momentum vectors. Discretisation of these dimensions involves generating a discrete set of basis functions which are capable of approximately representing the directions

in which particles may travel. The most commonly used angular discretisations for the BTE are discrete ordinates (S_N), spherical harmonics (P_N) and to a lesser extent wavelets, which are all discussed in this section.

2.3.1 Discrete Ordinates

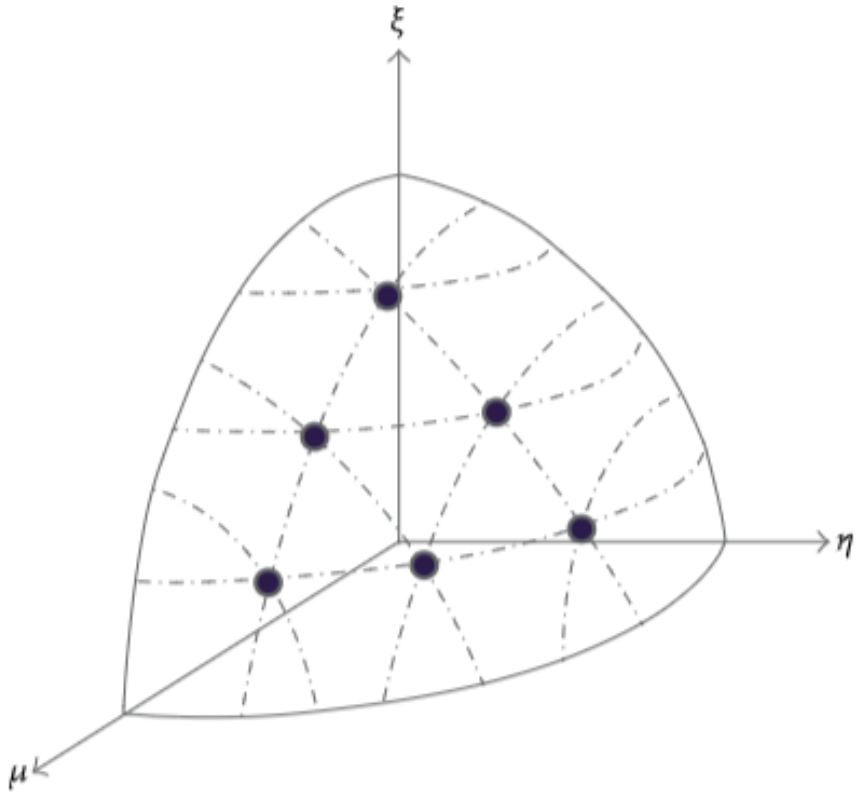
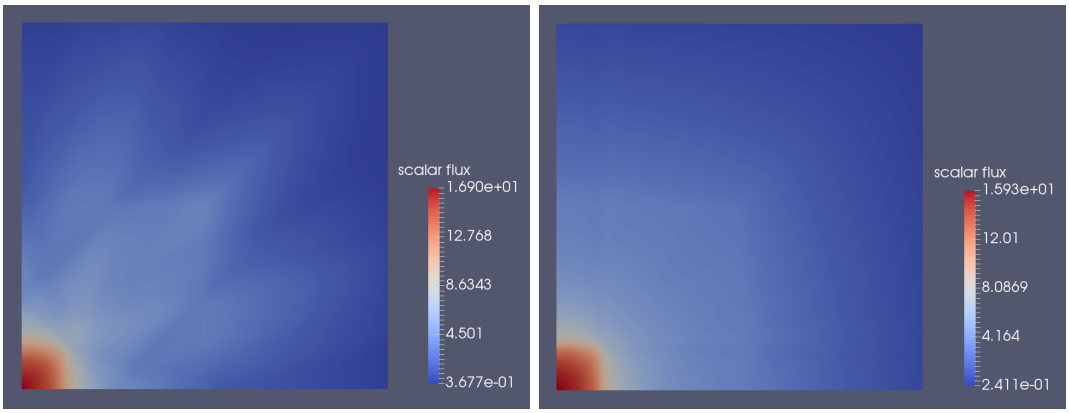


Figure 2.1: Angular basis vectors in an octant for the S_6 angular discretisation [1].

The method of discrete ordinates was first described by Chandrasekhar in 1950 [46]. It describes a class of quadrature schemes which convert the continuous variable of solid angle into a discrete set of angular vectors, and treat the angular flux as a delta distribution which is nonzero only in the directions of the angular vectors. There are many options for the arrangement and weights of each discrete vector, but here the level symmetric quadrature, referred to as S_N, will be used exclusively [47]. N refers to the order of the basis, which has $N(N+2)$ basis vectors, halved in two dimensions by symmetry. Figure 2.1 shows the arrangement of basis functions in one octant for the S_6 angular discretisation, and the other octants are reflections on the coordinate planes. A notable downside of this method is the possibility for ray effects - in a low-scattering medium with insufficient angular resolution, the angular flux may exhibit unphysical

fluctuations due to its inability to travel in directions other than those described by the angular vectors. Ray effects can be eliminated by increasing the number of angular vectors, but doing so also increases the computational cost of finding a solution. An example of this can be seen in figure 2.2, which depicts S_6 and S_{30} solutions to the Watanabe-Maynard extrapolation problem, as described in section 3.7.2. The S_6 solution exhibits significant ray effects, while in the S_{30} solution they are present but relatively minor. As the number of angular vectors approaches infinity, S_N is guaranteed to converge towards the exact solution [48].



(a) S_6 solution (b) S_{30} solution

Figure 2.2: Discrete ordinates solutions to the Watanabe-Maynard extrapolation problem

Unlike many other angular discretisations, the S_N method has a well-defined direction for each angular basis function. As a result, it is possible to determine for each ordinate which spatial elements are 'upstream' and 'downstream' of each other - where downstream elements receive advected flux from elements which are upstream of them. This enables the implementation of an efficient sweep-based solver using first-order upwinding, which traces the path of each ray through the domain element by element, and propagates flux in the direction it is travelling [49–51]. Sweep-based solvers are discussed further in section 2.5.1. The S_N quadrature is used by several major neutron transport solvers, including DENOVO [52], Rattlesnake [53] and DRAGON [54].

2.3.2 Spherical Harmonics

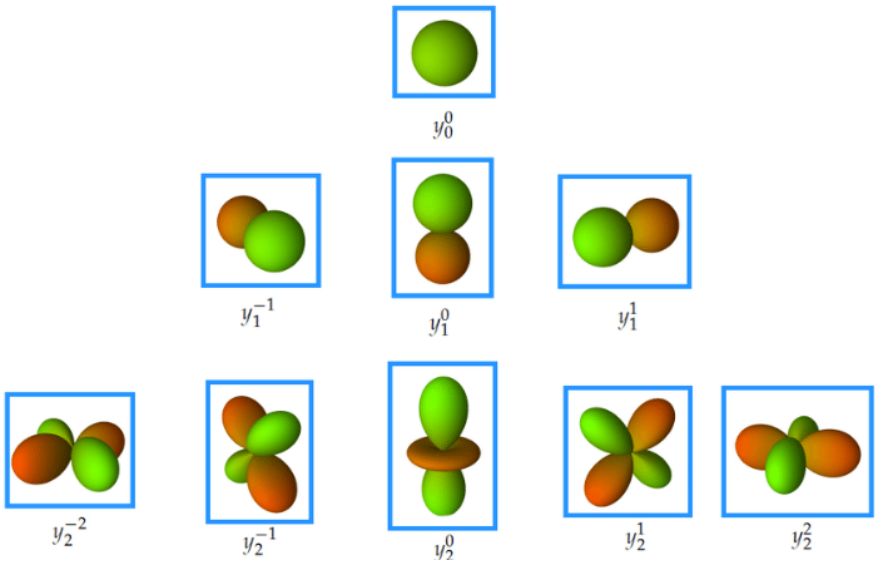


Figure 2.3: The spherical harmonic functions for $m \in \{-2, -1, 0, 1, 2\}$ and $l \in \{0, 1, 2\}$ [2].

The spherical harmonic method (P_N) uses Laplace's spherical harmonic functions $Y_l^m(\theta, \phi)$ as basis functions to represent the angular dimension. Figure 2.3 depicts these functions for $m \in \{-2, -1, 0, 1, 2\}$ and $l \in \{0, 1, 2\}$ [2]. They represent a complete, orthonormal basis set on the sphere, and as such have the ability to accurately represent any angular flux distribution if sufficient functions are utilised. They do not suffer from ray effects as the S_N method does, and may be more efficient to solve when few functions are utilised. However, the P_N method has several disadvantages. Gibbs oscillations can occur near discontinuities in physical properties, which introduces error and can potentially lead to unphysical negative values [55]. In addition, highly anisotropic flux distributions may require a very large number of spherical harmonic functions to accurately represent. Compounding this issue, the P_N functions are highly coupled and so models which use them scale poorly with high angular resolution. Finally, implementing vacuum boundary conditions is difficult for P_N approximations because each function spans the sphere and only the components pointing out of the domain are removed, whereas it is trivial when using the S_N discretisation as each ray represents a single direction which either leaves the domain or doesn't [56]. Some major neutron transport solvers use the P_N discretisation, including Rattlesnake [53] and EVENT [35].

2.3.3 Wavelets

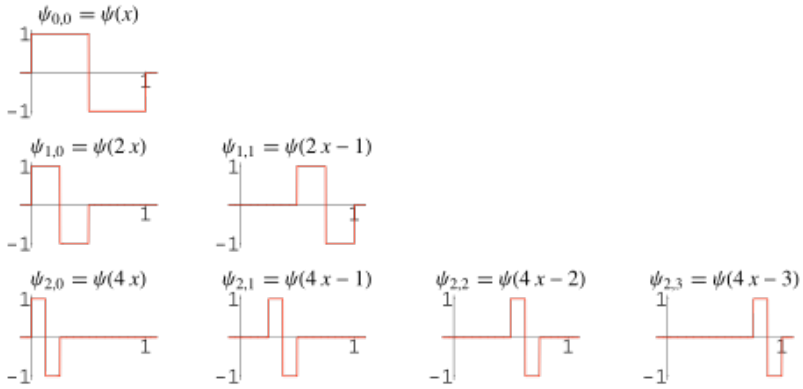


Figure 2.4: The first three Haar functions, with offsets [3].

Wavelet methods represent angular flux distributions as a linear combination of wavelet functions, which begin and end at zero and have some non-zero oscillation about the origin. The simplest and most commonly used for radiation transport are Haar wavelets, first described by Haar in 1910 [57]. They may be be defined on the interval $[0, L)$ as,

$$\Psi(x) = \begin{cases} 1 & 0 \leq x < \frac{L}{2} \\ -1 & \frac{L}{2} \leq x < L \\ 0 & \text{otherwise.} \end{cases} \tag{2.18}$$

A linear combination of these functions (along with shifted versions) can be used to approximate any continuous, real function, with the accuracy of the approximation increasing as additional wavelets are added [58]. Due to the discontinuous nature of the wavelets, they are also capable of representing discontinuous functions in some cases. While initially described on a line, equivalent functions on the sphere can readily be constructed. A major advantage of the Haar wavelet discretisation is that, since the functions are hierarchical, they enable the implementation of angular adaptivity [59]. However, wavelet discretisations are a relatively new technique, and not as widespread or well-studied in the field of radiation transport as the aforementioned angular discretisations. They have been implemented in the FETCH neutron transport solver [60], and at least one experimental code [49].

2.4 Full-order Spatial Discretisation Methods for the Boltzmann Transport Equation

The spatial dimensions of the BTE represent the spatial distribution of flux inside the domain. Discretising these dimensions typically involves dividing space into discrete partitions, and representing the spatial distribution within each partition with a finite number of degrees of freedom. In order to model the advection of particles through space, each partition must communicate with its neighbours. A wide variety of discretisations have been applied to the spatial dimensions of the BTE, including the relatively common finite volume, finite difference and finite element methods, and the less frequently used spectral and wavelet-based methods. Each of these methods will be described in this section.

2.4.1 The Finite Difference Method

The finite difference method (FDM) is the oldest method of numerically solving partial differential equations, and as such it was employed by early neutron transport solvers to discretise the spatial dimensions of the BTE [61]. The FDM divides the domain into regularly spaced points, and approximates terms in the governing equation using the finite differences between points. For example, first order derivatives can be approximated by the difference quotient when the distance between points is relatively small:

$$\lim_{dx\to 0} \frac{dF}{dx} = \frac{F(x+dx) - F(x)}{dx} \implies \frac{dF}{dx} \approx \frac{F(x+dx) - F(x)}{dx}. \tag{2.19}$$

This enables the solutions at each point to be formulated as algebraic equations.

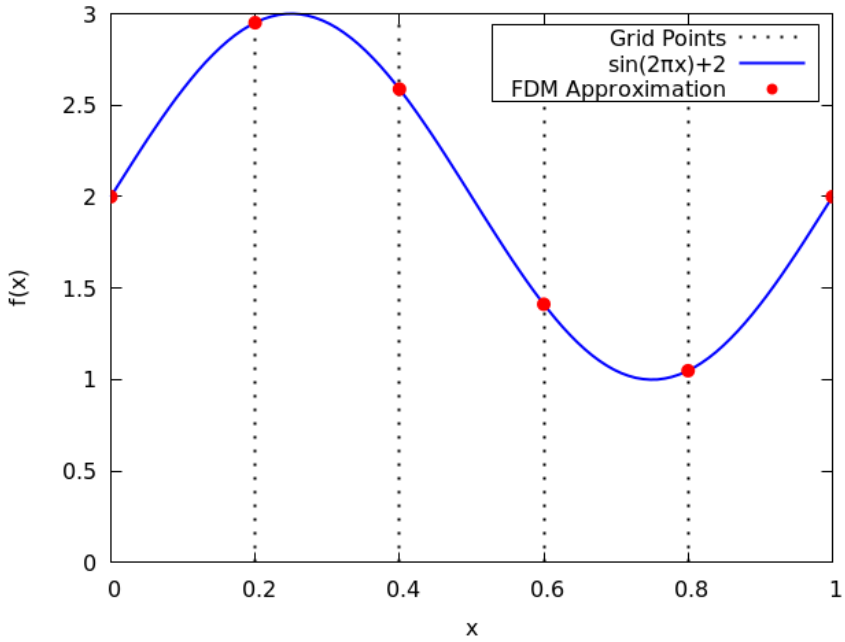


Figure 2.5: A finite difference approximation to $sin(2\pi x) + 2$ with 6 grid points.

Figure 2.5 depicts a one-dimensional space discretised with the FDM using 6 points, a continuous solution function $sin(2\pi x) + 2$, and a finite difference representation of the solution function. In this example, the finite difference representation is exact at each grid point, since the solution function is known rather than calculated numerically from a differential equation. The figure is merely intended to show how the FDM discretises the domain and defines approximate solutions, to be contrasted with figures 2.6 and 2.7.

The FDM typically produces sparse matrices, and so it tends to be computationally efficient compared to other methods. It is also relatively simple to implement and extend to higher-order accuracy. However, the FDM is usually restricted to operating on structured grids, and as such it is not conducive to rounded or otherwise irregular geometries, which are found frequently in nuclear reactors. This structured nature also makes localised grid refinement difficult. In addition, the FDM is not guaranteed to be conservative and special care must be taken to ensure that this property holds. Most of these issues can be overcome with some effort, but the efficiency benefits of the FDM tend to be reduced or lost in the process. Recent work on the FDM includes the development of acceleration techniques based on coarse mesh finite difference (CMFD) models, which have been applied to a wide range of solvers [62–64].

2.4.2 The Finite Volume Method

The finite volume method (FVM) divides the domain into non-overlapping volumes, and represents the solution in terms of the mean value over each volume, creating a piecewise constant solution over the domain.

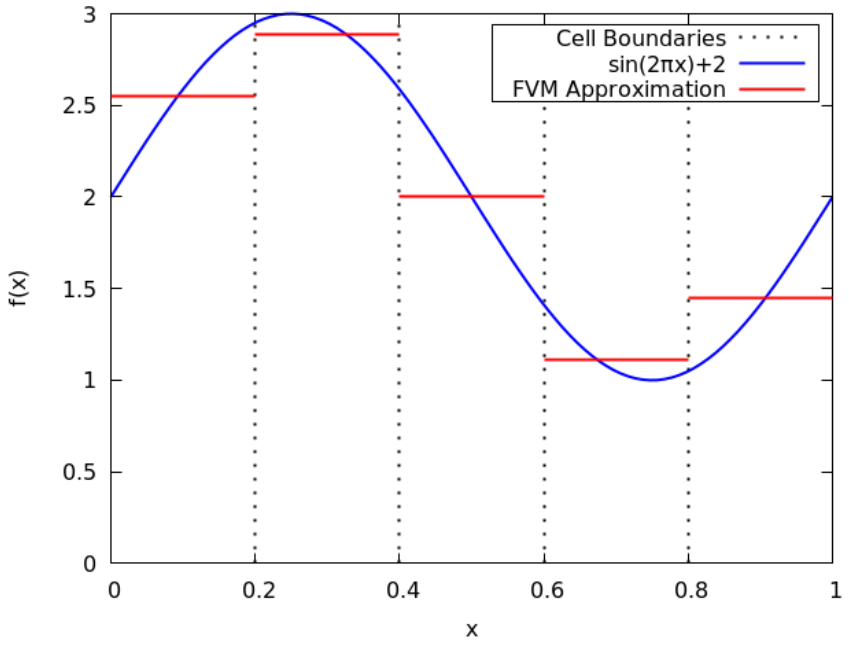


Figure 2.6: A finite volume approximation to $sin(2\pi x) + 2$ with 5 cells.

For example, figure 2.6 depicts a one-dimensional space discretised using the FVM, a continuous solution function $sin(2\pi x)+2$, and an approximation to the solution function constructed with the FVM. Communication between volumes is achieved by converting volume integrals to surface integrals using the divergence theorem, and evaluating them at the surface of each finite volume. These methods are guaranteed to be conservative, as the flux entering a given volume through its surface is equal to the flux leaving adjacent volumes. In addition, finite volume methods are able to handle unstructured meshes, which are common in radiation transport problems. However, the method can become complicated and difficult to implement when dealing with such meshes. Finally, it can be difficult to implement boundary conditions which are easy if not trivial to implement in the other spatial discretisation schemes.

2.4.3 The Finite Element Method

The finite element method (FEM) originated in the works of Hrennikoff [65] and Courant [66] in the early 1940s, and was first applied to neutron transport by Ohnishi [67, 68] in 1971. In 1973, Reed described the process of using a triangular spatial mesh to solve the BTE for neutrons [69], and the following year Lesaint implemented a similar method in order to model neutron population distributions [70]. The FEM operates by dividing the spatial domain into small, geometrically simple elements, and approximating the solution over each element with piecewise functions called basis functions, which are each nonzero only in a localised region of the domain. This leads to a set of equations associated with each element, which can be used to find an approximate solution for the corresponding element, expressed in terms of the coefficients of the local basis functions. By solving the local equations for each element and combining the solutions, an approximation of the solution function across the entire problem domain is constructed.

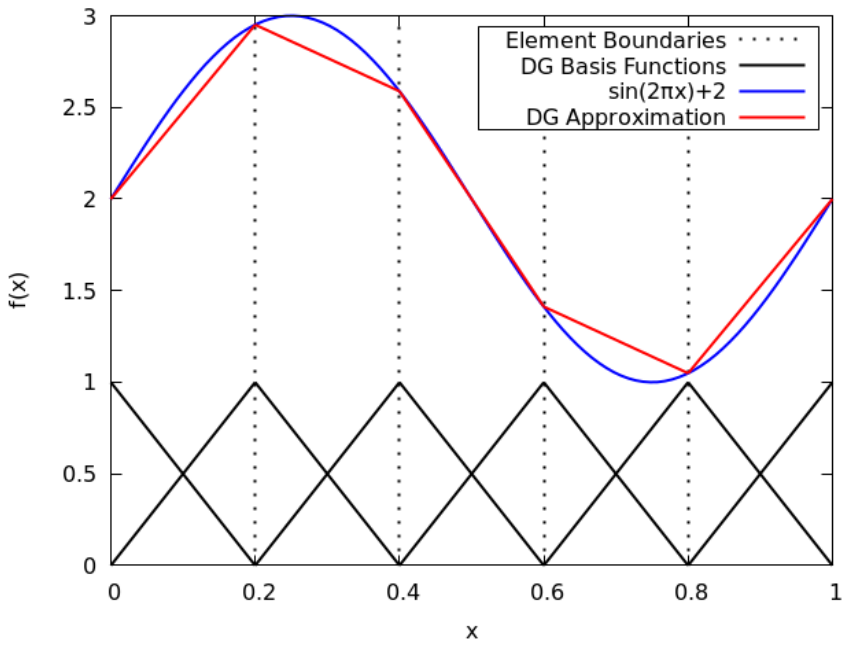


Figure 2.7: A discontinuous Galerkin finite element approximation to $sin(2\pi x) + 2$ with 5 elements.

For example, figure 2.7 depicts a one-dimensional space spanned by discontinuous linear basis functions, a continuous solution function $sin(2\pi x) + 2$, and an approximation to the solution function constructed with these basis functions. More complex basis functions can be used, which may increase accuracy if suitable functions are selected, but will increase the computational complexity of solving the problem. For example, the

use of quadratic polynomials as basis functions requires an additional node in the middle of each element, as solutions at three points are required in order to uniquely define a quadratic function. Unlike the FDM, the FEM allows nodes to be placed arbitrarily within a domain, enabling more complex geometries to be closely approximated. It is also relatively easy to increase the accuracy of the FEM approximation, and thus the model. This can be achieved by increasing either the number of elements in critical regions of the problem, or the order of the basis functions. The primary disadvantage of the FEM is that the matrices generated are not as well-structured as those generated by the FDM, and as such it can be more computationally expensive to solve. Recent work on the FEM includes [71], which used B-spline wavelet functions as spatial basis functions rather than the more commonly used polynomials. In [72], two procedures were developed which enabled the use of different approximations of the transport operator over the spatial domain. [73] used a finite element model to calculate thermal and mechanical displacement of a fuel assembly duct, then used a Monte Carlo code to calculate k_{eff} for the displaced assembly. [74] used a variation of the coarse mesh finite difference method mentioned in section 2.4.1 to accelerate a finite element model.

2.4.4 Spectral Methods

Spectral methods were introduced in 1969 by Orszag, and developed further by him and others in the following years [75, 76]. Like finite element methods, they represent solutions as a sum of basis functions, and solve problems by calculating the coefficients of each basis function. However, spectral methods differ in that the basis functions are, in general, nonzero throughout the domain. For example, a common spectral method uses a finite number of Fourier modes which span the spatial domain to discretise a solution in space [75]. This works because Fourier modes represent a complete, orthonormal basis, which theoretically allows any function to be represented. Many other bases can be used for this purpose, as long as they fulfil the same requirements. An early example is [77], which used Chebyshev polynomials to solve radiative transfer problems. Chebyshev spectral methods have also been used to solve heat transfer and fluid flow problems [78]. In 2017, [79] evaluated the use of Lobatto, Legendre and Chebyshev polynomials for solving the transport equation with a P_N angular discretisation. Lagrange polynomials have also been used to solve general two-term differential equations [80]. These methods

tend to excel at representing smooth functions, in which case their error can decrease exponentially with additional basis functions. However, they are much less effective in the presence of discontinuities, which can cause them to exhibit unphysical oscillations in the solution, known as Gibbs oscillations. This makes them poorly suited for general nuclear transport problems.

2.4.5 Wavelet Methods

Various types of wavelet have also been used as spatial basis functions, including Haar [81] and Daubechies [82] wavelets. These methods utilise hierarchical spatial representations, such that the lowest order basis functions span large spatial regions, and further basis functions have increasingly compact support. Such methods have numerous advantages. They are capable of producing well-conditioned bases which decrease in error rapidly, and correctly chosen wavelets can produce sparse matrices which enables efficient solutions. Their compact support and hierarchical nature also allows for the implementation of adaptive resolution. However, determining the correct wavelets to use for a particular problem can be challenging, and poorly-chosen wavelet bases lose the advantages which make their use appealing.

2.5 Solver Technologies for the Boltzmann Transport Equation

The result of discretising the BTE is a system of equations of the form $Ax = b$. Since the matrix A, source vector b and solution vector x can be extremely large, direct solution methods are not practical and iterative methods are generally used [42]. This section will discuss several techniques which have been applied to solve the BTE, some of which were used in this project.

2.5.1 Sweep-based Solution Methods

Sweep-based solvers are a class of iterative solution methods based on the Gauss-Seidel method, which are widely used for radiation transport applications [49, 51]. They work by generating a system of equations for each element, and solving them one by one, following the path of information flow throughout the domain [69]. This allows neutrons

to propagate through the entire domain in a single sweep, since the solution for each element relies only on coefficients that have already been solved in the current iteration, and coefficients from previous iterations are not needed. In the BTE, flux is transferred between elements via advection, and the incoming flux at each element is determined by its upstream neighbours. Elements must therefore be swept in an order which ensures that the upstream neighbours of each element are solved before the elements themselves. On unstructured grids, finding a suitable sweep order may be difficult. Algorithms for determining sweep orders exist, such as [4] which uses a list ordering heuristic, but cyclic dependencies are still problematic. For example, consider figure 2.8, which depicts elements in an unstructured mesh.

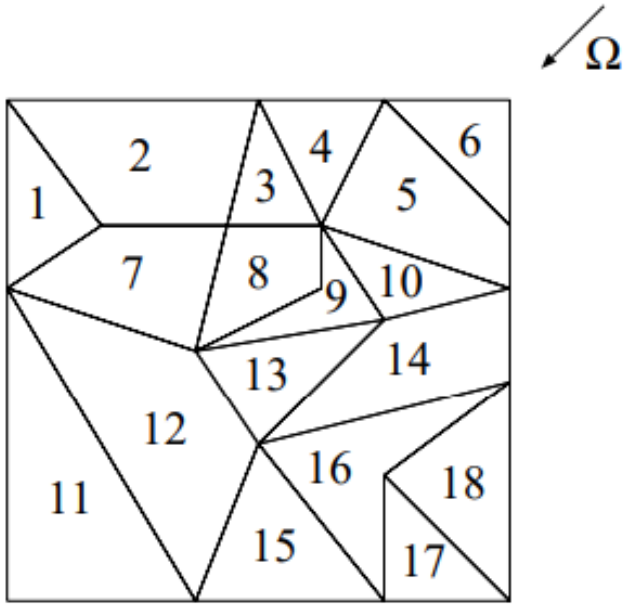
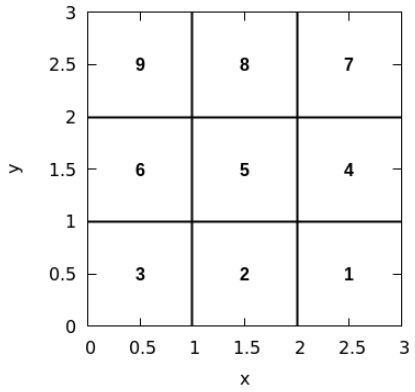
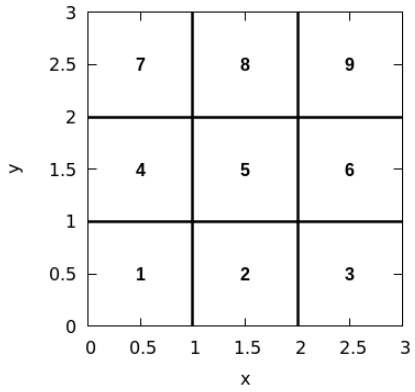


Figure 2.8: Elements in an unstructured mesh [4].

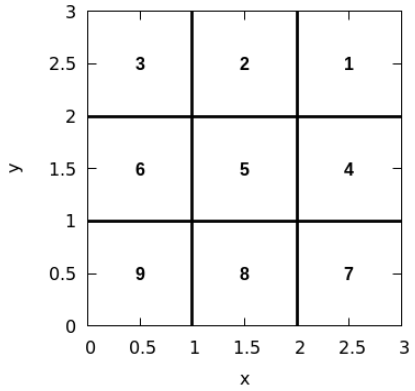
Flux travelling in the direction Ω must be transferred between elements 8 and 9 in both directions. This issue is typically resolved by using previous coefficients in one of the two cases, reducing the speed at which the solution converges. However, the process of determining an appropriate sweep order is trivial on structured grids. For the regular, rectangular 2D grids used in this project, each solve stage consists of four sweeps, as depicted in figure 2.9.



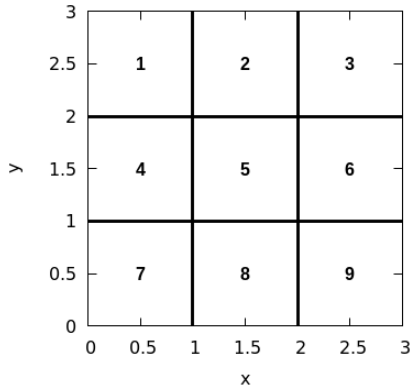
(a) Sweep order for rays travelling in -x, +y direction.



(b) Sweep order for rays travelling in +x, +y direction.



(c) Sweep order for rays travelling in -x, -y direction.



(d) Sweep order for rays travelling in +x, -y direction.

Figure 2.9: The order in which elements are visited in each sweep for an example problem with 3x3 square elements.

Since discrete ordinates have well-defined directions, each ray is swept individually in the order corresponding to its direction of travel. This ensures that flux will be correctly transported via advection, since the upstream neighbours of each element are guaranteed to be solved when they are required. For the standard angular POD method mentioned in section 1.1, this is not possible - since each basis function spans the sphere, each must be swept in all four directions for advection to be correctly calculated.

2.5.2 Multigrid Methods

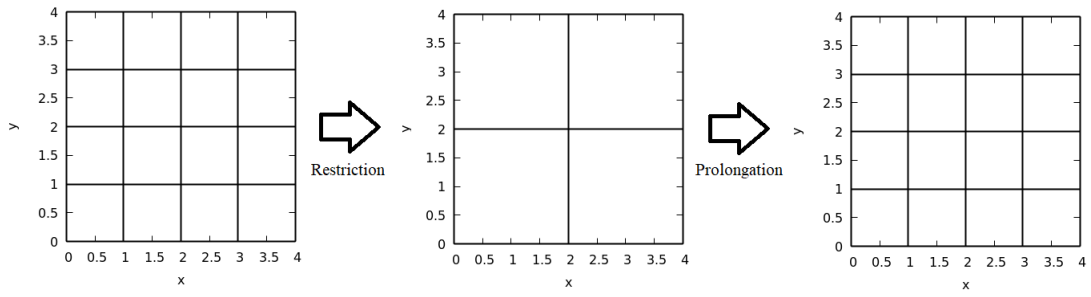


Figure 2.10: A simple three-stage multigrid V-cycle.

Multigrid methods are a class of iterative solvers which use a hierarchy of discretisations to improve their rate of convergence [83,84]. While many variations of such methods exist, they typically involve some combination of steps known as restriction and prolongation. Restriction is the process of projecting from a fine grid to a coarser grid, and prolongation is the reverse process of projecting from a coarse grid to a finer grid, as shown in figure 2.10. Iteration on a coarse grid reduces the low frequency error and is computationally relatively inexpensive, while solving on a fine grid is required to reduce the high frequency error and produce a more accurate solution overall, but is also more computationally expensive. By using multiple grids with varying resolutions, it is possible to produce a solution with the accuracy of a fine grid, but reduced computational cost. Consider a problem of the form,

$$Ax = b, \tag{2.20}$$

where A is an operator matrix, x is the exact solution, and b is a constant vector. For an approximate solution x_g on a fine grid, with error ϵ,

$$x_g = x + \epsilon, \tag{2.21}$$

and the residual error r is given by,

$$Ax_g - b = r. \tag{2.22}$$

Substituting equation 2.21 into equation 2.22, rearranging, then applying equation 2.20 gives,

$$A(x + \epsilon) - b = r \implies Ax - b + A\epsilon = r \implies A\epsilon = r. \tag{2.23}$$

Let ϵ_c be the error on the coarse grid, which when multiplied by the prolongation operator P gives the error on the fine grid,

$$\epsilon = P\epsilon_c. \tag{2.24}$$

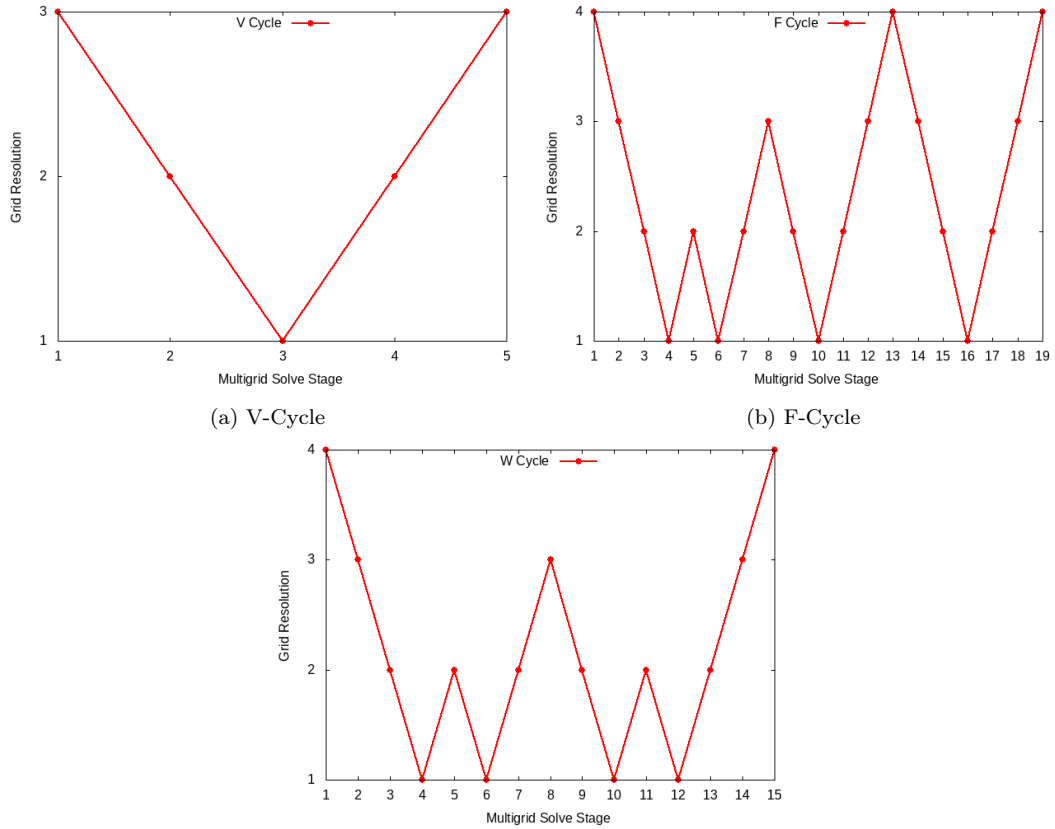
Making this substitution into equation 2.23 and premultiplying by the restriction matrix R to project the residual error and operator matrix onto the coarse grid gives,

$$RAP\epsilon_c = Rr. \tag{2.25}$$

The matrix RAP is the coarse grid operator matrix, which is denoted A_c, and the vector Rr is the coarse grid residual error, denoted r_c. This leads to the equation,

$$A_c\epsilon_c = r_c. \tag{2.26}$$

This equation allows a correction to the approximate solution x_g to be calculated rapidly on the coarse grid to minimise the residual error. The correction can then interpolated back up to the fine grid to adjust the solution, and then iterated further. This process is known as a V-cycle, since a graph of the resolution at each stage is shaped like a V. An example of this type of operation performed on a simple spatial domain is shown in figure 2.10, and a plot of the resolution at each stage of a V-cycle can be seen in figure 2.11a. More than two resolution levels may be used, in which case alternate procedures such as as W-cycles and F-cycles are also possible, as shown in figures 2.11c and 2.11b respectively. Each type of cycle represents a different ordering of restriction and prolongation operations, with its own advantages and disadvantages.



(a) V-Cycle

(b) F-Cycle

(c) W-Cycle

Figure 2.11: Examples of three common multigrid cycle procedures.

Multigrid methods have been applied to a wide range of problems across the field of numerical modelling. Early applications include solving the incompressible Navier-Stokes equations for high Reynolds number flows [85] and transonic viscous flows past airfoils [86], as well as solving the linearised Poisson-Boltzmann equation with applications to molecular biophysics [87]. More recently, an element agglomeration multigrid method with applications to the BTE was introduced [88], and multigrid methods were applied to model high voltage DC transmission lines [89] and the heat equation [90]. Multigrid methods can also be used to precondition Krylov solvers [91], which are discussed further in section 2.5.3. They have also been applied to train video models more efficiently [92].

2.5.3 Krylov Subspace Methods

For a matrix system of the form $Ax = b$, the residual r_n of an approximation x_n is given by $Ax_n - b = r_n$. Krylov subspaces are typically constructed by producing an approximation x_0 and finding the residual r_0. The Krylov subspace $\kappa_n(A, r_0)$ is the

linear subspace spanned by r_0 multiplied by successive powers of A,

$$\kappa_n(A, r_0) = \text{span}\{r_0, Ar_0, A^2r_0, ..., A^{n-1}r_0\}. \tag{2.27}$$

Krylov methods use this subspace as a basis for calculating corrections to the initial guess in order to refine their solution. Many algorithms based on Krylov subspaces have been employed, including the conjugate gradient method [93] and GMRES [94], among others. They typically work by constructing an orthogonal basis in the Krylov subspace, then following some iterative process which attempts to minimise the residual. For example, the generalised minimal residual method (GMRES) approximates the solution x_n at iteration n as a vector in the Krylov subspace κ_n which minimises the L^2-norm of the residual at each iteration,

$$\min_{x_n \in \kappa_n(A,b)} \|b - Ax_n\|_2. \tag{2.28}$$

Krylov subspace methods are frequently used to solve the system $Ax = b$ in many fields. Early implementations of Krylov subspace methods in neutron transport include the work of Patton, who implemented the GMRES method to solve S_N neutron transport problems in 1996 [94] and continued to develop the method thereafter [95].

2.6 Optimisation Techniques for the Boltzmann Transport Equation

2.6.1 Angular Adaptivity

Adaptivity is an attempt to improve the efficiency of numerical models by focusing resolution in a problem-specific manner, rather than uniformly throughout the domain. It can be applied to either angular or spatial resolution, and has also been applied to both simultaneously.

The goal of angular adaptivity is to increase the angular resolution in regions where it will be of the most benefit, such as in the presence of highly anisotropic flux distributions, while keeping it at a minimum in regions where additional resolution will not make a significant difference, such as highly scattering media with near-isotropic flux distributions.

Hierarchical angular discretisations are generally used for this purpose, as they make the implementation of adaptivity significantly easier. In this context, a hierarchical basis set is one in which increasing the order of the discretisation adds new basis functions but does not change the existing ones, enabling the existing coefficients to be retained. For example, increasing the order of spherical harmonic and Haar wavelet discretisations is simply a matter of including more harmonic functions or wavelets, respectively. By contrast, increasing the angular resolution of the S_N discretisation changes the existing angular vectors, making the implementation of angular adaptivity difficult since existing coefficients cannot simply be carried forwards to the next adaptive step.

In [39], an adaptive method was developed for the BTE using octahedral and hexahedral wavelet bases, and more recently [96] implemented angular adaptivity using the Haar wavelet angular discretisation. It has also been applied to the P_N angular discretisation in 2005 [97], and more recently to a filtered P_N basis [98]. An adaptive P_N algorithm based on a variational method is presented in [99]. [100] developed an adaptive algorithm based on a novel angular quadrature, which split the faces of an octahedron into triangles and projected the triangles onto the unit sphere. Angular adaptivity has also been implemented in other contexts, such as the analysis of structural and acoustic vibrations in a wave-based model [101], and the simulation of semiconductor devices using adaptive wavelets [102]. In addition to angular adaptivity which aims to reduce the overall error of a solution, goal-based angular adaptivity has also been developed. These methods aim to minimise the error within a particular region, such as a detector, rather than throughout the whole domain. For example, [103] presented a goal-based adaptive algorithm for the spherical harmonic angular discretisation, and compared it to standard angular adaptivity. [59] presented an implementation of goal-based adaptivity for the Haar wavelet angular discretisation, and [9] used a low-order spherical harmonic model to determine the appropriate distribution of angular resolution for a Haar wavelet model of the same problem. Kópházi and Lathouwers implemented adaptivity using a basis which was constructed by subdividing each angular octant into increasingly smaller triangles [49]. Angular adaptivity has also been implemented for the S_N discretisation, which required the development of techniques to transfer angular flux between different quadratures since S_N is not hierarchical [104].

2.6.2 Spatial Adaptivity

Spatial adaptivity is a similar concept to angular adaptivity, except that instead of focusing angular resolution in particular areas to improve performance, it distributes spatial basis functions for the same purpose. This typically means that resolution will be focused in areas with high spatial variance in the solution, while regions where the solution varies more gradually over space will have lower resolutions. There are two general approaches to spatial adaptivity - either the spatial mesh may be adjusted by adding or removing nodes in a process known as mesh refinement or h-adaptivity, or higher order basis functions may be used in particular regions, known as p-adaptivity.

Mesh refinement typically works by using an error metric to determine where new nodes would be most beneficial, and altering the spatial mesh by splitting the elements in that location into multiple new elements. Some of the nodes will be co-located with nodes on the previous mesh, and can retain their coefficients, wheras the newly added nodes will require interpolation to find initial values. There can be complications with this method, including the need to re-calculate some properties - for example, the order in which elements in the mesh are swept may change. In addition, mesh refinement is more difficult for some implementations of structured grids, such as the FDM. Despite these issues, h-adaptivity represents an effective option in many cases. Examples of mesh refinement-based spatially adaptive methods include [105, 106], which both developed mesh refinement algorithms for the spherical harmonic angular discretisation. In [107], it was applied to a mesh which included curved element boundaries, and [108] used it to solve the even-parity form of the BTE with a P_N angular discretisation.

