REDUCED ORDER MODELING OF NEUTRON DIFFUSION AND TRANSPORT USING PROPER GENERALIZED DECOMPOSITION

A Dissertation

by

ZACHARY MERRITT PRINCE

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Jean C. Ragusa
Committee Members,	Jim E. Morel
	Marvin L. Adams
	Bojan Popov
Head of Department,	John E. Hurtado

August 2019

Major Subject: Nuclear Engineering

ABSTRACT

This research explores the application of a reduced order modeling technique known as proper generalized decomposition (PGD) to models commonly employed in nuclear science and engineering. PGD is an *a priori* reduced order modeling technique that seeks a separated representation of a multi-dimensional variable. A separated representation involves decomposing a multi-dimensional variable into a sum of products of 1-D dimensional functions. It is conjecture that this representation can significantly reduce the burden of evaluating multi-dimensional linear systems. To investigate PGD's capability for this computational expediency and reduction in dimensionality, this research applies a PGD approach to four different types of problems: nuclear reactor criticality, multigroup neutron diffusion, neutron transport, and parameterized neutron diffusion. This dissertation first discusses the impetus of reduced order modeling and the methodology behind PGD. It then details the mathematics of the PGD algorithm and its application to several simple examples, including tailoring the algorithm for heterogeneous domains. The rest of the dissertation discusses the various new applications of this PGD approach.

In the criticality application, PGD is utilized to reduce the computational burden of evaluating multigroup neutron diffusion eigenvalue problems. In this application, each multigroup flux is sought as a finite sum of separable one-dimensional functions. With this representation, PGD is used to evaluate the linear systems within the power iteration process of the eigenvalue problem. The dissertation discusses the implementation of PGD to these eigenvalue systems including a derivation of PGD operators for multigroup neutron diffusion problems with standard power iteration and power iteration accelerated with adaptive Wielandt shift. To illustrate PGD's effectiveness, the implementation is applied to eigenvalue problems ranging from homogeneous to highly heterogeneous geometries with one-, two-, and four-group material properties. With comparison to full-order model evaluation with MOOSE, the effectiveness of PGD is found to be problem-dependent. PGD always out performs the full-order model with close to homogeneous problems, but its performance degrades with more realistic reactor problems.

In the space-energy approach, two different approaches are analyzed that utilize PGD to evaluate multigroup neutron diffusion problems, or more generally, coupled diffusion-reaction problems. This dissertation gives an overview of the PGD methodology and neutron diffusion with multigroup energy discretization. The first PGD approach performs a space-only decomposition, where a spatial separated representation is sought for each multigroup flux. The second approach is a full space-energy decomposition, where the energy dimension is included in the PGD separated representation. The dissertation also explores the prospect of performing a decomposition for different energy regions, effectively creating macro groups that retain fine-group structure. An algorithm for decomposing the linear operators to create an efficient PGD iteration process is explained for each of the approaches. The results include two 2-D, two-group examples and a 3-D seven-group example. When comparing with the full-order model, evaluated using MOOSE, both PGD approaches prove effective for mildly heterogeneous geometries, but show difficulty when dealing with more complex geometries. Furthermore, the space-energy representation is much slower than the space-only approach for the two-group problem, but proves more effective for the seven-group problem. The results also include a 145-group graphite block example, where PGD with space-energy separation significantly reduces the computational time compared to a specialized `deal.II` implementation.

In the neutron transport application, two different PGD approaches are utilized to evaluate the linear systems involved with S_N neutron transport. In the first approach, each S_N angular flux is sought as a finite sum of separable one-dimensional functions. In the second appoach, a space-angle decomposition is investigated, whereby including the angular decomposition in the separated representation. PGD has been applied extensively to advection-diffusion problems, but none that include pure advection and scattering-type variable coupling. This discussion discusses these implementations of PGD to the source iteration strategy for solving the neutron transport equation. To illustrate the effectiveness of PGD to evaluate these problems, it is applied a two-dimensional homogeneous example with a volumetric source with various scattering ratios. It is found that PGD is ineffective for pure absorption problems due to the extensive number of terms

required in the separated representation, which is verified by singular value decomposition of the full-order model. However, potential is found in utilizing PGD for problems requiring source iteration where the difference in two iterations' solution is much more separable.

In the parameterization application, a PGD approach is employed for uncertainty quantification purposes. The neutron diffusion equation with external sources, a diffusion-reaction problem, is used as the parametric model. The uncertainty parameters include the zone-wise constant material diffusion and reaction coefficients as well as the source strengths, yielding a large uncertain space in highly heterogeneous geometries. The PGD solution, parameterized in all uncertain variables, can then be used to compute mean, variance, and more generally probability distributions of various quantities of interest. In addition to parameterized properties, parameterized geometrical variations of 3D models are also considered. To achieve and analyze a parametric PGD solution, algorithms are developed to decompose the model's parametric space and semi-analytically integrate solutions for evaluating statistical moments. Varying dimensional problems are evaluated in order to showcase PGD's ability to solve high-dimensional problems and analyze its convergence.

DEDICATION

To my mother and father.

ACKNOWLEDGMENTS

First, I would like to thank my adviser Dr. Jean Ragusa. His direction, patience, and passion were the ultimate reasons for the success in my graduate study. I appreciate the fact that he brought to light the PGD method and recognized its importance for proper study in nuclear engineering. I would also like to thank my committee members Dr. Jim Morel, Dr. Marvin Adams, and Dr. Bojan Popov. Their lectures, discussions, and insights helped guide me to properly research numerical methods.

Thank you also to my fellow graduate students Logan Scott, Pablo Vaquer, Tarek Ghaddar, James Tompkins, Jijie Lou, Ian Halvic, and Logan Harbour. They made my graduate studies feel more like a team effort. I would especially like to thank Gabriella Morales for her unwavering support and adulation; she has undeniably been a benevolent constant in two of the most turbulent years of my life. Without you around, I surely would have gone insane.

Finally, and most importantly, I would like to thank my family. First and foremost my mother, Claudia Prince, you always made sure I took care of myself and thank you so much for your love and support. My father, Dan Prince, thank you for always taking interest in my studies and pushing me to always be better. My grandmother and grandfather, Jim and Judy Prince, I would not have been able to pursue my degrees if it was not for your support. My aunt, Jennifer Hall, thank you for taking of care me during my summer internship, I will never live that lavish again.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Professors Jean C. Ragusa, Jim E. Morel, and Marvin Adams and of the Department of Nuclear Engineering and Professor Bojan Popov of the Department of Mathematics.

All other work conducted for the dissertation was completed by the student independently with the direction of his adviser Dr. Jean C. Ragusa.

Funding Sources

Graduate study was supported by a fellowship from the U.S. Department of Energy's Nuclear Energy University Program.

NOMENCLATURE

AMG	Algebraic Multigrid
CFEM	Continuous Finite Element Method
CG	Conjugate Gradient
DDDAS	Dynamic Data-Driven Application System
DE	Decomposed Energy (Space-Energy Separated Representation)
`deal.II`	Differential Equations Analysis Library II
DFEM	Discontinuous Finite Element Method
DSA	Diffusion Synthetic Acceleration
FEM	Finite Element Method
FOM	Full Order Model
GMRES	Generalized Minimum Residual
HYPRE	Scalable Linear Solvers and Multigrid Methods
MG	Multigroup
MMS	Method of Manufactured Solutions
MOOSE	Multiphysics Object-Oriented Simulation Environment
MSE	Multi-Space-Energy
PGD	Proper Generalized Decomposition
PI	Power Iteration
POD	Proper Orthogonal Decomposition
QOI	Quantity of Interest
ROM	Reduced Order Model

SVD	Singular Value Decomposition
UQ	Uncertainty Quantification

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

1.1 Curse of Dimensionality

One of the most significant contributions to the complexity of computational science and engineering is the multi-dimensional nature of physical systems, or the so called “curse of dimensionality". The computational burden of solving highly-dimensional problems is especially prevalent in nuclear science and engineering. Despite the impressive advances in computer science and numerical modeling, many problems remain intractable. The following itemizes various phenomena in order of ascending difficulty:

- To simulate physics in a realistic fashion, variables often must be expressed as spatially dependent quantities. Typically this involves evaluating the variable in a three dimensional space, (x, y, z) in Cartesian geometries. If this space was discretized with M nodes in each dimension, the resulting system would involve M^3 unknowns. Therefore, if a more accurate answer is desired, a uniform refinement would increase the size of the system by a factor of eight at each refinement.

- The neutron transport equation, the gold standard for modeling neutron population behavior, includes a variable (angular flux) which lives in a phase-space of seven dimensions: three for position ($\vec{r}$), two for travel direction ($\vec{\Omega}$), one for energy (E), and one for time (t). This high-dimensional physics is often formidable to simulate with the system size easily reaching into the trillions of unknowns. Furthermore, the time dependence leads to a stiff system of equations and thus must use implicit time discretization. It is common in nuclear reactor analysis to remove the angular dependence ($\vec{\Omega}$). However, this 4-D+time neutron diffusion problem often still requires high performance computing resources to solve.

- Mathematical models used to represent physical systems often have many parameters whose values are not known exactly, but with some level of uncertainty. A technique for gathering the dependence of the solution with these parameters for uncertainty propagation is to

parameterize the solution, whereby adding the the parameters as extra dimensions to the natural dimensions, discussed in the previous points. When the number of parameters is large, the propagation of uncertainty on the system's output becomes computationally challenging; this is known as the curse of dimensionality in parametric uncertainty quantification (UQ). Using common methods like polynomial chaos, a modest uncertain space may require millions of system evaluations. Further, when the model itself is expensive to evaluate, the problem as posed may become intractable.

This list is by no means an exhaustive description of all the difficulties in modeling and simulation of nuclear science and engineering. However, it is meant to show how the complexity of problems increase dramatically as more realistic and accurate simulations are desired. The purpose of this dissertation is to investigate the potential of a reduced order modeling technique known as proper generalized decompositions to alleviate the computation expense of evaluating these problems.

1.2 Background on Reduced Order Modeling

Reduced order modeling (ROM) is a technique that aims to decrease the size of a system by extracting the relevant information into a much smaller subspace. Typically, the application of ROMs reduces the solution space *a posteriori*, that is, after a full-order (or high-dimensional) model has been exercised for a range of parameters to determine the proper subspace where the physics solution evolves. Many techniques for *a posteriori* ROM exist, one example is Proper Orthogonal Decomposition (POD) [1, 2, 3]. Some applications of POD include computational fluid dynamics [4, 5, 6], shallow water [7, 8], and mechanics [9, 10]. Some recent applications of POD for ROMs in the nuclear engineering community include reactor kinetics [11], particle transport [12], and criticality [13], for instance. In order to further illustrate the purpose of ROM, the following subsections gives a brief overview of POD with simple examples.

1.2.1 Reduced Order Modeling with Proper Orthogonal Decomposition

Suppose, after discretization, that a solution to a given physical system involves K degrees of freedom. POD finds and truncates an orthogonal basis to this solution such that the number of degrees of freedom for the truncated system is less than K. For a simple example, suppose there is a RGB image that is 1186-by-2048 pixels, which equates to a solution with $K = 7.3 \times 10^6$. To find the ROM of this image, it is restructured as a 2-D array were the columns are RGB values of the pixels in the y direction and the rows are values in the x direction, shown by the array $\boldsymbol{A}$:

$$\boldsymbol{A} = \begin{bmatrix} R_{1,1} & R_{1,2} & \dots & R_{1,2048} \\ R_{2,1} & R_{2,2} & \dots & R_{2,2048} \\ \vdots & \vdots & \ddots & \vdots \\ R_{1186,1} & R_{1186,2} & \dots & R_{1186,2048} \\ G_{1,1} & G_{1,2} & \dots & G_{1,2048} \\ G_{2,1} & G_{2,2} & \dots & G_{2,2048} \\ \vdots & \vdots & \ddots & \vdots \\ G_{1186,1} & G_{1186,2} & \dots & G_{1186,2048} \\ B_{1,1} & B_{1,2} & \dots & B_{1,2048} \\ B_{2,1} & B_{2,2} & \dots & B_{2,2048} \\ \vdots & \vdots & \ddots & \vdots \\ B_{1186,1} & B_{1186,2} & \dots & B_{1186,2048} \end{bmatrix}. \tag{1.1}$$

Singular value decomposition (SVD) is then used to find the orthogonal basis of the array, which is shown by Equation (1.2a) [14, 15]. $\boldsymbol{U}$ is the left orthonormal basis of size 1186$\times$3-by-2048, $\boldsymbol{V}$ is the right orthonormal basis of size 2048-by-2048, and $\boldsymbol{\Sigma}$ is a diagonal matrix containing the 2048 singular values. The ROM is then found by truncating the bases such that some percentage of the singular values are still present, shown by Equation (1.2b).

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\boldsymbol{T}} \tag{1.2a}$$

$$\boldsymbol{A} \approx \tilde{\boldsymbol{U}}\tilde{\boldsymbol{\Sigma}}\tilde{\boldsymbol{V}}^{\boldsymbol{T}} \tag{1.2b}$$

Assuming $\boldsymbol{\Sigma}$ is truncated to n singular values, $\tilde{\boldsymbol{U}}$ is then 1186×3-by-n and $\tilde{\boldsymbol{V}}$ is 2048-by-n. Figure 1.1 shows an example image and what the image looks like when truncating to various numbers of singular values. Figure 1.1d shows that when $n = 100$, the image is visually identical. With this truncation, the number of values in the resulting reduced order model is a tenth of the number of values in the original full-order image.

1.2.2 Proper Orthogonal Decomposition for a Physical System

A common application of POD is to produce a ROM of a physical system whose given input parameters are not known exactly. The idea is that this ROM can be evaluated at arbitrary parameter values much more quickly than the full-order counterpart. As an example, Equation (1.3) shows a discretized system with a solution $\boldsymbol{u}$, operator $\boldsymbol{L}$, and source $\boldsymbol{q}$ that depend on the parameter values $\vec{\mu}$ with K unknowns.

$$\boldsymbol{L}(\vec{\mu})\boldsymbol{u}(\vec{\mu}) = \boldsymbol{q}(\vec{\mu}) \tag{1.3}$$

The goal is to find a new operator $\tilde{\boldsymbol{L}}$ that is much smaller than $\boldsymbol{L}$ so that the solution can be evaluated more quickly for different values of $\vec{\mu}$. POD performs this model order reduction by first solving the solution by evaluating the full-order model at various parameter values or snapshots ($\vec{\mu}_i$). These solutions are then stored as an array:

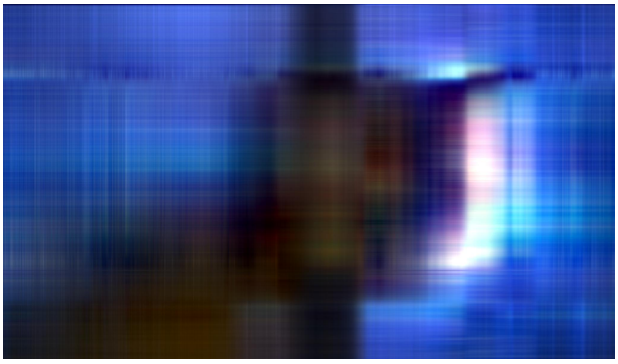
$$\boldsymbol{A} = \begin{bmatrix} u_1(\vec{\mu}_1) & u_1(\vec{\mu}_2) & \dots & u_1(\vec{\mu}_n) \\ u_2(\vec{\mu}_1) & u_2(\vec{\mu}_2) & \dots & u_2(\vec{\mu}_n) \\ \vdots & \vdots & & \vdots \\ u_K(\vec{\mu}_1) & u_K(\vec{\mu}_2) & \dots & u_K(\vec{\mu}_n) \end{bmatrix}, \tag{1.4}$$

and an orthogonal basis is found by performing SVD on matrix $\boldsymbol{A}$, shown by Equation (1.2a). The decomposition is then truncated to $\tilde{n}$ based on a percentage of the singular values, shown by Equation (1.2b), such that $\tilde{n} < \min(n, K)$. POD makes the assumption that $\boldsymbol{u}$ is a linear

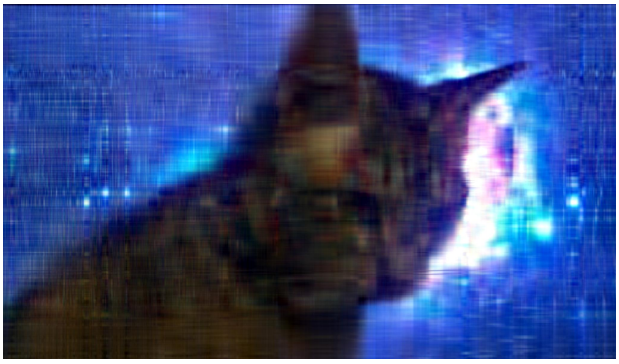
combination of the columns of $\tilde{\boldsymbol{U}}$:

$$u_i = \sum_{j=1}^{\tilde{n}} \alpha_j \tilde{U}_{ij} \ . \tag{1.5}$$

(a) Original Image



(b) 5 singular values (27%)



(c) 20 singular values (39%)

(d) 100 singular values (61%)

(e) 500 singular values (90%)

Figure 1.1: Image when performing various POD approximations

The system defined by Equation (1.3) is then projected onto the subspace described by Equation (1.5):

$$\tilde{\boldsymbol{U}}^T \boldsymbol{L}(\vec{\mu})\tilde{\boldsymbol{U}}\boldsymbol{\alpha} = \tilde{\boldsymbol{U}}^T \boldsymbol{q}(\vec{\mu}) \; . \tag{1.6}$$

This system now involves $\tilde{n}$ unknowns. This reduced order system, or model, can then be used to sample more values of $\vec{\mu}$ at a significantly reduced computational cost.

These two examples are meant to illustrate the potential of ROM, particularly POD, to reduce the computational rigor of evaluating large systems for UQ purposes. A major drawback of POD is that producing an initial basis by taking snapshots of different parameter values can be overwhelming when the number of unknowns K is very large. Contrary to POD, this research utilizes an *a priori* ROM, known as proper generalized decomposition, whereby producing a reduced basis on-the-fly to system simulation.

1.3 Introduction to Proper Generalized Decomposition

Proper generalized decomposition (PGD) aims at reducing the dimensionality of a system. To combat the difficulty in evaluating a multi-dimensional system, PGD decomposes the system solution into a separated representation [16, 17]. PGD represents a multi-variate function as a sum of products of one-dimensional functions. A D-dimensional solution is sought as N sums of D products, shown in Equation (1.7a), where N is large enough to represent the solution to a certain tolerance.

$$u(x_1, x_2, ..., x_D) = \sum_{n=1}^{\infty} \prod_{j=1}^{D} u_n^{x_j}(x_j) \approx \sum_{n=1}^{N} \prod_{j=1}^{D} u_n^{x_j}(x_j) \; . \tag{1.7a}$$

For instance, a two-dimensional variable $u(x, y)$ can be decomposed into N sums of products of functions of x and y, shown in Equation (1.7b).

$$u(x, y) = \sum_{n=1}^{\infty} X_n(x) Y_n(y) \approx \sum_{n=1}^{N} X_n(x) Y_n(y) \; . \tag{1.7b}$$

As such, the advantage of PGD is readily apparent. If a solution to a problem involves D dimensions with M nodes in each dimension, the problem will involve solving a system with M^D

unknowns. With PGD, the number of unknowns would be $N \times D \times M$. Put simply, usually a system size will increase exponentially with the number of dimensions, while PGD increases linearly. Additionally, if the system is uniformly refined, the system size increases as $2 \times D$, while an unreduced system size increases as 2^D. One of the disadvantages of PGD is that a linear problem now requires a non-linear solution process because the solution is sought as products of one-dimensional solutions.

1.3.1 Current PGD Applications

As stated in the previous section, PGD seeks a separated representation of a multi-dimensional variable. This representation is not a new concept. This form as been utilized in quantum chemistry for decades, particularly the Hartee-Fock representation of wave functions [18]. Additionally, a space-time separated representation has been utilized by Ladeveze's group in the development of the LATIN method for nonlinear structural mechanics [19, 20, 21]. However, PGD has only been developed in the past decade. Despite its adolescence, PGD has been applied to a wide range of subjects. Some of the first applications of PGD include the multi-bead-spring model of polymetric fluids [22, 23, 24]. PGD for the separation of physical space has been applied to heat flow [25, 26] and structural mechanics [27, 28]; especially interesting is the x-y/z separated representation for plate geometries [28, 29, 30, 31].

One of the hallmarks for the PGD method is its capability to parameterize models by adding uncertain parameters as extra coordinates. Thus, the dimensionality of a model includes the natural space and temporal coordinates, but also extra dimensions for each parameter under investigation. These parameters and types of models investigated using PGD vary widely across literature. Some of these applications include parametrizations of source locations [32], boundary conditions [27], geometric parameters [33], and material properties [34]. Most of these applications involve parameterizing only one or two properties, where the material properties are homogeneous. One notable exception can be found in Lamari et al. [35], where a parameterization of the heat conduction coefficient in every region of a heterogeneous domain is performed, with homogenization as the ultimate goal of their study. These parameterized models can then be utilized to perform many

useful types of analyses for physical systems. Some of these include process optimization [36], shape optimization [33, 37], inverse problems for real-time simulation [29, 32, 38, 39, 40], and uncertainty quantification [41, 42, 43].

This review is by no means a exhaustive list of current applications and developments of the PGD method. Luckily, PGD books [16, 44, 45] and review papers [46, 47, 48] have detailed lists of PGD methods and applications.

1.3.2 PGD in Nuclear Science and Engineering

PGD is a relatively new method and its applications are currently very scarce in the field of nuclear science and engineering. There are mainly two publications applying PGD to neutron diffusion models. Gonzalez et al. [49] implemented PGD to solve a 2-D, mono-energetic k-eigenvalue problem. This paper investigated the use of an Arnoldi-like algorithm within the PGD process to resolve the eigenvalue problem. Senecal & Wi [50] applied PGD using the MOOSE library (Multi-physics Object Oriented Simulation Environment) to 2-D steady-state neutron diffusion problems, including two-group diffusion. This paper is very detailed in the PGD implementation for two-group diffusion, but no generalized procedure for multigroup problems exists and only mildly heterogeneous problems have been investigated.

Beyond published manuscripts, there are some presented research on PGD application to transient diffusion and transport. Alberti & Palmer [51, 52] investigated application to space-time separation in 1-D transient diffusion, with nonlinear model considerations. Dominesey et al. [53] investigated space-angle separation for 1-D homogeneous transport problems.

1.4 Objectives of the Dissertation

The goal of this research is to tailor the PGD algorithm for application in nuclear science and engineering and analyze its capability for reducing the computational cost of evaluating these models. In achieving this goal, the PGD approach presented in this dissertation is applied to problems and governing laws that have not been investigated in previous PGD works. The following subsections describe each of these applications.

1.4.1 Multigroup Neutron Diffusion Criticality Calculations

In this application, PGD is used to evaluate the linear systems involved in the generalized eigenvalue problem for nuclear reactor criticality calculations. This application goes beyond the work by Gonzalez et al. [49] by investigating the multigroup version of the neutron diffusion problems. This extension to PGD applications is crucial for analyzing the capability of PGD to solve realistic reactor-physics problems. The research investigates the use of PGD within the standard power iteration procedure for resolving the nonlinear eigenvalue-eigenvector coupling.

1.4.2 Fixed-Source Neutron Diffusion with Space-Energy Separated Representations

In this application, PGD is used to evaluate fixed-source, steady-state multigroup neutron diffusion problems. These types of problems are subject of the work by Senecal & Wi in [50]. In their approach, each multigroup flux was separated in space-only, like in Equation (1.7b). The work presented in this dissertation compares the space-only approach with a full space-energy separation, whereby including the energy dimension in the PGD separated representation. This approach has the potential to significantly reduce the computational burden of evaluating fine-group calculations. This application applies both approaches to heterogeneous multigroup problems, where the potential of PGD to solve realistic models is analyzed.

1.4.3 Neutron Transport

As stated in Section 1.1, the neutron transport equation lives in a seven-dimensional phase space, which indicates that PGD could significantly reduce the dimensionality of these problems. In this research, the capability of PGD to serve as a reduced order model is analyzed by investigating the separability of a transport solution.

1.4.4 Parameterized Neutron Diffusion

Often parameters of a model are not known exactly, but with some degree of uncertainty. These parameters can include material properties, geometric features, boundary conditions, etc. Parametric modeling considers these uncertain parameters by expanding the model in the parameter space.

Developing parameterized models are important for uncertainty quantification (UQ), which measures the impact of the uncertainty of a system's properties. Parameterizing a model can be viewed as adding many more dimensional coordinates. Typically, the dimensionality of a model includes space and temporal coordinates. Parametric models include these natural coordinates as well as the total number of properties being parameterized. Parametric modeling is inherently high dimensional and when the full-order model is expensive to evaluate, the problems as posed may become intractable. In this work, PGD is utilized to develop a parameterized neutron flux which has unambiguous dependence on the uncertain parameters. This solution can then be used to propagate uncertainties, compute statistical moments, and perform design optimization with little computational effort.

1.5 Dissertation Structure

The chapters of this dissertation are as follows:

- Section 2 introduces the PGD solution process. This includes a generalized algorithm that can be applied to any linear system. Additionally, the chapter includes a methodology to decompose material properties for heterogeneous geometries. Several results are also included that apply PGD to the Poisson equation with method of manufactured solutions (MMS) for verification purposes. Finally, this chapter includes several diffusion-reaction examples which compare run-times and error with full-order model evaluation.

- Section 3 involves the PGD application to multigroup neutron diffusion criticality calculations. Here, the multigroup neutron diffusion equation is presented with descriptions of various power iteration techniques to evaluate the generalized eigenvalue problem. The technique in which PGD is applied to the power iteration procedure is also discussed with modifications to the underlying algorithm, including solution compression, eigenvalue projection, and adaptive enrichment tolerance. Results are presented for several reactor criticality problems and performance comparisons are made with full-order model evaluation.

- Section 4 involves the PGD application to neutron diffusion with space-energy separated

representations. This chapter presents the fixed-source, stead-state multigroup neutron diffusion equation. The PGD system construction is also presented for the additional energy dimension to the PGD solution decomposition. Results are presented for varying number of energy groups comparing the space-energy representation with space-only PGD and full-order model evaluation.

- Section 5 investigates PGD's application to neutron transport problems. This chapter introduces the mono-energetic neutron transport equation with S_N discretization of the angular dimension. Two different PGD approaches are also explained, including space-only decomposition of each direction's flux and space-angle decomposition (adding angle into the PGD separated representation). These approaches are analyzed by observing the separability of the decomposed solution and convergence of the PGD iteration scheme.

- Section 6 involves utilizing PGD to produce parametric models for neutron diffusion. This chapter introduces parametric modeling in neutron diffusion and utilizing a parameterized solution for uncertainty quantification. Application of PGD to these models is described and the unique construction of PGD operators is explained. The results illustrate the ability of PGD to evaluate very high dimensional models, otherwise impossible for full-order evaluation.

- Section 7 makes the concluding remarks on all these applications and remarks on potential future work.

2.1 PGD Solution Process

Proper generalized decomposition (PGD) is an *a priori* reduced order modeling technique, where the subspace is built on-the-fly to system evaluation. This section describes the PGD evaluation process with two approaches: variational form and matrix form. For the sake of exposition, the governing law is written as,

$$\mathcal{L}(x,y)u(x,y) = q(x,y)\ , \tag{2.1}$$

where $\mathcal{L}(x,y)$ is a linear operator acting on the solution u and $q(x,y)$ is the source term. The derivation is restricted to two dimensions for brevity, but higher dimensional systems can easily be extracted.

2.1.1 Variational Form

As an introduction to the PGD methodology, the PGD solution process is explained for a 2-D model and thus seek a separated representation of the solution in x and y, as shown in Equation (2.2a) [16]. In the PGD process, both $\mathcal{L}$ and q are decomposed in sums of one-dimensional operators and source terms:

$$u(x,y) = \sum_{n=1}^{N} X_n(x)Y_n(y)\ , \tag{2.2a}$$

$$\mathcal{L}(x,y)u(x,y) = \sum_{n=1}^{N}\sum_{\ell=1}^{L} \mathcal{L}_\ell^x(x)X_n(x)\mathcal{L}_\ell^y(y)Y_n(y)\ , \tag{2.2b}$$

$$q(x,y) = \sum_{k=1}^{Q} q_k^x(x)q_k^y(y)\ . \tag{2.2c}$$

N is again the number of terms needed to represent the solution u to a certain tolerance. L and Q are the number of terms needed to fully represent the operator and source using products of one-dimensional operators and sources, respectively. For a homogeneous diffusion-reaction operator in

2D Cartesian coordinates, L would equal 3: one for the diffusion operator in each spatial dimension and one for the absorption/reaction operator, as shown in Equation (2.3).

$$\mathcal{L}u(x,y) := -\nabla \cdot D\nabla u + \Sigma_a u = \sum_{n=1}^{N} \left[-D\frac{d^2X_n}{dx^2}Y_n - DX_n\frac{d^2Y_n}{dy^2} + \Sigma_a X_n Y_n \right] . \tag{2.3}$$

For an operator whose coefficients are dependent on material properties, L and Q are dependent on the decomposition of the heterogeneous geometry. Section 2.3.1 describes the method used for obtaining this decomposition.

The solution u is built iteratively, one term in the summation at a time, starting with the first term, $X_1(x)Y_1(y)$ all the way to $X_N(x)Y_N(y)$. This process is known as enriching the solution. For the first enrichment, the solution is assumed to be a direct separation of variables ($u(x,y) = X_1(x)Y_1(Y)$). To evaluate $X_1(x)$ and $Y_1(y)$, the separated solution is substituted into the governing law, the equation is multiplied by a test function ($v(x,y)$), and the equation is integrated over the full domain,

$$\sum_{\ell=1}^{L} \int_{\Omega_x} dx \mathcal{L}_\ell^x(x) X_1(x) \int_{\Omega_y} dy \mathcal{L}_\ell^y(y) Y_1(y) v(x,y) = \sum_{k=1}^{Q} \int_{\Omega_x} dx q_k^x(x) \int_{\Omega_y} dy q_k^y(y) v(x,y) dy \ . \tag{2.4}$$

Equation (2.4) is now a nonlinear equation between $X_1(x)$ and $Y_1(y)$. To resolve this nonlinearity, $X_1(x)$ and $Y_1(y)$ are solved separately by defining different test functions. When solving for $X_1(x)$, the test function is $v(x,y) = X^*(x)Y_1(y)$; and when solving for $Y_1(y)$, the test function is $v(x,y) = X_1(x)Y^*(y)$. Where $X^*(x)$ and $Y^*(y)$ are the respective 1-D test functions in x and y, and $X_1(x)$ and $Y_1(y)$ are the last computed solutions. Applying these test functions, the equations for $X_1(x)$ and $Y_1(y)$ are shown by Equations (2.5a) and (2.5b), respectively.

$$\sum_{\ell=1}^{L} \beta_{1,1,\ell}^y \int_{\Omega_x} X^*(x) \mathcal{L}_\ell^x(x) X_1(x) dx = \sum_{k=1}^{Q} \gamma_{1,k}^y \int_{\Omega_x} X^*(x) q_k^x(x) dx \ , \tag{2.5a}$$

$$\sum_{\ell=1}^{L} \beta^x_{1,1,\ell} \int_{\Omega_y} Y^*(y)\mathcal{L}^y_\ell(y)Y_1(y)dy = \sum_{k=1}^{Q} \gamma^x_{1,k} \int_{\Omega_y} Y^*(y)q^y_k(y)dy \,, \tag{2.5b}$$

where,

$$\beta^x_{i,j,\ell} = \int_{\Omega_x} X_i(x)\mathcal{L}^x_\ell(x)X_j(x)dx \,, \tag{2.6a}$$

$$\gamma^x_{i,k} = \int_{\Omega_x} X_i(x)q^x_k(x)dx \,, \tag{2.6b}$$

$$\beta^y_{i,j,\ell} = \int_{\Omega_y} Y_i(y)\mathcal{L}^y_\ell(y)Y_j(y)dy \,, \tag{2.6c}$$

$$\gamma^y_{i,k} = \int_{\Omega_y} Y_i(y)q^y_k(y)dy \,. \tag{2.6d}$$

Once $X_1(x)$ and $Y_1(y)$ have converged to a certain tolerance, the enrichment process continues by then evaluating $X_2(x)$ and $Y_2(y)$.

Suppose all terms up to index $N-1$ are known, and now enrichment N is being evaluated. This is done similarly as before by applying the assumed solution $u(x,y) = \sum_{i=1}^{N} X_i(x)Y_i(y)$, multiplying by the test function $v(x,y) = X^*(x)Y_N(y)$ or $v(x,y) = X_N(x)Y^*(y)$, and integrating over the full domain,

$$\sum_{\ell=1}^{L} \beta^y_{N,N,\ell} \int_{\Omega_x} X^*(x)\mathcal{L}^x_\ell(x)X_N(x)dx = \sum_{k=1}^{Q} \gamma^y_{N,k} \int_{\Omega_x} X^*(x)q^x_k(x)dx - \sum_{j=1}^{N-1}\sum_{\ell=1}^{L} \beta^y_{N,j,\ell} \int_{\Omega_x} X^*(x)\mathcal{L}^x_k(x)X_j(x)dx \,. \tag{2.7a}$$

$$\sum_{\ell=1}^{L} \beta^x_{N,N,\ell} \int_{\Omega_y} Y^*(y)\mathcal{L}^y_\ell(y)Y_N(y)dy = \sum_{k=1}^{Q} \gamma^x_{N,k} \int_{\Omega_y} Y^*(y)q^y_k(y)dy - \sum_{j=1}^{N-1}\sum_{\ell=1}^{L} \beta^x_{N,j,\ell} \int_{\Omega_y} Y^*(y)\mathcal{L}^y_\ell(y)Y_j(y)dy \,. \tag{2.7b}$$

Note that all the previously computed enrichments ($j = 1, \ldots, N-1$) appeared on the right-hand-side and are known quantities when solving for enrichment N.

To summarize, the solution process for PGD involves sequentially adding terms to the sum in Equation (2.2a), also known as enriching the solution. The first enrichment is a direct separation of variables ($u(x, y) = X_1(y)Y_1(y)$) and each dimension is solved individually using Equation (2.4). However, this is a nonlinear process because the resulting system for $X_1(x)$ depends on $Y_1(y)$, and vice versa. This nonlinear process is resolved using a fixed-point iteration or an alternating direction strategy [16]. Once resolved, a new enrichment term is added to the decomposed solution, until the contribution of the latest term is less than a certain tolerance. The PGD process is shown in Figure 2.1 for the case of a two-dimensional x-y problem. This PGD process, adapted to heterogeneous domains, is further discussed in Section 2.3.1. For additional details regarding PGD, see the following PGD books [16, 44, 45] and review papers [46, 47].

Since the PGD solution process involves two nested loops, enrichment and alternating direction, convergence criteria must be defined. This work defines the enrichment criteria as,

$$E_{\text{enr}}^{N} = \frac{||X_N(x)||\,||Y_N(y)||}{\sum_{n=1}^{N}||X_n(x)||\,||Y_n(y)||} < \epsilon_{\text{enr}} \text{ (enrichment tolerance)}, \tag{2.8}$$

and the alternating direction criteria as,

$$E_{\text{ad}}^{p+1} = \frac{||X_N^{p+1}(x) - X_N^{p}(x)||\,||Y_N^{p+1}(y) - Y_N^{p}(y)||}{||X_N^{p+1}(x)||\,||Y_N^{p+1}(x)||} < \epsilon_{\text{ad}} \text{ (fixed-point tolerance)}, \tag{2.9}$$

where p is the iteration index.

2.1.2 Matrix Form

This section describes the PGD solution process where the variables and operators are in discretized form, namely matrix form. The majority of this research utilizes continuous finite elements (CFEM) to discretize variables. For reference, Equation (2.10) shows the CFEM discretization of a 2-D variable in a full-order model on a uniform mesh.

$$u(x, y) \approx \hat{u}(x, y) = \sum_{i=1}^{I}\sum_{j=1}^{J} u_{i,j}\varphi_{i,j}(x, y)\ , \tag{2.10}$$

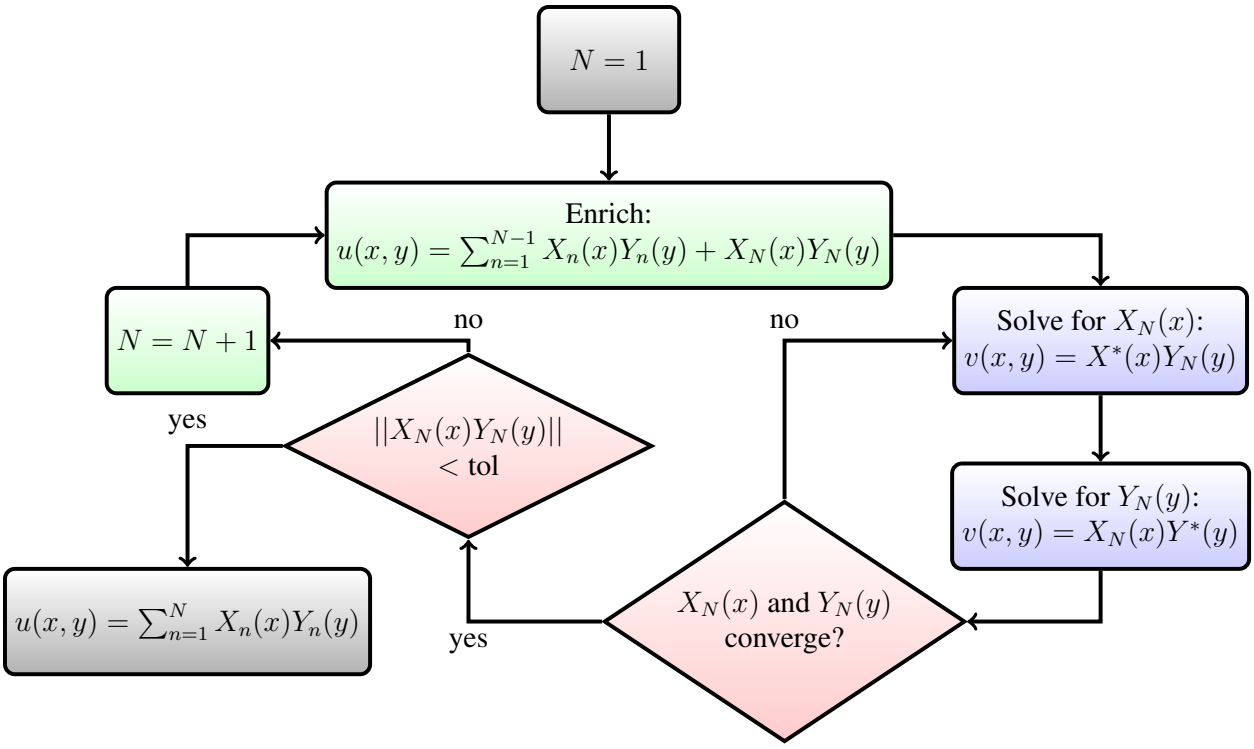


Figure 2.1: Visualization of the PGD solution process when solving a two-dimensional x-y problem

where $u_{i,j}$ are coefficients and $\varphi_{i,j}(x,y)$ are 2-D basis functions, with I and J nodes in x and y, respectively. Within the PGD framework, each term in Equation (2.2a) is discretized using 1-D CFEM:

$$X_n(x) \approx \hat{X}_n(x) = \sum_{i=1}^{I} X_{n,i}\varphi_i^x(x) \ , \tag{2.11a}$$

$$Y_n(y) \approx \hat{Y}_n(y) = \sum_{j=1}^{J} Y_{n,j}\varphi_j^y(y) \ . \tag{2.11b}$$

The coefficients in Equations (2.10) and (2.11) can be related if the PGD model is designed such that $\varphi_{i,j}(x,y) = \varphi_i^x(x)\varphi_j^y(y)$:

$$u_{i,j} = \sum_{n=1}^{N} X_{n,i}Y_{n,j} \ . \tag{2.12}$$

The relation shown by Equation (2.12) is known as projecting the PGD solution into the full-order space.

Here, it is convenient to represent the unknowns in vector form, where $u_{i,j} \rightarrow \boldsymbol{U}$, $X_{n,i} \rightarrow \boldsymbol{X}_n$, and $Y_{n,i} \rightarrow \boldsymbol{Y}_n$. The PGD and full-order model can now be related by,

$$\boldsymbol{U} = \sum_{n=1}^{N} \boldsymbol{X}_n \otimes \boldsymbol{Y}_n \ , \tag{2.13}$$

where $\otimes$ is the Kronecker product. The 2-D governing law, shown in variational form in Equation (2.1), can now be represented in matrix form:

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

with corresponding PGD representation:

$$\boldsymbol{L}\boldsymbol{U} = \left(\sum_{\ell=1}^{L} \boldsymbol{L}_\ell^x \otimes \boldsymbol{L}_\ell^y \right) \left(\sum_{n=1}^{N} \boldsymbol{X}_n \otimes \boldsymbol{Y}_n \right) = \sum_{n=1}^{N} \sum_{\ell=1}^{L} \left(\boldsymbol{L}_\ell^x \boldsymbol{X}_n \otimes \boldsymbol{L}_\ell^y \boldsymbol{Y}_n \right) \ , \tag{2.15a}$$

$$\boldsymbol{Q} = \sum_{k=1}^{Q} \boldsymbol{Q}_k^x \otimes \boldsymbol{Q}_k^y \ . \tag{2.15b}$$

For information regarding the construction of the multi-dimensional and 1-D matrices with CFEM, please see [54].

Now, regarding the evaluation of the 1-D PGD coefficient vectors, these unknowns are solved by the enrichment and alternating direction strategies described in the previous section. Equation (2.16) shows the matrix form of Equation (2.7).

$$\boldsymbol{M}_N^x \boldsymbol{X}_N = \boldsymbol{R}_N^x \ , \tag{2.16a}$$

$$\boldsymbol{M}_N^y \boldsymbol{Y}_N = \boldsymbol{R}_N^y \ , \tag{2.16b}$$

where $\boldsymbol{M}$ and $\boldsymbol{R}$ are matrices and residual vectors, respectively, defined by,

$$\boldsymbol{M}_n^x = \sum_{\ell=1}^{L} \left(\boldsymbol{Y}_n \cdot \boldsymbol{L}_\ell^y \boldsymbol{Y}_n \right) \boldsymbol{L}_\ell^x \ , \tag{2.17a}$$

$$\boldsymbol{M}_n^y = \sum_{\ell=1}^{L} \left(\boldsymbol{X}_n \cdot \boldsymbol{L}_\ell^x \boldsymbol{X}_n\right) \boldsymbol{L}_\ell^y \,, \tag{2.17b}$$

$$\boldsymbol{R}_n^x = \sum_{k=1}^{Q} \left(\boldsymbol{Y}_n \cdot \boldsymbol{Q}_k^y\right) \boldsymbol{Q}_k^x - \sum_{j=1}^{n-1} \left(\boldsymbol{Y}_n \cdot \boldsymbol{L}_\ell^y \boldsymbol{Y}_j\right) \boldsymbol{L}_\ell^x \boldsymbol{X}_j \,, \tag{2.17c}$$

$$\boldsymbol{R}_n^y = \sum_{k=1}^{Q} \left(\boldsymbol{X}_n \cdot \boldsymbol{Q}_k^x\right) \boldsymbol{Q}_k^y - \sum_{j=1}^{n-1} \left(\boldsymbol{X}_n \cdot \boldsymbol{L}_\ell^x \boldsymbol{X}_j\right) \boldsymbol{L}_\ell^y \boldsymbol{Y}_j \,. \tag{2.17d}$$

In this formulation, $\boldsymbol{M}$ and $\boldsymbol{R}$ need to be computed at each alternating direction iteration. However, if all the $\boldsymbol{L}^x$, $\boldsymbol{L}^y$, $\boldsymbol{Q}^x$, and $\boldsymbol{Q}^y$ terms are constructed before the PGD process is initiated, then the computation of $\boldsymbol{M}$ and $\boldsymbol{R}$ involves matrix-vector and vector-vector operations; very efficient algorithms have been developed for these operations [55, 56].

2.2 PGD for the Poisson Equation

This section analyzes the PGD application to the multi-dimensional Poisson equation,

$$-\Delta u(\vec{x}) = q(\vec{x}) \,, \tag{2.18}$$

where $\vec{x}$ is of arbitrary dimensionality: in two dimension $\vec{x} \equiv [x, y]$, in three dimensions $\vec{x} \equiv [x, y, z]$, and in D dimensions $\vec{x} \equiv [x_1, x_2, ..., x_D]$. The linear operators, described in Equation (2.2b), for the 2-D Poisson equation are defined as:

$$\mathcal{L}_1^x X_n(x) \mathcal{L}_1^y Y_n(y) = -\frac{d^2 X_n}{dx^2} Y_n \tag{2.19a}$$

$$\mathcal{L}_2^x X_n(x) \mathcal{L}_2^y Y_n(y) = -X_n \frac{d^2 Y_n}{dy^2} \tag{2.19b}$$

In the following subsections, PGD is applied to four different Poisson problems. The first is a 2-D and 3-D problem with a homogeneous source, where run-times are compared with full-order model evaluation. The second and third are problems that involve the method of manufactured solutions (MMS), where error convergence is verified and compared with full-order evaluation. The fourth problem analyzes PGD with D dimensional Poisson problems. The PGD process, described

in the previous section, is implemented in MATLAB where the 1-D solution terms are discretized using CFEM and the linear systems are evaluated using the MATLAB sparse LU solver. The full-order models are evaluated using the Multiphysics Object Oriented Simulation Environment (MOOSE), which utilizes a conjugate gradient solver preconditioned with algebraic multi-grid (AMG). The MOOSE solver is a HYPRE implementation with default numerical parameters.

2.2.1 Homogeneous Poisson Results

2.2.1.1 Visualization of Enrichment Procedure

To illustrate the enrichment procedure, PGD is applied to a 2-D homogeneous Poisson problem. For comparison purposes, the solution of this problem is decomposed using three different techniques. The first technique is using PGD to solve the original equation. The second technique performs SVD on the converged solution. SVD is known to produce the optimal decomposition for a 2-D array of values [57], see Appendix A for a description on how SVD is used for this purpose. The third technique uses the 2-D Fourier series representation of the exact solution,

$$u(x,y) = \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} \frac{4\left[1-(-1)^m\right]\left[1-(-1)^n\right]}{\pi^4 mn\left(m^2+n^2\right)} \sin(m\pi x)\sin(n\pi y)\ , \tag{2.20}$$

see Appendix B for the derivation of this representation. This representation produces a sum of separable functions, similar to a PGD formulation, that are orthogonal. Figure 2.2 shows the enrichment convergence of each of these techniques with the L_∞ error at each enrichment step. The contribution and L_∞ error at enrichment n are defined as,

$$\text{Contribution}(n) = \frac{||X_n||\,||Y_n||}{\sum_{i=1}^{n}||X_i||\,||Y_i||}\ , \tag{2.21a}$$

and,

$$L_\infty \text{ Error} = \max_{x\in\Omega_x\,,y\in\Omega_y}\left|\sum_{i=1}^{n} X_i(x)Y_i(y) - \sum_{i=1}^{N} X_i(x)Y_i(y)\right|\ , \tag{2.21b}$$

respectively. From Figure 2.2, PGD is far from the optimal decomposition of SVD and the convergence is not uniform. However, the error between each enrichment decreases, which indicates that

each enrichment term contributes to reducing the error. Furthermore, both PGD and SVD converge much more quickly than the analytical Fourier series, which implies that orthogonal decomposition does not necessarily provide the best decomposition. Figures 2.3a and 2.3b show the first six enrichment terms of PGD for x and y, respectively.

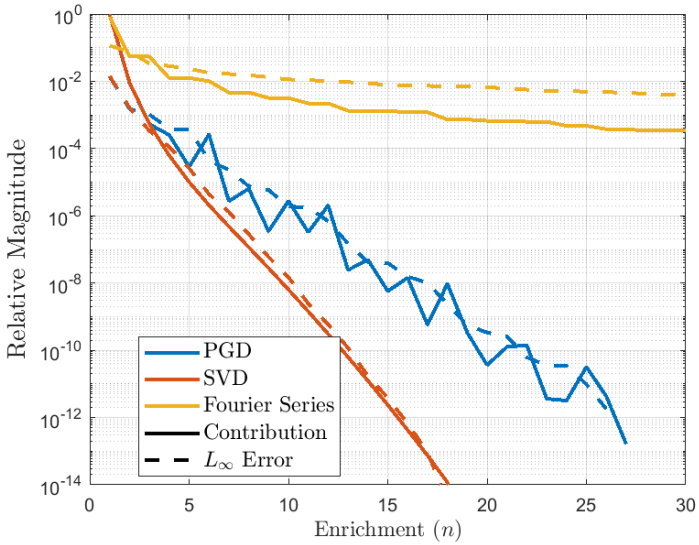
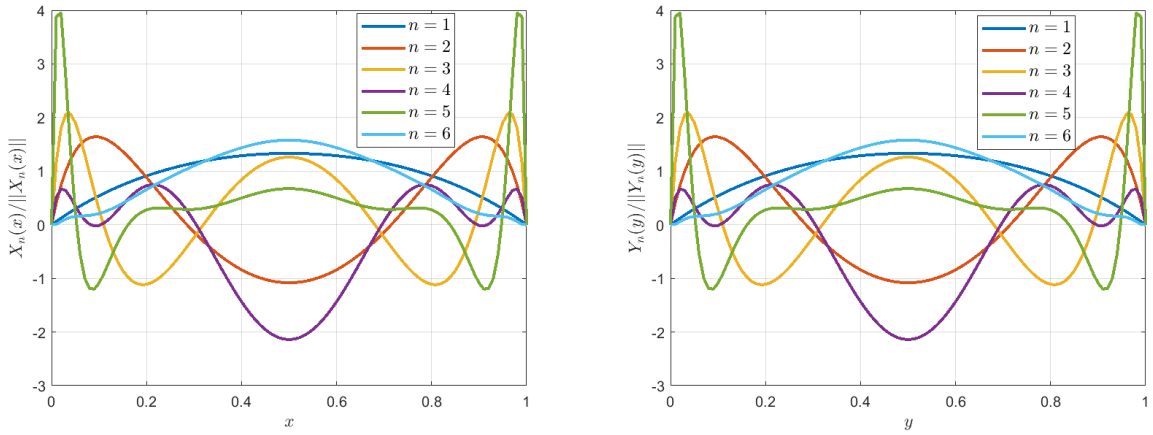


Figure 2.2: Enrichment convergence for PGD, SVD, and the Fourier series representation of the exact solution to the 2-D homogeneous Poisson equation.

2.2.1.2 Speed Comparison with MOOSE

In order to show the computational efficiency of PGD, a homogeneous two- and three-dimensional Poisson problem was executed with PGD and MOOSE with various mesh sizes. The linear tolerance in MOOSE and PGD enrichment tolerance were set to 10^{-6}. Table 2.1 shows the timing results for each mesh. These results illustrate how PGD can significantly reduce the computational effort in evaluating multi-dimensional problems. Additionally, the increase in timing confirms that PGD scales linearly with number of elements in each dimension ($I+J$), while the full-order model scales by the power of the number of dimensions ($I \times J$).



(a) $X_\ell(x)$ (b) $Y_\ell(y)$

Figure 2.3: First six 1-D terms in PGD solution, scaled by their L_2 norm.

Table 2.1: Homogeneous Poisson problem execution times

Elements per Dimension	Execution Time (sec)			
	2-D		3-D	
	PGD	MOOSE	PGD	MOOSE
100	0.0229	0.2649	0.3135	78.662
200	0.0447	1.0974	0.3309	653.08
400	0.0415	4.3070	0.4705	-
800	0.0620	20.392	0.8091	-
1,600	0.0968	81.285	1.4265	-
3,200	0.1827	324.05	1.6930	-
6,400	0.3613	-	4.1863	-
12,800	0.7810	-	11.677	-
25,600	2.1407	-	29.493	-
51,200	6.1579	-	56.872	-
102,400	17.491	-	112.95	-

Results with "-" could not be computed, system did not have enough memory

2.2.2 Method of Manufactured Solutions Results

In order to verify the PGD implementation and compare error convergence with full-order modeling, two different manufactured solutions are constructed and executed with PGD and MOOSE. The first is shown by Equation (2.22a), which can analytically be decomposed into two terms.

The second is shown by Equation (2.22b), which cannot be expressed by a finite sum of separable functions.

$$u(x,y) = \sin(x\pi)y^2(1-y) + \sin(y\pi)x^2(1-x) \tag{2.22a}$$

$$u(x,y) = \sin(xy\pi)\sin(x\pi)\sin(y\pi) \tag{2.22b}$$

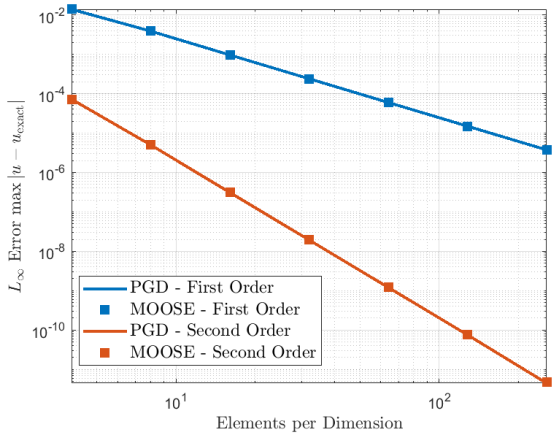
Figures 2.4a and 2.5a shows the L_∞ error as a function of mesh refinement for the first and second manufactured solutions, respectively. These plots show that the PGD implementation has proper error convergence for first and second order CFEM, even with an inseparable reference solution. Figures 2.4b and 2.5b show the enrichment convergence for the PGD solution at 256 elements per dimension. These plots also show the relative magnitude of the singular values, obtained from singular value decomposition (SVD), of the MOOSE solution and the exact solution projected on the same mesh. SVD of the first MMS shows the separability of the equation, where both the MOOSE and exact decompositions converge to approximately machine precision after the second term. However, PGD needs many more enrichments to converge. SVD of the second MMS shows the inseparability of the resulting solution. Additionally, PGD has more difficulty converging, compared to the first MMS, which indicates that the performance of PGD is dependent on the separability of the solution itself.

