REDUCED-ORDER MODELING OF NEUTRON TRANSPORT
BY PROPER GENERALIZED DECOMPOSITION

Kurt A. Dominesey

Submitted in Partial Fulfillment of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

Approved by:
Dr. Wei Ji, Chair
Dr. Yaron Danon
Dr. Suvranu De
Dr. Fengyan Li
Dr. Steven Douglass

Department of Mechanical, Aerospace, and Nuclear Engineering
Rensselaer Polytechnic Institute
Troy, New York

[August 2022]
Submitted July 2022
CONTENTS

LIST OF TABLES ................................................. ix
LIST OF FIGURES ............................................. x
ACKNOWLEDGMENT ............................................. xiv
ABSTRACT .................................................. xvi
1. INTRODUCTION ............................................ 1
  1.1 Motivation .............................................. 1
  1.2 Research Objectives .................................... 2
  1.3 Overview .............................................. 4
2. REDUCED-ORDER MODELS IN REACTOR PHYSICS .......... 5
  2.1 Models Original to Reactor Physics .................... 6
    2.1.1 Flux Synthesis ................................... 6
    2.1.2 Nodal Methods .................................... 6
    2.1.3 Response Matrix Methods ......................... 8
    2.1.4 Fission Matrix Methods .......................... 8
  2.2 Projection-Based Reduced-Order Modeling ............ 9
    2.2.1 Proper Orthogonal Decomposition ................. 9
    2.2.2 Dynamic Mode Decomposition .................... 10
    2.2.3 Dynamical Low Rank Approximation ............... 11
    2.2.4 Proper Generalized Decomposition .............. 12
3. PRELIMINARIES OF NEUTRON TRANSPORT ............... 14
  3.1 Physics of Neutron Transport ......................... 14
  3.2 Simplification and Numerical Discretization ....... 17
    3.2.1 Legendre Expansion of the Scattering Kernel .... 17
    3.2.2 Energetic Discretization by Multigroup Approximation 18
    3.2.3 Angular Discretization by Discrete Ordinates ...... 22
    3.2.4 Spatial Discretization by Discontinuous Galerkin Finite Elements . 22
  3.3 Iterative Solution .................................... 24
    3.3.1 Fixed-Source Problems ........................... 25
    3.3.2 Transport Sweeps ................................ 25
    3.3.3 Block Gauss-Seidel Iteration in Energy .......... 26
    3.3.4 Criticality Problems ............................ 27
4. MODEL ORDER REDUCTION IN ENERGY ............................................ 29
  4.1 Introduction .............................................................................. 29
  4.2 Motivation ............................................................................... 30
    4.2.1 Cross Section Condensation .................................................. 31
      4.2.1.1 Spatio-angular Condensation Errors .................................... 32
      4.2.1.2 Validity of the Condensation Spectrum ................................ 33
    4.2.2 Reduced-Order Modeling for Fine-Group Neutron Transport ........ 34
  4.3 Previous Work ........................................................................... 34
  4.4 Methodology ............................................................................ 36
    4.4.1 Separation of Coefficients ................................................... 36
    4.4.2 Separated Representation .................................................... 37
    4.4.3 Spatio-angular Submodel ...................................................... 37
    4.4.4 Energetic Submodel ............................................................. 40
    4.4.5 Galerkin and Minimax Formulations ....................................... 41
    4.4.6 Progressive PGD ................................................................. 42
    4.4.7 Progressive PGD with Update .............................................. 43
    4.4.8 Multigroup Discretization ..................................................... 45
    4.4.9 Energetic Inner Products ..................................................... 46
    4.4.10 Spatio-angular Inner Products and Caching ......................... 47
  4.5 Numerical Results ..................................................................... 48
    4.5.1 Numerical Sundries ............................................................. 48
    4.5.2 Computational Costs ........................................................... 49
    4.5.3 C5G7 UO$_2$ Pin-Cell .......................................................... 50
      4.5.3.1 Method of Manufactured Solutions ..................................... 51
      4.5.3.2 Uniform Fission Source .................................................. 52
      4.5.3.3 Reduced-Order Error Indicators ........................................ 53
    4.5.4 Cathalau UO$_2$ & MOX Pin-Cells ......................................... 56
      4.5.4.1 Cross Section Generation ............................................... 60
    4.5.5 Comparison to Coarse-Group Model ..................................... 68
      4.5.5.1 Exact Coarse-Group Cross Sections .................................. 68
      4.5.5.2 Approximate Coarse-Group Cross Sections ....................... 69
  4.6 Conclusions ............................................................................. 71
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5. MODEL ORDER REDUCTION IN ENERGY FOR EIGENVALUE PROBLEMS</td>
<td>75</td>
</tr>
<tr>
<td>5.1 Introduction</td>
<td>75</td>
</tr>
<tr>
<td>5.2 Motivation</td>
<td>75</td>
</tr>
<tr>
<td>5.3 Previous Work</td>
<td>76</td>
</tr>
<tr>
<td>5.4 Methodology</td>
<td>78</td>
</tr>
<tr>
<td>5.4.1 Separated Representation</td>
<td>78</td>
</tr>
<tr>
<td>5.4.2 Spatio-angular Submodel</td>
<td>79</td>
</tr>
<tr>
<td>5.4.3 Energetic Submodel</td>
<td>80</td>
</tr>
<tr>
<td>5.4.4 Galerkin and Minimax Formulations</td>
<td>81</td>
</tr>
<tr>
<td>5.4.5 Application to Eigenvalue Problems</td>
<td>82</td>
</tr>
<tr>
<td>5.4.6 Recursive LU Factorization</td>
<td>82</td>
</tr>
<tr>
<td>5.5 Numerical Results</td>
<td>84</td>
</tr>
<tr>
<td>5.5.1 Numerical Sundries</td>
<td>85</td>
</tr>
<tr>
<td>5.5.2 Computational Cost</td>
<td>85</td>
</tr>
<tr>
<td>5.5.3 Comparison to Fine-Group Model</td>
<td>86</td>
</tr>
<tr>
<td>5.5.4 Comparison to Coarse-Group Model</td>
<td>90</td>
</tr>
<tr>
<td>5.6 Conclusions</td>
<td>92</td>
</tr>
<tr>
<td>5.6.1 Future Work</td>
<td>92</td>
</tr>
<tr>
<td>6. MODEL ORDER REDUCTION IN AXIAL SPACE AND POLAR ANGLE</td>
<td>94</td>
</tr>
<tr>
<td>6.1 Motivation</td>
<td>94</td>
</tr>
<tr>
<td>6.2 Methodology</td>
<td>95</td>
</tr>
<tr>
<td>6.2.1 Separation of Coefficients</td>
<td>96</td>
</tr>
<tr>
<td>6.2.2 Separation of Axial Space from Polar Angle</td>
<td>98</td>
</tr>
<tr>
<td>6.2.3 Separated Representation</td>
<td>99</td>
</tr>
<tr>
<td>6.2.4 Axial-Polar Decomposition, 2D/1D(\mu)</td>
<td>101</td>
</tr>
<tr>
<td>6.2.4.1 Radial Submodel, 2D</td>
<td>101</td>
</tr>
<tr>
<td>6.2.4.2 Axial-Polar Submodel, 1D(\mu)</td>
<td>103</td>
</tr>
<tr>
<td>6.2.5 Axial Decomposition, 2D(\mu)/1D</td>
<td>105</td>
</tr>
<tr>
<td>6.2.5.1 Radial-Polar Submodel, 2D(\mu)</td>
<td>105</td>
</tr>
<tr>
<td>6.2.5.2 Axial Submodel, 1D</td>
<td>107</td>
</tr>
<tr>
<td>6.2.6 Radial or Axial Energy Dependence</td>
<td>108</td>
</tr>
<tr>
<td>6.2.7 Proper Generalized Decomposition Algorithm</td>
<td>110</td>
</tr>
<tr>
<td>6.2.8 Similarity to Conventional 2D/1D Methods</td>
<td>110</td>
</tr>
<tr>
<td>6.3 Numerical Results</td>
<td>112</td>
</tr>
</tbody>
</table>
6.3.1 Takeda Light Water Reactor ........................................... 112
6.4 Conclusion ........................................................................... 115
6.4.1 Future Work ................................................................. 117