P-adaptivity involves increasing the order of the polynomials used as basis functions in a particular region. For example, a problem could be solved with constant basis functions in some regions, linear in others, and quadrilateral functions in the most spatially varying regions. A major advantage of this method is that the mesh does not have to be adjusted, which can make it relatively simple to implement, and allows it to be used on structured grids. Examples of p-adaptivity include [109], which implemented it on a discontinuous finite element mesh in one dimension with discrete ordinates in space; and [110, 111], which both used it for compressible flow problems in three dimensions.

2.6.3 Reduced Order Modelling

In recent years, reduced order models have been developed which fulfil a similar purpose to adaptivity - reducing the computational cost of solving a problem while minimising the resultant error. These models rely on formulating new sets of basis functions which can be used to describe the solution, with fewer degrees of freedom than the original. This will inevitably introduce error, as an approximation is made, but if the basis sets are well-chosen for the problem in question this error can be minimised. As with adaptivity, reduced order modelling techniques can be applied to the spatial or angular dimensions, or potentially both at the same time.

There are several techniques for constructing reduced basis sets, which can be categorised as either *a priori* or *a posteriori* methods. *A priori* methods such as proper generalised decomposition aim to generate reduced bases without solutions to a full model [112], whereas *a posteriori* methods such as the empirical interpolation method [113, 114], neural networks [115] and POD use existing solutions to produce a reduced basis for use in further simulation. Many other model order reduction strategies have been developed, including the Volterra [116] and Fourier [117] series expansions, space mapping [118], the Kriging [119, 120] and harmonic balance [121] methods, and radial basis functions [120].

ROMs have been applied to a vast range of problems, from signal analysis and pattern recognition [122], to statistics [123], and geophysical fluids [124]. They have been particularly successful in simulating fluid flows following the early work of Lumley [125], and have been applied to turbulent pipe flows [126], wakes behind a cylinder [127], flows across air foils [128], the mixing of fluid layers [129], thermal currents [130, 131] and ocean models [132]. Neural network-based ROMs were proposed for general nonlinear systems as early as 1995 [133], and have since been applied to various problems, including the modelling of biomass fast pyrolysis in a fluidised bed reactor [134] and unsteady fluid flows [135]. ROMs have also been developed that use and apply basis functions over a domain decomposition - that is, they split the problem into multiple spatial partitions, and form optimised basis functions for each partition. Such models have been used to solve advection-diffusion problems [136], Maxwell's equations [137], and the Stokes equations [138], among many other applications. Mixed approaches have also been described using both high-order methods and reduced order models over a domain

decomposition, applying ROMs over regions that they can resolve well and utilising high order methods elsewhere. Applications include solving the Laplace equations [139] and, more recently, the compressible Euler equations [140]. These systems are significantly smaller, and thus more efficient to solve, than high dimensional models.

This project will focus on the method of proper orthogonal decomposition, applied to the angular dimension of the BTE. In 1873, Beltrami derived the singular value decomposition (SVD) [141]. Though originally presented in bilinear form, the method allows matrices to be decomposed into three component matrices which each contain information about the data in the original matrix. The next relevant development came in 1901, when Pearson introduced the technique of principal component analysis [142]. This presents a method for finding an optimised set of basis vectors, known as the principle components, to represent a particular set of data. It can be shown that the principle components of a set of data are given by the SVD of the data matrix [143]. This enables the technique of POD, which uses the SVD to decompose a matrix of full order solutions, known as the snapshot matrix. This generates a set of basis vectors which are optimised for representing the data contained in the snapshot matrix. While the set of basis functions generated can be large, most of the variance in the snapshot data can typically be captured by a relatively small number of basis functions. The rest can be discarded, as described in section 3.4. In the field of radiation transport, among others, POD is often used to reduce the degrees of freedom of spatial dimensions. In this context, snapshots are vectors containing the coefficients at each spatial node, with each snapshot typically taken at a different time or with varying material properties [144]. [11] instead populated the snapshot matrix with angular flux vectors taken from full order model solutions, and thus generated optimised angular basis functions capable of efficiently representing the angular flux distributions which they have been produced for. As previously mentioned, this method was effective at reducing the degrees of freedom required to represent the angular dimension, but it caused solver instability which reduced its effectiveness. The primary focus of this research was to fix the solver instability of angular POD and improve its efficacy.

Recent developments include the application of POD to model transient heat conduction in problems with temperature-dependent thermal conductivity [145], turbulent supersonic

jets [146], acoustic waves inside enclosures [147], incompressible magnetohydrodynamics [148], and many other problems. POD-based methods have also been combined with neural networks to model elasticity and plasticity [149], unsteady flows in a combustion problem [150], the viscous Burgers equation [151], and in detecting and quantifying structural damage [152].

In the field of nuclear engineering, ROMs have begun to make progress towards providing efficient and accurate predictions for both reactor physics and shielding problems. Early work in reactor physics includes [153], which applied POD to resolve the spatial dependence of reactor physics eigenvalue problems, and [154], which used it to perform transient analysis of accelerator driven systems. A comparison of POD to modal methods in transient analysis is given in [144], and the use of ROMs to resolve sub-channels in lead cooled fast reactors is developed in [155]. Further developments include the application of POD to the angular dimension of the BTE [11, 156, 157], space-angle ROMs for radiative heat transfer [158], and the use of range-finding algorithms for the linear transformation of parameters in multiphysics problems [159]. POD has also been applied to resolve the angular dependence of the neutron transport equation [11], which was followed by similar work to resolve the angular dimension in radiative heat transfer problems [160]. More recent work has applied POD to non-linear feedback effects within lead-cooled reactors [161], fuel burnup within benchmark systems [162], molten salt reactor analysis [163], natural thermohydraulic circulation [164] and reactor power distributions [165]. Many articles have also considered the problem of control rod movement using various ROMs, including early work based on proper generalised decomposition [166], and recent publications which employed POD [167], the empirical interpolation method [168], and neural networks [169].

2.6.4 Adaptive Reduced Order Models

Reduced order models which are hierarchical in nature allow for the implementation of adaptivity, since additional resolution can be added without disrupting the existing solution. This provides a significant advantage, as adaptivity can increase the efficiency even further than the model order reduction alone. In particular, if a hierarchical ROM is constructed from a non-hierarchical full order discretisation, then adaptivity can be implemented relatively easily in a context where it was previously difficult. This is the

case with POD models based on S_N full order models, which are the subject of this work. Angular adaptivity is difficult to implement for the full order model because increasing the order of the angular discretisation completely redistributes the ordinates rather than merely adding new ordinates, as in a hierarchical system. However, the construction of POD bases which are hierarchical enables angular adaptivity to be implemented in the reduced order model.

Many adaptive ROMs based on POD have been implemented, due to its hierarchical nature and widespread usage. Similar techniques have been applied to a wide range of physical systems. [132] combined spatial mesh refinement with a POD-based ROM in the context of ocean modelling. In [170], an adaptive POD model for the Reynolds-averaged Navier Stokes equations was produced, which included additional basis functions based on an a posteriori error estimator. [171] used adaptive POD to model population balance in a simulation of the crystallisation process. Other methods of producing adaptive reduced order models have also been demonstrated. For example, [163] recently produced an adaptive ROM of a molten salt reactor based on neural networks, and [172] used an adaptive finite element mesh along with a neural network to generate a correction term, which was applied to the coarse mesh to produce the final reduced order model.

Chapter 3

A Discontinuous Angular Reduced Order Model for the Boltzmann Transport Equation

This chapter describes a novel reduced order model for the angular discretisation of the BTE, known here as Discontinuous Proper Orthogonal Decomposition (DPOD). It represents a modification of the angular Proper Orthogonal Decomposition method described by Buchan in 2015 [11], which will henceforth be known simply as POD. The motivation for this modification was an attempt to solve the issue of solver instability which affects POD. Specifically, the number of iterations required to converge to a solution using the POD method oscillates drastically as the number of basis functions used is varied. In an attempt to fix this issue, a new type of angular POD basis function was developed. In contrast to the POD functions which span the entire angular domain, basis functions with compact support were developed. These functions are nonzero only in a particular section of the sphere - in this project, the basis functions were restricted to supporting individual octants, though in principle any partitioning of the angular domain could be used. A complete basis spanning the sphere was then constructed by a linear combination of basis functions from every octant. It will be demonstrated that partitioning the angular domain in such a manner eliminates the solver instability of the POD method, and additionally decreases the angular flux error for a given basis size in many cases. A further advantage of this method is that it allows for the implementation

of a more sophisticated form of angular adaptivity, as the number of basis functions utilised can differ between angular octants as well as spatial elements. It also enables the use of a more efficient sweep-based solver - since the angles represented by each basis function lie within a single sweep direction, it should be possible to produce a solution by sweeping one direction per basis function rather than all four. This possibility is discussed in more detail in section 1.2, along with an explanation of why it was not implemented in this project.

The chapter is organised as follows: First, the steady state, mono-energetic BTE is defined. A complete derivation of the angular POD method described by Buchan is then presented. Next, the DPOD method is derived in detail, along with a simple yet effective implementation of angular adaptivity using the DPOD basis functions. Numerical results for a range of problems are then presented to demonstrate the effectiveness of DPOD compared to POD and the full order discrete ordinates method. Finally, conclusions are drawn regarding the effectiveness of DPOD, based on the numerical results presented.

3.1 The Steady State, Mono-Energetic Boltzmann Transport Equation

The steady state, mono-energetic BTE describes the angular flux $\Psi(\vec{r}, \Omega)$ in direction Ω at position $\vec{r}$:

$$\Omega \cdot \nabla \Psi(\vec{r}, \Omega) + \Sigma_t(\vec{r}) \Psi(\vec{r}, \Omega) = q_{ex}(\vec{r}) + q_s(\vec{r}, \Omega' \rightarrow \Omega). \tag{3.1}$$

The first term in equation 3.1 is the advection operator $\Omega \cdot \nabla$, which can be written in Cartesian coordinates as:

$$\Omega \cdot \nabla = (\Omega_x, \Omega_y, \Omega_z) \cdot (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}) = \Omega_x \frac{\partial}{\partial x} + \Omega_y \frac{\partial}{\partial y} + \Omega_z \frac{\partial}{\partial z}, \tag{3.2}$$

where Ω_x, Ω_y and Ω_z are the components of the unit vector Ω in Cartesian space. The second term is the removal operator, which accounts for losses from both absorption and scattering. q_{ex} denotes the isotropic external source, and the scattering source term

q_s is given by:

$$q_s(\vec{r}, \Omega' \rightarrow \Omega) = \int_{\Omega'} \Sigma_s(\vec{r}, \Omega' \rightarrow \Omega)\Psi(\vec{r}, \Omega')d\Omega', \tag{3.3}$$

and represents the scattering into the angle Ω from all angles Ω'. The macroscopic scattering, absorption and total removal cross sections are given by Σ_s, Σ_a and Σ_t respectively, and are related by:

$$\Sigma_t = \Sigma_s + \Sigma_a. \tag{3.4}$$

Throughout this work, variables in brackets will denote continuous dependence, so that $\Psi(\vec{r}, \Omega)$ is continuous in space in angle and $\Psi(\vec{r})$ is continuous in space but discretised in angle. ψ will be used to denote angular fluxes which are discretised in both space and angle.

3.2 Angular Discretisation of the Boltzmann Transport Equation

To discretise the angular dimension of equation 3.1, the angular flux $\Psi(\vec{r}, \Omega)$ is approximated by a sum of N_a angular basis functions $\mathcal{G}_j(\Omega)$ multiplied by the coefficients $\Psi_j(\vec{r})$:

$$\Psi(\vec{r}, \Omega) \approx \sum_{j=1}^{N_a} \mathcal{G}_j(\Omega)\Psi_j(\vec{r}). \tag{3.5}$$

The approximation in equation 3.5 is inserted into equation 3.1, which is then weighted and integrated over all angles. The Bubnov-Galerkin method is applied, which uses the angular basis functions as weights, this time denoted $\mathcal{G}_i(\Omega)$. The angularly discretised form of the BTE can therefore be written as:

$$\sum_{j=1}^{N_a} \left(\Big(\int_\Omega \mathcal{G}_i(\Omega)\Omega \cdot \nabla \mathcal{G}_j(\Omega)d\Omega + \int_\Omega \mathcal{G}_i(\Omega)\Sigma_t(\vec{r})\mathcal{G}_j(\Omega)d\Omega \Big)\Psi_j(\vec{r}) \right) - \int_\Omega \mathcal{G}_i(\Omega)q_s(\vec{r}, \Omega' \rightarrow \Omega)d\Omega = \int_\Omega \mathcal{G}_i(\Omega)q_{ex}(\vec{r})d\Omega \quad \forall\ i \in \{1, N_a\} \tag{3.6}$$

The Cartesian components of the advection operator $\Omega \cdot \nabla$, given by equation 3.2, are substituted into equation 3.6:

$$\sum_{j=1}^{N_a} \Bigg(\Big(\int_\Omega \mathcal{G}_i(\Omega)\Omega_x \frac{\partial}{\partial x}\mathcal{G}_j(\Omega)d\Omega + \int_\Omega \mathcal{G}_i(\Omega)\Omega_y \frac{\partial}{\partial y}\mathcal{G}_j(\Omega)d\Omega + \int_\Omega \mathcal{G}_i(\Omega)\Omega_z \frac{\partial}{\partial z}\mathcal{G}_j(\Omega)d\Omega + \int_\Omega \mathcal{G}_i(\Omega)\Sigma_t(\vec{r})\mathcal{G}_j(\Omega)d\Omega \Big)\Psi_j(\vec{r}) \Bigg) - \int_\Omega \mathcal{G}_i(\Omega)q_s(\vec{r},\Omega' \rightarrow \Omega)d\Omega = \int_\Omega \mathcal{G}_i(\Omega)q_{ex}(\vec{r})d\Omega, \quad \forall\ i \in \{1, N_a\}. \tag{3.7}$$

Equation 3.7 can be written in matrix form as:

$$A_x \frac{\partial \Psi(\vec{r})}{\partial x} + A_y \frac{\partial \Psi(\vec{r})}{\partial y} + A_z \frac{\partial \Psi(\vec{r})}{\partial z} + H\Psi(\vec{r}) = Q(\vec{r}) \tag{3.8}$$

Where A_x, A_y, A_z and H are matrices of size $N_a \times N_a$. $\Psi(\vec{r})$ and $Q(\vec{r})$ are vectors of size N_a, containing the coefficients of the angular basis functions and the contributions of the source term, respectively. The components of each matrix and vector at row i, column j are given by:

$$\begin{aligned}
A_x^{ij} &= \int_\Omega \mathcal{G}_i(\Omega)\Omega_x\mathcal{G}_j(\Omega)d\Omega \\
A_y^{ij} &= \int_\Omega \mathcal{G}_i(\Omega)\Omega_y\mathcal{G}_j(\Omega)d\Omega \\
A_z^{ij} &= \int_\Omega \mathcal{G}_i(\Omega)\Omega_z\mathcal{G}_j(\Omega)d\Omega \\
H^{ij} &= \int_\Omega \mathcal{G}_i(\Omega)\Sigma_t(\vec{r})\mathcal{G}_j(\Omega)d\Omega - \int_\Omega \mathcal{G}_i(\Omega)q_s(\vec{r},\Omega' \rightarrow \Omega)d\Omega \\
Q^i &= \int_\Omega \mathcal{G}_i(\Omega)q_{ex}(\vec{r})d\Omega
\end{aligned}$$

A vector of matrices A is defined as $A = (A_x, A_y, A_z)$, which allows equation 3.8 to be rewritten as:

$$(A \cdot \nabla + H)\Psi(\vec{r}) = Q(\vec{r}). \tag{3.9}$$

This form of the equation is fully discretised in angle, but has yet to be discretised in space.

3.3 Spatial Discretisation of the Boltzmann Transport Equation

The discontinuous Galerkin finite element method is applied to discretise the spatial dimensions of equation 3.9. The equation is converted to its weak form by weighting it with a set of N_s spatial basis functions $\mathcal{N}_i(\vec{r})$ and integrating over the volume of each element, V_e:

$$\Big(\int_{V_e} \mathcal{N}_i(\vec{r}) A \cdot \nabla dV_e \; + \int_{V_e} \mathcal{N}_i(\vec{r}) H dV_e \Big) \Psi(\vec{r}) = \int_{V_e} \mathcal{N}_i(\vec{r}) Q(\vec{r}) dV_e$$

$$\forall \; i \in \{1, N_s\}. \quad (3.10)$$

The divergence theorem is applied to the advection term in equation 3.10, splitting it into an integral over V_e and another over the element boundary Γ_e:

$$\Big(\int_{\Gamma_e} \mathcal{N}_i(\vec{r})(A \cdot \hat{n}) d\Gamma_e - \int_{V_e} \nabla \mathcal{N}_i(\vec{r}) \cdot A dV_e + \int_{V_e} \mathcal{N}_i(\vec{r}) H dV_e \Big) \Psi(\vec{r}) = \int_{V_e} \mathcal{N}_i(\vec{r}) Q(\vec{r}) dV_e,$$

$$\forall \; i \in \{1, N_s\}, \quad (3.11)$$

where $\hat{n}$ is the unit vector normal to the element boundary. The angularly discretised flux $\Psi(\vec{r})$ is approximated as a sum of the spatial basis functions $\mathcal{N}_j(\vec{r})$ multiplied by the coefficients $\Psi_j(\vec{r})$:

$$\Psi(\vec{r}) \approx \sum_{j=1}^{N_s} \psi_j \mathcal{N}_j(\vec{r}). \quad (3.12)$$

The Bubnov-Galerkin method is applied, which uses the same set of functions as both weights and basis functions. Inserting equation 3.12 into equation 3.11 gives:

$$\sum_{j=1}^{N_s} \Big(\int_{\Gamma_e} \mathcal{N}_i(\vec{r})(A \cdot \hat{n}) \mathcal{N}_j(\vec{r}) d\Gamma_e - \int_{V_e} \nabla \mathcal{N}_i(\vec{r}) \cdot A \mathcal{N}_j(\vec{r}) dV_e + \int_{V_e} \mathcal{N}_i(\vec{r}) H \mathcal{N}_j(\vec{r}) dV_e \Big) \psi_j = \int_{V_e} \mathcal{N}_i(\vec{r}) Q(\vec{r}) dV_e,$$

$$\forall \; i \in \{1, N_s\}. \quad (3.13)$$

First-order upwinding is used to calculate the flow at the element boundaries [49]. The boundary term must therefore be split into inflow and outflow components [50]:

$$\sum_{j=1}^{N_s} \int_{\Gamma_e} \mathcal{N}_i(\vec{r})(A \cdot \hat{n})\mathcal{N}_j(\vec{r}) d\Gamma_e \psi_j$$
$$= \sum_{j=1}^{N_s} \Big(\int_{\Gamma_e} \mathcal{N}_i(\vec{r})(A^{in} \cdot \hat{n})\mathcal{N}_j(\vec{r}) d\Gamma_e \psi_j^{in} + \int_{\Gamma_e} \mathcal{N}_i(\vec{r})(A^{out} \cdot \hat{n})\mathcal{N}_j(\vec{r}) d\Gamma_e \psi_j^{out} \Big),$$
$$\forall\ i \in \{1, N_s\}. \quad (3.14)$$

ψ_j^{out} is the outflow, given by the angular flux vector of the element in question; and ψ_j^{in} is the inflow, given by the angular flux vectors of the element's upstream neighbours. The matrices $(A^{in} \cdot \hat{n})$ and $(A^{out} \cdot \hat{n})$ are formed to pass the correct incoming and outgoing information, respectively, through the element's surface. For S_N, they are produced by simply retaining the negative and positive diagonal elements, respectively, of the matrix $(A \cdot \hat{n})$. In the general case, a Riemann approach can be employed [173]. Equation 3.14 is inserted into equation 3.13 to give the final form of the fully discretised equations:

$$\sum_{j=1}^{N_s} \Bigg(\Big(- \int_{V_e} \nabla \mathcal{N}_i(\vec{r}) A \mathcal{N}_j(\vec{r}) dV + \int_{V_e} \mathcal{N}_i(\vec{r}) H \mathcal{N}_j(\vec{r}) dV \Big) \psi_j$$
$$+ \int_{\Gamma_e} \mathcal{N}_i(\vec{r})(A^{in} \cdot \hat{n})\mathcal{N}_j(\vec{r}) d\Gamma_e \psi_j^{in} + \int_{\Gamma_e} \mathcal{N}_i(\vec{r})(A^{out} \cdot \hat{n})\mathcal{N}_j(\vec{r}) d\Gamma_e \psi_j^{out} \Bigg)$$
$$= \int_{V_e} \mathcal{N}_i(\vec{r}) Q(\vec{r}) dV,$$
$$\forall\ i \in \{1, N_s\}. \quad (3.15)$$

3.4 Proper Orthogonal Decomposition in Angle

This section describes the method of POD in angle, first presented by Buchan et al [11]. The ROM is based on the use of POD and the method of snapshots to form optimal angular basis functions for a particular class of problem. Each snapshot is given by the angular flux vector of its associated node, and a snapshot matrix S is formed whose

columns are given by the snapshot vectors:

$$\mathrm{S} = \begin{bmatrix} | & | & & | \\ \psi^1 & \psi^2 & \ldots & \psi^{n_h} \\ | & | & & | \end{bmatrix}, \tag{3.16}$$

where n_h is the total number of snapshots. Each snapshot is of length N_a, which is the number of angles defined by the discrete ordinate angular discretisation. The snapshot matrix is therefore of size $N_a \times n_h$. The model may be trained to account for variation in one or more parameters by including snapshots from multiple solutions, each of which is produced on the same domain by varying the parameter(s) in question. This allows the model to interpolate and, to a certain extent, extrapolate to new values of the parameter(s). The number of snapshots n_h is therefore given by the number of nodes in the domain, N_n, multiplied by the number of problems which the model is trained on, N_p.

Once the snapshot matrix has been generated, the singular value decomposition (SVD) is performed,

$$\mathrm{S} = \mathrm{U}\Sigma V^T, \tag{3.17}$$

where U and V are orthogonal matrices of size $N_a \times N_a$ and $n_h \times n_h$, respectively, and Σ is of size $N_a \times n_h$. The matrix U contains the POD basis functions as its columns, ordered such that the first n_a columns form the optimal n_a basis vectors in the Frobenius norm. Σ is a diagonal matrix containing the singular values, which give the fraction of the information in the snapshot matrix which can be represented by the first n_a basis functions as,

$$I(n_a) = \frac{\sum_{i=1}^{n_a} (\Sigma_{i,i})^2}{\sum_{i=1}^{N_a} (\Sigma_{i,i})^2}, \tag{3.18}$$

where I varies from 0 to 1, with 1 being total capture of the snapshot information. V^T contains information which is extraneous to this application of the SVD, and is discarded.

The reduced order basis matrix $\mathcal{U}$ is formed by truncating the matrix U such that only

the first n_a columns are retained:

$$\mathcal{U}^{jk} = \mathrm{U}^{jk}, \forall j \in \{1, N_a\}, \forall k \in \{1, n_a\}, \tag{3.19}$$

where $\mathcal{U}^{jk}$ and U^{jk} denote the j^{th} row and k^{th} column of each matrix. The basis matrix $\mathcal{U}$ can be used to map the angular coefficients between the full and reduced order models through the relationship,

$$\Psi(\vec{r}) \approx \mathcal{U}\alpha(\vec{r}), \tag{3.20}$$

where $\alpha(\vec{r})$ is the vector of reduced order angular coefficients at each point in space. Substituting equation 3.20 into equation 3.8 and premultiplying by $\mathcal{U}^T$ projects the angularly discretised equations onto the POD space:

$$\Big(\mathcal{U}^T A_x \mathcal{U}\frac{\partial}{\partial x} + \mathcal{U}^T A_y \mathcal{U}\frac{\partial}{\partial y} + \mathcal{U}^T A_z \mathcal{U}\frac{\partial}{\partial z} + \mathcal{U}^T H \mathcal{U}\Big)\alpha(\vec{r}) = \mathcal{U}^T Q \tag{3.21}$$

Spatial discretisation can then be performed on equation 3.21 as described in section 3.3, with $\alpha(\vec{r})$ taking the place of $\Psi(\vec{r})$ in the derivation.

3.5 The Discontinuous Angular Reduced Order Model

This section describes a novel modification to the angular POD method which was first proposed in [11], and described in section 3.4. The previous method used full angular flux vectors to form the snapshot matrix, resulting in the formation of POD basis functions that spanned the entire surface of the sphere. The fundamental difference in this work is to relax the global nature of the POD basis functions, and to instead form basis sets with compact support - that is, they span subsets of the sphere. The selection of these angular regions can be arbitrary, but in this project each octant of the sphere defines a separate region, which is shown to stabilise the basis functions and as such should be considered the starting point for this method. Further subdivision may be beneficial in some contexts, such as highly anisotropic flux distributions, but has not been investigated here. The angular domain Ω is partitioned into the regions,

$$\Omega = \bigcup_{q=1}^{8} \Omega_q, \tag{3.22}$$

where Ω_q defines the q^{th} octant, which will have its own set of optimised basis functions. Figure 3.1 depicts one such octant.

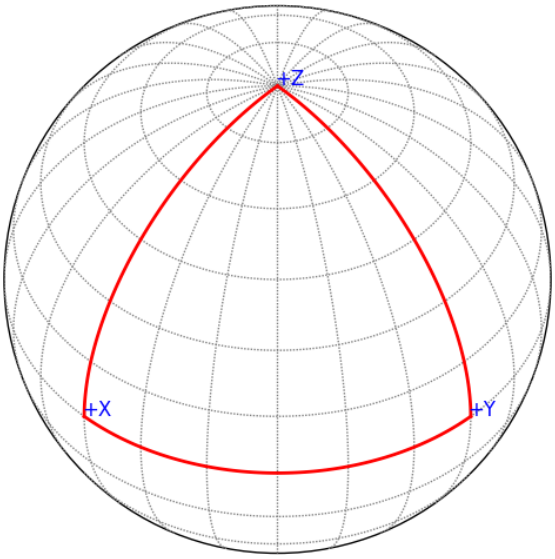


Figure 3.1: An octant on the sphere, which defines the boundaries of an angular region.

As previously, each class of problem may be resolved for N_p cases with varying conditions, such as perturbations to the nuclear material cross-sections. The vectors of angular coefficients formed at the nodes of the spatial mesh are then partitioned into sets according to their angular region,

$$\psi = \begin{bmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_8 \end{bmatrix}, \tag{3.23}$$

where ψ_q is a vector containing the N_q coefficients with associated directions within Ω_q. Note that $N_a = \sum^{q} N_q$. This partitioning of the angular vector is used to form separate snapshot matrices for each Ω_q. When using the octants of the sphere as angular regions, eight separate snapshot matrices are formed from the vectors of angular coefficients at the mesh nodes, for all problems used in creation of the ROM. In two dimensions, half of these octants are reflections of the other half and can be ignored. The snapshot matrices

produced are given by

$$\mathrm{S}_q = \begin{bmatrix} | & | & & | \\ \psi_q^1 & \psi_q^2 & ... & \psi_q^{n_h} \\ | & | & & | \end{bmatrix}, \ \forall q \in \{1, 8\}. \tag{3.24}$$

The POD basis sets for each angular region Ω_q can now be formed through the SVD of each snapshot matrix,

$$\mathrm{S}_q = \mathrm{U}_q \Sigma_q V_q^T \tag{3.25}$$

where U_q and V_q^T are unitary matrices of sizes $N_q \times N_q$ and $n_h \times n_h$, respectively. As previously, the column vectors of U_q contain the basis vectors which optimally represent the snapshot data, ordered such that the first n_a columns form the optimal n_a basis vectors in the Frobenius norm. The angular POD basis matrices $\mathcal{U}_q$ are formed by truncating each snapshot matrix such that only the first n_a columns are retained:

$$\mathcal{U}_q^{jk} = \mathrm{U}_q^{jk} \qquad \forall q \in \{1, 8\}, \forall j \in \{1, N_q\}, \forall k \in \{1, n_a\}, \tag{3.26}$$

where $\mathcal{U}_q^{jk}$ and U_q^{jk} denote the j^{th} row and k^{th} column of the matrix associated with the angular region q. Note that there is freedom to set the expansion size of each angular region independently - that is, n_a may vary with q. The fraction of the information in U_q which is retained in $\mathcal{U}_q$ can again be determined from the singular values:

$$I_q = \frac{\sum_{j=1}^{n_a} (\Sigma_q^{jj})^2}{\sum_{j=1}^{N_q} (\Sigma_q^{jj})^2}, \tag{3.27}$$

where I_q varies from 0 to 1, with 1 being total capture of the snapshot information. The matrices $\mathcal{U}_q$ can be used to map the angular coefficients between the full and reduced order models through the relationship,

$$\Psi_q(\vec{r}) \approx \mathcal{U}_q \alpha_q(\vec{r}), \tag{3.28}$$

for each angular partition Ω_q. The combined mapping over all angular partitions can be

expressed as,

$$\mathcal{U} = \begin{bmatrix} \mathcal{U}_1 & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 \\ 0 & 0 & \mathcal{U}_q & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & \mathcal{U}_8 \end{bmatrix}, \ \alpha(\vec{r}) = \begin{bmatrix} \alpha_1(\vec{r}) \\ \vdots \\ \alpha_q(\vec{r}) \\ \vdots \\ \alpha_8(\vec{r}) \end{bmatrix} \implies \mathcal{U}\alpha(\vec{r}) = \begin{bmatrix} \mathcal{U}_1\alpha_1(\vec{r}) \\ \vdots \\ \mathcal{U}_q\alpha_q(\vec{r}) \\ \vdots \\ \mathcal{U}_8\alpha_8(\vec{r}) \end{bmatrix}. \quad (3.29)$$

Equations 3.23 and 3.28 enable this to be compactly written as,

$$\Psi(\vec{r}) \approx \mathcal{U}\alpha(\vec{r}) \quad (3.30)$$

Substituting equation 3.30 into equation 3.8 and premultiplying by $\mathcal{U}^T$ projects the angular discretised equations onto the POD space:

$$\left(\mathcal{U}^T\mathbf{A}_x\mathcal{U}\frac{\partial}{\partial x} + \mathcal{U}^T\mathbf{A}_y\mathcal{U}\frac{\partial}{\partial y} + \mathcal{U}^T\mathbf{A}_z\mathcal{U}\frac{\partial}{\partial z} + \mathcal{U}^T\mathbf{H}\mathcal{U}\right)\alpha(\vec{r}) = \mathcal{U}^T\mathbf{Q} \quad (3.31)$$

Spatial discretisation can then be performed on equation 3.31 as described in section 3.3, with $\alpha(\vec{r})$ taking the place of $\Psi(\vec{r})$ in the derivation. Note that in the case of a single angular region spanning the full sphere, the angular POD method implemented by Buchan et al. [11] and described in section 3.4 is recovered.

3.6 Adaptivity in Angle

This section presents an adaptive algorithm using the DPOD basis functions. The standard DPOD method uses the same number of basis functions n_a throughout the domain, whereas in the adaptive DPOD (ADPOD) method each octant q and element e has an associated number of basis functions n_{qe}, which can be modified independently. The adaptive algorithm seeks to determine where to add basis functions in order to maximise their effectiveness in reducing total error. An initial solution is required in order to calculate the error metric and begin to add basis functions, and so the problem in question is first solved with

$$n_{qe} = 2, \ \forall q, \ \forall e. \quad (3.32)$$

The DPOD basis functions form a hierarchical set, with each function describing a larger fraction of the data than those which follow it, and so successive coefficients tend towards zero. It is therefore possible to infer which locations are likely to benefit most from additional basis functions by comparing their final coefficients. If the final coefficient in a particular location is large, it is likely that an additional basis function would also have a large coefficient, and would thus significantly impact the solution. By contrast, if the final coefficient is small, then successive coefficients are also likely to be small and have little impact on the overall solution. However, examination of solution vectors at various points showed that the magnitude of successive coefficients tends to oscillate, and so a single small coefficient does not guarantee convergence in that location. As a result, the final coefficient alone was not a reliable metric. To account for this, the larger of the final two coefficients at each location is compared. A list of these coefficients L_{qe} is compiled:

$$\mathrm{L}_{qe} = \mathrm{Max}(\alpha_{qe}^{n_{qe}-1}, \alpha_{qe}^{n_{qe}}),\ \forall q \in \{1,8\},\ \forall e \in \{1,N_e\},\ \forall j \in \{1,N_n\} \tag{3.33}$$

Where α_{qj}^{k} denotes the k^{th} angular coefficient for octant q at node j, and N_e is the number of elements in the spatial mesh. The list L_{qe} is sorted by magnitude, and the largest n_+ entries have their associated n_{qe} values increased, while the smallest n_- have their n_{qe} values decreased (to a minimum of two). Once n_{qe} has been adjusted, the process repeats until the problem has converged to within the desired tolerance. The complete adaptive method is described by algorithm 1.

Algorithm 1: Adaptive DPOD

```
/* Find an initial solution with the minimum number of basis
   functions.                                                  */
```

$n_{qe} = 2$ for all q and e.

Solve the ROM.

Calculate the error ϵ.

```
/* Iterate until the desired convergence is reached.          */
```

while $\epsilon >$ tolerance **do**

Create the list L_{qe} and sort by magnitude.

Increment n_{qe} by 1 for each of the first n_+ entries in L_{qe}.

Decrement n_{qe} by 1 to a minimum of 2, for each of the last n_- entries in L_{qe}.

Solve the ROM.

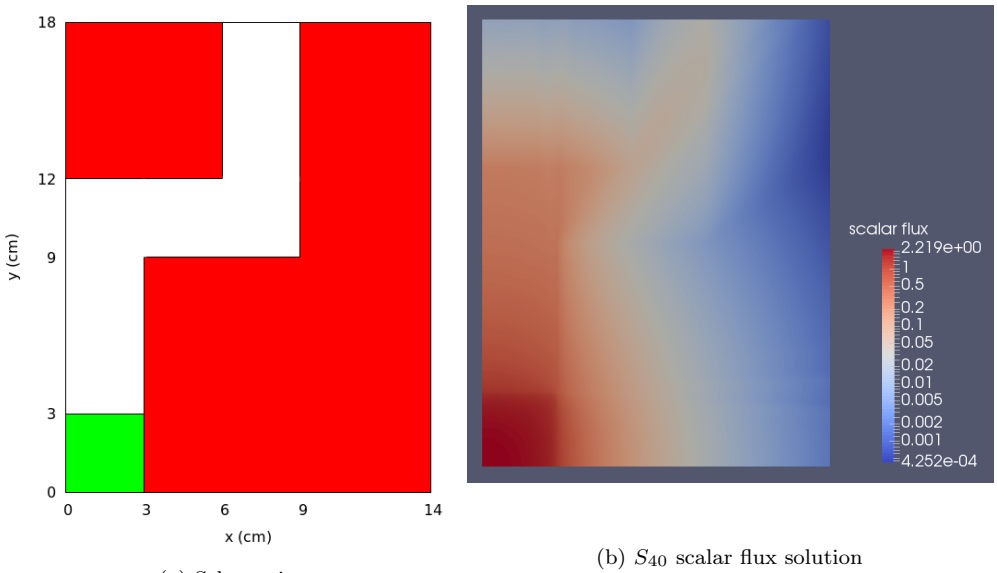
Calculate the error ϵ.

end

3.7 Numerical Examples

In this section, two numerical examples are presented in order to compare POD, DPOD and ADPOD. Uniform quadrilateral FEM spatial meshes are employed, using discontinuous bilinear basis functions. The full order method employs the S_N discretisation, with a sufficiently high angular resolution to ensure that the solutions have converged in angle. A sweep based solver is used to resolve the S_N and all POD methods. This solver was poorly optimised when results were gathered, and so computational efficiency is determined based on the number of solver iterations required for solution convergence, and the error for a given number of angular degrees of freedom. It should be noted that for the POD and DPOD methods the complexity, and hence computation time, of each solver iteration is the same for a given number of basis functions. Furthermore, the sparsity within the DPOD angular streaming matrices (75% for 2D and 87% for 3D) has not been considered, and so DPOD could potentially be implemented more efficiently than POD. The purpose of this analysis is to demonstrate the novel methods' improved solver convergence and increased accuracy for a given basis size when compared with S_N and previous angular POD methods.

3.7.1 The Dog-Leg Duct Problem



(a) Schematic

(b) S_{40} scalar flux solution

Figure 3.2: Schematic (a) and S_{40} scalar flux solution (b) for the dog-leg duct extrapolation problem. Green is the source, white is the duct and red is a highly absorbing material.

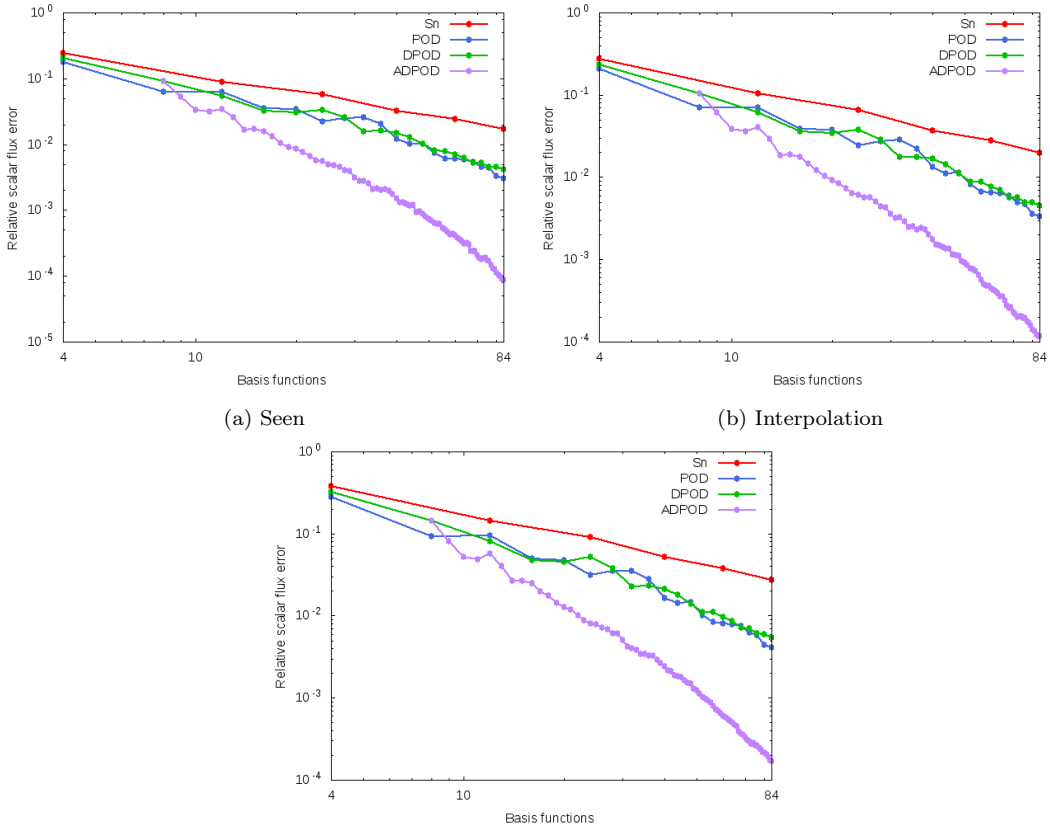
The first example is an advective dog-leg duct problem [12] described in section 1.3.1. Figure 3.2a shows a schematic of the domain. The green region is an isotropic source, the white region is the duct, and the red regions are heavy absorbers. Vacuum boundary conditions are applied to the top and right boundaries, and reflective boundary conditions to the bottom and left boundaries. The spatial dimension is discretised with a 140×180 mesh of discontinuous bilinear quadrilateral elements. The full order solutions used for snapshots and error calculations employed the S_{40} angular discretisation. Figure 3.2b shows the scalar flux distribution of the S_{40} solution to the extrapolation problem.

Problem	Region	Source ($cm^{-2}s^{-1}$)	Σ_a (cm^{-1})	Σ_s (cm^{-1})
1	Green	1.00	0.40	0.00
	White	0.00	0.00	0.00
	Red	0.00	0.40	0.00
2	Green	1.00	0.50	0.00
	White	0.00	0.00	0.00
	Red	0.00	0.50	0.00
3	Green	1.00	0.60	0.00
	White	0.00	0.00	0.00
	Red	0.00	0.60	0.00
Seen	Green	1.00	0.50	0.00
	White	0.00	0.00	0.00
	Red	0.00	0.50	0.00
Interpolate	Green	1.00	0.45	0.00
	White	0.00	0.00	0.00
	Red	0.00	0.45	0.00
Extrapolate	Green	1.00	0.35	0.00
	White	0.00	0.00	0.00
	Red	0.00	0.35	0.00

Table 3.1: Material properties for the dog-leg duct problems in chapter 3.

Table 3.1 lists the material cross sections for both the training and test problems. As the problem is purely advective, altering its scattering coefficient would qualitatively change the solutions. It was therefore left at 0 in all cases, and only the absorption coefficient was altered. The snapshot matrix was formed from all three training solutions, and the resulting POD bases were used to solve the test problems. The first test problem is referred to as the *seen* problem, as its material properties are identical to one of the training solutions. The other two are both *unseen* problems - the *interpolation* problem has its Σ_a value within the range for which snapshots were produced, while Σ_a for the *extrapolation* problem lies outside this range. Reduced order solutions to all three test problems were used in producing results, in order to compare the models' ability to cope

with variations in material properties.