2.2.3 Results for Poisson with Varying Dimensionality

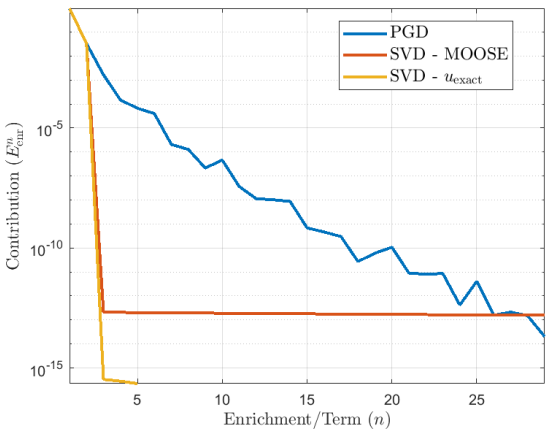
In this section, PGD is applied to the Poisson equation with various number of dimensions. The purpose of this analysis is to observe how PGD scales with dimensionality. For the sake of exposition, the Laplacian operator (Δ) in D dimensions is expressed as:

$$\Delta = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \cdots + \frac{\partial^2}{\partial x_D^2} \,. \tag{2.23}$$

The problem was executed with PGD for 2, 3, 4, 5, 10, and 20 dimensions with 100 elements in each dimension. Included is an 11 dimension result which has 144 elements in 10 dimensions and

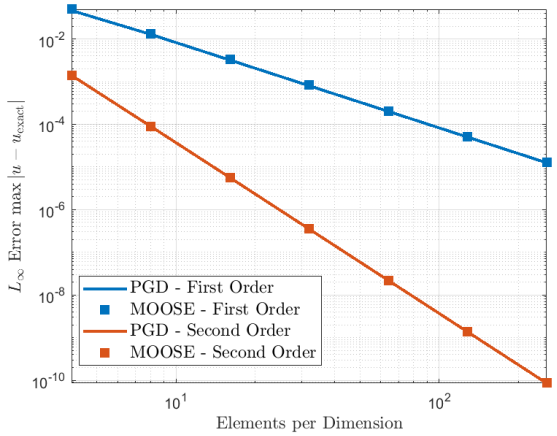


(a) L_∞ Error

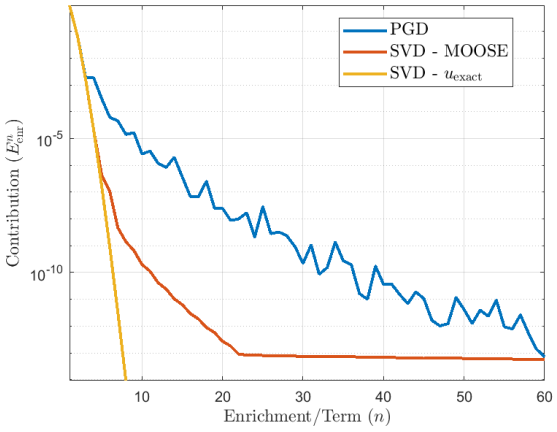


(b) Enrichment convergence

Figure 2.4: Error and enrichment convergence of first Poisson manufactured solution (Equation (2.22a)



(a) L_∞ Error



(b) Enrichment convergence

Figure 2.5: Error and enrichment convergence of first Poisson manufactured solution (Equation (2.22b)

157 in the other. Table 2.2 details the results of each of PGD simulations including run-times, the number of enrichments required for a tolerance of 10^{-5}, and a comparison between the size of the PGD solution and the expanded full-order solution. These results show that PGD does not scale linearly with the number of dimensions, due to the increase in the number of enrichments required

when the dimensionality is increased. However, difference between the FOM and PGD solution size is improved significantly at each increase in dimensionality.

Table 2.2: Results for PGD execution with varying dimensionality of the Poisson equation

Number of Dimensions	Run-time (s)	Enrichments	PGD DOF	FOM DOF
2	0.0176	7	1.4e+03	10^4
3	0.129	17	5.1e+03	10^6
4	0.427	35	1.4e+04	10^8
5	1.71	63	3.15e+04	10^{10}
10	1.3e+03	722	7.22e+05	10^{20}
11	1.41e+04	1304	2.08e+06	6.02×10^{23}
20	5.85e+04	1760	3.52e+06	10^{40}

2.3 PGD for Diffusion Reaction

This section discusses the PGD application for diffusion reaction problems. The purpose of this application is two-fold. Diffusion-reaction problems involve heterogeneous material properties, which gives an opportunity for analysis of PGD performance in heterogeneous materials. Additionally, this application serves as a first step toward application to neutron diffusion, which is a type of diffusion-reaction problem. The diffusion-reaction equation is defined as,

$$-\nabla \cdot D(\vec{r})\nabla\phi(\vec{r}) + \Sigma_a(\vec{r})\phi(\vec{r}) = Q(\vec{r}) \ , \tag{2.24}$$

where D is the diffusion coefficient (cm), Σ_a is the reaction coefficient or the absorption cross section in neutron diffusion (cm^{-1}), and Q is the source ($\text{n}\cdot\text{cm}^{-3}\cdot\text{s}^{-1}$). In neutron diffusion, the solution ϕ is known as the neutron flux ($\text{n}\cdot\text{cm}^{-2}\cdot\text{s}^{-1}$).

In the following subsections, the PGD operator construction is discussed where heterogeneous properties are considered, namely piece-wise constant gridded domains. Furthermore, results are presented for several diffusion-reaction problems in nuclear reactor-like geometries and an analysis is performed for PGD performance in varying amounts of heterogeneity.

2.3.1 Spatial Decomposition of Material Properties

This section discusses how to decompose an operator that contains a spatially dependent material property, such that it conforms to separated form in Equation (2.2b). Suppose that a linear operator is defined as a material constant ($k(x,y)$) times the solution ($u(x,y)$). For diffusion reaction, this material constant represents $D(x,y)$, $\Sigma_a(x,y)$, and $Q(x,y)$. In order to find the separated form of the operator, the property must be decomposed as,

$$k(x,y) = \sum_{\ell=1}^{L_k} k_\ell^x(x) k_\ell^y(y) \ . \tag{2.25}$$

With this description of the material constants, the PGD operator, Equation (2.2b), can be expressed as,

$$\mathcal{L}_\ell^x X_i \mathcal{L}_\ell^y Y_i = \frac{d}{dx}\left[D_\ell^x(x)\frac{dX_i}{dx}\right] D_\ell^y(y) Y_i, \ \ell = 1,..,L_d \ , \tag{2.26a}$$

$$\mathcal{L}_{\ell+L_d}^x X_i \mathcal{L}_{\ell+L_d}^y Y_i = D_\ell^x(x) X_i \frac{d}{dx}\left[D_\ell^y(y)\frac{dY_i}{dy}\right], \ \ell = 1,..,L_d \ , \tag{2.26b}$$

$$\mathcal{L}_{\ell+2L_d}^x X_i \mathcal{L}_{\ell+2L_d}^y Y_i = \Sigma_\ell^x(x) X_i \Sigma_\ell^y(y) Y_i, \ \ell = 1,...,L_\sigma \ , \tag{2.26c}$$

where L_d and L_σ are the number of terms required to represent D and Σ_a in a separated representation, respectively. Consequently, $L = 2L_d + L_\sigma$ from Equation (2.2b), and more generally in $\texttt{dim}$ spatial dimensions, $L = \texttt{dim} L_d + L_\sigma$.

In reactor physics, it is common for material properties to be defined in material zones, i.e. piecewise constant for separate material zones. Obtaining the decomposed spatial description of a given material, $k(x,y)$, requires a general decomposition of the property zones. In two dimensions, this entails finding slices in one dimension that are common in the other dimension. Each product of the decomposition will have one dimension describing the quantities of the slice, where the other describes the position of the slice. For instance, if there is a domain describing the cross section shown in Figure 2.6, then the resulting material decomposition is shown in Equation (2.27), where $H(x)$ is the Heaviside step function.

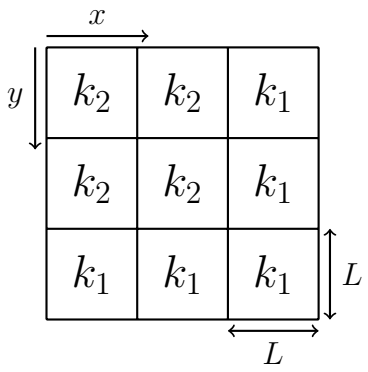


Figure 2.6: Domain showing property values for an example of property decomposition

$$K_1^x(x) = H(2L - x) \ , \tag{2.27a}$$

$$K_1^y(y) = k_2 H(2L - x) + k_1 H(x - 2L) \ , \tag{2.27b}$$

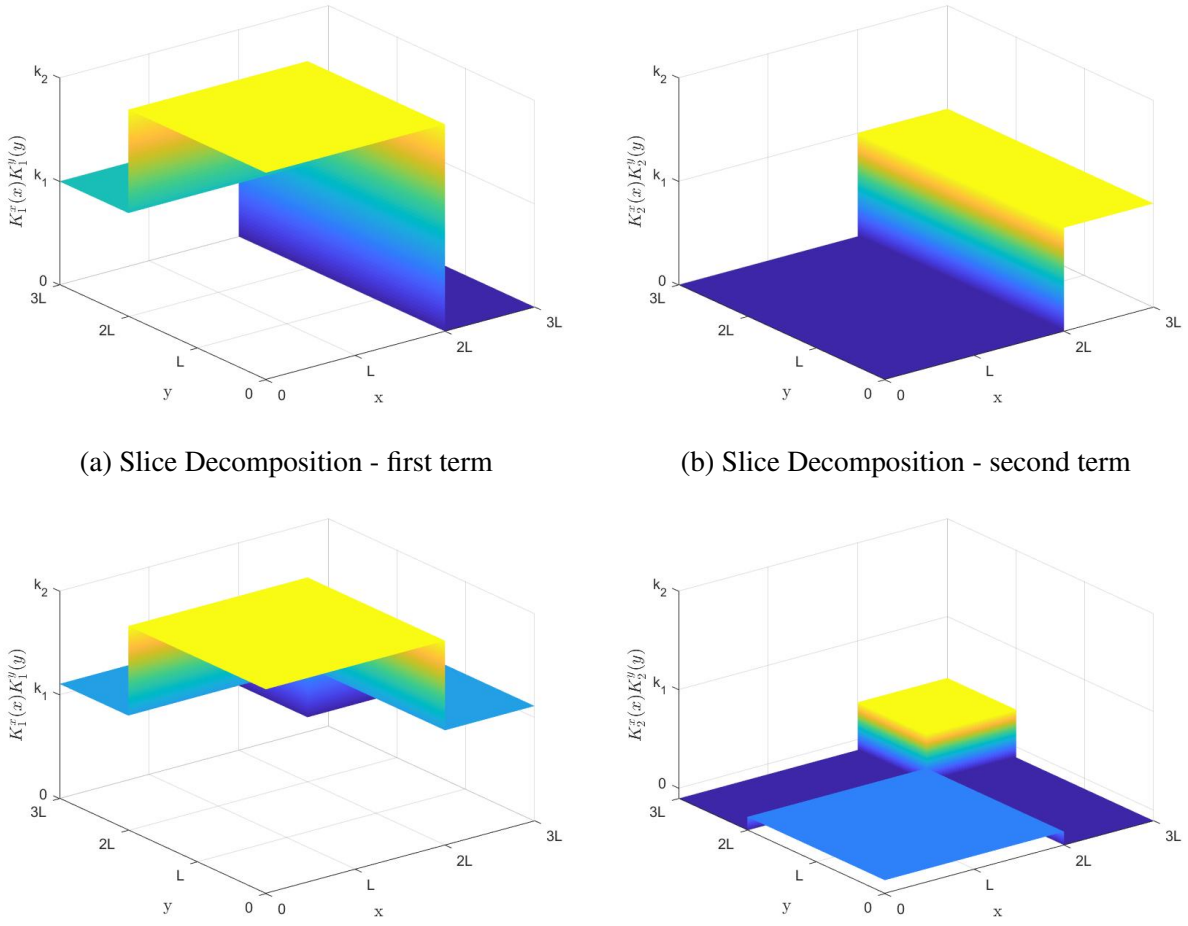
$$K_2^x(x) = H(x - 2L) \ , \tag{2.27c}$$

$$K_2^y(y) = k_1 \ . \tag{2.27d}$$

In three-dimensional geometries, common planes are found for a given dimension; then, the same two-dimensional decomposition is applied to each of such planes, resulting in a product of the two-dimensional decomposition and a function describing the position of the plane. This process is general for any given set of material properties.

The slice decomposition shown by Equation (2.27) is not the only possible choice when determining a separated representation for a material property. Other approaches include (1) singular value decomposition (SVD) and (2) PGD decomposition of the material property. As an example, the resulting decompositions for each of these three approaches in Figure 2.7 for the material layout of Figure 2.6. However, the slice-decomposition approach can yield a significant advantage when integrating the decomposed operators (right-hand-side of Equation (2.7)); indeed, computing these inner products is computationally cheaper due to the sparse nature afforded by the use of

step/Heaviside functions. When many enrichment terms are needed, these integrations take up a sizable amount of time to evaluate. Additionally, decomposing a 3-D heterogeneous domain is not as simple when employing SVD and is prone to round-off errors when using a PGD process, thus this implementation prefers the slice-decomposition process presented here.



(a) Slice Decomposition - first term

(b) Slice Decomposition - second term

(c) SVD/PGD - first term

(d) SVD/PGD - second term

Figure 2.7: Visualization of terms from decomposition of material properties using analytical slices, SVD, and PGD methods. The SVD and PGD methods resulted in the exact same decomposition.

2.3.2 2-D Heterogeneous Diffusion Reaction Results

In this section, the PGD algorithm and the material property decomposition is tested for a heterogeneous diffusion reaction problem in a reactor-like geometry. The chosen problem geometry is based on the IAEA benchmark from the ANL Benchmark Problem Book problem 11-A1 [58]. The 2-D geometry is a cross-section of the core between 20-100 cm along its height where the reflector is extended to make the domain rectangular, shown in Figure 2.8. The material properties are described by Table 2.3. To illustrate the material property decomposition used, Figure 2.9 shows each 1-D function in the decomposition of Σ_a and the resulting slices each product produces.

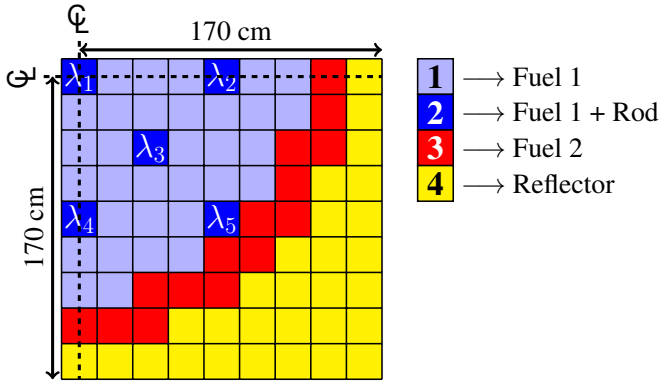
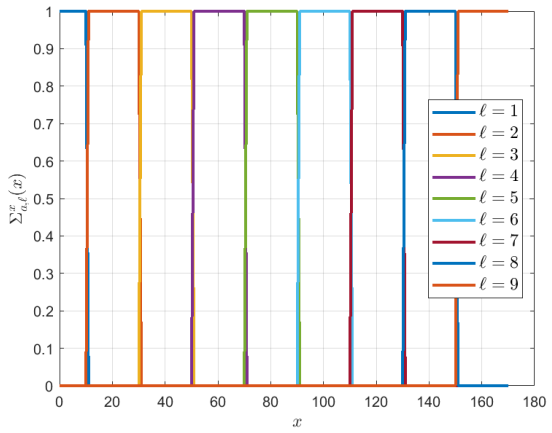


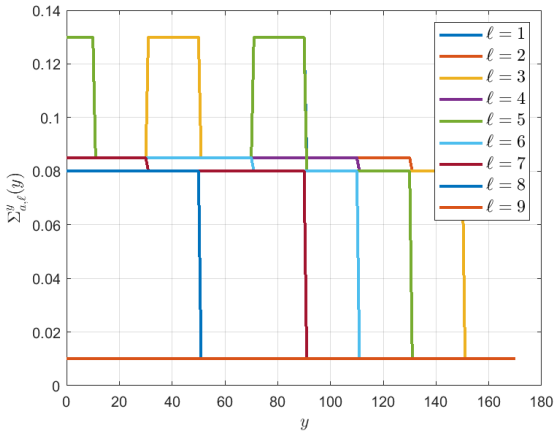
Figure 2.8: Geometry of 2-D IAEA Benchmark problem

Table 2.3: Material properties of 2-D IAEA Benchmark problem

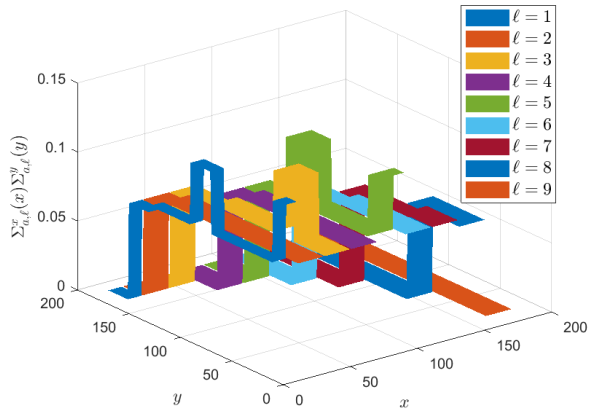
Region	Material	D	Σ_a	Q
1	Fuel 1	0.4	0.085	10
2	Fuel 1 + Rod	0.4	0.13	10
3	Fuel 2	0.4	0.08	10
4	Reflector	0.3	0.01	0
5	Reflector + Rod	0.3	0.055	0



(a) $\Sigma^x_{a,\ell}(x)$



(b) $\Sigma^y_{a,\ell}(y)$



(c) $\Sigma^x_{a,\ell}(x)\Sigma^y_{a,\ell}(y)$

Figure 2.9: Visualization of terms from decomposition of $\Sigma_a(x, y)$ using slice technique for the IAEA 2-D problem

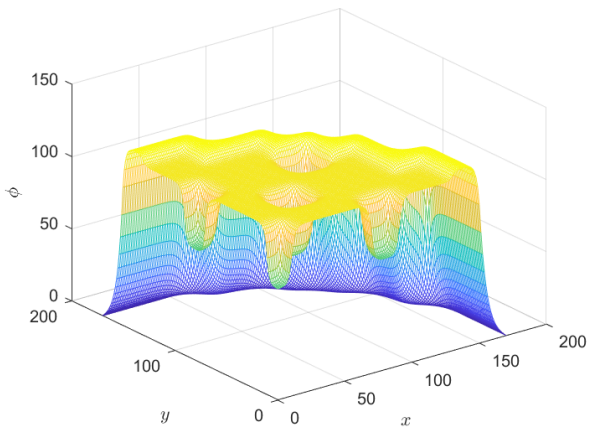
To illustrate the performance of PGD, the 2-D model was executed with various spatial refinements. The same configurations were solved using standard multi-dimensional FEM techniques in MOOSE as well. The MOOSE linear tolerance and PGD enrichment tolerance were set to 10^{-6}. Table 2.4 shows the execution times for the various spatial resolutions and the corresponding L_∞ error computed against a highly refined reference solution, obtained using MOOSE. These results show that PGD performance is poor on the coarse mesh, but surpasses the full-order model evaluation at high refinements because the PGD run-time increase linearly with the number of elements per dimension, while MOOSE increases quadratically. Figure 2.10 shows the resulting solution

and relative error at each nodal point for the 2-D problem. The error plot shows that much of the difference between the MOOSE and PGD occurs at the material discontinuities, indicating the PGD model has difficulty resolving the heterogeneous regions.

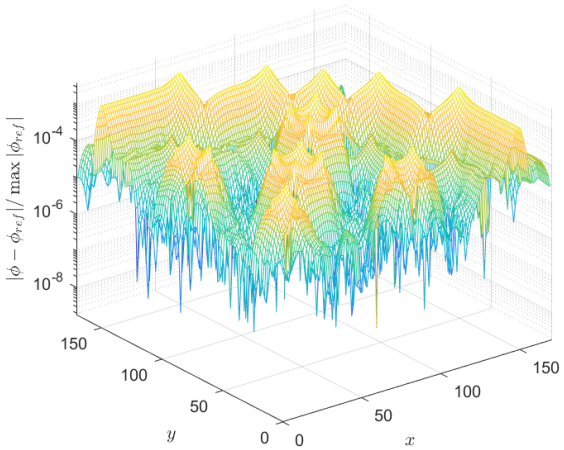
Table 2.4: 2-D IAEA benchmark problem execution times and error comparison

Elements per Dim. (x, y)	Execution Time (s)		Relative L_∞ Difference $\max\|\phi - \phi_{ref}\| / \max\|\phi_{ref}\|$	
	PGD	MOOSE	PGD	MOOSE
(34,34)	1.1104	0.1444	4.078E-2 (100)	4.081E-2
(68,68)	2.3357	0.1975	1.498E-2 (121)	1.483E-2
(136,136)	3.1999	0.7727	3.941E-3 (112)	3.917E-3
(272,272)	6.0488	3.1535	1.021E-3 (106)	9.981E-4
(544,544)	10.358	13.604	2.862E-4 (102)	3.431E-4
(1088,1088)	51.120	230.60	1.422E-4 (94)	

$(\cdot)$ Indicates the number of enrichment terms needed



(a) PGD solution



(b) Difference from MOOSE model

Figure 2.10: Solution and node-by-node relative difference of 2-D heterogeneous problem with (136,136) elements.

2.3.3 3-D Heterogeneous Diffusion Reaction Results

In this section, PGD is applied to the 3-D version of the IAEA benchmark problem described in the previous section. The 3-D extrusion into the z dimension is shown by Figure 2.11 and the material properties are desribed by Table 2.3.

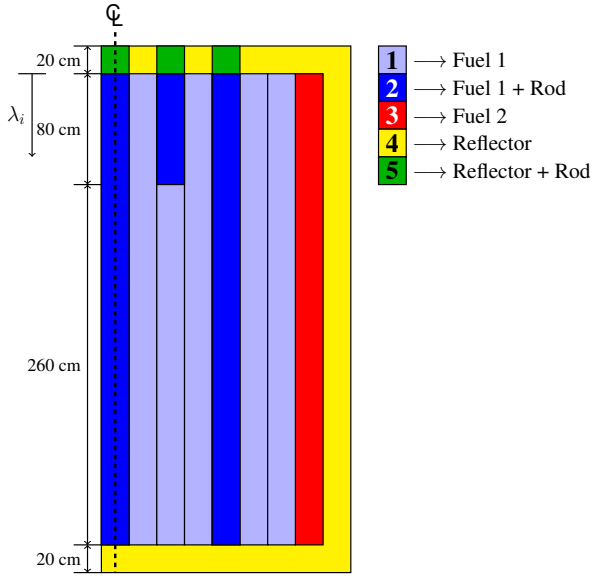


Figure 2.11: Axial geometry for the IAEA 3-D Benchmark

Again, the 3-D model was executed with PGD and MOOSE with various spatial refinements. The MOOSE linear tolerance and PGD enrichment tolerance were set to 10^{-6}. Table 2.5 shows the execution times for the various spatial resolutions for the 3-D problem. PGD again performs worse than MOOSE for coarse meshes, but significantly improves as the mesh is refined. MOOSE, as expected, scales cubically with the number of elements per dimension, while PGD scales approximately linearly, slightly better because less enrichments are required at higher refinements. Additionally, the number of enrichments for the 3-D version is significantly higher than the 2-D version, which is expected based on the results from Section 2.2.3.

Table 2.5: 3-D IAEA benchmark problem execution times

Elem. per Dim. (x, y, z)	Execution Time (s)		No. of PGD Terms
	PGD	MOOSE	
(34,34,38)	157.22	5.6114	350
(68,68,76)	250.40	49.476	339
(136,136,152)	365.22	403.79	311
(272,272,304)	555.22	$\sim$3200	289
(544,544,608)	770.66	$\sim$25600	252

Results with "$\sim$" could not be computed, system did not have enough memory. Values are extrapolated based on AMG scaling properties.

2.3.4 Study of PGD in Heterogeneous Domains

Throughout the results presented thus far, PGD seems to require more terms in the separated representation (more enrichments) when solving models with heterogeneous domains. This can easily be seen when comparing the number of enrichments required for 2-D homogeneous Poisson problem and the 2-D diffusion-reaction problem. This section analyzes the effect heterogeneity has on PGD. For this study, two different parameters are investigated: contrast of the material properties and the degree of heterogeneity. Two different problem types are investigated. The first involves only two material zones in a checkerboard patter, similar to geometries seen in reactor physics applications, and is meant to study effect of contrast only. The second takes into account the degree of heterogeneity and contrast.

2.3.4.1 Study of Contrast on a Checkerboard Domain

This study involves a five-by-five checkerboard domain shown by Figure 2.12. Two different parameters are varied in this experiment: Σ_a and the diffusion length window. Diffusion length is defined as $L_d = \sqrt{D/\Sigma_a}$. In this experiment, the efect of the variation in D and Σ_a are studied independently. The variation of these properties in each zone is defined by a factor κ, for instance, when varying D,

$$D_1 = \kappa \, , \quad D_2 = \frac{1}{\kappa} \, , \tag{2.28a}$$

and when varying Σ_a,

$$\Sigma_{a,1} = \frac{1}{\kappa}\,, \quad \Sigma_{a,2} = \kappa\,. \tag{2.28b}$$

From these definitions, it is apparent that when $\kappa = 1$, the domain is homogeneous and when $\kappa = 10$, the property varies by a factor of 100. When varying D or Σ_a, the counterpart is homogeneous with a value defined by a set maximum diffusion length ($L_{d,\max}$).

Figures 2.13a and 2.13b show the number of enrichments required for PGD process to reach an enrichment tolerance of 10^{-6} with varying D and Σ_a, respectively. From these plots, it is immediately apparent that having any type of heterogeneity affects the number of required enrichments significantly. For the variation in D, the magnitude of κ after 1 has very little impact, while a more diffusive domain requires more enrichments. This observation indicates that a smoother transition within the material discontinuities is more difficult to capture with a separated representation, but the inflection at the boundaries of the discontinuities is inconsequential. For the variation in Σ_a, at small diffusion lengths, the the variation does not affect the separability of the solution. This observation indicates that when a solution has sharp transitions (visually blocky), the difference in the solution peaks does not affect the separability. However, when the solution is smooth at higher diffusion lengths, this magnitude affects separability significantly.

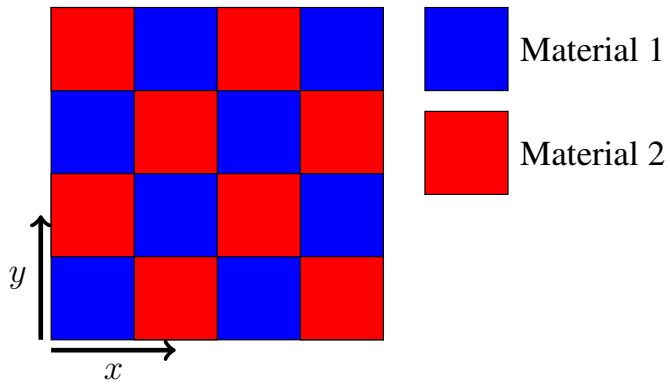
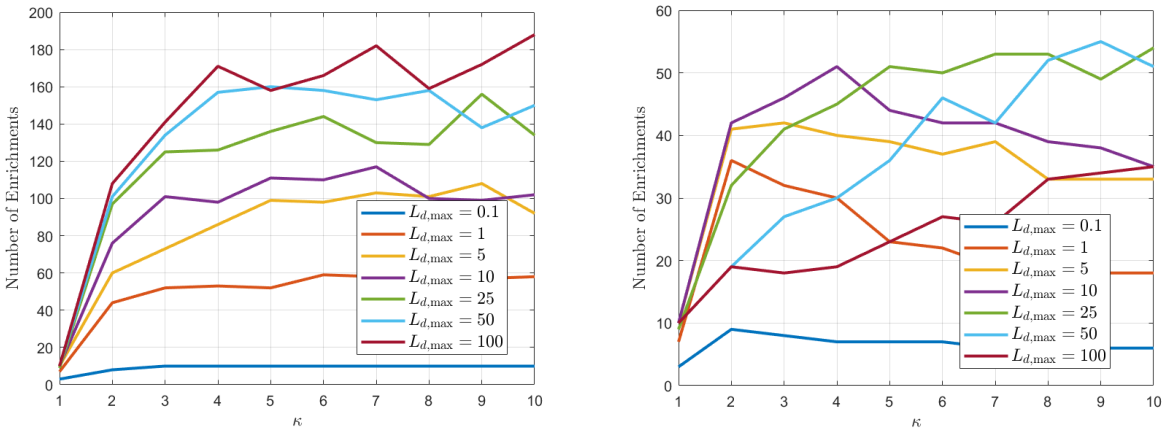


Figure 2.12: 100-by-100 cm checkerboard geometry, all boundaries are set such that $\phi = 0$.



(a) Varying D

(b) Varying Σ_a

Figure 2.13: Number of enrichments required for varying contrast in D and Σ_a with various diffusion length windows.

For a simpler look, the domain was decreased to 4-by-4 cm, where each block is 1-by-1 cm. D and Σ_a are varied independently, with definitions described by Equation (2.28). When one property is heterogeneous, the other is homogeneous with a value of 1. Figure 2.14 shows the resulting number of enrichments required for various values of κ. This result shows the significant difference in effect that D and Σ_a have on the separability of the solution.

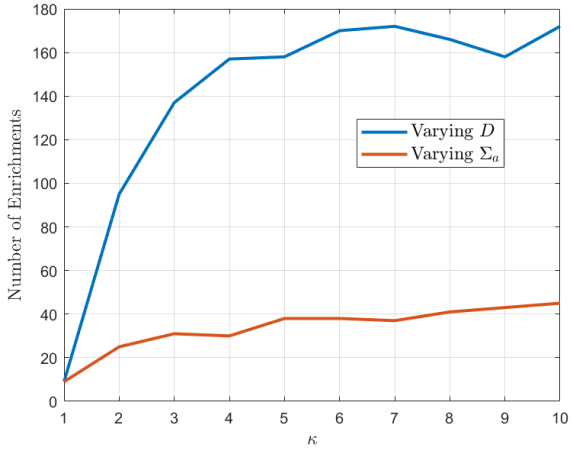


Figure 2.14: Number of enrichments required for varying contrast in D and Σ_a.

2.3.4.2 *Study of Degree of Heterogeneity and Contrast*

This study investigates the effect of the degree of heterogeneity as well as the contrast in material properties. The degree of heterogeneity is defined as simply the number of different material zones. For simplicity, this study only involves a 2-D geometry shown by Figure 2.15 with varying degrees of heterogeneity. The contrast in the material properties is based on the ratio of the maximum and minimum values of the property between the material zones:

$$\text{Contrast} = \frac{k_{\max}}{k_{\min}} - 1 \, . \tag{2.29}$$

The value of the property in each zone is picked randomly between the maximum and minimum values. To reduce uncertainty in the results, each configuration is run 25 times with different randomly selected material properties.

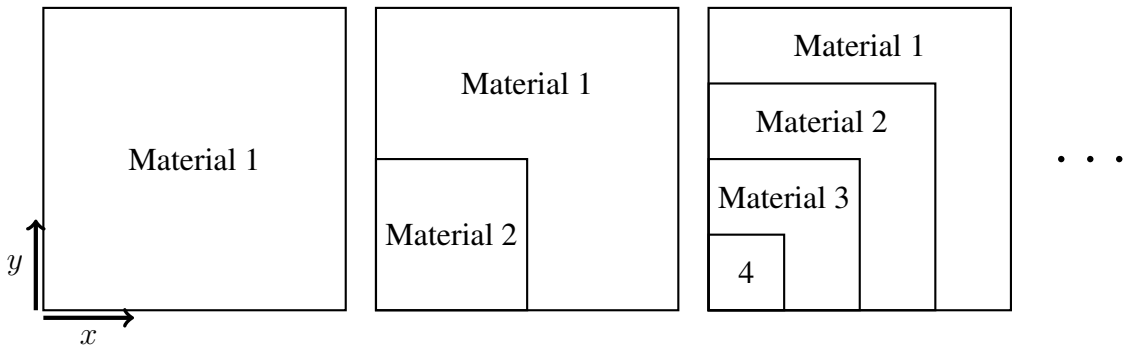
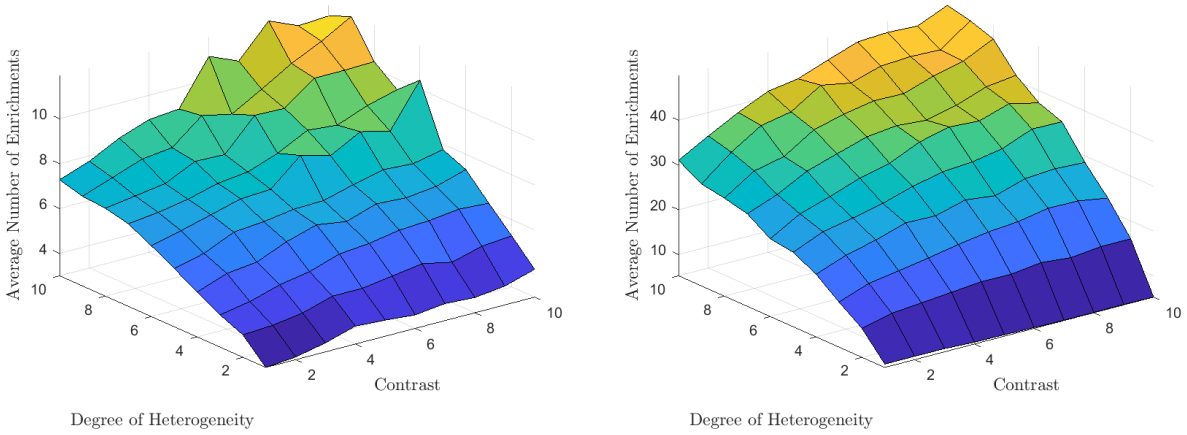


Figure 2.15: 2-D geometry with varying degrees of heterogeneity: 1, 2, 4, ... The geometry is 100-by-100 cm with reflecting boundaries along the x and y axis.

For this numerical experiment, the effect of D and Σ_a are studied independently. For D, $D_{\min} = 0.2$ where Σ_a and Q are homogeneous with values of 0.1 and 10, respectively. For Σ_a, $\Sigma_{a,\min} = 0.02$ where D and Q are homogeneous with values of 0.5 and 10, respectively. The quantity of interest is the number of enrichments required for an enrichment tolerance of 10^{-6}. Figure 2.16 shows this quantity as a function of the degree of heterogeneity and contrast for D

and Σ_a. It is apparent from these results that heterogeneous domains have a significant effect on PGD performance. The number of materials have a larger effect than the contrast, but the impact of contrast increases when more materials are present. Furthermore, having Σ_a be heterogeneous is much more impactful than having varying D. If a full-order model was executed on these models, there would be no difference in system size between each configuration. However, the configuration could affect the PGD system size by a factor of more than 50.



(a) D is heterogeneous

(b) Σ_a is heterogeneous

Figure 2.16: Number of enrichments required for varying degrees of heterogeneity and contrast. Each point is the average of 25 different property values.

2.4 Discussion

This chapter gives an overview of the PGD methodology and the solution process for partial differential equations. The theory of the PGD solution process is generalized for any given linear operator and the discretized formulation includes an methodology to maximize the efficiency of the process. Included are results that apply PGD to the Poisson equation with varying dimensionality and two manufactured solution. Additionally, PGD is applied to a diffusion-reaction equation, which serves as a first step towards applying PGD to neutron diffusion problems.

Even from these preliminary results, it is evident that PGD does not scale linearly with dimensionality and exhibits poor performance in heterogeneous domains. Although PGD needs significantly more enrichments to resolve higher dimensional problems, the reduction in solution size at high dimensions shows PGD is still a highly effective reduced order model. The issue of heterogeneity for PGD could prove problematic for reactor-physics applications, since the geometries involved are highly heterogeneous with a myriad of different materials. However, the run-time comparisons with the reactor geometries shows PGD has promise when the mesh is highly refined.

3. APPLICATION OF PROPER GENERALIZED DECOMPOSITION TO MULTIGROUP NEUTRON DIFFUSION EIGENVALUE CALCULATIONS

3.1 Introduction

Despite the advances in computer architectures and numerical analysis, eigenvalue calculations for full-core nuclear reactors are still CPU-intense task, even more so when in the context of core reload optimization. The computational burden in these computations is partly due to the dimensionality of the problem, or the size of the phase space. A neutron diffusion eigenvalue problem involves a four dimensional phase-space, three for space and one for energy.

The multigroup neutron diffusion criticality equations form an eigenvalue problem where the scalar flux ($\Phi \equiv [\phi^1, ..., \phi^G]$) is the eigenvector and the neutron multiplication factor (k) is the eigenvalue [59, 60]. Being an eigenvalue problem, evaluating ϕ and k requires a nonlinear process. A standard approach to evaluating the eigenvalue problem is power iteration. However, the convergence rate of power iteration is directly related to the dominance ratio (the ratio of second largest to the first largest eigenvalue), which can be close to unity for nuclear core problems [61, 62]. Therefore, acceleration techniques are often employed, including utilizing Chebyshev polynomials [63, 64, 65] or Wielandt shift [66, 67, 68]. Projection-based approaches, like the Arnoldi algorithm, are also commonly employed, particularly the implicit restarted Arnoldi method (IRAM) [69, 70, 71]. Krylov subspace methods are also utilized, including the Davidson method [72, 73, 74] and Jacobian-free Newton-Krylov approaches [75, 76, 77], which is a Newton-based method based on Krylov subspaces. Gonzalez et al. in [49] investigated the use of PGD with a Arnoldi-like process for a single-group k-eigenvalue problem. However, this process does not guarantee the evaluation of the largest eigenvalue or the convergence of the eigenvector. This work utilizes standard power iteration accelerated with a Wielandt shift, more specifically, an adaptive shift [68].

The goal of this work is to analyze and comment on the capability of the PGD approach for

evaluating criticality problems for reactor-physics calculations. This chapter presents three extensions to the PGD technique: (1) the application of PGD to accelerated power iterations for eigenvalue problems, (2) the design of an effective iteration procedure for PGD applied to eigenvalue problems, including compression of the fission integral at each iteration, (3) the construction of PGD operators for multigroup neutron diffusion in highly heterogeneous configurations and its effective implementation. The outline of the chapter is as follows: Section 3.2 briefly reviews multigroup diffusion criticality problems and their numerical solution using accelerated power iteration techniques. Section 3.3 discusses the the application of the PGD solution process to power iterations. Results are presented in Section 3.4.

3.2 Overview of the Multigroup Eigenvalue Problem

This section recalls the multigroup k-eigenvalue problem and discusses the power iteration and shifted power iteration process. For criticality calculations, the eigenvalue-eigenvector pair of interest is the one associated with the largest eigenvalue (fundamental mode). The largest value of k, known as the effective neutron multiplication factor or k_{eff}, can be computed using power iteration, for example, or an accelerated variant of power iteration.

3.2.1 Multigroup Eigenvalue Problem

The multigroup neutron diffusion k-eigenvalue equations are

$$-\nabla\cdot D^g\nabla\phi^g+\Sigma_t^g\phi^g=\sum_{g'=1}^{G}\Sigma_s^{g\leftarrow g'}\phi^{g'}+\frac{1}{k}\chi^g\sum_{g'=1}^{G}\nu\Sigma_f^{g'}\phi^{g'}\ ,\ g=1,...,G\,. \tag{3.1}$$

Each term in this equation is defined as:

G - total number of energy groups

ϕ^g - multigroup scalar neutron flux

D^g - multigroup diffusion coefficient

Σ_t^g - total macroscopic cross section

$\Sigma_s^{g\leftarrow g'}$ - macroscopic scattering cross section from group g' to group g

χ^g - fraction of neutrons generated in group g from fission (fission spectrum)

$\nu\Sigma_f^{g'}$ - product of the macroscopic fission cross section and the average number of neutrons emitted per fission in group g'

The equations in (3.1) can then be written into matrix form,

$$(\boldsymbol{D} + \boldsymbol{A} - \boldsymbol{S})\,\Phi = \frac{1}{k}\boldsymbol{F}\Phi\,, \tag{3.2}$$

where

$$(\Phi)_g = \phi^g\,, \tag{3.3a}$$

$$(\boldsymbol{D})_{gg'} = -\nabla\cdot D^g\nabla\delta_{gg'}\,, \tag{3.3b}$$

$$(\boldsymbol{A})_{gg'} = \Sigma_t^g\delta_{gg'}\,, \tag{3.3c}$$

$$(\boldsymbol{S})_{gg'} = \Sigma_s^{g\leftarrow g'}\,, \tag{3.3d}$$

$$(\boldsymbol{F})_{gg'} = \chi^g\nu\Sigma_f^{g'}\,. \tag{3.3e}$$

The neutron diffusion criticality equations form a generalized eigenvalue problem, which can be written in operator notation as:

$$\mathcal{A}\Phi = \frac{1}{k}\mathcal{B}\Phi\,, \tag{3.4}$$

with

$$\mathcal{A} = \boldsymbol{D} + \boldsymbol{A} - \boldsymbol{S} \;\text{ and }\; \mathcal{B} = \boldsymbol{F}\,.$$

This is a nonlinear problem for the eigenpair (k, Φ). Numerous methods have been devised to solve this eigenproblem; see, for instance, the references cited in the introduction. This work will focus on the application of power iteration and shifted power iteration.

3.2.2 Power Iteration and Shifted Power Iteration

The standard power iteration process proceeds as follows. Given a previous iterate ℓ for the pair (k, Φ), the new iteration is obtained by solving

$$\mathcal{A}\Phi^{\ell+1} = \frac{1}{k^{\ell}}\mathcal{B}\Phi^{\ell}\,, \tag{3.5a}$$

for the new estimate of the eigenvector; the next estimate of the eigenvalue is then computed using

$$k^{\ell+1} = \frac{\left\langle \Phi^{\ell+1}, \mathcal{B}\Phi^{\ell+1} \right\rangle}{\left\langle \Phi^{\ell+1}, \mathcal{A}\Phi^{\ell+1} \right\rangle} = \frac{\sum_{g=1}^{G} \int_{\mathcal{D}} (\Phi^{\ell+1})_g (\mathcal{B}\Phi^{\ell+1})_g d\vec{r}}{\sum_{g=1}^{G} \int_{\mathcal{D}} (\Phi^{\ell+1})_g (\mathcal{A}\Phi^{\ell+1})_g d\vec{r}}, \tag{3.5b}$$

where ℓ is the iteration index and the definition of k is known as the (inverse of the) Rayleigh quotient [78]. This iterative technique is simple to implement, but suffers from slow convergence behavior in reactor physics where the dominance ratio can be close to one [61]. Shifted power iteration is an example of accelerated power iterations whereby a shifting parameter is introduced to move the dominance ratio away from unity. This shift is applied as

$$\left(\mathcal{A} - \sigma^{\ell}\mathcal{B}\right)\Phi^{\ell+1} = \left(\frac{1}{k^{\ell}} - \sigma^{\ell}\right)\mathcal{B}\Phi^{\ell}\,, \tag{3.6}$$

where $k^{\ell+1}$ is computed using Equation (3.5b). The parameter σ is known as the Wielandt shift. Selecting $\sigma = 0$ would equate to a standard power iteration process. Application of shift can greatly improve the convergence of the iteration procedure, a value of σ closest to the value of $\frac{1}{k_{\text{eff}}}$ is optimal [78]. The Rayleigh quotient iteration uses a shift equal to the last computed eigenvalue, i.e., $\sigma^{\ell} = 1/k^{\ell}$. This choice of the Wielandt shift can result in a cubic convergence of the eigenvalue, while a constant shift converges linearly [79]. However, since the eigenvector typically converges more slowly than the eigenvalue, the system defined in Equation (3.6) can become ill-conditioned before the eigenvector is fully converged. Yee et. al [68] analyzed several other

adaptive approaches for σ, the one chosen for this work is defined as,

$$\sigma^{\ell} = \max\left(\frac{1}{k^{\ell}} - c_1\left|\frac{1}{k^{\ell}} - \frac{1}{k^{\ell-1}}\right| - c_0, \frac{1}{k_{\max}}\right), \tag{3.7}$$

where c_1, c_0, and $k_{\max}$ are tuning parameters chosen to be 10, 0.01, and 3, respectively, in this paper. These specific values for the parameters were chosen for analysis in [68]. Different values may prove to be more optimal for specific eigenvalue problems, but finding these optimal values is beyond the scope of this work.

The systems of equations defined by Equations (3.5a) and (3.6) are linear systems that can be represented in the form $\mathcal{L}\Phi = q$. Solving such systems requires an inversion of the operator $\mathcal{L}$, given a source q. It is at this stage that the PGD method is applied. PGD seeks a separated representation as an approximation to Φ. As discussed earlier, there are iterative steps associated with PGD, (1) solving for the various one-dimensional components of the separated representation and (2) enrichment of the representation with enough terms. These PGD iterations will thus be nested inside the power iteration process. The nesting of power iteration and PGD solver, as well as special considerations regarding the PGD operator decompositions, are discussed in the next section.

3.3 Proper Generalized Decomposition

So far, this chapter discussed the derivation multigroup criticality as a generalized eigenvalue problem and the power iteration procedure to resolve the nonlinearities of the eigenvalue-eigenvector pair. This section discusses the application of PGD within a power iteration. Figure 3.1 illustrates the design of this application.

As stated earlier, PGD seeks a separated representation, as shown in Equation (1.7b), for the solution of a system. For brevity, PGD process is described in the case of a two-dimensional problem, but the 3-D derivation can easily inferred from this exposé. Note that once a solution is approximated (and sought) as products of one-dimensional functions, it is advantageous from an implementation point-of-view to represent the operators in a similar manner. This operator

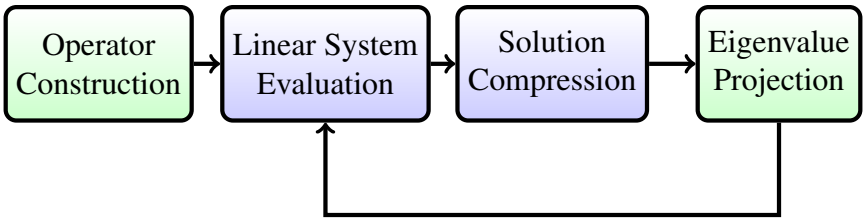


Figure 3.1: Design of PGD application to power iteration procedure

decomposition is discussed in Section 3.3.1. After the operator has been defined and a certain iteration's solution established, the resulting linear system can be solved using the PGD technique, discussed in Section 3.3.2. Once a solution from the linear system is resolved, the separated form can be compressed in order to alleviate the cost of eigenvalue evaluation and source construction. PGD compression is explained in Section 3.3.3. Finally, Section 3.3.4 discusses the computation of the eigenvalue after a PGD projection step; the error definitions for the iteration procedures are also introduced at that moment.

3.3.1 Operator Decomposition

In the case of a two-dimensional problem, a PGD representation (decomposition) of the eigenvector as given by Equation (3.8a) [16].

$$\Phi(x,y) = \sum_{i=1}^{N} X_i(x) Y_i(y)\,. \tag{3.8a}$$

The operators of the generalized eigenvalue problem can be similarly decomposed:

$$\mathcal{A}(x,y) = \sum_{j=1}^{A} \mathcal{A}_j^x(x) \mathcal{A}_j^y(y)\,. \tag{3.8b}$$

$$\mathcal{B}(x,y) = \sum_{j=1}^{B} \mathcal{B}_j^x(x) \mathcal{B}_j^y(y)\,. \tag{3.8c}$$

The generalized eigenvalue problem of Equation (3.4) can be re-written using a PGD representation:

$$\sum_{i=1}^{N}\sum_{j=1}^{A}\mathcal{A}_j^x(x)X_i(x)\mathcal{A}_j^y(y)Y_i(y) = \frac{1}{k}\sum_{i=1}^{N}\sum_{j=1}^{B}\mathcal{B}_j^x(x)X_i(x)\mathcal{B}_j^y(y)Y_i(y)\,, \tag{3.8d}$$

where A and B are the number of terms needed to fully decompose the $\mathcal{A}$ and $\mathcal{B}$ operators, respectively. Details on the formulation of these operators is discussed in Section 3.3.5.

3.3.2 Linear System Solves using PGD

At each power iteration (Equations (3.5) and (3.6)), one needs to solve a linear system, where the source term is the lagged fission integral from the previous power iteration. These linear system can be generalized by Equation (2.1) with the corresponding PGD representation in Equation (2.2b). In the context of shifted power iteration, these decomposed operators are defined as

$$\mathcal{L}_m^{x,\ell+1}\mathcal{L}_m^{y,\ell+1} = \begin{cases} \mathcal{A}_m^x\mathcal{A}_m^y & m = 1,..,A \\ -\sigma^{\ell+1}\mathcal{B}_{m-A}^x\mathcal{B}_{m-A}^y & m = A+1,...,A+B \end{cases}, \tag{3.9a}$$

$$q_n^{x,\ell+1}q_n^{y,\ell+1} = \mathcal{B}_j^x X_i^\ell(x)\mathcal{B}_j^y Y_i^\ell(y)\,, \quad n = (i-1)B + j\,. \tag{3.9b}$$

The total number of terms in the linear operator is then $L = A + B$. It should be noted that Equation (3.9a) applies for the shifted power iteration approach; in the unshifted case ($\sigma = 0$) the terms due to $\mathcal{B}$ are not present and thus $L = A$. The total number of terms in the source is $Q = B \times N$, which can become quite large if N is large, which can occur in highly heterogeneous media. Decreasing N in a power iteration decreases the computational effort for future iterations. Methods for minimizing N are discussed in Section 3.3.3. The rest of the PGD solution process is discussed in detail in Section 2.1, which can be applied directly here.

3.3.3 Solution Compression

In the power iteration process, the latest iterate is used to compute the next fission source. Here, the latest iteration is a PGD-decomposed solution and may not be optimal in terms of the number of expansion terms when used to compute the fission integral. Therefore, there is an opportunity

to compress the PGD solution, i.e., find a new decomposition such that the number of summation terms, N in Equation (3.8a), is reduced. This compression would reduce the number of terms in the source term of the next power iteration (Q in Equation (2.2c)) resulting in faster evaluation of all the terms in Equation (3.9b) and the right-hand-side of Equation (2.7). An obvious candidate for PGD compression is employing Singular Value Decomposition (SVD) on the two-dimensional array representation of the full solution; SVD is known to yield an optimal decomposition [57]. However, SVD is only applicable to two-dimensional array representations and requires the full-order solution, which PGD attempts to avoid in the first place. Another possibility is to use a PGD approach to compress the PGD solution (see, for instance, Chapter 3 of [16]). This work utilizes the latter approach.

To compress the current power iteration iterate obtained from the PGD process described in Section 3.3.2, the PGD process from Section 2.1 is utilized, but applied to a purely algebraic equation this time:

$$\sum_{i=1}^{\tilde{N}<N} \tilde{X}_i(x)\tilde{Y}_i(y) = \sum_{i'=1}^{N} X_{i'}(x)Y_{i'}(y)\,. \tag{3.10}$$

The right-hand-side of Equation (3.10) is known and the terms on left-hand-side are solved progressively through the aforementioned alternating direction and enrichment process. For instance, when evaluating the term $\tilde{X}_{\tilde{N}}(x)$, a test function of $v(x,y) = X^*(x)\tilde{Y}_{\tilde{N}}(y)$ is applied and integrated over the domain, resulting in:

$$\begin{aligned}\int_{\Omega_x} X^*(x)\tilde{X}_{\tilde{N}}(x)dx \int_{\Omega_y} \tilde{Y}_{\tilde{N}}(y)\tilde{Y}_{\tilde{N}}(y)dy = &\sum_{i'=1}^{N} \int_{\Omega_x} X^*(x)X_{i'}(x)dx \int_{\Omega_y} \tilde{Y}_{\tilde{N}}(y)Y_i(y)dy \\ &- \sum_{i=1}^{\tilde{N}-1} \int_{\Omega_x} X^*(x)\tilde{X}_i(x)dx \int_{\Omega_y} \tilde{Y}_{\tilde{N}}(y)\tilde{Y}_i(y)dy\,. \end{aligned} \tag{3.11}$$

Noticeably, Equation (3.11) involves no differential operators, thus the strong form of this equation for $\tilde{X}_{\tilde{N}}$ can be written as,

$$\beta^y_{\tilde{N},\tilde{N}}\tilde{X}_{\tilde{N}}(x) = \sum_{i'=1}^{N} \gamma^y_{\tilde{N},i'}X_{i'}(x) - \sum_{i=1}^{\tilde{N}-1} \beta^y_{\tilde{N},i}\tilde{X}_i(x)\,, \tag{3.12a}$$

where β and γ are scalars defined as,

$$\beta_{i,j}^{y} = \int_{\Omega_y} \tilde{Y}_i(y)\tilde{Y}_j(y)dy \,, \tag{3.12b}$$

$$\gamma_{i,j}^{y} = \int_{\Omega_y} \tilde{Y}_i(y)Y_j(y)dy \,. \tag{3.12c}$$

An equation for $\tilde{Y}_{\tilde{N}}$ can be found similarly. The additional computational effort in evaluating this compressed solution is minuscule relative to cost of a single power iteration solve since Equation (3.12a) only requires the evaluation of 1-D integrals and does not result in a linear system of equations.