7. NUMERICAL IMPLEMENTATION IN AETHER .......................... 118
7.1 Motivation ............................................................................ 118
7.2 Methodology ........................................................................ 119
  7.2.1 Fixed-Source Problems .................................................. 120
  7.2.2 Criticality Problems ....................................................... 121
  7.2.3 Multigroup Cross Sections ............................................. 122
  7.2.4 Angular Quadratures ..................................................... 122
7.3 Future Work ........................................................................ 123
  7.3.1 Parallelism ...................................................................... 123
  7.3.2 Preconditioning ............................................................. 123
  7.3.3 Anisotropic Scattering ................................................... 124

8. CONCLUSION ........................................................................... 125
8.1 Future Work ......................................................................... 126
  8.1.1 Algorithmic Enhancements ............................................. 126
  8.1.2 Further Validation ........................................................ 127
  8.1.3 Incorporating Prior Information ...................................... 128
  8.1.4 Complementary Techniques ......................................... 129

REFERENCES ............................................................................ 130

APPENDICES ............................................................................ 149

A. COMPUTATIONAL COMPLEXITY ........................................ 149
  A.1 Introduction ..................................................................... 149
  A.2 General Example in Finite-Dimensional Tensor Space ....... 150
    A.2.1 Full-Order Model ....................................................... 150
    A.2.2 Progressive Proper Generalized Decomposition ......... 152
  A.3 Neutron Transport .......................................................... 157
    A.3.1 Full-Order Model ....................................................... 157
      A.3.1.1 Up- and Downscattering .................................... 158
      A.3.1.2 Within-Group Blocks ....................................... 160
    A.3.2 Progressive PGD in Energy ....................................... 162
      A.3.2.1 Fixed-Point Iteration ......................................... 163
      A.3.2.2 Spatio-angular Calculations ............................... 165
      A.3.2.3 Energetic Calculations .................................... 167
I. MODEL ORDER REDUCTION IN ANGLE FOR 1D NEUTRON TRANSPORT 212

I.1 Introduction ......................................................... 212

I.2 Methodology ......................................................... 212

I.2.1 Bidirectional Separation ................................. 213

I.2.2 Spatial Submodel ............................................. 214

I.2.3 Angular Submodel ............................................ 215

I.2.4 Fixed-Point Iteration ..................................... 216

I.3 Numerical Results ................................................. 217

I.3.1 Verification by Method of Manufactured Solutions .......... 217

I.3.2 Verification by Analytical Benchmark ......................... 218

I.3.3 Study Over Scattering Ratio ................................ 219

I.4 Conclusions .......................................................... 221

J. SUPPLEMENTAL FILES ............................................... 223

J.1 Permissions for Chapter 4 and Appendices E, H, and I .......... 223
LIST OF TABLES

4.1 Normalized $L^2$ errors of progressive PGD with and without update, by fuel and group structure, Cathalau pin-cells. ........................................... 61

4.2 Normalized $L^2$ condensation and coarsened PGD errors, Cathalau pin-cell(s), SHEM-361 to CASMO-SH-70. Condensation errors are labeled CP or IP to indicate either a Consistent- or Inconsistent-P transport correction. ............ 72

5.1 Reference $k$-eigenvalues from OpenMC and multigroup full-order models. The error represents the difference from OpenMC. .......................... 89

E.1 CASMO-SH-70 structure as inclusive ranges of SHEM-361 groups. ........... 181

F.1 Cathalau benchmark statistics for pin-cell $k$-eigenvalues. ....................... 194
LIST OF FIGURES

1.1 The six-dimensional phase space of steady-state radiation transport consists of all possible spatial positions (a), angular trajectories (b), and energies (c). 1

1.2 Spatial decomposition of a 3D mesh (a) into 2D/1D meshes (b), as by Proper Generalized Decomposition. 2

1.3 Number of degrees-of-freedom in a full-order model of 3D, multigroup, neutron transport (a) versus a reduced-order, 2D/1D(μ) model by Proper Generalized Decomposition (b). 2

3.1 Coordinate system of energy-dependent radiation transport in 3D Cartesian geometry. 14

3.2 An energy mesh, conventionally termed a group structure in multigroup methods. 20

3.3 A product Gauss-Legendre-Chebyshev angular quadrature (a) is formed by a Gauss-Legendre polar and midpoint azimuthal quadrature (b). 23

4.1 Neutron cross sections of uranium and plutonium, including nuclear resonances. 30

4.2 The traditional domain of ultrafine-group calculations. 31

4.3 An example of cross section condensation (adapted from [1]). 32

4.4 Spatio-angular disparities caused by resonance self-shielding. Dashed lines represent condensed cross sections. Percentages indicate difference relative to reference cross section (in black). 33

4.5 C5G7 UO₂ pin-cell, 5:1 scale. 50

4.6 C5G7 UO₂ pin-cell meshes, 4:1 scale, with zero to three refinements (left to right), each of which subdivides every element into four. 51

4.7 Rate of convergence for full- and reduced-order models, C5G7 UO₂ pin-cell. 52

4.8 Convergence of progressive Proper Generalized Decomposition with and without update, C5G7 UO₂ pin-cell. 56

4.9 Cathalau pin-cell, 5:1 scale. 57

4.10 Fine-group Cathalau cross sections by fuel and group structure. 58

4.11 Convergence of Proper Generalized Decomposition with and without update, by group structure, Cathalau UO₂ pin-cell. 59