(a) Seen

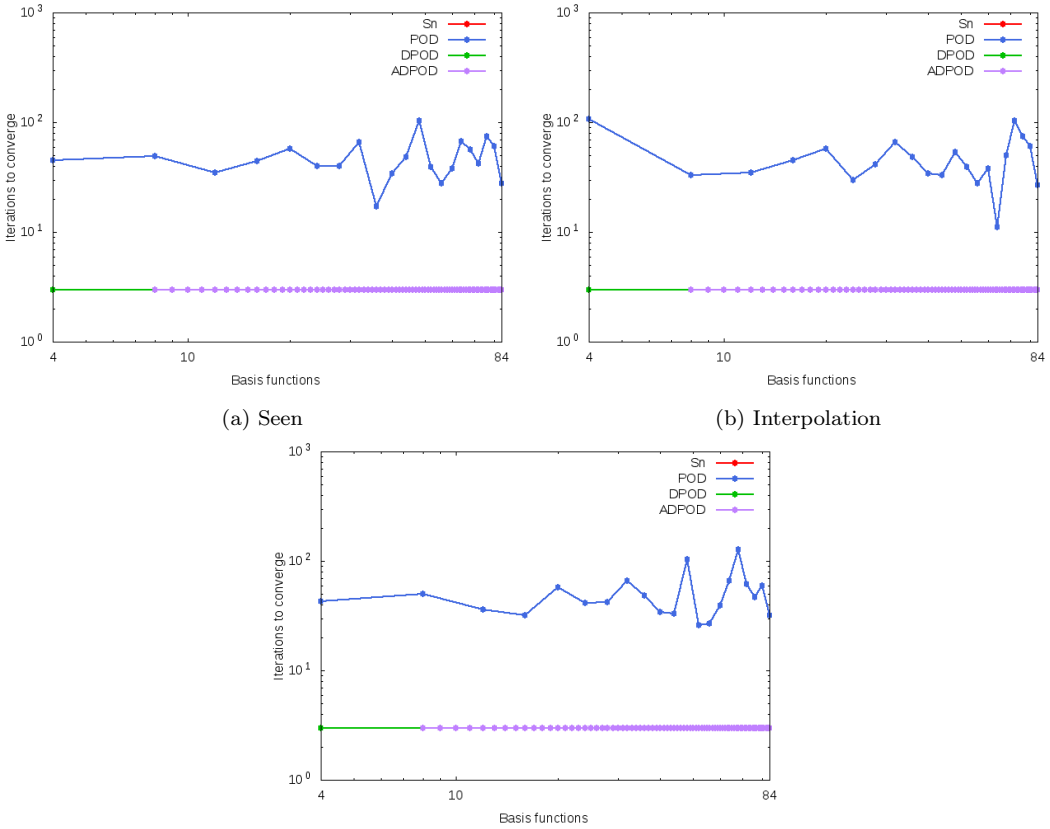
(b) Interpolation

(c) Extrapolation

Figure 3.3: The effects of increasing basis function count on relative scalar flux error for the dog-leg duct problem.

Figure 3.3 presents the L^1-norm of the relative scalar flux error in solutions of the dog-leg duct problem against the mean number of basis functions per node. Low order S_N solutions with degrees of freedom equivalent to the ROMs' are also shown, with their number of basis functions equal to the number of S_N angular directions, given by $N_a = N(N+2)/2$ in two dimensions, where N is the order of the S_N discretisation. This provides a comparison between the ROMs and a full order model of equivalent computational complexity. For example, the S_{12} angular discretisation has 84 basis functions, and the matrix systems it produces are equal in size and similar in computational cost to POD and DPOD with a total of 84 basis functions per node. In the case of adaptive DPOD, the number of basis functions varies by octant and element, so the mean number of basis functions is used. This is not exactly equivalent, since the computational cost scales with the square of the number of basis functions, but it provides an approximate comparison. POD consistently has lower error than S_N with

the same number of basis functions. For this problem, DPOD outperforms POD in some cases but underperforms in others. Overall, the two methods perform approximately equally in terms of error per basis function. ADPOD is a huge improvement upon all three of the other methods, particularly for larger numbers of basis functions, where it has an error almost two orders of magnitude smaller than POD and DPOD. This demonstrates the ability of angular adaptivity to drastically reduce the computational cost required to reach a given level of error.



(a) Seen

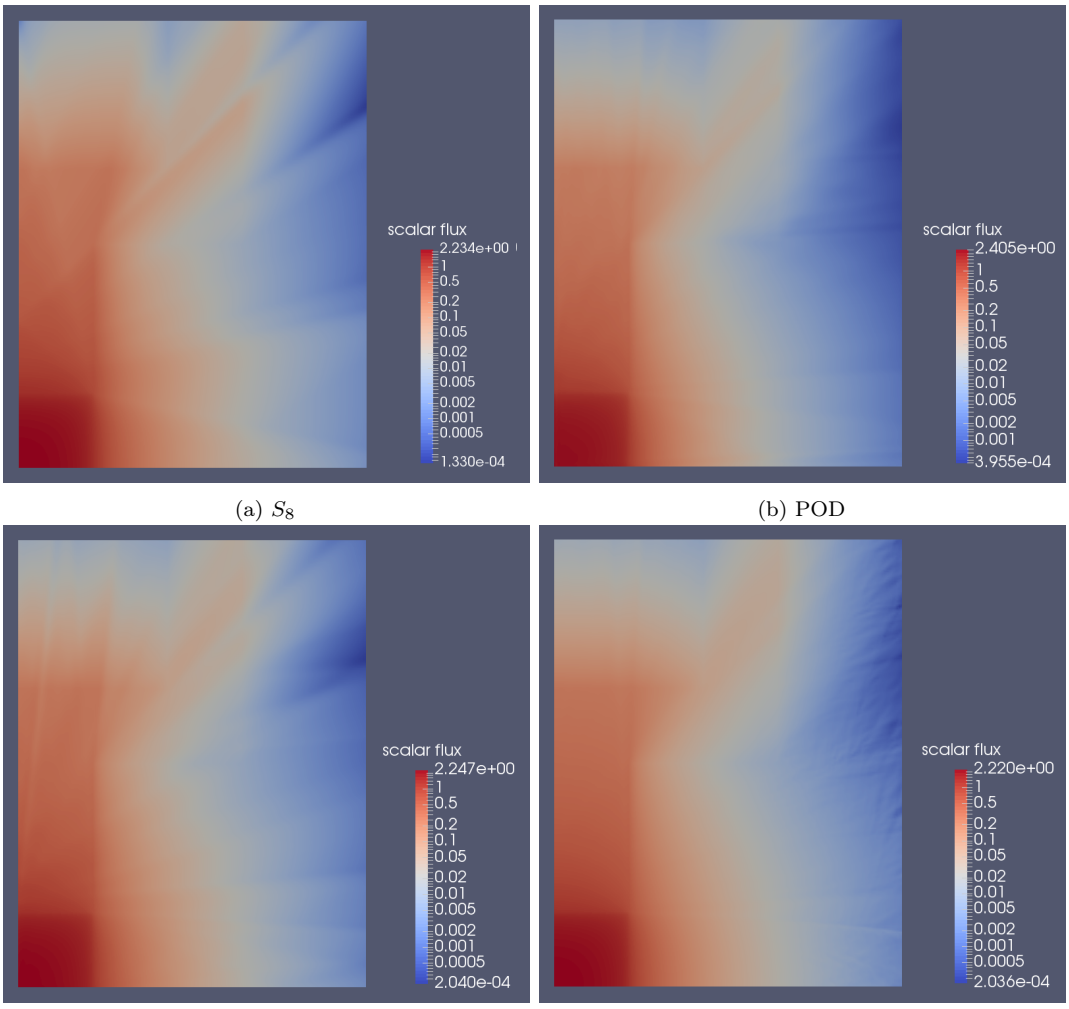
(b) Interpolation

(c) Extrapolation

Figure 3.4: The effects of increasing basis function count on the number of iterations required to converge to a solution for the dog-leg duct problem.

Figure 3.4 shows the number of solver iterations required to converge the solution for each method. A ROM solution was considered converged when the L^1-norm of the difference between current and previous iterations' scalar flux solutions decreased below 10^{-6}. In the case of ADPOD, the number of iterations is presented for each adaptive step, in order to demonstrate the stability of the ADPOD bases throughout the adaptive process. Due to the non-scattering media, the S_N discretisation required just two iterations to converge to a solution, though a third was performed to verify its convergence. By

contrast, POD required up to 1.8 orders of magnitude more iterations to converge. The number of iterations required was also inconsistent, with a range of approximately an order of magnitude, which caused unpredictable variation in the solve time. This demonstrates the aforementioned problem with previous angular POD methods, which this research was intended to solve. As the graph shows, DPOD and ADPOD do not suffer from this disadvantage. Instead, they converge in two sweeps as with S_N. This provides the first evidence that DPOD and ADPOD may overcome the solver instability issue of angular POD, at least in the case of non-scattering problems. As the solve time is dependant on the number of iterations required to converge the solution, DPOD provides significant benefits over POD despite their similar levels of error. ADPOD also resolved the solver instability issue, and it also had significantly reduced error compared to the non-adaptive methods, demonstrating that it provides significant benefits in this case.



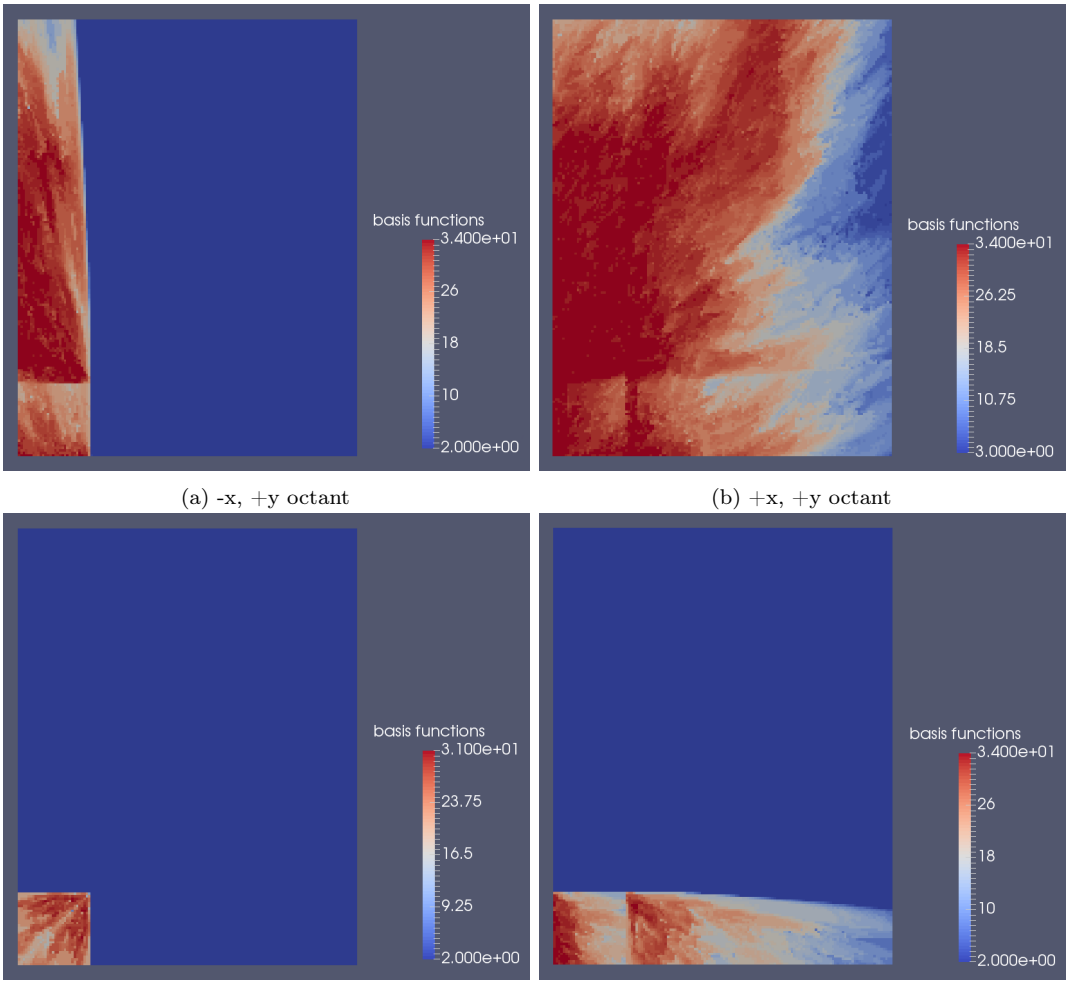
(a) S_8 (b) POD

(c) Discontinuous POD (d) Adaptive Discontinuous POD

Figure 3.5: Solutions to the dog-leg duct extrapolation problem for all four methods with 40 basis functions per node, on average in the case of ADPOD.

Figure 3.5 depicts the scalar flux solutions for the extrapolation problem produced by each method, using 40 basis functions per node (on average in the case of ADPOD). The full order solution to this problem is shown in figure 3.2b. The S_8 solution exhibits significant ray effects, which results in a highly inaccurate flux distribution within and outside of the duct. POD reduces these ray effects somewhat, though they are still visible in some areas. However, it has the most inaccurate peak flux by far, overestimating the flux in the source region by almost 10%. DPOD suffers from significant ray effects in this case, but its peak flux is much closer to the expected value than that of POD. It seems that the increased ray effects found in the DPOD solution somewhat counteracted its more accurate approximation of the peak flux, but in this case the DPOD error was still slightly lower overall, as figure 3.3c shows. ADPOD drastically reduces ray effects

compared to every other method, and has the smallest error in its peak flux. Minor distortions are present, but only in the regions with the least flux, where their effect on the total error is insignificant. This is to be expected, since these regions should be of low priority for additional basis functions. These results demonstrate the benefits of DPOD and particularly ADPOD in resolving non-scattering problems with few basis functions, when compared to previous methods. It also provides evidence that the adaptive method worked as intended.



(a) -x, +y octant

(b) +x, +y octant

(c) -x, -y octant

(d) +x, -y octant

Figure 3.6: Number of ADPOD basis functions per octant per node for the duct extrapolation problem, with a mean of 40 basis functions per node in total.

Figure 3.6 depicts the number of basis functions per node of each element and octant in the domain for the adaptive method. A large section of the domain had zero flux in three out of four octants, since it was only accessible by flux travelling in the +x, +y direction. The adaptive algorithm successfully accounted for this by keeping the number of basis functions in these regions at the minimum of two. Most basis functions were

focused inside the duct, which is to be expected since it is also the location of most of the flux, and so basis functions in this region are likely to have larger coefficients and contribute most to the overall solution. However, the absorbing regions had a relatively low cross section, and so flux penetrated into these regions rather than being immediately attenuated. The adaptive algorithm therefore added basis functions to these regions as well. These results provide further evidence of ADPOD's ability to focus basis functions where they are most beneficial, which explains its success in drastically reducing error for a given number of basis functions.

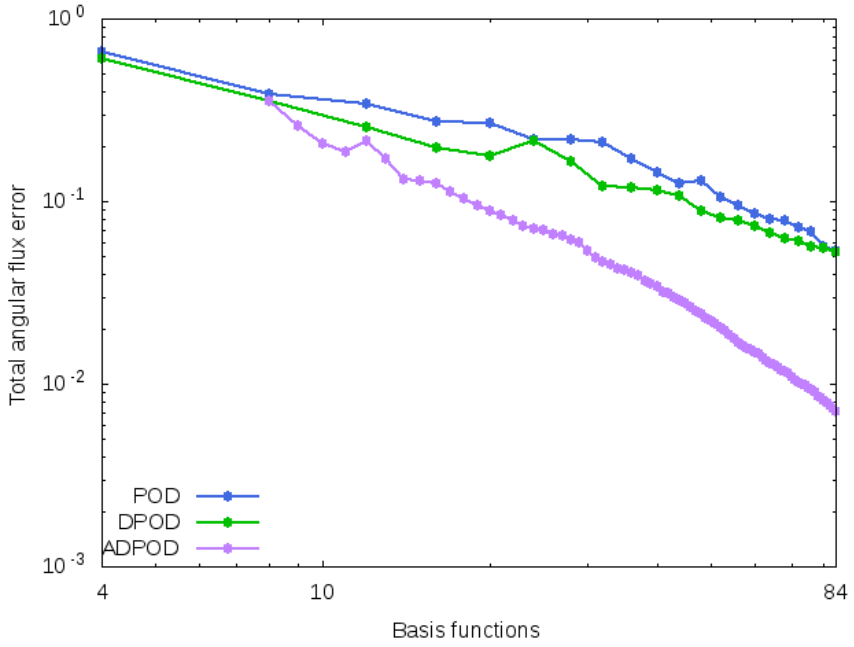
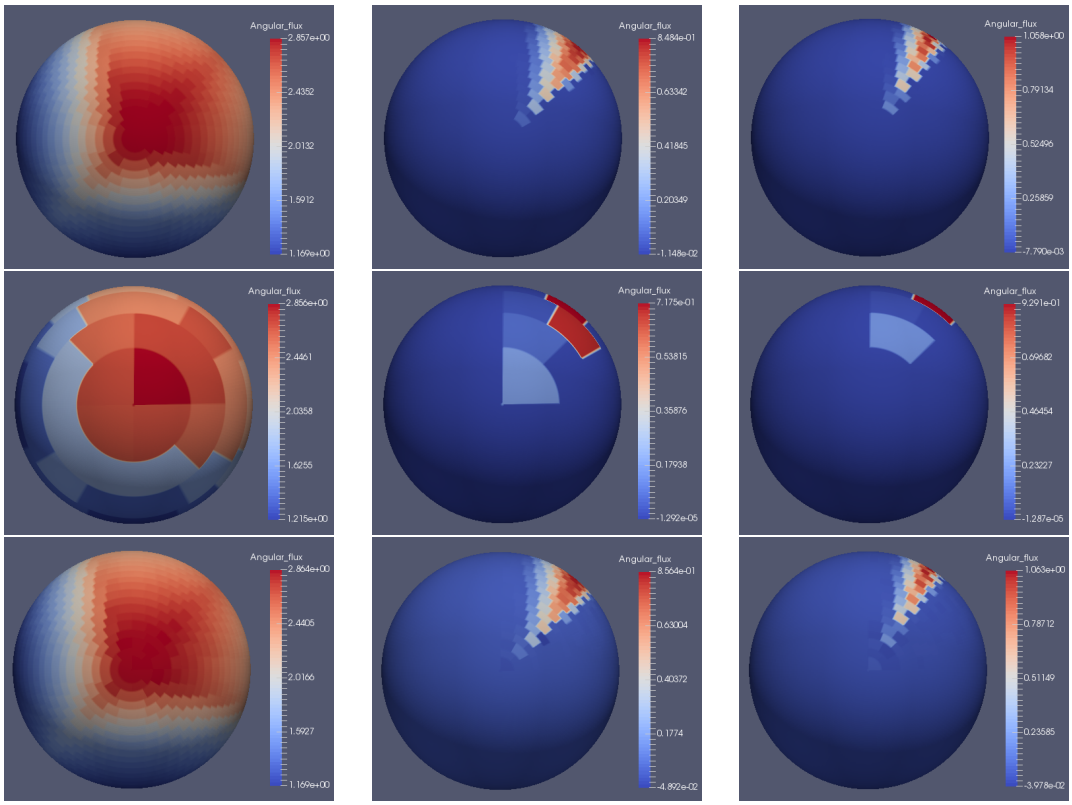


Figure 3.7: The effect of increasing basis function count on relative angular flux error for the dog-leg duct extrapolation problem.

Figure 3.7 presents the relative angular flux error against the number of basis functions per node, which is a more accurate error metric than the scalar flux error. The results are similar to the scalar flux error plots in figure 3.3, but in this plot DPOD is a slight improvement upon POD in every case, and the error oscillates less about its trend line. This suggests that cancellation of errors was responsible for the oscillating error, and for POD outperforming DPOD in some cases. Since cancellation of errors does not occur with angular flux, these problems are not present and the resulting graph is therefore more reliable. However, it is not feasible to calculate the scalar flux error between two different full order solutions in this case, as the S_N model is not hierarchical, and so both scalar and angular flux plots have been presented. Once again, ADPOD performs

significantly better than either method, particularly for large basis function counts. Since the angular flux error is a more reliable metric, this provides further evidence that the adaptive method worked as expected.



(a) x = 1.5cm, y = 1.5cm (b) x = 6.0cm, y = 10.5cm (c) x = 7.5cm, y = 16.0cm

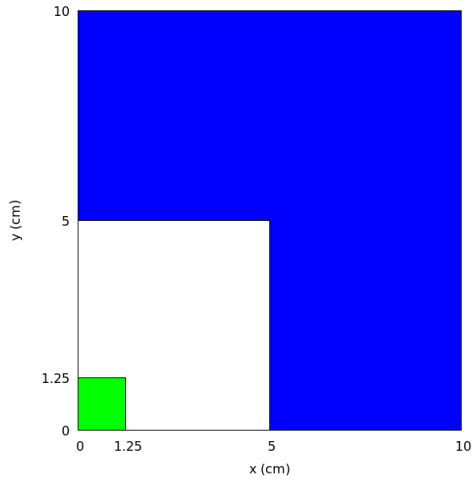
Figure 3.8: Angular flux distributions at various points in the dog leg duct problem. From top to bottom, S_{40}, S_8, and adaptive discontinuous POD. Both S_8 and ADPOD have a mean of 40 basis functions per node, while S_{40} has 840 basis functions per node.

Figure 3.8 depicts the angular flux profiles at three points in the domain for the extrapolation problem. The S_{40} solution has 840 basis functions in two dimensions, while the S_8 and ADPOD solutions both have 40. The S_8 angular discretisation is highly inaccurate. The solid angles subtended by each basis function are large, and so complex angular flux profiles can not be accurately represented. By contrast, ADPOD was able to reproduce the S_{40} angular flux profiles with only minor differences. The ADPOD solution has a higher angular resolution as its snapshots were produced with the S_{40} angular discretisation, and any angular flux profile it produced had the same resolution. However, appropriate basis functions are required for ADPOD to accurately reproduce the full order solution, as it has significantly fewer degrees of freedom. In this case, the model produced useful basis functions and the adaptive algorithm included

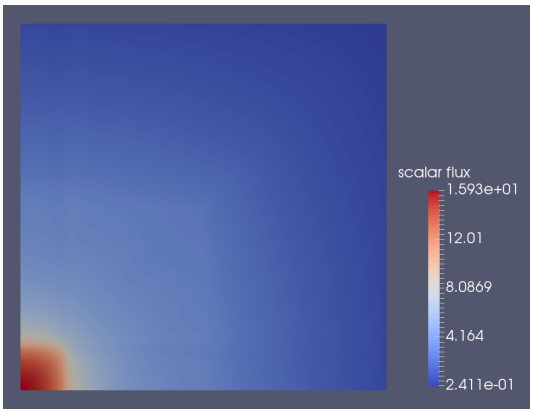
enough of them at each point to reproduce the S_{40} solution with only minor errors. This demonstrates the benefits of ROMs (and ADPOD in particular) over low order models when producing approximate solutions to higher order models.

3.7.2 The Watanabe-Maynard Problem

The second example is the Watanabe-Maynard problem, which was introduced in [13] and is described in section 1.3.2. A schematic is shown in Figure 3.9a. The green region is an isotropic square source, the white region is a void, and the blue region is a highly scattering material. Vacuum boundary conditions are applied to the top and right boundaries, and reflective boundary conditions to the bottom and left boundaries. The domain is discretised using an 80×80 mesh of discontinuous bilinear quadrilateral elements in space. Full order solutions for snapshot generation and error comparison were produced using S_{30} in angle. Figure 3.9b shows the scalar flux distribution of the S_{30} solution to the extrapolation problem.



(a) Schematic



(b) S_{30} scalar flux solution

Figure 3.9: Schematic (a) and S_{30} scalar flux solution (b) for the Watanabe-Maynard extrapolation problem. Green is the source, white is a void and blue is a highly scattering material.

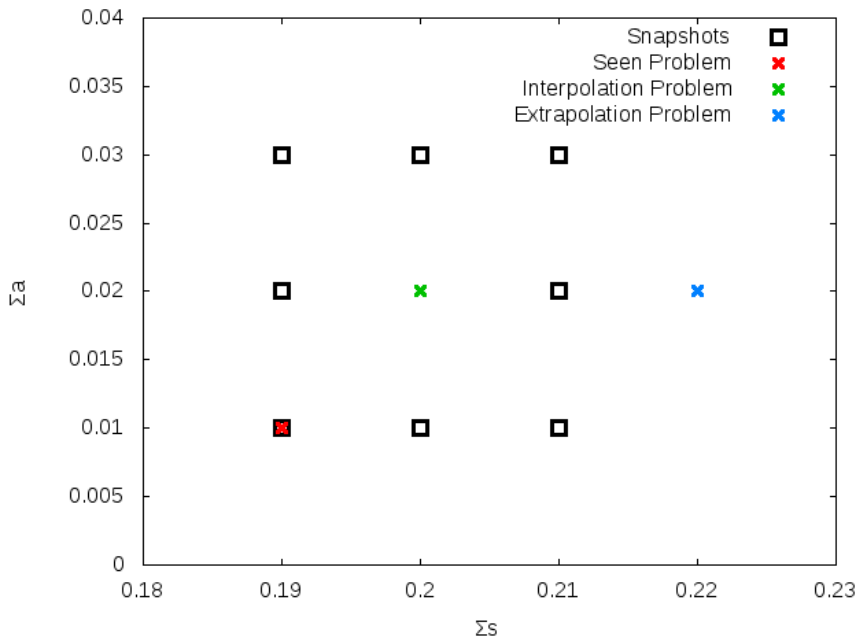
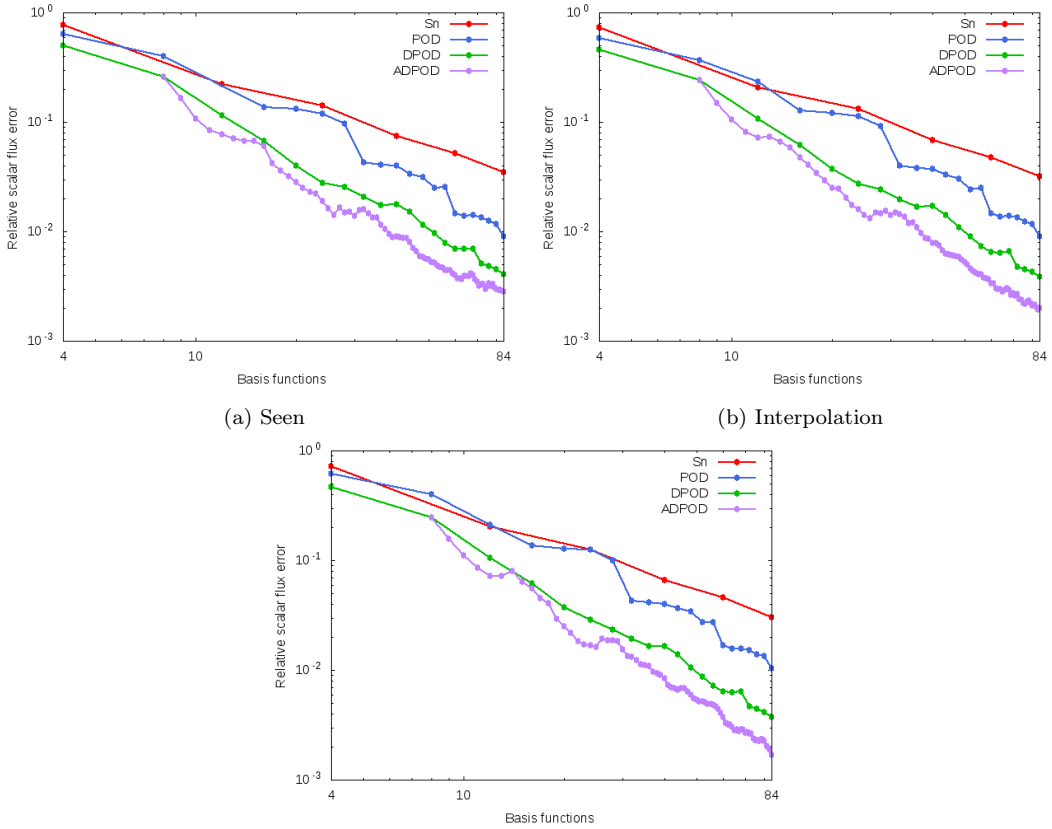


Figure 3.10: Σ_a and Σ_s values used to produce snapshots and solutions for the Watanabe-Maynard problem.

Figure 3.10 shows Σ_a and Σ_s values in the non-void regions for eight solutions and three test problems. While previously the scattering cross-section was kept at 0 since the problem was intended to be purely advective, this problem includes scattering, and so both the scattering and absorption cross-sections were modified. The snapshot matrix was formed by combining all eight training solutions, and the POD bases produced were used to solve all three test problems. As previously, one set of material properties had been seen in the offline stage, one was an interpolation between seen properties, and one was an extrapolation outside the range of seen material properties.



(a) Seen

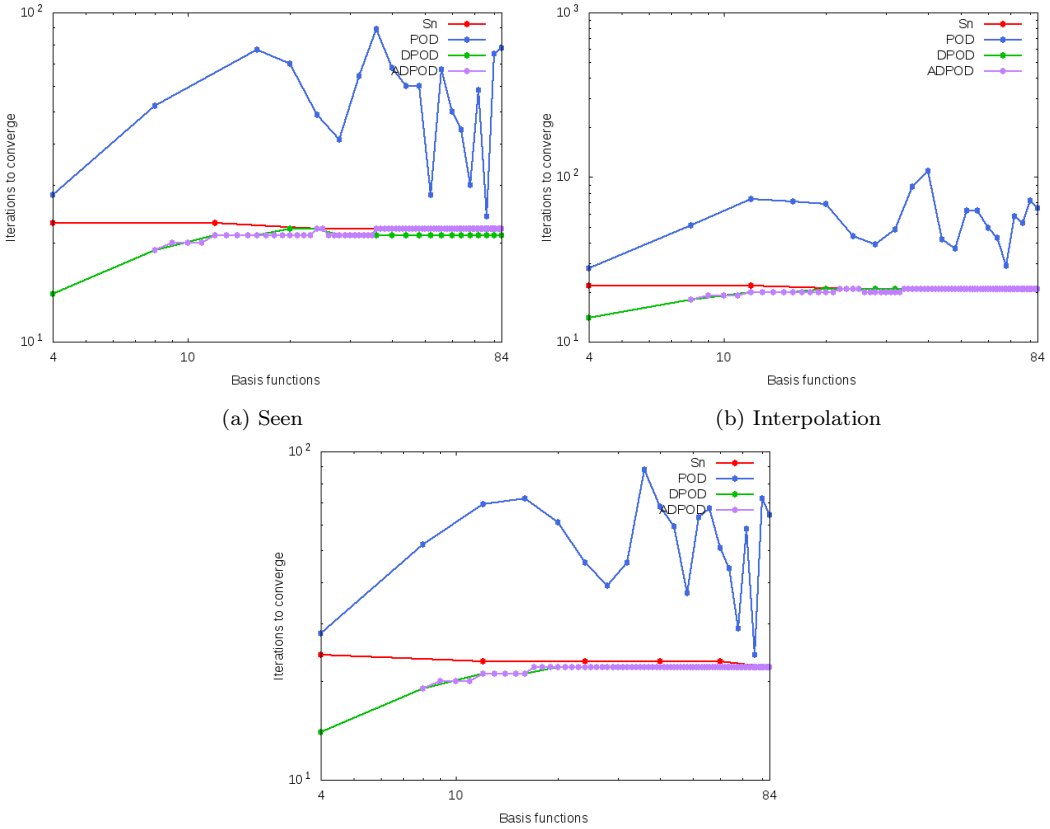
(b) Interpolation

(c) Extrapolation

Figure 3.11: The effects of increasing basis function count on relative scalar flux error for the Watanabe-Maynard problem.

Figure 3.11 presents the L^1-norm of the relative scalar flux error in solutions of the Watanabe-Maynard problem against the mean number of basis functions per node for each method. Results from low order S_N solutions are shown for comparison, as previously. In this case, POD generally had a lower error than S_N with the same number of basis functions, though it struggled with 12 and fewer. In particular, POD could not solve the seen problem with 12 basis functions, as the system diverged. These issues are likely due to the problem's complexity when compared to the dog-leg duct. The Watanabe-Maynard problem contains both void and scattering regions, and with few basis functions the flux in some areas could not be resolved properly. DPOD did not suffer from this issue, likely because the decoupling of angular fluxes in each octant ensured that information could always flow in approximately the right direction. DPOD also had consistently less error for a given number of basis functions than POD, usually by a factor of two to three. ADPOD improved upon this, reducing error by a further factor of two to three compared to DPOD. However, it did not provide the huge benefits

seen previously. This is likely because the scattering present in this problem ensures that flux travels throughout the domain, and so basis functions are required everywhere. Therefore, the adaptive method could not simply ignore large portions of the domain. By contrast, in the dog-leg duct problem flux is primarily limited to the duct itself, and so basis functions can be focused in that region.



(a) Seen

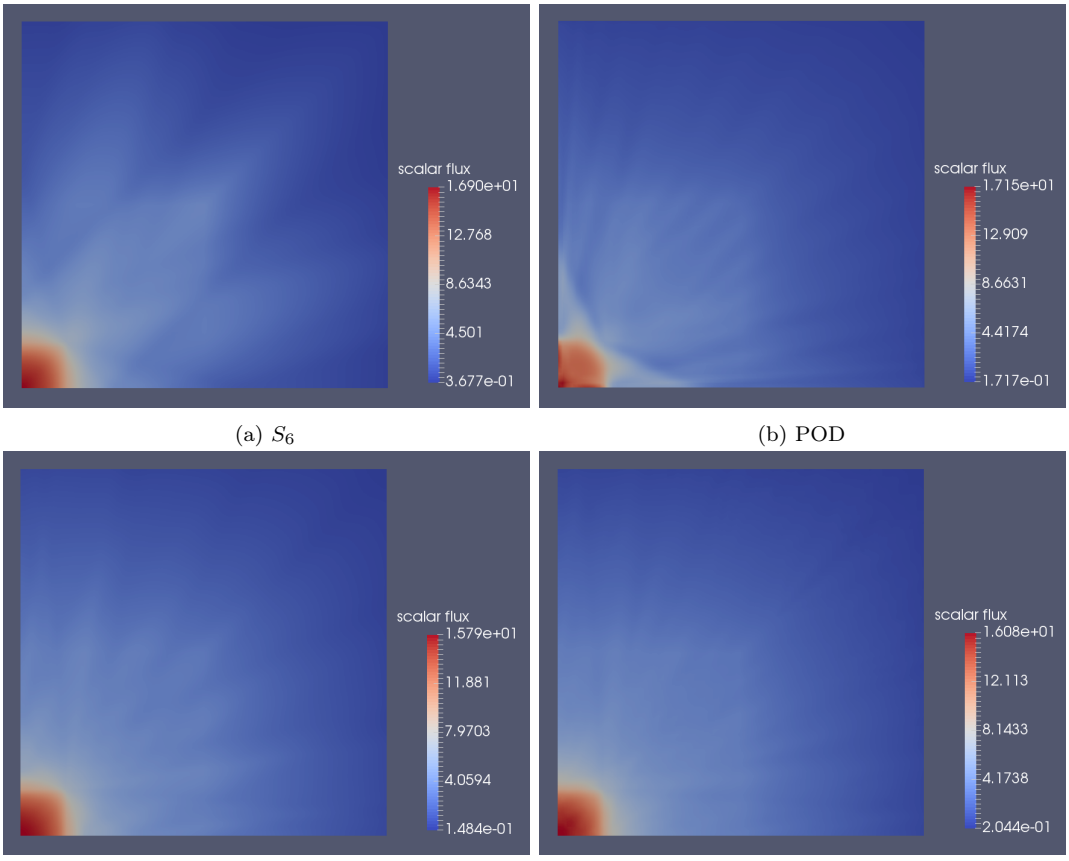
(b) Interpolation

(c) Extrapolation

Figure 3.12: The effects of increasing basis function count on the number of iterations required to converge a solution for the Watanabe-Maynard problem.

Figure 3.12 shows the number of solver iterations required to converge the solution for each method. As with the dog-leg duct problem, a solution was considered converged when the L^1-norm of the difference between current and previous iterations' scalar flux solutions decreased below 10^{-6}. However, the Watanabe-Maynard problem contains scattering regions which must be resolved by iteration, and so every method required more iterations to converge than the dog-leg duct problem. As previously, POD performed much worse than any other method by this metric. It consistently required more iterations to solve than the other three methods, and the exact number of iterations varied significantly. The other methods all converged faster and more consistently, with

similar numbers of iterations required for each, though DPOD and ADPOD converged slightly faster than S_N in most cases. This provides further evidence that the solver instability issue exhibited by POD has been resolved by the DPOD method, including for problems with scattering materials. Since the solve time is dependant on the number of iterations required to converge the solution, this constitutes a clear benefit of DPOD and ADPOD over POD.



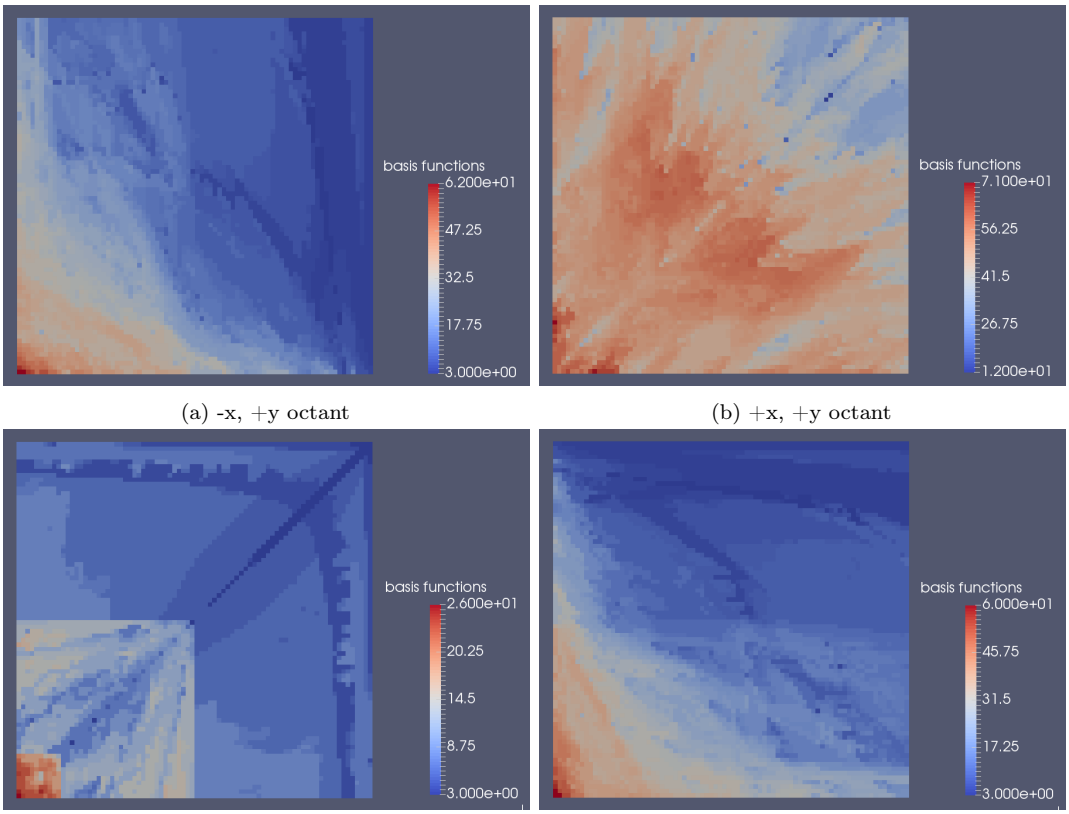
(a) S_6 (b) POD

(c) Discontinuous POD (d) Adaptive Discontinuous POD

Figure 3.13: Solutions to the Watanabe-Maynard extrapolation problem for all four methods with 24 basis functions per node, on average in the case of ADPOD.

Figure 3.13 depicts the scalar flux solutions produced for the extrapolation problem by each method with 24 basis functions per node, on average in the case of ADPOD. The full order solution to this problem is shown in figure 3.9b. The S_6 solution exhibits significant ray effects which severely impact the accuracy of the solution. POD reduces these ray effects somewhat, but makes them less regular and so increases error in some regions. It is particularly poor at resolving the source region and its surroundings, which leads to a more inaccurate peak flux than even the S_6 solution. DPOD improves upon the previous method - ray effects are significantly reduced, the source region is well

resolved and the peak flux is relatively accurate. However, some visible ray effects are still present due to the low number of basis functions used. ADPOD further reduces ray effects compared to DPOD, and has a similar error in its peak flux, resulting in the most accurate solution of the four. This further demonstrates the advantages of DPOD and ADPOD in resolving scattering problems with few basis functions, and provides more evidence that the adaptive method is effective.



(a) -x, +y octant (b) +x, +y octant

(c) -x, -y octant (d) +x, -y octant

Figure 3.14: Number of ADPOD basis functions per octant per node for the duct extrapolation problem, with a mean of 84 basis functions per node in total.

Figure 3.14 depicts the number of ADPOD basis functions per node for each element and octant in the domain, with a mean of 84 basis functions per node in total. In contrast to the dog-leg duct results, a significant amount of flux was present in all four octants due to scattering, which explains the reduced effectiveness of the adaptive algorithm for this problem when compared with the dog-leg duct problem. Despite this, the algorithm was able to distribute the added basis functions in order to maximise their contribution to the solution. Most basis functions were added to the top right octant, which contained the majority of the flux due to the problem's geometry. In the other three octants, basis functions were primarily added downstream of the source and scattering regions and the

reflective boundaries. This optimisation enabled ADPOD to improve upon POD and DPOD in terms of error per basis function, despite the broad angular flux distribution in the Watanabe-Maynard problem.