3.3.4 Eigenvalue Projection

This section describes the projection step of the PGD solution, whereby each term in the PGD solution is rescaled optimally. This projection is similar to the projection described in early PGD works [22, 27, 49]; for this application, this projection is exploited too obtain an improved eigenvector through a reduced eigenvalue problem. In this projection, one attempt at improving the PGD decomposition by assuming the optimal separated representation is a linear combination of the one found from the enrichment process. Using this methodology, the solution found from enrichment and compression is defined as $\Phi^{\ell+1/2}$ and the projected solution as:

$$\Phi^{\ell+1}(x,y) = \sum_{i'=1}^{N} \alpha_{i'} X_{i'}^{\ell+1/2}(x) Y_{i'}^{\ell+1/2}(y) \,, \tag{3.13}$$

where the α coefficients are to be determined. To compute the α coefficients, Equation (3.13) is applied to the generalized eigenvalue problem Equation (3.5a). Multiplying by different enrichment terms from the current PGD solution ($X_i^{\ell+1/2}(x)Y_i^{\ell+1/2}(y)$) and integrating over space, an eigenvalue problem appears for these coefficients:

$$\bar{\mathcal{A}}\boldsymbol{\alpha} = \frac{1}{k^{\ell+1}}\bar{\mathcal{B}}\boldsymbol{\alpha} \,, \tag{3.14}$$

where

$$(\boldsymbol{\alpha})_i = \alpha_i \,, \tag{3.15a}$$

$$(\bar{\mathcal{A}})_{ii'} = \sum_{j=1}^{A} \int_{\Omega_x} X_i^{\ell+1/2} \mathcal{A}_j^x X_{i'}^{\ell+1/2} dx \int_{\Omega_y} Y_i^{\ell+1/2} \mathcal{A}_j^y Y_{i'}^{\ell+1/2} dy \,, \tag{3.15b}$$

$$(\bar{\mathcal{B}})_{ii'} = \sum_{j=1}^{B} \int_{\Omega_x} X_i^{\ell+1/2} \mathcal{B}_j^x X_{i'}^{\ell+1/2} dx \int_{\Omega_y} Y_i^{\ell+1/2} \mathcal{B}_j^y Y_{i'}^{\ell+1/2} dy \,, \tag{3.15c}$$

$$i = 1, ..., N \,, \quad i' = 1, ..., N \,.$$

The eigenvalue problem given in Equation (3.14) is much smaller than the full-order version because the spatial variables have been integrated out; it is therefore significantly cheaper to evaluate than a single PGD linear system solve. After this projection step, the improved PGD solution ($\Phi^{\ell+1}$) is the latest power iteration solution and it is employed to obtain the latest eigenvalue ($k^{\ell+1}$).

With the combination of the PGD solution process and the power iteration technique, various nested iterations are present. To summarize the resulting iteration loops next: (1) eigenvalue (power) iteration, (2) PGD enrichment, and (3) alternating direction iteration for a given PGD enrichment step. The power iteration convergence criteria is defined as,

$$E_k^{\ell+1} = \frac{|k^{\ell+1} - k^{\ell}|}{k^{\ell}} < \epsilon_k \text{ (eigenvalue tolerance)}, \tag{3.16a}$$

$$E_\phi^{\ell+1} = \frac{||\Phi^{\ell+1} - \Phi^{\ell}||}{||\Phi^{\ell}||} < \epsilon_\phi \text{ (eigenvector tolerance)}, \tag{3.16b}$$

and the enrichment and alternating direction criteria are defined by Equations (2.8) and (2.9), respectively. Because of the nested iteration, it may be beneficial to tighten the inner iteration tolerances as the power iteration converges, allowing for looser convergence checks in the earlier stages of the power iteration process. This "inexact" approach to the PGD solution process prevents "over solving" each power iteration. Therefore, the enrichment error in this work for a particular power iteration is defined as:

$$\epsilon_{\text{enr}}^{\ell+1} = 0.01 \times \max\left(E_k^{\ell}, E_\phi^{\ell}\right) \,. \tag{3.17}$$

The resulting procedure for shifted power iteration with PGD is shown by Algorithm 1.

Algorithm 1 2D Generalized Eigenvalue Problem using PGD

procedure [Φ,k] = PGDEIGENVALUE($\mathcal{A}$,$\mathcal{B}$)
  $\mathcal{A} := [\mathcal{A}_1^x, ..., \mathcal{A}_A^x; \mathcal{A}_1^y, ..., \mathcal{A}_A^y]$
  $\mathcal{B} := [\mathcal{B}_1^x, ..., \mathcal{B}_B^x; \mathcal{B}_1^y, ..., \mathcal{B}_B^y]$
    Initialization: $\ell = 0$, $E_k^\ell = 1$, $E_\phi^\ell = 1$, $\sigma^\ell = 1/k_{\max}$
    Define initial eigenvector: Φ^ℓ
    Compute eigenvalue: $k^\ell \leftarrow$ Equation (3.5b)
    while $E_k^\ell < \epsilon_k$ and $E_\phi^\ell < \epsilon_\phi$ **do**
      Build linear operator: $\mathcal{L}^\ell \leftarrow$ Equation (3.9a)
      Build linear source: $q^\ell \leftarrow$ Equation (3.9b)
      Define enrichment tolerance: $\epsilon_{\text{enr}}^\ell \leftarrow$ Equation (3.17)
      Solve linear system: $\Phi^{\ell+1/2} := \sum_{i=1}^{N} X_i(x)Y_i(y) \leftarrow$Figure 2.1
      Compress eigenvector: $\Phi^{\ell+1/2} := \sum_{i=1}^{\tilde{N}\leq N} X_i(x)Y_i(y) \leftarrow$ Equation (3.10)
      Project solution and compute eigenvalue: $k^{\ell+1}$, $\Phi^{\ell+1} \leftarrow$ Equation (3.14)
      Compute shift: $\sigma^{\ell+1} \leftarrow$ Equation (3.7)
      $\ell = \ell + 1$
    end while
end procedure

3.3.5 PGD for Multigroup Neutron Diffusion

This section discusses the application of PGD to the operators and variables in the case of multigroup equations. Generally speaking, the application of the PGD process to the multigroup setting is relatively straightforward, just like extending a traditional one-group solver to the multi-group case, as in [80, 81], for instance. To begin, the scalar neutron flux and the linear operators is described as a decomposed spatial variable for each group (using a 2D example for brevity),

$$\phi^g(x,y) = \sum_{i=1}^{N} X_i^g(x) Y_i^g(y)\,, \tag{3.18a}$$

$$-\nabla \cdot D^g(x,y)\nabla = -\sum_{j=1}^{N_d} \left[D_j^{g,y}(y)\frac{\partial}{\partial x} D_j^{g,x}(x)\frac{\partial}{\partial x} + D_j^{g,x}(x)\frac{\partial}{\partial y} D_j^{g,y}(y)\frac{\partial}{\partial y} \right]\,, \tag{3.18b}$$

$$\Sigma_t^g(x,y) = \sum_{j=1}^{N_t} \Sigma_{t,j}^{g,x}(x)\Sigma_{t,j}^{g,y}(y)\,, \tag{3.18c}$$

$$\Sigma_s^{g\leftarrow g'}(x,y) = \sum_{j=1}^{N_s} \Sigma_{s,j}^{g\leftarrow g',x}(x)\Sigma_{s,j}^{g\leftarrow g',y}(y)\,, \tag{3.18d}$$

$$\nu\Sigma_f^g(x,y) = \sum_{j=1}^{N_f} \nu\Sigma_{f,j}^{g,x}(x)\nu\Sigma_{f,j}^{g,y}(y)\,. \tag{3.18e}$$

The decomposed description of the material properties are described in Section 2.3.1. Note that the number of terms in each expansion (N, N_d, N_s, N_f) is independent of the group or group pair. Therefore, the number of terms for each group property description is equal to the number of terms of the group that has the largest expansion. In this work, the fission spectrum χ^g is independent of the material zone.

The one-dimensional variables for this work are discretized using CFEM. With this framework, the multigroup flux and operators can be represented in matrix notation:

$$X_i^g(x)Y_i^g(y) \to (\boldsymbol{X}_i)_g \otimes (\boldsymbol{Y}_i)_g\,, \tag{3.19a}$$

$$\frac{\partial}{\partial x}D_j^{g,x}(x)\frac{\partial}{\partial x} \to (\boldsymbol{K}_j^x)_g \otimes \boldsymbol{M}^y\,, \quad \frac{\partial}{\partial y}D_j^{g,y}(y)\frac{\partial}{\partial y} \to \boldsymbol{M}^x \otimes (\boldsymbol{K}_j^y)_g\,, \tag{3.19b}$$

$$\Sigma_{t,j}^{g,x}(x)\Sigma_{t,j}^{g,y}(y) \to (\boldsymbol{M}_{t,j}^x)_{gg} \otimes (\boldsymbol{M}_{t,j}^y)_{gg}\,, \tag{3.19c}$$

$$\Sigma_{s,j}^{g\leftarrow g',x}(x)\Sigma_{s,j}^{g\leftarrow g',y}(y) \to (\boldsymbol{M}_{s,j}^x)_{gg'} \otimes (\boldsymbol{M}_{s,j}^y)_{gg'}\,, \tag{3.19d}$$

$$\chi^g\nu\Sigma_{f,j}^{g',x}(x)\nu\Sigma_{f,j}^{g',y}(y) \to (\boldsymbol{M}_{f,j}^x)_{gg'} \otimes (\boldsymbol{M}_{f,j}^y)_{gg'}\,. \tag{3.19e}$$

The group structure of the resulting matrices are equivalent to the ones described in Equation (3.3).

One can now easily define the eigenvalue problem operators described in Equation (3.8):

$$\mathcal{A}_j^x = \begin{cases} \boldsymbol{K}_j^x & j = 1, ..., N_d \\ \boldsymbol{M}^x & j = N_d + 1, ..., 2N_d \\ \boldsymbol{M}_{t,j-2N_d}^x & j = 2N_d + 1, ..., 2N_d + N_t \\ \boldsymbol{M}_{s,j-2N_d-N_t}^x & j = 2N_d + N_t + 1, ..., 2N_d + N_t + N_s \end{cases}, \tag{3.20a}$$

$$\mathcal{A}_j^y = \begin{cases} \boldsymbol{M}^y & j = 1, ..., N_d \\ \boldsymbol{K}_{j-N_d}^y & j = N_d + 1, ..., 2N_d \\ \boldsymbol{M}_{t,j-2N_d}^y & j = 2N_d + 1, ..., 2N_d + N_t \\ \boldsymbol{M}_{s,j-2N_d-N_t}^y & j = 2N_d + N_t + 1, ..., 2N_d + N_t + N_s \end{cases}, \tag{3.20b}$$

$$\mathcal{B}_j^x = \boldsymbol{M}_{f,j}^x \,, \quad j = 1, ..., N_f \,, \tag{3.20c}$$

$$\mathcal{B}_j^y = \boldsymbol{M}_{f,j}^y \,, \quad j = 1, ..., N_f \,. \tag{3.20d}$$

Using these discretized operators for the multigroup neutron diffusion equation, the generalized eigenvalue problem can be evaluated using Algorithm 1.

3.4 Results

To illustrate the capability of our PGD algorithm for multigroup criticality problems, it is applied to four different problems. The first is a bare homogeneous cube with four-group cross sections, which is meant to verify the implementation with an analytic solution. The second example is a mildly heterogeneous 2-D two-group problem that Senecal & Wi analyzed as a fixed source problem in [50]. The third and final problems use the geometry and material properties of the BIBLIS benchmark [82], the analysis includes the one-group version from [49] and the two-group benchmark version.

The PGD method was implemented in MATLAB and utilizes first-order CFEM for the 1D solutions. The linear systems in the PGD iteration process are evaluated using MATLAB's sparse LU solver. To produce a reference solution and compare performance with a full-order model, each

problem was also implemented in the MOOSE. The MOOSE implementation also uses first-order CFEM and linear solvers are performed using algebraic multi-grid techniques from the HYPRE package with default numerical parameters. Both PGD and MOOSE solve each problem on the same uniform meshes with rectangular elements. The meshes are uniformly refined for each problem to gauge accuracy versus run-time, these refinements are identified by the number of elements in the x and y dimensions, the total number of elements is therefore this number squared.

3.4.1 Four-Group Bare Homogeneous Reactor

This first example is a 3-D four-group, homogeneous reactor, which has an derivable analytical solution. The reactor is $300 \times 300 \times 300$ cm with cross-section values described in Table 3.1. The analytical k_{eff} is 1.13530390 and eigenfunction is

$$\phi^g(x, y, z) = C^g \cos\left(\frac{\pi}{300}x\right) \cos\left(\frac{\pi}{300}y\right) \cos\left(\frac{\pi}{300}z\right) , \tag{3.21}$$

where C^g are group-dependent constants.

Table 3.1: Material property values for homogeneous bare reactor

Group	D^g	Σ_r^g	$\nu\Sigma_f^g$	χ^g	$\Sigma_s^{g\leftarrow 1}$	$\Sigma_s^{g\leftarrow 2}$	$\Sigma_s^{g\leftarrow 3}$	$\Sigma_s^{g\leftarrow 4}$
1	2.500	0.0668	0.00835	0.75	0	0	0	0
2	1.050	0.0623	7.89e-4	0.25	0.668	0	0	0
3	0.676	0.0707	0.0101	0	4.77e-4	0.0630	0	0.00161
4	0.379	0.0909	0.117	0	0	3.0e-6	0.0482	0

The full-order model, evaluated using MOOSE, was run with 10 to 40 elements per dimension, and the PGD model was run with 10 to 160 elements per dimension. The eigenvalue and eigenvector convergence tolerance (ϵ_k,ϵ_ϕ) were set to 10^{-9}. Figure 3.2 shows the resulting eigenvalue and eigenvector error for MOOSE with power iteration and PGD with and without adaptive Wielandt shift. These errors show that both the full-order and PGD models have proper error convergence when the spatial mesh is refined. Figure 3.3 shows the resulting run-time for each simulation.

For this problem, PGD shows far superior performance compared to MOOSE evaluation. Since the exact solution is analytically separable, only one or two enrichments are needed as the power iteration converges, making this a computationally trivial problem for the PGD method. Table 3.2 shows the relative computational efficiency of linear system evaluation for each spatial refinement for the unshifted PGD model and full-order MOOSE model. The efficiency of a numerical scheme is defined as

$$\text{Efficiency} = \frac{1}{k_{\text{eff}}\,\text{Error} \times \text{CPU Time per PI}}\,. \tag{3.22}$$

Efficiency is higher for methods with low error and low CPU time. Because unaccelerated and accelerated power iterations are employed, the efficiency values use the average CPU time per power iteration. Table 3.2 shows the efficiency of the MOOSE solver and the PGD unshifted solver, relative to that of the PGD with shift approach. The efficiency of the unshifted PGD solver is greater than that of the shifted one is because the shifted operator appearing on the left-hand-side is more complex due to the Wiedlandt shift acting on the fission operator, making the unshifted PGD operator quicker to invert. The relative efficiency of the unshifted PGD scheme also increases with mesh refinement because the PGD solver spends less time in the tight enrichment tolerance range, due to the decrease in power iterations needed. When compared with the full multi-D solves performed using MOOSE, the PGD approaches become significantly more efficient as the mesh is refined.

Table 3.2: Relative computational efficiency for unshifted PGD and MOOSE, homogeneous four-group problem (baseline is the PGD method with shift; values < 1 denote a less efficient technique)

Elements per Dim.	PGD	MOOSE
10	1.3002	0.3919
20	1.3826	0.0528
40	1.5803	0.0103
80	1.7943	—
160	1.9274	—

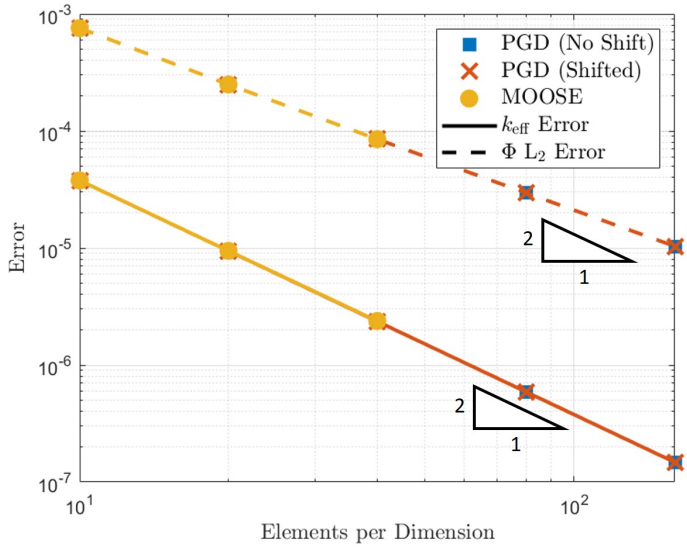


Figure 3.2: Relative eigenvalue and L_2 eigenvector error for the homogeneous four-group problem

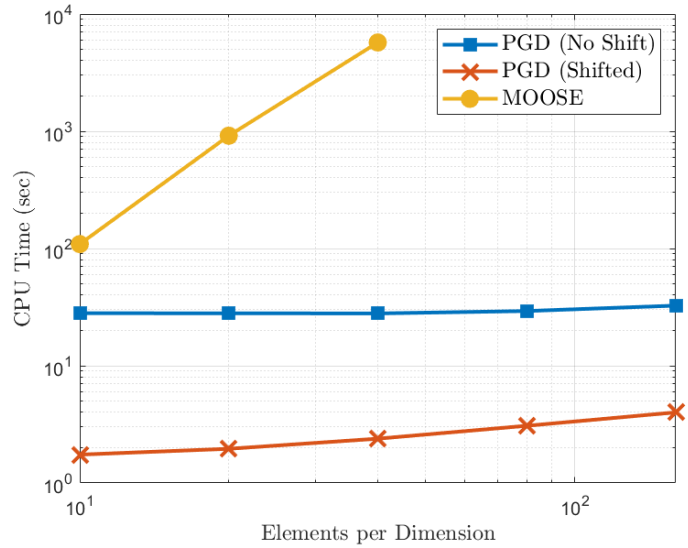


Figure 3.3: Run-time in seconds for each simulation of the homogeneous four-group problem

3.4.2 2-D Two-Group Problem

This next example is a simple heterogeneous 2-D, two-group example based on Problem 5.7 from [83]; a fixed-source version of this problem was analyzed in [50]. Figure 3.4 shows the ge-

ometry of the problem and the material properties can be found in Table 3.3. The full-order model, evaluated using MOOSE, was run with 35 to 1,120 elements per dimension, and the PGD model was run with 35 to 2,240 elements per dimension. The eigenvalue convergence tolerance (ϵ_k) was set to 10^{-9}. A MOOSE model with 2,240 elements per dimension was used for the reference eigenvalue and eigenvector with $k_{\text{eff}} = 0.99010305$; this solution required high performance computing to evaluate, so the performance results for this simulation are not included.

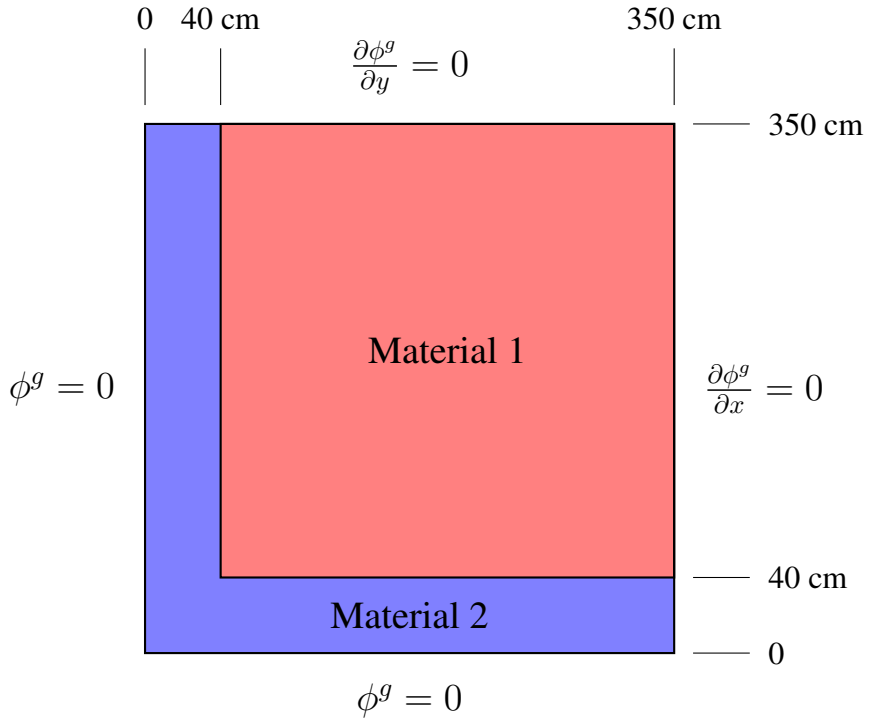


Figure 3.4: Geometry of 2-D two-group problem (figure not to scale)

Figure 3.5 shows the resulting eigenvalue and eigenvector error for MOOSE with power iteration and PGD with and without adaptive Wielandt shift. These errors show that both the full-order and PGD models have proper error convergence when the spatial mesh is refined. The eigenvalue errors in the last two refinements begin to deviate slightly between each method. This is most likely due to the fact that the error at these points are relatively close to the tolerance set for the power iteration. Figure 3.6 shows the resulting run-time for each simulation. For this problem,

Table 3.3: Material properties of 2-D two-group problem

Material	Group	D^g (cm)	Σ_a^g (cm^{-1})	$\nu\Sigma_f^g$ (cm^{-1})	$\Sigma_s^{g\leftarrow 1}$ (cm^{-1})
1	1	1.264	7.86e-4	4.562e-3	0.0
	2	0.9328	4.10e-3	0.0	7.368e-3
2	1	1.310	0.0	0.0	0.0
	2	0.8695	2.117e-4	0.0	1.018e-2

$\chi^1 = 1.0, \chi^2 = 0.0$

PGD shows far superior performance compared to MOOSE execution. Since this problem has very little heterogeneity, it is another relatively simple problem for PGD. Figure 3.7 shows the contribution of each enrichment for the PGD runs at 560 elements per dimension and the magnitude of each term of the SVD of the full-order model solution. Both the SVD and PGD enrichment convergence shows that a relatively few number of terms are needed to represent the converged solution to the iteration tolerance. The shifted PGD requires more enrichments than the unshifted simulation because the last power iteration had a tighter enrichment tolerance for the shifted simulation. This is due to the adaptive enrichment tolerance procedure defined by Equation (3.17). Table 3.4 shows the relative computational efficiency for each refinement of the unshifted PGD model and full-order MOOSE model. The unshifted PGD efficiency shows the same trend as in the previous example. As noted earlier, the efficiency of the full multi-D solves (using MOOSE) decreases significantly as the mesh is refined, compare to that of PGD-based approaches.

Table 3.4: Relative computational efficiency for unshifted PGD and MOOSE, 2-D two-group problem (baseline is the PGD method with shift; values < 1 denote a less efficient technique)

Elements per Dim.	PGD	MOOSE
70	0.3901	0.2598
140	0.4544	0.0447
280	0.6578	0.0260
560	0.5371	0.0135
1120	0.7804	0.0072

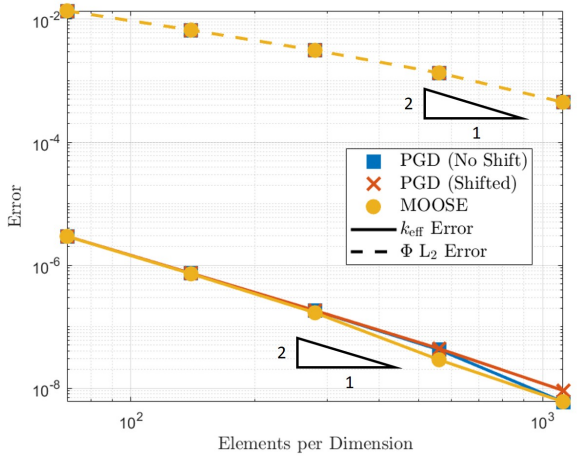


Figure 3.5: Relative eigenvalue and L_2 eigenvector error for the 2-D two-group problem

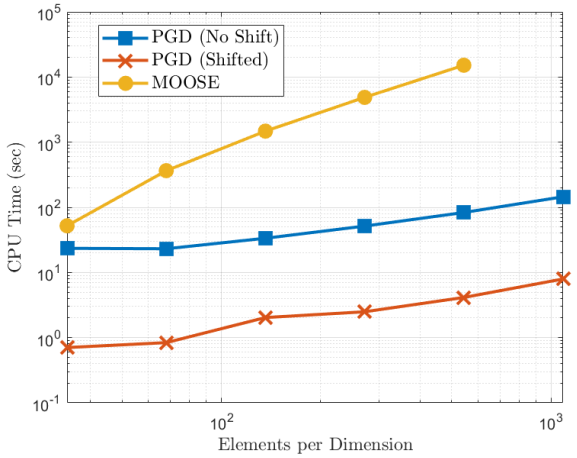


Figure 3.6: Run-time in seconds for each simulation of the 2-D two-group problem

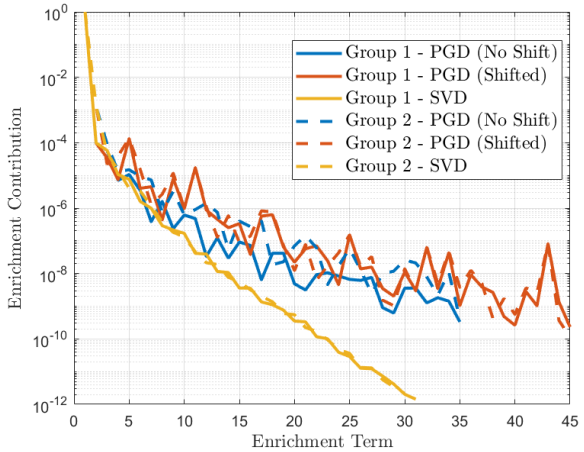


Figure 3.7: Enrichment contribution of converged PGD solutions and SVD of the full-order solution 2-D two-group problem

3.4.3 BIBLIS Benchmark Problem

These next two examples use a geometry based on the BIBLIS benchmark problem, which is shown in Figure 3.8. This problem is meant to illustrate PGD performance for highly heterogeneous geometries and compare between mono-energetic and multigroup systems. The first of these examples is the single-group version and second is the two-group version.

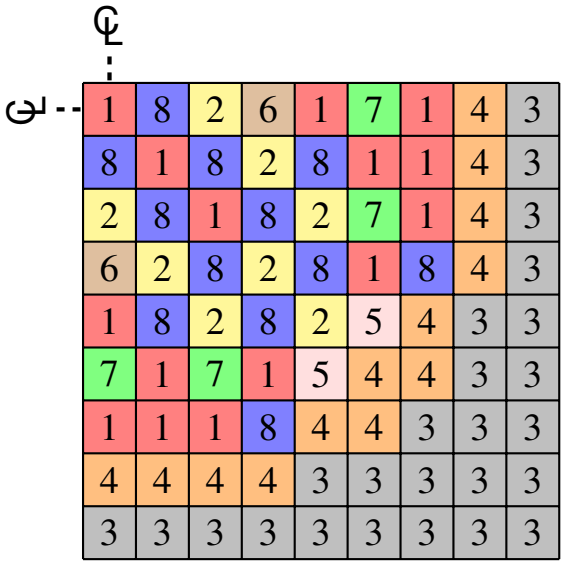


Figure 3.8: BIBLIS benchmark geometry: each block is 23.1226 cm $\times$ 23.1226 cm

3.4.3.1 One-Group Version

The material property values for one-group version of the BIBLIS benchmark can be found in Table 3.5. The full-order model, evaluated using MOOSE, was run with 34 to 1,088 elements per dimension, and the PGD model was run with 34 to 2,176 elements per dimension. The eigenvalue convergence tolerance (ϵ_k) were set to 10^{-9}. A MOOSE model with 2,176 elements per dimension was used for the reference eigenvalue and eigenvector with $k_{\text{eff}} = 0.618502168$; this solution required high performance computing to evaluate, so the performance results for this simulation are not included.

Table 3.5: Material properties for one-group BIBLIS benchmark problem

Material	D (cm)	Σ_a (cm^{-1})	$\nu\Sigma_f$ (cm^{-1})
1	1.4360	0.0095042	0.0058708
2	1.4366	0.0096785	0.0061908
3	1.3200	0.0026562	0.0000000
4	1.4389	0.0103630	0.0074527
5	1.4381	0.0100030	0.0061908
6	1.4385	0.0101320	0.0064285
7	1.4389	0.0101650	0.0061908
8	1.4393	0.0102940	0.0061908

Figure 3.9 shows the resulting eigenvalue and eigenvector error for MOOSE with power iteration and PGD with and without adaptive Wielandt shift. These errors show that both the full-order and PGD models, again, have proper error convergence when the spatial mesh is refined. To compare the solution between each model, Figure 3.10 shows the full-order solution at 272 elements per dimension with the difference from the PGD models at the same refinement. These plots show that the MOOSE and PGD evaluations converge to approximately the same solution in the power iteration process.

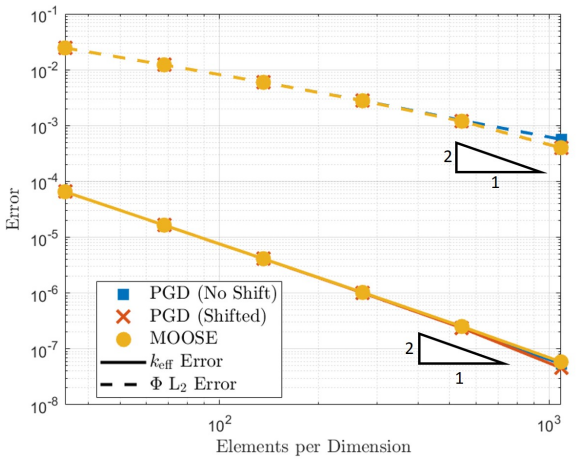
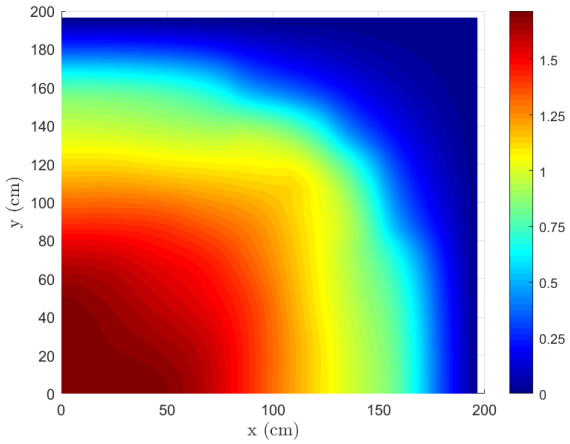
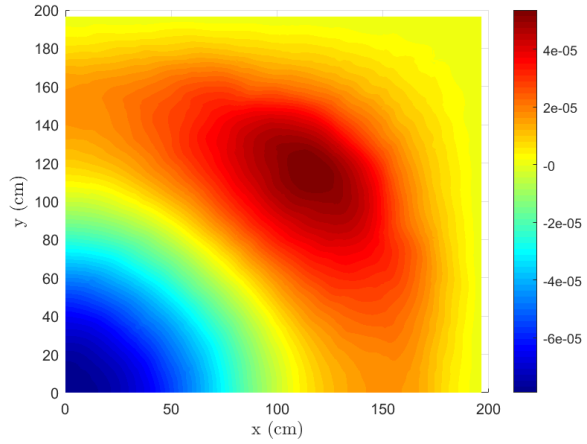


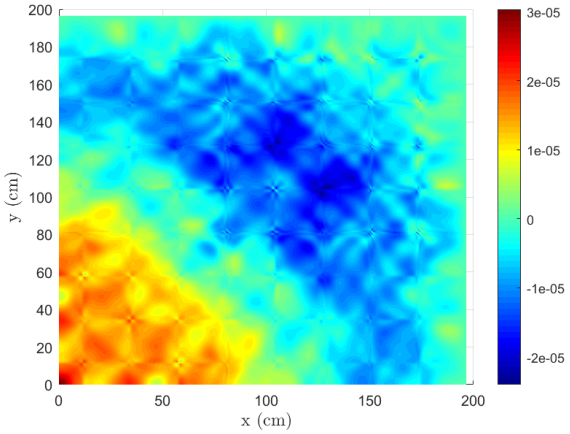
Figure 3.9: Relative eigenvalue and L_2 eigenvector error for one-group BIBLIS problem



(a) Full-order solution



(b) PGD unshifted solution difference



(c) PGD shifted solution difference

Figure 3.10: Full-order (MOOSE) solution and difference from PGD solutions at 272 elements per dimension, one-group BIBLIS problem

Figure 3.11 shows the resulting run-time for each simulation. These run-time results show that PGD has more marginal performance than the previous problem, which is mainly due to the increase in the number of enrichment terms needed to represent the converged solution. Table 3.6 shows the relative computational efficiency for each refinement of the unshifted PGD model and full-order MOOSE model, definition of these numbers are described in Section 3.4.1. It can be seen here that the unshifted PGD efficiency is far smaller than the shifted one, this due to the fact that there is no group coupling and inverting the shifted operator is essentially the same as inverting the

unshifted one. As noted previously, the multi-D solve efficiency gets worse with mesh refinement. Figure 3.12 shows the contribution of each enrichment for the PGD runs at 272 elements per dimension and the magnitude of each term of the SVD of the full-order model solution. The PGD enrichment contributions follow very closely with the SVD reference, but significantly more terms are needed fully characterize the solution when compared to the previous problem.

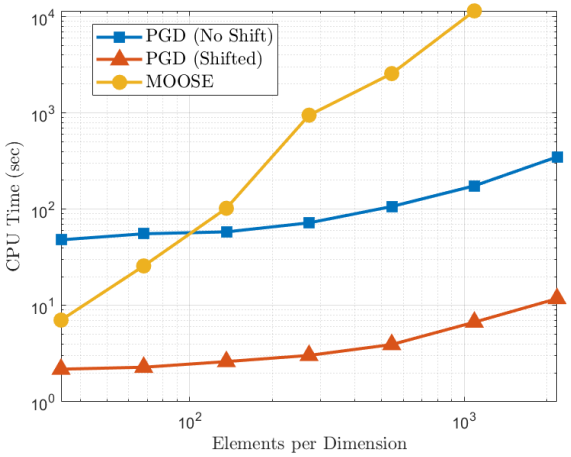


Figure 3.11: Run-time in seconds for each simulation of the one-group BIBLIS problem

Table 3.6: Relative computational efficiency for unshifted PGD and MOOSE, one-group BIBLIS problem (baseline is the PGD method with shift; values < 1 denote a less efficient technique)

Elements per Dim.	PGD	MOOSE
34	0.3737	6.0264
68	0.3384	1.7264
136	0.3699	0.4959
272	0.3441	0.0613
544	0.2971	0.0282
1088	0.2633	0.0089

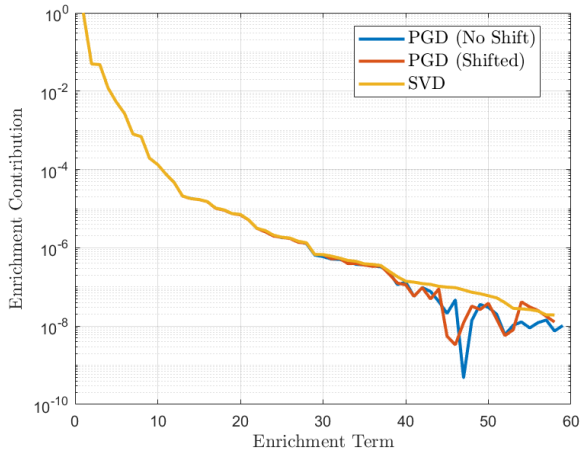


Figure 3.12: Enrichment contribution of converged PGD solutions and SVD of the full-order solution, one-group BIBLIS problem

3.4.3.2 Two-Group Version

To illustrate the performance of PGD for a multigroup example with more realistic reactor heterogeneities, the PGD algorithm is applied to the two-group version of the BIBLIS benchmark. The material property values for this problem can be found in Table 3.7. The full-order model, evaluated using MOOSE, was run with 34 to 1,088 elements per dimension, and the PGD model was run with 34 to 2,176 elements per dimension. The eigenvalue convergence tolerance (ϵ_k) were set to 10^{-9}. A MOOSE model with 2,176 elements per dimension was used for the reference eigenvalue and eigenvector with $k_{\text{eff}} = 1.02510305$; this solution required high performance computing to evaluate, so the performance results for this simulation are not included.

Figure 3.13 shows the resulting eigenvalue and eigenvector error for MOOSE with power iteration and PGD with and without adaptive Wielandt shift. These errors show that both the full-order and PGD models, again, have proper error convergence when the spatial mesh is refined. To compare the solution between each model, Figure 3.14 shows the full-order solutions at 272 elements per dimension with the difference from the PGD models at the same refinement. These plots show that the MOOSE and PGD evaluations converge to approximately the same solution in the power iteration process.

Table 3.7: Material properties of two-group BIBLIS benchmark problem

Material	Group	D^g (cm)	Σ_a^g (cm^{-1})	$\nu\Sigma_f^g$ (cm^{-1})	$\Sigma_s^{g\leftarrow 1}$ (cm^{-1})
1	1	1.4360	0.0095042	0.0058708	0
	2	0.3635	0.0750580	0.0960670	0.017754
2	1	1.4366	0.0096785	0.0061908	0
	2	0.3636	0.0784360	0.1035800	0.017621
3	1	1.3200	0.0026562	0	0
	2	0.2772	0.0715960	0	0.023106
4	1	1.4389	0.0103630	0.0074527	0
	2	0.3638	0.0914080	0.1323600	0.017101
5	1	1.4381	0.0100030	0.0061908	0
	2	0.3665	0.0848280	0.1035800	0.017290
6	1	1.4385	0.0101320	0.0064285	0
	2	0.3665	0.0873140	0.1091100	0.017192
7	1	1.4389	0.0101650	0.0061908	0
	2	0.3679	0.0880240	0.1035800	0.017125
8	1	1.4393	0.0102940	0.0064285	0
	2	0.3680	0.0905100	0.1091100	0.017027

$\chi^1 = 1.0, \chi^2 = 0.0$

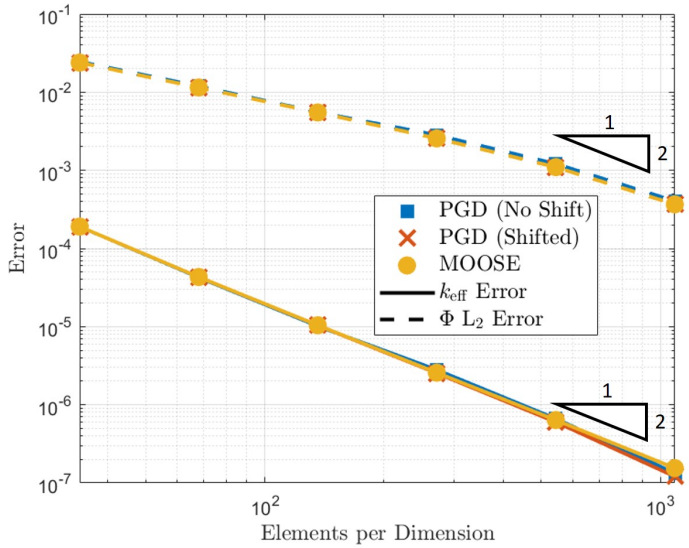


Figure 3.13: Relative eigenvalue and L_2 eigenvector error for two-group BIBLIS problem

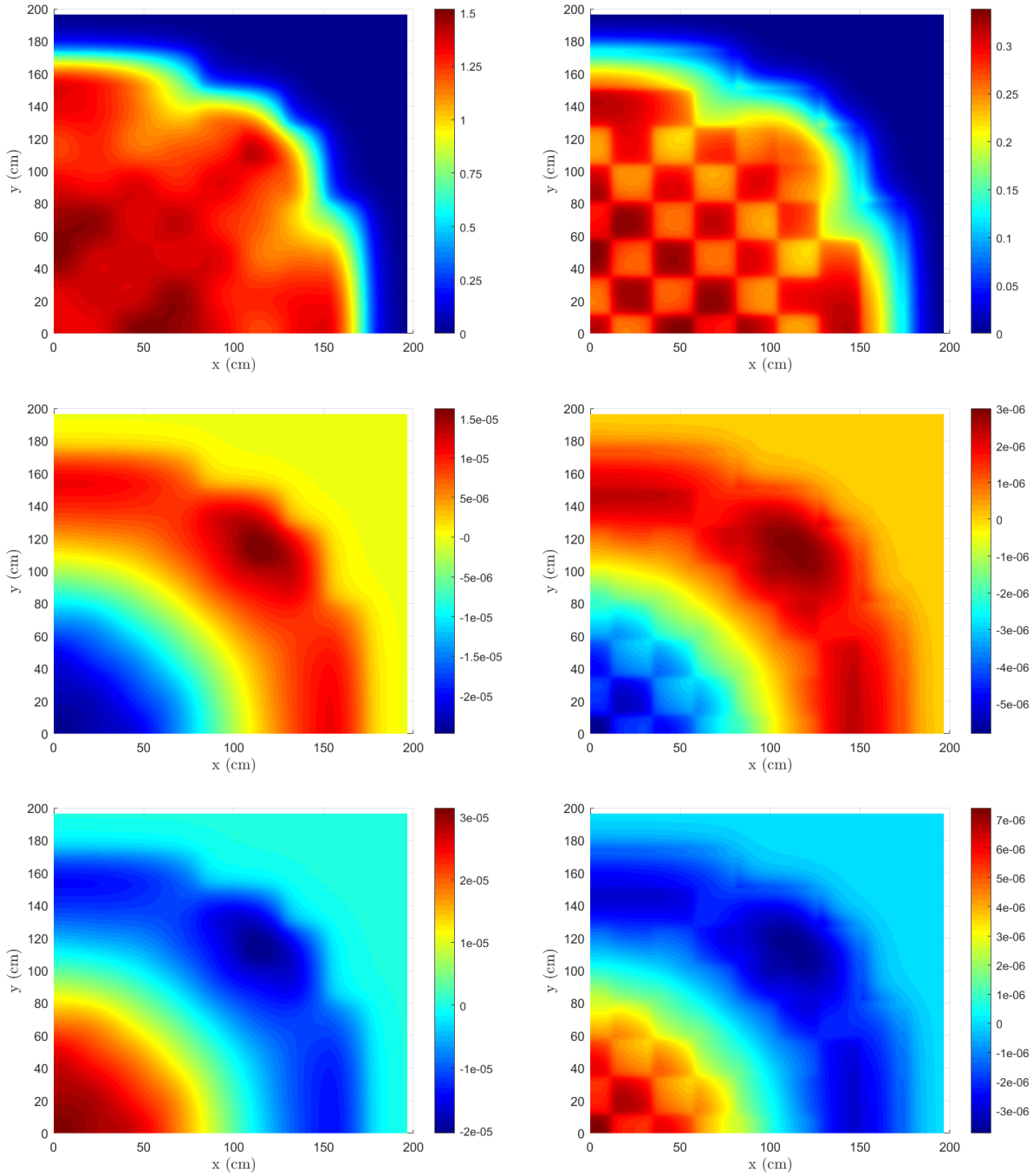


Figure 3.14: Full-order (MOOSE) solution and difference from PGD solutions at 272 elements per dimension, two-group BIBLIS problem. Left column: group 1 flux. Right column: group 2 flux. Top row: solution. Middle row: unshifted PGD difference. Bottom Row: shifted PGD difference

Figure 3.15 shows the resulting run-time for each simulation. These run-time results show that PGD has even more marginal performance than the one-group version. Table 3.8 shows the relative computational efficiency for each refinement of the unshifted PGD model and full-order MOOSE model. Figure 3.16 shows the contribution of each enrichment for the PGD runs at 272 elements per dimension and the magnitude of each term of the SVD of the full-order model solution. SVD has similar enrichment convergence as the one-group version, but PGD does not show the same convergence. From the poorer convergence of the shifted PGD model, the difference from SVD appears to be caused by the coupling between group fluxes in the PGD linear system evaluation. The unshifted operator only has scattering, coupling from group 1 to 2, for a given source term; while the shifted operator has scattering and fission, coupling both groups together. The group-1 convergence is better than the group-2 for unshifted operator because group 1 is independent, but they are approximately the same for the shifted operator.

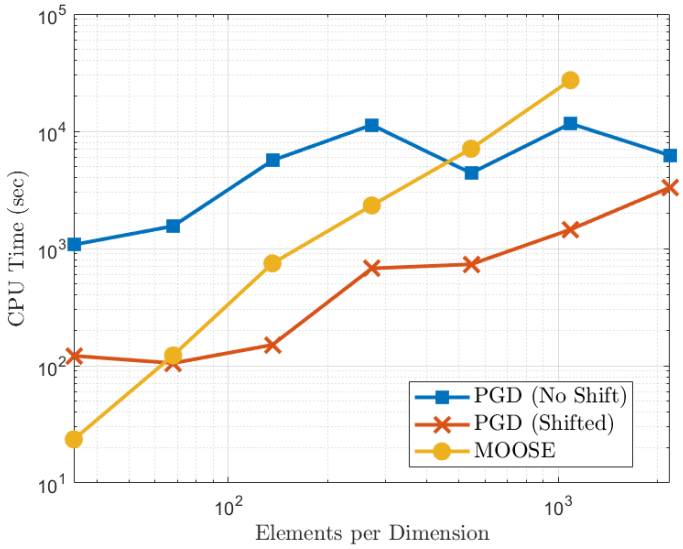


Figure 3.15: Run-time in seconds for each simulation of the two-group BIBLIS problem

Table 3.8: Relative computational efficiency for unshifted PGD and MOOSE, two-group BIBLIS problem (baseline is the PGD method with shift; values < 1 denote a less efficient technique)

Elements per Dim.	PGD	MOOSE
34	1.2906	138.3303
68	0.8220	25.4305
136	0.5271	5.9563
272	0.9528	6.5240
544	1.4271	2.4053
1088	1.2823	0.9485

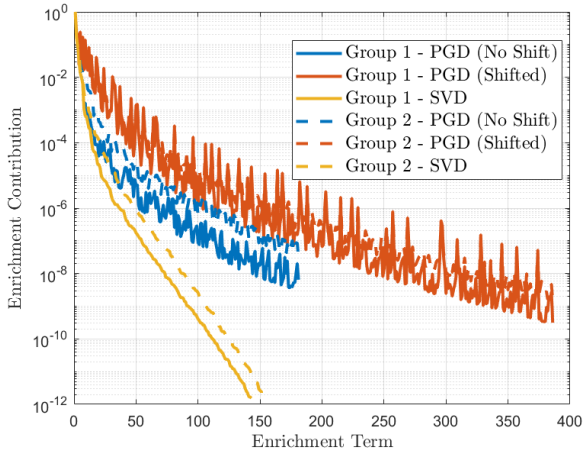


Figure 3.16: Enrichment contribution of converged PGD solutions and SVD of the full-order solution, two-group BIBLIS problem

3.4.4 Analysis of PGD Algorithm

The purpose of this section is to analyze some of the features of the implemented PGD algorithm. These include the shifting parameter, solution compression, eigenvalue projection, and the adaptive enrichment tolerance. The two-group version of BIBLIS example from Section 3.4.3 with 34 elements per dimension will be used for this analysis.

3.4.4.1 Wielandt Shift

As stated in Section 3.2.2, there are many different choices for the Wielandt shift parameter σ. In the previous sections, an adaptive shift was utilized, defined by Equation (3.7). Here the

adaptive shift results are compared with fixed shifts:

- No shift: $\sigma_0 = 0$
- Adaptive shift: $\sigma_{\text{adapt}} \longleftarrow$ Equation (3.7)
- Maximum k_{eff}: $\sigma_{\min} = 1/k_{\max} = 1/3$
- Maximum infinite medium: $\sigma_\infty = 1/\max(k_\infty) = 0.8525$

$\max(k_\infty)$ is determined by the material zone with the largest k_∞ Table 3.9 shows the number of iterations required for 10^{-7}, eigenvalue and eigenvector error, eigenvalue tolerance, total run-time, and number of enrichments required for the last power iteration. These results show that the adaptive shift is the best method for this problem, with a tenth of the number of power iterations compared to no shift. However, these results also indicate that having a shift for this problem increases the run-time per power iteration significantly, due to the increase in the number of enrichment evaluations per power iteration. It is also worth noting that the number of enrichments increases as the efficiency of the shift improves.

Table 3.9: Performance results of PGD with various shifting parameters for two-group BIBLIS problem

Shift Type	P.I.	k_{eff}Error	Φ Error	Run-time (s)	No. of Enrichments
σ_0	117	0.0001906	0.005983	1031	73
σ_{adapt}	11	0.0001906	0.005003	149.9	175
$\sigma_{\min}$	96	0.0001907	0.006334	1523	89
σ_∞	28	0.0001906	0.005100	423.9	108

This is by no means an exhaustive list of all the possible shifting parameters. See [78] and [68] for more options and analysis, especially for criticality problems. It is also important to note that a shift based on the Rayleigh quotient was attempted, explained in Section 3.2.2, but the PGD systems became severely ill conditioned after a few power iterations.

3.4.4.2 Solution Compression, Eigenvalue Projection, and Adaptive Enrichment Tolerance

In Sections 3.3.3 and 3.3.4, several methods are explained to potentially improve PGD performance for generalized eigenvalue problems. This analysis attempts to quantify how much these methods improve our PGD algorithm. Five different variations were run with no shift:

- Type 1 - All methods implemented
- Type 2 - No compression, projection, or adaptive tolerance
- Type 3 - No compression
- Type 4 - No projection
- Type 5 - No adaptive tolerance

Table 3.10 shows the number of iterations required for 10^{-7}, eigenvalue and eigenvector error, eigenvalue tolerance, difference in run-time from Type 1, and number of enrichments required for the last power iteration. These results show that each of the methods implemented improve PGD performance significantly, without affecting the final error of the solution. Most significant is the adaptive enrichment tolerance, which shows that the method prevents “over-solving".

Table 3.10: Performance results of PGD with various efficiency methods turned off for two-group BIBLIS problem. The run-time for the Type 1 simulation is 1031 s.

Type	P.I.	k_{eff}Error	Φ Error	Δ Run-time (s)	No. of Enrichments
1	117	0.0001906	0.005983	0	73
2	139	0.0001906	0.005823	+7586 (+736%)	143
3	117	0.0001906	0.006157	+260 (+25%)	73
4	139	0.0001906	0.005957	+769 (+75%)	77
5	117	0.0001906	0.005970	+4143 (+402%)	143

3.5 Discussion

This chapter investigates the implementation of PGD to multigroup neutron diffusion criticality problems. The eigenproblems are solved using standard power iteration and accelerated power iteration with an adaptive Wielandt shift. PGD is utilized as a linear system solver to obtain the eigenvector at each iteration. The theory sections discuss the PGD solution process and its implementation within the power iteration process. In order to avoid re-evaluating the PGD operators for each spatial dimension, a PGD-favorable decomposition of the multigroup diffusion operators is discussed for an efficient implementation of the technique. This is particularly needed for highly heterogeneous geometries, such as the ones of found in modern nuclear cores. Because the iterative PGD process is located in the innermost loop (power iteration being the traditional outer loop), several techniques to reduce the computational burden of the PGD iterations have been proposed, including compression of the current PGD iteration, solution projection to optimize the current PGD solution, and adaptive tolerance criteria.

The results for this chapter involve four different criticality problems. For comparison purposes, a non-PGD approach is also employed, solved using the MOOSE FEM library. The first test case is a four-group 3D homogeneous bare cube. PGD shows far superior performance compared to MOOSE; this problems admits an (analytical) solution that is separable in space, so the PGD evaluation requires very few terms to converge. The second problem is a two-group, 2-D two-material geometry. The domain is large and mostly consists of one material. Again, PGD shows superior performance compared to MOOSE for this problem because relatively few number of enrichments are needed to represent each power iteration's eigenvector; this problem was used in previous PGD applications [50]. The third and fourth problems have the same highly heterogeneous 2D geometry, representative of a more realistic core layout. For these problems, PGD shows a much more marginal performance compared to the previous examples, especially for the two-group version (fourth problem).

The difference in the PGD performance for these more heterogeneous problems highlights that the effectiveness of PGD may hinge on the problem's heterogeneity. The material-layout depen-

dence is evident when comparing the results from the two-material problem and the two-group BIBLIS problem. Both of the eigenvectors have a similar enrichment convergence, as assessed by an SVD analysis. This is compounded in the multigroup case, for example when comparing the 1-g and 2-g BIBLIS results where about 4 times more enrichments are needed in the PGD process than the ideal value (computed by performing an SVD of the solution). This increase in the number of enrichments renders PGD slower because added enrichments require more alternating direction iterations, each one being more computationally demanding than the previous one (residual evaluation with more terms). In addition, the multigroup coupling dependence is evident when comparing the one-group and two-group versions of the BIBLIS problem. The SVD convergence for these problems are very similar. However, the PGD enrichment convergence for the two-group version is much slower than the one-group version.