4.12 Convergence of Proper Generalized Decomposition with and without update, by group structure, Cathalau MOX pin-cell. 60
4.13 Number of SHEM-361 groups per CASMO-SH-70 group. Highlight denotes resonance range, groups 21 to 27. ................................................................. 64

4.14 Homogeneous and heterogeneous spectra, Cathalau UO$_2$ pin-cell. Stripes denote CASMO-SH-70 groups 21 to 27, from right to left. Note $\xi_1$ is plotted as first computed, prior to subsequent updates. ...................................................... 65

4.15 Homogeneous and heterogeneous spectra, Cathalau MOX pin-cell. Stripes denote CASMO-SH-70 groups 21 to 27, from right to left. Note $\xi_1$ is plotted as first computed, prior to subsequent updates. ...................................................... 65

4.16 Relative errors of total cross section moments (Consistent-P and Inconsistent-P transport cross sections) by modes, Cathalau pin-cells, collapsed from SHEM-361 to CASMO-SH-70 by PGD. Errors for each $M$ are computed prior to subsequent updates. ................................................................. 66

4.17 Relative group-wise condensation errors, Cathalau pin-cells. Cross sections are collapsed from SHEM-361 to CASMO-SH-70 by PGD (colored) or the full-order model (black), each with a Consistent-P (solid) or Inconsistent-P (dotted) correction. ................................................................. 67

4.18 Workflow for comparing flux errors between fine-group PGD ROM and coarse-group full-order model. Despite being depicted as single boxes, each solution consists of two fluxes (angular and scalar). Likewise, the “coarse full-order model” refers to both that with Consistent- and Inconsistent-P transport corrections. ................................................................. 68

4.19 Group-wise condensation (black and gray) and PGD (colored) errors, Cathalau UO$_2$ pin-cell, SHEM-361 to CASMO-SH-70. ................................................................. 69

4.20 Group-wise condensation (black and gray) and PGD (colored) errors, Cathalau MOX pin-cell, SHEM-361 to CASMO-SH-70. ................................................................. 70

4.21 Cathalau UO$_2$/MOX checkerboard and stripes, 4:1 scale. ................................. 70

4.22 Group-wise condensation (plus dissimilarity if dotted, black and gray) and PGD (colored) errors, Cathalau UO$_2$/MOX checkerboard, SHEM-361 to CASMO-SH-70. ................................................................. 71

4.23 Group-wise condensation (plus dissimilarity if dotted, black and gray) and PGD (colored) errors, Cathalau UO$_2$/MOX stripes, SHEM-361 to CASMO-SH-70. ................................. 72

5.1 Convergence of Galerkin and Minimax PGD (with eigenvalue update), by group structure, UO$_2$ pin-cell. ................................................................. 87

5.2 Convergence of Galerkin and Minimax PGD (with eigenvalue update), by group structure, MOX pin-cell. ................................................................. 88

5.3 Error of the $k$-eigenvalues computed by PGD ROMs and full-order models relative to OpenMC reference values. ................................................................. 89
F.3 Normalized fission source error, PGD with eigenvalue update, Cathalau MOX pin-cell. .................................................. 195
F.4 Cathalau benchmark results (black and red) and original results (green). Curves represent Gaussian Kernel Density Estimates (KDEs) and highlights denote standard deviations from the mean. .................................................. 195
H.1 Numerical convergence with grid refinement. .................................. 208
H.2 Numerical convergence with modal enrichment. .................................. 208
H.3 Scalar flux induced by prototypical $^{235}$U fission source. .................. 210
H.4 Estimated convergence with modal enrichment for the slowing down of fission neutrons. .................................................. 211
I.1 Verification of first-order convergence with grid refinement (b) to the rank-one manufactured solution (a). .................................................. 218
I.2 Verification of transmitted (a) and reflected (b) collided fluxes by semi-analytical benchmark. .................................................. 219
I.3 Angular fluxes computed by Proper Generalized Decomposition for scattering ratios $c$ of 0.5, 0.9, and 1.0 (from top to bottom). Stars ($\ast$) denote nonlinear non-convergence. .................................................. 221
I.4 Scalar fluxes computed by Proper Generalized Decomposition for varying scattering ratios $c$. .................................................. 221
Acknowledgment

Firstly, I would like to thank Dr. Wei Ji for his guidance and mentorship throughout my academic career. Without him, none of this could have been accomplished. Likewise, I give my sincere thanks to the rest of my doctoral committee: Dr. Yaron Danon, for introducing me to undergraduate research and convincing me to pursue my PhD; Dr. Suvranu De, for making time in his busy schedule and for being one of the handful of academics already familiar with my thesis topic; Dr. Fengyan Li, for her many insights into the mathematics of radiation transport; and Dr. Steven Douglass, for helping me navigate the ins-and-outs of NNL and for his indispensable advice on reactor physics and wordsmithing alike. I would also like to thank Troy Becker, Michael Hackemack, and Tim Trumbull for showing me the ropes of deterministic neutron transport on HPC architectures and generally making my practicum at NNL as productive, educational, and enjoyable as possible.

Additionally, I would like to thank my fellow graduate students from Dr. Ji’s research group: Jaron Senecal, for introducing me to the arcane subject that would one day eclipse all other worldly matters in my life; Sam Walker, for our many late-night study-sessions and for consistently demonstrating a zest for life I did not previously think possible of a graduate student; Joey McPherson, for his encyclopedic knowledge about semiconductors and many pointed words about the simulation thereof; Matt Eklund, for the half-hour epics about Fortran compilers, Star Wars, or theology that were sure to follow whenever he walked into our office; Peter Kowal, for his many creative ideas about reactor moderation; and Jon Eugenio, for fearlessly leading the rest of us through our first mycological expedition.

I am, of course, forever grateful to my parents, Cathy and Andy, who are retired, and thus I’m sure have better things to do than sift through 200 pages of inscrutable equations; as such, I hereby absolve them of any duty, express or implied, to read anything past this acknowledgment (and maybe the abstract). Since I know no words can dissuade my (soon-to-be) father-in-law, Plinio, I will instead merely thank him and Lana for their support, hospitality, and bottomless cups of coffee. And likewise, no words can express my gratitude to my fiancée, Hannah, whose unwavering love, support, and occasional copyediting were instrumental to the completion of this dissertation.