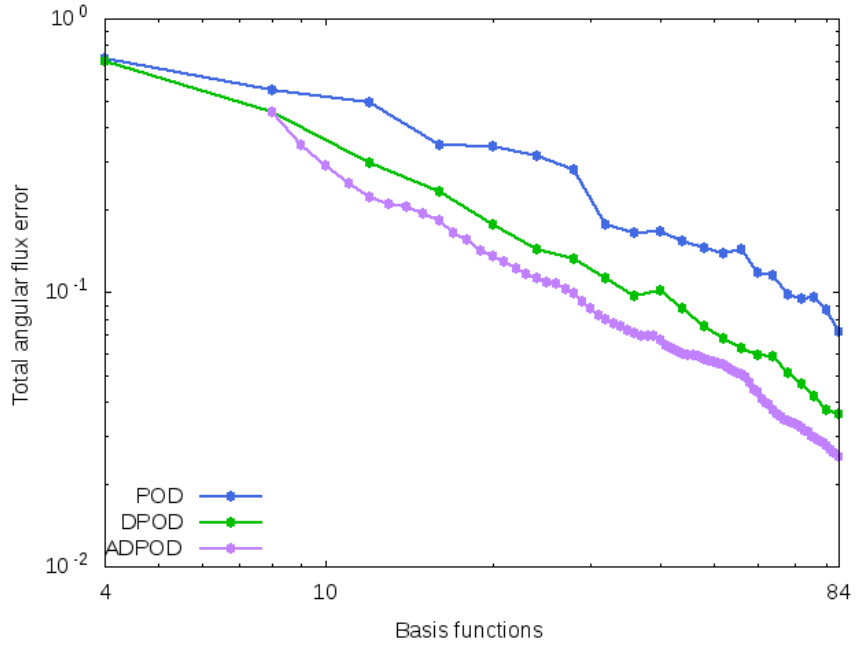
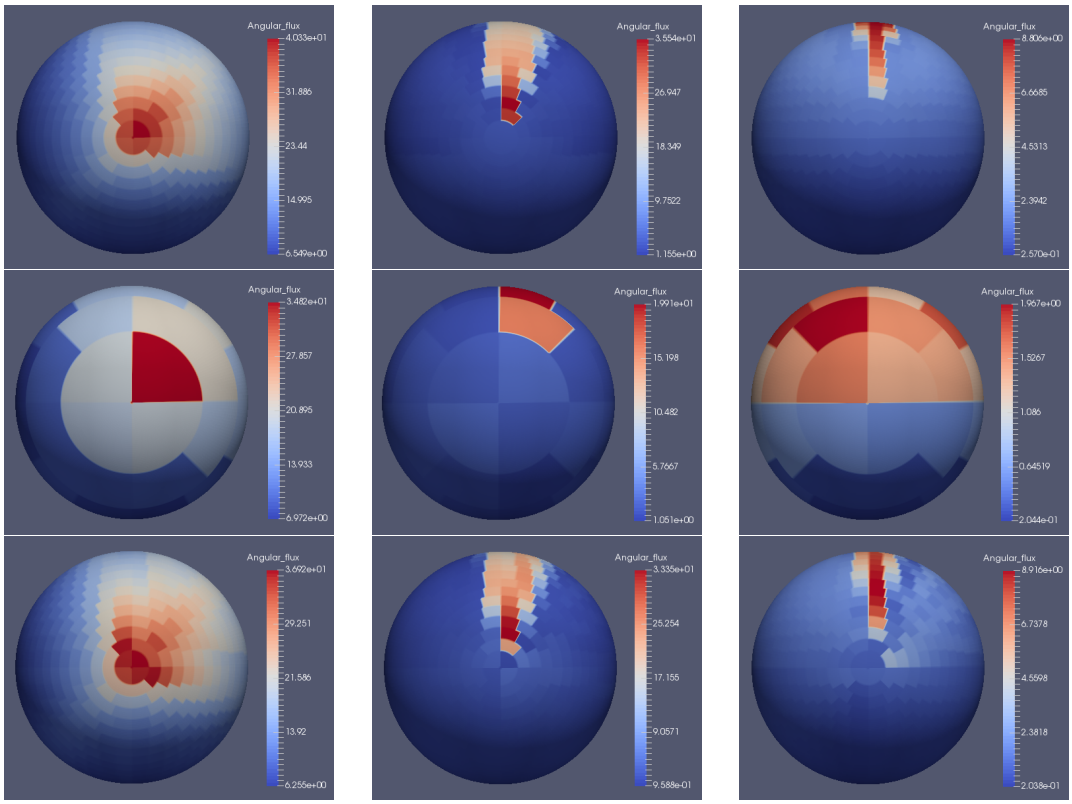


Figure 3.15: The effect of increasing basis function count on relative angular flux error for the Watanabe-Maynard extrapolation problem.

Figure 3.15 presents the relative angular flux error against the mean number of basis functions per node. These are similar to the results for the dog-leg duct problem - POD has the highest error, then DPOD, and ADPOD has the lowest, providing further evidence for the efficacy of DPOD and the adaptive method. However, ADPOD does not improve upon DPOD to the same extent as in the dog-leg duct problem. As mentioned previously, this is expected because the scattering which is present ensures that some amount of flux is present in most octants for most elements, and so basis functions must be distributed more evenly.



(a) x = 0.625cm, y = 0.625cm (b) x = 0.625cm, y = 4.375cm (c) x = 0.625cm, y = 8.750cm

Figure 3.16: Angular flux distributions at various points in the Watanabe-Maynard extrapolation problem. From top to bottom, S_{30}, S_6, and adaptive discontinuous POD. ADPOD and S_6 both use a mean of 24 basis functions per node, while S_{30} has 480.

Figure 3.16 depicts the angular flux profiles at three points within the domain. The locations of each point are shown in red in figure 3.9a. S_6 and ADPOD both used 24 basis functions, while S_{30} has 480. The S_6 discretisation is highly inaccurate, as expected. It performs particularly poorly in the scattering region (c), where the flux is distributed over a much wider range of angles than it should be. ADPOD performs far better, producing a reasonable approximation to the expected result. However, the errors in the ADPOD solutions are more visible in this case than previously, particularly for the +x, +y octant. This is likely because the problem geometry ensures that this octant contains the most flux, and scattering complicates the angular flux distributions. The basis functions in that octant must therefore describe the widest range of angular flux distributions, and will likely struggle the most. Further addition of basis functions to that octant by the adaptive algorithm should alleviate this issue.

3.8 Discussion

This chapter has presented a new reduced order model for discretising the angular dimension of the BTE, which is based upon POD and the method of snapshots [15]. The snapshots are formed from angular flux vectors taken from each spatial node, in a similar manner to previous work on angular POD [11]. Snapshots are included from multiple variations of the same problem with differing material properties, in order to enhance the ability of the model to solve problems with unseen material properties. The novel aspects of this work are the discontinuous formulation of the POD basis functions, which each span one octant in angle rather than the entire sphere, and the implementation of angular adaptivity using these discontinuous basis functions.

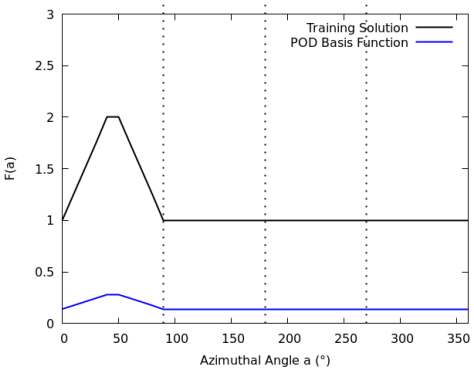
For the same angular size, the method of DPOD is shown to consistently reduce the error for a given basis size slightly when compared to POD, which is itself more accurate than S_N by up to half an order of magnitude. Additionally, the number of solver iterations required to converge each solution was both reduced by up to 1.8 orders of magnitude, and made more stable - solver iterations oscillated with basis size when using POD, while with DPOD they converged to the same value and did not oscillate. When combined, these two factors reduced solve times by up to 2 orders of magnitude compared to POD for the dog-leg duct problem, and by up to 1.5 orders of magnitude for the Watanabe-Maynard problem. This data demonstrates that the solver stability issue with POD is fixed by the introduction of DPOD, which was the primary goal of this chapter, and that the new method also reduced error slightly, which was a secondary goal. However, the reduction in error was relatively minor in most cases, excluding adaptivity.

In both numerical examples, the implementation of angular adaptivity via ADPOD provided further reductions in error. This was particularly successful in the dog-leg duct problem, where ADPOD reduced error for a given number of basis functions by up to an order of magnitude over DPOD, and thereby reduced solve times by up to 3 orders of magnitude in total compared to POD. For the Watanabe-Maynard problem, the reduction in error compared to DPOD was by a factor of 2-3, and the total reduction in solve time compared to POD was by up to 1.8 orders of magnitude. As mentioned previously, the disparity is likely a result of the scattering which was present in the Watanabe-

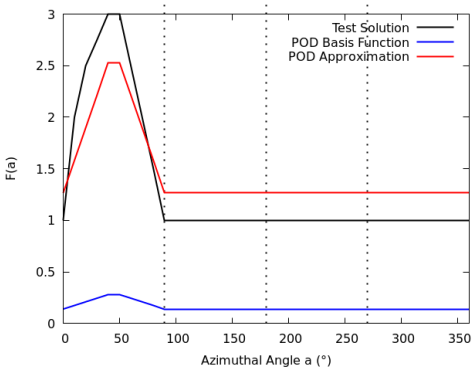
Maynard problem, which ensured that angular flux was present in most combinations of octant and element. The adaptive method therefore distributed angular resolution more broadly across the space-angle phase-space, resulting in reduced effectiveness. However, adaptivity was still effective in both cases, as expected. From the flux profiles examined, it seems likely that the reduction of ray effects provided by ADPOD was a significant contributor to the decreased error.

The benefits of DPOD and ADPOD compared to both POD and low-order S_N approximations have been clearly demonstrated in this chapter. In addition to fixing the solver instability and drastically reducing the iterations required to converge as a result, the new methods also reduced the error per basis function somewhat. As both of these factors contribute to the computational cost of producing an approximation with a given level of error, when they are both reduced the computation required decreases significantly.

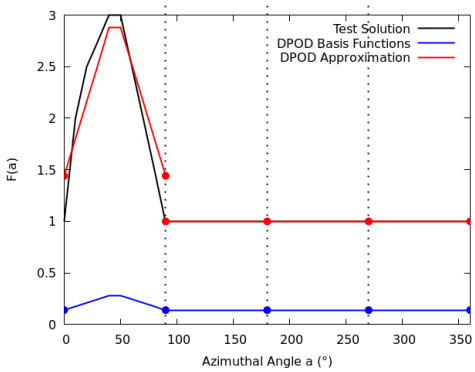
Since the DPOD method does not merely reduce the aforementioned solver instability but completely eliminates it, there must be a qualitative difference between the two methods in this regard. It is suspected that the solver instability of the previous angular POD method is caused by the incompleteness of the POD basis, combined with the global nature of the basis functions it produces. As the basis is incomplete, it is not possible to accurately represent all possible angular flux distributions. Since the basis functions are global, a change in the coefficient of a particular function will, in general, change the solution across the entire sphere. Figure 3.17 provides a simple demonstration which will be used to discuss this issue further.



(a) A training solution on the sphere, and the POD basis function which describes it exactly.



(b) A test solution on the sphere, and an approximate solution produced with the POD basis function from 3.17a.



(c) A test solution on the sphere, and an approximate solution produced with DPOD basis functions from 3.17a.

Figure 3.17: POD vs DPOD demonstration.

Figure 3.17a plots the value of a single training solution in black. The solution is dependent only on the azimuthal angle for simplicity, though the argument can easily be extended to additional dimensions. In blue, the optimal POD basis function for representing this solution is shown - in this case, it is simply a normalised version of the function itself. Figures 3.17b and 3.17c show a test solution, which has a different shape and a larger magnitude than the training solution in octant 1. In the rest of the domain, both the training and test solutions are equal to one.

The red line in figure 3.17b shows a POD approximation to the test solution, which is simply a scalar multiple of the single basis function, such that the POD approximation has the same magnitude as the test function. However, the approximation erroneously contains additional flux in octants 2-4, as if unphysical scattering had occurred. Figure 3.17c depicts a DPOD approximation to the same test solution, with the same training solution as the previous plots. In this case, discontinuities in the basis and solution functions are present at the locations with dots. Decoupling the fluxes in each octant

ensures that the angular flux in octants 2-4 is not affected by the change in octant 1. While the solution does become inaccurate, the error is contained within the octant which changed, and no false scattering into other octants occurs. On a structured quadrilateral mesh, the octant boundaries line up with the element boundaries such that each octant will advect into a different element. Therefore, the additional flux in octants 2-4 for the POD solution will erroneously advect into the wrong elements, increasing the number of iterations required to converge to a solution. By contrast, the DPOD solution ensures that the correct amount of flux advects into each element even if the distribution within each octant is not exactly correct.

Chapter 4

A Space-Angle Partitioned Reduced Order Model for the Angular Dimension of the Boltzmann Transport Equation

This chapter describes a novel reduced order model for the angular discretisation of the BTE, known here as Regional Discontinuous Proper Orthogonal Decomposition (RDPOD). It develops upon the DPOD method presented in chapter 3, and intends to improve the computational efficiency of the previous model by reducing the number of basis functions required to reach a given level of accuracy. The new model partitions the domain in space as well as angle, and forms independent sets of angular basis functions for each partition of the space-angle phase-space of the BTE. As the neutron flux distribution often varies substantially over space and angle, capturing the characteristics of the neutron flux using a single set of basis functions can place high demands on the original ROM formulation. Partitioning the spatial and angular dimensions and creating separate anguar ROMs for each partition can help to overcome this issue. As the variation in neutron flux distributions within each partition is considerably smaller than the variation across the entire problem, the number of basis functions required to resolve each distribution is reduced substantially. This can lead to smaller systems of

equations, and therefore reduce solve times, as demonstrated in this chapter.

The chapter is organised as follows: First, the process of discretising the BTE is briefly summarised. Next, the RDPOD method is derived in detail, along with a simple yet effective implementation of angular adaptivity using the RDPOD basis functions. Numerical results for a range of problems are then presented to demonstrate the effectiveness of RDPOD compared to previous methods. Finally, conclusions are drawn regarding the effectiveness of RDPOD, based on the numerical results presented.

4.1 Discretisation of the Boltzmann Transport Equation

The processes of angular and spatial discretisation closely follow sections 3.2 and 3.3, respectively. A summary of the discretisation process is presented here.

The steady-state, mono-energetic BTE is given by:

$$\Omega \cdot \nabla \Psi(\vec{r}, \Omega) + \Sigma_t(\vec{r}) \Psi(\vec{r}, \Omega) = q_{ex}(\vec{r}) + q_s(\vec{r}, \Omega' \rightarrow \Omega). \tag{4.1}$$

The angular flux $\Psi(\vec{r}, \Omega)$ is approximated by a sum of N_a angular basis functions $\mathcal{G}_j(\Omega)$ multiplied by the coefficients $\Psi_j(\vec{r})$,

$$\Psi(\vec{r}, \Omega) \approx \sum_{j=1}^{N_a} \mathcal{G}_j(\Omega) \Psi_j(\vec{r}). \tag{4.2}$$

The approximation in equation 4.2 is inserted into equation 4.1, which is then weighted and integrated over all angles. The basis functions $\mathcal{G}$ are also used as weighting functions. The resulting angularly discretised equations can be expressed in matrix form as,

$$(A \cdot \nabla + H(\vec{r})) \Psi(\vec{r}) = Q(\vec{r}), \tag{4.3}$$

where A is a vector of matrices (A_x, A_y, A_z), the removal and scattering terms are grouped into the matrix H, the coefficients of the angular expansion are contained in the vector $\Psi(\vec{r})$, and the angularly discretised source term is represented by the vector $Q(\vec{r})$. All matrices are of size $N_a \times N_a$, and all vectors besides A are of size N_a.

The discontinuous Galerkin finite element method is applied to discretise the spatial

dimensions of the BTE, as described in section 3.3. The angularly discretised flux $\Psi(\vec{r})$ in equation 4.3 is approximated as a sum of the N_s spatial basis functions $\mathcal{N}_j(\vec{r})$ multiplied by the coefficients ψ_j,

$$\Psi(\vec{r}) \approx \sum_{j=1}^{N_s} \psi_j \mathcal{N}_j(\vec{r}). \tag{4.4}$$

This approximation is inserted into equation 4.3, which is then converted to its weak form by weighting with the set of basis functions $\mathcal{N}(\vec{r})$ and integrating over the volume of each element, V_e. Green's theorem is applied to the advection term, splitting it into an integral over V_e and another over the element boundary Γ_e. Finally, the boundary term is split into inflow and outflow components, as first order upwinding is used to calculate the flow at element boundaries [49, 50]. The result is a full order discretised formulation of the BTE,

$$\begin{aligned}\sum_{j=1}^{N_s} \Bigg\{ \Big(-\int_{\mathrm{V}_e} \nabla\mathcal{N}_i(\vec{r}) A \mathcal{N}_j(\vec{r}) d\mathrm{V} + \int_{V_e} \mathcal{N}_i(\vec{r}) H(\vec{r}) \mathcal{N}_j(\vec{r}) d\mathrm{V} \Big)\psi_j \\ + \int_{\Gamma_e} \mathcal{N}_i(\vec{r})(A^{in}\cdot\hat{n})\mathcal{N}_j(\vec{r}) d\Gamma_e \psi_j^{in} + \int_{\Gamma_e} \mathcal{N}_i(\vec{r})(A^{out}\cdot\hat{n})\mathcal{N}_j(\vec{r}) d\Gamma_e \psi_j^{out} \Bigg\} \\ = \int_{\mathrm{V}_e} \mathcal{N}_i(\vec{r}) Q(\vec{r}) d\mathrm{V},\end{aligned}$$

$$\forall i \in \{1, N_s\}, \tag{4.5}$$

where $\hat{n}$ is the unit vector normal to the element boundary; ψ_j^{out} is the outflow, given by the angular flux vector of the element in question; and ψ_j^{in} is the inflow, given by the angular flux vectors of the element's upstream neighbours.

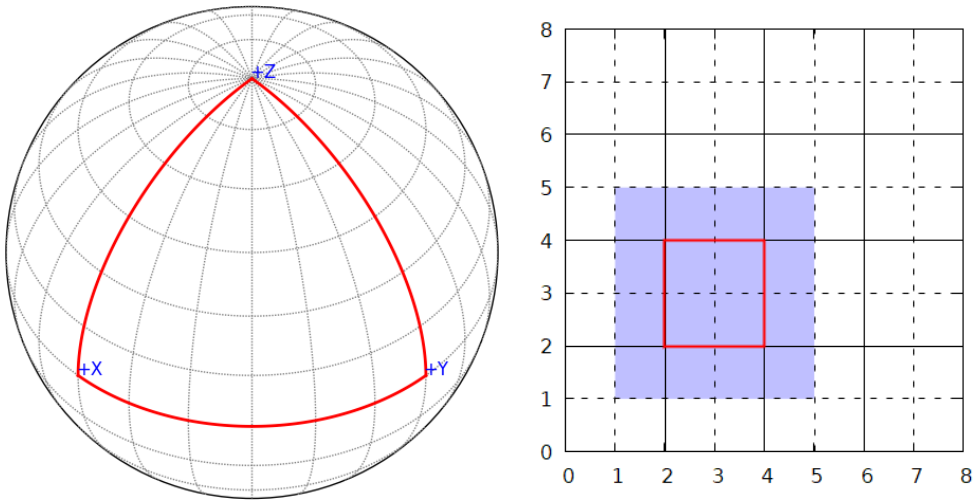
4.2 The Discontinuous Angular Reduced Order Model

This section presents a new angular POD method that forms independent angular basis sets, which are each optimised to resolve a specific partition of the space-angle phase-space of the BTE. The angular domain Ω and spatial domain V are partitioned into the two sets,

$$\Omega = \bigcup_{q=1}^{8} \Omega_q \qquad \text{and} \qquad V = \bigcup_{r=1}^{n_r} V_r, \tag{4.6}$$

where Ω_q and V_r represent subsets of the angular and spatial domains, respectively, as illustrated in figure 4.1. n_r is the number of spatial regions. The DPOD method

described in chapter 3 corresponds to the use of a single spatial region spanning the domain, that is, n_r=1. If a single angular partition spanning the full sphere is also used, then the angular POD method described in [11] is recovered.



(a) An angular octant on the sphere. The red lines delimit the boundaries. Each basis function is non-zero in one out of eight octants.

(b) A spatial region within an 8x8 element domain which is partitioned into 4x4 regions. Dashed black lines are element boundaries, solid black lines are element and region boundaries, and the red square encloses the region which takes snapshots from all elements shaded blue.

Figure 4.1: The spatial and angular discontinuities imposed by the RDPOD method.

In line with the DPOD method, the angular domain is partitioned into eight octants in three dimensions, though solutions for two dimensional problems are mirrored through the spatial plane, and so only four octants must be solved for. Each spatial partition may in principle contain any arbitrary set of elements in the spatial domain, though in this project regular rectangular regions were used. Henceforth, the number of regions used for a particular ROM will often be expressed as $n_{rx} \times n_{ry}$, where n_{rx} and n_{ry} denote the number of columns and rows which the domain is divided into, respectively. The total number of regions n_r is therefore given by $n_r = n_{rx} \times n_{ry}$. A cross-product of the two partitions in equation 4.6 forms the space-angle partition of the complete phase-space, which is given by,

$$Z = \bigcup_{qr} Z_{qr}, \tag{4.7}$$

where the phase-space partition spanning the angular octant Ω_q and spatial region V_r is denoted Z_{qr}.

The POD functions are formed for each partition via the method of snapshots [143]. The

angular coefficients from the S_N model are used to form the snapshot matrices for each partition. The vectors of angular coefficients formed at the nodes of the spatial mesh are first partitioned into sets according to their angular region as described in section 3.5,

$$\psi = \begin{bmatrix} \psi_1 \\ \vdots \\ \psi_q \\ \vdots \\ \psi_8 \end{bmatrix}, \tag{4.8}$$

where ψ_q is a vector containing the n_q coefficients with associated directions within Ω_q. Each component of ψ_q is then assigned to a spatial region V_r based on its position. Separate snapshot matrices S_{qr} are formed for each partition Z_{qr}, resulting in $8 \times n_r$ matrices in three dimensions or half as many in two.

The snapshot matrices S_{qr} also include vectors of angular coefficients from the elements adjacent to V_r, as this helps to preserve information when mapping between regions. This is shown in figure 4.1, where the region enclosed in red takes snapshots from all elements shaded in blue. The first numerical example in section 4.4 presents the reasoning behind this decision.

For each Z_{qr}, the associated snapshot set is therefore defined as,

$$\mathrm{S}_{qr} = \begin{bmatrix} | & | & & | \\ \psi_{q,r,1} & \psi_{q,r,2} & \dots & \psi_{q,r,n_h} \\ | & | & & | \end{bmatrix}, \qquad \forall q \in \{1,8\}, \forall r \in \{1, n_r\}, \tag{4.9}$$

where each $\psi_{q,r,i}$ is a vector of size $N_a/8$ containing the angular coefficients of the ith snapshot associated with the partition Z_{qr}. The term n_h, which may vary for each snapshot set, denotes the total number of snapshots in S_{qr}. This is given by $n_h = N_r \times N_p$, where N_r is the number of FEM basis nodes in partition V_r, including the neighbouring nodes as shown in blue in figure 4.1, and N_p is the number of problem variations used to train the ROM.

The RDPOD basis sets for each partition Z_{qr} can now be formed through the SVD of

each snapshot matrix,

$$\mathrm{S}_{qr} = \mathrm{U}_{qr}\Sigma_{qr}\mathrm{V}_{qr}^{T}, \tag{4.10}$$

where U_{qr} and V_{qr} are unitary matrices of sizes $N_a/8 \times N_a/8$ and $n_h \times n_h$, respectively. The column vectors of U_{qr} contain the optimised basis vectors that best represent the snapshot data, ordered such that the first n_a columns form the optimal n_a basis vectors in the Frobenius norm. Here, n_a is the same for all q and r, though this is not required. A method for varying the number of basis functions by region and octant is described in section 4.3. The RDPOD basis matrices $\mathcal{U}_{qr}$ are formed by truncating each snapshot matrix such that only the first n_a columns are retained. The fraction of the information in U_{qr} which is retained in $\mathcal{U}_{qr}$ can be determined from the singular values,

$$I_{qr}(n_a) = \frac{\sum_{i=1}^{n_a}(\Sigma_{q,r,i,i})^2}{\sum_{i=1}^{N_a}(\Sigma_{q,r,i,i})^2}, \tag{4.11}$$

where I_{qr} varies from 0 to 1, with 1 being total capture of the snapshot information. The matrices $\mathcal{U}_{qr}$ can be used to map the angular coefficients between the full and reduced order models through the relationship,

$$\Psi_q(\vec{r}) \approx \mathcal{U}_{qr}\alpha_{qr}(\vec{r}), \qquad \forall q \in \{1,8\}, \forall r \in \{1,n_r\}, \tag{4.12}$$

where $\alpha_{qr}(\vec{r})$ contains a vector of n_a coefficients of $\mathcal{U}_{qr}$, for each node within Z_{qr}. The combined mapping over all angular partitions in each region V_r can be expressed as,

$$\mathcal{U}_r\alpha_r(\vec{r}) = \begin{bmatrix} \mathcal{U}_{1,r} & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 \\ 0 & 0 & \mathcal{U}_{qr} & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & \mathcal{U}_{8,r} \end{bmatrix} \begin{bmatrix} \alpha_{1,r}(\vec{r}) \\ \vdots \\ \alpha_{qr}(\vec{r}) \\ \vdots \\ \alpha_{8,r}(\vec{r}) \end{bmatrix} = \begin{bmatrix} \mathcal{U}_{1,r}\alpha_{1,r}(\vec{r}) \\ \vdots \\ \mathcal{U}_{qr}\alpha_{qr}(\vec{r}) \\ \vdots \\ \mathcal{U}_{8,r}\alpha_{8,r}(\vec{r}) \end{bmatrix}, \qquad \forall r \in \{1,n_r\}. \tag{4.13}$$

Equations 4.8 and 4.12 enable this to be compactly written as,

$$\mathcal{U}_r\alpha_r(\vec{r}) \approx \Psi(\vec{r}), \qquad \forall r \in \{1,n_r\}. \tag{4.14}$$

Substituting equation 4.14 into equation 4.3 and premultiplying by $\mathcal{U}_r^T$ projects the angularly discretised equations onto the POD space,

$$\mathcal{U}_r^T(A\cdot\nabla+H(\vec{r}))\mathcal{U}_r\alpha_r(\vec{r})=\mathcal{U}_r^TQ(\vec{r}),\qquad\forall r\in\{1,n_r\}. \tag{4.15}$$

This projection is applied to each region V_r separately. Equation 4.15 is then spatially discretised as described in section 4.1, with $\alpha_r(\vec{r})$ taking the place of $\Psi_r(\vec{r})$ in the derivation. This results in the fully discretised RDPOD formulation of the BTE.

The spatially discretised forms of equation 4.15, equivalent to projections of equation 4.5 onto the RDPOD bases, are constructed separately for each spatial partition V_r using their own optimised RDPOD angular basis sets. Communication between the elements within each partition, and between neighbouring partitions, is implemented through the surface integrals of equation 4.5. As mentioned, a Riemann approach can be employed to form the incoming and outgoing matrices inside the surface integrals for a general angular discretisation. They can also be obtained through pre- and post-multiplication of the full order matrices A^{in} and A^{out} by the RDPOD mapping matrices $\mathcal{U}_r^T$ and $\mathcal{U}_r$. For adjacent elements within the same region V_r, the incoming and outgoing surface matrices of equation 4.5 are given by,

$$A_r^{in}=\mathcal{U}_r^TA^{in}\mathcal{U}_r,\qquad\text{and}\qquad A_r^{out}=\mathcal{U}_r^TA^{out}\mathcal{U}_r, \tag{4.16}$$

respectively. To obtain an element's incoming surface information from an adjacent element belonging to a different region, say $V_{r'}$, one must use the correct mapping to account for the fact that the incoming vector employs a different RDPOD basis. This is simply achieved by mapping the RDPOD coefficients $\alpha_{r'}$ from the incoming element to the full model space, applying the (S_N) incoming advection operator, then projecting the resulting vector onto the basis of region V_r. That is, the incoming advection matrices in the RDPOD formulation are given by,

$$A_{r,r'}^{in}=\mathcal{U}_r^TA^{in}\mathcal{U}_{r'}. \tag{4.17}$$

The matrices $A_{r,r'}^{in}$ are precomputed for each pair of regions r and r' with a common border. They can then be employed in equation 4.5, where the incoming surface integral

can be explicitly written as,

$$\sum_{j=1}^{N_s} \int_{\Gamma_e} \mathcal{N}_i(\vec{r}) A^{in}_{r,r'} \mathcal{N}_j(\vec{r}) d\Gamma_e \alpha^{in}_{r',j}, \qquad \forall i \in \{1, N_s\}. \tag{4.18}$$

As stated previously, in the case of a single spatial region spanning the domain, the DPOD method presented in [15] is recovered. Using a single spatial region together with a single angular region spanning the sphere, the original angular POD method in [11] is recovered.

4.3 Adaptivity in Angle

This section presents an adaptive algorithm using the RDPOD basis functions, known as adaptive RDPOD (ARDPOD). Instead of utilising the same number of basis functions n_a throughout the domain, each partition Z_{qr} has an associated number of basis functions n_{qr}, which can be modified independently. The adaptive algorithm uses an error metric to estimate the effect of increasing n_{qr} for each Z_{qr}, and adds basis functions where they are likely to minimise the total error.

The contribution of each basis function to the solution is dependent only on the magnitudes of its coefficients. The basis functions form a hierarchical set, which implies that the coefficients of each successive function will tend towards zero as the approximation converges. It can thus be inferred that additional basis functions are likely to be most beneficial in partitions where the coefficient of the final basis function currently included is large. However, successive coefficients have been observed to oscillate in magnitude, and so a single small coefficient does not guarantee convergence. The error metric therefore considers the final two coefficients.

An initial solution is required in order to calculate the error metric and begin the adaptive process, and so the problem in question is first solved with,

$$n_{qr} = 2, \qquad \forall q \in \{1, 8\}, \forall r \in \{1, n_r\}. \tag{4.19}$$

Next, the relative contribution from the final two basis functions is calculated for each

partition Z_{qr},

$$\mathrm{F}_{qr} = \frac{(\alpha_{q,r,n_{qr}-1} + \alpha_{q,r,n_{qr}})}{\sum_{q=1}^{8}\sum_{r=1}^{n_r}\sum_{i=1}^{n_{qr}} \alpha_{q,r,i}}, \qquad \forall q \in \{1,8\}, \forall r \in \{1,n_r\}, \tag{4.20}$$

where $\alpha_{q,r,i}$ denotes the i^{th} angular coefficient in the partition Z_{qr}. A threshold value τ is set to some inital, relatively large value, typically 1. A variable known as the threshold divisor, δ_τ, is set to some value greater than 1. In this chapter, $\delta_\tau = 2$. Each iteration, basis functions are added according to the equation,

$$n_{qr} = n_{qr} + 1, \qquad \forall\ (q,r) \in \{\ (q',r') \mid \mathrm{F}_{q',r'} > \tau\}. \tag{4.21}$$

The number of basis functions added in this adaptive stage, n_Δ, is counted and compared to the minimum number of basis functions to add per iteration, n_+. If $n_\Delta < n_+$, then $\tau = \tau/\delta_\tau$ and equation 4.21 is applied again with the new value of τ. The process repeats until $n_\Delta > n_+$, at which point the next iteration can begin. Once n_{qr} has been adjusted, the process repeats until a desired number of basis functions is reached. The complete adaptive method is described by algorithm 2.

Algorithm 2: Adaptive RDPOD

```
n_qr = 2 for all q and r.
/* Iterate until the desired number of basis functions is reached.
   */
while sum(n_qr) < maximumBasisFunctions do
    Solve the ROM.
    Calculate F_qr.
    n_Δ = 0;
    while n_Δ < n_+ do
        /* Loop through partitions and add basis functions.        */
        for q = 0 to 8 do
            for r = 0 to n_r do
                if F_qr > τ then
                    n_qr += 1;
                    n_Δ += 1;
                end
            end
        end
        /* Decrease the threshold value τ if necessary.            */
        if n_Δ < n_+ then
            τ = τ/δ_τ;
        end
    end
end
```

4.4 Numerical Examples

In this section, two numerical examples are presented in order to compare the performance of DPOD, RDPOD and ARDPOD. Uniform quadrilateral FEM spatial meshes are employed, using discontinuous bilinear basis functions. The full order method employs the S_N discretisation, with a sufficiently high angular resolution to ensure that the solutions have converged in angle. As mentioned in section 1.2, the sweep-based solver employed throughout is not fully optimised. Hence, whilst solve times are provided, they should be considered conservative estimates, and it is highly likely that reductions can and will be made in future. In addition, it was not practical to obtain exclusive usage of a computation node for each test, and therefore some error was introduced by the variance in other work performed by each node during the tests. However, this variance was found to be small enough that it did not significantly impact the quality of the data. The main purpose of this analysis is to demonstrate the methods' increased accuracy for a given basis size when compared to DPOD, which has been demonstrated to outperform both S_N and angular POD by the same metric in chapter 3.

In the examples presented, the spatial partitions used to construct the ROMs are of regular structure, despite equation 4.6 allowing for any arbitrary partitioning to be used. Whilst the ideal implementation would minimise the variation in angular flux profiles within each partition, generating partitions in this manner would require significant additional work. This chapter will instead focus on demonstrating that even regularly shaped partitions can drastically improve modelling efficiency. This demonstration therefore forms a foundation for future work on the generation of optimal partitions for further improvements to efficiency.

Finally, it is worth mentioning that the computer memory requirements of the RDPOD method are relatively small. The memory usage increases linearly with the number of regions, n_r, since each sub-region requires its own set of discretised angular matrices. However, as shown in the numerical examples, the angular sizes of the reduced systems are typically small, on the order of a few tens of basis functions. Thus, taking a large angular ROM with 100 basis functions for example, and using a reasonably large set of partitions, say $n_r = 1000$, only 1.5 GiB of memory is required for storage.

4.4.1 The Region Test Problem

The first example in this chapter is an extremely simple setup which demonstrates the benefit of including elements from the boundaries of each region in their snapshot matrices, known as boundary elements (BEs), as discussed in section 4.2. The problem is a 30×10cm material with Σ_t=1 and q_{ex}=1 throughout. Vacuum boundary conditions are applied to all domain boundaries. The domain is discretised using a 3×1 mesh of discontinuous bilinear quadrilateral elements in space, and 3x1 RDPOD regions are used for the reduced order model.

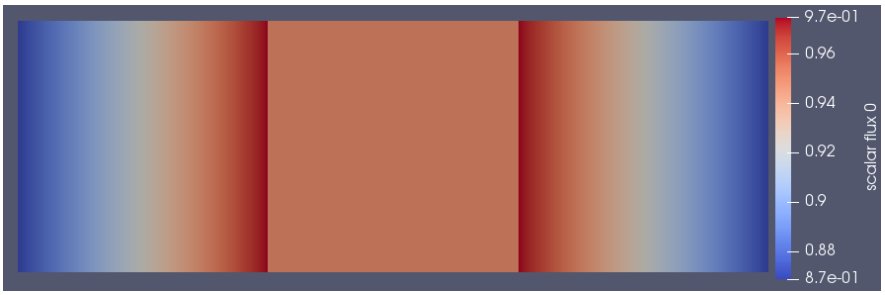


Figure 4.2: S_{20} scalar flux solution to the region test problem.

Figure 4.2 shows the scalar flux distribution of the S_{20} solution to the problem. The full order solution is quite poor due to the extremely low spatial resolution, but the relevant point here is how well the RDPOD model is able to reconstruct the solution, not the accuracy of the solution itself.

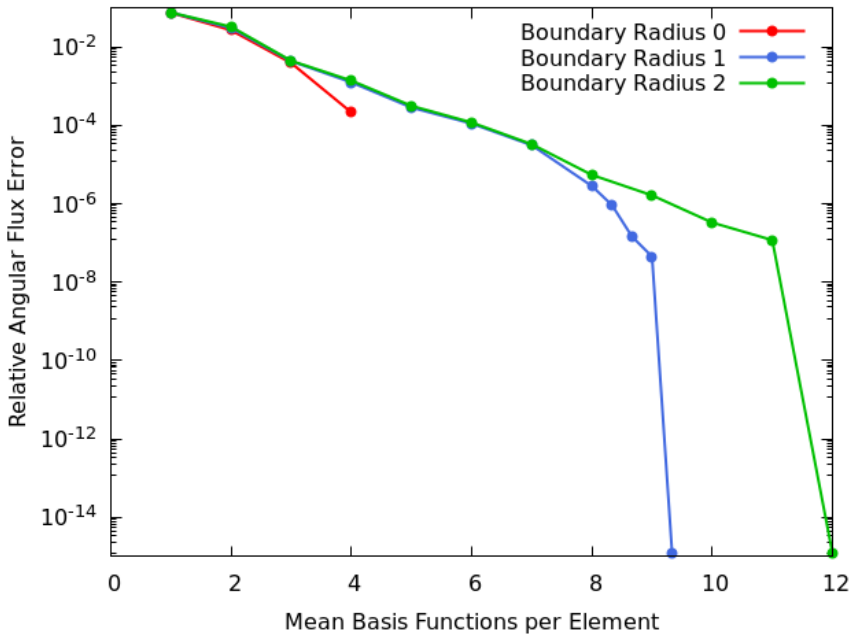


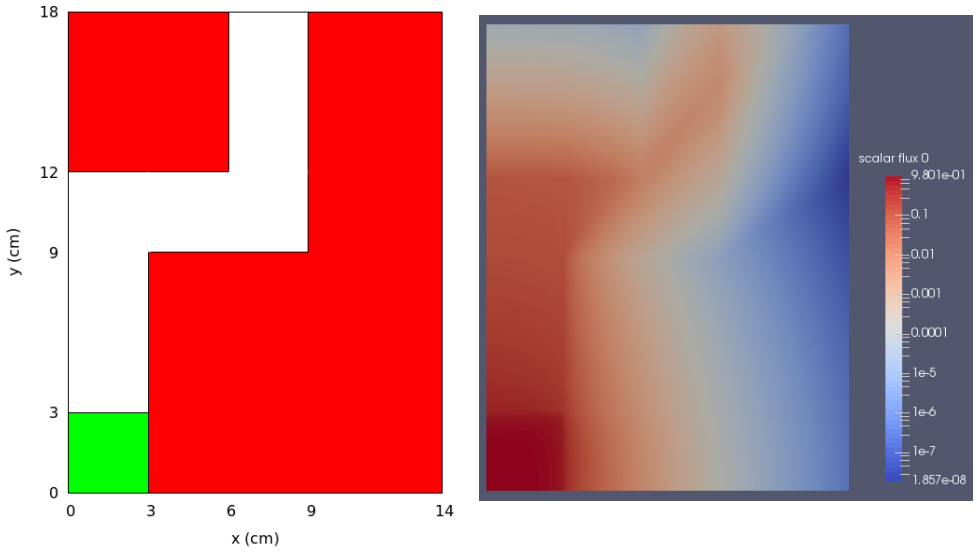
Figure 4.3: Relative angular flux error vs mean basis functions per node for the region test problem, with varying numbers of boundary elements.

Figure 4.3 shows the relative angular flux error against the mean number of basis functions per node for the region test problem, with 3x1 spatial elements. Each line represents a different boundary radius (BR), which governs the number of BEs included in the snapshot matrices. With a BR of 0, no neighbours are included. Each matrix therefore contains 4 snapshots, and the SVD only produces 4 basis functions per region. With a BR of 1, elements which are adjacent to each region are included as snapshots for that region. Therefore, the two side elements have 8 snapshots in their snapshot matrices, since they only have one adjacent element. The central element contains 12 snapshots, as it is adjacent to both of the other elements. The total number of basis functions available is therefore 28, and the mean is 9.3 recurring when all 28 are used. With a BR of 2, elements which are adjacent to the previous BEs are also included as snapshots for each region. Each of the snapshot matrices therefore contains 12 snapshots, which constitutes the entire problem. This is equivalent to solving the problem with non-regional DPOD, since all three regions' basis matrices are identical and contain the entire solution as snapshots.

As the plot shows, all three solutions produce very similar results at first. Upon reaching its maximum number of basis functions, 4, the solution with BR 0 performs slightly better than the other two, which continue to produce near-identical results. However,

the solutions with BEs quickly improve upon the peak performance of the solutions with none. When the final basis function for each solution with BEs is included, the error drops precipitously, decreasing by over 6 orders of magnitude to approximately machine error. However, this occurs later for the solution with a BR of 2, as more basis functions are available. This suggests that including BEs removes a limitation on the peak performance of the method, while having a negligible impact on solutions which are yet to reach this limit.

4.4.2 The Dog-Leg Duct Problem



(a) Dog-leg duct schematic

(b) S_{50} scalar flux solution to interpolation problem

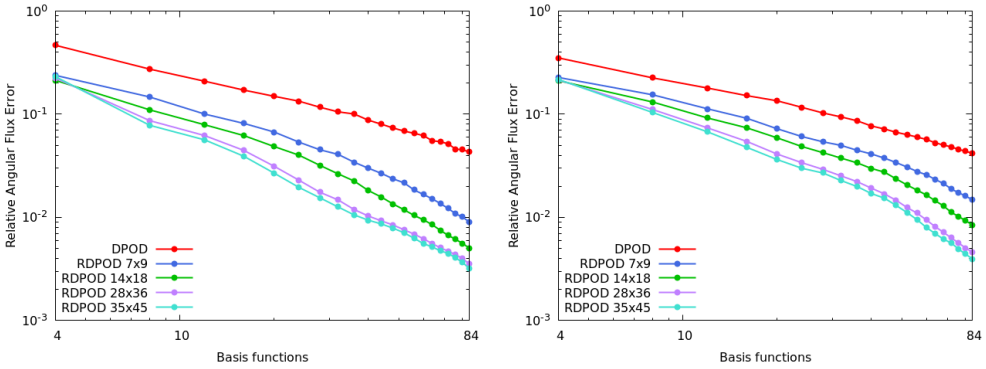
Figure 4.4: Schematic (a) and S_{50} scalar flux solution (b) for the dog-leg duct interpolation problem. The green region is the source, the white region is the duct, and the red region is a highly absorbing material.