To conclude, PGD can be an effective technique for a certain class of neutron diffusion problems. Some intuition must be applied concerning the nature of the resulting solution in order to determine if PGD will have better performance than a full multi-D model evaluation. In general, this PGD approach performs more poorly for highly heterogeneous geometries. For more homogeneous geometries, PGD appears to always outperform its full-order modeling counterpart in a significant manner.

4. SPACE-ENERGY SEPARATED REPRESENTATIONS FOR MULTIGROUP NEUTRON DIFFUSION USING PROPER GENERALIZED DECOMPOSITION

4.1 Introduction

In this chapter, the PGD methodology is formulated to include the energy dimension of neutron diffusion into the PGD separated representation. This application serves as an extension to the PGD approaches from Chapter 3 and the work of Senecal & Wi [50]. The model problem in this work is the fixed-source, steady-state multigroup neutron diffusion equation. The variable of interest is the space-energy dependent neutron flux ($\phi(\vec{r}, E)$). The work by Senecal & Ji used the PGD methodology to seek a space-only separated representation for each group flux. This work takes a different approach, whereby seeking a full space-energy separated representation. The chapter analyzes the properties of such a phase-space PGD decomposition and provide results for various energy discretizations, from 2 groups to 145 groups.

This chapter includes three key extensions of the PGD method. Firstly, it includes a derivation of the construction of a PGD operator for space-energy separated representation that decomposes the four-dimensional (space$\times$ group) cross sections. This operator construction allows for the construction of 1-D linear systems before the PGD solution process is initiated; this is especially optimized for highly heterogeneous two- and three-dimensional spatial geometries. Second, PGD is applied to a variety of multigroup neutron diffusion problems, ranging from a course two-group problem to a fine-group problem on the order of 100 energy groups. Finally, this chapter includes a characterization of the performance of PGD for these problems, remarking on its capability for realistic calculations.

4.2 PGD for Neutron Diffusion

The model problem for this work is the steady-state neutron diffusion equation with a fixed-source, shown in Equation (4.1).

$$-\nabla\cdot D(\vec{r},E)\nabla\phi(\vec{r},E)+\Sigma_t(\vec{r},E)\phi(\vec{r},E)=\int_0^\infty \Sigma_s(\vec{r},E'\rightarrow E)\phi(\vec{r},E')dE' + \chi(E)\int_0^\infty \nu\Sigma_f(\vec{r},E')\phi(\vec{r},E')dE'+Q(\vec{r},E)\ . \quad (4.1)$$

The most common discretization technique for the energy dimension is multigroup, whereby separating the flux and cross sections into energy groups:

$$\phi(\vec{r},E)=\phi^g(\vec{r})\ ,\quad E\in[E_g,E_{g-1}]\ . \quad (4.2)$$

With this discretization, Equation (4.1) becomes the multigroup neutron diffusion equations:

$$-\nabla\cdot D^g(\vec{r})\nabla\phi^g(\vec{r})+\Sigma_t^g(\vec{r})\phi^g(\vec{r})=\sum_{g'=1}^{G}\Sigma_s^{g\leftarrow g'}(\vec{r})\phi^{g'}(\vec{r}) + \chi^g\sum_{g'=1}^{G}\nu\Sigma_f^{g'}(\vec{r})\phi^{g'}(\vec{r})+Q^g(\vec{r})\ ,\quad g=1,...,G\ . \quad (4.3)$$

For brevity in the remaining sections and without loss of generality, all the reaction terms are combined into a single cross section,

$$\Sigma^{gg'}(\vec{r})=\Sigma_t^g(\vec{r})\delta_{gg'}-\Sigma_s^{g\leftarrow g'}(\vec{r})-\chi^g\nu\Sigma_f^{g'}(\vec{r})\ . \quad (4.4)$$

The multigroup neutron diffusion equations then become:

$$-\nabla\cdot D^g(\vec{r})\nabla\phi^g(\vec{r})+\sum_{g'=1}^{G}\Sigma^{gg'}(\vec{r})\phi^{g'}(\vec{r})=Q^g(\vec{r})\ ,\quad g=1,...,G\ . \quad (4.5)$$

With multigroup discretization, there are two different ways to perform the PGD separation of

variables: First, one may perform PGD for each multigroup flux separately,

$$\phi^g(x,y) = \sum_{i=1}^{N} X_i^g(x) Y_i^g(y) \ , \quad g = 1, ..., G \ , \tag{4.6a}$$

or the entire phase-space (spatial coordinates + energy variables) can be treated with PGD all at once

$$\phi(x,y,E) = \sum_{i=1}^{N} X_i(x) Y_i(y) \mathcal{E}_i(\boldsymbol{E}) \ , \boldsymbol{E} = [E_G, ..., E_1] \tag{4.6b}$$

Equation (4.6a) represents a PGD representation in space only, this type is implemented in the previous chapter, for example. The representation in Equation (4.6b) is a full dimensional decomposition and is the subject of analysis in this work. The following sections detail the construction of a PGD operator for the multigroup neutron diffusion with space-energy separation.

4.2.1 Operator Construction for Space-Only Separation

This section gives a brief overview on the construction of PGD operators for space-only separated multigroup neutron diffusion (Equation (4.6a)). In this representation, each group-by-group operator must be decomposed separately in the form of Equation (2.2b). Utilizing the method of decomposing material properties in Section 2.3.1, the linear operator for group g' to g becomes:

$$\begin{aligned}\mathcal{L}^{gg'}(x,y)\phi^{g'}(x,y) = \sum_{i=1}^{N}\sum_{\ell=1}^{L^{gg'}} \mathcal{L}_\ell^{gg',x}(x) X_i^{g'}(x) \mathcal{L}_\ell^{gg',y}(y) Y_i^{g'}(y) = \\ - \sum_{i=1}^{N} \left\{ \sum_{\ell=1}^{M_d^g} \left[D_\ell^{g,x} \frac{d^2 X_i^g}{dx^2} D_\ell^{g,y} Y_i^g + D_\ell^{g,x} X_i^g D_\ell^{g,y} \frac{d^2 Y_i^g}{dy^2} \right] \delta_{gg'} \right. \\ \left. + \sum_{\ell=1}^{M_\sigma^{gg'}} \Sigma_\ell^{gg',x} X_i^{g'} \Sigma_\ell^{gg',y} Y_i^{g'} \right\} \ . \end{aligned} \tag{4.7}$$

The group solution and group-by-group operator can then be concatenated to construct a single system:

$$(\boldsymbol{\mathcal{L}})_{gg'} = \mathcal{L}^{gg'} = \sum_{\ell=1}^{L^{gg'}} \mathcal{L}_\ell^{gg',x} \otimes \mathcal{L}_\ell^{gg',y} \ , \tag{4.8a}$$

$$(\Phi)_g = \phi^g = \sum_{i=1}^{N} X_i^g \otimes Y_i^g \ . \tag{4.8b}$$

With this multigroup construction of the PGD operator, and similarly for the source, the PGD solution process from Section 2.1 can be applied directly.

4.2.2 Operator Construction for Space-Energy Separation

This section gives an overview on the construction of PGD operators for space-energy separated multigroup neutron diffusion (Equation (4.6b)). In order represent the operator shown in Equation (2.2b), the domain describing each material property must be decomposed in products of one-dimension functions, similarly to the flux representation of Equation (4.6b). For a space-energy dependent material property, the separated representation is,

$$k(x,y,E) = \sum_{i=1}^{M} k_i^{x,y}(x,y) k_i^e(E) = \sum_{i=1}^{M} \left(\sum_{j=1}^{M_i} k_j^x(x) k_j^y(y) \right) k_i^e(E) \ . \tag{4.9}$$

In this representation, $k_i^e(E)$ is the energy-dependent property value of material zone i and $k_i^{x,y}(x,y)$ describes the position of that zone. $k_i^{x,y}(x,y)$ is obtained by utilizing the process in Section 2.3.1 with material property in zone i set to 1 and all others to 0. In the simple example of Figure 2.6, one can readily obtain the spatial representation of k_1, i.e., k_1^{xy} from Equation (4.9) by setting $k_1 = 1$ and $k_2 = 0$. For the application of multigroup material properties, k_i^e becomes a matrix of size $G \times G$. For the diffusion coefficient: $(k_i^e)_{gg'} \rightarrow D_i^g \delta_{gg'}$. For the combined cross section: $(k_i^e)_{gg'} \rightarrow \Sigma_i^{gg'}$.

The resulting description of the linear operator is shown in Equation (4.10), where $\hat{M}$ is the number of terms required to fully represent each material property, $\hat{M} = \sum_{i=1}^{M} M_i$ from Equation (4.9). Consequently, $L = 3\hat{M}$ from Equation (2.2b), and more generally in `dim` spatial dimensions, $L = (\texttt{dim} + 1) \times \hat{M}$.

$$\mathcal{L}_\ell^x X_i \mathcal{L}_\ell^y Y_i \mathcal{L}_\ell^e \mathcal{E}_i = -\frac{d}{dx}\left[D_\ell^x(x) \frac{dX_i}{dx} \right] D_\ell^y(y) Y_i \boldsymbol{D}_\ell \mathcal{E}_i, \tag{4.10a}$$

$$\mathcal{L}^x_{\ell+\hat{M}} X_i \mathcal{L}^y_{\ell+\hat{M}} Y_i \mathcal{L}^e_{\ell+\hat{M}} \mathcal{E}_i = -D^x_\ell(x) X_i \frac{d}{dy}\left[D^y_\ell(y)\frac{dY_i}{dy}\right] \boldsymbol{D}_\ell \mathcal{E}_i, \tag{4.10b}$$

$$\mathcal{L}^x_{\ell+2\hat{M}} X_i \mathcal{L}^y_{\ell+2\hat{M}} Y_i \mathcal{L}^e_{\ell+2\hat{M}} \mathcal{E}_i = \Sigma^x_\ell(x) X_i \Sigma^y_\ell(y) Y_i \boldsymbol{\Sigma}_\ell \mathcal{E}_i, \tag{4.10c}$$

with

$$\ell = 1, ..., \hat{M} \ .$$

In multigroup discretization, the energy dimension operators $\boldsymbol{D}_\ell$ and $\boldsymbol{\Sigma}_\ell$ are matrices describing the within group and cross group material properties:

$$(\boldsymbol{D}_\ell)_{gg'} = D^g_i \delta_{gg'} \ , \tag{4.11a}$$

$$(\boldsymbol{\Sigma}_\ell)_{gg'} = \Sigma^g_{t,i}\delta_{gg'} - \Sigma^{g\leftarrow g'}_{s,i} - \chi^g \nu \Sigma^{g'}_{f,i} \ , \tag{4.11b}$$

for

$$\sum_{i'=1}^{i-1} M_{i'} < \ell \leq \sum_{i'=1}^{i} M_{i'} \ .$$

With this space-energy separated construction of the PGD operator, and similarly for the source, the PGD solution process from Section 2.1 can be applied directly.

4.2.3 Multi-Space-Energy Separation

This section introduces multi-space-energy PGD, which is where different space-energy separated representations are sought for the neutron flux for different energy ranges. The impetus of this modified decomposition is that the spatial profile of the neutron flux can vary significantly for different energy groups. The worst case scenario is that each group flux has a completely different separated representation, which would require an unwieldy number of enrichments while using a unique space-energy decomposition. In the four-group problem in [84] there is a significant difference in the spatial profile between the fast and thermal energy ranges. In [85] there is a difference between the fast, epithermal, and thermal energy ranges. However, the fact that, for energy groups within a certain energy range, the flux spectrum exhibits similar profiles provides an opportunity for space-energy decomposition to prove effective, especially as the number of energy

groups increases.

For instance, the energy-dependent neutron flux can be separated into fast and thermal energy ranges, which is analogous to multigroup:

$$\phi(\vec{r}, E) = \begin{cases} \phi^{\text{fst}}(\vec{r}, E) & E \in \text{fast range} \\ \phi^{\text{th}}(\vec{r}, E) & E \in \text{thermal range} \end{cases} . \tag{4.12}$$

A space-energy separation is then sought for each of these energy ranges, which in multigroup discretization looks like:

$$\phi^{\text{fst}}(x, y, E) = \sum_{i=1}^{N} X_i^{\text{fst}}(x) Y_i^{\text{fst}}(y) \mathcal{E}_i^{\text{fst}}(\boldsymbol{E}_{\text{fst}}) \, , \boldsymbol{E}_{\text{fst}} = [E_{\text{fst}}, ..., E_1] \, , \tag{4.13a}$$

$$\phi^{\text{th}}(x, y, E) = \sum_{i=1}^{N} X_i^{\text{th}}(x) Y_i^{\text{th}}(y) \mathcal{E}_i^{\text{th}}(\boldsymbol{E}_{\text{th}}) \, , \boldsymbol{E}_{\text{th}} = [E_G, ..., E_{\text{th}}] \, , \tag{4.13b}$$

where E_{fst} is the lowest energy group of the fast region and E_{th} is the highest energy group of the thermal region. The linear operator is defined by combining the methodologies from Section 3.3.5 and Section 4.2.2. A separate linear operator is found for each energy region and the coupling between them, in the form similar to Equation (4.10). These operators are then concatenated similarly to Equation (4.8). Thus, the PGD solution process in Section 2.1 can be applied.

It should be noted that the differentiation between fast and thermal energy ranges is for illustration; separation into any number of energy ranges is possible and is straightforward to infer from the above example.

4.3 Results

In order to test our PGD implementation with space-energy separation, five different examples are applied. The first example is a mildly heterogeneous 2-D two-group problem that is analyzed as a criticality problem in Section 3.4.2. The second example is a 2-D, two-group heterogeneous reactor geometry based on the IAEA benchmark in the ANL Benchmark Problem book [58]. The third and fourth examples are 3-D, seven-group problems based on an example from [86]. Each

of these four problems were evaluated using three different techniques: full-order modeling using MOOSE [87], PGD with space-only separation (PGD-MG, Equation (4.6a)), and PGD with space-energy decomposition (PGD-DE, Equation (4.6b)). The fifth example is a 3-D 145-group graphite block with an embedded AmBe source. This fourth example only analyzes the PGD-DE technique and full-order model is evaluated using the `deal.II` library, more details on this implementation are in Section 4.3.4.

The PGD implementations use CFEM for each of its 1-D spatial functions. The full-order model (FOM) is applied with an equivalent multi-dimensional CFEM discretization. The linear systems incurred by the PGD solution process are evaluating using MATLAB's LU solver. The MOOSE linear solves are performed using the generalized minimal residual method (GMRES) with algebraic multi-grid (AMG) techniques from the HYPRE package with default numerical parameters.

Two different types of error are analyzed in the following results. One is a full phase-space error defined in Equation (4.14a). The other is a group dependent error, defined by Equation (4.14b).

$$\text{Error} = \frac{\sum_{g=1}^{G} ||\phi^g - \phi^g_{\text{ref}}||_{L_2}}{\sum_{g=1}^{G} ||\phi^g_{\text{ref}}||_{L_2}} \,, \tag{4.14a}$$

$$\epsilon(E) = \frac{||\phi^g - \phi^g_{\text{ref}}||_{L_2}}{||\phi^g||_{L_2}} \,, \quad E \in [E_g, E_{g-1}] \,. \tag{4.14b}$$

4.3.1 2-D Two-Group Problem

This example is a simple heterogeneous 2-D, two-group example based on Problem 5.7 from [83]; the criticality version of this problem was analyzed in Section 3.4.2. The geometry is shown by Figure 3.4. The material properties can be found in Table 3.3 with the addition of a group one source term in Material 1, $Q^1 = 0.01$. The full-order model and both PGD models involve a uniform mesh and run with various levels of refinement, ranging from 35 to 2240 elements in the x and y dimensions. The MOOSE linear tolerance was set to 10^{-5} and the PGD enrichment tolerance was set to 10^{-6}. To compute error, a reference solution (ϕ^g_{ref}) is used from evaluating the

full-order model on a 2240-by-2240 element mesh.

Table 4.1 shows the error of each simulation and number of enrichments required for the PGD methods. Figure 4.1 shows the run-time for each simulation. All the methods show very similar error, but the run-times for PGD-MG are significantly lower than the MOOSE and PGD-DE evaluations. Although both PGD methods are significantly faster than MOOSE at higher refinements, the space-energy separated PGD suffers heavily from the increase in required enrichments. The increase in the number of enrichments is due to the addition of the energy dimension and this two-group addition does not significantly decrease the dimensionality of the PGD solution.

Table 4.1: Resulting error and number of enrichments required for 2-D two-group example

Elem. per Dim.	MOOSE	PGD-MG		PGD-DE	
(x, y)	Error	Error	Enrichments	Error	Enrichments
(35,35)	0.002067	0.002068	14	0.001941	237
(70,70)	0.0005192	0.0005197	17	0.0004721	282
(140,140)	0.0001297	0.0001305	16	7.778e-05	222
(280,280)	3.207e-05	3.245e-05	12	2.788e-05	197
(560,560)	7.637e-06	9.029e-06	12	6.598e-06	241
(1120,1120)	1.527e-06	3.72e-06	15	9.045e-07	246

4.3.2 Two-Group Reactor Geometry

This example involves a 2-D heterogeneous reactor geometry with two-group cross sections. The chosen problem geometry is based on the ANL Benchmark Problem Book problem 11-A1 [58]. The one-group version of this problem is analyzed in Section 2.3.2. The 2-D geometry is shown in Figure 2.8. The two-group material properties are detailed in Table 4.2.

The system was evaluated with the three techniques on various levels of spatial refinement. Figure 4.2 shows the run-time for each method and each refinement level. Table 4.3 details resulting error for each evaluation, compared to a highly refined full-order solution, as well as number of enrichments required for the PGD methods. As expected, the MOOSE run-times have a quadratic dependence on the number of spatial elements per dimension, while the PGD methods have a linear

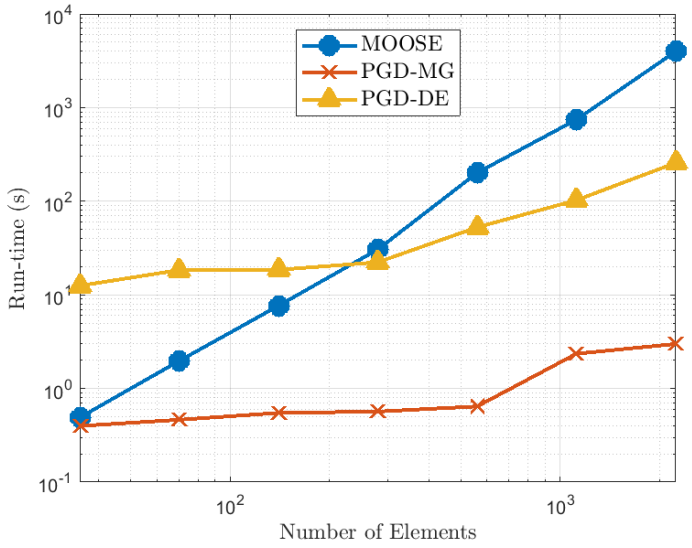


Figure 4.1: Run-time results for 2-D two-group example

Table 4.2: Material properties of two-group, 2-D IAEA Benchmark problem

Material	Group	D	Σ_r	$\Sigma_s^{2\leftarrow1}$	$\chi^1\nu\Sigma_f^2$	Q
Fuel 1	1	1.5	0.03	0.02	0.135	0
	2	0.4	0.22			10
Fuel 1 + Rod	1	1.5	0.03	0.02	0.135	0
	2	0.4	0.265			10
Fuel 2	1	1.5	0.03	0.02	0.135	0
	2	0.4	0.215			10
Fuel 1 + Rod	1	2.0	0.04	0.04	0	0
	2	0.3	0.01			0

dependence. Although both PGD methods are significantly faster than MOOSE at higher refinements, the space-energy separated PGD suffers heavily from the increase in required enrichments, due to addition of the energy dimension in the decomposition.

4.3.3 3-D Seven-Group Example

These next two examples are based on the problem in Section 4.3 of [86], where the geometry and material properties can be found. Because the multiplicity for this problem is greater than 1, all the fission cross sections are halved to avoid negative flux values. Two different instances

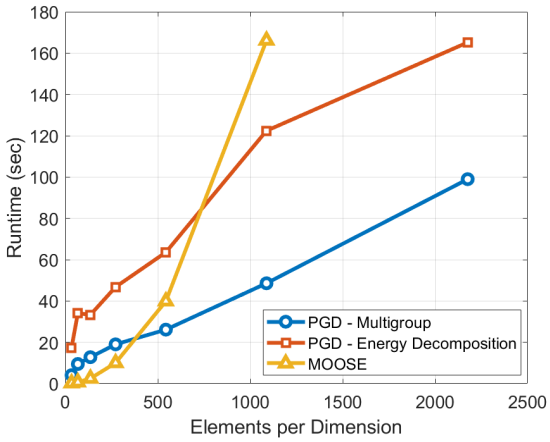


Figure 4.2: Runtime results for the 2-D IAEA Problem

Table 4.3: Relative error of fast group flux for various refinements. Error defined by Equation (4.14a)

Elm. per Dim. (x, y)	Error		
	MOOSE	PGD-MG	PGD-DE
(34,34)	1.229E-2	1.227E-2 (99)	1.229E-2 (161)
(68,68)	4.059E-3	4.076E-3 (118)	4.001E-3 (177)
(136,136)	1.263E-3	1.193E-3 (112)	1.330E-3 (149)
(272,272)	4.634E-4	3.903E-4 (97)	6.391E-4 (137)
(544,544)	2.761E-4	3.759E-4 (91)	3.756E-4 (136)

$(\cdot)$ Indicates the number of enrichment terms needed

of the geometry are considered. First, the fuel region is identical, but the reflector is extended to make the geometry rectangular. Second, to see how PGD performs in a more homogeneous 3-D geometry, the fuel region is homogenized so that the entire region consists of only Material 1. The full-order model and both PGD models involve a uniform mesh and run with various levels of refinement. The full-order model was run with (17,17,20) to (68,68,80) elements in (x, y, z). The PGD models were run with (17,17,20) to (136,136,160) elements. The MOOSE linear tolerance was set to 10^{-5} and the PGD enrichment tolerance was set to 10^{-6}. To compute error, a reference solution (ϕ^g_{ref}) is used from evaluating the full-order model on a 272-by-272-by-320 element mesh. Due to the burden of evaluating this reference model, it was run on a high performance computer

with 8 processors; consequently, the performance of this evaluation is not included.

Table 4.4 shows the error of each simulation and number of enrichments for the fully heterogeneous fuel region example. Figure 4.3a shows the run-time for each simulation in this geometry. For this example, MOOSE greatly out performs the PGD methods because the number of enrichments required for PGD to evaluate is quite large, especially for the PGD-DE method. However, if extrapolating the MOOSE run-time for higher refinements based on AMG scaling properties, PGD would have eventually out performed MOOSE. The difference in run-time between PGD-MG and PGD-DE is marginal. In this seven-group example, PGD-DE reduces the dimensionality significantly for a given enrichment term. However, the larger number of enrichments required for PGD-DE eliminates any run-time savings from this reduction.

Table 4.5 shows the error of each simulation and number of enrichments for the homogeneous fuel region example. Figure 4.3b shows the run-time for each simulation in this geometry. The MOOSE performance for this example is identical to the previous example because the size of the system is unchanged. However, the performance of PGD is significantly improved because this more homogeneous material requires less enrichments to evaluate. Furthermore, the difference in the number of enrichments between PGD-DE and PGD-MG is smaller, which is the reason PGD-DE is faster in this example.

Table 4.4: Resulting error and number of enrichments for 3-D seven-group example with heterogeneous fuel region. Error defined by Equation (4.14a)

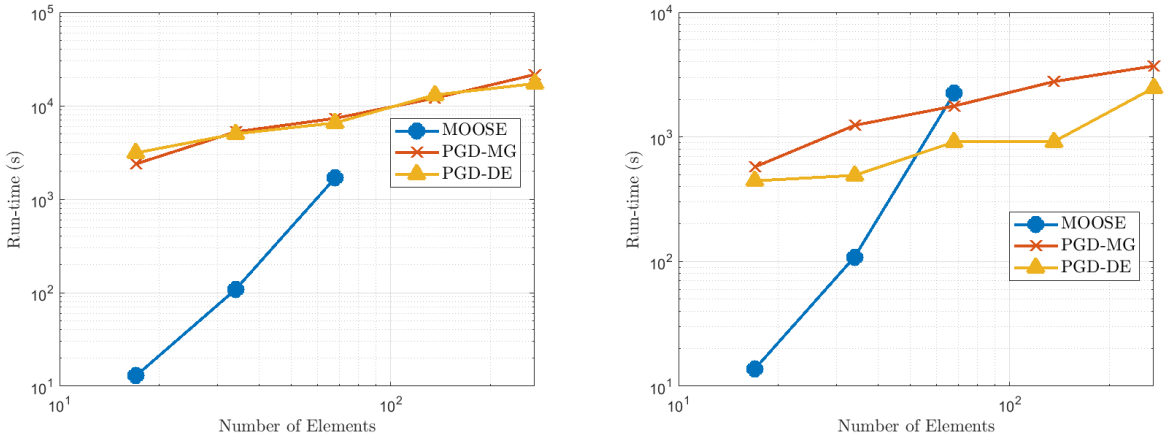
Elm per Dim.	MOOSE	PGD-MG		PGD-DE	
(x, y, z)	Error	Error	Enrichments	Error	Enrichments
(17,17,20)	0.08994	0.07162	333	0.07178	1537
(34,34,40)	0.05564	0.02382	468	0.02396	1737
(68,68,80)	0.004522	0.006574	492	0.007089	1577
(136,136,160)	—	0.001518	501	0.003165	1727

Values with — could not be run because system did not have enough memory

Table 4.5: Resulting error and number of enrichments for 3-D seven-group example with homogeneous fuel region. Error defined by Equation (4.14a)

Elm per Dim. (x, y, z)	MOOSE Error	PGD-MG		PGD-DE	
		Error	Enrichments	Error	Enrichments
(17,17,20)	0.07073	0.07073	192	0.07098	633
(34,34,40)	0.02295	0.02295	264	0.02298	790
(68,68,80)	0.006109	0.006116	277	0.006555	413
(136,136,160)	—	0.001321	285	0.002162	506

Values with — could not be run because system did not have enough memory



(a) Heterogeneous fuel region

(b) Homogeneous fuel region

Figure 4.3: Run-time results for 3-D seven-group examples

4.3.4 145-Group Graphite Block with AmBe source

This example involves a 3-D graphite block with 145-group cross sections, the source of neutrons comes from a small AmBe block embedded in the graphite. The geometry is shown in Figure 4.4. The materials assumed to have no fission, total cross section is shown in Figures 4.5a and 4.5b, and the AmBe source spectrum is shown in Figure 4.5c.

The full-order model solution was evaluated using a specialized `deal.II` implementation [88], most similar to the step-28 tutorial, which is described in [86]. Typically, this type of simulation utilizes a block Gauss-Seidel iteration to resolve the thermal upscattering. However, this

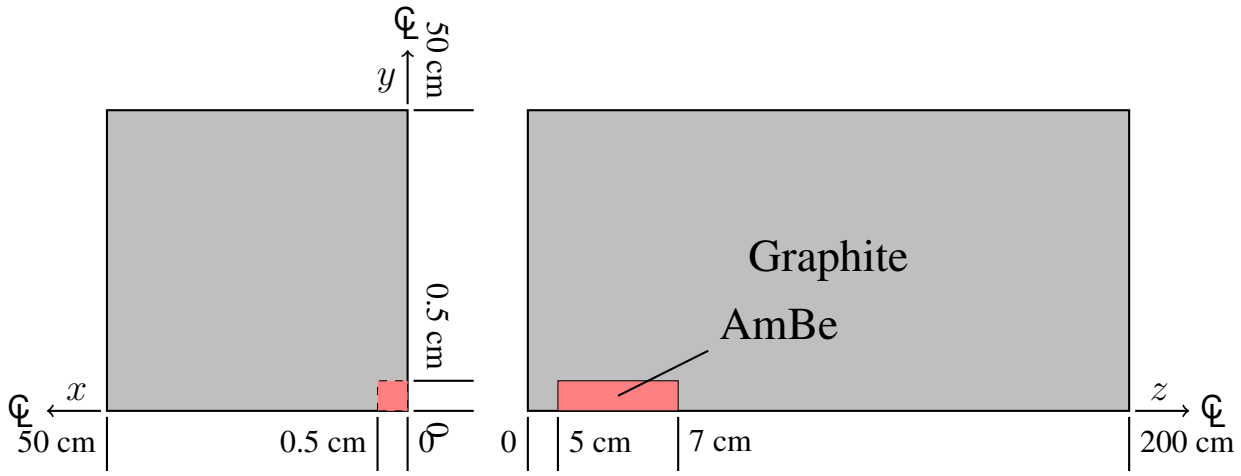


Figure 4.4: 3-D graphite block geometry (not to scale)

technique's spectral radius for the corresponding infinite medium version of this problem is approximately 0.99. Therefore, we elected to utilize the two-grid acceleration method, which has a corresponding spectral radius of 0.53 [89, 90]. The full-order model utilizes a conjugate gradient (CG) solver for each group flux with an AMG preconditioner. The PGD implementation uses a direct LU solver for the energy dimension linear system evaluation.

Four types of PGD separation are implemented for this problem: full space-energy decomposition (Equation (4.6b), PGD-DE) and three different mult-space-energy decomposition (Equation (4.13), PGD-MSE). The three different range separations for PGD-MSE are:

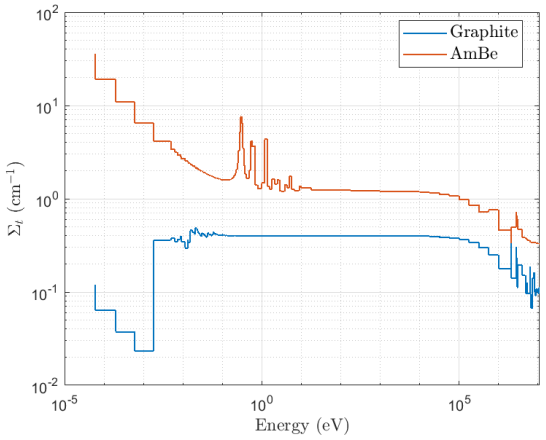
1. Separation of pure down-scattering and upscattering regions. The down-scattering region is defined as groups 1 through 39 (11.0 MeV $\geq E >$ 316 eV) and the upscattering region is defined as groups 40 through 145 (316 eV $\geq E >$ 0 eV).

2. Separation of the graphite resonance region and thermal region. The resonance region is defined as groups 1 through 26 (11.0 MeV $\geq E >$ 0.562 MeV) and the thermal region is defined as groups 27 through 145 (0.562 MeV $\geq E >$ 0 MeV).

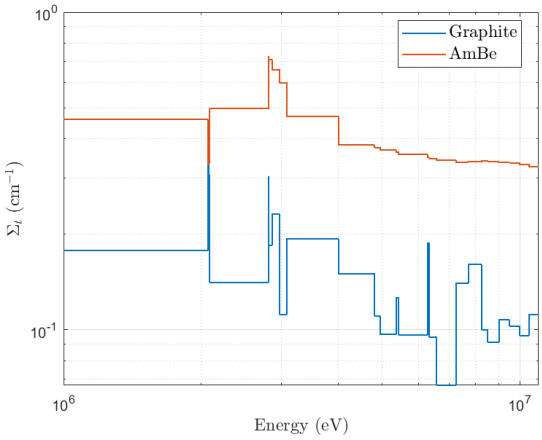
3. Separation of resonance region, pure down-scattering region, and upscattering region: 1 through 26 (11.0 MeV $\geq E >$ 0.562 MeV), groups 27 through 39 (0.562 MeV $\geq E >$ 316

eV), and groups 40 through 145 (316 eV $\geq E >$ 0 eV).

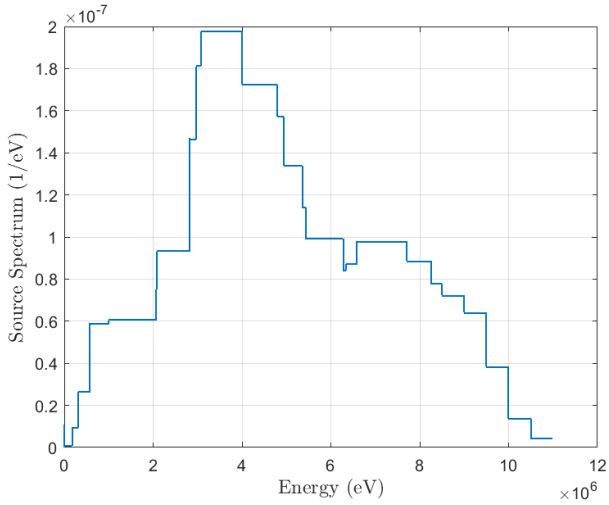
The geometry was discretized using a uniform grid of 100 elements in the x and y direction and 400 in the z direction. The two-grid accelerated Gauss-Seidel iteration tolerance for the full-order model was set to 10^{-6}, which resulted in 21 thermal upscattering iterations, with a total of 2,286 single-group diffusion solves for the entire problem. The PGD enrichment tolerance was set to 10^{-7} for all types. The full-order evaluation was performed with domain decomposition across 12 processors. The PGD evaluation was performed on a single processor in MATLAB.



(a) Total cross section



(b) Total cross section in fast region



(c) AmBe source spectrum

Figure 4.5: Material properties for graphite block example

Table 4.6 details the performance of each method and the error for each of the PGD methods. Error is a comparison between the full-order `deal.II` solution (FOM) and the resulting PGD solutions. From these results, all PGD methods show far superior performance compared to the `deal.II` evaluation. All the errors are lower than tolerance set for the Gauss-Seidel iteration and the run-time is less than 10% of the full-order evaluation. Additionally, without two-grid acceleration, the full-order evaluation would have likely required an order of magnitude more diffusion solves and increasing the run-time to the scale of weeks. However, there appears to be no significant difference in the performance between each of the PGD methods for this problem. The PGD-MSE approaches indicate that PGD has an exorbitant amount of difficulty resolving the fast region, requiring four to five times the number of enrichments as the thermal region. Furthermore, splitting between the graphite resonance and slowing down regions actually impairs PGD performance significantly, indicated by the Type 2 and 3 results. To further the analysis, the individual multigroup flux errors across the spectrum are inspected.

Table 4.6: Performance and error comparison of full-order model evaluation and PGD with space-energy separation for 145-group graphite problem. Number of enrichments are for each energy range. Error computed with FOM solution as ϕ_{ref}.

System Evaluation Type	Enrichment Tolerance	Run-time (sec)	Error Equation (4.14a)	Number of Enrichments
FOM	—	1.61e+05	—	—
PGD-DE	10^{-5}	197	6.387e-05	289
	10^{-6}	1860	5.595e-06	1027
	10^{-7}	7920	8.159e-07	2337
PGD-MSE Type 1	10^{-5}	252	2.992e-05	320, 90
	10^{-6}	1305	3.935e-06	898, 162
	10^{-7}	7490	9.996e-08	2290, 402
PGD-MSE Type 2	10^{-5}	339	3.099e-05	277, 253
	10^{-6}	2443	4.290e-06	710, 883
	10^{-7}	1.14e+04	6.862e-07	1695, 2017
PGD-MSE Type 3	10^{-5}	312	2.186e-05	277, 252, 71
	10^{-6}	1723	3.599e-06	710, 548, 117
	10^{-7}	9070	1.245e-07	1695, 1367, 389

Figures 4.6a and 4.6b show the resulting space integrated spectrum for the full energy range and fast region, respectively. The integrated spectrum is defined as:

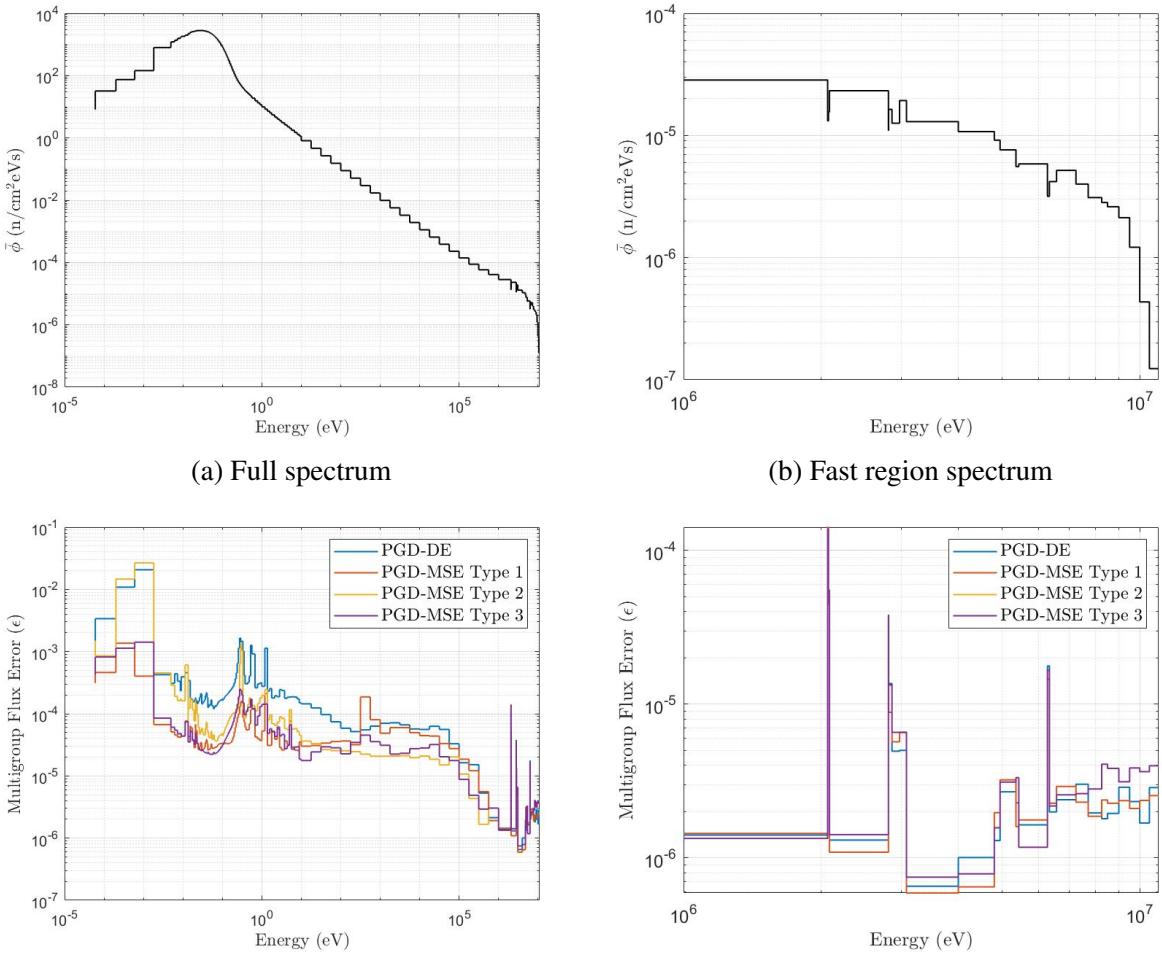
$$\bar{\phi}(E) = \frac{\int_{\Omega_x}\int_{\Omega_y}\int_{\Omega_z}\phi^g(x,y,z)dxdydz}{\Omega_x \times \Omega_y \times \Omega_z}\frac{1}{\Delta E_g}\ , \quad E \in [E_g, E_{g-1}]\ . \tag{4.15}$$

Figures 4.6c and 4.6d show the multigroup flux errors in the corresponding regions. These results show that all the PGD methods have the same difficulty resolving the fast region, which is where PGD-MSE spends most of its time solving. Each method especially has difficulty solving the flux at very specific energies. The peaks in error in Figure 4.6d align with the graphite resonances (Figure 4.5b) and the dips in the flux spectrum (Figure 4.6b). The peaks in the thermal spectrum align with the AmBe resonances in Figure 4.5a. The flux in the thermal region is more accurate when utilizing PGD-MSE. Additionally, the thermal region solution is improved when separating between pure down-scattering and upscattering regions versus thermal-resonance separation.

The PGD-MG approach was also attempted for this 145-group problem, but the process took too long to produce viable results. One enrichment with this method took 3 hours and evaluating only the group 1 flux required 100 enrichments. Therefore, a very conservative estimate for the run-time with PGD-MG is about two weeks. Thus, a PGD approach which includes the energy dimension is a significantly more efficient approach than a PGD solution tackling only the spatial coordinates, for large number of energy groups.

4.4 Discussion

This chapter introduces and investigates the utilization of proper generalized decomposition to evaluate multigroup neutron diffusion problems with space-energy separation. The theory section discusses two types of decomposition for the multigroup neutron flux: space-only and space-energy separated representation. The former is utilized for criticality problems in Section 3 and the latter is the subject of analysis in this work. The chapter also discussed how to decompose the multigroup linear operators for both types of flux representations which is generalized for heterogeneous media. Finally, the section introduced the multi-space-energy decomposition of the flux



(a) Full spectrum

(b) Fast region spectrum

(c) Full spectrum error

(d) Fast region error

Figure 4.6: Visualization of 145-group graphite block spectrum and PGD error. Error defined by Equation (4.14a) with FOM solution as ϕ_{ref}

which allows for different separated representation for different regions of the energy domain.

In order to test the PGD implementations, they are applied to five different examples: two 2-D two-group examples with mildly heterogeneous and reactor geometries, two based on a 3-D seven-group reactor geometry, and a 3-D 145-group graphite block. For performance comparison, the examples also employ a full-order model equivalent, evaluated using MOOSE and `deal.II`. The results from the two-group problems show that PGD performance surpasses the full-order evaluation when the mesh is highly refined. The space-only representation is much better for this problem than the space-energy representation. This is due to the fact that the addition of the energy dimen-

sion does not reduce the dimensionality of the problem significantly, since it is only two-group, and this representation requires significantly more enrichments to evaluate to fully represent the multigroup solution. The third and fourth examples are differentiated by heterogeneous and homogeneous fuel regions. In the heterogeneous version, the PGD approaches have nearly identical performance and significantly slower than MOOSE. In the homogeneous version, PGD performance is significantly improved and the space-energy approach proves most effective. These results indicate that PGD has the best performance when the geometry is more homogeneous, which requires less enrichments to evaluate in the solution process. The space-energy decomposition also shows the best results in more homogeneous geometries and improves on the performance when more energy groups are discretized. The final example investigates the multi-space-energy representation. The results of this 145-group problem showed that the space-energy decomposition was a far more efficient technique than the full-order evaluation, even with iteration acceleration and parallel processing. The results also showed that the multi-space-energy representation did not significantly affect the global error or the run-time. However, the MSE technique did affect the spatial error in different energy regions. In general, PGD had difficulty resolving the flux at the resonance energies and the errors in the fast region were unaffected by the MSE technique. However, separation of fast and thermal energies reduced the error in the thermal region and required significantly less enrichments.

The results presented in this chapter show that PGD is a highly efficient method and the space-energy separation is especially effective for fine-group calculations. For the fine-group results, the global error for PGD was impressive, but some of the multigroup flux errors were large. If these fluxes need to be more accurate, significantly more enrichments are required, which may deem PGD ineffective. The multi-space-energy representation helps the error in the thermal regions, but does not affect the fast region. Since the full-order evaluation is very efficient in the fast region, a possible alternative would be to use the full-order model in this region and PGD with space-energy decomposition in the upscattering region.

5. INVESTIGATING THE APPLICATION OF PROPER GENERALIZED DECOMPOSITION TO NEUTRON TRANSPORT

5.1 Introduction

Modeling and simulation of neutron transport problems is often a daunting task due to the high dimensionality of its phase-space. Steady-state, mono-energetic neutron transport has a five-dimensional phase-space: three for spatial description and two for particle direction. Due to its large phase-space, neutron transport suffers heavily from the “curse of dimensionality". This chapter investigates the capability of PGD to reduce the dimensionality of this phase space. Two different PGD approaches are investigated: space-only decomposition of each directional angular flux and full space-angle decomposition. For reference, the model equation is shown in Equation (5.1).

$$\vec{\Omega} \cdot \nabla \psi(\vec{r}, \vec{\Omega}) + \Sigma_t(\vec{r}) \psi(\vec{r}, \vec{\Omega}) = \frac{1}{4\pi} \Sigma_s(\vec{r}) \phi(\vec{r}) + \frac{1}{4\pi} \mathcal{Q}(\vec{r}) \, . \tag{5.1}$$

This chapter is broken up into two sections. Section 5.2 utilizes PGD for the spatial decomposition of the S_N neutron transport equations. This section includes different PGD operator formulations and analysis of various angular quadratures and scattering ratios. Section 5.3 explores the prospect of space-angle decomposition. Dominesey et al. in [53] provide an analysis for 1-D transport problems; therefore, this work focuses on extending the implementation to multi-dimensional transport problems. PGD is not directly applied to these problems; however, a formulation of the PGD operators is presented and a SVD analysis on 2-D problems is included. In this work, the investigation is limited to pure absorption and isotropic scattering problems with unaccelerated source iteration.

5.2 Space-Only Decomposition of Neutron Transport

The model problem for this work is the steady-state, mono-energetic neutron transport equation, shown in Equation (5.1). A common discretization technique for the angular dimensions

utilizes discrete ordinates or S_N, where the angular flux (ψ) is evaluated at discrete directions [91]. The resulting approximation to the angular flux is then:

$$\psi(\vec{r},\vec{\Omega}) = \sum_{n=1}^{N} \psi^n(\vec{r})\delta(\vec{\Omega}-\vec{\Omega}_n) \ , \tag{5.2a}$$

$$\phi(\vec{r}) = \sum_{n=1}^{N} w_n \psi^n(\vec{r}) \ . \tag{5.2b}$$

With this discretization, Equation (5.1) is then converted into an equation for each direction (N total):

$$\vec{\Omega}_n \cdot \nabla\psi^n(\vec{r}) + \Sigma_t(\vec{r})\psi^n(\vec{r}) = \frac{1}{4\pi}\Sigma_s(\vec{r})\phi(\vec{r}) + \frac{1}{4\pi}\mathcal{Q}(\vec{r}) \ , \quad n = 1,..,N \ . \tag{5.3}$$

With this model equation, PGD with space-only decomposition represents each S_N angular flux and scalar flux as spatially decomposed variables,

$$\psi^n(x,y) = \sum_{i=1}^{R_n} X_i^n(x) Y_i^n(y) \ , \tag{5.4a}$$

$$\phi(x,y) = \sum_{n=1}^{N} w_n \sum_{i=1}^{R_n} X_i^n(x) Y_i^n(y) \ . \tag{5.4b}$$

The rest of this section presents the PGD operator formulation necessary to apply the PGD solution process to Equation (5.4a), operator symmeterization typically used for unsymmetrical operators, source iteration approach, and several results.

5.2.1 Operator Construction

This section describes the “Galerkin" operator formulation specifically for neutron transport with heterogeneous properties, shown by Equation (5.3), where the cross section and source term are spatially dependent (e.g., piece-wise constant per zone). In order to represent the operator shown in Equation (2.2b), the domain describing each material property must be decomposed into products of one-dimension functions, similarly to the flux representation of Equation (5.4a). For a

space-dependent cross section, the separated representation is,

$$\Sigma(x,y) = \sum_{j=1}^{R_\sigma} \Sigma_j^x(x)\Sigma_j^y(y) \ . \tag{5.5}$$

The resulting description of the linear operator is shown in Equation (5.6). Consequently, $L = 2 + R_\sigma$ from Equation (2.2b), and more generally in $\texttt{dim}$ spatial dimensions, $L = \texttt{dim} + \texttt{R}_\sigma$.

$$\mathcal{L}_1^x X_i \mathcal{L}_1^y Y_i = \Omega_x \frac{\partial X_i}{\partial x} Y_i \ , \tag{5.6a}$$

$$\mathcal{L}_2^x X_i \mathcal{L}_2^y Y_i = \Omega_y X_i \frac{\partial Y_i}{\partial y} \ , \tag{5.6b}$$

$$\mathcal{L}_{\ell+2}^x(X_i)\mathcal{L}_{\ell+2}^y(Y_i) = \Sigma_{t,\ell}^x(x) X_i \Sigma_{t,\ell}^y(y) Y_i, \ \ \ell = 1, ..., R_\sigma \ . \tag{5.6c}$$

Acquiring the one-dimensional material property functions for the description of the material zones requires decomposition of the heterogeneous media. This technique is described in Section 2.3.1. With this description of the PGD-decomposed operator, the process from Section 2.1 can be directly applied here. However, the “Galerkin" formulation described from this section can cause convergence difficulties in the solution process because of the asymmetry of the differential operator. The following subsection describes a technique to symmeterize PGD operators.

5.2.2 Residual Minimization

The differential operators described by Equation (5.6) produce a non-symmetric system of equations. PGD applications to these types of systems show very poor convergence of PGD alternating direction iterations [16, 17]. Therefore, techniques to symmetrize the PGD operator are often employed to overcome this difficulty. Gonzalez et al. in [92] utilizes the streamline-upwind/Petrov-Galerkin (SUPG) method to perform the symmetrization on advection-diffusion problems. A more generic method is the residual minimization technique. The minimal residual technique is used exclusively in the LATIN multiscale method [20, 19]. However, for more general PGD applications, the method was proposed in [93] and further investigated in [34, 94, 95].

Residual minimization entails finding the minimum of the Euclidean norm of the residual. The

PGD residual for enrichment R is defined as:

$$\text{Res} = \sum_{\ell=1}^{L} \mathcal{L}_\ell^x X_R \mathcal{L}_\ell^y Y_R - \sum_{j=1}^{J} \mathcal{R}_j^x \mathcal{R}_j^y \,, \tag{5.7}$$

where $J = Q + L \times R$ and,

$$\mathcal{R}_j^x = \begin{cases} q_k^x \,, & j = k \\ -\mathcal{L}_\ell^x X_i \,, & j = k + \ell + L(i-1) \end{cases} \,, \tag{5.8a}$$

$$\mathcal{R}_j^y = \begin{cases} q_k^y \,, & j = k \\ \mathcal{L}_\ell^y Y_i \,, & j = k + \ell + L(i-1) \end{cases} \,. \tag{5.8b}$$

The euclidean norm ($||\cdot||$) of this residual is then defined as:

$$\begin{aligned} ||\text{Res}||^2 = \sum_{\ell=1}^{L}\sum_{\ell'=1}^{L} \left(\langle \mathcal{L}_\ell^x X_R, \mathcal{L}_{\ell'}^x X_R \rangle \langle \mathcal{L}_\ell^y Y_R, \mathcal{L}_{\ell'}^y Y_R \rangle\right) - \\ 2\sum_{\ell=1}^{L}\sum_{j=1}^{J} \left(\left\langle \mathcal{L}_\ell^x X_R, \mathcal{R}_j^x \right\rangle \left\langle \mathcal{L}_\ell^y Y_R, \mathcal{R}_j^y \right\rangle\right) + \sum_{j=1}^{J}\sum_{j'=1}^{J} \left(\left\langle \mathcal{R}_j^x, \mathcal{R}_{j'}^x \right\rangle \left\langle \mathcal{R}_j^y, \mathcal{R}_{j'}^y \right\rangle\right) \,, \end{aligned} \tag{5.9}$$

where $\langle \cdot, \cdot \rangle$ is defined as an integration over the one-dimensional phase space,

$$\langle a(x), b(x) \rangle = \textstyle\int_{\Omega_x} a(x) b(x) dx \quad , \quad \langle c(y), d(y) \rangle = \textstyle\int_{\Omega_y} c(y) d(y) dy \,. \tag{5.10}$$

This residual is then minimized dimension by dimension, an alternating direction approach, by setting the derivative of the residual norm with respect to variable being solved equal to zero:

$$\tfrac{\partial ||\text{Res}||^2}{\partial X_R} = 0 \quad \text{and} \quad \tfrac{\partial ||\text{Res}||^2}{\partial Y_R} = 0 \,. \tag{5.11}$$

With this formulation, a system of equations can be written for X_R and Y_R. For instance, the

system for X_R is then:

$$\sum_{\ell=1}^{L}\sum_{\ell'=1}^{L}\hat{\beta}^{y}_{\ell,\ell'}\left\langle \mathcal{L}^{x}_{\ell},\mathcal{L}^{x}_{\ell'}X_R\right\rangle = \sum_{\ell=1}^{L}\sum_{j=1}^{J}\hat{\gamma}^{y}_{\ell,j}\left\langle \mathcal{L}^{x}_{\ell},\mathcal{R}^{x}_{j}\right\rangle \ , \tag{5.12}$$

where,

$$\hat{\beta}^{y}_{\ell,\ell'} = \langle \mathcal{L}^{y}_{\ell}Y_R, \mathcal{L}^{y}_{\ell'}Y_R\rangle \tag{5.13a}$$

$$\hat{\gamma}^{y}_{\ell,j} = \left\langle \mathcal{L}^{y}_{\ell}Y_R, \mathcal{R}^{y}_{j}\right\rangle \ . \tag{5.13b}$$

The equation for Y_R can be derived similarly.