This research was performed under appointment to the Rickover Fellowship Program in Nuclear Engineering sponsored by the Naval Reactors Division of the National Nuclear
Security Administration.
ABSTRACT

The numerical simulation of neutron transport within a nuclear reactor—for brevity, reactor physics—is the foundation on which analysts: design new reactors; license existing designs; prolong the licenses of operating plants; optimize loading patterns of fuel and burnable absorbers; perform fuel cycle analyses; evaluate a reactor’s ability to self-regulate in response to perturbations; model reactivity control systems; design adequate shielding; and predict the isotopic composition of spent fuel. It is, in short, how humans understand the neutronic inner workings of a nuclear reactor, second only to physical experimentation. It is also, unfortunately, extremely resource-intensive to deterministically compute at high fidelities, being described as a six- or seven-dimensional integro-differential equation. These dimensions refer to the neutron field’s position in space $\vec{r} \equiv (x, y, z)$, direction of travel $\vec{\Omega}$ (in angular coordinates $\mu$ and $\omega$), speed (energy) $E$, and instant in time $t$ (for transients). As such, discretizing the neutron transport equation with even a scant ten unknowns in each dimension would yield a seven-dimensional mesh containing ten million degrees-of-freedom. Unsurprisingly, in practical simulations of large reactors, this number often runs into the billions or trillions, requiring either vast High Performance Computing (HPC) resources or drastic simplifications. This phenomenon is known generally as the “curse of dimensionality” and is common to many fields of numerical analysis.

Presently, we aim to circumvent the curse of dimensionality by seeking a separable, low-rank approximation of the neutron flux, or solution. Moreover, unlike a posteriori, or data-driven, Reduced-Order Models (ROMs), this decomposition will be computed progressively by way of a greedy algorithm, eliminating the need for full-order reference solutions. Specifically, the a priori model order reduction technique here applied to reactor physics is Proper Generalized Decomposition (PGD). Because PGD approximates the solution to high-dimensional problems like radiation transport as a finite series of $M$ products of low(er)-dimensional modes, one avoids solving the high-dimensional (full-order) problem entirely. Instead, only $M$ nonlinear systems of low-dimensional subproblems need be solved; as such, the PGD ROM may be drastically cheaper to compute than the original problem, especially if few modes $M$ are needed. Particularly, we here separate energy (yielding spatio-angular and energetic subproblems) and axial space (yielding 2D and 1D subproblems). Both ROMs are then validated in prototypical reactor physics benchmarks.
Our first application—model order reduction in energy by PGD—is motivated by the extreme (ultrafine) resolution required to resolve individual nuclear resonances—sharp peaks across a narrow range of energy—in the neutron interaction probabilities, or cross sections, of nuclear fuels and other materials. This regime of fidelity is so demanding as to be typically impractical for geometries larger than a 1D or 2D slice across a single fuel pin—let alone the hundreds of pins comprising an assembly (or the hundreds of assemblies comprising a core). For now, however, we consider the more modest energy meshes (70 to 361 groups) usually employed for infinite lattices of pins or assemblies (that is, lattice physics), such that it is tractable to compute the full-order solution for comparison. Benchmark cases are taken to be representative light water reactor (LWR) pins of UO$_2$ or Mixed Oxide fuel with CASMO-70, XMAS-172, and SHEM-361 energy meshes. To begin, we establish that both the Galerkin and Minimax PGD ROMs are able to compute the flux at sufficient precision (0.36% $L^2$ error or less) in a tractable number of modes ($M = 50$). Next, we apply the ROM to cross section generation—often the objective of lattice physics—achieving results comparable to a homogenized, infinite medium model with 1 to 3 modes and comparable to the full-order model by 10 to 20 modes, as assessed by the error of the coarse-group model. Subsequently, we compare the coarse(ned)-group flux against that given by cross section condensation, finding similar $L^2$ errors (0.5%) with 10 modes. Given additional modes, the ROM is able to converge below this threshold. Finally, this ROM is extended to criticality (eigen)problems by means of an original algorithm, which achieves $k$-eigenvalue errors less than $2 \times 10^{-4}$ by $M = 50$. Further, the eigenvalue ROM again compares favorably to the coarse-group model with as few as 10 to 20 modes. Based on these results, we anticipate this PGD ROM may be able to calculate detailed flux distributions and cross sections more economically than the full-order model, at a marginal or negligible detriment to accuracy. Moreover, the ROM presents an alternative means of approximation to cross section condensation, preferable in that it introduces neither a loss of fidelity nor irrecoverable error.

Secondly, we apply PGD to separate the axial and (optionally) polar dimensions of neutron transport. As nuclear reactors (especially LWRs) tend to be tall, but geometrically simple, in the axial, or $z$ direction, we expect this ROM may save substantial effort and rapidly converge to a low-rank approximation. Moreover, we anticipate this approach may compare favorably to the methodologically distinct, but practically analogous 2D/1D methods already practiced in reactor physics. First, we derive two original models: that of axial PGD—which
separates only $z$ and the axial streaming direction $\vartheta \in \{-1, +1\}$—and axial-polar PGD—which separates both $z$ and polar angle $\mu$ from the radial domain. Additionally, we grant that the energy dependence $E$ may be ascribed to either radial or axial modes, or both, bringing the total number of candidate 2D/1D ROMs to six. To assess performance, these PGD ROMs are then applied to two few-group benchmarks characteristic of LWRs. Therein, we find each ROM to be convergent and the axial-polar PGD to be often more economical than the axial PGD. Ultimately, given the popularity of 2D/1D methods in reactor physics, we expect a PGD ROM which achieves a similar effect, but perhaps with superior accuracy, a quicker runtime, and/or broader applicability, would be eminently useful, especially for full-core problems.

Finally, we discuss the neutron transport software developed to implement both the the full-order and PGD models, Aether. More specifically, in order to meaningfully apply these ROMs it was necessary to first establish a basic set of features—namely, unstructured mesh geometry, spatial discretization by finite elements, and hyperbolic transport with matrix-free sweeps. Since no software was available that met these requirements, we here develop an original, C++ library, in turn using the deal.II finite element package. Despite the specialized research objectives above, the software is organized such that the particularities of PGD do not appear in the full-order model, but rather are implemented as wrappers around or modifications of it. This allows the library to serve as a general-purpose research tool for deterministic radiation transport, even outside of applications in PGD. Ultimately, we intend to release Aether as a permissively open-source software library, such that others can use and modify this implementation at will. Moreover, while some outstanding features (preconditioning, parallelism) would be practically required, we envision with a modest effort, Aether could be made a useful application for end-users, not just developers, akin to OpenMC or OpenMOC.
CHAPTER 1
INTRODUCTION

1.1 Motivation

Computational simulation of neutron transport is hampered by inherently high dimensionality, as depicted in Figure 1.1. In spite of the governing equation being well-known and involving few assumptions, calculating the neutron flux over all positions $\vec{r}$, angles $\vec{\Omega}$, and energies $E$ of interest in a nuclear reactor can be computationally intractable, even on a modern computing cluster. These analyses are only made practical by clever simplifications devised by reactor physicists—of special note to this thesis, cross section condensation [1] and 2D/1D methods [2]. However, we here propose a means of approximation largely unexplored with regard to neutron transport: Reduced-Order Models (ROMs) by Proper Generalized Decomposition (PGD) [3]–[5].