The second example in this chapter is a dog-leg duct problem [12]. Section 1.3.1 introduces the problem, and section 3.7.1 gives more detail about the discretisations used in this work. However, in this chapter a wider range of material properties were examined in order to investigate the limitations of the model further. The new set of material properties used for both training and test problems is given in table 4.1. In addition, the S_{50} angular discretisation was used rather than S_{40}, as the efficiency of the full order solver had been improved since the previous data was gathered. Figure 4.4a shows a schematic of the dog-leg duct problem, and figure 4.4b shows the scalar flux distribution of the S_{50} solution to the interpolation problem.

Problem	Material	Source ($cm^{-2}s^{-1}$)	Σ_a (cm^{-1})	Σ_s (cm^{-1})
1	Green	1.00	0.50	0.00
	White	0.00	0.00	0.00
	Red	0.00	0.50	0.00
2	Green	1.00	1.50	0.00
	White	0.00	0.00	0.00
	Red	0.00	1.50	0.00
3	Green	1.00	0.50	0.00
	White	0.00	0.05	0.00
	Red	0.00	0.50	0.00
4	Green	1.00	1.50	0.00
	White	0.00	0.05	0.00
	Red	0.00	1.50	0.00
Interpolation	Green	1.00	1.00	0.00
	White	0.00	0.025	0.00
	Red	0.00	1.00	0.00
Extrapolation	Green	1.00	2.00	0.00
	White	0.00	0.10	0.00
	Red	0.00	2.00	0.00

Table 4.1: Material properties for the dog-leg duct problems in chapter 4.

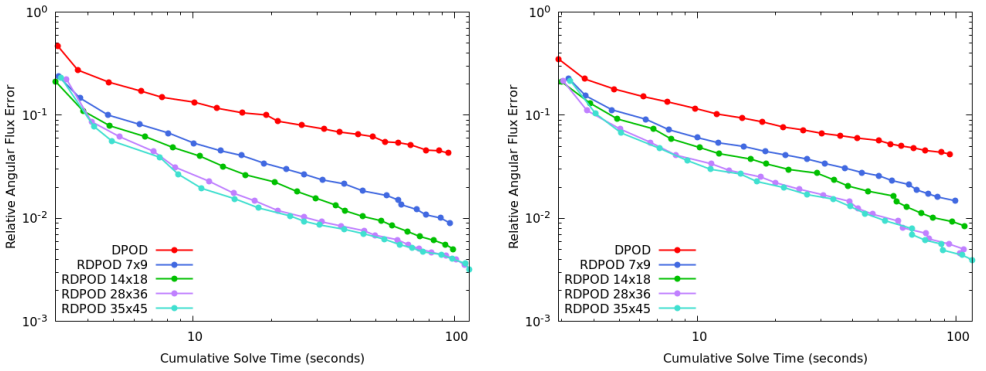
The snapshot matrix was formed from all four training solutions, and the resulting RDPOD bases were used to solve the test problems. As in prior chapters, the first test problem is referred to as the *interpolation* problem, as its material properties are within the range for which snapshots were produced, and the second is referred to as the *extrapolation* problem, as its material properties lie outside of this range.



(a) Interpolation (b) Extrapolation

Figure 4.5: Angular flux error vs number of basis functions for S_{50} solutions to the dog-leg duct problems with varying numbers of spatial regions. "$a \times b$" indicates a spatial partitions in x and b in y, for a total of $a \times b$.

Figure 4.5 compares the L^2-norms of the relative angular flux errors for the dog-leg duct interpolation and extrapolation problems using DPOD and RDPOD, with varying numbers of basis functions and regions. The results show that RDPOD consistently reduced error compared to DPOD for a given angular basis size. Furthermore, increasing the number of spatial partitions reliably reduced the error further. For the largest set of spatial partitions, the error was reduced by approximately 1 order of magnitude with 16 basis functions. The reduction in error compared to DPOD continued to grow as the angular resolution was increased further.

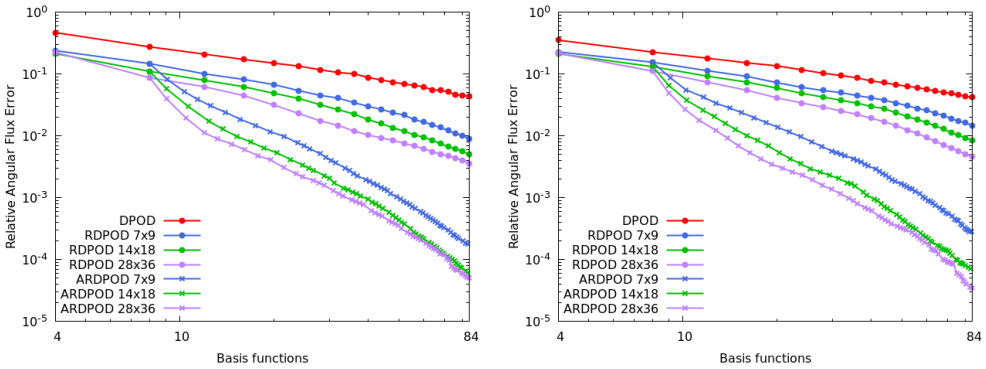


(a) Interpolation (b) Extrapolation

Figure 4.6: Angular flux error vs solve time in seconds for S_{50} solutions to the dog-leg duct problems with varying numbers of spatial regions. "$a \times b$" indicates a spatial partitions in x and b in y, for a total of $a \times b$.

Figure 4.6 presents the L^2-norm of the relative angular flux error against the solve time for both interpolation and extrapolation problems, for varying numbers of basis functions and regions. Similar trends are observed, with RDPOD reducing the solve

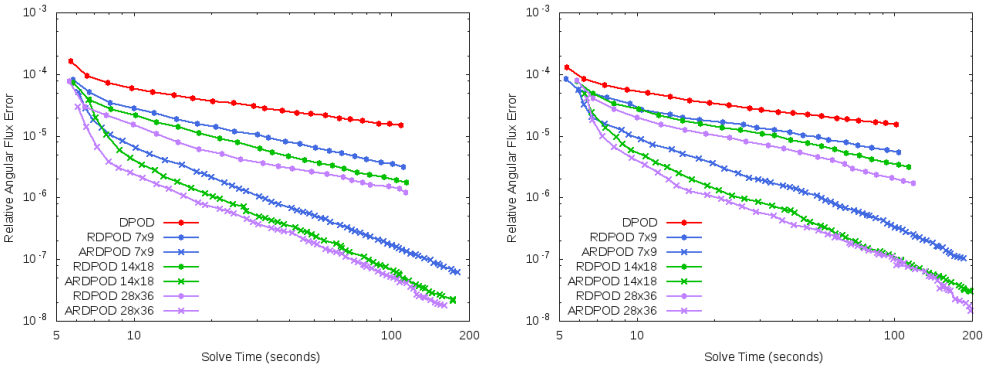
time required to achieve a given error. For the largest spatial partition sets, the solve times were reduced by over an order of magnitude with a relative error of around 10^{-5}. The trends in the graph also indicate that greater efficiency would be achieved for lower error tolerances.



(a) Interpolation (b) Extrapolation

Figure 4.7: Angular flux error vs number of basis functions for S_{50} solutions to the dog-leg duct problems with varying numbers of spatial regions using angular adaptivity.

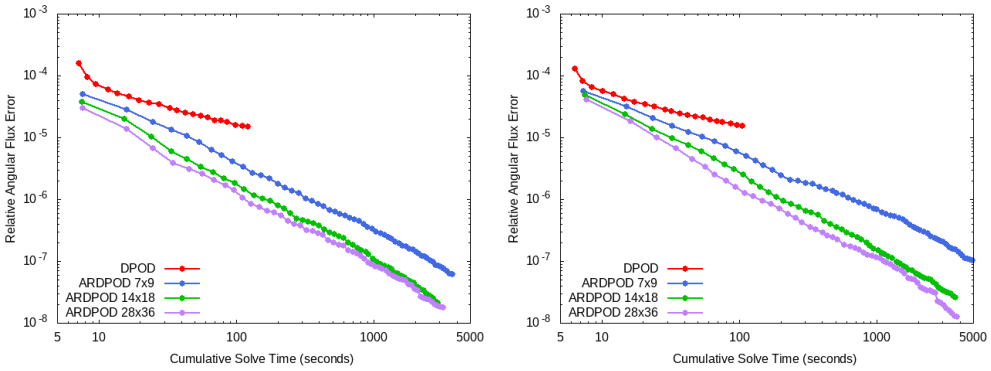
Figure 4.7 presents the relative angular flux error vs the mean number of basis functions per node for DPOD, and for RDPOD and ARDPOD with varying numbers of regions. It is shown that ARDPOD significantly reduced the error for a given angular basis size. With just 12 basis functions, the error was reduced by approximately half an order of magnitude compared to RDPOD, and by more than an order of magnitude compared to DPOD. By 84 basis functions, the reductions in error for a given basis size had increased to almost 2 orders of magnitude compared to RDPOD and approximately 2.5 orders of magnitude compared to DPOD.



(a) Interpolation (b) Extrapolation

Figure 4.8: Angular flux error vs solve time in seconds for S_{50} solutions to the dog-leg duct problems with varying numbers of spatial regions using angular adaptivity. "$a \times b$" indicates a spatial partitions in x and b in y, for a total of $a \times b$.

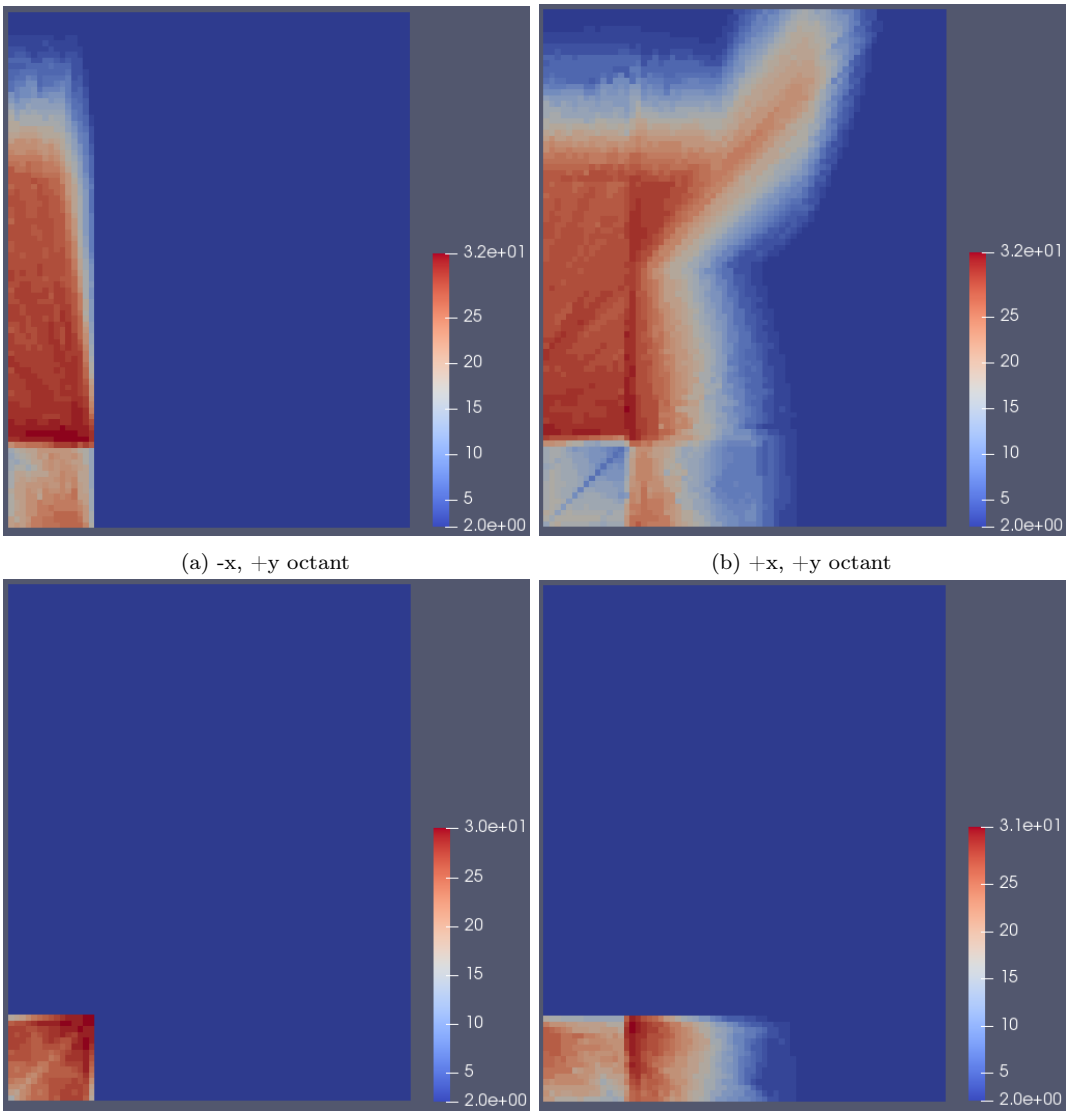
Figure 4.8 presents the relative angular flux error against the solve time for DPOD, RDPOD and ARDPOD. The adaptive solve times show the time taken to complete the final iteration of the adaptive process, starting from a zero solution. This gives an indication of the performance of ARDPOD once the adaptive stage is complete and an efficient basis function distribution has been generated. The adaptive method drastically reduced the solve time required to reach a given level of error in comparison to both DPOD and RDPOD. At around 20 seconds of solve time, the ARDPOD errors were reduced by just over an order of magnitude in comparison to RDPOD, and by around 2 orders of magnitude when compared to DPOD.



(a) Interpolation (b) Extrapolation

Figure 4.9: Angular flux error vs cumulative solve time in seconds for S_{50} solutions to the dog-leg duct problems with varying numbers of spatial regions using angular adaptivity. "$a \times b$" indicates a spatial partitions in x and b in y, for a total of $a \times b$.

Figure 4.9 presents the relative angular flux error against the total solve times using ARDPOD. The total solve time is the time to complete the entire adaptive solution process, up to and including the specified basis size. While the ARDPOD solver was far from optimised, as discussed in section 1.2, the results still demonstrate that ARDPOD reduced the angular flux error by up to an order of magnitude for a given solve time when compared to DPOD, and that the error exhibited an increase in order of convergence.



(a) -x, +y octant

(b) +x, +y octant

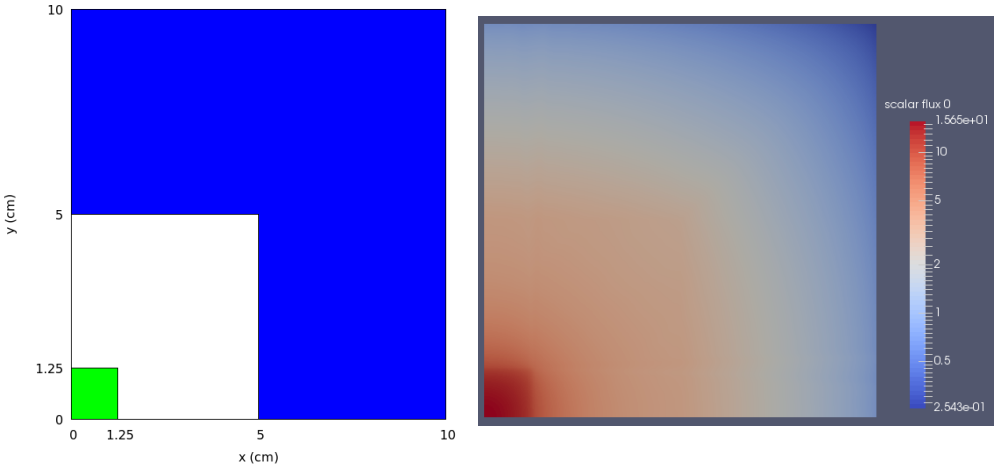
(c) -x, -y octant

(d) +x, -y octant

Figure 4.10: The number of ARDPOD basis functions per octant in each spatial region, for the dog-leg duct extrapolation problem, with 70x90 regions and a mean of 84 basis functions per node in total.

Figure 4.10 shows the number of basis functions associated with each octant in each spatial region, for the dog-leg duct extrapolation problem. The plots show that basis functions were predominantly added to high flux regions, such as inside the duct and at its borders. This demonstrates that the adaptive method successfully increased the resolution in regions where additional basis functions were likely to be most beneficial, and thus explains the method's success in reducing the error for a given number of basis functions.

4.4.3 The Watanabe-Maynard Problem



(a) Schematic

(b) S_{50} scalar flux solution

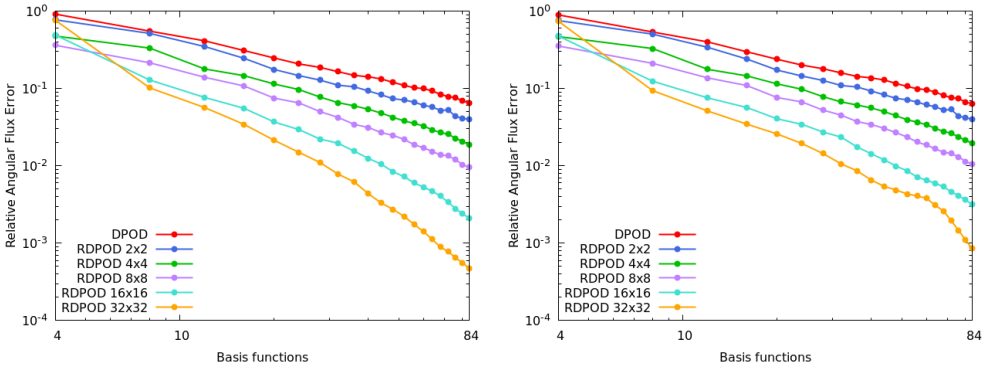
Figure 4.11: Schematic (a) and S_{50} scalar flux solution (b) for the Watanabe-Maynard interpolation problem. The green region is the source, the white region is a void and the blue region is a moderately absorbing and scattering material.

The third example in this chapter is the Watanabe-Maynard problem, introduced in [13] and discussed previously in section 1.3.2. Figure 4.11a shows a schematic of the spatial domain, and section 3.7.2 describes the problem in detail. Table 4.2 gives the material properties of each region for the training and test solutions, which are the same as the previous chapter. The spatial domain was discretised with a 160×160 mesh of discontinuous bilinear quadrilateral elements, and the S_{50} angular discretisation was employed to produce full order angular solutions. Both the spatial and angular resolutions were higher than in chapter 3, as the efficiency of the full order model was improved since gathering the previous set of data. Figure 4.11b depicts the scalar flux distribution of the S_{50} solution to the interpolation problem.

Problem	Material	Source ($cm^{-2}s^{-1}$)	Σ_a (cm^{-1})	Σ_s (cm^{-1})
1	Green	6.40	0.01	0.19
	White	0.00	0.00	0.00
	Blue	0.00	0.01	0.19
2	Green	6.40	0.01	0.21
	White	0.00	0.00	0.00
	Blue	0.00	0.01	0.21
3	Green	6.40	0.03	0.19
	White	0.00	0.00	0.00
	Blue	0.00	0.03	0.19
4	Green	6.40	0.03	0.21
	White	0.00	0.00	0.00
	Blue	0.00	0.03	0.21
Interpolation	Green	6.40	0.02	0.20
	White	0.00	0.00	0.00
	Blue	0.00	0.02	0.20
Extrapolation	Green	6.40	0.02	0.22
	White	0.00	0.00	0.00
	Blue	0.00	0.02	0.22

Table 4.2: Material properties for the Watanabe-Maynard problems in chapters 4 and 5.

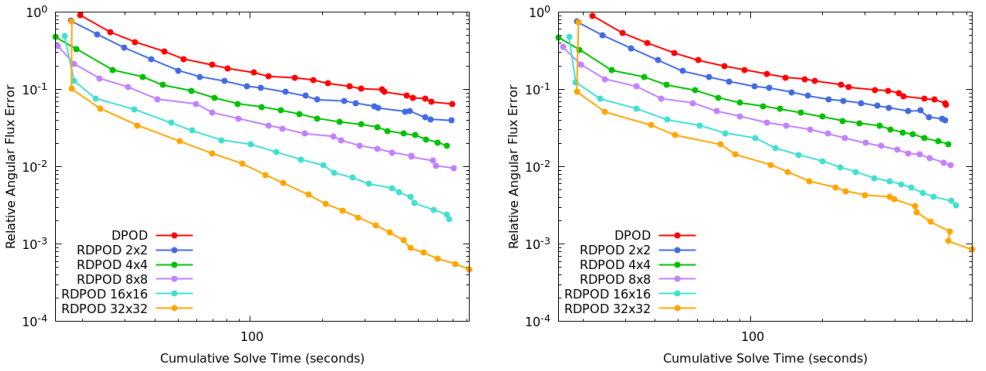
Table 4.2 lists the material cross sections for the training and test problems. As previously, the snapshot matrix was formed from all four training solutions, and the resulting bases were used to solve both an interpolation and an extrapolation problem.



(a) Interpolation (b) Extrapolation

Figure 4.12: Angular flux error vs number of basis functions for S_{50} solutions to the Watanabe-Maynard problems with varying numbers of spatial regions.

Figures 4.12a and 4.12b depict the L^2-norm of the angular flux error for the Watanabe-Maynard problems, for both DPOD and RDPOD with varying numbers of regions. As the figures show, increasing the number of RDPOD regions consistently reduced the angular flux error for a given basis size, excluding the results with just 4 basis functions per node. With 8 basis functions per node, the maximum reduction in error was an order of magnitude, which increased steadily to 2 orders of magnitude by 84 basis functions.

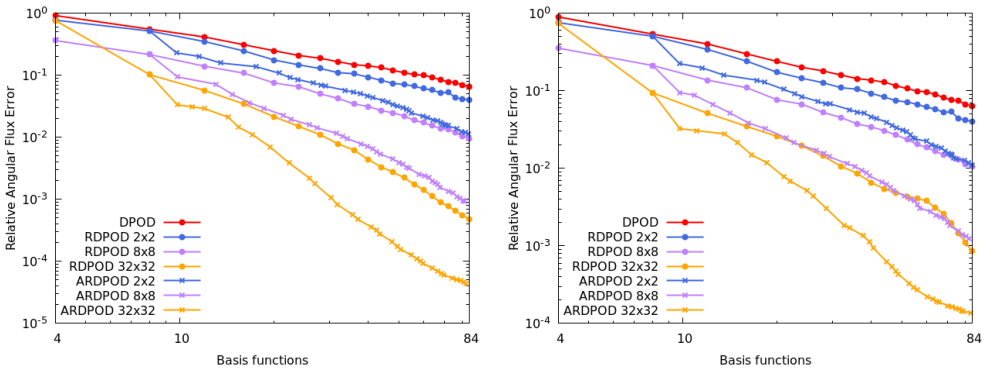


(a) Interpolation (b) Extrapolation

Figure 4.13: Angular flux error vs solve time in seconds for S_{50} solutions to the Watanabe-Maynard problems with varying numbers of spatial regions. "$a \times b$" indicates a spatial partitions in x and b in y, for a total of $a \times b$.

Figure 4.13 plots the relative angular flux error against the solve time for DPOD and RDPOD. It is again shown that increasing the number of spatial regions consistently decreased the error for a given solve time, with a maximum reduction of more than 2 orders of magnitude. In some cases, moving from 4 to 8 basis functions decreased both the error and the solve time, which was not expected. This is likely because the iterative solver converged slowly for the bases with 4 functions, so adding a basis function to each

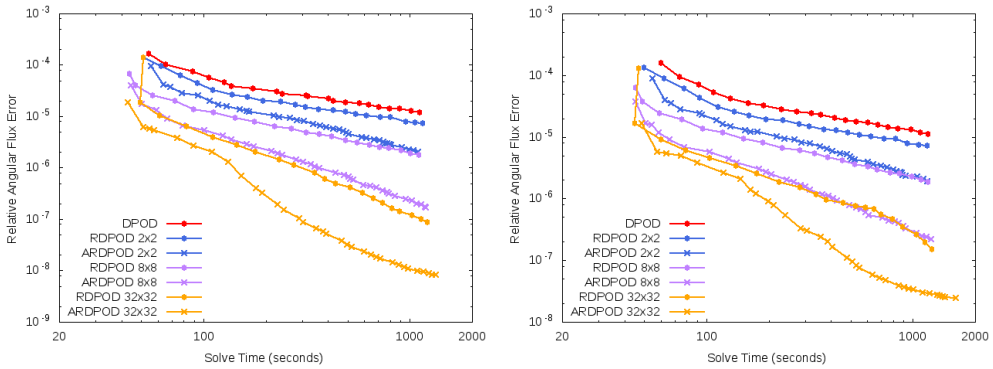
octant not only reduced the error as expected, but also allowed the solver to converge in fewer iterations and thereby reduced the solve time.



(a) Interpolation (b) Extrapolation

Figure 4.14: Angular flux error for adaptive S_{50} solutions to the Watanabe-Maynard problems with varying numbers of spatial regions. "$a \times b$" indicates a spatial partitions in x and b in y, for a total of $a \times b$.

Figure 4.14 plots the relative angular flux error against the mean number of basis functions per node for all three methods with varying numbers of regions. As the figure shows, ARDPOD reduced the error compared to RDPOD in all cases, with a peak reduction of approximately an order of magnitude. Compared to DPOD, ARDPOD reduced the error by approximately 2.5 orders of magnitude. The figure also demonstrates, once again, that increasing the number of regions consistently reduced error for both RDPOD and ARDPOD.

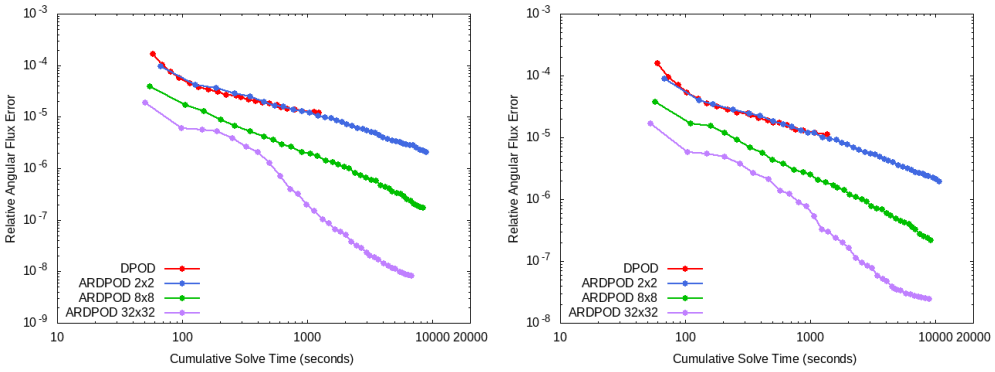


(a) Interpolation (b) Extrapolation

Figure 4.15: Angular flux error vs solve time in seconds for S_{50} solutions to the Watanabe-Maynard problems with varying numbers of spatial regions using angular adaptivity. "$a \times b$" indicates a spatial partitions in x and b in y, for a total of $a \times b$.

Figure 4.15 compares the relative angular flux error to the solve times for DPOD, RDPOD and ARDPOD. As explained in the discussion of figure 4.8, the solve times given are

for a single iteration, and are merely intended to be indicative rather than precise measurements of optimal performance. The figure shows that ARDPOD drastically decreased the error for a given solve time, with a peak reduction of more than 3 orders of magnitude. This demonstrates that, while RDPOD and ARDPOD both reduced the error compared to DPOD for a given basis size, they did not significantly affect the solve time. Similar anomalies to figure 4.13 were seen with four basis functions per node, likely for the same reason.

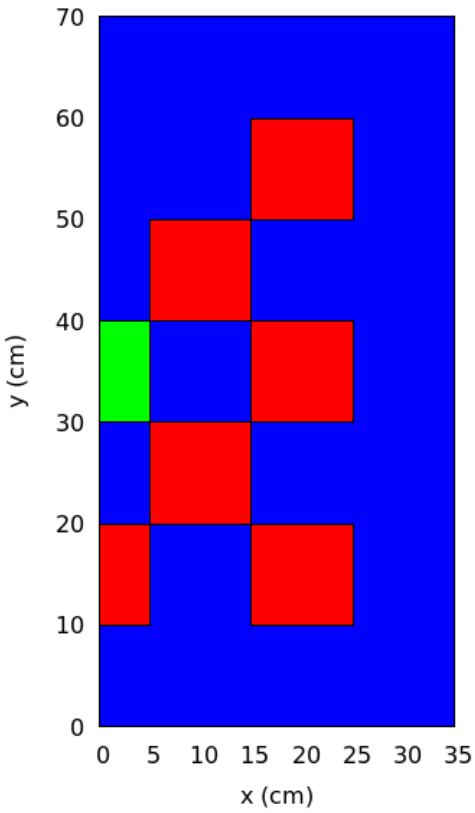


(a) Interpolation (b) Extrapolation

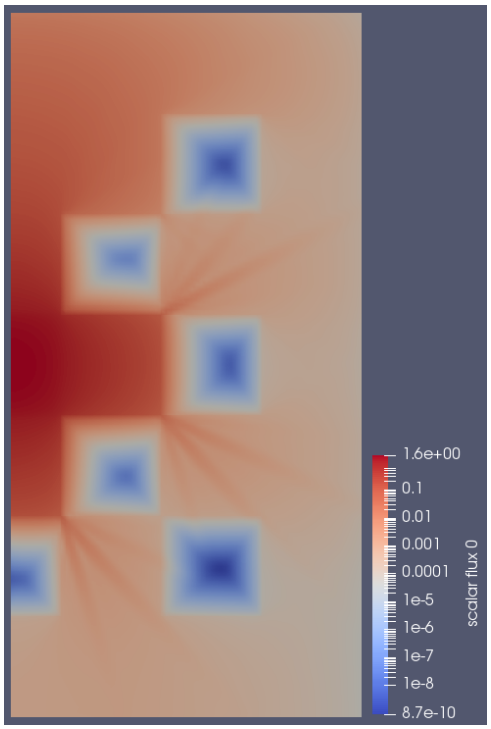
Figure 4.16: Angular flux error vs cumulative solve time in seconds for S_{50} solutions to the Watanabe-Maynard problems with varying numbers of spatial regions using angular adaptivity. "$a \times b$" indicates a spatial partitions in x and b in y, for a total of $a \times b$.

Figure 4.16 presents the relative angular flux error against the cumulative solve time for ARDPOD. As mentioned in the discussion of figure 4.9, the adaptive method is not yet fully optimised, and significant improvements by this metric are likely possible. Despite this, the graphs show that ARDPOD performed at least as well as DPOD in the case of 2x2 regions, and significantly better with more regions. At best, it offered over an order of magnitude reduction in error with the same solve time as DPOD.

4.4.4 The Checkerboard Problem



(a) Schematic



(b) S_{50} scalar flux solution

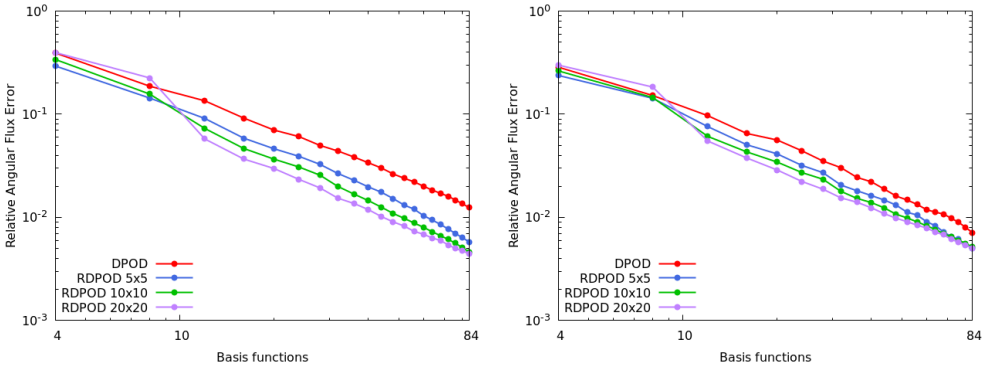
Figure 4.17: Schematic (a) and S_{50} scalar flux solution (b) for the checkerboard extrapolation problem. The green region is the source, the blue regions are scattering and the red regions are highly absorptive.

The fourth example in this chapter is a checkerboard problem first proposed by Brunner [14], which is discussed further in section 1.3.3. A schematic is shown in Figure 4.17a. The green region is an isotropic source, the blue regions are purely scattering, and the red regions are highly absorptive. The full problem presented in [14] is symmetrical about a vertical line through the centre, and so here the right half of the problem has been solved with reflective boundary conditions on the left boundary. All other boundaries have vacuum conditions applied. The domain is discretised using a 70×140 mesh of discontinuous bilinear quadrilateral elements in space. Full order solutions for snapshot generation and error comparison were produced using S_{50} in angle. Figure 4.17b shows the scalar flux distribution of the S_{50} solution to the extrapolation problem.

Problem	Region	Source ($cm^{-2}s^{-1}$)	Σ_a (cm^{-1})	Σ_s (cm^{-1})
1	Green	1.00	0.00	0.50
	Blue	0.00	0.00	0.50
	Red	0.00	5.00	0.00
2	Green	1.00	0.00	1.50
	Blue	0.00	0.00	1.50
	Red	0.00	5.00	0.00
3	Green	1.00	0.00	0.50
	Blue	0.00	0.00	0.50
	Red	0.00	15.00	0.00
4	Green	1.00	0.00	1.50
	Blue	0.00	0.00	1.50
	Red	0.00	15.00	0.00
Interpolate	Green	1.00	0.00	1.00
	Blue	0.00	0.00	1.00
	Red	0.00	10.00	0.00
Extrapolate	Green	1.00	0.00	2.00
	Blue	0.00	0.00	2.00
	Red	0.00	20.00	0.00

Table 4.3: Material properties for the checkerboard problems in chapters 4 and 5.

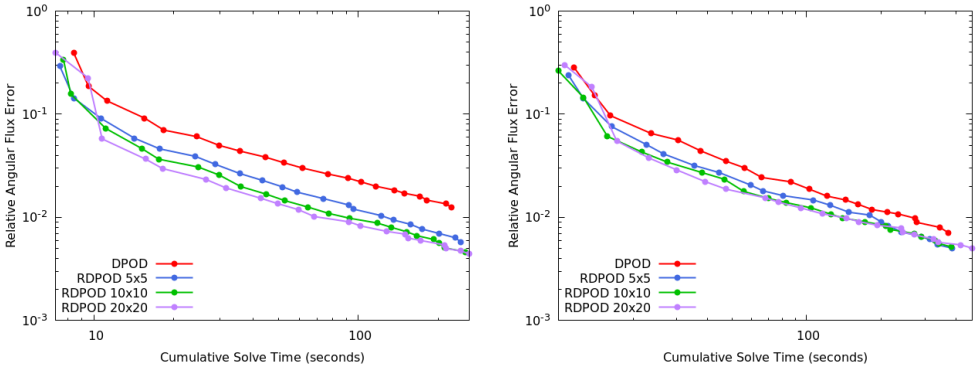
Table 4.3 lists the material cross sections for both the training and test problems. The snapshot matrix was formed from all four training solutions, and the resulting POD bases were used to solve the test problems. As reproducing a seen solution is relatively simple and has already been demonstrated, both of the test problems in this example are *unseen* problems - the *interpolation* problem has its material properties within the range for which snapshots were produced, while the material properties of the *extrapolation* problem lie outside this range.



(a) Interpolation (b) Extrapolation

Figure 4.18: Angular flux error vs number of basis functions for S_{50} solutions to the checkerboard problems with varying numbers of spatial regions. "$a \times b$" indicates a spatial partitions in x and b in y, for a total of $a \times b$.

Figure 4.18 compares the L^2-norms of the relative angular flux errors for the checkerboard interpolation and extrapolation problems using DPOD and RDPOD, with varying numbers of basis functions and regions. The results demonstrate that RDPOD generally has a lower error than DPOD for a given number of basis functions, and that increasing the number of RDPOD basis functions reduces the error in most cases, as expected. However, the reduction in error is relatively low compared to other problems, at less than half of an order of magnitude at best. This is somewhat unexpected, as it was assumed that the complex nature of the checkerboard problem would benefit significantly from RDPOD. One possible explanation is that the angular flux profiles involved are sufficiently complicated that both DPOD and RDPOD struggle to compress them effectively, whereas other problems were simple enough that RDPOD allowed for significant improvement over DPOD. There are also some data points with few basis functions which perform poorly compared to the general trend. This issue has been observed for other problems as well, and is likely due to the potential instability of solvers with a very small number of basis functions.



(a) Interpolation (b) Extrapolation

Figure 4.19: Angular flux error vs solve time in seconds for S_{50} solutions to the checkerboard problems with varying numbers of spatial regions. "$a \times b$" indicates a spatial partitions in x and b in y, for a total of $a \times b$.

Figure 4.19 plots the relative angular flux error against the solve time for DPOD and RDPOD. As previously, increasing the number of spatial regions tends to result in a lower error for a given solve time. This trend is not perfect, however - in some cases the RDPOD solutions all have comparable error, and the 20×20 RDPOD model performed poorly with few basis functions. As mentioned previously, the reduction in error is relatively low for this problem, but the RDPOD method still improves computational efficiency compared to DPOD in almost all cases, and matches it at worst.

4.5 Discussion

This chapter has developed upon the method of DPOD proposed in a recent article [15]. A new reduced order model for the angular dimension of the BTE, known as RDPOD, has been described. The novelty of RDPOD lies in its separation of the spatial domain into multiple regions, each of which has its own optimised DPOD basis set. A method of projecting flux between each reduced order basis without full order calculations has also been derived and implemented. Finally, an adaptive algorithm based on the RDPOD bases has been presented.

The RDPOD method is shown to consistently decrease the relative angular flux error for a given solve time and basis size when compared to DPOD, by an amount proportional to the number of spatial regions. This was the expected result, as increasing the number of regions reduces the number of elements per region, and therefore allows each basis function to be better optimised for the angular flux profiles it represents. The relative

angular flux error for a given number of basis functions was reduced by up to an order of magnitude for the dog-leg duct problem, and 2 orders of magnitude for the Watanabe-Maynard problem, demonstrating that the method can benefit both advection and scattering problems. Similar results were observed when comparing the error to the solve time, as RDPOD did not significantly affect the solve time for a given basis size compared to DPOD.

The adaptive method, known as ARDPOD, was demonstrated to further reduce error for a given basis size compared to RDPOD. The effect of adaptivity was significant - for both the dog-leg duct and Watanabe-Maynard problems, the error was reduced by up to 3 orders of magnitude compared to DPOD. Compared to RDPOD with the same number of regions, ARDPOD typically reduced the error by up to an order of magnitude, and more in some cases. The same was true when comparing the error to the solve time for each iteration. However, the cumulative time to reach a given adaptive stage was not optimised, as the algorithm instead aimed to form as close to an optimal basis set as possible at each adaptive stage. Additional optimisation of the adaptive method could significantly reduce the cumulative solve time, but this is left for future work.

The results for the region test problem demonstrate the purpose of including boundary elements in the snapshot matrices of each region. When no BEs are included, the minimum possible error is approximately 10^{-6} when every possible basis function is used. Even though the RDPOD model has seen the angular flux distribution of each element, it is not capable of reproducing the solution to machine error without boundary elements. It is suspected that some error is induced in the transfer of flux between elements, since the incoming flux uses a different basis and must be converted to the local one, which can not represent it perfectly. By contrast, when boundary elements are included, each region's basis is capable of representing incoming fluxes without inducing error, since they have seen the flux distributions of adjacent elements. While the effect of including BEs is much less significant for larger problems, it was judged that including them was a sensible precaution to ensure that the performance of the method was never constrained by the aforementioned induced error.

Chapter 5

Solution Acceleration Methods for the Boltzmann Transport Equation

This chapter will present two methods of solver acceleration which aim to reduce the computational cost of converging solutions to the BTE. Both methods utilise the DPOD and RDPOD methods presented in chapters 3 and 4, respectively.

The first method presented is a simple multigrid algorithm which aims to accelerate the solution of the ROMs. Since POD basis functions are hierarchical, low order basis functions generally contribute more to the overall solution than higher order basis functions. The multigrid method solves DPOD and RDPOD models on a coarse grid, then increases the number of basis functions in the model while retaining the solution coefficients of existing basis functions. This allows the low order basis functions to be resolved at a reduced computational cost, before introducing higher order basis functions which contribute relatively small corrections to the overall solution and therefore require fewer iterations to converge to a given level of accuracy.

The second method presented aims to accelerate the solution of full order models. The ROMs developed in previous chapters are used to produce an initial solution to a full order model, which is then iterated further to improve the accuracy of the solution. Since the reduced order models have significantly fewer degrees of freedom, they are able to converge to a given level of accuracy more quickly than the full order model. However, the process of dimensionality reduction inevitably introduces error due to

the approximations made, which reduces the maximum accuracy which the models can attain. By iterating ROM solutions in the full order model, this error can be eliminated while retaining some of the computational cost reductions provided by the ROMs.

The chapter is split into two sections, the first describing the multigrid method, and the second describing the use of ROMs to accelerate full order solutions. Each section is organised as follows: First, the motivation and theory behind the method is described. Numerical results for a range of problems are then presented to demonstrate its effectiveness. Finally, conclusions are drawn on the utility of the method in question, based on the numerical results.

5.1 Multigrid Reduced Order Modelling

This section will present a highly simplified implementation of a multigrid solution algorithm, as introduced in section 2.5.2. It is intended to improve the solve times of the reduced order models implemented in previous chapters.