Although this derivation is complete, it is quite tedious to implement. Another approach is to redefine the linear operator $\mathcal{L}$ and source q in its PGD residual minimization form. This is done by multiplying each side of Equation (2.1) by the adjoint of $\mathcal{L}$. The new PGD operators, $\hat{\mathcal{L}}^{x}_{\hat{\ell}}$ and $\hat{\mathcal{L}}^{y}_{\hat{\ell}}$, and source, $\hat{q}^{x}_{\hat{k}}$ and $\hat{q}^{y}_{\hat{k}}$, are then defined as:

$$\hat{\mathcal{L}}^{x}_{\hat{\ell}}X_i\,\hat{\mathcal{L}}^{y}_{\hat{\ell}}Y_i = \mathcal{L}^{x,\dagger}_{\ell}\mathcal{L}^{x}_{\ell'}X_i\,\mathcal{L}^{y,\dagger}_{\ell}\mathcal{L}^{y}_{\ell'}Y_i \ ,\ \hat{\ell} = \ell + L(\ell'-1) \ , \tag{5.14a}$$

$$\hat{q}^{x}_{\hat{k}}\,\hat{q}^{y}_{\hat{k}} = \mathcal{L}^{x,\dagger}_{\ell}q^{x}_{k}\,\mathcal{L}^{y,\dagger}_{\ell}q^{y}_{k} \ ,\ \hat{k} = \ell + L(k-1) \ , \tag{5.14b}$$

where $\dagger$ signifies the adjoint of the operator. In discrete form, the adjoint is simply the transpose of the operator matrix. With this definition of the linear operators and source, the process from Section 2.1 can be applied directly. However, the number of terms in the decomposed operators becomes L^2, which could become quite taxing for the PGD process.

This work utilizes both the “Galerkin" and residual minimization formulations for the S_N transport operator and compares their convergence. For more information regarding the residual minimization approach, see Chapter 6 of [16] or Section 3.4 of [17].

5.2.3 Source Iteration

Although the neutron transport equation represented by Equation (5.3) is linear, an iteration scheme is required to evaluate the angular fluxes effectively, due to the integral scattering operator. Source iteration is the most traditional and historically robust iteration technique. The source iteration scheme is represented by,

$$\vec{\Omega}^n \cdot \nabla \psi^{n,k+1}(\vec{r}) + \Sigma_t(\vec{r})\psi^{n,k+1}(\vec{r}) = \frac{1}{4\pi}\mathcal{S}^k(\vec{r}) \ , \tag{5.15}$$

where k is the iteration index and,

$$\mathcal{S}^k(\vec{r}) = \Sigma_s(\vec{r})\phi^k(\vec{r}) + \mathcal{Q}(\vec{r}) \ . \tag{5.16}$$

The convergence of this iteration technique is dependent on the scattering ratio ($c = \Sigma_s/\Sigma_t$), which is often very close to unity. There are many types of acceleration methods, including diffusion synthetic acceleration (DSA) [96], nonlinear diffusion acceleration (NDA) [97], and nonlinear Krylov acceleration (NKA) [98], see [99] for an overview of common acceleration schemes. However, this work will only address the performance of PGD with straight source iteration without acceleration.

Due to the versatility of the PGD methodology, there are numerous theoretical techniques to perform this type of iteration within the PGD solution process. This section will discuss two such techniques, which are described in the following subsections.

5.2.3.1 PGD Source Iteration: Type 1

The first type of source iteration with PGD is essentially using the full PGD solution process as a linear system evaluation method. This is done by evaluating the system defined by Equation (5.6) with a source term defined by the previous iteration's scalar flux. The resulting angular flux from this evaluation is then,

$$\psi^{n,k+1}(x,y) = \sum_{i=1}^{R_n^{k+1}} X_i^{n,k+1}(x) Y_i^{n,k+1}(y) \ , \tag{5.17}$$

which is computed from the source term,

$$\mathcal{S}^k(x,y) = \sum_{n=1}^{N} w_n \sum_{\ell=1}^{R_\sigma} \sum_{i=1}^{R_n^k} \Sigma_{s,\ell}^x(x) X_i^{n,k}(x) \Sigma_{s,\ell}^y(y) Y_i^{n,k}(y) + \sum_{\ell=1}^{R_q} q_\ell^x(x) q_\ell^y(y) \ . \tag{5.18}$$

5.2.3.2 PGD Source Iteration: Type 2

This second type of source iteration with PGD attempts to optimize the linear system evaluation discussed in the previous section. The idea is to utilize the information gained from the previous iteration's solution to ease the evaluation of future iterations. This is done by defining the next iteration's angular flux as:

$$\psi^{n,k+1} = \psi^{n,k} + \delta\psi^{n,k+1} = \sum_{k'=1}^{k} \sum_{i=1}^{R_n^{k'}} X_i^{n,k'} Y_i^{n,k'} + \sum_{i=1}^{R_n^{k+1}} X_i^{n,k+1} Y_i^{n,k+1} \ . \tag{5.19}$$

In this scheme, the PGD process is only solving for $\delta\psi^{n,k+1}$ which presumably requires significantly less separated terms than $\psi^{n,k+1}$, and therefore requires less enrichments to solve for. This scheme starts every PGD evaluation in the source iteration process by assuming all the previous iterations are part of solution, then enriching from there.

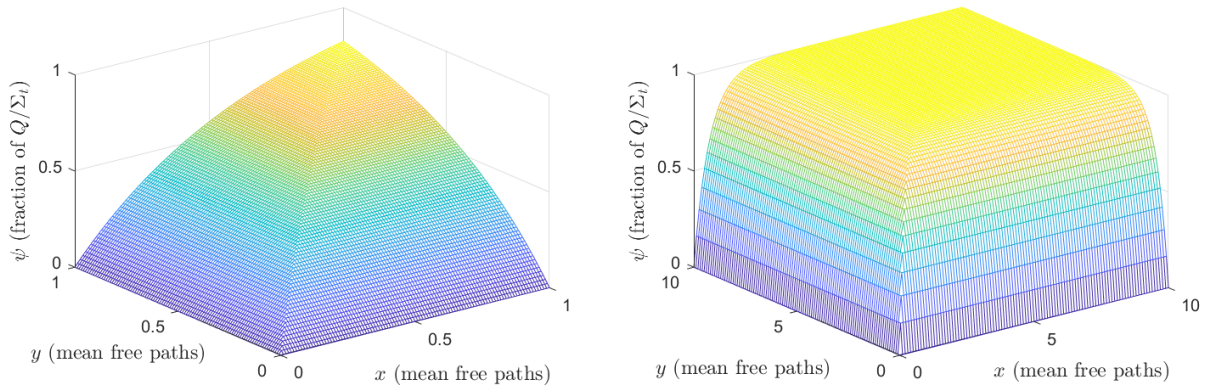
5.2.4 Results

To analyze the performance of PGD, it is applied to a simple 2-D example, a homogeneous geometry with a volumetric source. Two different instances of this problem are analyzed: a purely absorbing media where the solution in a single direction is evaluated and a scattering material where the two different source iteration techniques are employed. The performance metric for PGD is based on the number of enrichments that are required to evaluate. Although PGD does not have a rigorous definition of the rate of enrichment convergence, singular value decomposition (SVD) is used to illustrate the optimal convergence. Diamond difference discretization is used for each 1-D variable in the PGD solution. SVD is performed on the same mesh but with a full-order model (FOM) solution.

5.2.4.1 *Pure Absorber*

In this example, PGD and SVD are used to decompose 2-D angular flux in a homogeneous, purely absorbing geometry with volumetric source. Two different instances of the domain are chosen: a domain consisting of 1 mean free path between each edge and a domain with 10 mean free paths. These problems employ the “Galerkin" and residual minimization operator construction with diamond difference discretization. Application of PGD to this problem is meant for analysis of the different operator formulations and enrichment convergence.

A single simulation was run for each operator construction on a uniform mesh with 1000 uniform nodes in each dimension. Since this problem is purely advective, only a single angle was considered with $\mathbf{\Omega} = \left[\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right]$. An analytic solution is also available utilizing ray tracing. To portray a optimized decomposition, this solution was projected on the same mesh and decomposed using SVD. The resulting angular fluxes are shown by Figure 5.1.



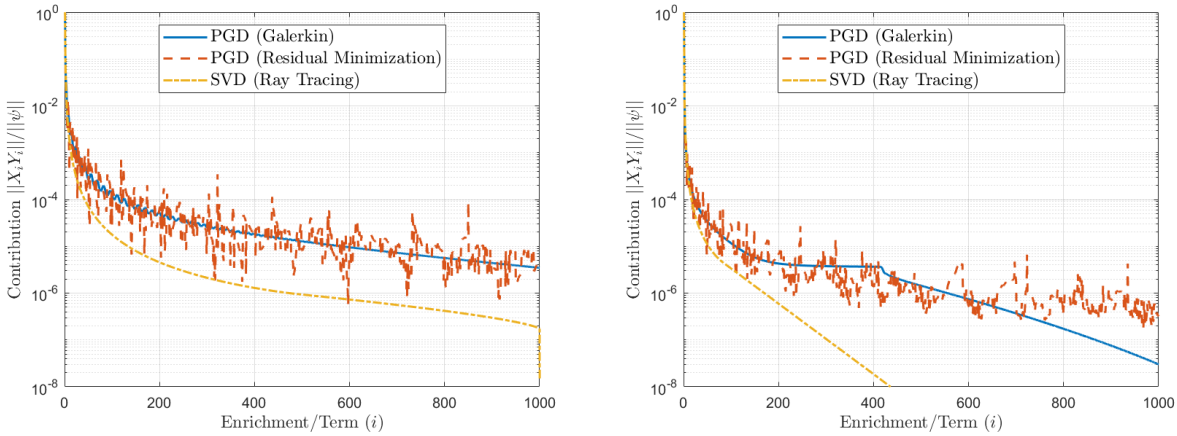
(a) Domain with 1 mean free paths

(b) Domain with 10 mean free paths

Figure 5.1: Visualization of angular flux for pure absorber problem

Figure 5.2 shows the normalized contribution of each enrichment term for each of the methods in both geometries. Figure 5.2a shows that PGD has very poor convergence for this problem. After 1000 enrichments, the PGD solution is no longer a reduced order model, as the size of the solution

is larger than the full order solution. This poor convergence is caused by the fact that the full order solution is non-separable, which is indicated by the poor convergence of SVD on the analytical solution. Figure 5.2b shows an improved convergence for both PGD and SVD. It is also important to note that the residual minimization approach allows for better convergence of the alternating direction iteration, but more sporadic convergence of the enrichment procedure.



(a) Domain with 1 mean free path

(b) Domain with 10 mean free paths

Figure 5.2: Enrichment convergence for PGD and SVD techniques

5.2.4.2 Isotropic Scattering

In this example, PGD is utilized within the source iteration process to evaluate angular flux in a isotropic scattering material. The domain with 10 mean free paths was chosen for this analysis. The purpose is to see how both source iteration techniques perform and analyze the separability of the solution with various scattering ratios and angular quadratures. Nine different scenarios are presented including S_2, S_4, and S_8 angular discretizations and scattering ratios of 0.1, 0.5, and 1.0. The PGD functions use a 1-D diamond difference discretization with 100 elements in each dimension. The FOM uses 2-D diamond difference on an equivalent mesh.

Figure 5.3 shows the number of enrichments required to solve for an enrichment error of 10^{-6}

at each source iteration. Enrichment error is defined as:

$$\epsilon_{\text{enr}} = \frac{||X_R||\,||Y_R||}{\sum_{i=1}^{R} ||X_i||\,||Y_i||} \,. \tag{5.20}$$

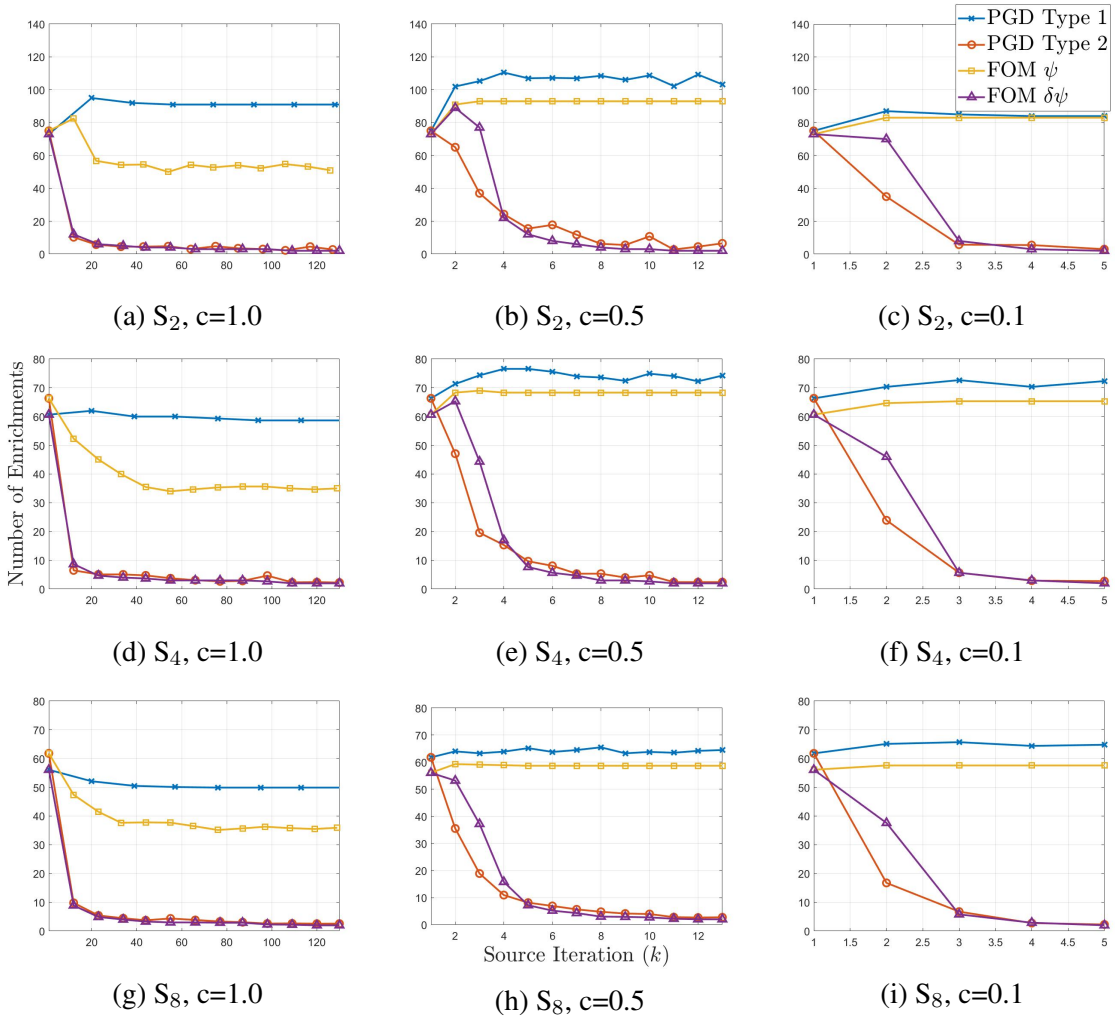
The FOM ψ and $\delta\psi$ results are meant to show the SVD equivalent to the type 1 and 2 source iteration techniques, respectively. Figure 5.4 shows the L_2 error of the scalar flux at each source iteration (k), where the reference is the fully converged flux at iteration K. From these plots, one can conclude that the type 2 source iteration is a much more efficient process than type 1. The separability of the flux does not seem to be dependent on the amount of scattering, the number of discrete ordinates, or the number of source iterations. However, $\delta\psi$ is highly dependent on all these quantities. In general, PGD type 2 performs best with a more refined angular quadrature and larger scattering ratio.

5.2.5 Discussion

This section introduces the prospect of utilizing PGD for two-dimensional S_N transport. It presents an overview of the PGD solution process in its “Galerkin" formulation and introduce residual minimization to symmeterize the PGD operator, which is common in advection-diffusion PGD applications. The main difference in this transport application is that the problems are purely advective. Two techniques are also introduced to incorporate the PGD process within source iteration procedure in order to resolve the angular coupling in the scattering term.

To investigate the implementation, PGD is applied to a 2-D homogeneous, isotropic scattering domain with a fixed volumetric source. Two types of problems are applied: pure absorption and isotropic scattering. Analysis of the pure absorbing application shows that PGD convergence is very poor for both “Galerkin" and residual minimization formulations in a small domain. Utilizing SVD on the analytical solution indicates that the cause of the poor convergence is the non-separability of the solution. However, convergence of enrichment terms significantly improves with a larger domain. The scattering application investigates nine different cases with various angular quadratures and scattering ratios. The results indicate that the amount of scattering does not

affect the separability of the angular flux, but the separability of $\delta\psi$, computed from source iteration, is vastly improved through the source iteration procedure. Therefore, evaluating $\delta\psi$ instead of ψ using PGD could prove effective.

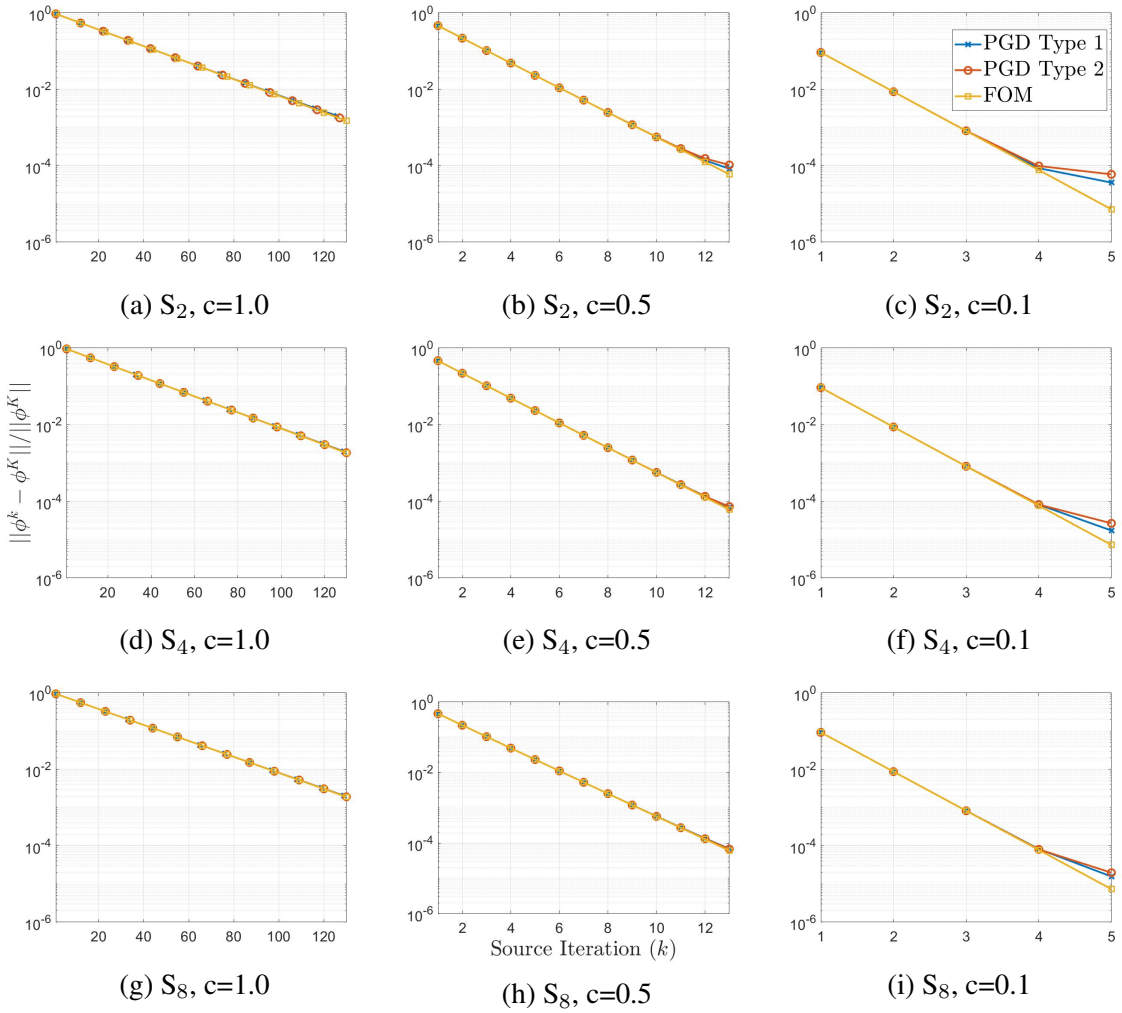


(a) S_2, c=1.0 (b) S_2, c=0.5 (c) S_2, c=0.1

(d) S_4, c=1.0 (e) S_4, c=0.5 (f) S_4, c=0.1

(g) S_8, c=1.0 (h) S_8, c=0.5 (i) S_8, c=0.1

Figure 5.3: Number of enrichments required for an enrichment tolerance of $\mathbf{10^{-6}}$ at each source iteration, averaged over all angular flux solutions.



(a) S_2, c=1.0 (b) S_2, c=0.5 (c) S_2, c=0.1

(d) S_4, c=1.0 (e) S_4, c=0.5 (f) S_4, c=0.1

(g) S_8, c=1.0 (h) S_8, c=0.5 (i) S_8, c=0.1

Figure 5.4: Scalar flux error at each source iteration. Error is computed relative to the fully converged FOM solution.

5.3 Space-Angle Decomposition for Neutron Transport

In this section, the solution $\psi(\vec{r}, \vec{\Omega})$ is decomposed in both space and angle. The differentiation between space-only and space-angle decomposition is similar to that of the space-energy decomposition in Section 4. Now, there are several possible representations that serve this purpose. The least intrusive technique is to only separate space and angle as,

$$\psi(\vec{r}, \vec{\Omega}) \approx \sum_{i=1}^{R} X_i(\vec{r}) \mathcal{O}_i(\vec{\Omega}) \ . \tag{5.21a}$$

The next technique is to perform a full spatial decomposition,

$$\psi(\vec{r},\vec{\Omega}) \approx \sum_{i=1}^{R} X_i(x)Y_i(x)Z_i(z)\mathcal{O}_i(\vec{\Omega}) \; . \tag{5.21b}$$

However, based on the results of the previous section, this decomposition would most likely prove ineffective. The angular dimensions are typically represented as $\vec{\Omega} = [\mu, \varphi]$, where μ is the cosine of the polar angle and φ is the azimuthal angle. If the discrete ordinate grid is uniform in these angles, then the PGD representation of these angular dimensions can be split into 1-D functions,

$$\psi(\vec{r},\vec{\Omega}) \approx \sum_{i=1}^{R} X_i(\vec{r})\mathcal{M}_i(\mu)\Phi_i(\varphi) \; . \tag{5.21c}$$

From these descriptions of the angular flux, any type of discretization technique for the angular dimensions is feasible. For this work, S_N is utilized where the μ dimension uses Gauss-Legendre quadrature and φ uses Chebyshev quadrature [91].

These three representations include the angular variables in the separated representation, resulting in a full space-angle decomposition. However, in order to account for boundary conditions in a Cartesian mesh, there needs to be a different PGD solution for each octant of the angular domain. The solution for the angular flux in the octant where $\Omega_x > 0$, $\Omega_y > 0$, and $\Omega_z > 0$ is then

$$\psi(\vec{r},\vec{\Omega} > [0,0,0]) \approx \sum_{i=1}^{R^{+++}} X_i^{+++}(\vec{r})\mathcal{O}_i^{+++}(\mathbf{\Omega}^{+++}) \; , \tag{5.22a}$$

for the Equation (5.21a) representation and

$$\psi(x,y,z,\vec{\Omega} > [0,0,0]) \approx \sum_{i=1}^{R^{+++}} X_i^{+++}(x)Y_i^{+++}(y)Z_i^{+++}(z)\mathcal{O}_i^{+++}(\mathbf{\Omega}^{+++}) \; , \tag{5.22b}$$

for the Equation (5.21b) representation. where $\mathbf{\Omega}^{+++} \subset [\vec{\Omega}_1, ..., \vec{\Omega}_N]$. The $(+++)$ superscripts indicate that the solution lives in the all positive octant; the solution where it is $(++-)$ lives

where $\Omega_z < 0$. For the Equation (5.21c) representation,

$$\psi(\vec{r}, \mu > 0, 0 < \varphi < \pi/2) \approx \sum_{i=1}^{R^{+++}} X_i^{+++}(\vec{r}) \mathcal{M}_i^{+++}(\boldsymbol{\mu}^{+}) \Phi_i^{+++}(\boldsymbol{\varphi}^{++}) \ , \tag{5.22c}$$

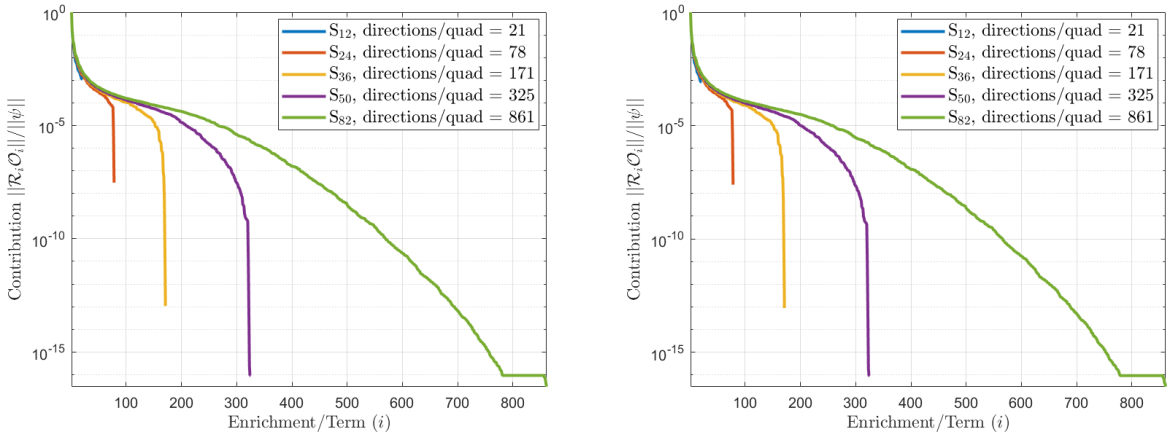
where $\boldsymbol{\mu}^{+} \subset [\mu_1, .., \mu_N]$ and $\boldsymbol{\varphi}^{++} \subset [\varphi_1, ..., \varphi_N]$.

This section is meant to introduce the possibility of performing a space-angle decomposition of a multi-dimension angular flux. Making an informed decision of the best type of decomposition would require extremely extensive analysis and any type of conclusion on this question would be premature at this stage of PGD application. Consequently, this chapter focuses on investigating the possibility of using a space-angle decomposition to serve as a reduced order model. Therefore, the following section utilizes the least intrusive decomposition of Equation (5.21a) and investigate the separability of simple angular flux solution in this form.

5.3.1 Results

To investigate the separability of a angular flux solution, these results utilize the same geometry from Section 5.2.4 with a domain of 10 mean free paths. Two different instances are considered: pure absorption and pure isotropic scattering. For this analysis, a full-order model is executed with varying angular quadratures and a diamond-difference discretization. These solutions are then decomposed using SVD to determine the optimal decomposition for the representation by Equation (5.21a). Figures 5.5a and 5.5b shows the convergence of the SVD execution for the pure absorption and pure scattering cases, respectively. These results show that optimal separability of the space-angle decomposition, thus PGD will undeniably perform worse. The convergence of each angular quadrature shows a steep dive at the enrichment equal to the number of directions per quadrature, which is where the separated representation is no longer a reduced order model. Only the S_{50} and S_{82} have an enrichment error below 10^{-6} before this threshold. However, at least 250 enrichments are needed for this error. In the PGD process, approximately 50 alternating directions are needed for this difficult, asymmetric problem for each enrichment, meaning a total of 12,500 2-D sweeps are required, conservatively. Therefore, performing the full-order model evaluation

would be significantly faster than a PGD implementation.



(a) Pure absorption case

(b) Pure scattering case

Figure 5.5: Enrichment convergence for space-angle SVD with various angular quadratures

6. PARAMETRIC UNCERTAINTY QUANTIFICATION USING PROPER GENERALIZED DECOMPOSITION APPLIED TO NEUTRON DIFFUSION [1]

6.1 Introduction

Mathematical models used to represent physical systems often have many parameters whose values are not known exactly, but with some level of uncertainty. When the number of parameters is large, the propagation of uncertainty on the system's output becomes computationally challenging; this is known as the curse of dimensionality in parametric uncertainty quantification. Further, when the model itself is expensive to evaluate, the problem as posed may become intractable. Surrogate models, or ROMs, are often sought to reduce the computational cost of the full-order physical models, while retaining its parametrization, i.e., its dependency on the input parameters. Typically, the application of ROMs aims at reducing the solution space *a posteriori*, that is, after a full-order, or high-dimensional, model has been exercised for a range of parameters to determine the proper subspace where the physics solution evolves. Section 1.2.2 gives a simple application of POD.

This chapter presents a different perspective, whereby parameterizing a model can be viewed as adding extra dimensions, in addition to the standard spatial and temporal dimensions typically present in governing laws. Thus, the dimensionality of a model includes the natural space and temporal coordinates, but also extra dimensions for each parameter under investigation. The solution from such a model lives, by definition, in a high dimensional space. With PGD, a high-dimensional problem is solved through a series of one-dimensional problems. When the extra dimensions include model parameters, PGD can be viewed as an *a priori* ROM method. This chapter applies PGD to uncertainty quantification for a diffusion-reaction (neutron diffusion) problem often employed in nuclear reactor physics. One of the features of nuclear reactor core modeling is that the physical domain is highly heterogeneous, with often a large number of material zones. Each material zone can have different properties, each of them only known within a certain accuracy due

[1] Reprinted with permission from "Parametric uncertainty quantification using proper generalized decomposition applied to neutron diffusion" by Z. M. Prince and J. C. Ragusa, 2019. *International Journal for Numerical Methods in Engineering*, Copyright 2019 by John Wiley and Sons.

to uncertainty in nuclear models and experimental data. Furthermore, strong neutron absorbers may be inserted in localized zones of the nuclear core, resulting in strong solution gradients. The main goal of this chapter is to assess the ability of PGD to handle large heterogeneous physical domains and to carry out uncertainty propagation with PGD for a certain number of input/model parameters.

So far, the PGD methodology for decomposing natural coordinates is presented in detail, with a separated representation similar to Equation (6.1a). A parametric model would add more dimensions to the decomposition. For example, parameterizing a property k present in the model results in a four-dimensional solution variable $u(x, y, z, k)$. The PGD separated representation of that variable is sum of products of functions of x, y, z, and k, shown in Equation (6.1b). In that sense, PGD can be seen as an *a priori* ROM technique, whereby the solution u is explicitly parameterized in the uncertain model parameters. Once the PGD representation is obtained, evaluating u at any point x, y, z in space and for any value of k is a straightforward application of Equation (6.1b).

$$u(x, y, z) = \sum_{i=1}^{\infty} X_i(x) Y_i(y) Z_i(z) \approx \sum_{i=1}^{N} X_i(x) Y_i(y) Z_i(z) \,, \tag{6.1a}$$

$$u(x, y, z, k) = \sum_{i=1}^{\infty} X_i(x) Y_i(y) Z_i(z) \mathcal{K}_i(k) \approx \sum_{i=1}^{N} X_i(x) Y_i(y) Z_i(z) \mathcal{K}_i(k) \,. \tag{6.1b}$$

The main advantage to the representation given by Equation (6.1b) is that the resulting PGD solution will have an unambiguous dependence on the parameterized properties, resulting in a trivial post-processing step for Uncertainty Quantification (UQ) purposes.

The model problem for this work is represented by the one-group, steady-state, neutron diffusion equation with an external source, given by Equation (2.24). Parameterizing this model involves including D, Σ_a, and Q as extra dimensions in the PGD solution process. In a homogeneous domain, this parameterization yields the following solution decomposition:

$$\phi(\vec{r}, D, \Sigma_a, Q) \approx \sum_{i=1}^{N} X_i(x) Y_i(y) Z_i(z) \mathcal{D}_i(D) \mathcal{S}_i(\Sigma_a) \mathcal{Q}_i(Q) \,. \tag{6.2}$$

In reactor physics, the material properties (diffusion coefficient, absorption cross section) and sources are typically piece-wise constant per material zone (for instance, when fuel assemblies are loaded with different fissile contents, or when neutron absorbers are locally inserted). In such situations, the PGD expansion of Equation (6.2) needs to be amended to reflect the heterogeneous character of the material layout in the domain; this is further discussed in Section 6.2. For the reader interested in the generation of material properties for neutron diffusion, please see Chapter 5 of the textbook by Duderstadt and Hamilton [59]. For the purpose of this dissertation, a 3-D heterogeneous nuclear core layout and the associated (possibly uncertain) material properties are assumed given.

Having a parameterized solution, as shown in Equation (6.2), is advantageous for uncertainty quantification and design of nuclear reactors. Several important quantities of interest (QOI) can be readily obtained as functional of the solution and model parameters can be optimized for a desired QOI. Some examples of QOIs in reactor physics include (1) the average flux solution over a region of interest, (2) the maximum value of the flux in the domain (the "peaking factor"), and (3) the total neutron population in the core (proportional to the volume integral of the flux). These QOIs are obviously dependent upon model parameters. With a PGD expansion that accounts for the parameters, uncertainty quantification for these QOIs is straightforward.

Parametrizing models has become a hallmark of the PGD method, with applications to a myriad of problems, see 1.3.1 for a detailed overview. Most of these applications involve parameterizing only one or two properties, where the material properties are homogeneous. One notable exception can be found in Lamari et al. [35], where a parameterization of the heat conduction coefficient in every region of a heterogeneous domain is performed, with homogenization as the ultimate goal of their study. The work presented here is similar to this parameterization of material properties. The key differences in this work are the subject of application and the capability of parameterizing select and discontinuous material zones. Furthermore, none of the neutronic applications from Section 1.3.2 involve the investigation of a parametric model with PGD. This chapter includes three key extensions of the PGD method. First, this chapter presents an application of PGD to a

high-dimensional uncertainty quantification problem. Second, this work applies parametric PGD to a model problem not considered previously. Finally, it presents an algorithm for decomposing parametric material properties that allows for the construction of 1-D linear systems before the PGD solution process is initiated; this is especially optimized for highly heterogeneous two- and three-dimensional geometries.

The outline of the remainder of the chapter is as follows. In Section 6.2, the PGD process is expanded upon to include extra dimensions to enable parametric studies for neutron diffusion problems. Section 6.3 describes the uncertain quantification procedures performed in the chapter, including descriptions of the quantities of interest and the integration method for computing mean and variance. Section 6.4 describes the decomposition of material properties for parametric models, so that the 1-D matrices of the PGD-decomposed multi-D operator can be computed once and for all. Results are presented in Section 6.5; a one-dimensional parametric neutron diffusion problem (for which an analytical solution can be obtained) is presented first, followed by a more complex and realistic parametric model for a full 3-D nuclear reactor core. Conclusions and outlook are proposed in Section 6.6.

6.2 PGD for Parametric Models

The equation of interest in this work is the parameterized one-group neutron diffusion equation, shown in Equation (2.24), where its solution ϕ is the neutron flux, D is the diffusion coefficient, Σ_a is the absorption cross section, and Q is a neutron source. In the case of a homogeneous domain, the parameterized solution, ϕ, can be expressed as $\phi(\vec{r}, D, \Sigma_a, Q)$. The expression for the resulting decomposed solution in a three-dimensional domain is given in Equation (6.2). However, for the vast majority of applications, and the problems at hand here, the physical domain is not homogeneous and material variations should be accounted for. Here, it is assumed that the material properties are piece-wise constant in each material zone. In this case, the parameterized solution is expressed as $\phi(\vec{r}, D_1, ..., D_{M_d}, \Sigma_{a,1}, ..., \Sigma_{a,M_\sigma}, Q_1, ..., Q_{M_q})$. The subscripts of the material properties are the indices for the material zones, with $M_d,\ M_\sigma\ , M_q$ materials for $D\ , \Sigma_a$, and Q. The

resulting decomposed flux is represented by Equation (6.3) :

$$\phi(\vec{r}, D_1, ..., D_{M_d}, \Sigma_{a,1}, ..., \Sigma_{a,M_\sigma}, Q_1, ..., Q_{M_q}) = \sum_{i=1}^{N} \left(X_i(x) Y_i(y) Z_i(z) \prod_{j=1}^{M_d} \mathcal{D}_i^{D_j}(D_j) \prod_{j=1}^{M_\sigma} \mathcal{S}_i^{\sigma_j}(\Sigma_{a,j}) \prod_{j=1}^{M_q} \mathcal{Q}_i^{q_j}(Q_j) \right) . \quad (6.3)$$

The dimensionality of the parametric problem is now $3+M_d+M_\sigma+M_q$ (3 for $x, y,$ and z and $M_d+M_\sigma+M_q$ for the model parameters). The PGD process, described in Section 2.1, is applied here, but on a higher dimensional space. The weak forms for each decomposed spatial term (X_i, Y_i, Z_i) yield three 1-D neutron diffusion equations, as before. The weak forms for the decomposed parameter terms ($\mathcal{D}_i$, $\mathcal{S}_i$, $\mathcal{Q}_i$) result in purely algebraic equations due to the lack of derivative operators. They are obtained by integrating the three spatial dimensions and the other parameter dimensions. The parameter basis functions are simply finite volume (order-0 discontinuous Galerkin method). For instance, a simple example for a one-dimensional homogeneous heat conduction equation is given in Chapter 5 of Chinesta et al. [16].

6.3 Uncertainty Quantification

Uncertainty quantification (UQ) is an integral part of computational science and engineering. This section describes several ways a PGD solution can apply to UQ. In general, UQ applies to a set of quantities of interest (QOIs). For example, a QOI, either evaluated at a point in space or over a region of interest, will depend on the input/model parameters of a system. This process is depicted below for a generic QOI (denoted by I), in the case of a homogeneous 1-D single-material neutron diffusion problem:

$$\phi(x, y, z, D, \Sigma_a, Q) = \sum_{i=1}^{N} X_i(x) Y_i(y) Z_i(z) \mathcal{D}_i(D) \mathcal{S}_i(\Sigma_a) \mathcal{Q}_i(Q) \longrightarrow I(\phi) \equiv I(\vec{\mathcal{D}}, \vec{\mathcal{S}}, \vec{\mathcal{Q}}) , \quad (6.4)$$

with

$$\vec{\mathcal{D}} = [\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_N] \ , \ \vec{\mathcal{S}} = [\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_N] \ , \ \vec{\mathcal{Q}} = [\mathcal{Q}_1, \mathcal{Q}_2, ..., \mathcal{Q}_N] \ .$$

Once the parametric PGD solution is obtained, sampling for any QOI only involves a trivial evaluation of each decomposed term in the PGD decomposition for I. The parametric description for the QOI, as represented by the expression for I, is often straightforward to derive, as discussed in [100]. In cases where it is not, an alternate approach based on regression techniques can be used; see [101], for instance. This latter approach is more general but has not been pursued in this work.

6.3.1 Mean and Variance

Statistical moments (such as mean and variance) of an uncertain QOI can be computed using the PGD solution as well, without the need for Monte Carlo sampling of the distribution. Denoting a given joint probability distribution for the uncertain parameters by $f(D, \Sigma_a, Q)$, the mean and variance of a QOI (μ_I and σ_I^2) can be computed as follows:

$$\mu_I = \mathbb{E}\left[I\right] \ , \tag{6.5a}$$

$$\sigma_I^2 = \mathbb{E}\left[(I - \mu_I)^2\right] = \mathbb{E}[I^2] - \mu_I^2 \, , \tag{6.5b}$$

where the expectation for a generic function is given by

$$\mathbb{E}\left[g(D, \Sigma_a, Q)\right] = \int_{-\infty}^{\infty} dD \int_{-\infty}^{\infty} d\Sigma_a \int_{-\infty}^{\infty} dQ \, g(D, \Sigma_a, Q) \, f(D, \Sigma_a, Q) \, . \tag{6.5c}$$

In the case of independent uncertain parameters,

$$f(D, \Sigma_a, Q) = f_D(D) f_{\Sigma_a}(\Sigma_a) f_Q(Q) \, , \tag{6.6}$$

and the expectation becomes

$$\mathbb{E}\left[g(D, \Sigma_a, Q)\right] = \int_{-\infty}^{\infty} dD f_D(D) \int_{-\infty}^{\infty} d\Sigma_a f_{\Sigma_a}(\Sigma_a) \int_{-\infty}^{\infty} dQ f_Q(Q) \, g(D, \Sigma_a, Q) \, . \tag{6.7}$$

Now, if the QOI is based on a PGD expansion of the solution as shown below

$$I(D, \Sigma_a, Q) = \sum_{i=1}^{N} \alpha_i \mathcal{D}_i(D) \mathcal{S}_i(\Sigma_a) \mathcal{Q}_i(Q) \, , \tag{6.8}$$

where α_i are scalar coefficients, then the mean and variance expressions can be obtained through one-dimensional integrations:

$$\mu_I = \sum_{i=1}^{N} \alpha_i \int_{-\infty}^{\infty} dD \mathcal{D}_i(D) f_D(D) \int_{-\infty}^{\infty} d\Sigma_a \mathcal{S}_i(\Sigma_a) f_{\Sigma_a}(\Sigma_a) \int_{-\infty}^{\infty} dQ(Q) \mathcal{Q}_i f_Q(Q) \, , \tag{6.9a}$$

$$\sigma_I^2 = \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j \int_{-\infty}^{\infty} dD \mathcal{D}_i(D) \mathcal{D}_j(D) f_D(D) \int_{-\infty}^{\infty} d\Sigma_a \mathcal{S}_i(\Sigma_a) \mathcal{S}_j(\Sigma_a) f_{\Sigma_a}(\Sigma_a) \times \int_{-\infty}^{\infty} dQ \mathcal{Q}_i(Q) \mathcal{Q}_j(Q) f_Q(Q) - \mu_I^2 \, . \tag{6.9b}$$

Equation (6.9) indicates that the mean and variance from a PGD-evaluated parametric quantity can be computed from straight integration of the separated solution. In this chapter, these integrations are performed on the same finite-volume grid used for the parameter discretization. The calculation of the statistical moments using the separated representation of the parameter distribution and QOI is discussed more generally in [100].

Alternatively, a standard Monte Carlo sampling technique [102] can be used to obtain these statistical moments, by sampling from the joint probability distribution $f_D(D) f_{\Sigma_a}(\Sigma_a) f_Q(Q)$ and evaluating Equation (6.8) at these samples.

Finally, recall that the above expressions have been provided in the case of a homogeneous domain. Starting from the parameterized PGD solution for a heterogeneous domain (see Equation (6.3)), the same procedure can be repeated here. The numerical results for heterogeneous cases employ such expressions (they are omitted here for brevity but are straightforward to derive).

6.3.2 Quantities of Interest

Defining quantities of interest is pivotal for uncertainty quantification and design optimization. QOIs in reactor analysis typically include zone-averaged flux values, peaking factors, and total neutron population (other QOIs are used in the field but this exposition is limited to a few of these). The average flux in a region of interest (ROI) is given by:

$$\bar{\phi}_{\text{RoI}} = \frac{\int_{\text{RoI}} \phi(\vec{r}) d^3r}{\int_{\text{RoI}} d^3r} \, . \tag{6.10}$$

When the ROI is the entire domain (ROI$= \Omega$), this simply denoted as $\bar{\phi}$. The peaking factor is the ratio of the largest flux in the spatial domain to the average flux:

$$P_f = \frac{\max_{\vec{r}\in\Omega} \phi(\vec{r})}{\bar{\phi}} \, . \tag{6.11}$$

Also of interest is the change in the total neutron population, say, for example, when an absorber material (labelled "rod" here) is introduced in the core. A good measure for this quantity is the difference in the average fluxes:

$$\Delta\bar{\phi} = \bar{\phi}_0 - \bar{\phi}_{\text{rod}} \, , \tag{6.12}$$

where $\bar{\phi}_0$ denotes the average flux before the absorber rod insertion and $\bar{\phi}_{\text{rod}}$ denotes the average flux after insertion.

With these quantities of interest defined, one can now specialize Equation (6.8) and formulate their PGD expressions, using the separated representation of the solution shown Equation (6.2), as follows:

$$\bar{\phi}(D, \Sigma_a, Q) = \sum_{i=1}^{N} \alpha_i^x \alpha_i^y \alpha_i^z \mathcal{D}_i(D) \mathcal{S}_i(\Sigma_a) \mathcal{Q}_i(Q) \, , \tag{6.13a}$$

$$P_f(D, \Sigma_a, Q) = \frac{\max_{(x,y,z)\in\Omega} \left[\sum_{i=1}^{N} X_i(x) Y_i(y) Z_i(z) \mathcal{D}_i(D) \mathcal{S}_i(\Sigma_a) \mathcal{Q}_i(Q) \right]}{\sum_{i=1}^{N} \alpha_i^x \alpha_i^y \alpha_i^z \mathcal{D}_i(D) \mathcal{S}_i(\Sigma_a) \mathcal{Q}_i(Q)} \, , \tag{6.13b}$$

with

$$\alpha_i^x \alpha_i^y \alpha_i^z = \frac{\int_{\Omega_x} X_i(x)dx}{\int_{\Omega_x} dx} \times \frac{\int_{\Omega_y} Y_i(y)dy}{\int_{\Omega_y} dy} \times \frac{\int_{\Omega_z} Z_i(z)dz}{\int_{\Omega_z} dz} \ . \tag{6.13c}$$

For the change in total neutron population, the material zones received the absorber rod must be parameterized separately. Here, it is assume that only the absorption cross section value is affected by the presence of the inserted rod. Thus, the functional representation for functions $\mathcal{S}$ is as follows, $\mathcal{S}(\Sigma_a) \rightarrow \mathcal{S}^{\text{fuel}}(\Sigma_a^{\text{fuel}})\mathcal{S}^{\text{rod}}(\Sigma_a^{\text{fuel}})$ when the rod is not present and $\mathcal{S}(\Sigma_a) \rightarrow \mathcal{S}^{\text{fuel}}(\Sigma_a^{\text{fuel}})\mathcal{S}^{\text{rod}}(\Sigma_a^{\text{rod}})$ when it is inserted. This yields the following expression for the change in total neutron population:

$$\Delta\bar{\phi} = \sum_{i=1}^{N} \alpha_i^x \alpha_i^y \alpha_i^z \mathcal{D}_i(D) \mathcal{S}_i^{\text{fuel}}(\Sigma_a^{\text{fuel}}) \mathcal{Q}_i(Q) \left[\mathcal{S}_i^{\text{rod}}(\Sigma_a^{\text{fuel}}) - \mathcal{S}_i^{\text{rod}}(\Sigma_a^{\text{rod}}) \right] \ . \tag{6.14}$$

Statistical moments (e.g., mean and variance) of QOIs based on flux averages (i.e., for $\bar{\phi}$ and $\Delta\bar{\phi}$) can be computed directly with 1-D integrations in the parameter dimensions.

6.4 Material Property Decomposition

This section describes the construction of the PGD operator for neutron diffusion (diffusion+reaction) in the form shown in Equation (2.2). It is first worth noting that, in the parametric model, the material properties not only depend on space, but also on the magnitude of the property in each material zone. Therefore, a material property in a 3-D domain with M material zones will have $M + 3$ independent variables. In order to formulate the separated form of the operator, shown in Equation (2.2), these material properties must be decomposed similarly to the solution. This formulation allows for the 1-D matrices and source vectors to be constructed before the PGD process is started, instead of being re-computed on-the-fly during the PGD iterations. The number of terms used in the decomposition of a given material property (i.e., N_d,N_σ,N_q) depends on the number of material zones and the heterogeneity of the domain. A simple example is shown in Equation (6.15) for a generic property labeled k. Here, k corresponds to either D, Σ_a, or Q and M_k corresponds to the number of material zones for property k, (e.g., M_d , M_σ, or M_q). The goal is to find a material property decomposition $K(\vec{r}, k_1, ..., k_M)$ in a PGD manner; the argument k_j denotes the magni-

tude of property k in material zone j. The PGD decomposition reads as a product of a spatial description of the material zone and the magnitude of the material property itself.

$$K(\vec{r}, k_1, ..., k_{M_k}) = \sum_{j=1}^{M_k} K_j^{xyz}(x, y, z) k_j \ , \tag{6.15a}$$

where K_j^{xyz} describes the spatial position of material zone j. This characteristic function for material j is obtained via a PGD decomposition as follows :

$$K_j^{xyz}(x, y, z) = \sum_{n=1}^{N_j^{xyz}} K_{j,n}^x(x) K_{j,n}^y(y) K_{j,n}^z(z) = \begin{cases} 1 & \text{if} \quad (x, y, z) \in \text{ material zone } j \\ 0 & \text{otherwise} \end{cases} \ , \tag{6.15b}$$

where N_j^{xyz} is the number of terms needed to decompose it.

The methodology for finding $K_j^{xyz}(x, y, z)$ can be found in Section 2.3.1. This process is general for any given set of material properties. In the simple example of Figure 2.6, one can readily obtain the spatial representation of k_1, i.e., K_1^{xyz} from Equation (6.15b) by setting $k_1 = 1$ and $k_2 = 0$. An algorithm for decomposing a material property in a general domain is shown in Algorithm 2. This process consists of two parts. The first is to find the decomposed form of the property in the regions that are not parameterized (material property values are known). The second part finds the spatial description of each parameterized material zone (uncertain material property values).

6.4.1 Parameterized Insertion of an Absorber Rod

In order to assess the change in the total neutron population when an absorber is inserted locally by a certain amount (length) in the core, the section parameterizes that insertion length in a three-dimensional model. Parameterizing such control rod length is important in understanding the absorbing strength of the rod versus its location in the reactor. Parameterizing the insertion length is akin to a parametric geometrical condition. The decomposed form of the flux for this

Algorithm 2 Parameterized Material Property Decomposition

procedure $\boldsymbol{K}$ = PARAMETERIZEDDECOMPOSITION(G,P,$\vec{k}$)
$G \rightarrow$ Grid with M material zones
$P \subset [1, ..., M] \rightarrow$ Zones to be parameterized
$\vec{k} := [k_1, ..., k_M] \rightarrow$ Nominal values of material properties
$\boldsymbol{K} := \sum_{i=1}^{N_k} \left(K_i^x(x) K_i^y(y) K_i^z(y) \prod_{j \in P} K_i^{k_j}(k_j) \right) \rightarrow$ Output of decomposed parameter function
    $i = 0$
    Set $k_j = 0,\ j \in P$
    (2.27) $\longrightarrow \sum_{n=1}^{N_0^{xyz}} \tilde{K}_n^x(x) \tilde{K}_n^y(y) \tilde{K}_n^z(z)$ ▷ Decompose known (non-parametric) material properties.
    for all $n \leftarrow 1, N_0^{xyz}$ **do** ▷ Transfer decomposition to output
        $i = i + 1$
        $K_i^x(x) = \tilde{K}_n^x(x)$, $K_i^y(y) = \tilde{K}_n^y(y)$, $K_i^y(y) = \tilde{K}_n^y(y)$
        $K_i^{k_j}(k_j) = 1,\ j \in P$ ▷ All parametric dimensions are 1 for these terms
    end for
    for all $m \in P$ **do** ▷ Loop over parameterized zones
        Set $k_m = 1$
        Set $k_j = 0,\ j \neq m$
        (2.27) $\longrightarrow \sum_{n=1}^{N_m^{xyz}} \tilde{K}_n^x(x) \tilde{K}_n^y(y) \tilde{K}_n^z(z)$ ▷ Spatial description of material zone
        for all $n \leftarrow 1, N_m^{xyz}$ **do** ▷ Transfer decomposition to output
            $i = i + 1$
            $K_i^x(x) = \tilde{K}_n^x(x)$, $K_i^y(y) = \tilde{K}_n^y(y)$, $K_i^y(y) = \tilde{K}_n^y(y)$
            $K_i^{k_j}(k_j) = 1,\ j \in P,\ j \neq m$ ▷ Parametric dimensions not in zone m
            $K_i^{k_m}(k_m) = k_m$ ▷ Zone m dimension
        end for
    end for
    $N_k = i$
end procedure

parameterized problem is described then given by:

$$\phi(\vec{r}, \lambda_1, ..., \lambda_M) = \sum_{i=1}^{N} X_i(x) Y_i(y) Z_i(z) \prod_{j=1}^{M} \Lambda_i^j(\lambda_j) \ . \tag{6.16}$$

λ_j is the insertion length of control rod j, where there are M rods in total. This solution representation is analogous to the representation in Equation (6.3), except that it replaces the terms dependent on material property values with terms dependent on the insertion length. Again, this

solution decomposition requires that the material properties be similarly decomposed, as in Equation (6.15a). Since the insertion of a control rod is typically done along the z-dimension of the reactor, a material property can be decomposed as a sums of products of functions in x, y with functions in z:

$$K(\vec{r}, \lambda_1, ..., \lambda_M) = K_0(x, y, z) + \sum_{j=1}^{M} K_j(x, y)\chi_j(z, \lambda_j) \ . \tag{6.17}$$

$K_0(x, y, z)$ is the decomposition of the material properties with all rods fully extracted. K_j is the x-y description of the position of rod j, it does not have a z component. χ_j represents the change in the material properties due to the addition of the rod at location z, dependent on the insertion length of control rod j. These values can then be described explicitly as:

$$K_0(x, y, z) = \sum_{i=1}^{N_0} K_{0,i}^x(x) K_{0,i}^y(y) K_{0,i}^z(z) = k^{\text{no rod}}(x, y, z) \ , \tag{6.18a}$$

$$K_j(x, y) = \sum_{i=1}^{N_j^{xy}} K_{j,i}^x(x) K_{j,i}^y(y) = \begin{cases} 1 & \text{if} \quad (x, y) \in \text{ rod } j \text{ location} \\ 0 & \quad \text{otherwise} \end{cases} \ , \tag{6.18b}$$

$$\chi_j(z, \lambda_j) = \sum_{i=1}^{N_j^{z\lambda}} \chi_{j,i}^z(z) \chi_{j,i}^\lambda(\lambda_j) = \begin{cases} (k^{\text{rod}} - k^{\text{no rod}}) & \text{if } z \leq \lambda_j \\ 0 & \text{if } z > \lambda_j \end{cases} \ . \tag{6.18c}$$

The number of terms required to represent χ_j, i.e., $N_j^{z\lambda}$, is relatively large since it may be equal to the number of axial layers in the z-dimension for the discretization of λ_j. If the variable λ_j is discretized using finite volume with $N_j^{z\lambda}$ intervals ($\vec{\lambda}_j = [\tilde{\lambda}_{j,1}, ..., \tilde{\lambda}_{j,N_j^{z\lambda}}]$), the resulting decomposed form of χ_j is

$$\chi_{j,i}^z(z) = H(z - \tilde{\lambda}_{j,i-1/2}) - H(z - \tilde{\lambda}_{j,i+1/2}) \ , \quad i = 1, ..., N_j^{z\lambda} \ , \tag{6.19a}$$

$$\chi_{j,i}^\lambda(\lambda_j) = (k^{\text{rod}} - k^{\text{no rod}}) H(\lambda_j - \tilde{\lambda}_{j,i+1/2}) \ , \quad i = 1, ..., N_j^{z\lambda} \ , \tag{6.19b}$$

where $\tilde{\lambda}_{j,i+1/2}$ and $\tilde{\lambda}_{j,i-1/2}$ are the nodes of interval i in the "insertion length" parametric dimension.