![Figure 1.1: The six-dimensional phase space of steady-state radiation transport consists of all possible spatial positions (a), angular trajectories (b), and energies (c).](image)

In effect, PGD allows one to decompose a multidimensional model into multiple, lower-dimensional models. To illustrate, consider a 3D radiation transport problem—specifically, the Takeda Light Water Reactor (LWR) [6] analyzed in Section 6.3.1—with 50 spatial degrees-of-freedom along the $x$, $y$, and $z$ axes; four azimuthal by four polar angles per octant of the unit sphere; and two energy groups. Such a spatial mesh is plotted in Figure 1.2a—each cube representing $10^3$ unknowns—and likewise such an angular quadrature in Figure 3.3a. Evidently, despite this relatively coarse discretization, the total number of degrees-of-freedom is $50^3 \times 4^2 \times 8 \times 2$, or 32 million, as visualized in Figure 1.3a where each cube represents $50^3 = 125,000$ unknowns. This reveals both the “curse of dimensionality” and the advantage...
of PGD: separating the dimensions may yield a dramatically smaller problem. For example, separating in $z$ (as in Figure 1.2b) and axial streaming direction $\vartheta$ reduces this number to $M \times 320,100$; likewise, additionally separating the polar cosine $\mu \equiv \cos(\theta)$ (as in Figure 3.3b) and $E$ yields $M$ times either 80,400 or 40,800 respectively, the former case being depicted in Figure 1.3b for $M = 3$. The primary disadvantage is that the separated models are nonlinearly coupled and some number of modes, $M$, is required. Even so, PGD may prove a powerful tool for high-fidelity radiation transport, especially if few modes $M$ are found to be sufficient.

![Figure 1.2: Spatial decomposition of a 3D mesh (a) into 2D/1D meshes (b), as by Proper Generalized Decomposition.](image)

![Figure 1.3: Number of degrees-of-freedom in a full-order model of 3D, multigroup, neutron transport (a) versus a reduced-order, 2D/1D($\mu$) model by Proper Generalized Decomposition (b).](image)

1.2 Research Objectives

In this dissertation, we seek to characterize two major applications of PGD: model order reduction in energy and the same in axial space and polar angle. This decision is informed by two basic facts of reactor physics. First, the energy mesh must be exceedingly fine—20,000 to 100,000 energy groups [1]—to resolve individual nuclear resonances, which
form sharp peaks in neutron reaction probabilities (cross sections). This can be mitigated by cross-section condensation, a multistage, flux-weighted averaging of this data, but this introduces irrecoverable error and still requires simulations of formidable energy fidelity [7], [8]. Naturally then, we will apply energy-separated PGD to three distinct ends: for general-purpose reactor physics; to generate cross sections; and as an alternative means of approximation, obviating condensation. Moreover, we use PGD in energy as an example to study the advantage of PGD’s “update” step and a Petrov-Galerkin PGD (Minimax PGD [9]), neither of which have been previously applied to reactor physics. Likewise, we also use this opportunity to demonstrate an original algorithm for solving eigenproblems by PGD.

Secondly, nuclear reactors tend to be tall but “axially homogeneous,” in that there are few material interfaces in the axial dimension. This suggests the immense computational expense in moving from 2D to 3D simulations can be mitigated somehow. The prevailing approach is 2D/1D methods—such as those implemented in CRX [10], DeCART (Deterministic Core Analysis Based on Ray Tracing) [11], and MPACT (Michigan Parallel Characteristics Transport) [2]—but these are not always stable or accurate (necessitating several correction schemes) and are perhaps “best suited for problems where the axial gradient of the solution is relatively weak, such as [LWR] problems” [12]. An analogous method by PGD, which separates a 3D problem into 2D and 1D subproblems without incurring the above disadvantages, would then be eminently useful, especially for full-core analyses. To summarize, we state our research objectives as follows, categorized into three workscopes.

1. Achieve model order reduction in energy by PGD
   (a) Apply to general-purpose reactor physics, cross section generation, and as an alternative to the same
   (b) Apply to Galerkin and Minimax PGD, with and without update
   (c) Apply to fixed-source and eigenvalue problems—in the latter case, by way of an original algorithm, Progressive PGD with Eigenvalue Update

2. Achieve model order reduction in axial space and polar angle by PGD
   (a) Assess separation of axial space with and without polar angle
   (b) Assess axial, radial, and shared energy dependence
3. Develop a neutron transport simulator implementing the full- and reduced-order models

(a) Support multigroup, discrete ordinates ($S_N$), and Discontinuous Galerkin Finite Element Method (DGFEM) discretizations in 1D, 2D, and 3D geometry

(b) Support fixed-source and eigenvalue problems

1.3 Overview

To begin, Chapter 2 provides a literature review of reduced-order modeling in reactor physics. This includes not only general, projection-based ROMs—such as PGD—but a select few ad hoc methods invented by reactor physicists. Next, a broad overview of computational neutron transport (as it relates to the present research) is given by Chapter 3. We then delve into the technical contributions, divided into four chapters: Chapter 4 discusses model order reduction in energy; Chapter 5 extends this discussion to $k$-eigenvalue problems; Chapter 6 is devoted to model order reduction in axial space and polar angle; and Chapter 7 details Aether, an original radiation transport simulator developed to implement these models. Finally, concluding remarks and prospects for future work are given in Chapter 8.
CHAPTER 2
REDUCED-ORDER MODELS IN REACTOR PHYSICS

Reduced-order modeling, understood in the broad, literal sense of achieving a model where the computational cost scales at a reduced order, is commonplace in reactor physics. Arguably, diffusion could be considered a low-order model of transport, as could 2D/1D methods for 3D transport, or nodal methods for fine-mesh transport. This ubiquity reflects the fact that the computational demands of reactor physics have historically dwarfed the available computing resources. For example, even as LWRs have been in commercial operation since 1957 (or 1960, excluding the Shippingport reactor) modern simulations of the Watts Bar Unit 1 core in MPACT employ thousands of processors (4088 in 2017 [14]). And, in fact, this is just one instance of the Consortium for Advanced Simulation of LWRs’ (CASL’s) broader efforts to bring supercomputers to bear for LWR analysis [15].