5.1.1 Motivation and Theory

Multigrid methods use multiple discretisations of varying resolutions to accelerate the solution of a high-resolution model. For example, a 'V-cycle' begins by solving for a few iterations on a fine grid, then projecting the residual error onto coarser grids and solving to find corrections which minimise the residual, before projecting the corrections back up to the fine grid and iterating further. This minimises the approximation error by generating solutions on a fine grid, without incurring the full computational cost of converging such solutions. The most commonly used multigrid cycles are discussed in section 2.5.2.

In order to project between grids, prolongation and restriction operators are typically constructed. If a flux vector ψ_n is associated with resolution level n, where the resolution increases with increasing n, then the prolongation operator P_n is defined as,

$$\psi_{n+1} = P_n \psi_n, \tag{5.1}$$

and the restriction operator R_n is defined as,

$$\psi_{n-1} = R_n \psi_n. \tag{5.2}$$

In this case, the multigrid method is applied to the angular dimension of the reduced order model, meaning that the problem is solved with varying numbers of angular basis functions. The standard multigrid method described in section 2.5.2 was modified due to the properties of the existing models. Since the models constructed for this project utilise a sweep-based solver, the residual error is not available and would need to be constructed. Instead, the full solution vector is projected between grids and solved for at each stage. Therefore, only prolongation was used in this case, and not restriction. The result of these modifications was a model which begins by solving on a coarse angular grid, then prolongates the entire solution and solves it with additional basis functions. This process may be repeated several times before reaching and solving on the final grid, which has the highest resolution. A plot of the grid resolution at each stage of this simplified 'up-cycle' is shown in figure 5.1.

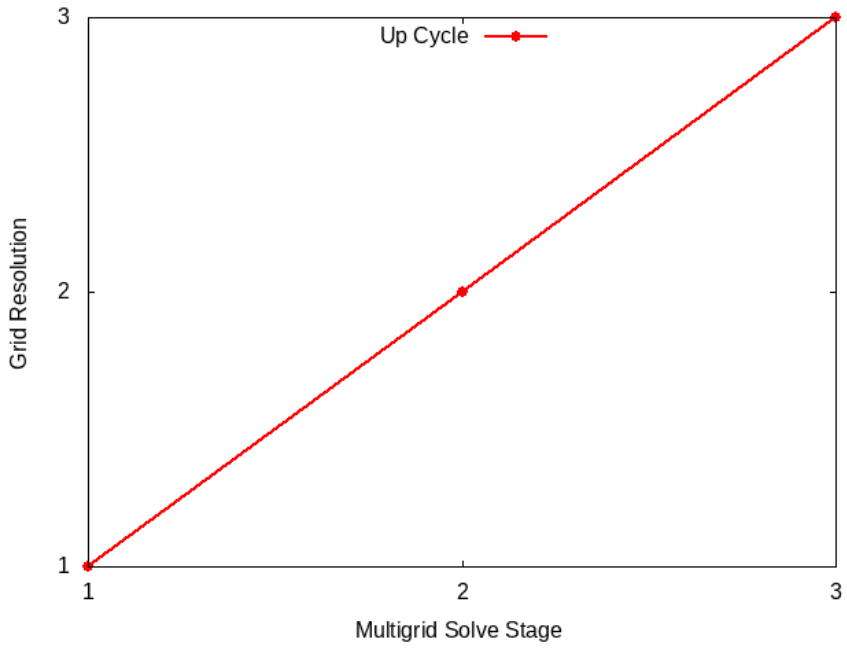


Figure 5.1: An example of the simplified 'up-cycle' used in this project.

For example, consider a two-stage multigrid method with n_0 basis functions in the first stage and n_1 in the second, where $n_1 > n_0$. The solution to the reduced order model

on a single node at stage n is given by the vector ψ_n. Before the model is run, ψ_0 is of length n_0, and is initialised with every coefficient set to zero:

$$\psi_0 = \begin{bmatrix} \psi_{0,1} \\ \vdots \\ \psi_{0,n_0} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{5.3}$$

After the first stage solution, the coefficients in ψ_0 are nonzero in general. The second stage begins by creating a ψ_1 vector of length n_1, and transferring the coefficients from the previous stage using the prolongation operator P_0. This will result in a vector with n_0 nonzero coefficients, and $n_1 - n_0$ coefficients of zero, which correspond to the newly added basis functions:

$$\psi_1 = P_0\psi_0 = \begin{bmatrix} \psi_{0,1} \\ \vdots \\ \psi_{0,n_0} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \implies P_0 = \begin{bmatrix} I \\ 0 \end{bmatrix}. \tag{5.4}$$

P_0 is therefore a rectangular matrix with dimensions $n_1 \times n_0$, containing the $n_0 \times n_0$ identity matrix I in a block at the top, and zeros elsewhere. It is not necessary to actually construct the prolongation matrices in this case, since merely transferring coefficients from ψ_0 to the top of ψ_1 has the same effect.

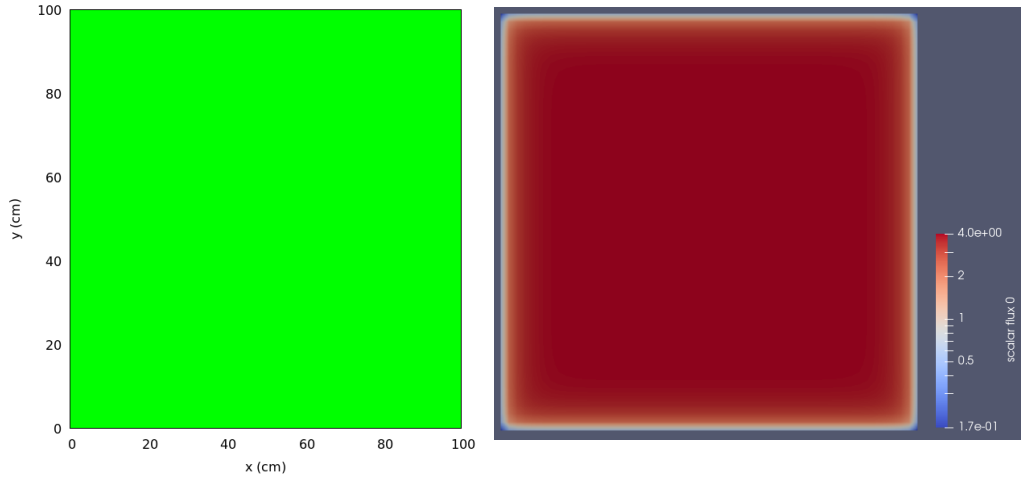
Solving for ψ_1 will converge the newly added coefficients, and slightly adjust the old ones due to contributions from the new basis functions. The second solution will have equivalent accuracy to simply solving with n_1 basis functions from the start, but requires fewer iterations of the high-order model to converge to a solution, because the early basis functions which contain most of the data are already close to converged when the model is initialised, and the new basis functions represent relatively small corrections which converge quickly.

5.1.2 Numerical Examples

This section presents several numerical examples which demonstrate the ability of the multigrid method to rapidly produce solutions to the BTE. The majority of the data presented uses two multigrid stages, a coarse grid followed by a fine grid. This is primarily because each multigrid stage can have its own resolution and tolerance, and the more stages used, the larger the number of possible combinations. With two stages and a final stage with 21 basis functions per node, the only possibilities are 1>21, 2>21 and so on, for a total of 20. With three stages, a first stage with n basis functions has $20 - n$ possible second stages - 1>2>21, 1>3>21, ..., 1>20>21, 2>3>21, 2>4>21 and so on, for a total of 190. The threshold value at each stage besides the last can also vary, further increasing the number of configurations. Since the search space of the two-stage method was relatively small, comprehensive data collection was possible. To gather this data, each problem was first solved with 4-80 basis functions per node and varying convergence thresholds. These solutions were then used as initial solutions for a solve with 84 basis functions per node and a threshold of 10^{-9}.

Some data gathered using more than two multigrid stages is shown for the first problem. However, this consists of a small subset of possible configurations due to the large number of configurations available, as explained above. The data which was gathered using more than two multigrid stages found minor benefits over a two stage process at best, and did not seem to justify more extensive data collection.

5.1.2.1 The Scattering Test Problem

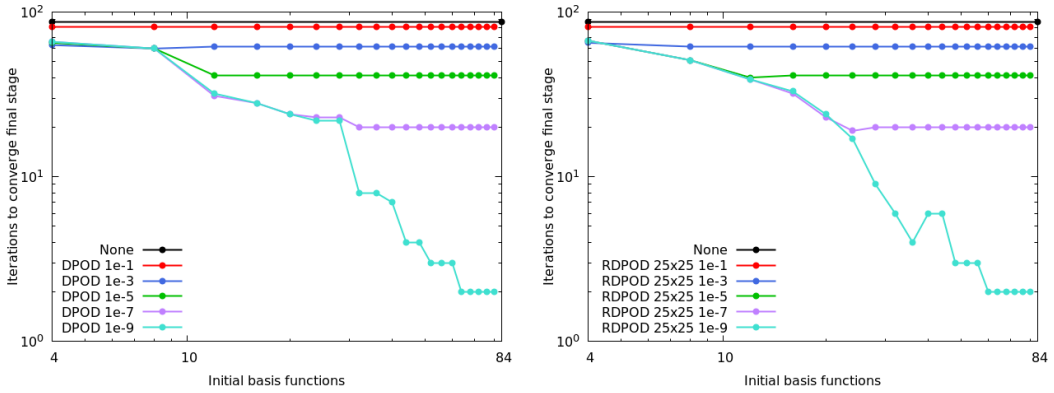


(a) Schematic

(b) S_{50} scalar flux solution to extrapolation problem

Figure 5.2: Schematic (a) and S_{50} scalar flux solution (b) for the scattering test extrapolation problem.

The first example is a highly scattering problem, which was designed to demonstrate the utility of the methods in this chapter. Problems shown in previous chapters have required relatively few iterations to converge the full order model, typically in the tens and as low as two for purely streaming problems such as the dog-leg duct. However, many problems which are of interest do not converge so quickly, and it is these problems which are likely to benefit most from this method of acceleration. The problem consists of a single material which is highly scattering, weakly absorbing and an isotropic neutron source. The material properties for each variation of the problem are listed in table 5.1. Vacuum boundary conditions are applied to all four boundaries. The problem was discretised with a 50x50 mesh of discontinuous bilinear quadrilateral elements, and the S_{50} angular discretisation was used to generate snapshots for the ROMs. As in prior chapters, the first test problem is referred to as the *interpolation* problem, as its material properties are within the range for which snapshots were produced, and the second is referred to as the *extrapolation* problem, as its material properties lie outside of this range.



(a) Interpolation problem, DPOD solver.

(b) Interpolation problem, RDPOD 25x25 solver.

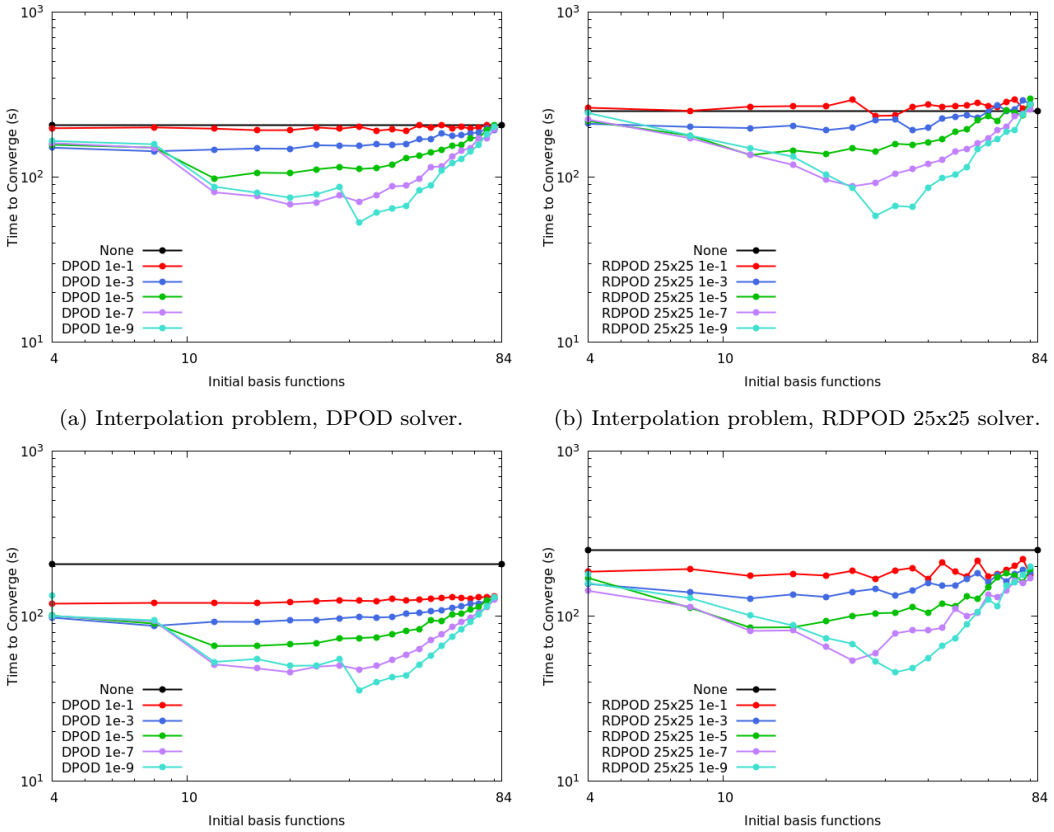
(c) Extrapolation problem, DPOD solver.

(d) Extrapolation problem, RDPOD 25x25 solver.

Figure 5.3: The number of iterations required to converge the final stage solution with 84 reduced order basis functions per node, for varying solver tolerances and basis functions.

Figure 5.3 shows the number of iterations required to converge a solution of the scattering test problem with 84 basis functions per node to a threshold of 10^{-9}, after solving with fewer basis functions and retaining that solution. The number of iterations required when solving for 84 basis functions alone without the multigrid method is also shown for comparison. As the figure shows, the multigrid method consistently reduced the number of iterations required in the final stage. For each first stage threshold value, increasing the number of basis functions used tended to decrease the number of iterations required up until a minimum. Once the minimum number of iterations for a particular threshold value was reached, adding more basis functions to the first solve stage made no difference to the number of iterations in the final stage. It appears that the effectiveness of the multigrid method in reducing the number of iterations in the final stage is limited by both the threshold value and the number of basis functions used in the first stage. This is to be expected, since the first stage solution asymptotically approaches the best possible solution for the number of basis functions it is using. If the number of basis functions is

too low, then error is caused by an inability to accurately represent the solution, and if the threshold is too low, insufficient iterations will be performed and the solution will not converge fully. In both cases, the resulting error forces the final stage to perform more iterations in order to converge to the full order solution.



(a) Interpolation problem, DPOD solver.

(b) Interpolation problem, RDPOD 25x25 solver.

(c) Extrapolation problem, DPOD solver.

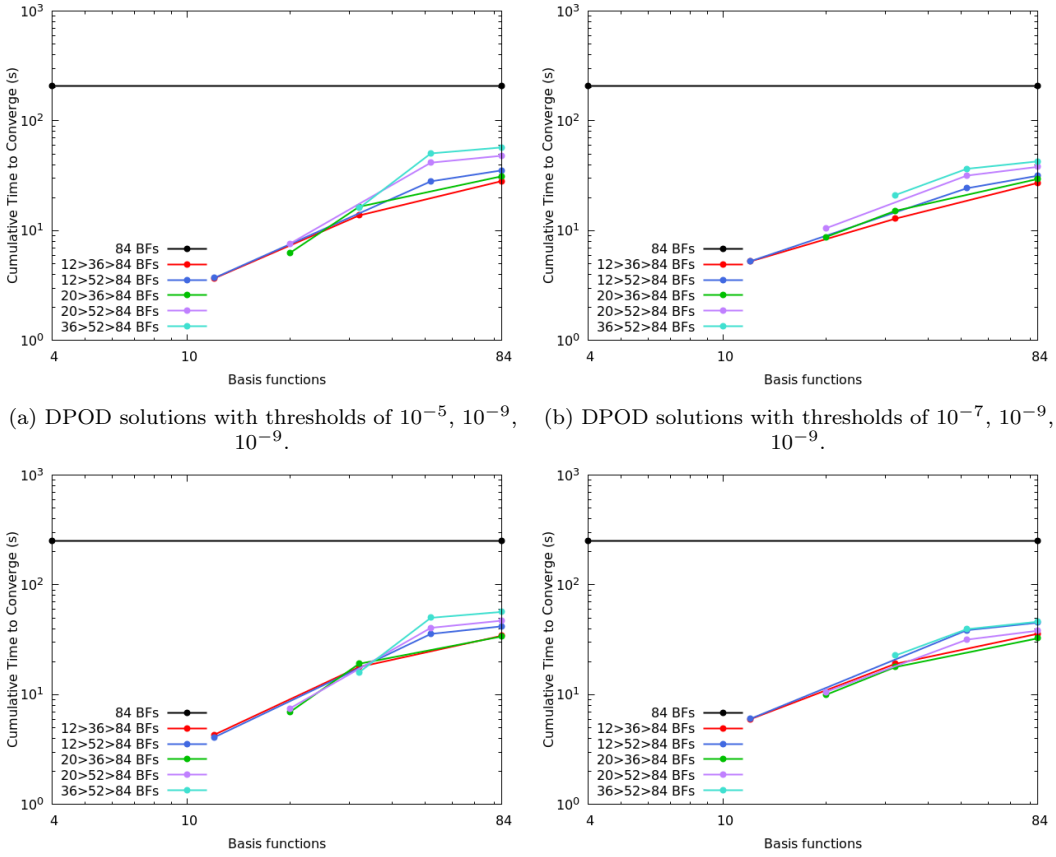
(d) Extrapolation problem, RDPOD 25x25 solver.

Figure 5.4: The total computation time required to converge a solution with 84 reduced order basis functions per node, for varying solver tolerances and basis functions.

Figure 5.4 shows the total computation time required to converge solutions to the scattering test problem with a threshold of 10^{-9}, as the threshold and number of basis functions used in stage 1 were varied. This is given by the sum of computation times for each multigrid stage. Increasing the number of basis functions used in the first stage decreased the total time to converge up to a point, then began to increase it again and decrease the overall efficiency of the method. At best, the convergence time was reduced by approximately 0.8 orders of magnitude.

Increasing the threshold tended to make the gradient steeper in both sections, with the solve time reducing faster initially and thus reaching a lower minimum, but also increasing faster once the minimum was reached. This can be understood by examining

figure 5.3 - while adding more basis functions in the first stage reduced the number of iterations in the final stage, it tended to be beneficial. Once the final stage iterations were at or close to their minimum, adding more basis functions increased the overall solve time. This is because the additional solve time required in the first stage was more than counteracted by the reduced solve time of the second stage if fewer iterations were required to solve it. At a certain point, however, adding more basis functions failed to decrease the number of iterations in the final stage, but continued to increase the solve time of the first stage, thus increasing the total solve time. These results suggest that the multigrid method works as expected, successfully reducing the overall solve time. They also demonstrate that there is an optimal setup in terms of threshold and basis function values for each combination of problem, solver and desired error. However, finding the optimal setup is not required in this case, as the method is beneficial for a wide range of settings. Both DPOD and RDPOD exhibited similar levels of reduction in solve times, though the final RDPOD solves resulted in much less error than the DPOD solves with equal numbers of basis functions, as previously.



(a) DPOD solutions with thresholds of 10^{-5}, 10^{-9}, 10^{-9}.

(b) DPOD solutions with thresholds of 10^{-7}, 10^{-9}, 10^{-9}.

(c) RDPOD 25x25 solutions with thresholds of 10^{-5}, 10^{-9}, 10^{-9}.

(d) RDPOD 25x25 solutions with thresholds of 10^{-7}, 10^{-9}, 10^{-9}.

Figure 5.5: The cumulative amount of time required to converge each stage of the extrapolation problem with three multigrid stages with varying angular resolutions and threshold values.

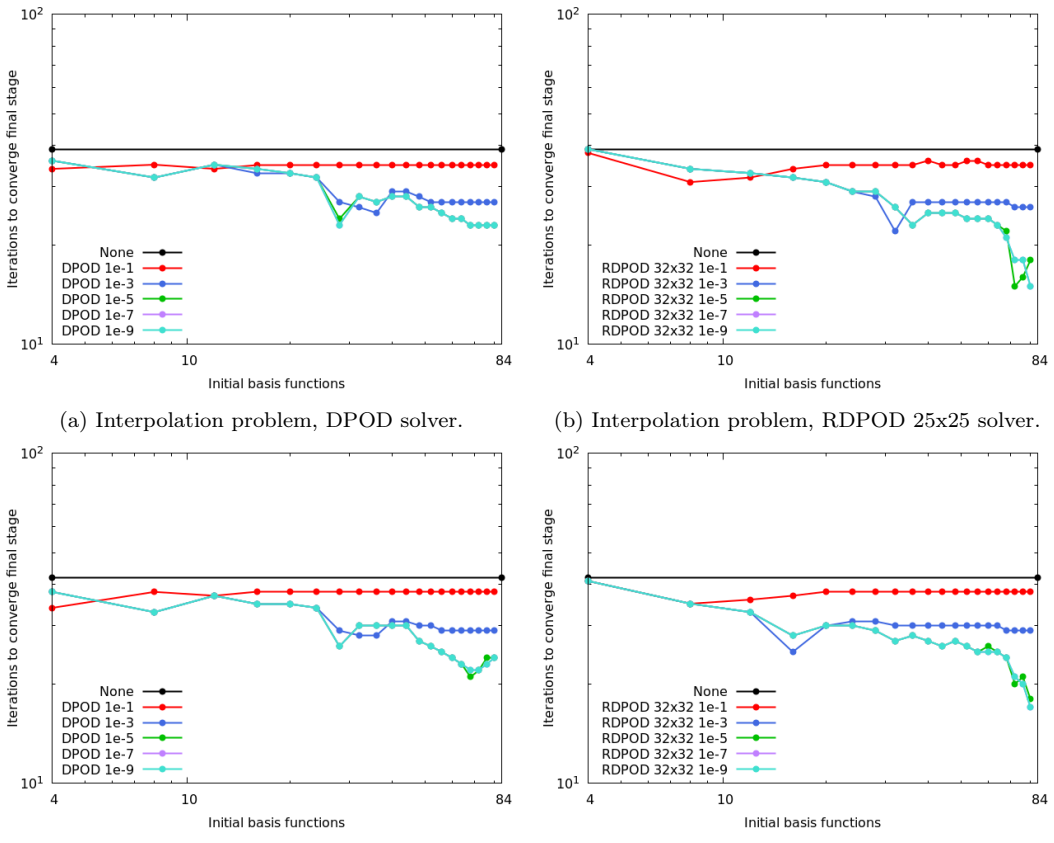
Figure 5.5 shows the cumulative time required to solve each step of the scattering test extrapolation problem using various 3-stage multigrid configurations, all of which had a final stage with 84 basis functions and a threshold of 10^{-9}. The times taken to solve the final stage alone, without using the multigrid method, are also shown for comparison. As explained in the introduction to section 5.1.2, each additional multigrid stage significantly increases the number of combinations of basis function numbers and threshold values. Due to the prohibitively large number of possible configurations, only a small subset were tested. The configurations were selected by examining figure 5.3 for inflection points where the derivative of the gradient changed from negative to positive, since these represent particularly efficient combinations of basis function count and threshold value. For example, there was a steep decline in iterations between 32 and 36 basis functions with a threshold of 10^{-9} in both DPOD plots, followed by a gradient of 0 between 36 and 40 basis functions. DPOD was therefore particularly efficient with 36 basis functions,

as confirmed by figure 5.4. As such, several of the three-stage configurations shown in figure 5.5 use 36 basis functions at one stage.

As the figures show, the selected configurations consistently converged faster than the non-multigrid solutions, by up to an order of magnitude. In some cases, they also improved upon the best two-stage multigrid results shown in figure 5.4, thus demonstrating the utility of using more than two multigrid stages. However, the improvements over the two-stage solves were minor even when present. More optimal configurations likely exist, but the increased search space compared to the two-stage process makes finding them difficult.

5.1.2.2 The Watanabe-Maynard Problem

The second example used to test the multigrid method is the Watanabe-Maynard problem [13], which is described in section 4.4.3. The problem setup in this case was identical to that presented in chapter 4, besides the use of the multigrid method.



(a) Interpolation problem, DPOD solver.

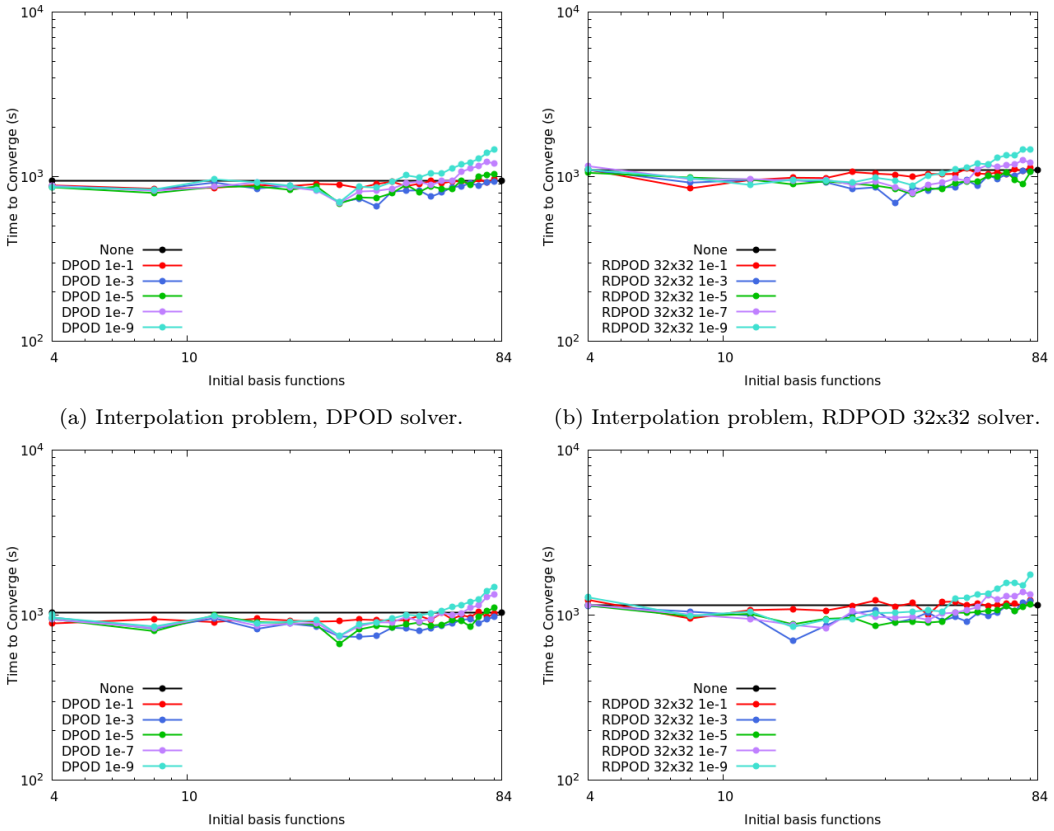
(b) Interpolation problem, RDPOD 25x25 solver.

(c) Extrapolation problem, DPOD solver.

(d) Extrapolation problem, RDPOD 25x25 solver.

Figure 5.6: The number of iterations required to converge the final stage solution with 84 reduced order basis functions per node, for varying solver tolerances and basis functions.

Figure 5.6 shows the number of iterations required to converge a solution of the Watanabe-Maynard problem with 84 basis functions per node to a threshold of 10^{-9}, after first solving with fewer basis functions. The number of iterations required when solving for 84 basis functions alone without the multigrid method is also shown for comparison. As previously, the multigrid method consistently reduced the number of iterations required in the final stage, and additional basis functions tended to reduce the number of iterations required further. However, in this case the reduction was only by 0.5 orders of magnitude at best. When the threshold was 10^{-5} or more, the number of iterations reached a minimum, but with a smaller threshold it continued to decrease throughout the graph, with minor oscillations. This is likely a result of the increased complexity of the Watanabe-Maynard problem compared to the scattering test problem, which meant that more basis functions were required to accurately represent its solutions. Adding basis functions therefore continued to be beneficial for longer, unless the first stage accuracy was limited by the solver threshold rather than the number of basis functions used. These results suggest that the optimal threshold and basis function count for reducing the number of iterations required to converge the final stage are problem-dependent, and increase with the complexity of the problem.



(a) Interpolation problem, DPOD solver.

(b) Interpolation problem, RDPOD 32x32 solver.

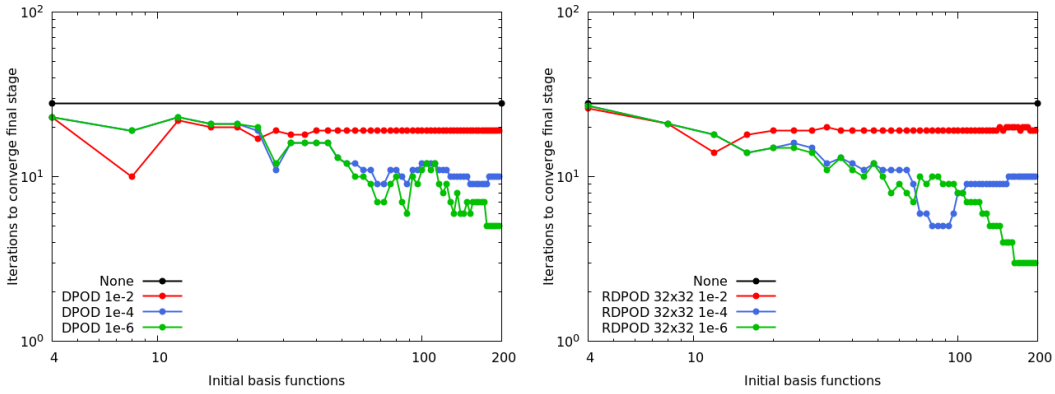
(c) Extrapolation problem, DPOD solver.

(d) Extrapolation problem, RDPOD 32x32 solver.

Figure 5.7: The total computation time required to converge a solution with 84 reduced order basis functions per node, for varying solver tolerances and basis functions.

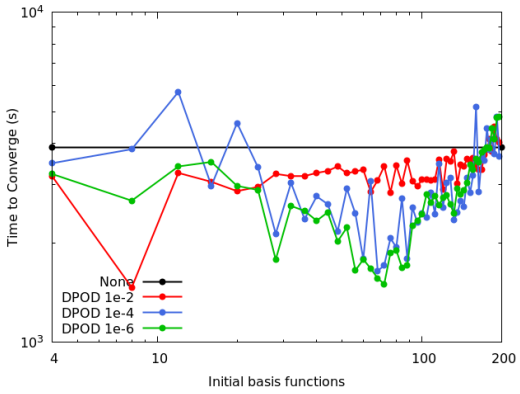
Figure 5.7 shows the total computation time required to converge the Watanabe-Maynard problem, including both stages of the multigrid method, as the threshold and number of basis functions used in the first stage were varied. While the multigrid method did tend to reduce the total solve time in most cases, the reduction was small, reaching approximately 0.2 orders of magnitude at best. This is likely because the increased complexity of this problem compared to the scattering problem meant that more basis functions had significant contributions to the solution. Therefore, the final stage was different enough from previous stages that a significant amount of iteration was required to converge the solution, regardless of the resolution and threshold used in the first stage. This implies that the method would be more beneficial if the final stage used significantly more basis functions, since a higher proportion of them would have small contributions and therefore the first stage solutions would be closer to the final stage solution. However, in this case the method was probably not worthwhile when considering the additional complexity introduced by its implementation, and the experimentation required to find

close-to-optimal settings.

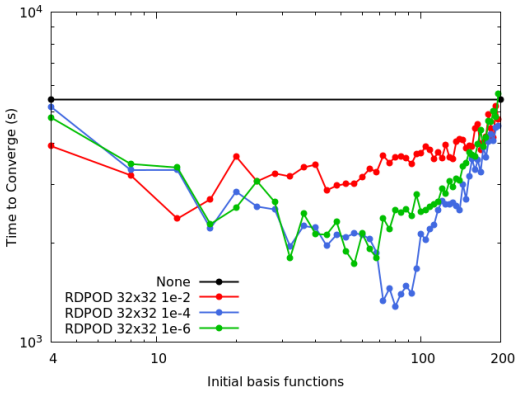


(a) DPOD solver iterations.

(b) RDPOD 32x32 solver iterations.



(c) DPOD total solve time.



(d) RDPOD 32x32 total solve time.

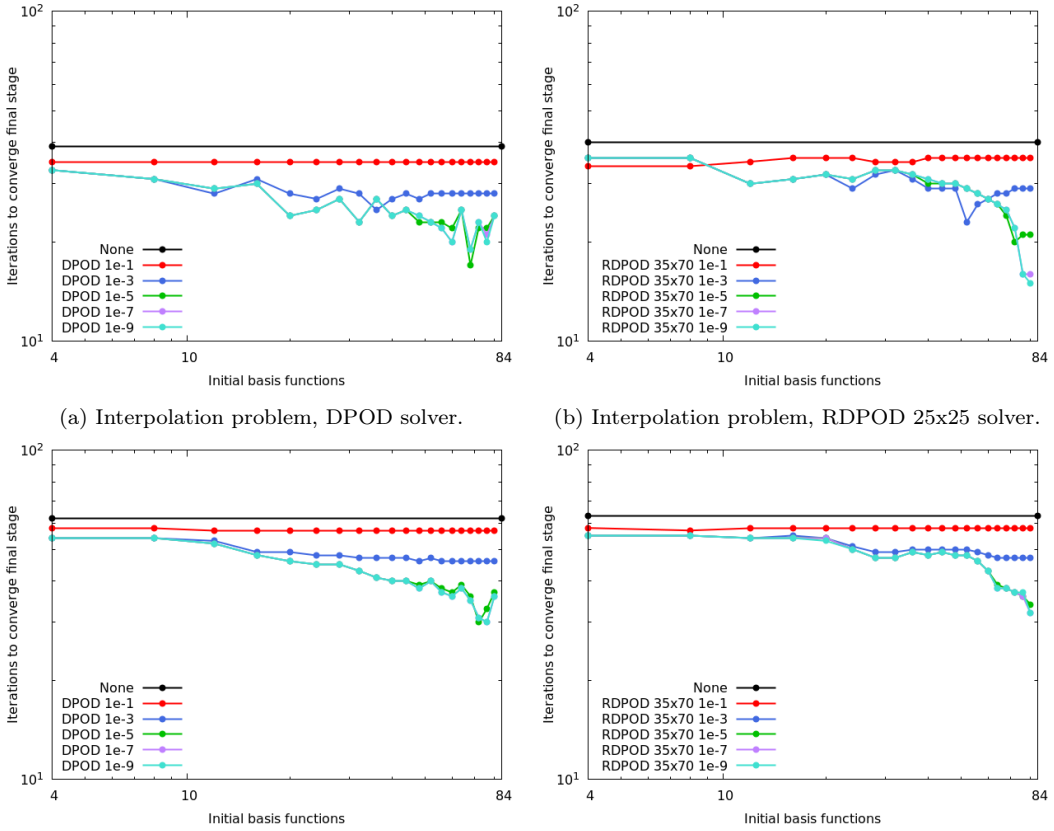
Figure 5.8: The number of final stage iterations and total time required to converge to a solution of the Watanabe-Maynard extrapolation problem with 200 basis functions per node, for varying solver tolerances and basis functions.

Figure 5.8 shows the number of iterations and total time required to converge the Watanabe-Maynard extrapolation problem, with 200 basis functions per node in the final stage. This investigation was intended to test the hypothesis that more complex problems would still benefit significantly from the multigrid method if a larger basis was used, as mentioned in the discussion of figure 5.7. Since the number of basis functions used was much larger, the convergence threshold was lowered to 10^{-6} so that the solve times would not be impractically large, and the interpolation problem was not examined. As expected, the multigrid method was significantly more beneficial in this case. The number of iterations in the final stage decreased by up to an order of magnitude, which was a significant improvement upon the 0.5 orders of magnitude improvement seen in figure 5.7. The reduction in total solve time was also much larger, reaching over 0.5 orders of magnitude at best compared to approximately 0.2 in the previous results. These results demonstrate that the multigrid method can have significant benefits even

for complex problems, and provide evidence that the reduction in solve time increases with the final stage resolution, as hypothesised.

5.1.2.3 The Checkerboard Problem

The third example used to test the multigrid method is the checkerboard problem [14], first described in section 4.4.4. The problem was set up as presented previously, besides the use of the multigrid method.



(a) Interpolation problem, DPOD solver.

(b) Interpolation problem, RDPOD 25x25 solver.

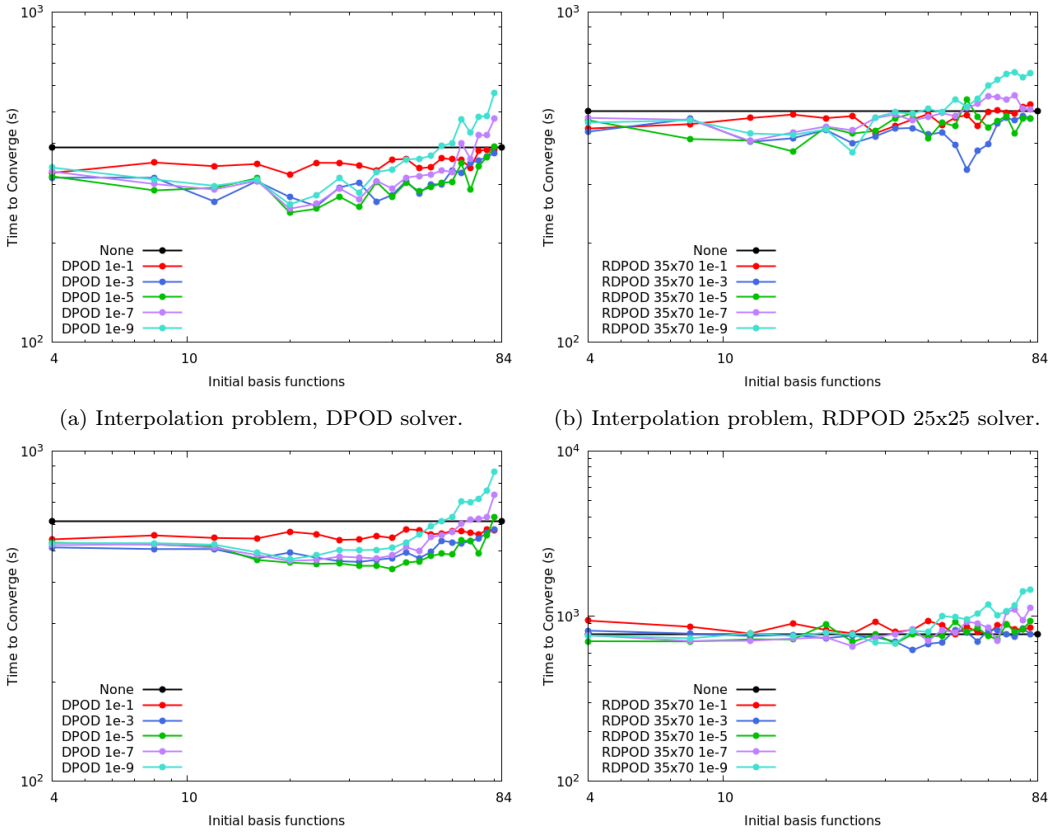
(c) Extrapolation problem, DPOD solver.

(d) Extrapolation problem, RDPOD 25x25 solver.

Figure 5.9: The number of iterations required to converge the final stage solution with 84 reduced order basis functions per node, for varying solver tolerances and basis functions.

Figure 5.9 shows the number of iterations required to converge a solution of the checkerboard problem with 84 basis functions per node to a threshold of 10^{-9}, after first solving with fewer basis functions. The number of iterations required when solving for 84 basis functions without the multigrid method is also shown for comparison. The results show many similarities with the Watanabe-Maynard results in figure 5.6, with the same implications. Once again, the multigrid method consistently reduced the number of iterations required to converge the final stage, and both the basis function count

and threshold value affected the extent of this reduction. However, the reduction was by approximately 0.5 orders of magnitude at best, and generally less. This suggests that, much like the Watanabe-Maynard problem, the checkerboard problem was too complicated to obtain the same benefits as the scattering test problem.



(a) Interpolation problem, DPOD solver.

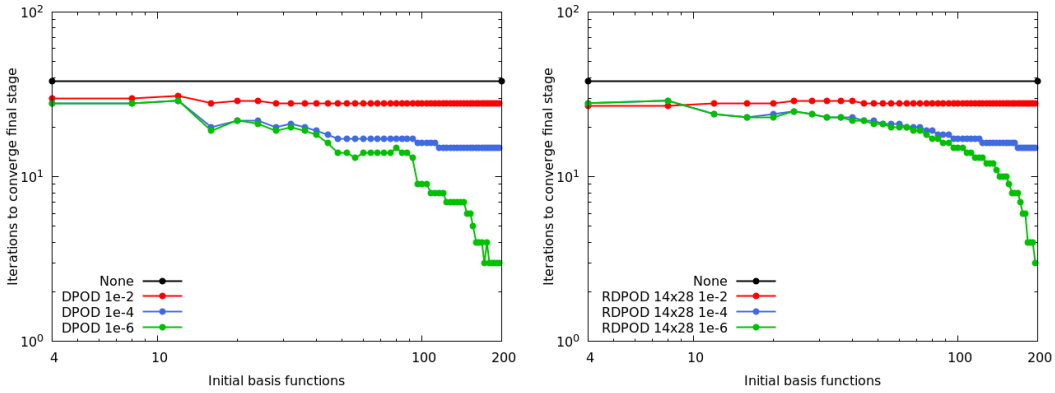
(b) Interpolation problem, RDPOD 25x25 solver.

(c) Extrapolation problem, DPOD solver.

(d) Extrapolation problem, RDPOD 25x25 solver.