6.5 Results

To illustrate the ability to perform parametric modeling and uncertainty quantification for neutron diffusion with PGD, this section presents four different problems. First, a 1-D homogeneous example, for which an analytical solution is available and used in order to verify the parametric solution. Second, a 1-D two-region problem, where the exact solution is also available, which is used to investigate how heterogeneity affects a PGD parametric solution. The third problem increases the problem complexity by employing a 2-D heterogeneous nuclear core geometry in order to analyze the ability to parameterize multiple material zones. Finally, a 3-D nuclear core geometry is used to showcase the PGD implementation for a problem that would be quite demanding for traditional uncertainty propagation methods. Each of these applications performs an error analysis of the PGD solution against either an analytical solution or a solution to the non-parametric model (the non-parametric situation is a much lower dimensional problem and can be solved effectively with traditional techniques and PGD equally). The application of the 3-D example goes further by parameterizing the insertion of control rods, which is only possible for 3-D geometries.

6.5.1 One-Dimensional Homogeneous Example

To begin the analysis of parameterizing properties in the neutron diffusion equation, PGD was applied to a parametric one-dimensional homogeneous problem. The neutron diffusion equation and its boundary conditions are given in Equation (6.20).

$$-D\frac{d^2\phi}{dx^2} + \Sigma_a\phi = Q\,, \quad x \in [0, 10]\,, \quad \phi(x=0) = \phi(x=10) = 0\,. \tag{6.20}$$

The analytical solution is provided in Equation (6.21), where the A and B constants depend on the boundary conditions and material properties.

$$\phi(x, D, \Sigma_a, Q) = A\text{exp}\left(\sqrt{\frac{\Sigma_a}{D}}x\right) + B\text{exp}\left(-\sqrt{\frac{\Sigma_a}{D}}x\right) + \frac{Q}{\Sigma_a}\,, \tag{6.21a}$$

with

$$A = -\frac{Q}{\Sigma_a}\left(\exp\left(10\sqrt{\frac{\Sigma_a}{D}}\right)+1\right)^{-1}, \tag{6.21b}$$

$$B = -\frac{Q}{\Sigma_a}\left(\exp\left(-10\sqrt{\frac{\Sigma_a}{D}}\right)+1\right)^{-1}. \tag{6.21c}$$

The parameters D, Σ_a, and Q are uncertain and can range between 0.1 and 10 (in units of cm, cm^{-1}, and n/cm^3·s, respectively). The probability distribution for each parameter can either be uniform or normal. The total number of dimensions in the parametric model is four (one for the physical space x, and three for the uncertain space in D, Σ_a, and Q). First-order continuous finite elements are used for the x dimension with 100 elements, while a finite volume discretization is used for the parameter dimensions, with 100 elements in each dimension, spanning 0.1 to 10. The enrichment tolerance is set to 10^{-6} and 119 enrichment terms are required to satisfy that tolerance value. The resulting decomposition to the neutron flux is shown by Equation (6.22). Figure 6.1 shows the solution evaluated for some material property values; the analytical solution is also plotted.

$$\phi = \sum_{i=1}^{N}\phi_i = \sum_{i=1}^{119} X_i(x)\mathcal{D}_i(D)\mathcal{S}_i(\Sigma_a)\mathcal{Q}_i(Q)\ . \tag{6.22}$$

Figure 6.2a shows the relative error of the PGD solution as a function of the full parameter space. This error plot illustrates where, in the parameter space, the PGD solution deviates from the exact solution. We note an independence of the error with respect to the source strength Q, as expected because the problem is linear in that parameter. The error is greater for large values of D and small values of Σ_a as the the solution has a steeper spatial variation for such values. Figure 6.2b shows the enrichment error (current enrichment level compared to current solution) and cumulative solution error at each enrichment step, respectively defined as:

$$\epsilon_e^{\text{enr}} = \frac{||\phi_e||}{||\sum_{i=1}^{e}\phi_i||}\ , \text{ and } \epsilon_e^{\text{sol}} = \frac{||\sum_{i=1}^{N}\phi_i - \sum_{i=1}^{e}\phi_i||}{||\sum_{i=1}^{N}\phi_i||}\ .$$

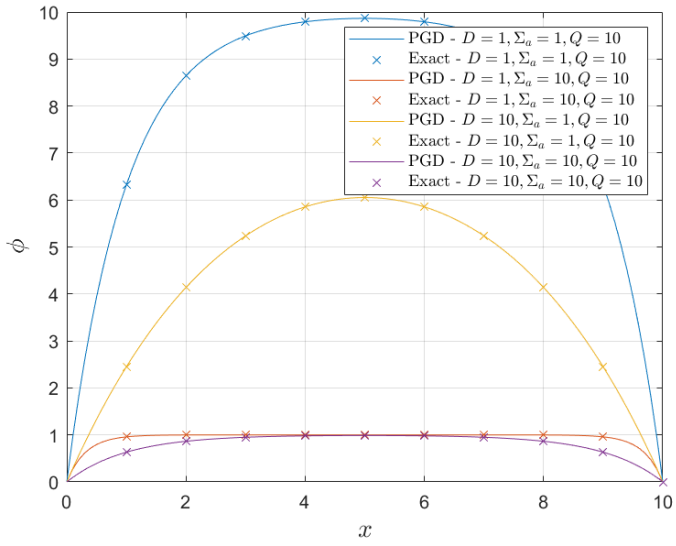
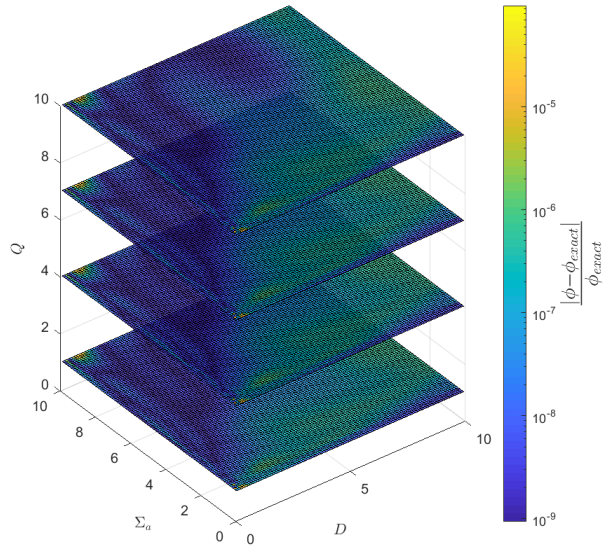
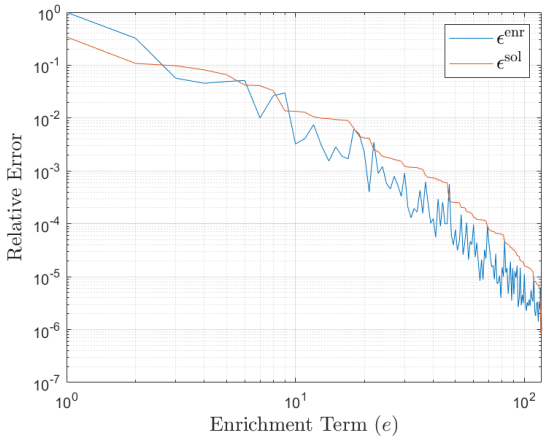


Figure 6.1: Flux solution for some chosen property values, PGD parametric model and analytical solution



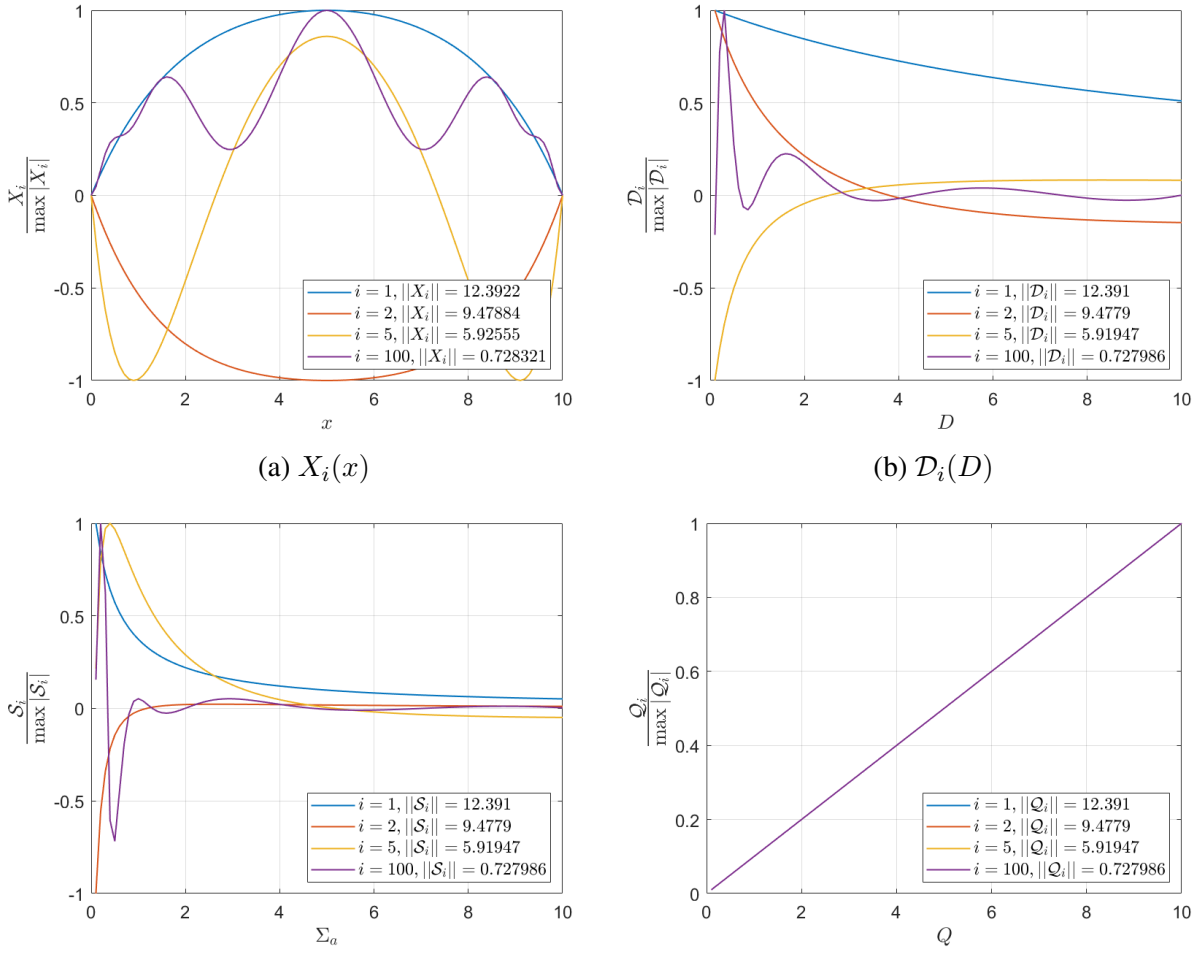
(a) Relative error $\frac{|\phi-\phi_{exact}|}{\phi_{exact}}$ for the full parameter space at $x = 5$



(b) Enrichment error and solution error at each enrichment step

Figure 6.2: Convergence of the parameterized PGD solution for homogeneous 1-D example.

The magnitude of the enrichment error for each enrichment step is an important point of comparison for higher dimensional and more complex problems in the following sections. The cumulative solution error is monotonically decreasing for each additional enrichment. Figure 6.3 shows four different normalized enrichment terms for each dimension. The first, second, fifth, and hundredth enrichment terms were selected for comparison to show how the terms evolve through the enrichment process. The legend of these plots show the L_2 of each enrichment function, illustrating how these functions decay as a function of the enrichment level.



(a) $X_i(x)$ (b) $\mathcal{D}_i(D)$

(c) $\mathcal{S}_i(\Sigma_a)$ (d) $\mathcal{Q}_i(Q)$

Figure 6.3: 1-D enrichment terms at 1, 2, 5, and 100 enrichments for each dimension for homogeneous 1-D example.

6.5.1.1 *Uncertainty Quantification for 1-D Homogeneous Diffusion*

This section shows the UQ analysis for the homogeneous neutron diffusion example using both the PGD and the analytical (exact) solutions, Equations (6.22) and (6.21). Two QOIs (average flux and peak flux) are sampled with a standard Monte Carlo technique, assuming either a uniform or a normal distribution for the uncertain material parameters. The parameter distributions are defined as:

$$f(D, \Sigma_a, Q) \sim \mathcal{U}(\mu_D - \sigma_D, \mu_D + \sigma_D)\mathcal{U}(\mu_{\Sigma_a} - \sigma_{\Sigma_a}, \mu_{\Sigma_a} + \sigma_{\Sigma_a})\mathcal{U}(\mu_Q - \sigma_Q, \mu_Q + \sigma_Q) \,, \tag{6.23a}$$

$$f(D, \Sigma_a, Q) \sim \mathcal{N}(\mu_D, \sigma_D)\mathcal{N}(\mu_{\Sigma_a}, \sigma_{\Sigma_a})\mathcal{N}(\mu_Q, \sigma_Q) \,, \tag{6.23b}$$

where $\mathcal{U}(a, b)$ is the uniform distribution in the interval $[a, b]$ and $\mathcal{N}(a, b)$ is the normal distribution with mean a and standard deviation b. In our numerical tests, we have chosen $\mu_D = \mu_{\Sigma_a} = \mu_Q = 5.0$ and $\sigma_D = \sigma_{\Sigma_a} = \sigma_Q = 2.5$. Table 6.1 provides the mean and variance for the QOIs. These sampled means and variances are also compared against the analytical integration of the statistical moments of the PGD solution and the exact solution, see Equation (6.9) for the integration technique. Figure 6.4 shows the probability distribution functions for the QOIs, obtained using 10^6 samples. There is an excellent agreement between the distributions obtained using the exact solution and those computed with the PGD solution. For this example, the uncertainty quantification with a parameterized PGD solution can yield accurate results. Generating a large number of samples from a PGD solution is a fast process, even in a high dimensional uncertain pace.

6.5.2 One-Dimensional Two-Region Example

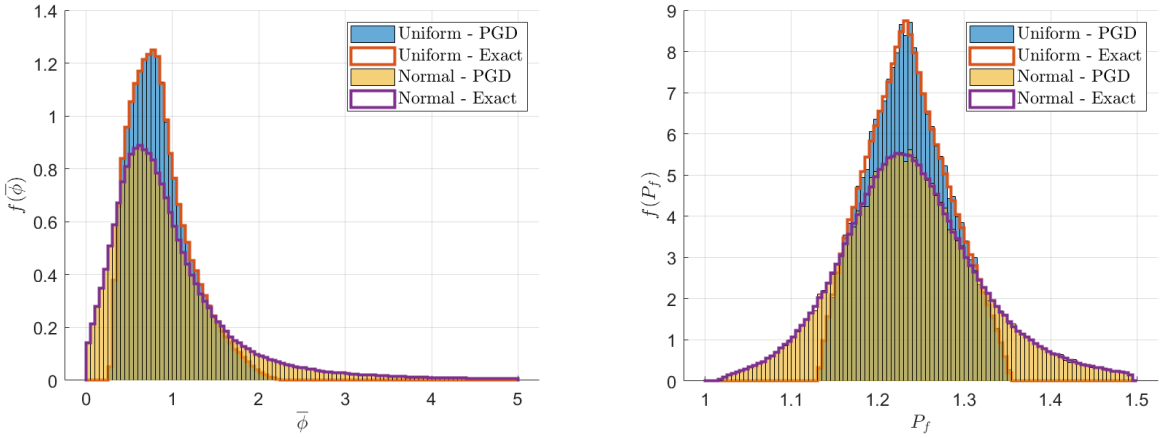
To further the analysis of parameterizing properties in the neutron diffusion equation, PGD was applied to a parametric one-dimensional two-region problem. The neutron diffusion equation and its boundary conditions are given in Equation (6.24).

$$-D_1 \frac{d^2\phi_1}{dx^2} + \Sigma_{a,1}\phi = Q_1 \,, \quad x \in [0, 2.5] \,, \tag{6.24a}$$

$$-D_2 \frac{d^2\phi_2}{dx^2} + \Sigma_{a,2}\phi = Q_2 \,, \quad x \in [2.5, 5] \,, \tag{6.24b}$$

Table 6.1: Mean and variance for average flux and peaking factor (10^6 samples) for homogeneous 1-D example.

	Average Flux $\bar{\phi}$				Peaking Factor P_f			
	Uniform		Normal		Uniform		Normal	
	μ	σ	μ	σ	μ	σ	μ	σ
Sampling - Exact	0.8624	0.3541	1.0447	1.0836	1.2349	0.0471	1.2377	0.0805
Sampling - PGD	0.8624	0.3543	1.0444	1.0785	1.2351	0.0471	1.2379	0.0805
Integration - Exact	0.8623	0.3545	1.0449	1.0865				
Integration - PGD	0.8637	0.3591	1.0445	1.0785				



(a) Average flux, $\bar{\phi}$

(b) Peaking Factor, P_f

Figure 6.4: Probability distributions for average flux and peaking factor(10^6 samples) for homogeneous 1-D example.

$$\left.\frac{d\phi_1}{dx}\right|_{x=0} = \phi_2(5) = 0\ , \tag{6.24c}$$

$$\phi_1(2.5) = \phi_2(2.5), \quad \left.\frac{d\phi_1}{dx}\right|_{x=2.5} = \left.\frac{d\phi_2}{dx}\right|_{x=2.5}\ . \tag{6.24d}$$

The parameters $D_{1,2}$, $\Sigma_{a,1,2}$, and $Q_{1,2}$ are uncertain and have uniform probability in the range between 0.1 and 10 cm, cm^{-1}, and n/cm^3·s, respectively. The resulting decomposition to the neutron

flux is shown by Equation (6.25).

$$\phi = \sum_{i=1}^{1000} \phi_i = \sum_{i=1}^{1000} X_i(x) \prod_{j=1}^{2} \mathcal{D}_i^{d_j}(D_j) \mathcal{S}_i^{\sigma_j}(\Sigma_{a,j}) \mathcal{Q}_i^{q_j}(Q_j) \; . \tag{6.25}$$

In a 1D M-region domain, the PGD system contains $2M$ linear spatial operators (one for diffusion in x in each region, one for reaction in x in each region). The total number of dimensions on the parametric model is $3M + 1$ (one for the physical space x, and three for the uncertain space in D, Σ_a, and Q for each region). First-order continuous finite elements are used for the x dimension with 100 elements, while a finite volume discretization is used for the parameter dimensions, with 100 elements in each dimension spanning from 0.1 to 10. The enrichment process was stopped at 1000 terms, which resulted in a final enrichment error of $\sim 10^{-5}$. Four snapshots of the solution were taken of the PGD parametric solution: one homogeneous and three heterogeneous. Table 6.2 details these snapshots and the resulting L_2 error from the exact solution. Figure 6.5 shows the resulting solutions from the snapshots; the analytical solution is also plotted. These global errors show that the accuracy of the PGD parametric solution is highly dependent on the inputted parameter values.

Table 6.2: Cases chosen to compute relative difference from exact solution

Case	D_1	D_2	$\Sigma_{a,1}$	$\Sigma_{a,2}$	Q_1	Q_2	$\|\|\phi - \phi_{\text{ref}}\|\|/\|\|\phi_{\text{ref}}\|\|$
1	5.0	5.0	5.0	5.0	5.0	5.0	0.000155867
2	9.2	7.6	0.8	7.8	5.7	3.4	0.0709736
3	2.9	3.9	0.6	9.4	4.7	1.7	0.0734167
4	7.6	5.7	5.4	1.3	0.2	8.0	0.0979055

To confer the results from the snapshots, the following shows how error depends on the full parameter space. Figures 6.6a and 6.6b show the error of the PGD solution as function of the diffusion coefficient and absorption in each region, respectively. This error plots are meant to show where in the parameter space the PGD solution deviates from the exact solution. These

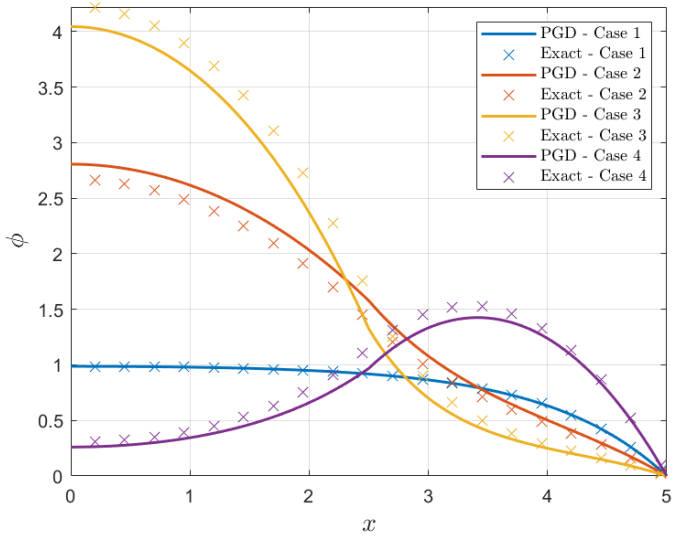
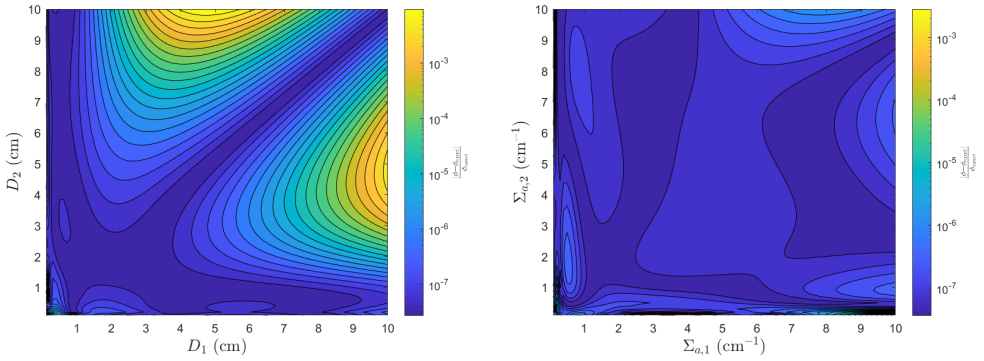


Figure 6.5: Flux solution for some chosen property values for two-region 1-D example, PGD parametric model and analytical solution

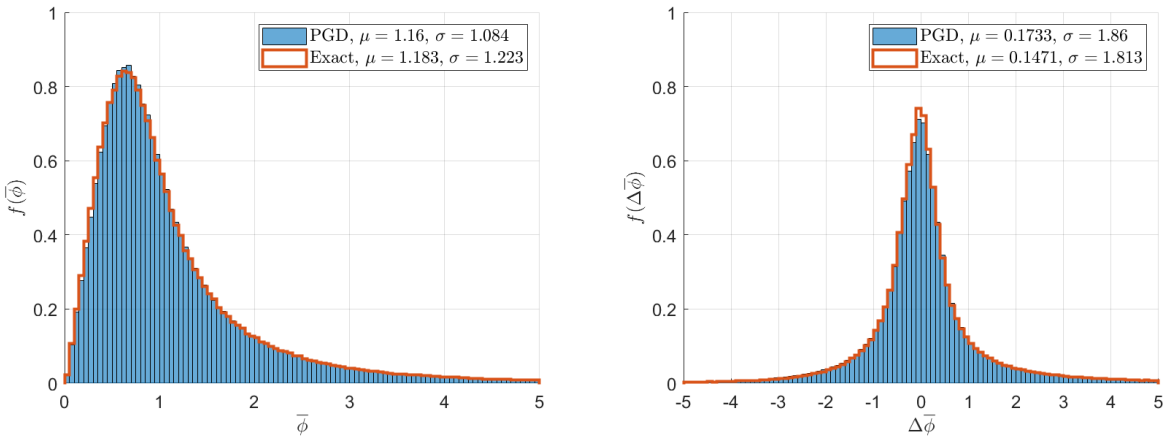
profiles indicate that error is highest when D is large and Σ_a is small, with a dip where the two regions' values are equivalent. This observation demonstrates two features of the accuracy of the PGD solution. First, the locations of highest error show that error from the spatial discretization is evident. Second, the dip where the two regions' properties equal indicate that the PGD solution is highly dependent on the degree of heterogeneity of the material.



(a) $D_{1,2}$ error, Σ_a and Q set to Case 1 (b) $\Sigma_{a,1,2}$ error, D and Q set to Case 1

Figure 6.6: Relative error $\frac{|\phi - \phi_{exact}|}{\phi_{exact}}$ for the full parameter space at $x = 5$ for 1-D two-region problem

To illustrate the applicability of the parametric PGD, we perform a UQ analysis of the 1D neutron diffusion example using the analytical (exact) solution and the PGD solution. As stated previously, the probability for each parameter is uniformly distributed between 0.1 and 10. Figures 6.7a and 6.7b show the probability density functions for average flux and changed flux, respectively, computed from sampling 10^6 times. The center region ($x \in [0,\ 2.5]$) is the rod region. The uncertainty distribution from sampling the PGD solution a million times has reasonable agreement with the distribution of the exact solution with the same samples. If more accuracy is needed, then more enrichment terms would be required. However, even for this example, the later enrichments converge quite slowly: about an order of magnitude more terms for another order of magnitude of accuracy. This issue is only exacerbated as the number of uncertain space dimensions (i.e., number of material regions) is increased [103].



(a) Average flux, $\bar{\phi}$

(b) Change in neutron population, $\Delta\bar{\phi}$

Figure 6.7: Probability distributions for average flux and change in neutron population (10^6 samples) for two-region 1-D example.

6.5.3 2-D IAEA Benchmark Problem

In this second case, we apply PGD to a heterogeneous 2-D configuration in order to examine its performance with a more complex geometry. The chosen problem is based on the ANL Benchmark Problem Book problem 11-A1 [58] (the 3-D version of this benchmark problem is analyzed in the next section). The 2-D geometry layout is shown in Figure 2.8, where the numbers indicate the the material index. Only a quarter core is represented, with symmetry lines at the top and left. Each square cell (of size 20 cm $\times$ 20 cm) contains one material. There are four material types in the 2-D version of the problem.

6.5.3.1 Parameterizing 2-D IAEA Problem

Parameterizing the properties of this geometry results in a PGD decomposition in 13 dimensions: two in space (x and y), four for the diffusion coefficient of the different materials, four for the absorption cross section, and three for neutron source (there are no sources in the reflector zones). After decomposing the material properties in a manner suitable for a fast PGD process, the neutron diffusion operator contained a total of 66 linear operators (L in Equation (2.2b)) in each dimension. The spatial and parameter dimensions are discretized using continuous finite elements and finite volume, respectively. Table 6.3 provides discretization details for each parameter (minimum and maximum value in each of the 13 dimensions). The resulting PGD solution is given by the expression shown in Equation (6.26):

$$\phi(x, y, D_1, ..., D_4, \Sigma_1, ..., \Sigma_4, Q_1, ..., Q_3) = \sum_{i=1}^{2000} X_i(x) Y_i(y) \prod_{j=1}^{4} \mathcal{D}_i^{D_j}(D_j) \prod_{j=1}^{4} \mathcal{S}_i^{\sigma_j}(\Sigma_{a,j}) \prod_{j=1}^{3} \mathcal{Q}_i^{q_j}(Q_j) \ . \quad (6.26)$$

The PGD solution required 2000 enrichment terms, for an enrichment error of approximately 10^{-4}. The memory consumed by the PGD solution is 23 MB, while a non-PGD solution in a 13-dimensional space would likely consume twenty orders more. Table 6.4 shows the parameter values of four arbitrary cases chosen to illustrate how well the parameterized PGD solution per-

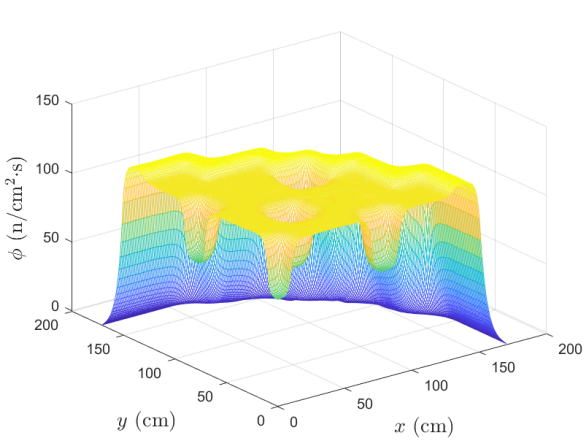
Table 6.3: Details on the discretization of the PGD parametric model

Dimension		No. of Elements	Min.	Max.
x	(cm)	170	0	170
y	(cm)	170	0	170
D_1, D_2, D_3	(cm)	100	0.2	0.8
D_4	(cm)	100	0.15	0.6
$\Sigma_{a,1}$	(cm^{-1})	100	0.0425	0.17
$\Sigma_{a,2}$	(cm^{-1})	100	0.065	0.26
$\Sigma_{a,3}$	(cm^{-1})	100	0.04	0.16
$\Sigma_{a,4}$	(cm^{-1})	100	0.005	0.02
Q_1, Q_2, Q_3	($n/cm^3 \cdot s$)	100	5.0	20

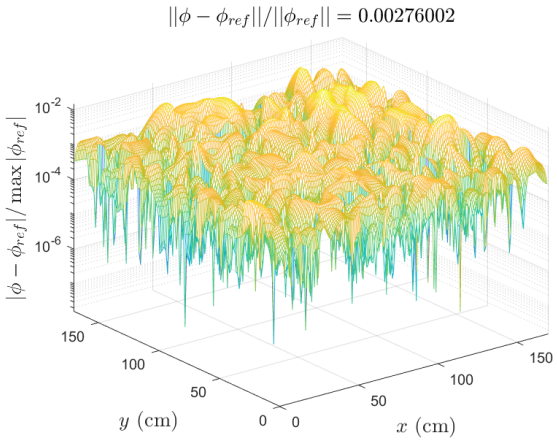
forms against a non-parametric solution; the relative error between the PGD parametric solution and a reference, non-parametric solution, are also provided. Figure 6.8 shows the parametric solution from the four cases as well as their relative errors when compared against non-parametric solutions. The non-parametric counterparts are computed using the PGD method with just a spatial decomposition of the flux (i.e., a 2-dimensional problem only), the material properties being set to the specific case values.

Table 6.4: Cases chosen to compute relative difference from non-parametric solution

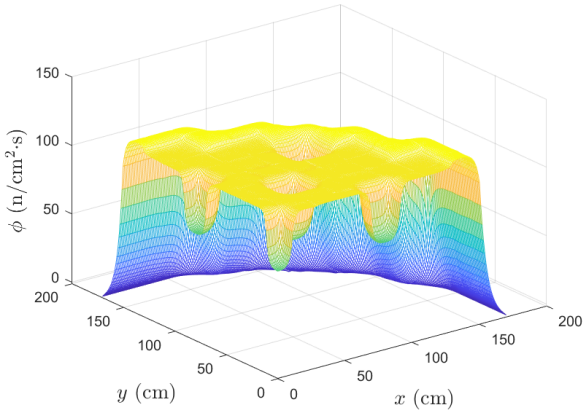
Case	Prop.	Value in Material Zone 1	2	3	4	Relative Error $\|\|\phi - \phi_{\text{ref}}\|\|/\|\|\phi_{\text{ref}}\|\|$
1	D	0.4	0.4	0.4	0.3	2.760e-3
	Σ_a	0.085	0.13	0.08	0.01	
	Q	10	10	10	0	
2	D	0.23	0.53	0.77	0.3	3.287e-3
	Σ_a	0.085	0.13	0.08	0.01	
	Q	10	10	10	0	
3	D	0.4	0.4	0.4	0.3	5.460e-3
	Σ_a	0.049	0.075	0.154	0.01	
	Q	10	10	10	0	
4	D	0.23	0.53	0.77	0.3	7.185e-3
	Σ_a	0.049	0.075	0.154	0.01	
	Q	10	10	10	0	



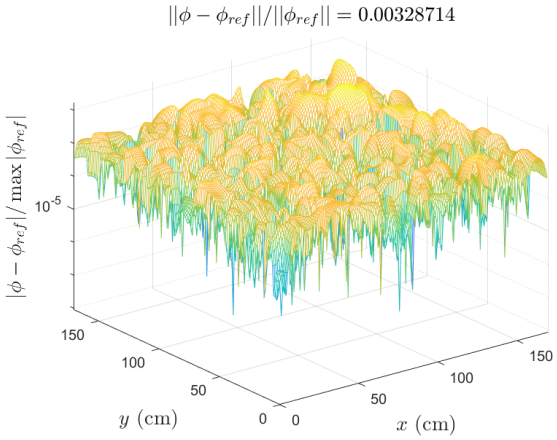
(a) Case 1 Solution



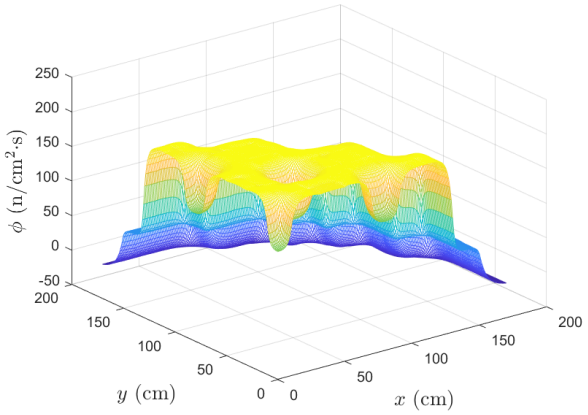
(b) Case 1 Difference



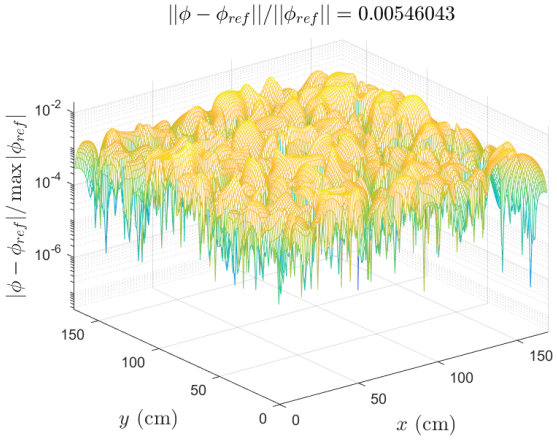
(c) Case 2 Solution



(d) Case 2 Difference

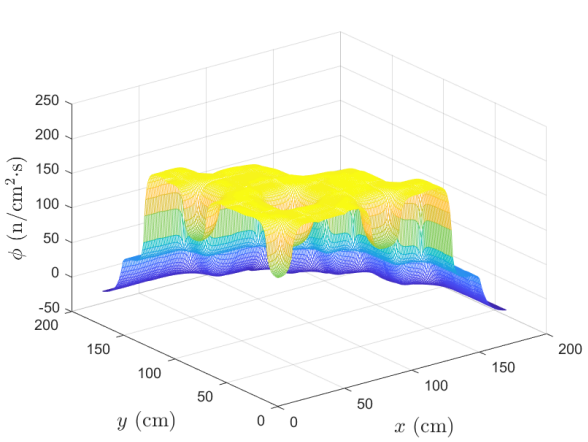


(e) Case 3 Solution

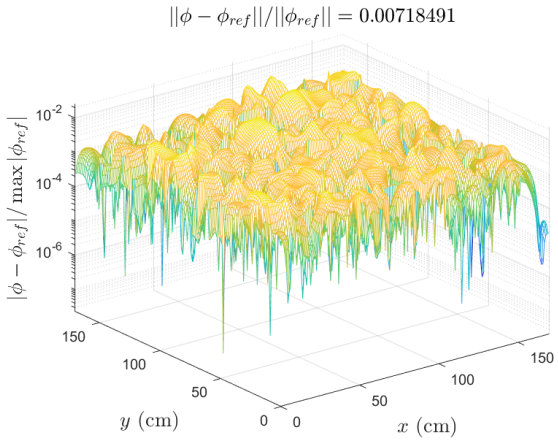


(f) Case 3 Difference

Figure 6.8: Parametric solutions and relative errors compared to non-parametric solutions



(g) Case 4 Solution



(h) Case 4 Difference

Figure 6.8: Continued

The errors given in Table 6.4 and Figure 6.8 show that the spatial error is approximately an order of magnitude higher than the enrichment error. Using the enrichment convergence rate shown in Figure 6.9, one can infer that reducing the enrichment error by another order of magnitude would require another order of magnitude in number of enrichments. This slow rate of convergence is due to the high-dimensionality of the problem; see also [103] for another such instance. However, in the context of UQ for neutronic applications, the uncertainty due to the parameters are often much larger than this level of convergence error.

Figure 6.10 shows how the parameterized PGD solution varies as a function of the material property only. To obtain these one-dimensional functions, an integration is performed over all other dimensions (space and the other parameters); the resulting one-dimension functions are normalized for plotting convenience. As expected, the dependence of the solution of the source parameter is linear; see Figure 6.10c. The rest of these one-dimensional functions are meant to give a qualitative relative sensitivity of the average flux with respect to each material property. Figure 6.10a shows that the closer the material zone is to the outer edge of the domain, the greater the effect the uncertainty of diffusion coefficient has on the flux average. Figure 6.10b shows that effect of the absorption cross-section uncertainty is directly related to the volume of each material zone.

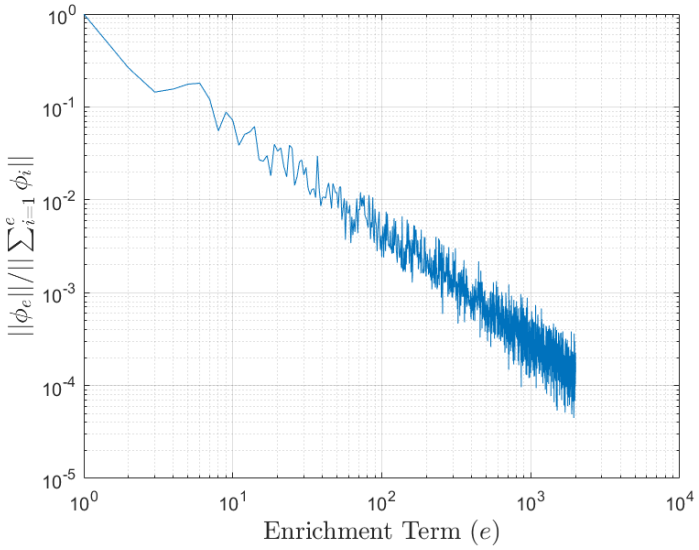
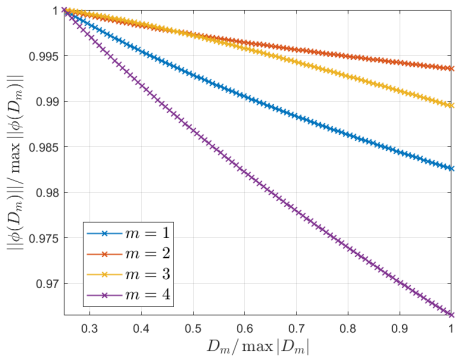
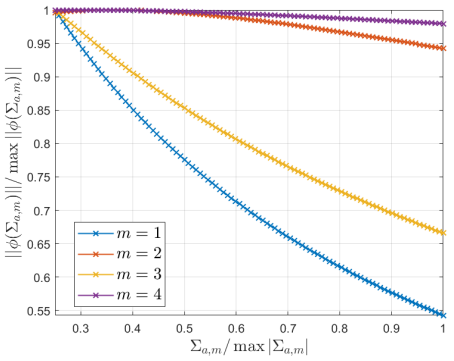


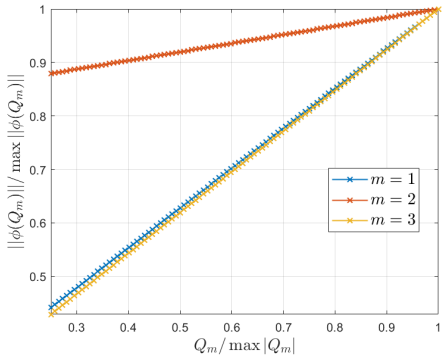
Figure 6.9: Enrichment error at each enrichment step



(a) D



(b) Σ_a

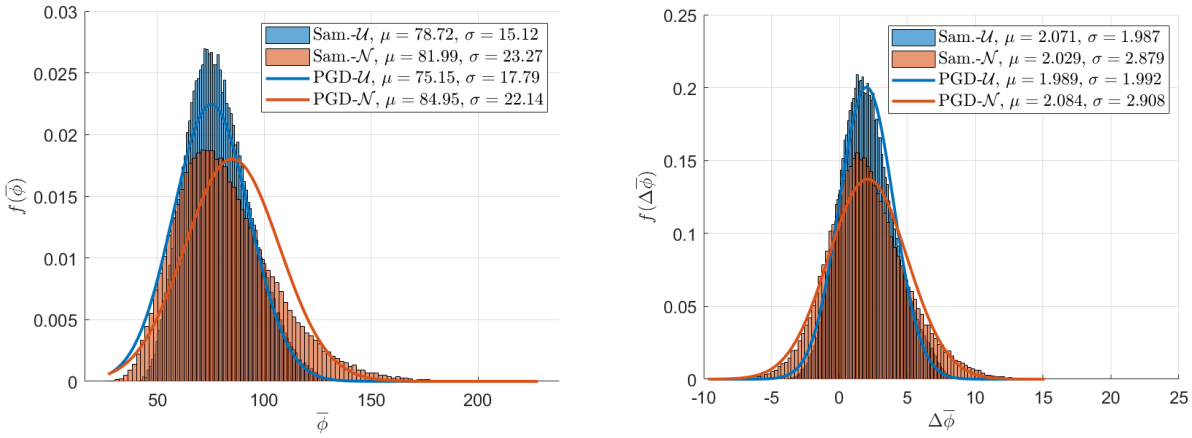


(c) Q

Figure 6.10: Relative norm of flux as a function of material variation

6.5.3.2 *Uncertainty Quantification of 2-D Heterogeneous Problem*

This section details the relevant UQ results from the obtained PGD solution of 2-D IAEA benchmark problem. The quantities of interest for this analysis are the average neutron flux and the change in total neutron population. The uncertain parameters are, again, assumed to be independent variables, thus the joint probability distribution function is the product of the univariate probability distribution functions. Uniform and normal probability distributions are employed. The mean for each parameter distribution is set to Case 1 of Table 6.4 and the standard deviation σ is chosen to be 33% of the mean. Figure 6.11 shows the probability distribution as a result of sampling the PGD solution as well as a normal distribution with mean value and variance computed using Equation (6.9) (direct integration of the PGD solution).



(a) Average Flux

(b) Change in Total Neutron Population

Figure 6.11: Probability distributions for average flux and changed population. Sampling results computed using 10^5 samples. PGD results are a normal distribution with the mean and variance computed using Equation (6.9).

The results in Figure 6.11 are meant to illustrate the ability of the PGD solution to generate a probability distribution for some quantities of interest and estimate the propagated uncertainty. The normal distributions in Figure 6.11 are meant to visually compare the actual probability dis-

tributions obtained from sampling against Gaussian distributions that have means and variances calculated using Equation (6.9). For the change in the total neutron population distribution, a Gaussian distribution would be an appropriate fit to the the actual probability distribution assumption. For the average flux QOI, the obtained probability distribution deviates significantly from a Gaussian distribution.

6.5.4 3-D IAEA Benchmark Problem

This fourth and final case examines the effectiveness of PGD for a 3-D heterogeneous problem, the 3-D version of the ANL Benchmark problem used in the previous section. The x–y layout of the 2-D geometry of Figure 2.8 is extruded in the z direction. This extrusion is shown in Figure 2.11 with the addition of a fifth material region. Material properties in each region of the domain are parameterized. Additionally, in Section 6.5.4.3, the position of the each control rod is also parameterized (the x–y locations of the control rods can be found in Figure 2.8).

6.5.4.1 Parameterizing 3-D IAEA Problem Material Properties

Parameterizing the properties of this geometry results in a PGD process in 16 dimensions: three in space, five zones for the diffusion coefficient, five zones for the absorption cross section, and three zones for the neutron source The PGD system contains 116 linear operators (L in Equation (2.2b)) in each dimension. The spatial and parameter dimensions are discretized using continuous finite elements and finite volume, respectively. Table 6.3 and 6.5 provide discretization details for each dimension. The PGD solution is given by Equation (6.27):

$$\phi(x,y,z,D_1,...,D_5,\Sigma_1,...,\Sigma_5,Q_1,...Q_3) = \sum_{i=1}^{1300} X_i(x)Y_i(y)Z_i(z)\prod_{j=1}^{5}\mathcal{D}_i^{D_j}(D_j)\prod_{j=1}^{5}\mathcal{S}_i^{\sigma_j}(\Sigma_{a,j})\prod_{j=1}^{3}\mathcal{Q}_i^{q_j}(Q_j)\ . \quad (6.27)$$

The solution contains 1300 enrichment terms, with in a relative enrichment error of approximately 5×10^{-4}. The memory consumed by the PGD solution is 21 MB, while a full solution in

Table 6.5: Details on the discretization of the 3-D PGD parametric model

Dimension		No. of Elements	Min.	Max.
z	(cm)	380	0	380
D_5	(cm)	100	0.15	0.6
$\Sigma_{a,5}$	(cm^{-1})	100	0.0275	0.11

16 dimensions would consume many orders of magnitude more. Table 6.6 shows the parameter values of four arbitrary cases chosen to illustrate how accurate the parameterized solution against a non-parametric solution. Figure 6.12 shows the parametric solution from the one of the four specified cases and its error when compared against a non-parametric solution. The non-parametric solutions are obtained using the PGD technique over spatial coordinates only, while the material properties are set to the values for each specific case.

The errors reported in Table 6.6 and Figure 6.12 show a performance similar to the 2-D results. The convergence shown in Figure 6.13 shows approximately the same convergence as for the 2-D, while the dimensionality of the problem (16 dimensions now, as opposed to 13 in the 2-D case) increased only slightly. One may infer that, for this type of problems, the rate of convergence is more dependent on the total number of dimensions in the problem rather than the number of expansion terms needed in the decomposition of the linear operator (the value of L in Equation (2.2b)).

6.5.4.2 Uncertainty Quantification for 3-D IAEA Problem

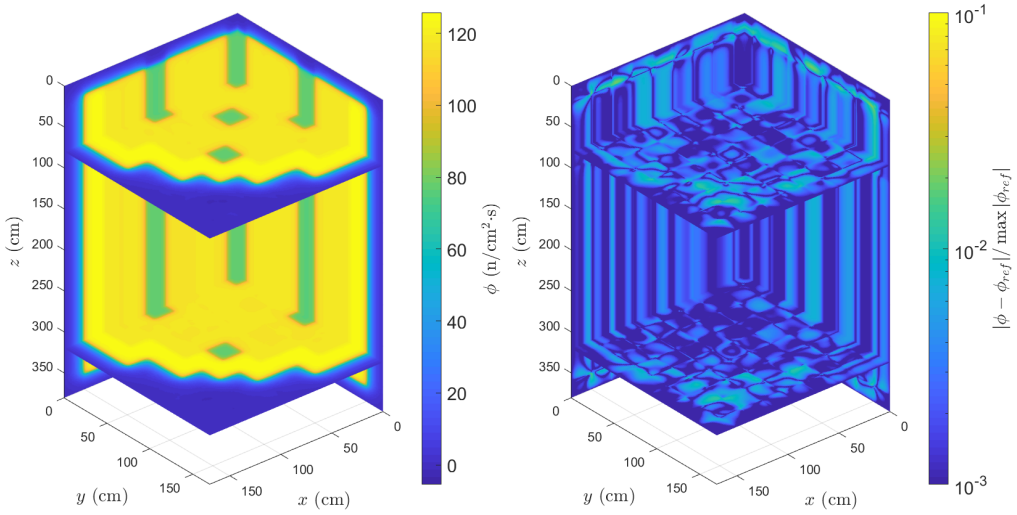
This section details the relevant UQ results from the obtained PGD solution of the 3-D IAEA benchmark problem. The quantities of interest and parameter distributions are the same as for the 2-D example in Section 6.5.3.2. Figure 6.14 shows the probability distributions for two QOIs, obtained from sampling the PGD solution. Gaussian distributions with the mean and variance computed directly from the PGD solution (Equation (6.9)) are also included in the graph. The graph legends display the mean and variance values, obtained from Monte Carlo sampling or direct integration of the PGD solution.

The results in Figure 6.14 are meant to exemplify the ability of sampling multiple times a PGD solution in a cost-effective manner, hence generating well-resolved probability distribution

for the quantities of interest and propagating uncertainty. These distributions show a very similar behavior as the 2-D results, with the same conclusions on the accuracy of the Gaussian distribution assumptions.

Table 6.6: Cases chosen to compute relative difference from non-parametric solution

Case	Prop.	Value in Material Zone					Relative Error
		1	2	3	4	5	$\|\|\phi - \phi_{\text{ref}}\|\|/\|\|\phi_{\text{ref}}\|\|$
1	D	0.4	0.4	0.4	0.3	0.3	4.515e-3
	Σ_a	0.085	0.13	0.08	0.01	0.055	
	Q	10	10	10	0	0	
2	D	0.23	0.53	0.77	0.3	0.3	5.343e-3
	Σ_a	0.085	0.13	0.08	0.01	0.055	
	Q	10	10	10	0	0	
3	D	0.4	0.4	0.4	0.3	0.3	1.142e-2
	Σ_a	0.049	0.075	0.154	0.01	0.055	
	Q	10	10	10	0	0	
4	D	0.23	0.53	0.77	0.3	0.3	1.366e-2
	Σ_a	0.049	0.075	0.154	0.01	0.055	
	Q	10	10	10	0	0	



(a) Case 1 Solution

(b) Case 1 Difference

Figure 6.12: Parametric solution and relative error compared to non-parametric solution

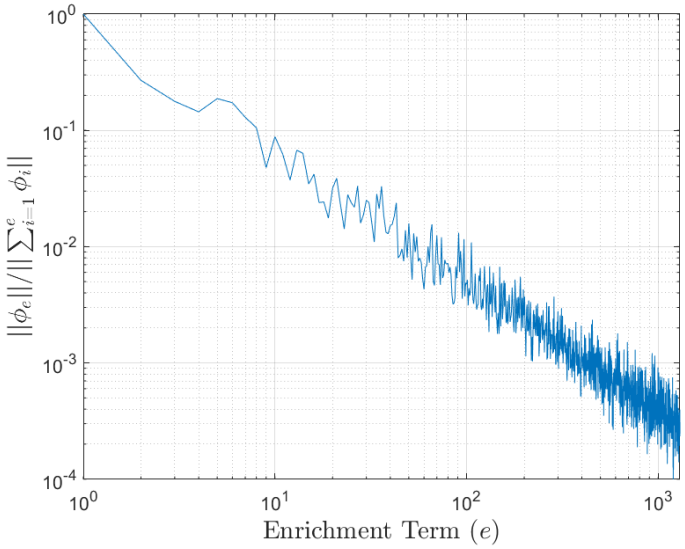
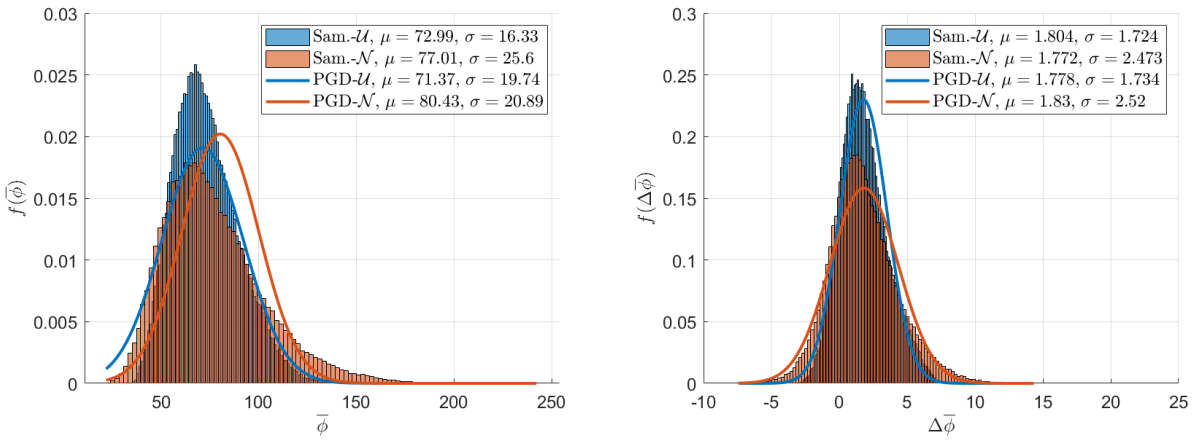


Figure 6.13: Enrichment error at each enrichment step



(a) Average Flux

(b) Change in Total Neutron Population

Figure 6.14: Probability distributions for the average flux and change in total neutron population. Sampling results computed using 10^5 samples. PGD results are a normal distribution with the mean and variance computed using integration method.