That said, reduced-order modeling often refers specifically to a class of methods which project a high-dimensional model onto a low-dimensional subspace. (Where the distinction is relevant, we refer to the latter as projection-based reduced-order modeling.) In practice, this can be understood as expanding a solution as a summation of basis functions, called modes; finding the approximate solution then requires only computing the coefficients of each mode. Generally, as in Proper Orthogonal Decomposition (POD) [16], [17], these modes are obtained from a decomposition of several full-order reference solutions, termed snapshots. For this reason, these approaches are described as \textit{a posteriori} or data-driven. By contrast, PGD [3], [5], a more recent method which is applied in this dissertation, is \textit{a priori}, requiring no reference solutions.

Given these distinctions, we divide our review into two sections. First, we discuss \textit{ad hoc} methods devised for reactor physics, which can be considered to be, or bear resemblance to, reduced-order models. Notable exclusions include cross section condensation and 2D/1D methods, since these are discussed more thoroughly in Chapters 4 and 6 respectively. Next, we discuss applications of projection-based ROMs, beginning with POD, Dynamic Mode Decomposition (DMD) [18], [19], and Dynamical Low Rank Approximation (DLRA) [20].

---

\footnotesize

[22], to radiation transport and diffusion. Finally, we discuss the same for PGD, including some original contributions by the author (summarized in Appendices H and I) which are superseded or complemented by the research of Chapters 4–6.

2.1 Models Original to Reactor Physics

2.1.1 Flux Synthesis

Flux (or modal) synthesis methods, investigated primarily during the 1960s to 1970s, seek some functional expansion of neutron transport, but generally do not prescribe a means of determining the modes. Exceptions include the temporal syntheses—sometimes described as modal methods—of [23]–[25] and [26] which set the modes to be the successive eigenfunctions of a full-order (α- or k-)eigenvalue problem. More often, modes are determined by engineering judgment, sometimes guided by reference solutions. Once an active research topic, synthesis was applied to each dimension of transport—space [27]–[31], angle [32], [33], energy [31], [34]–[41], and time [42]–[44]—and motivated several review documents [45]–[47].

However, these methods never entered common practice; speaking from 1974, Price and Duderstadt [51] ascribe this primarily to an unexplained phenomenon of “anomalous failures” occasionally reported by several practitioners of flux synthesis. Aggravating matters further, the computational savings—at least, in spectral synthesis—while appreciable, did not meet the expectations of their originators. As such, users were not swayed to abandon the conventional approach (at the time, of finite-difference multigroup diffusion). Today, synthesis methods can largely be considered to be superseded by projection-based ROMs, including POD, DMD, and PGD. Indeed, [23], [24], [52] and [25] explicitly compare flux synthesis in time with POD or DMD respectively, in both cases finding the latter to be more effective.

2.1.2 Nodal Methods

Nodal methods essentially aim to model large spatial domains as individual control volumes, with some intrinsic absorption or production of neutrons, coupled together by in/outflows on the interfaces. Often, this approach is used for full-core analyses where the nodes are taken as (axial portions of) fuel assemblies. Within each node, the cross sections are usually homogenized either by simple flux-weighted volume averages or Generalized

---

Equivalence Theory (GET) [53], which introduces a discontinuity factor at each surface (a correction factor for homogenization). Several nodal methods were later demonstrated to be equivalent to finite elements—a node corresponding to an element—including some based on transverse-integration [54]–[56]. This refers to the integration of governing equations over all but one dimension—for instance, the transverse-integrated equation in $x$ would be the 3D equation integrated in $y$ and $z$—and largely distinguishes modern from historical nodal methods. Further details and context are amply provided in review articles by Lawrence [57] and Sutton and Aviles [58].

This equivalence is also true of variational nodal methods [59], [60], which historically refer to hybrid finite element discretizations [61] of the diffusion and even-parity transport equations [62], ensuring particle conservation. Exploiting the geometric capabilities afforded, later authors would replace the homogeneous elements with a heterogeneous submesh of finite “subelements,” retaining conservation over each coarse element [63]. These would later be employed for full-core, “local-global” schemes [64], [65] where heterogeneous pin-cell nodes are coupled only by low-order spherical harmonic moments (higher-order moments being assumed as reflecting on nodal boundaries). Arguably, by attempting to embed fine-mesh phenomena in—and later, to partially decouple it from—coarse-mesh simulations, these methods are reminiscent of multiscale finite element methods (MsFEMs) [66]–[68]—including the generalized MsFEM [69], [70] which replaces the local problems with POD or DMD ROMs—multiscale mortar FEMs [71], and the response matrix methods of Section 2.1.3. Variational nodal methods were also recently invoked for a 2D/1D method, though to limited success [72].

Far from the relative disuse of flux synthesis, nodal methods are crucial for full-core and transient analyses, where the computational burden could be otherwise overwhelming. Routinely employed in industry and licensing, these methods are implemented in commercial (SIMULATE [73], PANTHER [74], [75]), academic (PARCS [76], NESTLE [77]) and laboratory (DIF3D [78], [79], PROTEUS-NODAL [80], DYN3D [81], [82]) software, in addition to many legacy simulators, perhaps the earliest being FLARE [83] in 1964. In particular, few-group—especially two-group—nodal diffusion is often the method of choice, which underscores the high demand for model order reduction in industrial reactor physics.
2.1.3 Response Matrix Methods

Response Matrix Methods (RMMs) are similar to nodal methods, except that each node is represented by a response matrix, which is tabulated based on high-fidelity reference calculations. As such, few assumptions are made about the physics inside a node, in contrast to nodal methods; instead, the simplifications apply only at nodal interfaces, where the incoming/outgoing flux is usually represented as an expansion of orthogonal polynomials (moments). Because the fixed-source neutron transport equation is linear, the exiting flux can be expanded as a weighted sum of response functions which each represent the exiting flux given a unit incident flux of a particular moment. However, the multiplication factor $k$ (of criticality problems) is nonlinear; therefore the response functions need be computed for several values of $k$ and interpolated when $k$ attains a value not considered in reference calculations. Owing to these pre-calculations, the global problem—coupling each node and finding $k$—can be solved quickly, on the order of an hour or less for a full-core analysis [84], [85]. By this, it is clear RMMs follow an analogous offline/online workload as in data-driven (projection-based) ROMs. That is, expensive reference pre-calculations (solving for response functions or snapshots) lead to a fast runtime later (solving the global problem or data-driven ROM). As noted in Section 4.3, Reed and Roberts [86] recently used POD to determine a basis set by which the expand the energetic dependence of nodal boundary conditions, finding success, defined as a maximum local fission density error of $\leq 0.1\%$, with as as few as 10 to 15 modes given a 238-group energy mesh.

Practically, RMMs are often intended as a modern, accurate replacement to nodal methods, with a small number of global unknowns. Notable implementations include Coarse Mesh Transport (COMET) [84], [85], [87] and Serment [88]. The underlying transport solvers used to compute response functions in each are MCNP [89] and Detran [88] respectively, where the former is Monte Carlo and the latter is deterministic.