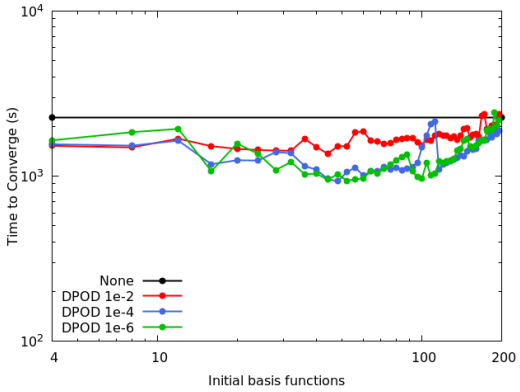
Figure 5.10: The total computation time required to converge a solution with 84 reduced order basis functions per node, for varying solver tolerances and basis functions.

Figure 5.10 shows the total computation time required to converge the checkerboard problem, including both stages of the multigrid method, as the threshold and number of basis functions used in stage 1 were varied. As in figure 5.7, the multigrid method successfully reduced the total time required to converge to a solution, but only by a relatively small amount, and not in every case. Since this problem is even more complicated than the Watanabe-Maynard problem, these results seem to confirm that the multigrid method is less effective the more complex the problem is. As explained in the discussion of figure 5.7, it is hypothesised that using more basis functions in the final stage would make the multigrid method more effective, but in this case the method does not seem worthwhile.

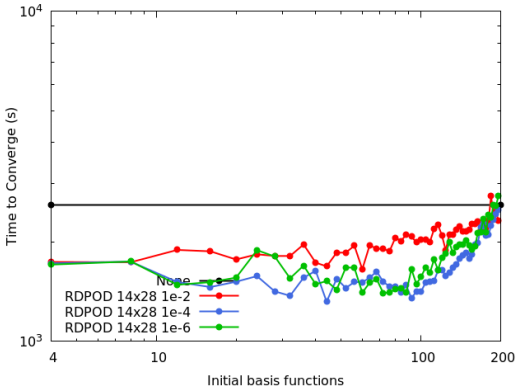


(a) DPOD solver iterations.

(b) RDPOD 32x32 solver iterations.



(c) DPOD total solve time.



(d) RDPOD 32x32 total solve time.

Figure 5.11: The number of final stage iterations and total time required to converge to a solution of the checkerboard extrapolation problem with 200 basis functions per node, for varying solver tolerances and basis functions.

Figure 5.11 shows the number of iterations and total time required to converge the checkerboard extrapolation problem, with 200 basis functions per node in the final stage.. This investigation was intended to provide further evidence for the hypothesis that more complex problems could still benefit significantly from the multigrid method if a larger basis was used, as explained previously. As with figure 5.8, the convergence threshold was lowered to 10^{-6} so that the solve times would not be impractically large, and the interpolation problem was not examined. As predicted, the multigrid method was significantly more beneficial in this case. Once again, the number of iterations in the final stage decreased by up to an order of magnitude, significantly improving upon the 0.5 order of magnitude reduction seen in figure 5.9. While not particularly large, the reduction in total solve time was more significant than in figure 5.10, reaching approximately 0.4 orders of magnitude at best compared to approximately 0.2 in the previous figure. These results provide further evidence that the multigrid method can benefit even complex problems, and that the reduction in solve time increases with the

final stage resolution, as hypothesised.

5.1.3 Discussion

This section has demonstrated that the multigrid method is able to decrease the total computation time required to generate a converged solution in most cases. However, with 84 basis functions in the final stage the effect was only significant enough to be worthwhile in the simplest case, the highly scattering test probem. The two more complex problems still benefited, but only by a small amount. As mentioned previously, this is likely because the simplicity of the scattering problem meant that higher order basis functions had relatively small contributions and thus the solutions required few iterations to converge when provided with approximate coefficients for the low order basis functions. By contrast, additional basis functions provided significant contributions to the solutions of the more complex problems. As a result, their final stages required more iterations for a given first stage configuration than the scattering problem, and therefore their solve times were reduced by less. This hypothesis led to speculation that solving with more basis functions in the final stage would likely increase the benefits of the multigrid method for the Watanabe-Maynard and checkerboard problems, since a higher proportion of the basis functions would have small contributions and converge quickly. In order to test this, both the Watanabe-Maynard and checkerboard problems were solved with 200 basis functions in the final stage. The expectation was that the multigrid method would be more beneficial in this context, since the high order basis functions would contribute less to the overall solution in a similar manner to the scattering test problem. In both cases, the multigrid method performed better with a larger number of basis functions, as predicted. This provides strong evidence for the hypothesis presented, and suggests that the multigrid method could be practical for large, complex problems despite its poor performance in initial testing.

Overall, these results seem to suggest that the multigrid method has potential for generating high-accuracy solutions in a shorter time than the standard method. However, the more complex the problem, the higher the model resolution required for the multigrid method to be of significant benefit. Optimisation of the solver and settings could likely mitigate this issue and make the method beneficial in a wider range of circumstances. It may also be worthwhile to experiment with more than two or three multigrid stages.

While this was attempted during testing with little benefit, the search space of possible configurations increases significantly with each additional stage, as explained in the introduction to section 5.1.2. As a result, it is almost certain that the optimal multi-stage configurations were not found. Further research into this technique should therefore consider a more rigorous method of determining which configurations to investigate.

5.2 Reduced Order Acceleration of the Full Order Model

5.2.1 Motivation and Theory

This section will present an application of the DPOD and RDPOD methods described in chapters 3 and 4, whereby reduced order models are used to accelerate the solution of full order models, as mentioned in section 2.2.1. Reduced order solutions to unseen problems are generated, and the angular flux vectors produced are used as the starting point for full order model solutions. The advantage of this method is that it generates a solution of equal accuracy to the full order model alone, while incurring significantly reduced computational costs due to the efficiency provided by the reduced order model.

The process begins by generating full order model solutions with a representative range of material properties, such that any unseen material properties which are to be applied will produce qualitatively similar solutions. DPOD and RDPOD models are then generated using these solutions as described in chapters 3 and 4. When a new solution for unseen material properties is required, the new problem is first solved using a reduced order model, generating a solution which consists of a vector $\alpha_{q,r}$ for each spatial node, containing the coefficients of the POD basis functions at that node. The coefficient vectors are projected back up to the full order model space using the basis matrices $\mathcal{U}_{q,r}$ associated with their corresponding node, as implied by equations 3.20 and 4.12 for DPOD and RDPOD respectively. This generates a reduced order approximation to the solution in the full order model's angular representation, S_N in this case. These S_N solution vectors ψ_q are then fed into a full order S_N angular model as the initial solution. The full order model is solved using an iterative upwind sweep solver, as described in sections 1.2 and 2.5.1.

As the initial solution is already an approximation to the full order solution, with its

accuracy dependent on the number and distribution of basis functions used in the ROM, the full order model converges in fewer iterations than if the solution vector had started from zero, as is usual. Since initial solutions are generated with reduced order models, they are produced much more quickly than the full order model would take to reach a similar level of accuracy. This ensures that the process has a reduced computational cost for the same level of accuracy, for any sensibly constructed reduced order model - that is, any which converges in less computational time than the full order model would take to reach the same level of accuracy.

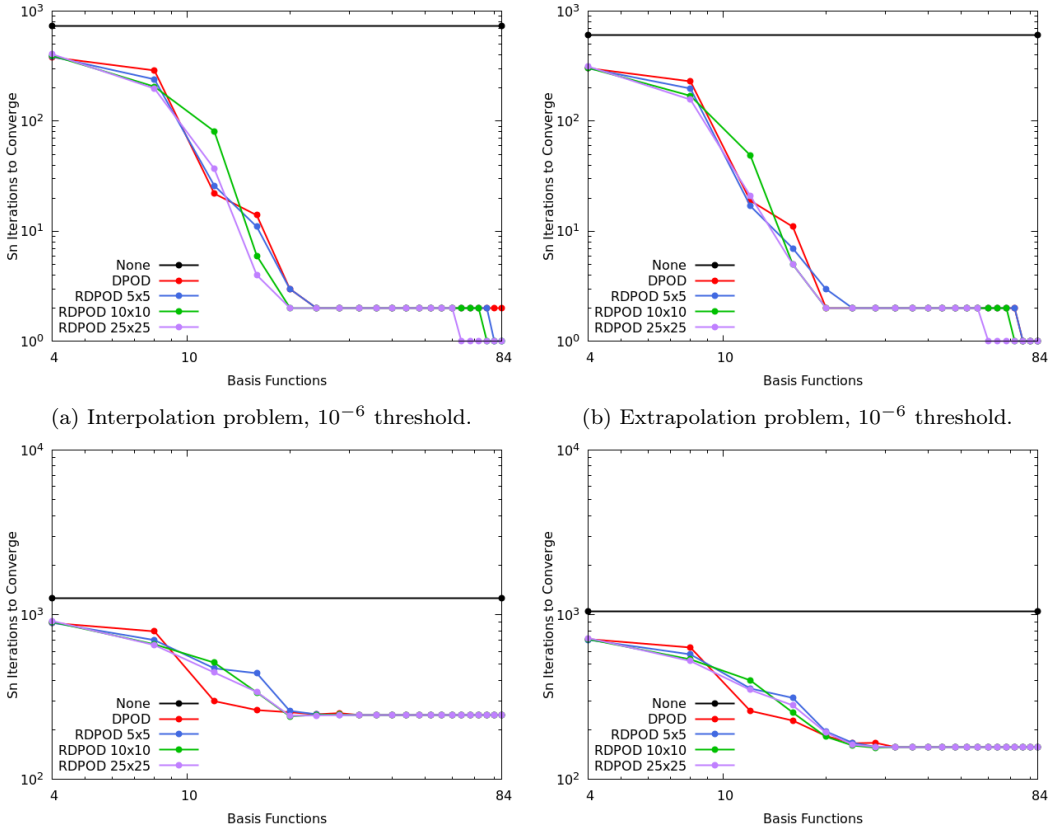
5.2.2 Numerical Examples

In this section, several numerical examples are presented which demonstrate the effectiveness of DPOD and RDPOD in accelerating solutions of the full order model, compared to the full order model alone. Uniform quadrilateral FEM spatial meshes are employed, using discontinuous bilinear basis functions. The full order method employs the S_{50} angular discretisation for all three problems. A reference solution for each problem was produced using the full order model and converged to within a tolerance of 10^{-12}. Reduced order models were then created by taking snapshots from the reference solutions, and used to solve unseen problems. These reduced order solutions were converged to a tolerance of 10^{-6}, and used as initial solutions for the full order models. The accelerated full order models were then solved with tolerances of 10^{-6} and 10^{-9}, in order to compare the results in each case. The following sections will compare the standard full order models to the accelerated models in order to demonstrate the utility of this method. Solve times will be presented, though as explained in section 1.2 the reduced order model is not fully optimised, and as such they should be taken as a general indication of whether the method works rather than a demonstration of its optimal performance. The dog-leg duct problem shown in other chapters is not considered, as it is a purely advective problem and as such only requires two iterations to converge - one for the initial rays, and one for reflected rays.

5.2.2.1 The Scattering Test Problem

The first example is a highly scattering problem, which was designed to demonstrate the utility of the methods in this chapter. A full description and explanation of the

problem is provided in section 5.1.2.1. Figure 5.2 shows a schematic of the problem and an S_{50} scalar flux solution to the extrapolation problem. The material properties of each problem are listed in table 5.1.



(a) Interpolation problem, 10^{-6} threshold.

(b) Extrapolation problem, 10^{-6} threshold.

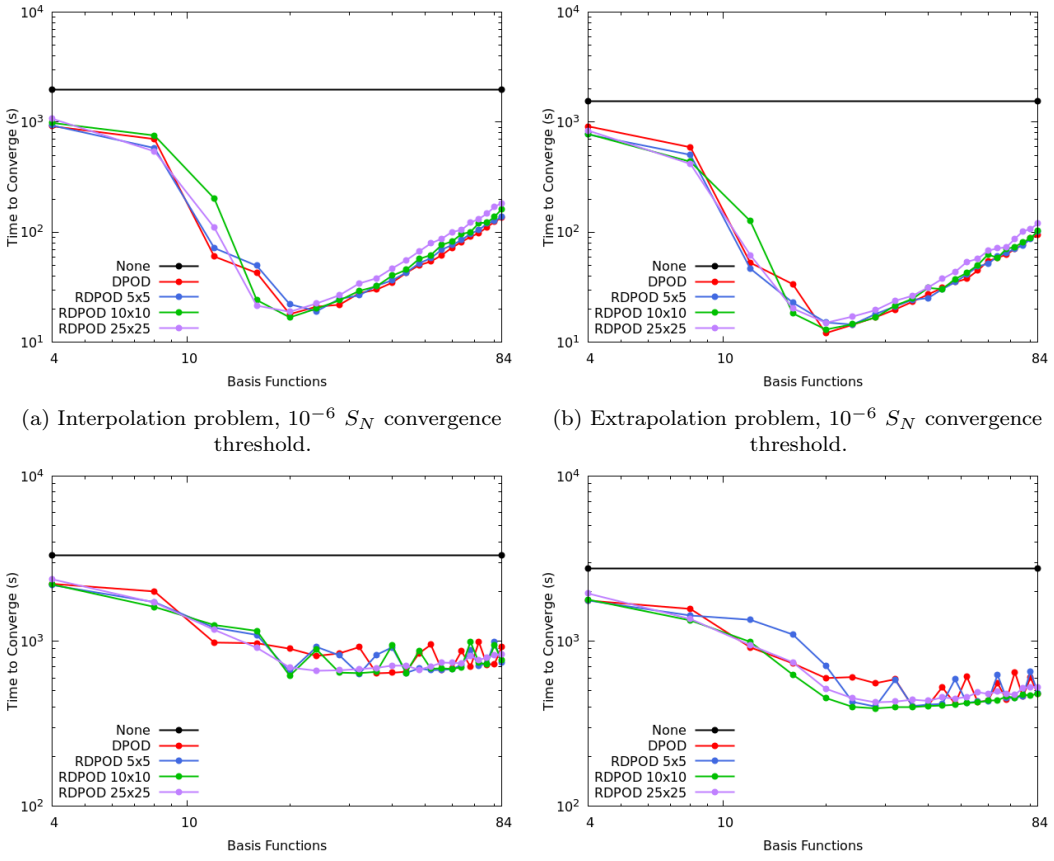
(c) Interpolation problem, 10^{-9} threshold.

(d) Extrapolation problem, 10^{-9} threshold.

Figure 5.12: The number of S_N iterations required to converge to a solution, for various accelerator configurations.

Figure 5.12 shows the number of S_N iterations required to converge to a solution of the scattering test problem after acceleration with both DPOD and RDPOD solutions, with varying numbers of basis functions and spatial regions used to generate the reduced order solutions. The number of iterations required to converge the full order model from an initial solution of zero is also shown for comparison. As the graphs demonstrate, acceleration with a reduced order model reduced the number of iterations required to converge to a solution in every case. Increasing the number of basis functions used in the reduced order model further reduced the number of full order model iterations. This was to be expected because adding basis functions reduces the ROM's error, thus ensuring that the initial solution is closer to the final solution. In the two graphs with thresholds of 10^{-6}, by around 12 BFs the reduced order model was able to approximate the S_N

solution so closely that the S_N model only required 1-2 iterations to converge, with 1 the minimum possible since an iteration must be performed to determine whether the model has converged past the threshold value. This represents a reduction of up to 3 orders of magnitude in the number of iterations required. In the 10^{-9} threshold case, the number of iterations required reached a minimum value greater than 1 and stayed there. This is likely because the reduced order model reached the smallest error it was capable of, which was not sufficient to reduce the number of iterations required to 1. Since the problem is simple, it is no surprise that this limit was reached with relatively few basis functions. As the variance in angular flux profiles across the snapshots is low, they can be accurately represented with a smaller basis than more complex problems.



(a) Interpolation problem, 10^{-6} S_N convergence threshold.

(b) Extrapolation problem, 10^{-6} S_N convergence threshold.

(c) Interpolation problem, 10^{-9} S_N convergence threshold.

(d) Extrapolation problem, 10^{-9} S_N convergence threshold.

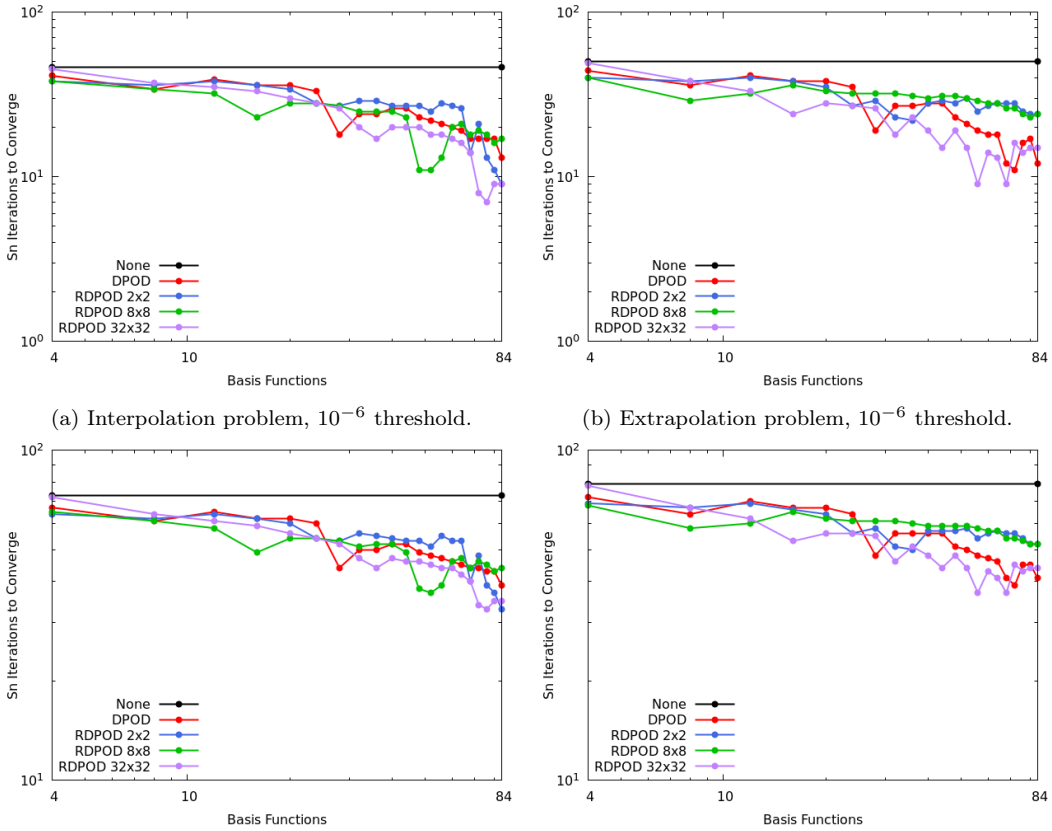
Figure 5.13: The total computation time required to converge to a solution, for various accelerator configurations.

Figure 5.13 depicts the total computation times required to converge S_N solutions of the scattering test problem, with varying numbers of basis functions used to generate the reduced order acceleration solutions. This includes the time required to solve the

reduced order model for an initial full order solution, plus the time required to iterate the accelerated S_N solution to the desired threshold. The plots demonstrate that ROM acceleration consistently reduces the total time required to converge a solution by up to 2 orders of magnitude with a threshold of 10^{-6}, and 0.5 orders of magnitude with a threshold of 10^{-9}. Increasing the number of basis functions used in the ROM was beneficial up to around 12-20 basis functions, at which point adding more basis functions begins to increase the total solve time. This can be understood by considering the results in figure 5.16. When increasing the number of basis functions used by the ROM reduces the number of S_N iterations required, it is beneficial to the overall solve time, since the reduction in S_N iterations more than counteracts the additional time required to produce the reduced order solution. Once the number of S_N iterations has reached a minimum, no further benefit can be gained from increasing the number of ROM basis functions. However, doing so will still increase the time required to produce ROM solutions, and so the total solve time begins to increase. It is also noteworthy that RDPOD with 25x25 regions performed worse than DPOD and RDPOD with fewer regions. The reason for this is not clear, but may be due to the reduced sparsity of the 25x25 matrices increasing the time required to perform matrix multiplications, without reducing the error sufficiently to provide a net positive effect. These results demonstrate the effectiveness of acceleration with reduced order models, and show that there exists an optimal number of ROM basis functions, beyond which additional basis functions are detrimental to the overall efficiency of the method.

5.2.2.2 The Watanabe-Maynard Problem

The second example in this section is the Watanabe-Maynard problem [13]. The problem setup in this case was identical to that presented in chapter 4.



(a) Interpolation problem, 10^{-6} threshold.

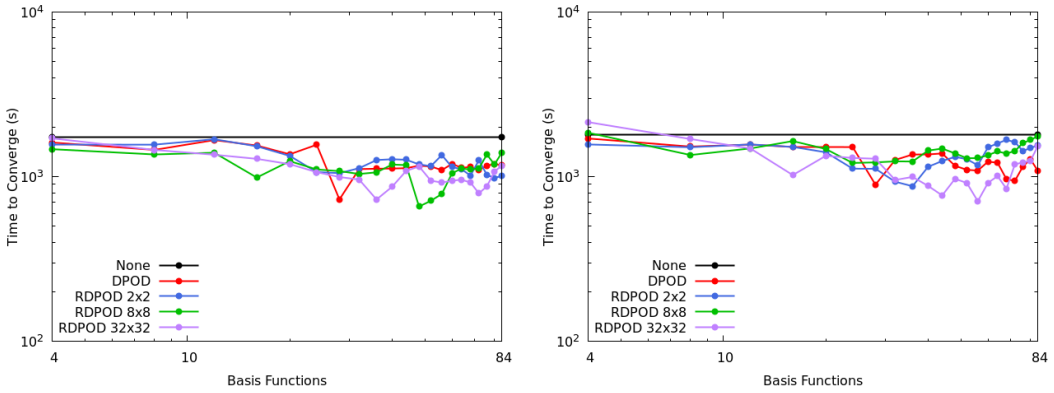
(b) Extrapolation problem, 10^{-6} threshold.

(c) Interpolation problem, 10^{-9} threshold.

(d) Extrapolation problem, 10^{-9} threshold.

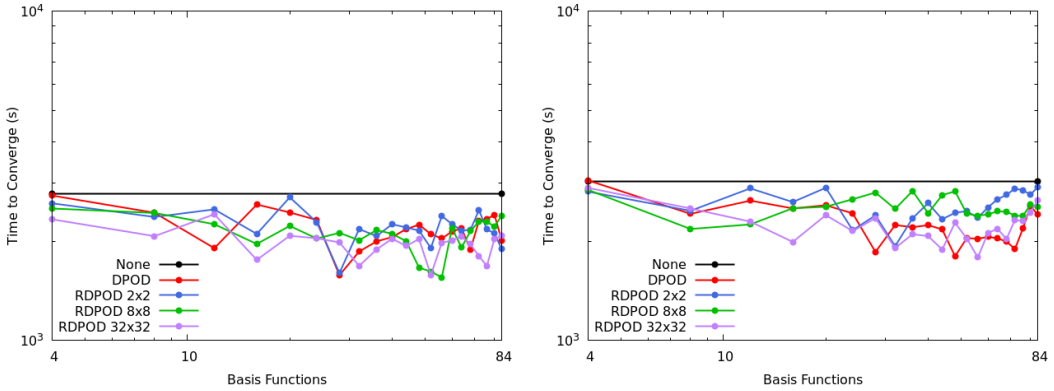
Figure 5.14: The number of S_N iterations required to converge to a solution, for various accelerator configurations.

Figure 5.14 shows the number of S_N iterations required to converge a solution to the Watanabe-Maynard problem after acceleration by both DPOD and RDPOD solutions, with varying numbers of basis functions used to generate the reduced order solutions. The number of iterations required with an initial solution of zero throughout the domain is also shown for comparison. As previously, acceleration consistently reduced the number of S_N iterations required to converge the full order model. Additional basis functions tended to reduce this number further, but oscillations about the general trend line meant that this was not always true. In this case, the reduction was not as large as for the scattering model, and never reached 1 iteration to converge. This is likely because the problem is significantly more complicated, and thus many more basis functions would be required to reach the required level of accuracy.



(a) Interpolation problem, 10^{-6} S_N convergence threshold.

(b) Extrapolation problem, 10^{-6} S_N convergence threshold.



(c) Interpolation problem, 10^{-9} S_N convergence threshold.

(d) Extrapolation problem, 10^{-9} S_N convergence threshold.

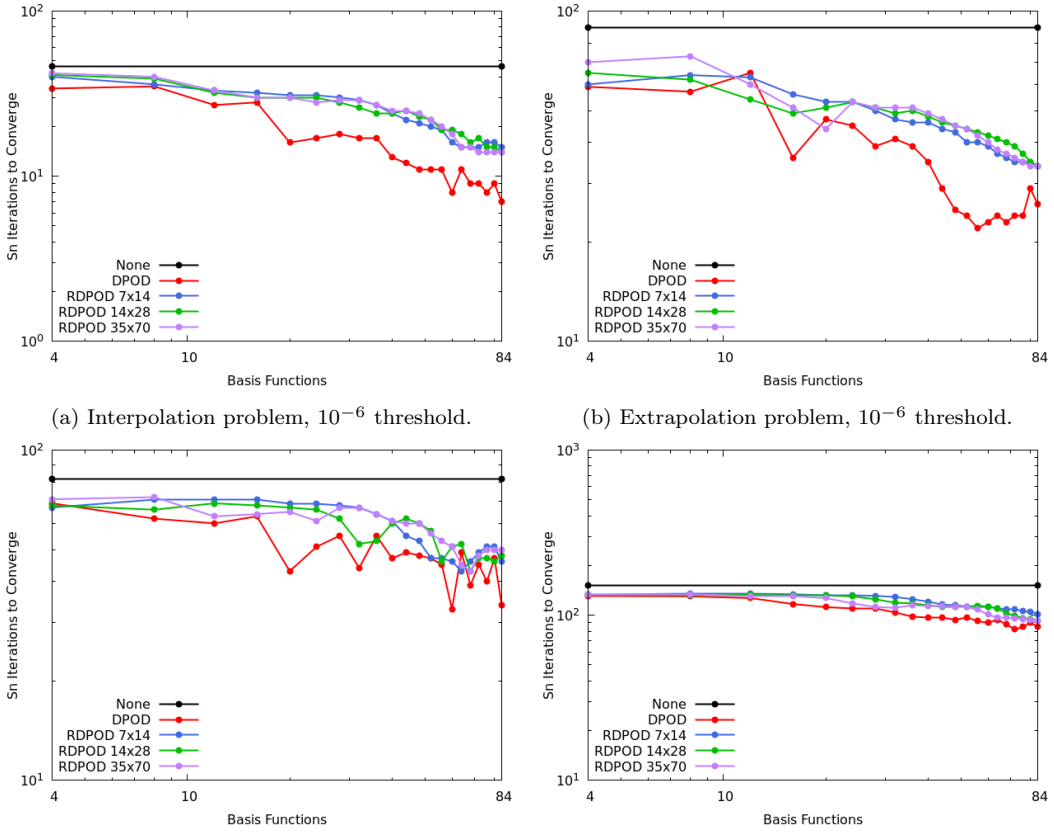
Figure 5.15: The total computation time required to converge to a solution, for various accelerator configurations.

Figure 5.15 depicts the total computation time required to converge an S_N solution to the Watanabe-Maynard problem, with varying numbers of basis functions used to generate the reduced order acceleration solutions. This includes the time required to solve the reduced order model, plus the time required to iterate the accelerated S_N solution to the desired threshold. The acceleration method reduced the total solve time in almost all cases, though it was unchanged or increased slightly for a few data points. However, the effect was much less significant than previously, reducing solve times by approximately 0.25 orders of magnitude at best. This is likely because the Watanabe-Maynard problem is more complex and thus the reduced order model requires more basis functions to reach the same level of accuracy, reducing its advantage over the full order model. The number of iterations required to solve the accelerated full order model therefore decreases relatively slowly, and as such acceleration provides less benefit. However, these results demonstrate that the acceleration method works, and

with further optimisation to the reduced order model it would provide a more significant reduction to the total solve times.

5.2.2.3 The Checkerboard Problem

The third example in this section is the checkerboard problem [14], first described in section 4.4.4. The problem was set up as presented previously.



(a) Interpolation problem, 10^{-6} threshold.

(b) Extrapolation problem, 10^{-6} threshold.

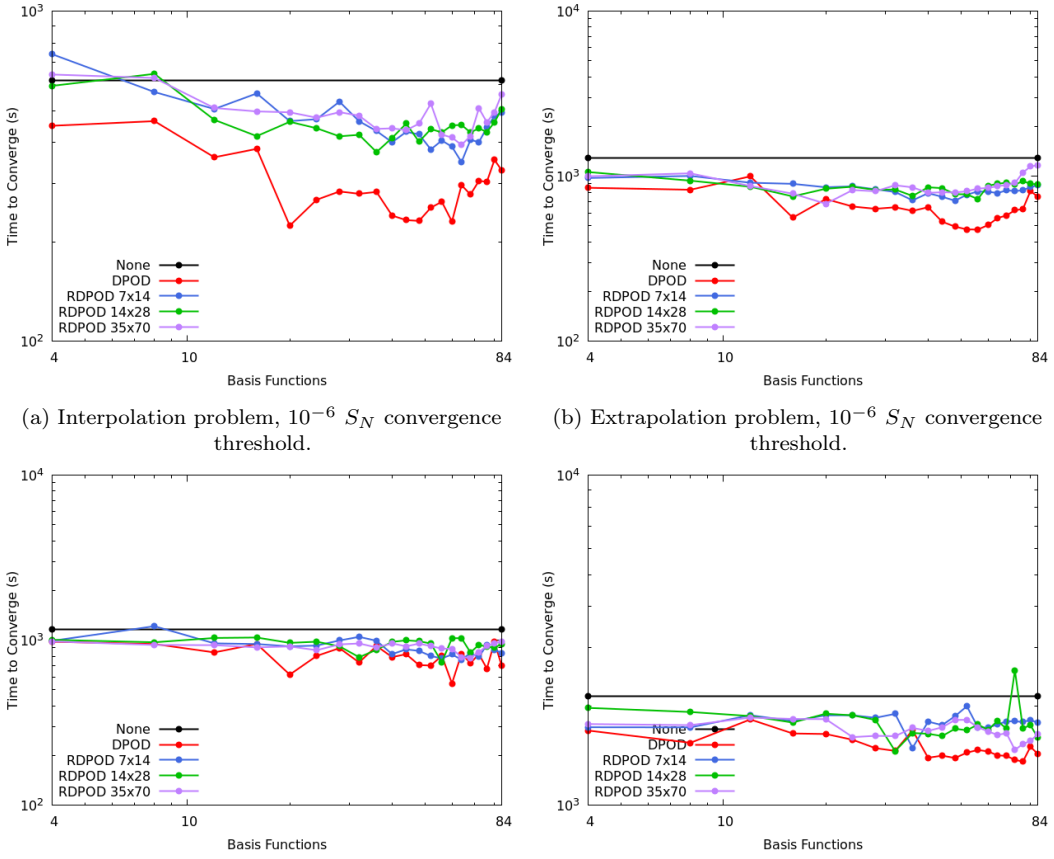
(c) Interpolation problem, 10^{-9} threshold.

(d) Extrapolation problem, 10^{-9} threshold.

Figure 5.16: The number of S_N iterations required to converge to a solution, for various accelerator configurations.

Figure 5.16 shows the number of S_N iterations required to converge to a solution for the checkerboard problem after acceleration with both DPOD and RDPOD solutions, with various reduced order model configurations used to generate the acceleration solutions. The number of iterations required without acceleration by a reduced order model is also shown for comparison. As previously, ROM acceleration consistently reduced the number of iterations required to converge the S_N model, and increasing the number of ROM basis functions tended to decrease the iterations required further, though not in every case. Interestingly, DPOD performed better than RDPOD in most cases for these

tests. It is not clear why, since RDPOD had a lower error than DPOD as expected. However, it may be that the DPOD and RDPOD models had qualitatively different error distributions, and the DPOD error was located such that it decreased more than the RDPOD error with each S_N iteration.



(a) Interpolation problem, 10^{-6} S_N convergence threshold.

(b) Extrapolation problem, 10^{-6} S_N convergence threshold.

(c) Interpolation problem, 10^{-9} S_N convergence threshold.

(d) Extrapolation problem, 10^{-9} S_N convergence threshold.

Figure 5.17: The total computation time required to converge to a solution, for various accelerator configurations.

Figure 5.17 depicts the total computation time required to converge an S_N solution to the checkerboard problem, with varying numbers of basis functions used to generate the reduced order solutions for acceleration. This includes the time required to solve the reduced order model, plus the time required to iterate the accelerated S_N solution until it converged to the desired threshold. As with the Watanabe-Maynard problem, the total solve time was reduced in most cases, but usually by a relatively small amount, approximately 0.5 orders of magnitude at best. This is likely for the same reason - the complexity of the problem meant that the ROM required many basis functions to approximate it accurately, which reduced the effectiveness of the acceleration. It should

once again be noted that the reduced order solver is not fully optimised, and that if it were, the benefit of accelerating the full order solution in this manner would be much greater.

5.2.3 Discussion

This section has demonstrated the effectiveness of using the reduced order models developed in earlier chapters to generate initial solutions to a full order model. As expected, the method was consistently able to reduce the number of S_N iterations required to converge to a solution. This resulted in a reduction in the total solve time in almost all cases. For the scattering test problem, this reduction was large, up to 2 orders of magnitude in some cases. However, the improvement was much smaller for the more complex problems. This is likely a result of the complex problems having much higher error for a given number of basis functions, which caused the initial solution to the full order model to be further from the converged solution and require more iterations to solve. However, as explained in section 1.2, the reduced order models were not fully optimised, and in fact there is an obvious way to reduce their solve times by a factor of four in two dimensions. If this was implemented, solving the ROM to generate an initial solution would take far less time, and therefore the total solve times with acceleration would all be reduced. Despite the lack of optimisation, it is clear from the results presented here that acceleration of a full order model by a reduced order model is effective in reducing overall solve times. As with the multigrid method, this type of acceleration seems to be more effective for relatively simple problems which the reduced order model can resolve with few basis functions.

5.3 Tables

Problem	Source ($cm^{-2}s^{-1}$)	Σ_a (cm^{-1})	Σ_s (cm^{-1})
1	1.00	0.10	10.00
2	1.00	0.10	14.00
3	1.00	0.20	10.00
4	1.00	0.20	14.00
Interpolation	1.00	0.15	12.00
Extrapolation	1.00	0.25	16.00

Table 5.1: Material properties for the scattering test problems in chapter 5.

Chapter 6

Summary, Conclusions and Future Research

This project has developed two novel reduced order models, known as DPOD and RDPOD. It described each method, explained the reasoning behind their implementation, and presented findings concerning their effectiveness. It then presented two applications of the aforementioned models in solver acceleration, the first being the implementation of a multigrid solver, and the second being the use of the reduced order models to find an initial solution to a full order model, in order to reduce the overall solve time. This chapter will summarise the findings of this project, draw conclusions from said findings, and propose avenues for future research.

6.1 Summary and Conclusions

6.1.1 Chapter 2

Chapter 3 presented a novel reduced order model based on angular POD, which produced basis functions with compact support, known as discontinuous POD. It demonstrated that the DPOD method was successful in eliminating the solver instability of previous angular POD techniques, which was in itself a significant improvement to computational efficiency and consistency. DPOD also reduced the error of solutions produced with a given number of basis functions in many cases, which provided further computational efficiency benefits. In addition, an adaptive DPOD method was implemented, which

took advantage of the hierarchical and discontinuous nature of the DPOD basis functions to allow the angular resolution to vary across both spatial element and angular octant. This method was highly effective at generating efficient bases with low error for the number of degrees of freedom. However, the adaptive process was optimised to generate as efficient a basis as possible at each stage, rather than to minimise the total time required to adapt and then produce a solution. In addition, the solver itself was not fully optimised, and could have been significantly improved. The adaptive method was therefore fairly inefficient in terms of total solve time, despite the fully adapted model's excellent efficiency. This could certainly be improved by altering settings, or the adaptive method itself, and by optimising the ROM solver. However, the implementation of adaptivity was intended to demonstrate that it could be effective in reducing the error for a given number of basis functions, and in this regard it was successful. Further improvements to the adaptive method are left for future work to consider.

6.1.2 Chapter 3

Chapter 4 presented a model known as Regional DPOD (RDPOD) which partitioned the spatial dimension and produced separate DPOD basis sets for each partition. This was intended to improve computational efficiency by reducing the variance in the angular flux profiles which each DPOD basis set would need to reproduce, thus requiring fewer basis functions to reach the same level of error. This method was demonstrated to significantly improve upon the computational efficiency of DPOD in many cases, and to equal it at worst. It was also found that increasing the number of spatial regions tended to consistently improve the computational efficiency of the model, which was expected since increasing the number of regions meant that each region was smaller and likely contained less variation in angular flux profiles. While this relationship was fairly consistent, in some cases the efficiency appeared to be asymptotically approaching a maximum as the number of regions was increased. An adaptive method was also implemented for the RDPOD basis functions, which allowed the number of basis functions used to vary by angular octant and spatial region. This method was demonstrated to be highly effective at reducing the error for a given basis size, to the extent that the cumulative time for the full adaptive process to reach a given level of error was faster than a single non-adaptive solve with the same level of error in many cases. As previously, the adaptive method

was optimised to reach the lowest error possible for a given basis size, rather than for a given amount of cumulative solve time, and as such its performance by that metric could likely be improved further. In theory, this method could also be modified to adapt on a per-element basis as the adaptive DPOD method did, which should produce better results, but this would be significantly more difficult to implement.

These modifications are believed to be effective because the neutron flux distribution often varies substantially over space and angle. Capturing the characteristics of the neutron flux using a single set of basis functions can therefore place high demands on the original ROM formulation. Partitioning the spatial and angular dimensions and creating separate angular ROMs for each partition can help to overcome this issue. As the variation in neutron flux distributions within each partition is considerably smaller than the variation across the entire problem, the number of basis functions required to resolve those distributions is reduced substantially. This can lead to smaller systems of equations, and therefore reduce solve times.

6.1.3 Chapter 4

Chapter 5 presented two solver acceleration methods which both utilised the DPOD and RDPOD models introduced in previous chapters. The first was a simple angular multigrid method, which solved problems in multiple stages with increasing angular resolution at each stage. The intent behind this was to allow the solution to converge to a given threshold faster. The low-order ROM coefficients which contributed most to the solution could be found relatively cheaply with low-resolution sweeps, and the high-order model would then converge faster since its initial solution was closer to the reference solution. The method was demonstrated to function as expected, successfully reducing the number of iterations required to converge the high-resolution model in all cases. However, the extent to which this reduction benefited the overall solve time varied. In the case of the relatively simple scattering test problem, the method provided a significant net benefit, reducing solve times by almost an order of magnitude. On the other hand, the Watanabe-Maynard and checkerboard problems were significantly more complex, and benefited far less overall, with their solve times reduced by approximately 0.2 orders of magnitude with 84 basis functions. As explained previously, this is suspected to be a result of the more complex problems requiring more basis functions to accurately

represent. This means that the difference between solutions using, say, 32 and 84 basis functions is much greater for complex problems like the checkerboard than simple problems like the scattering test. Therefore, when the former problem moves from a coarse to a fine grid the difference is substantial and significant iteration is required, whereas in the latter case the solutions are relatively similar and less iteration is needed. This hypothesis led to the suspicion that the multigrid method would be much more effective if the highest-resolution grid used more basis functions, since a coarse solve could then be performed which would be closer to the fine grid than was achievable with just 84 basis functions per node on the fine grid. In order to test this, the Watanabe-Maynard and checkerboard problems were solved with a fine grid that had 200 basis functions per node. As predicted, the multigrid method performed significantly better in this situation, providing evidence for the hypothesis and demonstrating its utility for even complex problems. This suggests that the multigrid method provides more benefit the more basis functions are used in the final stage, though insufficient evidence is available to determine whether this relationship holds in all circumstances.

The second method presented in this chapter was the use of DPOD and RDPOD models to generate initial solutions for a full order model, which was then used to reduce the error of the solution through further iteration. This was intended to generate solutions of equal accuracy to a full order model solve alone, but with reduced computational time. Reduced order models decrease the computation time required to generate a solution, but have the trade-off of increasing error due to the approximations involved in constructing the ROM. However, by taking a ROM solution and iterating it further with a full order model, this induced error can be eliminated. This does increase the computation time compared to solving the ROM alone, but should still be faster to reach a given level of error than solving on the full order model alone, thanks to the increased modelling efficiency provided by the ROM. As predicted, this method was successful in reducing the number of S_N iterations needed to converge a solution to a given level of accuracy, which resulted in a reduced total solve time compared to the full order model alone in almost all cases. For the scattering test problem, which was designed specifically to demonstrate the benefits of the methods in this chapter, the reduction in solve times was by up to 2 orders of magnitude. However, the Watanabe-Maynard and checkerboard problems benefited less, as they required more iteration of the full order

model to converge to a solution. This is suspected to be a result of the relative error of the more complex solutions being significantly larger, resulting in them converging to the reference solution more slowly. Despite this issue, the method was demonstrated to be effective in many situations. As discussed previously, it would also benefit from optimisation of the reduced order models. Significant optimisation of the ROM solver is expected to be possible, as discussed in section 1.2.

6.2 Future Research and Development

Many possibilities exist in terms of improvements to the models in this thesis, as well as future research based upon the methods presented. Some have been mentioned elsewhere in this document, while others are introduced for the first time here.

6.2.1 Improvements to Existing Models

This section will discuss potential improvements to the methods employed in this project. Some were apparent from initial testing but not implemented due to time limitations, and others did not become apparent until later stages of the project.