6.5.4.3 Parameterized Control Rod Movement

This section presents results from parmeterizing the insertion length of each control rod for the 3-D IAEA problem. The insertion length for control rod i is λ_i, where the index for each rod is

labeled in Figure 2.8 and the length is the distance from the top of the fuel height, as shown in Figure 2.11. Parameterizing this geometry results in eight dimensions: three for space and one for each of the five control rods, shown in Equation (6.28). The spatial dimensions are discretized similarly to those of the previous sections, the parameter dimensions are discretized using 18 finite volume elements spanning the entire length of the fuel region (340 cm). The resulting PGD system contains 118 linear operators in each dimension (L in Equation (2.2b)). The PGD solution contains 1000 enrichment terms, with an enrichment relative error of approximately 5×10^{-5}. Table 6.7 shows the parameter values of four arbitrary cases chosen to illustrate the solution and compute the relative difference from a reference, non-parametric solution. Figure 6.15 shows the z-direction profile of the parameterized and reference solution for each of the four cases in Table 6.7.

$$\phi(x, y, z, \lambda_1, ..., \lambda_M) = \sum_{i=1}^{1000} X_i(x) Y_i(y) Z_i(z) \prod_{j=1}^{5} \Lambda_i^j(\lambda_j) \; . \tag{6.28}$$

Table 6.7: Cases chosen to compute the relative error between parametric and non-parametric solutions

	Rod Insertion Length (cm)					Relative Error
Case	1	2	3	4	5	$\|\|\phi - \phi_{\text{ref}}\|\|/\|\|\phi_{\text{ref}}\|\|$
1	80	140	120	260	300	1.192e-3
2	80	100	360	140	60	1.702e-3
3	60	180	200	360	100	1.358e-3
4	140	160	200	60	200	1.076e-3

The errors reported in Table 6.7 (and also given in Figure 6.15) show that the PGD solution is quite accurate compared to a non-parametric solution, with errors less than 0.17%. The accuracy of these results surpass those of the material property parameterizations of Section 6.5.4 while using a smaller number of enrichment terms. This observation seems to indicate that the number of linear operators, the value of L in Equation (2.7) (118 terms for 3-D rod movement parameteri-

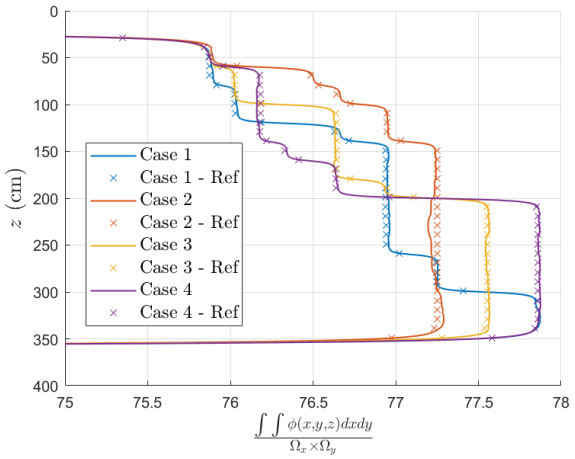


Figure 6.15: Axial profiles for parametric and reference solutions with different rod insertions.

zation versus 66 for 2-D property parameterization), may not directly impact the number of PGD enrichments terms necessary ($\sim$1000 for the 3-D rod parameterization versus $\sim$1300 for the 2-D material parameterization).

An important metric for reactor design is the change in neutron population depending on the insertion depth of each control rod. Figure 6.16 shows this quantity ($\Delta\bar{\phi}$) for each control rod. To compute $\Delta\bar{\phi}$ for rod i, $\bar{\phi}_0$ was computed by extracting all the rods ($\lambda_j = 0$), then $\bar{\phi}_{\text{rod}}$ was computing by extracting all rods except rod i. The worth in Figure 6.16 are scaled by the number of each rod in the entire reactor. The results in Figure 6.16 are meant to illustrate the ability of a PGD approach to produce design-relevant results in much simpler manner than traditional methods, which would require a full 3-D evaluation for every possible configuration of the control rods' layout and insertion length.

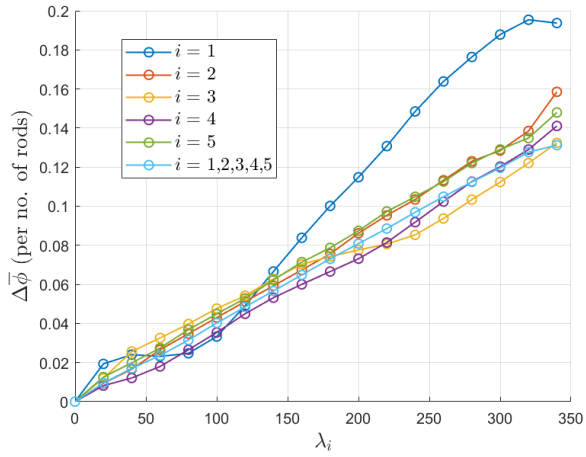


Figure 6.16: Change in total neutron population for each control rod location ($i = 1, \ldots, 5$ and all rods together).

6.6 Discussion

In this chapter, uncertainty quantification (UQ) and design optimization have been performed using PGD techniques applied to neutron diffusion problems. The neutron diffusion equation with external sources, a diffusion-reaction problem with forcing terms, is used as the parametric model. For heterogeneous domains, the dimensionality of the uncertain space can become quite large. Thus, a parameterized PGD solution is sought in a high dimensional space, the natural spatial coordinates as well as each zone-dependent material property. This PGD solution, parameterized in all uncertain variables, can then be used to compute mean, variance, and more generally probability distributions of various quantities of interest. In addition to parameterized properties, parameterized geometrical variations of 3D models have also been considered in this work.

The theory section of this chapter shows an in-depth derivation of the PGD system for parametric heterogeneous problems. The linear operators are decomposed into zone-dependent 1-D operators; although this system construction seems complex, it is purely analytical and completely contained in the pre-processing stage, ultimately resulting in a efficient solution process. The theory section also discusses the utilization of a PGD solution for uncertainty quantification. It shows how a resolved PGD solution can be sampled trivially with a Monte Carlo technique and also dis-

cuss how a PGD solution can be directly integrated to compute the mean and variance of some quantities of interest, given specific types of parameter probability distributions. Finally the section discusses the quantities of interest analyzed for this application of neutron diffusion, including the average solution (average flux), the maximum of the solution (peaking factor), and change in the total neutron population.

In the results section, parametric PGD solutions are applied to four different geometries. The first and second are 1-D problems with a known analytical solution, meant to verify the PGD system construction and solution process. The error and basis functions of the resulting PGD solution show that the solution follows the physics of the problem quite well, but are somewhat dependent on the heterogeneity of the geometry. UQ results are also presented for these problems and show that evaluating the parameterized PGD at sampled values of the uncertain parameters yield accurate results. The third and fourth geometries involve 2-D and 3-D heterogeneous domains, respectively. The results obtained using these models illustrates the applicability of PGD techniques to propagate uncertainties effectively in such high dimensional problems, a task often impractical with traditional methods. The 3-D application also investigates parameterizing control rod insertion for design optimization considerations. All of these application also show the behavior of the enrichment convergence for highly-dimensional problems. It is observed that convergence of the PGD solution depends noticeably on the total number of dimensions in the parametric problem.

In summary, this PGD application to parameterized neutron diffusion models shows promising results for practical application of realistic uncertainty propagation and design optimization. Considerations for the future development of this work would include increasing the complexity of the governing law, e.g., including application to multigroup criticality neutron diffusion and neutron transport problems. In addition, there is also a need to better understand enrichment convergence in highly dimensional spaces for UQ calculations.

7. CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

In this work, a proper generalized decomposition approach was implemented and investigated for various nuclear science and engineering applications. The goal of the research was to tailor the PGD algorithm for neutron diffusion and transport problems and asses the viability of the method to reduce the computational burden of evaluating these multi-dimensional systems. These applications include,

1. Multigroup neutron diffusion eigenvalue calculations
2. Fine-group neutron diffusion problems
3. Neutron transport problems
4. Parametric uncertainty quantification for neutron diffusion

The detailed conclusions for each these applications can be found in their corresponding chapters. However, several general conclusions can be made based on collective observations.

Broadly speaking, PGD is an highly efficient method and its ability to reduce the dimensionality of certain multi-dimensional models is without question. PGD performance is impressive when compared to its full-order counterpart for many of the problems presented in this dissertation and in the myriad of other PGD works. However, it is apparent from these applications that PGD performance is heavily dependent on several features of the models at hand: number of dimensions, heterogeneity of the domain, and separability of the full-order solution. Admittedly, these factors are not independent, but it is difficult to correlate them quantitatively.

Contrary to the idealistic vision of PGD, the burden of evaluating a model with PGD is not linearly dependent on the number of dimensions, as the number of enrichments required increases significantly when dimensionality is increased. However, the resulting run-time and system size when compared to the full-order model improves continuously with increasing dimensions, if a

reasonably accurate solution is desired. The efficiency of PGD evaluation suffers heavily if highly accurate solutions are needed. PGD can be highly useful if only a rough picture of the model is needed, which is generally the purpose of reduced order models.

A recurring theme throughout this dissertation is that PGD performance suffers heavily in its application to highly heterogeneous domains. PGD dependence on heterogeneity seems to be the primary drawback of its application to reactor physics. The domains involving nuclear reactors, especially for high fidelity simulations, contain multitudinous materials with severe discontinuities in property values. For these models, utilizing full-order evaluation methods is most likely the best approach. However, simplified models exist and are common practice for reactor analysis, these simplifications is where PGD can prosper.

The performance of PGD ultimately hinges on the separability of the model's solution, that is the number of separated terms, or enrichments, required to represent the solution to a certain accuracy. If the optimal decomposition of a solution requires a certain amount of terms, then PGD requires at least that many enrichment evaluations, most likely more. In the analysis of this dissertation, the separability of 2-D models is quantified using SVD, which provides the optimal decomposition. It is observed that PGD follows the SVD convergence closely for the first few enrichments, but diverges significantly when more terms are added. Furthermore, solutions that contain discontinuities or non-smoothness not aligned with any of the coordinates requires an exorbitant number of enrichments, to the point where the separated representation is no longer a reduced order model. These kind of solutions are especially prevalent in transport solutions, where no diffusion is present. To conclude, a decent amount of knowledge about the physics and the domain is needed in order to determine, *a priori*, if PGD will be an effective technique.

The PGD methodology is still very much in its adolescence, especially in its application to nuclear science and engineering. For the applications presented in this dissertation, PGD shows a great amount of promise as a reduced order modeling technique and a method for reducing the computational burden for neutron diffusion problems. However, much more development and analysis is required to conclusively determine the proper place in which PGD can reside in the

nuclear community. The decomposition of spatial dimensions is a huge crux in the PGD methodology; more promising is the decomposition of other coordinates including energy, angle, time, and parametric dimensions, where PGD can truly serve its purpose.

7.2 Recommendations for Future Work

The main recommendation for future work with PGD is to apply it to more complex examples and perform more analysis on its convergence. This section details potential avenues of PGD for each of the applications presented in this dissertation. The following subsections itemizes these recommendations.

7.2.1 Mulitgoup Criticality

For PGD in criticality calculations, more analysis and verification is needed on PGD for reactor-physics problems including more complex geometries and finer-group problems, which usually exist in realistic reactor benchmarks. Although, it is expected this PGD approach will have much more difficulty evaluating highly heterogeneous geometries. Some advanced PGD techniques exist that could mitigate this difficulty, including adaptive subspace methods [104], AMG methods [105], and domain decomposition [106], for instance. Furthermore, PGD algorithms should be compared with more traditional reactor physics codes, like TRIVAC and PARCS, as examples.

7.2.2 Space-Energy Decomposition

PGD with space-energy decomposition showed very promising results, especially when the domain was only mildly heterogeneous. To eliminate the performance ambiguity with heterogeity, a possile approach would be to remove the decomposition in space, where the solution has a separation of the 3-D spatial dimension and energy. Furthermore, since PGD seemed to have difficulty resolving the fast-resonance region in the fine-group calculation, a full-order model could be used for this region, since no Gauss-Seidel iteration is required, and use PGD for the thermal region. Additionally, different energy discretizations could be explored including multiband methods [107] and finite element methods [108, 109]. Since the evaluation of the energy dependent variables in

the PGD space-energy separation involve an algebraic equation, continuous energy representation is also feasible.

7.2.3 Neutron Transport

The results from Section 5 show that transport, in general, is a very difficult problem to capture using PGD with space separation. Using PGD by itself may not be viable method to improve performance for these problems. However, it is feasible that PGD could be used as a corrective measure to improve source iteration convergence. For instance, using a FOM evaluation technique for the uncollided flux, then using PGD for source iteration, due to the separability of $\delta\psi$. PGD could also be used as an accelerator, solving for $\delta\psi$ between each FOM source iteration using a small number of enrichments to correct the solution. Finally, the Residual Monte Carlo technique could also be utilized in conjunction with PGD, whereby performing a very coarse PGD evaluation then using Monte Carlo methods to correct the residual [110].

7.2.4 Parameterized Neutron Diffusion

In this chapter, PGD proved to be effective for producing parameterized solutions for very high-dimensional problems. The work only used these solutions for the purposes of rudimentary UQ. Therefore, potential avenues of future would be to apply these PGD parametric solutions for realistic and more useful engineering purposes. A very promising and exciting research involves the dynamic data-driven application system (DDDAS) [38, 39, 100, 111, 112]. DDDAS is real-time simulation tool that is able to represent physical models under real-time perturbations, develop data-driven models, and consider control environments. Including these functionalities individually is commonly employed with traditional computing platforms, but embracing all of them is significantly more difficult and a possible avenue of relief is utilizing PGD. In addition to UQ and data-based simulation, PGD can also be applied to more complex nuclear engineering models including multigroup problems, criticality calculations, and the megalomaniac that is neutron transport.

REFERENCES

[1] 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*, vol. 2, no. 11, pp. 559–572, 1901.

[2] R. Pinnau, *Model Reduction via Proper Orthogonal Decomposition*, ch. 5, pp. 95–109. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[3] S. Volkwein, "Proper orthogonal decomposition: Theory and reduced-order modelling," *Lecture Notes, University of Konstanz*, vol. 4, no. 4, 2013.

[4] J. Lumley, A. Yaglom, and V. Tatarski, "Atmospheric turbulence and radio wave propagation," *Jounal of Computational Chemistry*, vol. 23, no. 13, pp. 1236–1243, 1967.

[5] G. Berkooz, P. Holmes, and J. L. Lumley, "The proper orthogonal decomposition in the analysis of turbulent flows," *Annual review of fluid mechanics*, vol. 25, no. 1, pp. 539–575, 1993.

[6] C. W. Rowley, "Model reduction for fluids, using balanced proper orthogonal decomposition," *International Journal of Bifurcation and Chaos*, vol. 15, no. 03, pp. 997–1013, 2005.

[7] Y. Cao, J. Zhu, Z. Luo, and I. Navon, "Reduced-order modeling of the upper tropical pacific ocean model using proper orthogonal decomposition," *Computers & Mathematics with Applications*, vol. 52, no. 8, pp. 1373 – 1386, 2006.

[8] R. Ştefănescu and I. Navon, "Pod/deim nonlinear model order reduction of an adi implicit shallow water equations model," *Journal of Computational Physics*, vol. 237, pp. 95 – 114, 2013.

[9] G. Kerschen, J.-C. Golinval, A. F. Vakakis, and L. A. Bergman, "The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: An overview," *Nonlinear Dynamics*, vol. 41, pp. 147–169, Aug 2005.

[10] A. Corigliano, M. Dossi, and S. Mariani, "Model order reduction and domain decomposition strategies for the solution of the dynamic elastic-plastic structural problem," *Computer Methods in Applied Mechanics and Engineering*, vol. 290, pp. 127 – 155, 2015.

[11] 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*, vol. 71, pp. 217 – 229, 2014.

[12] A. Buchan, A. Calloo, M. Goffin, S. Dargaville, F. Fang, C. Pain, and I. Navon, "A pod reduced order model for resolving angular direction in neutron/photon transport problems," *Journal of Computational Physics*, vol. 296, pp. 138 – 157, 2015.

[13] 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*, vol. 95, no. 12, pp. 1011–1032, 2013.

[14] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," in *Linear Algebra*, pp. 134–151, Springer, 1971.

[15] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.

[16] F. Chinesta, R. Keunings, and A. Leygue, *The Proper Generalized Decomposition for Advanced Numerical Simulations*, vol. 1. Springer, 2014.

[17] F. Chinesta, A. Ammar, and E. Cueto, "Recent advances and new challenges in the use of the proper generalized decomposition for solving multidimensional models," *Archives of Computational Methods in Engineering*, vol. 17, pp. 327–350, 12 2010.

[18] E. Cancès, M. Defranceschi, W. Kutzelnigg, C. L. Bris, and Y. Maday, "Computational quantum chemistry: A primer," in *Special Volume, Computational Chemistry*, vol. 10 of *Handbook of Numerical Analysis*, pp. 3 – 270, Elsevier, 2003.

[19] P. Ladevèze, *Nonlinear computational structural mechanics: new approaches and non-incremental methods of calculation*. Springer Science & Business Media, 2012.

[20] P. Ladevèze, J.-C. Passieux, and D. Néron, "The latin multiscale computational method and the proper generalized decomposition," *Computer Methods in Applied Mechanics and Engineering*, vol. 199, no. 21-22, pp. 1287–1296, 2010.

[21] A. Nouy and P. Ladevèze, "Multiscale computational strategy with time and space homogenization: A radial-type approximation technique for solving microproblems," *International Journal for Multiscale Computational Engineering*, vol. 2, no. 4, 2004.

[22] A. Ammar, B. Mokdad, F. Chinesta, and R. Keunings, "A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids," *Journal of non-Newtonian fluid Mechanics*, vol. 139, no. 3, pp. 153–176, 2006.

[23] A. Ammar, B. Mokdad, F. Chinesta, and R. Keunings, "A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modelling of complex fluids: Part ii: Transient simulation using space-time separated representations," *Journal of Non-Newtonian Fluid Mechanics*, vol. 144, no. 2, pp. 98 – 121, 2007.

[24] B. Mokdad, E. Prulière, A. Ammar, and F. Chinesta, "On the simulation of kinetic theory models of complex fluids using the fokker-planck approach," *Applied Rheology*, vol. 2, pp. 1–14, 01 2007.

[25] E. Prulière, F. Chinesta, A. Ammar, A. Leygue, and A. Poitou, "On the solution of the heat equation in very thin tapes," *International Journal of Thermal Sciences*, vol. 65, pp. 148 – 157, 2013.

[26] C. Heyberger, P.-A. Boucard, and D. Néron, "Multiparametric analysis within the proper generalized decomposition framework," *Computational Mechanics*, vol. 49, no. 3, pp. 277–289, 2012.

[27] D. González, A. Ammar, F. Chinesta, and E. Cueto, "Recent advances on the use of separated representations," *International Journal for Numerical Methods in Engineering*, vol. 81, no. 5, pp. 637–659, 2010.

[28] F. Chinesta, A. Leygue, M. Beringhier, L. T. Nguyen, J. Grandidier, B. Schrefler, and F. Pesavento, "Towards a framework for non-linear thermal models in shell domains," *International Journal of Numerical Methods for Heat & Fluid Flow*, vol. 23, no. 1, pp. 55–73, 2013.

[29] B. Bognet, F. Bordeu, F. Chinesta, A. Leygue, and A. Poitou, "Advanced simulation of models defined in plate geometries: 3d solutions with 2d computational complexity," *Computer Methods in Applied Mechanics and Engineering*, vol. 201, pp. 1–12, 2012.

[30] B. Bognet, A. Leygue, and F. Chinesta, "On the fully 3d simulations of thermoelastic models defined in plate and shell geometries," *European Journal of Computational Mechanics*, vol. 21, no. 1-2, pp. 40–51, 2012.

[31] E. Giner, B. Bognet, J. J. Ródenas, A. Leygue, F. J. Fuenmayor, and F. Chinesta, "The proper generalized decomposition (PGD) as a numerical procedure to solve 3d cracked plates in linear elastic fracture mechanics," *International Journal of Solids and Structures*, vol. 50, no. 10, pp. 1710 – 1720, 2013.

[32] C. Quesada, D. González, I. Alfaro, E. Cueto, and F. Chinesta, "Computational vademecums for real-time simulation of surgical cutting in haptic environments," *International Journal for Numerical Methods in Engineering*, vol. 108, no. 10, pp. 1230–1247, 2016. nme.5252.

[33] A. Leygue and E. Verron, "A first step towards the use of proper general decomposition method for structural optimization," *Archives of Computational Methods in Engineering*, vol. 17, no. 4, pp. 465–472, 2010.

[34] E. Prulière, F. Chinesta, and A. Ammar, "On the deterministic solution of multidimensional parametric models using the proper generalized decomposition," *Mathematics and Computers in Simulation*, vol. 81, no. 4, pp. 791–810, 2010.

[35] H. Lamari, A. Ammar, P. Cartraud, G. Legrain, F. Chinesta, and F. Jacquemin, "Routes for efficient computational homogenization of nonlinear materials using the proper generalized decompositions," *Archives of Computational methods in Engineering*, vol. 17, no. 4, pp. 373–391, 2010.

[36] C. Ghnatios, F. Chinesta, E. Cueto, A. Leygue, A. Poitou, P. Breitkopf, and P. Villon, "Methodological approach to efficient modeling and optimization of thermal processes taking place in a die: Application to pultrusion," *Composites Part A: Applied Science and Manufacturing*, vol. 42, no. 9, pp. 1169 – 1178, 2011.

[37] L. Chamoin and H. Thai, "Certified real-time shape optimization using isogeometric analysis, pgd model reduction, and a posteriori error estimation," *International Journal for Numerical Methods in Engineering*, vol. 0, no. ja, 2019.

[38] 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*, vol. 213-216, pp. 29 – 41, 2012.

[39] D. González, F. Masson, F. Poulhaon, A. Leygue, E. Cueto, and F. Chinesta, "Proper generalized decomposition based dynamic data driven inverse identification," *Mathematics and Computers in Simulation*, vol. 82, no. 9, pp. 1677 – 1695, 2012.

[40] J. V. Aguado, A. Huerta, F. Chinesta, and E. Cueto, "Real-time monitoring of thermal processes by reduced-order modeling," *International Journal for Numerical Methods in Engineering*, vol. 102, no. 5, pp. 991–1017, 2015.

[41] A. Nouy, "Generalized spectral decomposition method for solving stochastic finite element equations: Invariant subspace problem and dedicated algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 197, no. 51, pp. 4718 – 4736, 2008.

[42] M. Chevreuil and A. Nouy, "Model order reduction based on proper generalized decomposition for the propagation of uncertainties in structural dynamics," *International Journal for Numerical Methods in Engineering*, vol. 89, no. 2, pp. 241–268, 2012.

[43] A. Nouy, M. Chevreuil, and E. Safatly, "Fictitious domain method and separated representations for the solution of boundary value problems on uncertain parameterized domains," *Computer Methods in Applied Mechanics and Engineering*, vol. 200, no. 45, pp. 3066 – 3082, 2011.

[44] F. Chinesta and P. Ladevèze, *Separated Representations and PGD-Based Model Reduction: Fundamentals and Applications*, vol. 1. Springer, 2014.

[45] F. Chinesta and E. Cueto, *PGD-Based Modeling of Materials, Structures and Processes*, vol. 554. Springer, 2014.

[46] F. Chinesta, A. Ammar, A. Leygue, and R. Keunings, "An overview of the proper generalized decomposition with applications in computational rheology," *Journal of Non-Newtonian Fluid Mechanics*, vol. 166, no. 11, pp. 578 – 592, 2011.

[47] F. Chinesta, P. Ladevèze, and E. Cueto, "A short review on model order reduction based on proper generalized decomposition," *Archives of Computational Methods in Engineering*, vol. 18, no. 4, p. 395, 2011.

[48] F. Chinesta, A. Huerta, G. Rozza, and K. Willcox, *Model Reduction Methods*, ch. 1, pp. 1–36. American Cancer Society, 2017.

[49] S. González-Pintor, D. Ginestar, and G. Verdú, "Using proper generalized decomposition to compute the dominant mode of a nuclear reactor," *Mathematical and Computer Modelling*, vol. 57, no. 7, pp. 1807 – 1815, 2013.

[50] J. P. Senecal and W. Ji, "Characterization of the proper generalized decomposition method for fixed-source diffusion problems," *Annals of Nuclear Energy*, vol. 126, pp. 68 – 83, 2019.

[51] A. J. Alberti and T. S. Palmer, "A-priori reduced order modeling for transient neutron diffusion," in *5th International Conference on Transport Theory*, pp. 1–3, 2017.

[52] A. J. Alberti and T. S. Palmer, "Reduced order modeling of non-linear radiation diffusion via proper generalized decomposition," in *Transactions of the American Nuclear Society*, vol. 117, pp. 691–694, 2018.

[53] K. A. Dominesey, J. P. Senecal, and W. Ji, "A reduced-order neutron transport model separated in space and angle," in *Transactions of the American Nuclear Society*, vol. 117, pp. 687–690, 2018.

[54] J. N. Reddy, *An introduction to the finite element method*. McGraw-Hill Mechanical Engineering, 2005.

[55] L. S. Blackford, J. Demmel, J. Dongarra, I. Duff, S. Hammarling, G. Henry, M. Heroux, L. Kaufman, A. Lumsdaine, A. Petitet, R. Pozo, K. Remington, and R. C. Whaley, "An updated set of basic linear algebra subprograms (BLAS)," *ACM Trans. Math. Softw.*, vol. 28, pp. 135–151, June 2002.

[56] "Basic linear algebra subprograms technical forum standard," *The International Journal of High Performance Computing Applications and Supercomputing*, vol. 16, no. 1, 2002.

[57] A. Dumon, C. Allery, and A. Ammar, "Simulation of heat and mass transport in a square lid-driven cavity with proper generalized decomposition (PGD)," *Numerical Heat Transfer, Part B: Fundamentals*, vol. 63, no. 1, pp. 18–43, 2013.

[58] American Nuclear Society, La Grange Park, Ill Computational Benchmark Problems Committee, "Argonne code center: benchmark problem book," Tech. Rep. ANL–7416(Suppl2), Argonne National Laboratory, 1977.

[59] J. J. Duderstadt and L. J. Hamilton, *Nuclear reactor analysis*. Wiley, 1976.

[60] W. Stacey, *Nuclear Reactor Physics*. John Wiley & Sons, 2007.

[61] E. L. Wachspress, *Iterative Solution of Elliptic Systems and Application to the Neutron Diffusion Equations of Reactor Physics*. Prentice-Hall, 1966.

[62] H. Park, D. A. Knoll, and C. K. Newman, "Nonlinear acceleration of transport criticality problems," *Nuclear Science and Engineering*, vol. 172, no. 1, pp. 52–65, 2012.

[63] Y. Saad, "Chebyshev acceleration techniques for solving nonsymmetric eigenvalue problems," *Mathematics of Computation*, vol. 42, no. 166, pp. 567–588, 1984.

[64] T. Rivlin, *Chebyshev Polynomials: From Approximation Theory to Algebra and Number Theory*. Pure and Applied Mathematics: A Wiley Series of Texts, Monographs and Tracts, Wiley, 1990.

[65] I. S. Duff and J. A. Scott, "Computing selected eigenvalues of sparse unsymmetric matrices using subspace iteration," *ACM Trans. Math. Softw.*, vol. 19, pp. 137–159, June 1993.

[66] Y. Saad, "Numerical solution of large nonsymmetric eigenvalue problems," *Computer Physics Communications*, vol. 53, no. 1, pp. 71 – 90, 1989.

[67] J. H. Wilkinson, ed., *The Algebraic Eigenvalue Problem*. New York, NY, USA: Oxford University Press, Inc., 1988.

[68] B. C. Yee, B. Kochunas, E. W. Larsen, and Y. Xu, "Space-dependent wielandt shifts for multigroup diffusion eigenvalue problems," *Nuclear Science and Engineering*, vol. 188, no. 2, pp. 140–159, 2017.

[69] V. Vidal, G. Verdú, D. Ginestar, and J. L. Muñoz Cobo, "Variational acceleration for subspace iteration method. application to nuclear power reactors," *International Journal for Numerical Methods in Engineering*, vol. 41, no. 3, pp. 391–407, 1998.

[70] G. Verdú, R. Miro, D. Ginestar, and V. Vidal, "The implicit restarted arnoldi method, an efficient alternative to solve the neutron diffusion equation," *Annals of Nuclear Energy*, vol. 26, no. 7, pp. 579 – 593, 1999.

[71] J. Munoz-Cobo, R. Miró, A. Wysocki, and A. Soler, "3d calculation of the lambda eigenvalues and eigenmodes of the two-group neutron diffusion equation by coarse-mesh nodal methods," *Progress in Nuclear Energy*, vol. 110, pp. 393 – 409, 2019.

[72] R. B. Morgan, "Generalizations of davidson's method for computing eigenvalues of large nonsymmetric matrices," *Journal of Computational Physics*, vol. 101, no. 2, pp. 287 – 291, 1992.

[73] G. Verdú, D. Ginestar, R. Miró, and V. Vidal, "Using the jacobi-davidson method to obtain the dominant lambda modes of a nuclear power reactor," *Annals of Nuclear Energy*, vol. 32, no. 11, pp. 1274 – 1296, 2005.

[74] C. Subramanian, S. V. Criekingen, V. Heuveline, F. Nataf, and P. Havé, "The davidson method as an alternative to power iterations for criticality calculations," *Annals of Nuclear Energy*, vol. 38, no. 12, pp. 2818 – 2823, 2011.

[75] D. A. Knoll, H. Park, and C. Newman, "Acceleration of k-eigenvalue/criticality calculations using the jacobian-free newton-krylov method," *Nuclear Science and Engineering*, vol. 167, no. 2, pp. 133–140, 2011.

[76] V. Mahadevan and J. Ragusa, "Novel hybrid scheme to compute several dominant eigenmodes for reactor analysis problems," in *Proc. Int. Conf. Physics of Reactors (PHYSOR)*, (Interlaken, Switzerland), p. 8, September 14-19 2008.

[77] D. F. Gill and Y. Y. Azmy, "Newton's method for solving k-eigenvalue problems in neutron diffusion theory," *Nuclear Science and Engineering*, vol. 167, no. 2, pp. 141–153, 2011.

[78] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*. Society for Industrial and Applied Mathematics, 2011.

[79] B. N. Parlett, *The Symmetric Eigenvalue Problem*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1998.

[80] L. A. Semenza, E. E. Lewis, and E. C. Rossow, "The application of the finite element method to the multigroup neutron diffusion equation," *Nuclear Science and Engineering*, vol. 47, no. 3, pp. 302–310, 1972.

[81] Y. Wang, S. Schunert, and V. Laboure, "Rattlesnake theory manual," tech. rep., Idaho National Laboratory, Idaho Falls, ID, 2018.

[82] E. Múller and Z. Weiss, "Benchmarking with the multigroup diffusion high-order response matrix method," *Annals of Nuclear Energy*, vol. 18, no. 9, pp. 535 – 544, 1991.

[83] A. Hébert, *Applied reactor physics*. Presses inter Polytechnique, 2009.

[84] L. Dubcova, P. Solin, G. Hansen, and H. Park, "Comparison of multimesh hp-fem to interpolation and projection methods for spatial coupling of thermal and neutron diffusion calculations," *Journal of Computational Physics*, vol. 230, no. 4, pp. 1182 – 1197, 2011.

[85] F. Rahnema, A. Haghighat, and A. Ougouag, "Consistent multigroup theory enabling accurate course-group simulation of gen iv reactors," tech. rep., U.S. Department of Energy, Nuclear Energy University Program, 11 2013.

[86] Y. Wang, W. Bangerth, and J. Ragusa, "Three-dimensional h-adaptivity for the multigroup neutron diffusion equations," *Progr. Nucl. Energy*, vol. 51, pp. 543–555, 2009.

[87] D. Gaston, C. Newman, G. Hansen, and D. Lebrun-Grandie, "Moose: A parallel computational framework for coupled systems of nonlinear equations," *Nucl. Engrg. Design*, vol. 239, pp. 1768 – 1778, 2009.

[88] G. Alzetta, D. Arndt, W. Bangerth, V. Boddu, B. Brands, D. Davydov, R. Gassmoeller, T. Heister, L. Heltai, K. Kormann, M. Kronbichler, M. Maier, J.-P. Pelteret, B. Turcksin, and D. Wells, "The `deal.II` library, version 9.0," *Journal of Numerical Mathematics*, vol. 26, no. 4, pp. 173–183, 2018.

[89] B. T. Adams and J. E. Morel, "A two-grid acceleration scheme for the multigroup sn equations with neutron upscattering," *Nuclear Science and Engineering*, vol. 115, no. 3, pp. 253–264, 1993.

[90] M. Hanuš, J. C. Ragusa, and M. Hackemack, "A study of various thermal upscattering acceleration schemes for massively parallel transport sweeps," in *Proc. Int. Conf. on Mathematics & Computational Methods Applied to Nuclear Science & Engineering (M&C)*, (Jeju, Korea), April 16-20 2017.

[91] G. I. Bell and S. Glasstone, *Nuclear Reactor Theory*. Krieger, 1970.

[92] D. González, E. Cueto, F. Chinesta, P. Diez, and A. Huerta, "Supg-based stabilization of proper generalized decompositions for high-dimensional advection-diffusion equations," *Int. J. Numer. Methods Eng*, vol. 94, no. 13, pp. 1216–1232, 2013.

[93] A. Ammar, F. Chinesta, and A. Falcó, "On the convergence of a greedy rank-one update algorithm for a class of linear systems," *Archives of Computational Methods in Engineering*, vol. 17, pp. 473–486, Dec 2010.

[94] L. Boucinha, A. Ammar, A. Gravouil, and A. Nouy, "Ideal minimal residual-based proper generalized decomposition for non-symmetric multi-field models – application to transient elastodynamics in space-time domain," *Computer Methods in Applied Mechanics and Engineering*, vol. 273, pp. 56 – 76, 2014.

[95] P.-E. Allier, L. Chamoin, and P. Ladevèze, "Proper generalized decomposition computational methods on a benchmark problem: introducing a new strategy based on constitutive relation error minimization," *Advanced Modeling and Simulation in Engineering Sciences*, vol. 2, p. 17, Jul 2015.

[96] E. W. Larsen, "Diffusion-synthetic acceleration methods for discrete-ordinates problems," *Transport Theory and Statistical Physics*, vol. 13, no. 1-2, pp. 107–126, 1984.

[97] E. N. Aristova and D. F. Baydin, "Implementation of the quasi-diffusion method for calculating the critical parameters of a fast neutron reactor in 3d hexagonal geometry," *Mathematical Models and Computer Simulations*, vol. 5, pp. 145–155, Apr 2013.

[98] J. Willert, H. Park, and W. Taitano, "Using anderson acceleration to accelerate the convergence of neutron transport calculations with anisotropic scattering," *Nuclear Science and Engineering*, vol. 181, no. 3, pp. 342–350, 2015.

[99] J. Willert, H. Park, and D. Knoll, "A comparison of acceleration methods for solving the neutron transport k-eigenvalue problem," *Journal of Computational Physics*, vol. 274, pp. 681 – 694, 2014.

[100] F. Chinesta, E. Cueto, E. Abisset-Chavanne, J. L. Duval, and F. E. Khaldi, "Virtual, digital and hybrid twins: A new paradigm in data-based engineering and engineered data," *Archives of Computational Methods in Engineering*, Nov 2018.

[101] R. Ibáñez, E. Abisset-Chavanne, A. Ammar, D. González, E. Cueto, A. Huerta, J. L. Duval, and F. Chinesta, "A multidimensional data-driven sparse identification technique: the sparse proper generalized decomposition," *Complexity*, vol. 2018, 2018.

[102] C. Robert and G. Casella, *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

[103] C. Paillet, D. Nèron, and P. Ladevèze, "A door to model reduction in high-dimensional parameter space," *Comptes Rendus Mécanique*, vol. 346, no. 7, pp. 524 – 531, 2018.

[104] A. Nouy, "A priori model reduction through proper generalized decomposition for solving time-dependent partial differential equations," *Computer Methods in Applied Mechanics and Engineering*, vol. 199, no. 23, pp. 1603 – 1626, 2010.

[105] H. D. Sterck and K. Miller, "An adaptive algebraic multigrid algorithm for low-rank canonical tensor decomposition," *SIAM Journal on Scientific Computing*, vol. 35, no. 1, pp. B1–B24, 2013.

[106] A. Huerta, E. Nadal, and F. Chinesta, "Proper generalized decomposition solutions within a domain decomposition strategy," *International Journal for Numerical Methods in Engineering*, vol. 113, no. 13, pp. 1972–1994, 2018.

[107] A. Hébert, "The ribon extended self-shielding model," *Nuclear Science and Engineering*, vol. 151, no. 1, pp. 1–24, 2005.

[108] A. T. Till, M. Hanuš, J. Lou, J. E. Morel, and M. L. Adams, "Comparisons of the finite-element-with-discontiguous-support method to continuous energy monte carlo for pin-cell problems," in *Proc. Int. Conf. Physics of Reactors (PHYSOR)*, (Sun Valley, Idaho), pp. 3409–3423, May 1-5 2016.

[109] P. A. Vaquer and R. G. McClarren, "Criticality benchmarks for the finite-element-with-discontiguous-support multigroup method," in *Proc. Int. Conf. on Mathematics & Computational Methods Applied to Nuclear Science & Engineering (M&C)*, (Jeju, Korea), April 16-20 2017.

[110] J. R. Peterson, J. E. Morel, and J. C. Ragusa, "Residual monte carlo for the one-dimensional particle transport equation," *SIAM J. Sci. Comput.*, vol. 38, pp. B941–B961, Nov 2016.

[111] A. Badías, D. González, I. Alfaro, F. Chinesta, and E. Cueto, "Local proper generalized decomposition," *International Journal for Numerical Methods in Engineering*, vol. 112, no. 12, pp. 1715–1732, 2017.

[112] G. Quaranta, E. Abisset-Chavanne, F. Chinesta, and J.-L. Duval, "A cyber physical system approach for composite part: From smart manufacturing to predictive maintenance," *AIP Conference Proceedings*, vol. 1960, no. 1, p. 020025, 2018.

APPENDIX A

PRODUCING A SEPARATED REPRESENTATION USING SINGULAR VALUE DECOMPOSITION

In many of the results throughout this dissertation, PGD is compared to SVD for analysis on the separability of a 2-D solution. This appendix explains explicitly how SVD is used to produce a separated representation of a full-order solution. To begin, it is assumed that the full-order solution is known and has the discretized form described by,

$$u(x,y) \approx \sum_{k=1}^{K} u_k \varphi_k(x,y) \; . \tag{A.1}$$

In all of the applications presented, the domain is discretized with a uniform mesh, that is the solution can be defined as,

$$u(x,y) \approx \sum_{i=1}^{I} \sum_{j=1}^{J} u_{i,j} \varphi_i^x(x) \varphi_j^y(y) \; , \tag{A.2}$$

where I and J are the number of nodes in the x and y dimension, respectively, and $K = I \times J$. Now, the unknown coefficients (u_k) are combined into a vector $\boldsymbol{u}$. In a uniform grid, this vector can be rearranged into a J-by-I matrix:

$$\boldsymbol{A} = \begin{bmatrix} u_{1,1} & u_{2,1} & \dots & u_{I,1} \\ u_{1,2} & u_{2,2} & \dots & u_{I,2} \\ \vdots & \vdots & \ddots & \vdots \\ u_{1,J} & u_{2,J} & \dots & u_{I,J} \end{bmatrix} \; . \tag{A.3}$$

Without loss of generality, the full-order solution on a uniform mesh can be described as,

$$u(x,y) = \sum_{n=1}^{\infty} X_n(x) Y_n(y) \approx \sum_{n=1}^{\infty} \left(\sum_{i=1}^{I} X_{n,i} \varphi_i^x(x) \right) \left(\sum_{j=1}^{J} Y_{n,j} \varphi_j^y(y) \right) \ . \tag{A.4}$$

The coefficients $X_{n,i}$ and $Y_{n,j}$ are combined into the vectors $\boldsymbol{X}_n$ and $\boldsymbol{Y}_n$, respectively. The array $\boldsymbol{A}$ can then be described as,

$$\boldsymbol{A} = \sum_{n=1}^{\infty} \boldsymbol{Y}_n \boldsymbol{X}_n^T \ . \tag{A.5}$$

To relate each term $\boldsymbol{X}_n$ and $\boldsymbol{Y}_n$ to the matrix $\boldsymbol{A}$, SVD is performed on the matrix, which produces,

$$\boldsymbol{A} = \boldsymbol{U} \boldsymbol{\Sigma} \boldsymbol{V}^T \ , \tag{A.6}$$

where $\boldsymbol{U}$ is a J-by-N array, $\boldsymbol{\Sigma}$ is a diagonal N-by-N array, $\boldsymbol{V}$ is a I-by-N array, and $N = \min(I, J)$. The coefficients $X_{n,i}$ and $Y_{n,j}$ can then be related to $\boldsymbol{A}$ by,

$$X_{n,i} = (\boldsymbol{V})_{i,n} \sqrt{(\boldsymbol{\Sigma})_{n,n}} \ , \ n = 1, ..., N \ , \ i = 1, ..., I \ , \tag{A.7a}$$

$$Y_{n,j} = (\boldsymbol{U})_{j,n} \sqrt{(\boldsymbol{\Sigma})_{n,n}} \ , \ n = 1, ..., N \ , \ j = 1, ..., J \ , \tag{A.7b}$$

APPENDIX B

DERIVATION OF FOURIER SERIES REPRESENTATION OF HOMOGENEOUS POISSON PROBLEM

This appendix section shows the derivation of the Fourier series solution to the 2-D homogeneous Poisson problem. The problems is defined as,

$$-\Delta u(x,y) = 1\ , x \in [0,1]\, y \in [0,1] \tag{B.1a}$$

$$u(0,y) = u(1,y) = u(x,0) = u(x,1) = 0 \tag{B.1b}$$

To begin, the source term is expanded into a 2-D Fourier series:

$$1 = \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} F_{mn}\sin(m\pi x)\sin(n\pi y)\ , \tag{B.2a}$$

where,

$$F_{mn} = 4\int_0^1 \sin(m\pi x)dx \int_0^1 \sin(n\pi y)dy = \frac{4}{\pi^2 mn}\left[1-(-1)^m\right]\left[1-(-1)^n\right]\ . \tag{B.2b}$$

With this representation for the source, it is easy to see that the solution must take the form:

$$u(x,y) = \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} U_{mn}\sin(m\pi x)\sin(n\pi y)\ . \tag{B.3}$$

This representation already satisfies the boundary conditions, so to find the coefficients U_{mn} Equations (B.2) and (B.3) are substituted into Equation (B.1a):

$$\sum_{m=1}^{\infty}\sum_{n=1}^{\infty}\left((m\pi)^2+(n\pi)^2\right)U_{mn}\sin(m\pi x)\sin(n\pi y) = \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} F_{mn}\sin(m\pi x)\sin(n\pi y)\ . \tag{B.4}$$

Relating the coefficients:

$$U_{mn} = \frac{4\left[1-(-1)^m\right]\left[1-(-1)^n\right]}{\pi^4 mn\left(m^2+n^2\right)} \,. \tag{B.5}$$

APPENDIX C

COMPARING CONTRIBUTION-BASED AND RESIDUAL-BASED PGD CONVERGENCE

Most of the results throughout this dissertation showed a spurious convergence of enrichment contribution, which was used as the stopping criteria for the enrichment process. Another common criteria in PGD works is to use a residual based approach. Here, these approach are compared to determine if the residual-based approached produces a more uniform convergence. For exposition, the contribution-based error estimator is defined as:

$$\text{Enrichment Magnitude}(n) = \frac{||X_n||\,||Y_n||}{\sum_{i=1}^{n}||X_i||\,||Y_i||}\ , \tag{C.1}$$

and the residual-based error estimator is defined as:

$$\text{Residual Magnitude}(n) = \frac{||\text{Res}_n||}{\sum_{i=1}^{n}||X_i||\,||Y_i||}\ , \tag{C.2a}$$

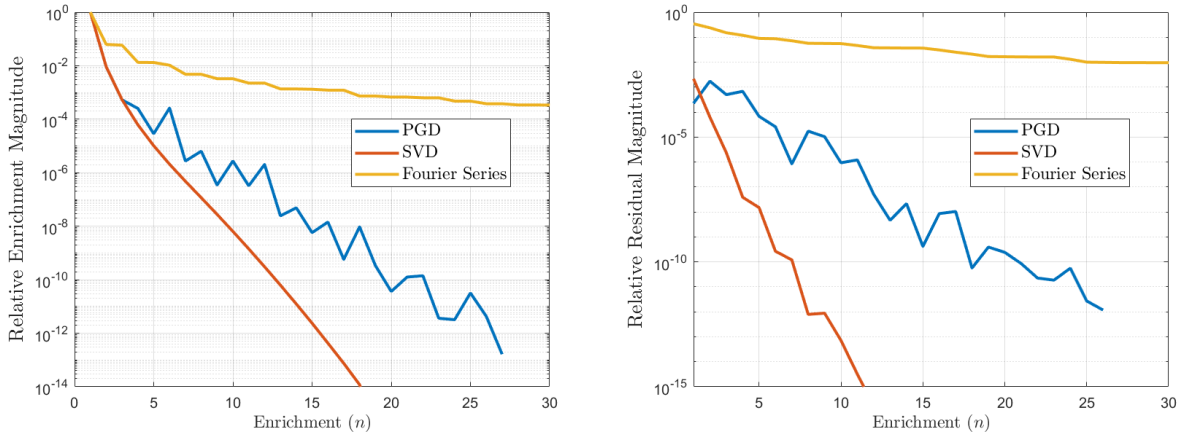
where,

$$\text{Res}_n = q(x,y) - \mathcal{L}(x,y)u_n(x,y) = \sum_{k=1}^{Q} q_k^x(x)q_k^y(y) - \sum_{\ell=1}^{L}\sum_{i=1}^{n}\mathcal{L}_\ell^x X_n \mathcal{L}_\ell^y Y_n\ , \tag{C.2b}$$

and,

$$\begin{aligned}||\text{Res}_n||^2 = \int_{\Omega_x}\int_{\Omega_y}\left[q(x,y) - \mathcal{L}(x,y)u_n(x,y)\right]^2 &= \langle q,q\rangle - 2\langle q,\mathcal{L}u_n\rangle + \langle \mathcal{L}u_n,\mathcal{L}u_n\rangle \\ = \sum_{k=1}^{Q}\sum_{k'=1}^{Q}\langle q_k^x,q_{k'}^x\rangle\langle q_k^y,q_{k'}^y\rangle - 2\sum_{k=1}^{Q}\sum_{\ell=1}^{L}\sum_{i=1}^{n}\langle q_k^x, L_\ell^x X_i\rangle\langle q_k^y, L_\ell^y Y_i\rangle &+ \\ \sum_{\ell=1}^{L}\sum_{i=1}^{n}\sum_{\ell'=1}^{L}\sum_{i'=1}^{n}\langle \mathcal{L}_\ell^x X_i, \mathcal{L}_{\ell'}^x X_{i'}\rangle\langle \mathcal{L}_\ell^y Y_i, \mathcal{L}_{\ell'}^y Y_{i'}\rangle\ .\end{aligned} \tag{C.2c}$$

The convergence of these quantities is studied for the 2-D homogeneous Poisson problem. Figure C.1a shows the contribution-based convergence and Figure C.1b shows the residual-based convergence. These results show that the residual-based convergence is just as spurious as the contribution-based one used throughout the results of the dissertation. This observation indicates that a residual-based enrichment tolerance is most likely a no better error estimator than a the contribution-based one.



(a) Contribution-based convergence

(b) Residual-based convergence

Figure C.1: 2-D Homogeneous Poisson enrichment convergence for PGD, SVD, and Fourier series.

APPENDIX D

PGD DERIVATION FOR SPECIFIC APPLICATIONS

This appendix is meant to show the specific derivations of the PGD algorithm of the governing laws specified in this dissertation. For the sake of exposition and brevity, the materials are homogeneous and the domain is 2-D. The weak form for each dimension's solve is presented for the Nth enrichment. For additional brevity, the one dimensional integrals are represented by:

$$\langle a, b \rangle_r = \int_{\Omega_r} ab \, dr, \tag{D.1}$$

where r is the dimension being integrated over, x, y, or E for instance.