2.1.4 Fission Matrix Methods

Fission matrix methods are analogous to RMMs, except that where previously the response was the outgoing flux induced by an incoming flux, here the tabulated response is the production of neutrons in one node ($i$) induced by the emission of a fission neutron in another ($j$). Evidently then, these coefficients can be assembled into an $N \times N$ matrix, where $N$ is the number of nodes. This leads to an eigenvalue problem which can be solved for the
multiplication factor $k$ and the fission rate in each node. Optionally, one can also recover the flux by solving a fixed-source problem (with the converged fission source) or compute higher-order eigenvalues and eigenfunctions. As in RMMs, high-fidelity—in practice, often Monte Carlo—reference calculations are required, though this cost can be mitigated by arguments of geometric similarity or correction factors to account for in-core environments. Notable implementations, building on previous work [90]–[92], include the Real-time Analysis for Particle-transport and In-situ Detection (RAPID) [93], which uses Serpent [94] for reference calculations, and that within MCNP [95], itself a Monte Carlo code [96].

2.2 Projection-Based Reduced-Order Modeling

2.2.1 Proper Orthogonal Decomposition

Beginning with POD ROMs which separate physical dimensions, Buchan et al. [97] applied POD to determine a basis set for the angular expansion of radiative transport, which is found to be more accurate than the conventional, but analogous, expansion by spherical harmonics, as in $P_N$ methods. In the context of transient diffusion, Sartori et al. [23] compares POD (in time) and a modal method devised in earlier work (by Xia et al. [98]), where the basis functions are selected as the successive eigenfunctions of a steady-state, $k$-criticality problem, finding the POD more effective. Later, Lorenzi [52] compares this modal method, an adjoint modal method,\(^2\) the POD, and a unique Adjoint POD (APOD), where the modal methods are found to be inadequate and APOD superior to POD. The former modal method is similar to that put forth by Wols [24], who also implements POD as a comparison, where the modes are the $\alpha$- (or $k$-)eigenfunctions of an accelerator-driven system. These modal methods are examples of the modal/flux synthesis methods of Section 2.1.1. For eigenvalue problems, Buchan et al. [99] presents a POD ROM of neutron diffusion parameterized by material cross-sections, unorthodox in that it recasts the model as time-dependent, converging to the steady-state eigenvalue. A similarly parameterized ROM is presented by German and Ragusa [100], using the more conventional $k$-eigenvalue formulation—where the POD projection is implemented either globally or locally by energy group—and supplemented with a greedy algorithm to add new snapshots. Subsequently, German et al. [101] extend this work to a multi-physical Molten Salt Fast Reactor (with neutronics given by six-group diffusion plus

\(^2\)Presented in Lorenzi et al. [26], where the test functions are adjoint $k$-eigenfunctions.
moving precursor balances and thermal-hydraulics by energy conservation) to account for
Doppler broadening (temperature dependence) of fuel cross-sections. For \( k \)-eigenvalue and
transient neutron transport (by discrete ordinates, \( S_N \)), Sun et al. [102] demonstrates a POD
ROM which reduces the spatial dimension; snapshot matrices—with varied cross-sections
for the steady-state, and timesteps for the transient case—are formed and decomposed
for each ordinate individually, finding acceptable results with as as few as two and three
modes respectively. Meanwhile, Choi et al. [103] apply POD to transient neutron transport,
achieving spatially- and spatio-temporally reduced models. Finally, Peng et al. [104] present a
Reduced Basis (RB) [105] method for a space-angle decomposition of radiation transport; RB
differs from POD in that the snapshots are sought by a greedy algorithm (rather than being
manually selected by the user) but, as the algorithms are otherwise comparable, including
this work here is appropriate. Furthermore, applications of POD pertaining to energy are
given in Section 4.3.

2.2.2 Dynamic Mode Decomposition

The DMD [18], [19] is a recent method which can be used to construct reduced-order
models based only on the action of an operator. The canonical use is time-series data;
repeated time-stepping will yield a series of snapshots, each of which is mapped to the
next by some operation. Then, using the Singular Value Decomposition (SVD) and the
relationship between snapshots, a low-rank approximation for the operator can be achieved.
In the radiation transport community, McClaren [106] uses DMD to calculate the time (\( \alpha \)-)
eigenvalues of neutron transport problems in sub- and supercritical slabs. Nishioka et al.
[107] and Endo et al. [108] also calculate the fundamental \( \alpha \)-eigenpair by DMD, but as
applied to reactor noise signals and neutron count rates measured in Rossi-\( \alpha \) and (in the
former article) pulsed-neutron source (PNS) experiments at the Kyoto University Critical
Assembly (KUCA). Hardy et al. [25] achieves a ROM of transient, three-group neutron
diffusion by DMD, which is found to outperform an \( \alpha \)-eigenfunction expansion—a modal
method, discussed in Sections 2.1.1 and 2.2.1—for a subcritical metal sphere. Similarly,
 transient two-dimensional, two-group neutron diffusion by DMD is attained by Abdo et al.
[109], demonstrated in the Laboratorium für Reaktorregelung und Anlagensicherung (LRA)
benchmark. McClaren and Haut [110] apply DMD to accelerate the classic Richardson
(source) iteration of discrete ordinates transport, later extended [111] to the (nonlinear)
positive source iteration which “zeros-and-rescales” elements with negative fluxes during a transport sweep. Roberts et al. [112] use DMD to accelerate the (also classic) power iteration for an eigenvalue neutron diffusion problem, finding that it does improve the power iteration, but not enough to outperform the Arnoldi method. To simulate a transient and analyze stability in a Molten Salt Fast Reactor, Di Ronco et al. [113] invoke DMD for a multi-physical (neutronic and thermal-hydraulic) model. For uncertainty quantification, Abdo et al. [114] seeks a DMD model of isotopic evolution in TRIGA (Training, Isotopes, Research, General Atomics) reactor fuel—both a single element and 2D assembly—validating the predictive ability in both cases, but noting a failure to adequately characterize uncertainty in the latter. Parameterizing the physical properties defining three case studies, Sharak et al. [115] use DMD to construct a ROM of $S_N$ radiative heat transfer in participating media. Meanwhile, Huhn et al. [116] considers the canonical case of a spatio-temporal ROM constructed by DMD, as applied to nonlinear radiative diffusion, but where the model is simultaneously parameterized by various material properties; to this end, two novel means of obtaining the eigenmodes at a given parameter realization are demonstrated.

Mathematically, the connection of DMD to POD is straightforward, since the SVD of the snapshot matrix\(^3\) yields the POD modes. If the operator is linear, the DMD operator is its projection onto this POD basis. This holds even though DMD never requires the operator to be formed, known, or even finite-dimensional (only its action computed). This relationship is discussed at length by Schmid [18].