6.2.1.1 Solver Optimisation

As introduced in section 1.2 and mentioned several times in later chapters, the sweep-based solver used in the reduced order model was not well optimised for angular POD. This project aimed to develop novel reduced order modelling techniques, and not to produce an efficient and well-optimised solver. However, such optimisation would provide significant benefit in terms of the time required to solve the reduced order models, which is a key metric in almost any real-world application. In addition, it would allow the reduced order models to compare more favourably to other models of a similar kind, and to the full order model itself. As previously mentioned, an obvious improvement would be to solve only for the basis functions which are pointed in the current sweep direction. This was not possible for previous angular POD methods since every basis function spanned the sphere, but DPOD introduced basis functions with compact support on the sphere. As a result, it should be possible to solve only for the basis functions in one octant for each sweep, corresponding to the current sweep direction. This should

immediately reduce solve times by a factor of four in two dimensions, and a factor of eight in three dimensions. Other improvements are likely possible, but this modification alone presents a clear opportunity for significant efficiency improvements.

6.2.1.2 Adaptivity

In chapter 3, an adaptive method which varies the number of basis functions by element and octant is presented. When the adaptive method in chapter 4 was developed, the number of basis functions was restricted to varying by region rather than element for simplicity of implementation. While this version of adaptivity was highly successful, it is likely that the ability to adapt on a per-element basis would provide further benefit.

In addition, the adaptive methods were developed with the goal of producing as close to an optimal basis at each adaptive stage as possible. That is, the goal was to minimise the error for a given number of basis functions in the domain. This goal was chosen because, as previously mentioned, the reduced order solver was not optimised. Therefore, the number of basis functions used was initially employed as an approximate stand-in for the solve time, as can be seen in chapter 3. As a result, a large number of adaptive stages were required, and therefore the total solve times were suboptimal. The adaptive methods frequently reduced overall solve times despite this, but if they were designed to get reasonably close to an optimised basis function distribution quickly, rather than closer but more slowly, they would likely be of more benefit to solve times.

6.2.2 Novel Research Opportunities

During the course of this project, several ideas have arisen which may be worthy of investigation. They are presented here in the hopes of inspiring future work.

6.2.2.1 Optimal Spatial Partitioning

The RDPOD method introduced in chapter 4 worked by partitioning the spatial domain of a problem and producing independent basis sets for each partition, referred to as 'regions'. In the chapter, each spatial region is rectangular (usually square) and regular. This made the implementation relatively simple, but is highly unlikely to be optimal in most cases. In theory, there is no requirement for each spatial region to be regular, or even contiguous. For example, in a grid of regularly spaced fuel pins, it may turn

out to be advantageous to group every pin into the same region despite their lack of proximity, due to similarities in their angular flux profiles. While there are various possibilities for achieving a more optimised distribution of partitions, one in particular is suspected to have promise. The singular values obtained when performing the SVD contain information on the contribution of each basis function to the overall solution, as described in equation 4.11. The goal is to minimise the number of basis functions required to represent the data in a particular region. This could be achieved by maximising the early singular values and minimising later ones, as this corresponds to the early basis functions having a larger contribution in the region. This is equivalent to maximising $I_{q,r}(n)$ for a given number of basis functions n. Therefore, a method which algorithmically assigned each element to the region whose singular values were least affected by its addition could be highly beneficial. Naively, this could be achieved by selecting a single element to begin with, and performing an SVD for every possible pair of elements including that one. Then, the pairing which resulted in the largest value of I(1) would both be included in the same region. Similar tests which attempted to maximise I for small values of n could then be performed, adding the element which resulted in the smallest change each time. At some threshold, a new region would be created and the same process performed until every element was assigned to a region. While the benefits of such a method would clearly be problem-dependent, and may be minor in some cases, it is suspected that this method could provide significant reductions in error for a given number of basis functions in the domain. In addition, as presented the algorithm would require significant computation, but far more efficient implementations of the same concept are likely possible.

6.2.2.2 Combining Reduced and Full Order Models

Since the reduced and full order models constructed for this project use the same spatial grid and differ only in angular discretisation, it should be possible to produce a model which uses reduced order modelling in some spatial regions and full order modelling in others. This may be highly beneficial for some applications, particularly when small portions of the domain are either of greater interest, or have more variance in their angular flux profile. By solving these small sections with a full order model and the rest with a reduced order model, their error could be minimised while retaining most of the

efficiency benefits of the ROM. This would require the implementation of appropriate boundary conditions between the two models so that flux could flow between them, which should be relatively simple. Such a method would likely only be worthwhile for some problems, and it is not clear how much benefit it would provide, but further investigation may be warranted.

6.2.2.3 Combining Angular and Spatial Reduced Order Modelling

This project has focused on the development of angular ROM methods - that is, methods which reduce the degrees of freedom of the angular dimensions of the model. Similar methods can be applied to the spatial domain, as discussed in section 2.6.3. These involve taking snapshots of the spatial domain, either over time or with varying material properties, and generating basis functions which can efficiently represent the spatial variance of the flux. In principle, it should be possible to combine the angular ROMs presented in this thesis with spatial order reduction in the same model. This would be a significant undertaking to implement, but may perform better than ROMs involving either the angular or spatial dimensions alone. As such, it represents a promising avenue for future research to explore.

Bibliography

[1] Y. Chen, B. Zhang, L. Zhang, J. Zheng, Y. Zheng, and C. Liu. Ares: A parallel discrete ordinates transport code for radiation shielding applications and reactor physics analysis. *Science and Technology of Nuclear Installations*, 2017, Mar 2017. 8, 38

[2] M. Wiercioch. Exploring the potential of spherical harmonics and pcvm for compounds activity prediction. *International Journal of Molecular Sciences*, 20:2175, May 2019. 8, 40

[3] E.W. Weisstein. Haar function. From MathWorld–A Wolfram Web Resource. Accessed 17/10/22. *https://mathworld.wolfram.com/HaarFunction.html*. 8, 41

[4] Shawn D Pautz. An algorithm for parallel sn sweeps on unstructured meshes. *Nuclear Science and Engineering*, 140(2):111–136, 2002. 8, 48

[5] M. Ragheb. Decay heat generation in fission reactors. *Probabilistic, Possibilistic, and Deterministic Safety Analysis, Nuclear Applications*, page 20, 2011. 20

[6] T.P. McLaughlin, S.P. Monahan, N.L. Pruvost, V.V. Frolov, B.G. Ryazanov, and V.I. Sviridov. A review of criticality accidents 2000 revision. *Los Alamos National Laboratory Report LA-13638*, 2000. 21

[7] L. Boltzmann. *Weitere Studien über das Wärmegleichgewicht unter Gasmolekülen*. k. und k. Hof- und Staatsdr., 1872. 21, 30

[8] P. Gressman and R. Strain. Global classical solutions of the boltzmann equation without angular cut-off. *Journal of the American Mathematical Society*, 24(3):771–847, Mar 2011. 21

[9] S. Dargaville, R.P. Smedley-Stevenson, P.N. Smith, and C.C. Pain. Goal-based angular adaptivity for boltzmann transport in the presence of ray-effects. *Journal of Computational Physics*, 421:109759, 2020. 22, 54

[10] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936. 23

[11] A.G. Buchan, A.A. Calloo, M.G. Goffin, S. Dargaville, F. Fang, C.C. Pain, and I.M. Navon. A pod reduced order model for resolving angular direction in neutron/photon transport problems. *Journal of Computational Physics*, 296:138 – 157, 2015. 23, 28, 57, 58, 60, 65, 67, 70, 90, 97, 101

[12] A.G. Buchan, C.C. Pain, M.D. Eaton, R.P. Smedley-Stevenson, and A.J.H. Goddard. Linear and quadratic octahedral wavelets on the sphere for angular discretisations of the boltzmann transport equation. *Annals of Nuclear Energy*, 32(11):1224 – 1273, 2005. 25, 73, 107

[13] Y. Watanabe and C.W. Maynard. The discrete cones method for two-dimensional neutron transport calculations. *Transport Theory and Statistical Physics*, 15(1-2):135–156, 1986. 26, 82, 113, 133, 145

[14] T.A. Brunner. Forms of approximate radiation transport. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States); Sandia ..., 2002. 27, 118, 137, 148

[15] A.C. Hughes and A.G. Buchan. A discontinuous and adaptive reduced order model for the angular discretization of the boltzmann transport equation. *International Journal for Numerical Methods in Engineering*, 121:5647–5666, 2020. 28, 90, 101, 121

[16] A.C. Hughes and A.G. Buchan. An adaptive reduced order model for the angular discretization of the boltzmann transport equation using independent basis sets over a partitioning of the space-angle domain. *International Journal for Numerical Methods in Engineering*, 2022. 28

[17] E.E. Lewis and W.F. Miller. *Computational methods of neutron transport*. John Wiley and Sons, Inc., New York, NY, 1984. 30, 32, 34, 35

[18] F. Golse. Applications of the boltzmann equation within the context of upper atmosphere vehicle aerodynamics. *Computer Methods in Applied Mechanics and Engineering*, 75(1):299 – 316, 1989. 31

[19] B. Evans and S.P. Walton. Aerodynamic optimisation of a hypersonic reentry vehicle based on solution of the boltzmann–bgk equation and evolutionary optimisation. *Applied Mathematical Modelling*, 52:215–240, 2017. 31

[20] M. Mohebbi and M.A. Rezvani. Multi objective optimization of aerodynamic design of high speed railway windbreaks using lattice boltzmann method and wind tunnel test results. *International Journal of Rail Transportation*, 6(3):183–201, 2018. 31

[21] C. Luzzato, J. Biermann, and R. Fouque. The aero-acoustic design and optimization of a ground transportation hvac system using lattice boltzmann methods. In *2019 National Agency for Finite Element Methods and Standards World Congress*, Jun 2019. 31

[22] S. Aoki. Hydrodynamics of a highly flattened galaxy, based on the collisionless boltzmann equation. *Astronomy and Astrophysics*, 148:1–11, 1985. 31

[23] S.R. Cranmer, G.B. Field, G. Noci, and J.L. Kohl. The impact of ion-cyclotron wave dissipation on minor ion velocity distributions in the solar corona. In *American Astronomical Society Meeting Abstracts*, volume 191, pages 74–11, 1997. 31

[24] M. Kim, S. Thibeault, L. Simonsen, and J. Wilson. *Comparison of Martian Meteorites and Martian Regolith as Shield Materials for Galactic Cosmic Rays*. National Aeronautics and Space Administration, Langley Research Center, Nov 1998. 31

[25] R. Maartens, T. Gebbie, and G. Ellis. Cosmic microwave background anisotropies: Nonlinear dynamics. *Physical Review D*, 59(8), Mar 1999. 31

[26] X. Cheng, F. Hu, and Q. Zeng. Simulation of wind gust structure in the atmospheric boundary layer with lattice boltzmann method. *Chinese Science Bulletin*, 57(10):1196–1203, 2012. 31

[27] Y. Feng, P. Boivin, J. Jacob, and P. Sagaut. Hybrid recursive regularized lattice boltzmann simulation of humid air with application to meteorological flows. *Phys. Rev. E*, 100:023304, Aug 2019. 31

[28] J. Jacob, L. Merlier, F. Marlow, and P. Sagaut. Lattice boltzmann method-based simulations of pollutant dispersion and urban physics. *Atmosphere*, 12(7), 2021. 31

[29] M.H. Hekmat and M. Mirzaei. Continuous and discrete adjoint approach based on lattice boltzmann method in aerodynamic optimization part i: Mathematical derivation of adjoint lattice boltzmann equations. *Advances in Applied Mathematics and Mechanics*, 6(5):570–589, 2014. 31

[30] I. Cheylan, G. Fritz, D. Ricot, and P. Sagaut. Shape optimization using the adjoint lattice boltzmann method for aerodynamic applications. *AIAA journal*, 57(7):2758–2773, 2019. 31

[31] W. Li, J. Carrete, N.A. Katcho, and N. Mingo. Shengbte: A solver of the boltzmann transport equation for phonons. *Computer Physics Communications*, 185(6):1747–1758, 2014. 31

[32] A. Puglisi, S. Plumari, and V. Greco. Electric conductivity from the solution of the relativistic boltzmann equation. *Phys. Rev. D*, 90:114009, Dec 2014. 31

[33] P. Silvennoinen. A self-adjoint form of the linear transport equation. *Journal of Mathematical Analysis and Applications*, 43(2):529–536, 1973. 35

[34] J.L. Liscum-Powell, A.B. Prinja, J.E. Morel, and L.J. Lorence Jr. Finite element solution of the self-adjoint angular flux equation for coupled electron-photon transport. *Nuclear Science and Engineering*, 142(3):270–291, 2002. 35

[35] C.R.E. de Oliveira. An arbitrary geometry finite element method for multigroup neutron transport with anisotropic scattering. *Progress in Nuclear Energy*, 18(1):227–236, 1986. Finite Element and Allied Methods. 35, 40

[36] L. Cao and H. Wu. A spherical harmonics—finite element discretization of the self-adjoint angular flux neutron transport equation. *Nuclear engineering and design*, 237(23):2232–2239, 2007. 35

[37] C.R. Drumm and J. Lorenz. Parallel fe approximation of the even/odd parity form of the linear boltzmann equation. *Mathematical and Computer Modeling*, 31(2):55–71, Jul 1999. 35

[38] J.E. Morel and J.M. McGhee. A self-adjoint angular flux equation. *Nuclear Science and Engineering*, 132(3):312–325, 1999. 35

[39] A.G. Buchan, C.C. Pain, M.D. Eaton, R.P. Smedley-Stevenson, and A.J.H. Goddard. Self-adaptive spherical wavelets for angular discretizations of the boltzmann transport equation. *Nuclear Science and Engineering*, 158(3):244–263, 2008. 36, 54

[40] P. Renard, A. Alcolea, and D. Ginsbourger. Stochastic versus deterministic approaches. *Environmental modelling: Finding simplicity in complexity*, pages 133–149, 2013. 36

[41] J.A. Willert. *Hybrid deterministic/Monte Carlo methods for solving the neutron transport equation and k-eigenvalue problem.* North Carolina State University, 2013. 36

[42] A. Tveit. On the complexity of matrix inversion. *Mathematical Note*, page 1, 2003. 36, 47

[43] J. Spanier and E.M. Gelbard. *Monte Carlo principles and neutron transport problems.* Courier Corporation, 2008. 37

[44] S.P. Hamilton and T.M. Evans. Continuous-energy monte carlo neutron transport on gpus in the shift code. *Annals of Nuclear Energy*, 128:236–247, 2019. 37

[45] T.E. Booth. Exponential convergence for monte carlo particle transport. *Trans. Am. Nucl. Soc.; (United States)*, 50, Nov 1985. 37

[46] S. Chandrasekhar. *Radiative Transfer.* Dover Publications, New York, 1960. 38

[47] R. Koch and R. Becker. Evaluation of quadrature schemes for the discrete ordinates method. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 84(4):423–435, 2004. Eurotherm Seminar 73 - Computational Thermal Radiation in Participating Media. 38

[48] J. Tencer. A comparison of angular discretization techniques for the radiative transport equation. In *ASME International Mechanical Engineering Congress and Exposition*, volume 57502, page V08BT10A034. American Society of Mechanical Engineers, 2015. 39

[49] J. Kópházi and D. Lathouwers. A space–angle dgfem approach for the boltzmann radiation transport equation with local angular refinement. *Journal of Computational Physics*, 297:637 – 668, 2015. 39, 41, 47, 54, 65, 96

[50] A.R. Owens, J.A. Welch, J. Kópházi, and M.D. Eaton. Discontinuous isogeometric analysis methods for the first-order form of the neutron transport equation with discrete ordinate (sn) angular discretisation. *Journal of Computational Physics*, 315:501 – 535, 2016. 39, 65, 96

[51] T.S. Haut, P.G. Maginot, V.Z. Tomov, B.S. Southworth, T.A. Brunner, and T.S. Bailey. An efficient sweep-based solver for the $s\_\{N\}$ equations on high-order meshes. *arXiv preprint arXiv:1810.11080*, 2018. 39, 47

[52] T.M. Evans, A.S. Stafford, R.N. Slaybaugh, and K.T. Clarno. Denovo: A new three-dimensional parallel discrete ordinates code in scale. *Nuclear Technology*, 171(2):171–200, 2010. 39

[53] Y. Wang, S. Schunert, J. Ortensi, V. Laboure, M. DeHart, Z. Prince, F. Kong, J. Harter, P. Balestra, and F. Gleicher. Rattlesnake: A moose-based multiphysics multischeme radiation transport application. *Nuclear Technology*, 207(7):1047–1072, 2021. 39, 40

[54] G. Marleau, A. Hebert, and R. Roy. New computational methods used in the lattice code dragon. Technical report, Ecole Polytechnique de Montreal, Quebec (Canada), United States, 1992. CONF-920308–Vol1. 39

[55] G.L. Olson. Spherical harmonic solutions of the radiation transport equation using angle-dependent artificial scattering to decrease oscillations. Technical report, Tech. Rep. LA-UR-12-22927, Los Alamos National Laboratory, Los Alamos, NM, 2012. 40

[56] L. Cao and H. Wu. A spherical harmonics—finite element discretization of the

self-adjoint angular flux neutron transport equation. *Nuclear Engineering and Design - NUCL ENG DES*, 237:2232–2239, Dec 2007. 40

[57] A. Haar. Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 69(3):331–371, Sep 1910. 41

[58] B.J. Adigun, A.G. Buchan, A. Adam, S. Dargaville, M.A. Goffin, and C.C. Pain. A haar wavelet method for angularly discretising the boltzmann transport equation. *Progress in Nuclear Energy*, 108:295–309, 2018. 41

[59] L. Soucasse, S. Dargaville, A.G. Buchan, and C.C. Pain. A goal-based angular adaptivity method for thermal radiation modelling in non grey media. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 200:215–224, 2017. 41, 54

[60] M.A. Goffin, A.G. Buchan, S. Dargaville, C.C. Pain, P.N. Smith, and R.P. Smedley-Stevenson. Goal-based angular adaptivity applied to a wavelet-based discretisation of the neutral particle transport equation. *Journal of Computational Physics*, 281:1032–1062, 2015. 41

[61] B.G. Carlson. Transport theory: Formulations and solutions by finite difference methods. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 1968. 42

[62] Z. Zhong, T.J. Downar, Y. Xu, M.D. DeHart, and K.T. Clarno. Implementation of two-level coarse-mesh finite difference acceleration in an arbitrary geometry, two-dimensional discrete ordinates transport method. *Nuclear Science and Engineering*, 158(3):289–298, 2008. 43

[63] M.J. Lee, H.G. Joo, D. Lee, and K. Smith. Coarse mesh finite difference formulation for accelerated monte carlo eigenvalue calculation. *Annals of Nuclear Energy*, 65:101–113, 2014. 43

[64] A. Zhu, M. Jarrett, Y. Xu, B. Kochunas, E. Larsen, and T. Downar. An optimally diffusive coarse mesh finite difference method to accelerate neutron transport calculations. *Annals of Nuclear Energy*, 95:116–124, 2016. 43

[65] A. Hrennikoff. Solution of problems of elasticity by the framework method. *Journal of Applied Mechanics*, 1941. 45

[66] R. Courant. Variational methods for the solution of problems of equilibrium and vibrations. *Bulletin of the American mathematical Society*, 49(1):1–23, 1943. 45

[67] T. Ohnishi. Application of finite element solution technique to neutron diffusion and transport equations. Technical report, Hitachi Ltd., Kawasaki (Japan). Atomic Energy Research Lab., Oct 1971. 45

[68] T. Ohnishi. Finite element method applied to reactor physics problems. *Journal of Nuclear Science and Technology*, 8(12):717–720, 1971. 45

[69] W.H. Reed and T.R. Hill. Triangular mesh methods for the neutron transport equation. Technical report, Los Alamos Scientific Lab., N. Mex.(USA), 1973. 45, 47

[70] P. Lesaint and P.A. Raviart. On a finite element method for solving the neutron transport equation. In Carl de Boor, editor, *Mathematical Aspects of Finite Elements in Partial Differential Equations*, pages 89–123. Academic Press, 1974. 45

[71] H. Guo, W. Chen, X. Jiang, D. Jiang, and X. Zhang. A discontinuous finite element method based on b-spline wavelet on the interval for solving first-order neutron transport equation with discrete ordinate (sn) angular discretisation. *Annals of Nuclear Energy*, 162:108511, 2021. 46

[72] V. Labouré, Y. Wang, J. Ortensi, N. Martin, and S. Schunert. Multischeme equivalence procedure for neutron transport finite element methods. *Annals of Nuclear Energy*, 166:108712, 2022. 46

[73] E. Lum and C.L. Pope. Simulation of the fast reactor fuel assembly duct-bowing reactivity effect using monte carlo neutron transport and finite element analysis. *Nuclear Technology*, 207(5):761–770, 2021. 46

[74] Y. Chan and S. Xiao. A linear prolongation cmfd acceleration for two-dimensional discrete ordinate k-eigenvalue neutron transport calculation with pin-resolved mesh using discontinuous galerkin finite element method. *Annals of Nuclear Energy*, 154:108103, 2021. 46

[75] S.A. Orszag. Numerical methods for the simulation of turbulence. *The Physics of Fluids*, 12(12):II–250, 1969. 46

[76] S.A. Orszag. Spectral methods for problems in complex geometrics. In *Numerical methods for partial differential equations*, pages 273–305. Elsevier, 1979. 46

[77] A.D. Kim and M. Moscoso. Chebyshev spectral methods for radiative transfer. *SIAM Journal on scientific computing*, 23(6):2074–2094, 2002. 46

[78] W. Guo, G. Labrosse, and R. Narayanan. *The application of the Chebyshev-Spectral method in transport phenomena*. Springer Science & Business Media, 2013. 46

[79] N. Nahavandi. The evaluation of different trial shape functions applied in spectral element method in analysis of multi-group pn transport equation. *Annals of Nuclear Energy*, 103:147–156, 2017. 46

[80] A. Shokri and S. Mirzaei. Numerical study of the two-term time-fractional differential equation using the lagrange polynomial pseudo-spectral method. *Alexandria Engineering Journal*, 59(5):3163–3169, 2020. 46

[81] Ü. Lepik. Numerical solution of differential equations using haar wavelets. *Mathematics and Computers in Simulation*, 68(2):127–143, 2005. 47

[82] S. Qian and J. Weiss. Wavelets and the numerical solution of partial differential equations. *Journal of Computational Physics*, 106(1):155–175, 1993. 47

[83] J. Brannick, M. Brezina, R. Falgout, T. Manteuffel, S. McCormick, J. Ruge, B. Sheehan, J. Xu, and L. Zikatanov. Extending the applicability of multigrid methods. *Journal of Physics: Conference Series*, 46(1):443, sep 2006. 50

[84] B. Chang, T. Manteuffel, S. McCormick, J. Ruge, and B. Sheehan. Spatial multigrid for isotropic neutron transport. *SIAM Journal on Scientific Computing*, 29(5):1900–1917, 2007. 50

[85] U. Ghia, K.N. Ghia, and C.T. Shin. High-re solutions for incompressible flow using the navier-stokes equations and a multigrid method. *Journal of Computational Physics*, 48(3):387–411, 1982. 52

[86] L. Martinelli. *Calculations of viscous flows with a multigrid method*. Princeton University, 1987. 52

[87] M. Holst and F. Saied. Multigrid solution of the poisson—boltzmann equation. *Journal of Computational Chemistry*, 14(1):105–113, 1993. 52

[88] A.G. Buchan, C.C. Pain, A.P. Umpleby, and R.P. Smedley-Stevenson. A sub-grid scale finite element agglomeration multigrid method with application to the boltzmann transport equation. *International journal for numerical methods in engineering*, 92(3):318–342, 2012. 52

[89] M.A. Abouelatta, S.A. Ward, A.M. Sayed, K. Mahmoud, M. Lehtonen, and M.M.F. Darwish. Fast corona discharge assessment using fdm integrated with full multigrid method in hvdc transmission lines considering wind impact. *Ieee Access*, 8:225872–225883, 2020. 52

[90] O. Steinbach and H. Yang. Comparison of algebraic multigrid methods for an adaptive space–time finite-element discretization of the heat equation in 3d and 4d. *Numerical Linear Algebra with Applications*, 25(3):e2143, 2018. e2143 nla.2143. 52

[91] R.N. Slaybaugh, T.M. Evans, G.G. Davidson, and P.P.H. Wilson. Multigrid in energy preconditioner for krylov solvers. *Journal of Computational Physics*, 242:405–419, 2013. 52

[92] C. Wu, R. Girshick, K. He, C. Feichtenhofer, and P. Krahenbuhl. A multigrid method for efficiently training video models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2020. 52

[93] A. Gupta and R.S. Modak. On the use of the conjugate gradient method for the solution of the neutron transport equation. *Annals of Nuclear Energy*, 29(16):1933–1951, 2002. 53

[94] B.W. Patton. *Application of Krylov subspace iterative techniques to the numerical solution of the neutron transport equation.* University of Michigan, 1996. 53

[95] B.W. Patton and J.P. Holloway. Application of preconditioned gmres to the numerical solution of the neutron transport equation. *Annals of Nuclear Energy*, 29(2):109–136, 2002. 53

[96] S. Dargaville, A.G. Buchan, R.P. Smedley-Stevenson, P.N. Smith, and C.C. Pain. Scalable angular adaptivity for boltzmann transport. *Journal of Computational*

Physics, 406:109124, 2020. 54

[97] H. Park and C.R.E. de Oliveira. Adaptive angular resolution for the finite element-spherical harmonics method. *Transactions of the American Nuclear Society*, 93:517–519, 2005. 54

[98] S. Dargaville, A.G. Buchan, R.P. Smedley-Stevenson, P.N. Smith, and C.C. Pain. Angular adaptivity with spherical harmonics for boltzmann transport. *Journal of Computational Physics*, 397:108846, 2019. 54

[99] K.H. Sadeghi, M. Abbasi, and A. Zolfaghari. Adaptive refinement for pn neutron transport equation based on conjoint variational formulation and discontinuous finite element method. *Progress in Nuclear Energy*, 104:16–31, 2018. 54

[100] J.J. Jarrell. *An adaptive angular discretization method for neutral-particle transport in three-dimensional geometries.* Texas A&M University, 2010. 54

[101] H. Riou, P. Ladevèze, B. Sourcis, B. Faverjon, and L. Kovalevsky. An adaptive numerical strategy for the medium-frequency analysis of helmholtz's problem. *Journal of Computational Acoustics*, 20(01):1250001, 2012. 54

[102] L.D. Marchi, E. Baravelli, F. Franze, and N. Speciale. 3d mesh generation with wavelet-driven adaptivity. In *2006 International Conference on Simulation of Semiconductor Processes and Devices*, pages 212–215, 2006. 54

[103] M.A. Goffin, A.G. Buchan, A.C. Belme, C.C. Pain, M.D. Eaton, P.N. Smith, and R.P. Smedley-Stevenson. Goal-based angular adaptivity applied to the spherical harmonics discretisation of the neutral particle transport equation. *Annals of Nuclear Energy*, 71:60–80, 2014. 54

[104] B. Zhang, L. Zhang, C. Liu, and Y. Chen. Goal-oriented regional angular adaptive algorithm for the sn equations. *Nuclear Science and Engineering*, 189(2):120–134, 2018. 54

[105] H. Zhang and E.E. Lewis. Spatial adaptivity applied to the variational nodal pn equations. *Nuclear Science and Engineering*, 142(1):57–63, 2002. 55

[106] O. Safarzadeh. Application of h-adaptivity to the interface current collision probability method. *Annals of Nuclear Energy*, 94:826–831, 2016. 55

[107] A.R. Owens, J.A. Welch, J. Kópházi, and M.D. Eaton. An adaptive, hanging-node, discontinuous isogeometric analysis method for the first-order form of the neutron transport equation with discrete ordinate (sn) angular discretisation. *Computer Methods in Applied Mechanics and Engineering*, 318:215–241, 2017. 55

[108] M. Abbassi, A. Zolfaghari, A. Minuchehr, and M. Yousefi. An adaptive finite element approach for neutron transport equation. *Nuclear Engineering and Design*, 241(6):2143–2154, 2011. (W3MDM) University of Leeds International Symposium: What Where When? Multi-dimensional Advances for Industrial Process Monitoring. 55

[109] J.S. Warsa and A.K. Prinja. A p-adaptive numerical transport method. *Transactions of the American Nuclear Society*, 79, Dec 1998. 55

[110] F. Naddei, M. de la Llave Plata, and V. Couaillier. A comparison of refinement indicators for p-adaptive discontinuous galerkin methods for the euler and navier-stokes equations. In *2018 AIAA Aerospace Sciences Meeting*, page 0368, 2018. 55

[111] W. Li, H. Luo, A. Pandare, and J. Bakosi. A p-adaptive discontinuous galerkin method for compressible flows using charm++. In *AIAA Scitech 2020 forum*, page 1565, 2020. 55

[112] F. Chinesta, P. Ladeveze, and E. Cueto. A short review on model order reduction based on proper generalized decomposition. *Archives of Computational Methods in Engineering*, 18(4):395, 2011. 56

[113] M. Barrault, Y. Maday, N.C. Nguyen, and A.T. Patera. An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique*, 339(9):667–672, 2004. 56

[114] N.C. Nguyen and J. Peraire. An efficient reduced-order modeling approach for non-linear parametrized partial differential equations. *International Journal for Numerical Methods in Engineering*, 76(1):27–55, 2008. 56

[115] D. Xiao, F. Fang, J. Zheng, C.C. Pain, and I.M. Navon. Machine learning-based rapid response tools for regional air pollution modelling. *Atmospheric Environment*,

199:463–473, 2019. 56

[116] W. Silva. Identification of nonlinear aeroelastic systems based on the volterra theory: progress and opportunities. *Nonlinear Dynamics*, 39(1):25–62, 2005. 56

[117] M. Rydel and R. Stanisławski. A new frequency weighted fourier-based method for model order reduction. *Automatica*, 88:107–112, 2018. 56

[118] J.W. Bandler, Q.S. Cheng, S.A. Dakroury, A.S. Mohamed, M.H. Bakr, K. Madsen, and J. Sondergaard. Space mapping: the state of the art. *IEEE Transactions on Microwave Theory and Techniques*, 52(1):337–361, 2004. 56

[119] G. Aversano, A. Bellemans, Z. Li, A. Coussement, O. Gicquel, and A. Parente. Application of reduced-order models based on pca & kriging for the development of digital twins of reacting flow applications. *Computers & Chemical Engineering*, 121:422–441, 2019. 56

[120] X. Chen, L. Liu, T. Long, and Z. Yue. A reduced order aerothermodynamic modeling framework for hypersonic vehicles based on surrogate and pod. *Chinese Journal of Aeronautics*, 28(5):1328–1342, 2015. 56

[121] J. Blahoš, A. Vizzaccaro, L. Salles, and F. el Haddad. Parallel harmonic balance method for analysis of nonlinear dynamical systems. In *Turbo Expo: Power for Land, Sea, and Air*, volume 84232, page V011T30A028. American Society of Mechanical Engineers, 2020. 56

[122] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Computer Science and Scientific Computing. Academic Press, second edition, 1990. 56

[123] I.T. Jolliffe. *Principal Component Analysis*. Springer, Berlin, second edition, Oct 2002. 56

[124] D.T. Crommelin and A.J. Majda. Strategies for model reduction: comparing different optimal bases. *Journal of the Atmospheric Sciences*, 61:2206–2217, 2004. 56

[125] J.L. Lumley. The structure of inhomogeneous turbulent flows. *Atmospheric Turbulence and Radio Wave Propagation*, pages 166–177, 1967. 56

[126] H.P. Bakewell and J.L. Lumley. Viscous sublayer and adjacent wall region in turbulent pipe flow. *The Physics of Fluids*, 10:1880–1889, 1967. 56

[127] F.R. Payne and J.L. Lumley. Large-eddy structure of the turbulent wake behind a circular cylinder. *The Physics of Fluids*, 10:194–196, 1967. 56

[128] A.E. Dean and C. Mavriplis. Low-dimensional description of the dynamics in separated flow past thick airfoil. *AIAA*, 32:1222–1227, 1994. 56

[129] J. Delville, S. Bellin, and J.P. Bonnet. *Use of the proper orthogonal decomposition in a plane turbulent mixing layer, in Turbulence and Coherent Structures*, pages 75–90. Kluwer Academic Publishers, Dordrecht, 1991. 56

[130] L. Sirovich and H. Park. Turbulent thermal convection in a finite domain: PartI. theory. *Physics of fluids*, 2:1649 –1658, 1990. 56

[131] H. Park and L. Sirovich. Turbulent thermal convection in a finite domain: PartII. numerical results. *Physics of fluids*, 2:1659 –1668, 1990. 56

[132] F. Fang, C.C. Pain, I.M. Navon, G.J. Gorman, M.D. Piggott, P.A. Allison, P.E. Farrell, and A.J.H. Goddard. A POD reduced order unstructured mesh ocean modelling method for moderate reynolds number flows. *Ocean Modelling*, 28:127 – 136, 2009. 56, 59

[133] J. Zhang and A.J. Morris. Fuzzy neural networks for nonlinear systems modelling. *IEE Proceedings - Control Theory and Applications*, 142:551–561(10), Nov 1995. 56

[134] H. Zhong, Q. Xiong, L. Yin, J. Zhang, Y. Zhu, S. Liang, B. Niu, and X. Zhang. Cfd-based reduced-order modeling of fluidized-bed biomass fast pyrolysis using artificial neural network. *Renewable Energy*, 152:613–626, 2020. 56

[135] O. San, R. Maulik, and M. Ahmed. An artificial neural network framework for reduced order modeling of transient flows. *Communications in Nonlinear Science and Numerical Simulation*, 77:271–287, 2019. 56

[136] J. Eftang, A. Patera, and E. Rønquist. An "hp" certified reduced basis method for parametrized elliptic partial differential equations. *SIAM J. Scientific Computing*, 32:3170–3200, Sep 2010. 56

[137] Y. Chen, J. Hesthaven, and Y. Maday. A seamless reduced basis element method for 2d maxwell's problem: An introduction. *Lecture Notes in Computational Science and Engineering*, 76:141–152, Oct 2010. 56

[138] P. Pacciarini, P. Gervasio, and A. Quarteroni. Spectral based discontinuous galerkin reduced basis element method for parametrized stokes problems. *Computers & Mathematics with Applications*, 72(8):1977–1987, 2016. Finite Elements in Flow Problems 2015. 56

[139] M. Buffoni, H. Telib, and A. Iollo. Iterative methods for model reduction by domain decomposition. *Computers & Fluids*, 38(6):1160–1167, 2009. 57

[140] S. Riffaud, M. Bergmann, C. Farhat, S. Grimberg, and A. Iollo. The dgdd method for reduced-order modeling of conservation laws. *Journal of Computational Physics*, 437:110336, 2021. 57

[141] E. Beltrami. Sulle funzioni bilineari. *Giornale di Matematiche ad Uso degli Studenti Delle Universita*, 11(2):98–106, 1873. 57

[142] K. Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901. 57

[143] L. Sirovich. Turbulence and the dynamics of coherent structures. *Quarterly of Applied Mathematics*, 5:561–590, 1987. 57, 97

[144] A. Sartori, D. Baroli, A. Cammi, D. Chiesa, L. Luzzi, R. Ponciroli, E. Previtali, M.E. Ricotti, G. Rozza, and M. Sisti. Comparison of a modal method and a proper orthogonal decomposition approach for multi-group time-dependent reactor spatial kinetics. *Annals of Nuclear Energy*, 71(0):217 – 229, 2014. 57, 58

[145] G. Jiang, H. Liu, K. Yang, and X. Gao. A fast reduced-order model for radial integral boundary element method based on proper orthogonal decomposition in nonlinear transient heat conduction problems. *Computer Methods in Applied Mechanics and Engineering*, 368:113190, 2020. 57

[146] A. Alomar, A. Nicole, D. Sipp, V. Rialland, and F. Vuillot. Reduced-order model of a reacting, turbulent supersonic jet based on proper orthogonal decomposition.

Theoretical and Computational Fluid Dynamics, pages 1–29, 2020. 58

[147] N. el Moçayd, M.S. Mohamed, D. Ouazar, and M. Seaid. Stochastic model reduction for polynomial chaos expansion of acoustic waves using proper orthogonal decomposition. *Reliability Engineering & System Safety*, 195:106733, 2020. 58

[148] M. Abbaszadeh, M. Dehghan, and I.M. Navon. A proper orthogonal decomposition variational multiscale meshless interpolating element-free galerkin method for incompressible magnetohydrodynamics flow. *International Journal for Numerical Methods in Fluids*, 92(10):1415–1436, 2020. 58

[149] D. Huang, J.N. Fuhg, C. Weißenfels, and P. Wriggers. A machine learning based plasticity model using proper orthogonal decomposition. *Computer Methods in Applied Mechanics and Engineering*, 365:113008, 2020. 58

[150] Q. Wang, J.S. Hesthaven, and D. Ray. Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem. *Journal of Computational Physics*, 384:289–307, 2019. 58

[151] O.R. San and R. Maulik. Neural network closures for nonlinear model order reduction. *Advances in Computational Mathematics*, 44(6):1717–1750, Jan 2018. 58

[152] S.E. Azam, A. Rageh, and D. Linzell. Damage detection in structural systems utilizing artificial neural networks and proper orthogonal decomposition. *Structural Control and Health Monitoring*, 26(2):e2288, 2019. e2288 STC-18-0042.R2. 58

[153] A.G. Buchan, C.C. Pain, F. Fang, and I.M. Navon. A POD reduced-order model for eigenvalue problems with application to reactor physics. *International Journal for Numerical Methods in Engineering*, 95(12):1011–1032, 2013. 58

[154] F. Wols, D. Lathouwers, and W. Uyttenhove. Transient analyses of accelerator driven systems using modal expansion techniques. *Delft University of Technology*, 2010. 58

[155] A. Sartori, A. Cammi, L. Luzzi, and G. Rozza. A multi-physics reduced order model for the analysis of lead fast reactor single channel. *Annals of Nuclear Energy*, 87:198 – 208, 2016. 58

[156] Z. Peng, Y. Chen, Y. Cheng, and F. Li. A reduced basis method for radiative transfer equation. *ArXiv*, abs/2103.07574, 2021. 58

[157] L. Soucasse, A.G. Buchan, S. Dargaville, and C.C. Pain. An angular reduced order model for radiative transfer in non grey media. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 229:23–32, 2019. 58

[158] J. Tencer, K. Carlberg, R. Hogan, and M. Larsen. Reduced order modeling applied to the discrete ordinates method for radiation heat transfer in participating media. In *Heat Transfer Summer Conference*, volume 50336, page V002T15A011. American Society of Mechanical Engineers, 2016. 58

[159] D. Huang, H. Abdel-Khalik, C. Rabiti, and F. Gleicher. Dimensionality reducibility for multi-physics reduced order modeling. *Annals of Nuclear Energy*, 110:526–540, Dec 2017. 58

[160] J. Tencer, K. Carlberg, M. Larsen, and R. Hogan. Accelerated Solution of Discrete Ordinates Approximation to the Boltzmann Transport Equation for a Gray Absorbing–Emitting Medium Via Model Reduction. *Journal of Heat Transfer*, 139(12), Aug 2017. 58

[161] S. Lorenzi. An adjoint proper orthogonal decomposition method for a neutronics reduced order model. *Annals of Nuclear Energy*, 114:245 – 258, 2018. 58

[162] C. Castagna, M. Aufiero, S. Lorenzi, G. Lomonaco, and A. Cammi. Development of a reduced order model for fuel burnup analysis. *Energies*, 13(4):890, Feb 2020. 58

[163] F. Alsayyari, M. Tiberga, Z. Perkó, D. Lathouwers, and J.L. Kloosterman. A nonintrusive adaptive reduced order modeling approach for a molten salt reactor system. *Annals of Nuclear Energy*, 141:107321, 2020. 58, 59

[164] R. Manthey, A. Knospe, C. Lange, D. Hennig, and A. Hurtado. Reduced order modeling of a natural circulation system by proper orthogonal decomposition. *Progress in Nuclear Energy*, 114:191 – 200, 2019. 58

[165] H. Gong, Y. Yu, and Q. Li. Reactor power distribution detection and estimation via a stabilized gappy proper orthogonal decomposition method. *Nuclear Engineering*

and Design, 370:110833, 2020. 58

[166] C. Ghnatios, F. Masson, A. Huerta, A. Leygue, E. Cueto, and F. Chinesta. Proper generalized decomposition based dynamic data-driven control of thermal processes. *Computer Methods in Applied Mechanics and Engineering*, 213-216:29–41, 2012. 58

[167] C.E. Heaney, A.G. Buchan, C.C. Pain, and S. Jewer. Reduced-order modelling applied to the multigroup neutron diffusion equation using a nonlinear interpolation method for control-rod movement. *Energies*, 14(5):1350, Mar 2021. 58

[168] G. Helin, C. Wei, Z. Chunyu, and C. Gong. Fast solution of neutron diffusion problem with movement of control rods. *Annals of Nuclear Energy*, 149:107814, 2020. 58

[169] T.R.F. Phillips, C.E. Heaney, P.N. Smith, and C.C. Pain. An autoencoder-based reduced-order model for eigenvalue problems with application to neutron diffusion. *International Journal for Numerical Methods in Engineering*, 122(15):3780–3811, 2021. 58

[170] T. Braconnier, M. Ferrier, J.C. Jouhaud, M. Montagnac, and P. Sagaut. Towards an adaptive pod/svd surrogate model for aeronautic design. *Computers & Fluids*, 40(1):195–209, 2011. 59

[171] L. Feng, M. Mangold, and P. Benner. Adaptive pod–deim basis construction and its application to a nonlinear population balance system. *AIChE Journal*, 63(9):3832–3844, 2017. 59

[172] J. Baiges, R. Codina, I. Castañar, and E. Castillo. A finite element reduced-order model based on adaptive mesh refinement and artificial neural networks. *International Journal for Numerical Methods in Engineering*, 121(4):588–601, 2020. 59

[173] A.G. Buchan, S.R. Merton, C.C. Pain, and R.P. Smedley-Stevenson. Riemann boundary conditions for the boltzmann transport equation using arbitrary angular approximations. *Annals of Nuclear Energy*, 38(5):1186–1195, 2011. 65