D.1 Multigroup Neutron Diffusion Criticality

This section presents the PGD derivation for space-only decomposition of the multigroup neutron diffusion equation within the power iteration process, described by Equation (3.1). For the Nth enrichment and the $\ell + 1$ power iteration, the solution is assumed to be,

$$\phi^{g,\ell+1}(x, y) = \sum_{i=1}^{N^{\ell+1}} X_i^{g,\ell+1}(x) Y_i^{g,\ell+1}(y) \ , \ g = 1, ..., G \ . \tag{D.2}$$

Solving for $X_i^{g,\ell+1}(x)$:

$$\begin{aligned}
&\left(D^g \left\langle \frac{dX^*}{dx}, \frac{dX_N^{g,\ell+1}}{dx} \right\rangle_x - D^g X^* \frac{dX_N^{g,\ell+1}}{dx}\bigg|_0^{\Omega_x} \right) \left\langle Y_N^{g,\ell+1}, Y_N^{g,\ell+1} \right\rangle_y \\
&\qquad + \left\langle X^*, X_N^{g,\ell+1} \right\rangle_x \left(D^g \left\langle \frac{dY_N^{g,\ell+1}}{dy}, \frac{dY_N^{g,\ell+1}}{dy} \right\rangle_y - D^g Y_N^{g,\ell+1} \frac{dY_N^{g,\ell+1}}{dy}\bigg|_0^{\Omega_y} \right) \\
&+ \Sigma_t^g \left\langle X^*, X_N^{g,\ell+1} \right\rangle_x \left\langle Y_N^{g,\ell+1}, Y_N^{g,\ell+1} \right\rangle_y - \sum_{g'=1}^{G} \Sigma_s^{g \leftarrow g'} \left\langle X^*, X_N^{g',\ell+1} \right\rangle_x \left\langle Y_N^{g,\ell+1}, Y_N^{g',\ell+1} \right\rangle_y
\end{aligned}$$

$$= \frac{1}{k^\ell} \sum_{i=1}^{N^\ell} \chi^g \sum_{g'=1}^{G} \nu\Sigma_f^{g'} \left\langle X^*, X_N^{g',\ell} \right\rangle_x \left\langle Y_N^{g,\ell+1}, Y_N^{g',\ell} \right\rangle_y$$

$$- \sum_{j=1}^{N-1} \Bigg[\left(D^g \left\langle \frac{dX^*}{dx}, \frac{dX_j^{g,\ell+1}}{dx} \right\rangle_x - D^g X^* \frac{dX_j^{g,\ell+1}}{dx} \Bigg|_0^{\Omega_x} \right) \left\langle Y_N^{g,\ell+1}, Y_j^{g,\ell+1} \right\rangle_y$$

$$+ \left\langle X^*, X_j^{g,\ell+1} \right\rangle_x \left(D^g \left\langle \frac{dY_N^{g,\ell+1}}{dy}, \frac{dY_j^{g,\ell+1}}{dy} \right\rangle_y - D^g Y_N^{g,\ell+1} \frac{dY_j^{g,\ell+1}}{dy} \Bigg|_0^{\Omega_y} \right)$$

$$+ \Sigma_t^g \left\langle X^*, X_j^{g,\ell+1} \right\rangle_x \left\langle Y_N^{g,\ell+1}, Y_j^{g,\ell+1} \right\rangle_y$$

$$- \sum_{g'=1}^{G} \Sigma_s^{g \leftarrow g'} \left\langle X^*, X_j^{g',\ell+1} \right\rangle_x \left\langle Y_N^{g,\ell+1}, Y_j^{g',\ell+1} \right\rangle_y \Bigg] \ , \ g = 1, ..., G \ , \quad \text{(D.3a)}$$

Solving for $Y_i^{g,\ell+1}(y)$:

$$\left(D^g \left\langle \frac{dX_N^{g,\ell+1}}{dx}, \frac{dX_N^{g,\ell+1}}{dx} \right\rangle_x - D^g X_N^{g,\ell+1} \frac{dX_N^{g,\ell+1}}{dx} \Bigg|_0^{\Omega_x} \right) \left\langle Y^*, Y_N^{g,\ell+1} \right\rangle_y$$

$$+ \left\langle X_N^{g,\ell+1}, X_N^{g,\ell+1} \right\rangle_x \left(D^g \left\langle \frac{dY^*}{dy}, \frac{dY_N^{g,\ell+1}}{dy} \right\rangle_y - D^g Y^* \frac{dY_N^{g,\ell+1}}{dy} \Bigg|_0^{\Omega_y} \right)$$

$$+ \Sigma_t^g \left\langle X_N^{g,\ell+1}, X_N^{g,\ell+1} \right\rangle_x \left\langle Y^*, Y_N^{g,\ell+1} \right\rangle_y - \sum_{g'=1}^{G} \Sigma_s^{g \leftarrow g'} \left\langle X_N^{g',\ell+1}, X_N^{g',\ell+1} \right\rangle_x \left\langle Y^*, Y_N^{g',\ell+1} \right\rangle_y$$

$$= \frac{1}{k^\ell} \sum_{i=1}^{N^\ell} \chi^g \sum_{g'=1}^{G} \nu\Sigma_f^{g'} \left\langle X_N^{g,\ell+1}, X_N^{g',\ell} \right\rangle_x \left\langle Y^*, Y_N^{g',\ell} \right\rangle_y$$

$$- \sum_{j=1}^{N-1} \Bigg[\left(D^g \left\langle \frac{dX_N^{g,\ell+1}}{dx}, \frac{dX_j^{g,\ell+1}}{dx} \right\rangle_x - D^g X_N^{g,\ell+1} \frac{dX_j^{g,\ell+1}}{dx} \Bigg|_0^{\Omega_x} \right) \left\langle Y^*, Y_j^{g,\ell+1} \right\rangle_y$$

$$+ \left\langle X_N^{g,\ell+1}, X_j^{g,\ell+1} \right\rangle_x \left(D^g \left\langle \frac{dY^*}{dy}, \frac{dY_j^{g,\ell+1}}{dy} \right\rangle_y - D^g Y^* \frac{dY_j^{g,\ell+1}}{dy} \Bigg|_0^{\Omega_y} \right)$$

$$+ \Sigma_t^g \left\langle X_N^{g,\ell+1}, X_j^{g,\ell+1} \right\rangle_x \left\langle Y^*, Y_j^{g,\ell+1} \right\rangle_y$$

$$- \sum_{g'=1}^{G} \Sigma_s^{g \leftarrow g'} \left\langle X_N^{g',\ell+1}, X_j^{g',\ell+1} \right\rangle_x \left\langle Y^*, Y_j^{g',\ell+1} \right\rangle_y \Bigg] \ , \ g = 1, ..., G \ , \quad \text{(D.3b)}$$

D.2 Space-Only Multigroup Neutron Diffusion

This section presents the PGD derivation for space-only decomposition of the multigroup neutron diffusion equation described by Equation (4.5). For the Nth enrichment, the solution is assumed to be,

$$\phi^g(x,y) = \sum_{i=1}^{N} X_i^g(x) Y_i^g(y) \ , \ g = 1, ..., G \ . \tag{D.4}$$

Solving $X_N^g(x)$:

$$\begin{aligned}
&\left(D^g \left\langle \frac{dX^*}{dx}, \frac{dX_N^g}{dx} \right\rangle_x - D^g X^* \frac{dX_N^g}{dx}\bigg|_0^{\Omega_x} \right) \langle Y_N^g, Y_N^g \rangle_y \\
&+ \langle X^*, X_N^g \rangle_x \left(D^g \left\langle \frac{dY_N^g}{dy}, \frac{dY_N^g}{dy} \right\rangle_y - D^g Y_N^g \frac{dY_N^g}{dy}\bigg|_0^{\Omega_y} \right) + \sum_{g'=1}^{G} \Sigma^{gg'} \left\langle X^*, X_N^{g'} \right\rangle_x \left\langle Y_N^g, Y_N^{g'} \right\rangle_y \\
&= \left\langle X^*, \sqrt{Q} \right\rangle_x \left\langle Y_N^g, \sqrt{Q} \right\rangle_y - \sum_{j=1}^{N-1} \Bigg[\left(D^g \left\langle \frac{dX^*}{dx}, \frac{dX_j^g}{dx} \right\rangle_x - D^g X^* \frac{dX_j^g}{dx}\bigg|_0^{\Omega_x} \right) \langle Y_N^g, Y_j^g \rangle_y \\
&\qquad + \langle X^*, X_j^g \rangle_x \left(D^g \left\langle \frac{dY_N^g}{dy}, \frac{dY_j^g}{dy} \right\rangle_y - D^g Y_N^g \frac{dY_j^g}{dy}\bigg|_0^{\Omega_y} \right) \\
&\qquad + \sum_{g'=1}^{G} \Sigma^{gg'} \left\langle X^*, X_j^{g'} \right\rangle_x \left\langle Y_N^g, Y_j^{g'} \right\rangle_y \Bigg] \ , \ g = 1, ..., G \ ,
\end{aligned} \tag{D.5a}$$

Solving $Y_N^g(y)$:

$$\begin{aligned}
&\left(D^g \left\langle \frac{dX_N^g}{dx}, \frac{dX_N^g}{dx} \right\rangle_x - D^g X_N^g \frac{dX_N^g}{dx}\bigg|_0^{\Omega_x} \right) \langle Y^*, Y_N^g \rangle_y \\
&+ \langle X_N^g, X_N^g \rangle_x \left(D^g \left\langle \frac{dY^*}{dy}, \frac{dY_N^g}{dy} \right\rangle_y - D^g Y^* \frac{dY_N^g}{dy}\bigg|_0^{\Omega_y} \right) + \sum_{g'=1}^{G} \Sigma^{gg'} \left\langle X_N^g, X_N^{g'} \right\rangle_x \left\langle Y^*, Y_N^{g'} \right\rangle_y \\
&= \left\langle X_1^g, \sqrt{Q} \right\rangle_x \left\langle Y^*, \sqrt{Q} \right\rangle_y - \sum_{j=1}^{N-1} \Bigg[\left(D^g \left\langle \frac{dX_N^g}{dx}, \frac{dX_j^g}{dx} \right\rangle_x - D^g X_N^g \frac{dX_j^g}{dx}\bigg|_0^{\Omega_x} \right) \langle Y^*, Y_j^g \rangle_y \\
&\qquad + \langle X_N^g, X_j^g \rangle_x \left(D^g \left\langle \frac{dY^*}{dy}, \frac{dY_j^g}{dy} \right\rangle_y - D^g Y^* \frac{dY_j^g}{dy}\bigg|_0^{\Omega_y} \right) \\
&\qquad + \sum_{g'=1}^{G} \Sigma^{gg'} \left\langle X_N^g, X_j^{g'} \right\rangle_x \left\langle Y^*, Y_j^{g'} \right\rangle_y \Bigg] \ , \ g = 1, ..., G \ .
\end{aligned} \tag{D.5b}$$

D.3 Space-Energy Multigroup Neutron Diffusion

This section presents the PGD derivation for space-energy decomposition of Equation (4.5). For the Nth enrichment, the solution is assumed to be,

$$\phi(x,y,E) = \sum_{i=1}^{N} X_i(x) Y_i(y) \mathcal{E}_i(E) \; . \tag{D.6}$$

Solving $X_N(x)$:

$$\begin{aligned}
&\left(\left\langle \frac{dX^*}{dx}, \frac{dX_N}{dx} \right\rangle_x - X^* \frac{dX_N}{dx}\bigg|_0^{\Omega_x} \right) \langle Y_N, Y_N \rangle_y \langle \mathcal{E}_N, \boldsymbol{D}\mathcal{E}_N \rangle_E \\
&\qquad + \langle X^*, X_N \rangle_x \left(\left\langle \frac{dY_N}{dy}, \frac{dY_N}{dy} \right\rangle_y - Y_N \frac{dY_N}{dy}\bigg|_0^{\Omega_y} \right) \langle \mathcal{E}_N, \boldsymbol{D}\mathcal{E}_N \rangle_E \\
&\quad + \langle X^*, X_N \rangle_x \langle Y_N, Y_N \rangle_y \langle \mathcal{E}_N, \boldsymbol{\Sigma}\mathcal{E}_N \rangle_E = \langle X^*, 1 \rangle_x \langle Y_N, 1 \rangle_y \langle \mathcal{E}_N, \boldsymbol{Q}\mathcal{E}_N \rangle_E \\
&\qquad - \sum_{j=1}^{N-1} \Bigg[\left(\left\langle \frac{dX^*}{dx}, \frac{dX_j}{dx} \right\rangle_x - X^* \frac{dX_j}{dx}\bigg|_0^{\Omega_x} \right) \langle Y_N, Y_j^g \rangle_y \langle \mathcal{E}_N, \boldsymbol{D}\mathcal{E}_j \rangle_E \\
&\qquad\quad + \langle X^*, X_j \rangle_x \left(\left\langle \frac{dY_N}{dy}, \frac{dY_j}{dy} \right\rangle_y - Y_N \frac{dY_j}{dy}\bigg|_0^{\Omega_y} \right) \langle \mathcal{E}_N, \boldsymbol{D}\mathcal{E}_j \rangle_E \\
&\qquad\qquad + \langle X^*, X_j \rangle_x \langle Y_N, Y_j \rangle_y \langle \mathcal{E}_N, \boldsymbol{\Sigma}\mathcal{E}_j \rangle_E \Bigg] \; .
\end{aligned} \tag{D.7a}$$

Solving $Y_N(y)$:

$$\begin{aligned}
&\left(\left\langle \frac{dX_N}{dx}, \frac{dX_N}{dx} \right\rangle_x - X_N \frac{dX_N}{dx}\bigg|_0^{\Omega_x} \right) \langle Y^*, Y_N \rangle_y \langle \mathcal{E}_N, \boldsymbol{D}\mathcal{E}_N \rangle_E \\
&\qquad + \langle X_N, X_N \rangle_x \left(\left\langle \frac{dY^*}{dy}, \frac{dY_N}{dy} \right\rangle_y - Y^* \frac{dY_N}{dy}\bigg|_0^{\Omega_y} \right) \langle \mathcal{E}_N, \boldsymbol{D}\mathcal{E}_N \rangle_E \\
&\quad + \langle X_N, X_N \rangle_x \langle Y^*, Y_N \rangle_y \langle \mathcal{E}_N, \boldsymbol{\Sigma}\mathcal{E}_N \rangle_E = \langle X_N, 1 \rangle_x \langle Y^*, 1 \rangle_y \langle \mathcal{E}_N, \boldsymbol{Q}\mathcal{E}_N \rangle_E \\
&\qquad - \sum_{j=1}^{N-1} \Bigg[\left(\left\langle \frac{dX_N}{dx}, \frac{dX_j}{dx} \right\rangle_x - X_N \frac{dX_j}{dx}\bigg|_0^{\Omega_x} \right) \langle Y^*, Y_j \rangle_y \langle \mathcal{E}_N, \boldsymbol{D}\mathcal{E}_j \rangle_E \\
&\qquad\quad \langle X_N, X_j \rangle_x \left(\left\langle \frac{dY^*}{dy}, \frac{dY_j}{dy} \right\rangle_y - Y^* \frac{dY_j}{dy}\bigg|_0^{\Omega_y} \right) \langle \mathcal{E}_N, \boldsymbol{D}\mathcal{E}_j \rangle_E
\end{aligned}$$

$$+ \langle X_N, X_j \rangle_x \langle Y^*, Y_j \rangle_y \langle \mathcal{E}_N, \boldsymbol{\Sigma} \mathcal{E}_j \rangle_E \Bigg] \ . \quad \text{(D.7b)}$$

Solving $\mathcal{E}_N(E)$:

$$\begin{aligned}
&\left(\left\langle \frac{dX_N}{dx}, \frac{dX_N}{dx} \right\rangle_x - X_N \frac{dX_N}{dx}\bigg|_0^{\Omega_x} \right) \langle Y_N, Y_N \rangle_y \langle \mathcal{E}^*, \boldsymbol{D} \mathcal{E}_N \rangle_E \\
&\qquad + \langle X_N, X_N \rangle_x \left(\left\langle \frac{dY_N}{dy}, \frac{dY_N}{dy} \right\rangle_y - Y^* \frac{dY_N}{dy}\bigg|_0^{\Omega_y} \right) \langle \mathcal{E}^*, \boldsymbol{D} \mathcal{E}_N \rangle_E \\
&\quad + \langle X_N, X_N \rangle_x \langle Y_N, Y_N \rangle_y \langle \mathcal{E}^*, \boldsymbol{\Sigma} \mathcal{E}_N \rangle_E = \langle X_N, 1 \rangle_x \langle Y_N, 1 \rangle_y \langle \mathcal{E}^*, \boldsymbol{Q} \mathcal{E}_N \rangle_E \\
&\qquad - \sum_{j=1}^{N-1} \Bigg[\left(\left\langle \frac{dX_N}{dx}, \frac{dX_j}{dx} \right\rangle_x - X_N \frac{dX_j}{dx}\bigg|_0^{\Omega_x} \right) \langle Y_N, Y_j \rangle_y \langle \mathcal{E}^*, \boldsymbol{D} \mathcal{E}_j \rangle_E \\
&\qquad\qquad \langle X_N, X_j \rangle_x \left(\left\langle \frac{dY_N}{dy}, \frac{dY_j}{dy} \right\rangle_y - Y^* \frac{dY_j}{dy}\bigg|_0^{\Omega_y} \right) \langle \mathcal{E}^*, \boldsymbol{D} \mathcal{E}_j \rangle_E \\
&\qquad\qquad + \langle X_N, X_j \rangle_x \langle Y_N, Y_j \rangle_y \langle \mathcal{E}^*, \boldsymbol{\Sigma} \mathcal{E}_j \rangle_E \Bigg] \ . \quad \text{(D.7c)}
\end{aligned}$$

With multigroup discretization of the energy dimension, the integrals in energy are defined as:

$$\langle \mathcal{E}_i, \boldsymbol{D} \mathcal{E}_j \rangle_E = \sum_{g=1}^{G} \mathcal{E}_i^g D^g \mathcal{E}_j^g \ , \quad \text{(D.8a)}$$

$$\langle \mathcal{E}_i, \boldsymbol{\Sigma} \mathcal{E}_j \rangle_E = \sum_{g=1}^{G} \sum_{g'=1}^{G} \mathcal{E}_i^g \Sigma^{gg'} \mathcal{E}_j^{g'} \ , \quad \text{(D.8b)}$$

$$\langle \mathcal{E}_i, \boldsymbol{Q} \mathcal{E}_j \rangle_E = \sum_{g=1}^{G} \mathcal{E}_i^g Q^g \mathcal{E}_j^g \ , \quad \text{(D.8c)}$$

where $\mathcal{E}_i^g$ is defined by:

$$\mathcal{E}_i(E) = \mathcal{E}_i^g \ , \quad E \in [E_g, E_{g-1}] \ . \quad \text{(D.9)}$$

Finally, the basis function $\mathcal{E}^*(E) = 1$.

D.4 Space-Only Neutron Transport

This section presents the PGD derivation for space-only decomposition of the S_N neutron transport equation described by Equation (5.3). For the Nth enrichment of $k+1$ source iteration, the angular flux is assumed to be,

$$\psi^{n,k+1}(x,y) = \sum_{i=1}^{R_{n,k+1}} X_i^{n,k+1}(x) Y_i^{n,k+1}(y) \tag{D.10}$$

Solving for $X_{R_{n,k+1}}^{n,k+1}(x)$:

$$\begin{aligned}
&\sqrt{1-\mu^2}\cos(\varphi)\left\langle X^*, \frac{dX_{R_{n,k+1}}^{n,k+1}}{dx}\right\rangle_x \left\langle Y_{R_{n,k+1}}^{n,k+1}, Y_{R_{n,k+1}}^{n,k+1}\right\rangle_y \\
&\qquad + \sqrt{1-\mu^2}\sin(\varphi)\left\langle X^*, X_{R_{n,k+1}}^{n,k+1}\right\rangle_x \left\langle Y_{R_{n,k+1}}^{n,k+1}, \frac{dY_{R_{n,k+1}}^{n,k+1}}{dy}\right\rangle_y \\
&= \frac{\Sigma_s}{4\pi}\sum_{n'=1}^{N} w_{n'} \sum_{i=1}^{R_{n',k}} \left\langle X^*, X_i^{n',k}\right\rangle_x \left\langle Y_{R_{n,k+1}}^{n,k+1}, Y_i^{n',k}\right\rangle_y + \frac{Q}{4\pi}\left\langle X^*, 1\right\rangle_x \left\langle Y_{R_{n,k+1}}^{n,k+1}, 1\right\rangle_y \\
&\qquad - \sum_{j=1}^{R_{n,k+1}-1}\left[\sqrt{1-\mu^2}\cos(\varphi)\left\langle X^*, \frac{dX_j^{n,k+1}}{dx}\right\rangle_x \left\langle Y_{R_{n,k+1}}^{n,k+1}, Y_j^{n,k+1}\right\rangle_y \right. \\
&\qquad\qquad \left. + \sqrt{1-\mu^2}\sin(\varphi)\left\langle X^*, X_j^{n,k+1}\right\rangle_x \left\langle Y_{R_{n,k+1}}^{n,k+1}, \frac{dY_j^{n,k+1}}{dy}\right\rangle_y\right]
\end{aligned} \tag{D.11a}$$

Solving for $Y_{R_{n,k+1}}^{n,k+1}(y)$:

$$\begin{aligned}
&\sqrt{1-\mu^2}\cos(\varphi)\left\langle X_{R_{n,k+1}}^{n,k+1}, \frac{dX_{R_{n,k+1}}^{n,k+1}}{dx}\right\rangle_x \left\langle Y^*, Y_{R_{n,k+1}}^{n,k+1}\right\rangle_y \\
&\qquad + \sqrt{1-\mu^2}\sin(\varphi)\left\langle X_{R_{n,k+1}}^{n,k+1}, X_{R_{n,k+1}}^{n,k+1}\right\rangle_x \left\langle Y^*, \frac{dY_{R_{n,k+1}}^{n,k+1}}{dy}\right\rangle_y \\
&= \frac{\Sigma_s}{4\pi}\sum_{n'=1}^{N} w_{n'} \sum_{i=1}^{R_{n',k}} \left\langle X_{R_{n,k+1}}^{n,k+1}, X_i^{n',k}\right\rangle_x \left\langle Y^*, Y_i^{n',k}\right\rangle_y + \frac{Q}{4\pi}\left\langle X_{R_{n,k+1}}^{n,k+1}, 1\right\rangle_x \left\langle Y^*, 1\right\rangle_y
\end{aligned}$$

$$- \sum_{j=1}^{R_{n,k+1}-1} \left[\sqrt{1-\mu^2}\cos(\varphi) \left\langle X_{R_{n,k+1}}^{n,k+1}, \frac{dX_j^{n,k+1}}{dx} \right\rangle_x \left\langle Y^*, Y_j^{n,k+1} \right\rangle_y \right.$$
$$\left. + \sqrt{1-\mu^2}\sin(\varphi) \left\langle X_{R_{n,k+1}}^{n,k+1}, X_j^{n,k+1} \right\rangle_x \left\langle Y^*, \frac{dY_j^{n,k+1}}{dy} \right\rangle_y \right] \quad \text{(D.11b)}$$

D.5 Parameterized Neutron Diffusion

This section presents PGD derivation of the parameterized version of the mono-energetic neutron diffusion equation. For the Nth enrichment, the solution is assumed to be:

$$\phi(x, y, D, \Sigma_a, Q) = \sum_{i=1}^{N} X_i(x) Y_i(y) \mathcal{D}_i(D) \mathcal{S}_i(\Sigma_a) \mathcal{Q}_i(Q) \quad \text{(D.12)}$$

Solving for $X_N(x)$:

$$\left(\left\langle \frac{dX^*}{dx}, \frac{dX_N}{dx} \right\rangle_x - X^* \frac{dX_N}{dx}\bigg|_0^{\Omega_x} \right) \alpha_{N,N}^y \alpha_{N,N}^D \alpha_{N,N}^{\Sigma_a} \alpha_{N,N}^Q + \langle X^*, X_N \rangle_x \beta_{N,N}^y \beta_{N,N}^D \beta_{N,N}^{\Sigma_a} \beta_{N,N}^Q$$
$$+ \langle X^*, X_N \rangle_x \gamma_{N,N}^y \gamma_{N,N}^D \gamma_{N,N}^{\Sigma_a} \gamma_{N,N}^Q = \langle X^*, 1 \rangle_x \delta_N^y \delta_N^D \delta_N^{\Sigma_a} \delta_N^Q$$
$$- \sum_{j=1}^{N-1} \left[\left(\left\langle \frac{dX^*}{dx}, \frac{dX_j}{dx} \right\rangle_x - X^* \frac{dX_j}{dx}\bigg|_0^{\Omega_x} \right) \alpha_{N,j}^y \alpha_{N,j}^D \alpha_{N,j}^{\Sigma_a} \alpha_{N,j}^Q + \langle X^*, X_j \rangle_x \beta_{N,j}^y \beta_{N,j}^D \beta_{N,j}^{\Sigma_a} \beta_{N,j}^Q \right.$$
$$\left. + \langle X^*, X_j \rangle_x \gamma_{N,j}^y \gamma_{N,j}^D \gamma_{N,j}^{\Sigma_a} \gamma_{N,j}^Q \right] . \quad \text{(D.13a)}$$

Solving for $Y_N(y)$:

$$\alpha_{N,N}^x \langle Y^*, Y_N \rangle_y \alpha_{N,N}^D \alpha_{N,N}^{\Sigma_a} \alpha_{N,N}^Q + \beta_{N,N}^x \left(\left\langle \frac{dY^*}{dy}, \frac{dY_N}{dy} \right\rangle_y - Y^* \frac{dY_N}{dy}\bigg|_0^{\Omega_y} \right) \beta_{N,N}^D \beta_{N,N}^{\Sigma_a} \beta_{N,N}^Q$$
$$+ \gamma_{N,N}^x \langle Y^*, Y_N \rangle_y \gamma_{N,N}^D \gamma_{N,N}^{\Sigma_a} \gamma_{N,N}^Q = \delta_N^x \langle Y^*, 1 \rangle_y \delta_N^D \delta_N^{\Sigma_a} \delta_N^Q$$
$$- \sum_{j=1}^{N-1} \left[\alpha_{N,j}^x \langle Y^*, Y_j \rangle_y \alpha_{N,j}^D \alpha_{N,j}^{\Sigma_a} \alpha_{N,j}^Q + \beta_{N,j}^x \left(\left\langle \frac{dY^*}{dy}, \frac{dY_j}{dy} \right\rangle_y - Y^* \frac{dY_j}{dy}\bigg|_0^{\Omega_y} \right) \beta_{N,j}^D \beta_{N,j}^{\Sigma_a} \beta_{N,j}^Q \right.$$
$$\left. + \gamma_{N,j}^x \langle Y^*, Y_j \rangle_y \gamma_{N,j}^D \gamma_{N,j}^{\Sigma_a} \gamma_{N,j}^Q \right] . \quad \text{(D.13b)}$$

Solving for $\mathcal{D}_N$:

$$\begin{aligned}\alpha^x_{N,N}\alpha^y_{N,N}\left\langle \mathcal{D}^*, D\mathcal{D}_N\right\rangle_D \alpha^{\Sigma_a}_{N,N}\alpha^Q_{N,N} + \beta^x_{N,N}\beta^y_{N,N}\left\langle \mathcal{D}^*, D\mathcal{D}_N\right\rangle_D \beta^{\Sigma_a}_{N,N}\beta^Q_{N,N} \\ + \gamma^x_{N,N}\gamma^y_{N,N}\left\langle \mathcal{D}^*, \mathcal{D}_N\right\rangle_D \gamma^{\Sigma_a}_{N,N}\gamma^Q_{N,N} = \delta^x_N\delta^y_N\left\langle \mathcal{D}^*, 1\right\rangle_D \delta^{\Sigma_a}_N\delta^Q_N \\ - \sum_{j=1}^{N-1}\Big[\alpha^x_{N,j}\alpha^y_{N,j}\left\langle \mathcal{D}^*, D\mathcal{D}_j\right\rangle_D \alpha^{\Sigma_a}_{N,j}\alpha^Q_{N,j} + \beta^x_{N,j}\beta^y_{N,j}\left\langle \mathcal{D}^*, D\mathcal{D}_j\right\rangle_D \beta^{\Sigma_a}_{N,j}\beta^Q_{N,j} \\ +\gamma^x_{N,j}\gamma^y_{N,j}\left\langle \mathcal{D}^*, \mathcal{D}_j\right\rangle_D \gamma^{\Sigma_a}_{N,j}\gamma^Q_{N,j}\Big] \quad . \end{aligned} \tag{D.13c}$$

Solving fo $\mathcal{S}_N$:

$$\begin{aligned}\alpha^x_{N,N}\alpha^y_{N,N}\alpha^D_{N,N}\left\langle \mathcal{S}^*, \mathcal{S}_N\right\rangle_{\Sigma_a} \alpha^Q_{N,N} + \beta^x_{N,N}\beta^y_{N,N}\beta^D_{N,N}\left\langle \mathcal{S}^*, \mathcal{S}_N\right\rangle_{\Sigma_a} \beta^Q_{N,N} \\ + \gamma^x_{N,N}\gamma^y_{N,N}\gamma^D_{N,N}\left\langle \mathcal{S}^*, \Sigma_a\mathcal{S}_N\right\rangle_{\Sigma_a} \gamma^Q_{N,N} = \delta^x_N\delta^y_N\delta^D_N\left\langle \mathcal{S}^*, 1\right\rangle_{\Sigma_a} \delta^Q_N \\ - \sum_{j=1}^{N-1}\Big[\alpha^x_{N,j}\alpha^y_{N,j}\alpha^D_{N,j}\left\langle \mathcal{S}^*, \mathcal{S}_j\right\rangle_{\Sigma_a} \alpha^Q_{N,j} + \beta^x_{N,j}\beta^y_{N,j}\beta^D_{N,j}\left\langle \mathcal{S}^*, \mathcal{S}_j\right\rangle_{\Sigma_a} \beta^Q_{N,j} \\ +\gamma^x_{N,j}\gamma^y_{N,j}\gamma^D_{N,j}\left\langle \mathcal{S}^*, \Sigma_a\mathcal{S}_j\right\rangle_{\Sigma_a} \gamma^Q_{N,j}\Big] \quad . \end{aligned} \tag{D.13d}$$

Solving for $\mathcal{Q}_N$:

$$\begin{aligned}\alpha^x_{N,N}\alpha^y_{N,N}\alpha^D_{N,N}\alpha^{\Sigma_a}_{N,N}\left\langle \mathcal{Q}^*, \mathcal{Q}_N\right\rangle_Q + \beta^x_{N,N}\beta^y_{N,N}\beta^D_{N,N}\beta^{\Sigma_a}_{N,N}\left\langle \mathcal{Q}^*, \mathcal{Q}_N\right\rangle_Q \\ + \gamma^x_{N,N}\gamma^y_{N,N}\gamma^D_{N,N}\gamma^{\Sigma_a}_{N,N}\left\langle \mathcal{Q}^*, \mathcal{Q}_N\right\rangle_Q = \delta^x_N\delta^y_N\delta^D_N\delta^{\Sigma_a}_N\left\langle \mathcal{Q}^*, Q\right\rangle_Q \\ - \sum_{j=1}^{N-1}\Big[\alpha^x_{N,j}\alpha^y_{N,j}\alpha^D_{N,j}\alpha^{\Sigma_a}_{N,j}\left\langle \mathcal{Q}^*, \mathcal{Q}_j\right\rangle_Q + \beta^x_{N,j}\beta^y_{N,j}\beta^D_{N,j}\beta^{\Sigma_a}_{N,j}\left\langle \mathcal{Q}^*, \mathcal{Q}_j\right\rangle_Q \\ +\gamma^x_{N,j}\gamma^y_{N,j}\gamma^D_{N,j}\gamma^{\Sigma_a}_{N,j}\left\langle \mathcal{Q}^*, \mathcal{Q}_j\right\rangle_Q\Big] \quad . \end{aligned} \tag{D.13e}$$

Where α, β, γ, and δ are coefficients defined as:

$$\alpha^x_{i,j} = \left\langle \frac{dX_i}{dx}, \frac{dX_j}{dx}\right\rangle_x - X_i\frac{dX_j}{dx}\bigg|_0^{\Omega_x} \ , \ \beta^x_{i,j} = \gamma^x_{i,j} = \left\langle X_i, X_j\right\rangle_x \ , \ \delta^x_i = \left\langle X_i, 1\right\rangle_x \tag{D.14a}$$

$$\beta^y_{i,j} = \left\langle \frac{dY_i}{dy}, \frac{dY_j}{dy} \right\rangle_y - Y_i \frac{dY_j}{dy}\bigg|_0^{\Omega_y} \ , \ \alpha^y_{i,j} = \gamma^y_{i,j} = \langle Y_i, Y_j \rangle_y \ , \ \delta^y_i = \langle Y_i, 1 \rangle_y \tag{D.14b}$$

$$\alpha^D_{i,j} = \beta^D_{i,j} = \langle \mathcal{D}_i, D\mathcal{D}_j \rangle_D \ , \ \gamma^D_{i,j} = \langle \mathcal{D}_i, \mathcal{D}_j \rangle_D \ , \ \delta^D_i = \langle \mathcal{D}_i, 1 \rangle_D \ , \tag{D.14c}$$

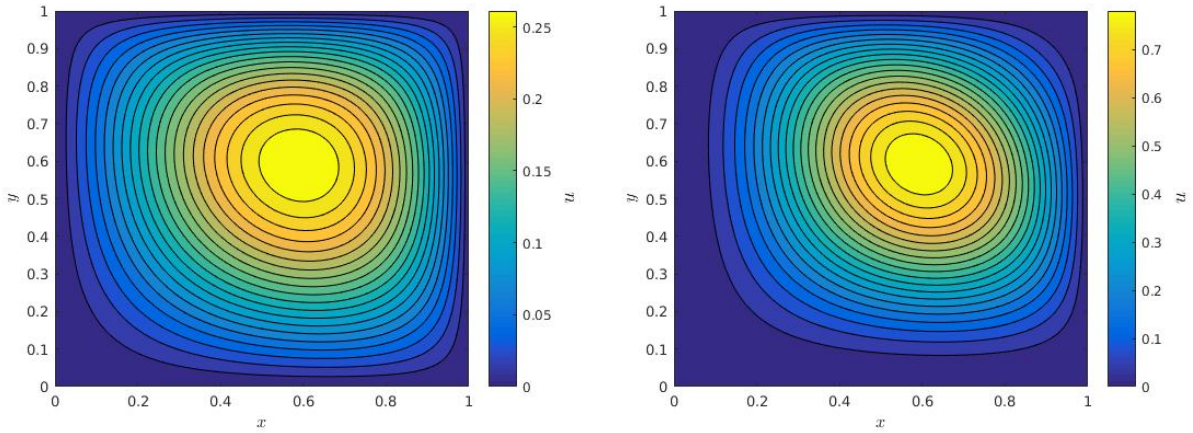
$$\alpha^{\Sigma_a}_{i,j} = \beta^{\Sigma_a}_{i,j} = \langle \mathcal{S}_i, \mathcal{S}_j \rangle_{\Sigma_a} \ , \ \gamma^{\Sigma_a}_{i,j} = \langle \mathcal{S}_i, \Sigma_a \mathcal{S}_j \rangle_{\Sigma_a} \ , \ \delta^{\Sigma_a}_i = \langle \mathcal{S}_i, 1 \rangle_{\Sigma_a} \ , \tag{D.14d}$$

$$\alpha^Q_{i,j} = \beta^Q_{i,j} = \gamma^Q_{i,j} = \langle \mathcal{Q}_i, \mathcal{Q}_j \rangle_Q \ , \ \delta^Q_i = \langle \mathcal{Q}_i, Q \rangle_Q \ . \tag{D.14e}$$

APPENDIX E

SOLUTIONS TO SELECT PROBLEMS

This appendix is meant to show the solutions to select problems presented in the dissertation that were not already disclosed. These results are not meant for analysis or draw any conclusions, they are included for the sake of exposition.



(a) First manufactured solution (Equation (2.22a)) (b) First manufactured solution (Equation (2.22b))

Figure E.1: Visualization of both Poisson manufactured solutions

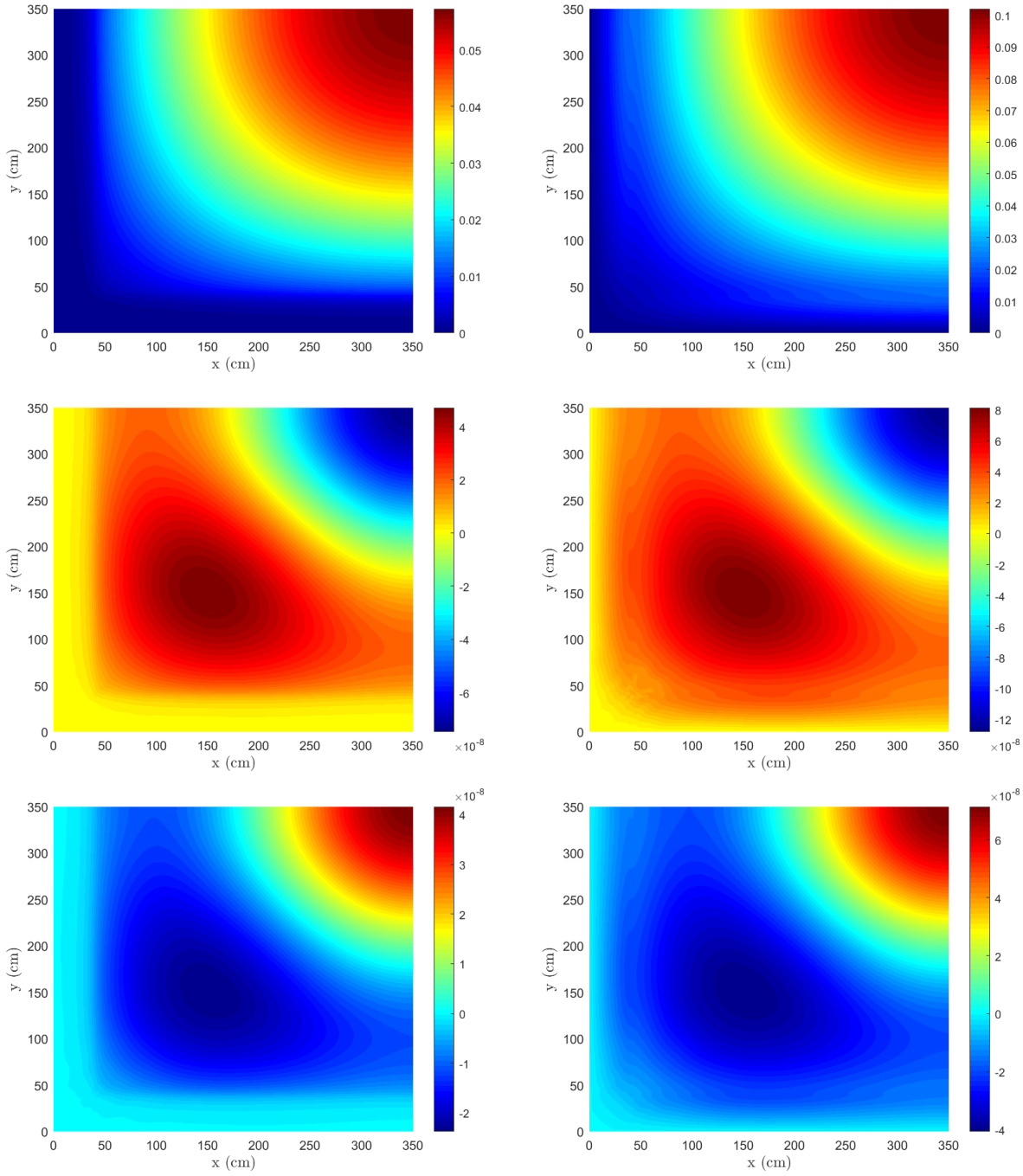


Figure E.2: Full-order (MOOSE) solution and difference from PGD solutions at 170 elements per dimension, two-group two-region eigenvalue problem. Left column: group 1 flux. Right column: group 2 flux. Top row: solution. Middle row: unshifted PGD difference. Bottom Row: shifted PGD difference

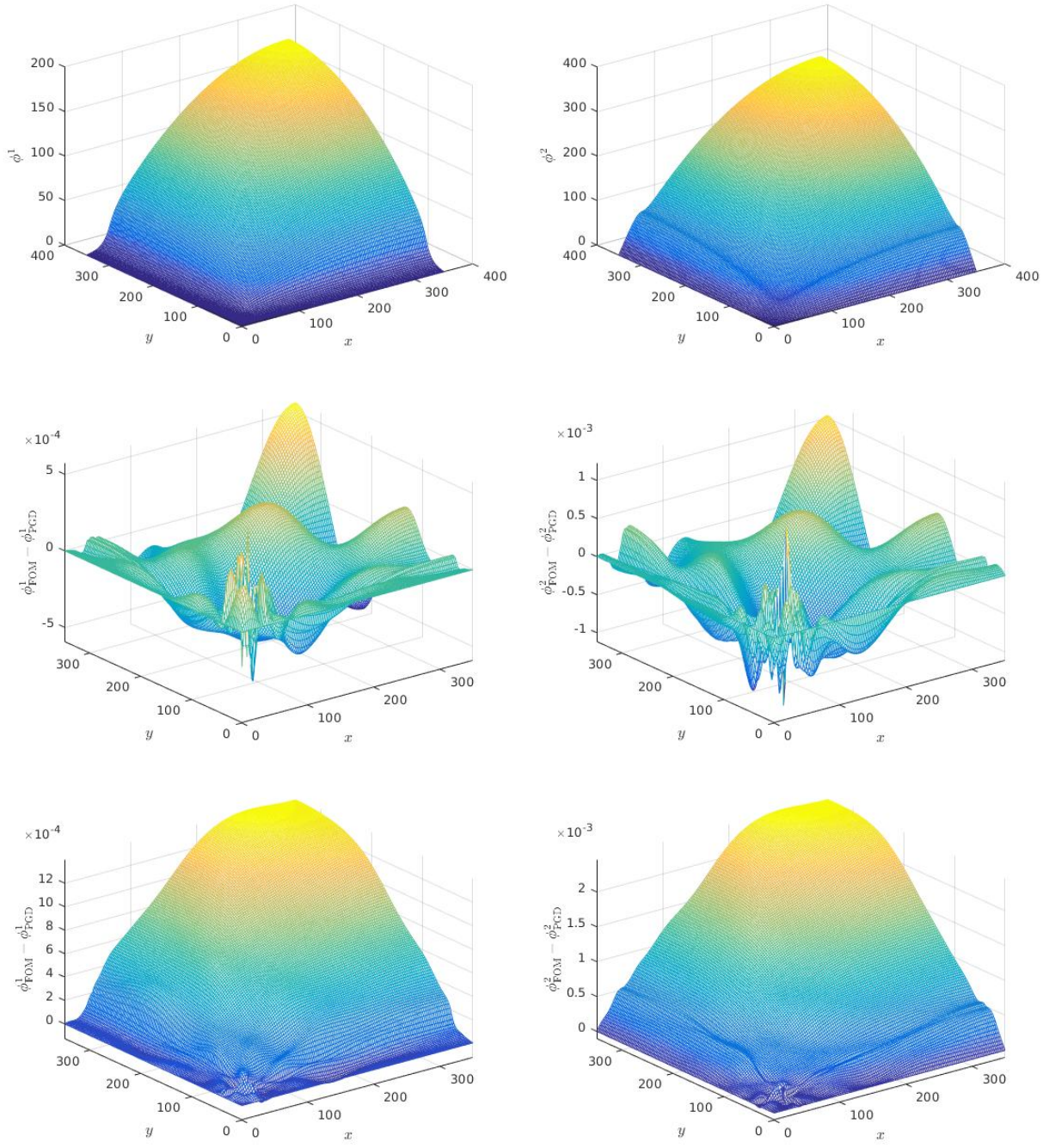


Figure E.3: Full-order (MOOSE) solution and difference from PGD solutions at 170 elements per dimension, two-group two-region fixed-source problem. Left column: group 1 flux. Right column: group 2 flux. Top row: solution. Middle row: PGD-MG difference. Bottom Row: PGD-DE difference

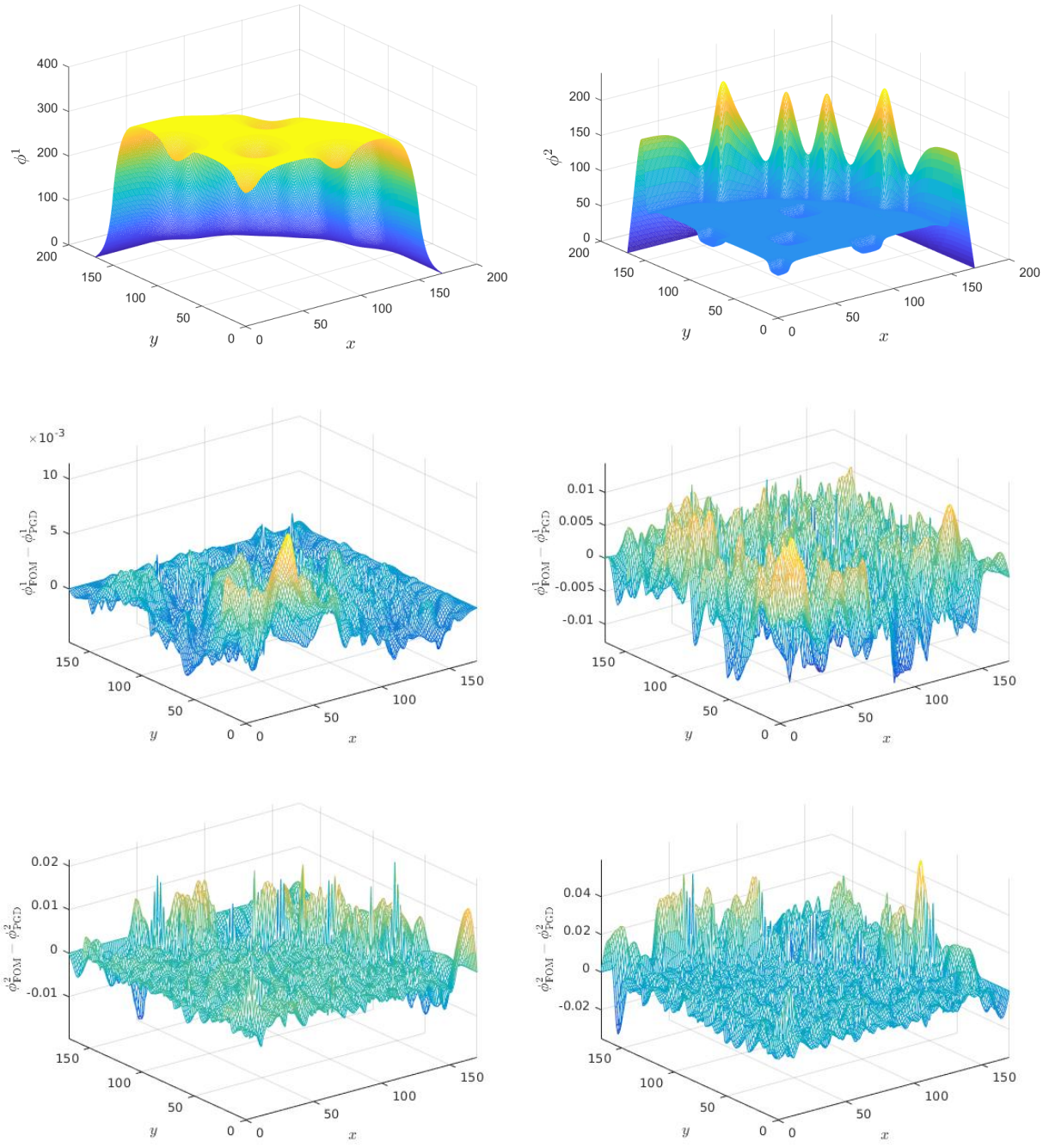
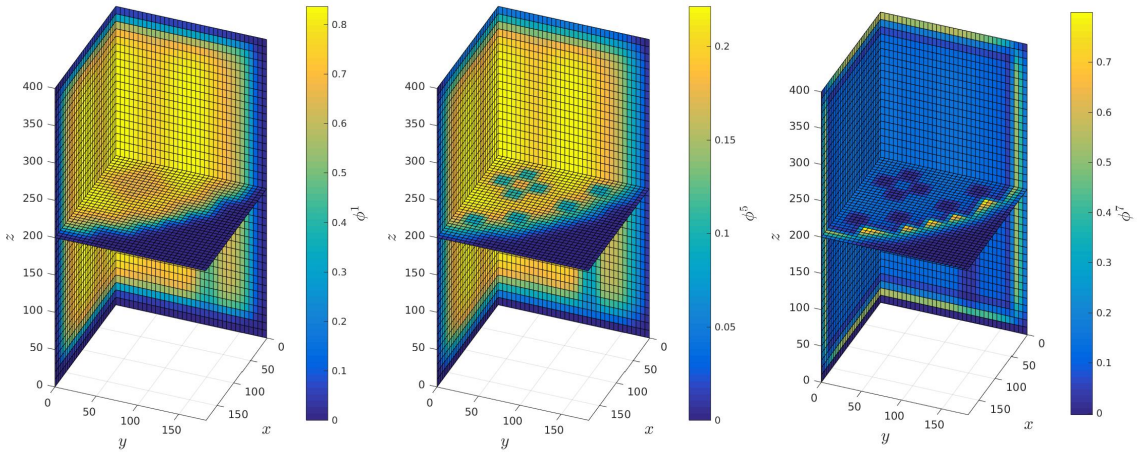
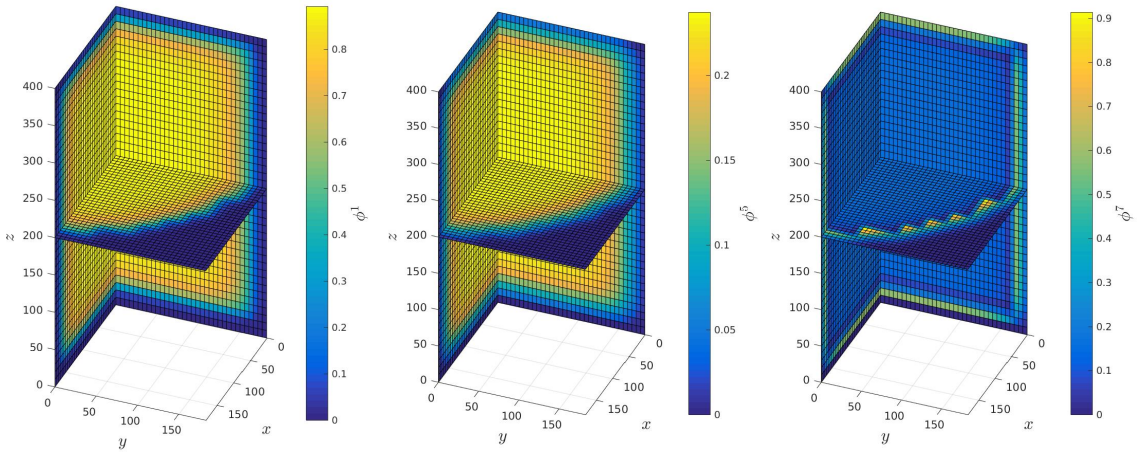


Figure E.4: Full-order (MOOSE) solution and difference from PGD solutions at 136 elements per dimension, 2-D IAEA multigroup problem. Left column: group 1 flux. Right column: group 2 flux. Top row: solution. Middle row: PGD-MG difference. Bottom Row: PGD-DE difference



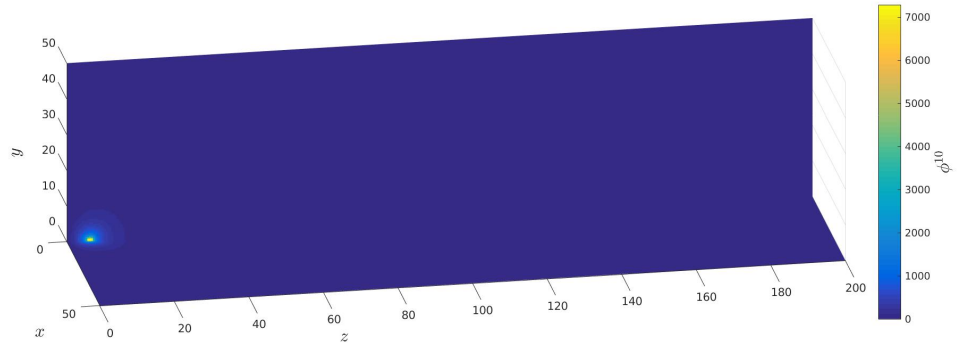
(a) Group 1 (b) Group 5 (c) Group 7

Figure E.5: Visualization of 3-D seven-group fixed-source problem with heterogeneous fuel region

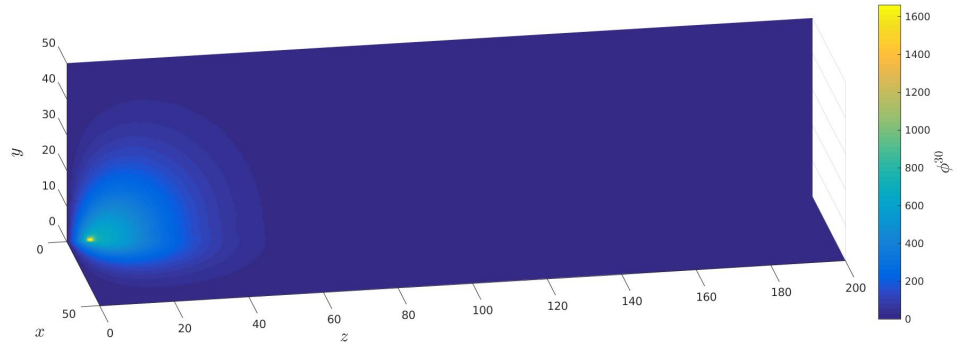


(a) Group 1 (b) Group 5 (c) Group 7

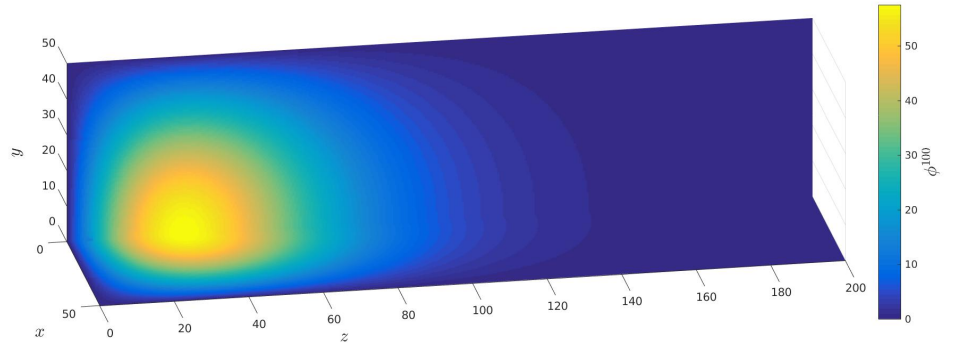
Figure E.6: Visualization of 3-D seven-group fixed-source problem with homogeneous fuel region



(a) Group 10

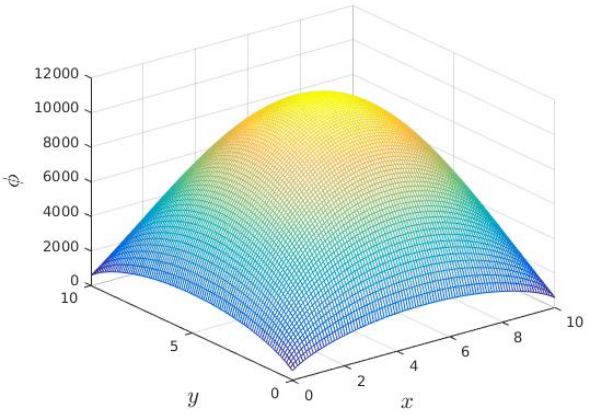


(b) Group 30

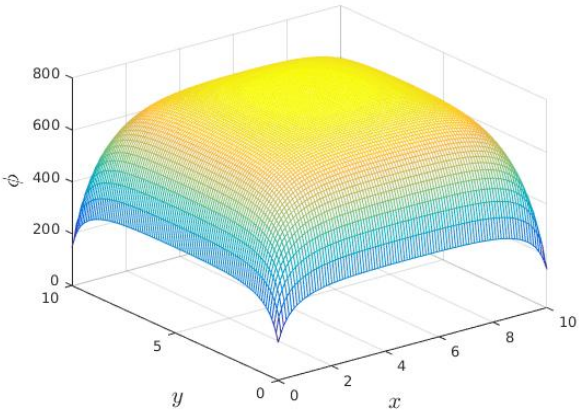


(c) Group 100

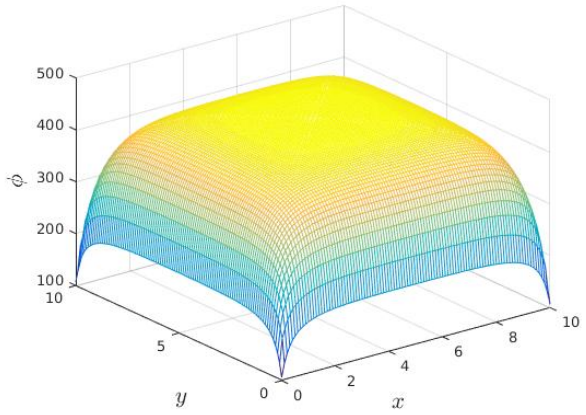
Figure E.7: Visualization of 145-group graphite block flux



(a) S_8, $c = 1$



(b) S_8, $c = 0.5$



(c) S_8, $c = 0.1$

Figure E.8: Visualization of scalar flux for isotropic scattering problem