### 2.2.3 Dynamical Low Rank Approximation

DLRA affords a means of evolving a low-rank factorization of a solution forward in time (or some other iterative sequence) without recourse to the expensive, full-order model [117]. To this end, several novel (numerical) integrators have recently been developed with the appealing features of rank-adaptivity and robust error bounds [20]–[22]. For neutral radiation transport by a $P_N$ method, Peng et al. [118] investigate a low-rank factorization of the flux in space and angle, evolved in time over several transient problems by DLRA. Subsequently, Peng and McClarren [119] propose a high-order/low-order (HOLO) scheme in which this low-rank $P_N$ model is coupled to a full-rank quasi-diffusion model, thereby

\(^3\)More precisely there are two matrices, those of the past and present snapshots. The SVD is applied to the former.
recovering the global conservation usually lost by DLRA. Similar applications to $S_N$ methods, iteratively solved via transport sweeps, are presented in [120]. An asymptotic-preserving (AP), multiscale (micro-macro) DLRA scheme for the linear transport equation, meanwhile, is proposed by Einekmmer et al. [121]. Further, Ding et al. [122] present a thorough, mathematical error analysis of DLRA in the diffusion limit, as applied to a similar transport equation. An $L^2$ stability analysis of DLRA for kinetic transport problems, encompassing several choices of numerical integrator, is provided by Kusch et al. [123], with special note regarding scattering terms. Meanwhile, Kusch et al. [124] consider a spectral decomposition which is evolved across Power Iterations for neutron diffusion criticality problems in spherical geometry. For radiation therapy, Kusch and Stammer [125] apply DLRA to compute the collided flux scattered from either a line source of neutral radiation or a realistic electron beam; there, the low-rank approximation in space and angle is evolved in energy (rather than time), as modeled by the continuous slowing down approximation.

As in DMD, the latter two examples demonstrate that while the “dynamic” evolution is conventionally across a series of timesteps, the same procedure applies to any sequence of iterates or any variable that can be interpreted as an analogue of time (a “pseudo-time”). Further, although DLRA requires some initial condition of the factorization, this may be known analytically (as in a transient simulation) or guessed arbitrarily (as in an iterative solver); in this sense, DLRA can be considered an a priori model. That said, DLRA differs from PGD—and, for that matter, POD and DMD as well—in that the former evolves a low-rank decomposition while the latter constructs a low-rank decomposition (typically by rank-one corrections in PGD). As such, the two do not appear immediately interchangeable, and, in fact, could be considered complementary, in that PGD could be used to compute the initial factorization supplied to DLRA.

### 2.2.4 Proper Generalized Decomposition

Despite the utility of these data-driven ROMs, it is foreseeable that the required reference solutions may be inconvenient or intractable to compute. As such, we break with most ROM literature to pursue a priori models—requiring no prior solutions—through PGD, a recent innovation first published in 2006 by Ammar et al. [3]. Since then, it has seen wide application to a variety of equations, especially in computational mechanics, and several review articles [4], [126], [127] and books [5], [128]–[130] have been published on the topic.
For neutron transport and diffusion, this method was first applied by González-Pintor et al. [131] in 2013, to separate the $x$ and $y$ dimensions of one-group (energy-independent) neutron diffusion for a critical reactor problem. Later, Alberti and Palmer [132] extended this exploration to multigroup (energy-dependent) diffusion and separation in time for transient problems, culminating in Alberti’s thesis [133]. Meanwhile, Senecal and Ji [134] elaborated on the work of González-Pintor and Alberti with respect to three-dimensional ($x$, $y$, and $z$) separation and non-intrusive implementation leveraging Idaho National Laboratory’s Multiphysics Object-Oriented Simulation Environment (MOOSE) [135], further discussed in Senecal’s thesis [136]. Still in the vein of spatially separated neutron diffusion, Prince and Ragusa [137] revisited the $k$-eigenvalue problem, now multigroup, applying PGD as an inner solver (compared to the outer fission iterations). Additionally, Prince and Ragusa [138] decomposed one-group material parameters for uncertainty quantification.

The first application towards (general) Boltzmann transport was presented by Chinesta et al. [139], considering pure advection, separated in space and velocity. Dominesey and Ji [140] extended this research to consider collision (reaction) and scattering terms specific to neutron transport, achieving space-angle separation, as summarized in Appendix I. Also related to neutron transport—discretized by discrete ordinates, $S_N$—Prince and Ragusa [141] decomposed the $x$ and $y$ dependencies of the flux along each ordinate. For separation in neutron energy, Dominesey and Ji [142] demonstrated a scheme using an analytical (isotropic) scattering kernel, coupled with one-dimensional neutron diffusion, reviewed in Appendix H. Prince and Ragusa [143] also pursued a similar method, but with multigroup energy and simultaneous decomposition of neutron diffusion in two or three spatial dimensions. For the remainder of this dissertation, we will build upon this literature by demonstrating separation in energy for neutron transport in Chapters 4 and 5 and several 2D/1D methods separating axial space—and, optionally, polar angle and/or energy—in Chapter 6.
CHAPTER 3
PRELIMINARIES OF NEUTRON TRANSPORT

At present, we briefly review the physics of neutron transport and mathematical methods for its numerical solution. Basic knowledge of this topic is essential to understand the challenges this dissertation seeks to address and the workings of the full- and reduced-order models employed therein. Further implementation details, meanwhile, are provided in Chapter 7 and Appendix C.

3.1 Physics of Neutron Transport

We begin by defining the primary object of our study, the neutron flux $\psi$, possessing units of neutrons per unit area per unit time—conventionally, cm$^{-2}$s$^{-1}$. Like any physical particle, the neutron flux must be located at a particular position in space $\vec{r} \equiv (x, y, z)$ at some instant in time $t$ and travel in a given direction $\vec{\Omega}$—defined by angular coordinates $\mu$ and $\omega$—with some speed $v$, corresponding to energy $E$. These variables—space, angle, energy, and time—comprise the seven-dimensional “phase space” of neutron transport, as depicted in Figure 3.1 (excluding $t$), and give rise to the aforementioned curse of dimensionality.

The basic phenomena of neutron transport, meanwhile, are fairly simple: neutrons may stream from one position to another, be removed by collision, scatter from one direction and

![Coordinate system of energy-dependent radiation transport in 3D Cartesian geometry.](image)

Figure 3.1: Coordinate system of energy-dependent radiation transport in 3D Cartesian geometry.

---

energy to another, or be born out of an exogenous source or fission [144]. Meanwhile, the imbalance between these mechanisms of gain or loss must equal the rate of change of the neutron flux, by definition. Accounting for each of these processes (starting with the change in time, then following their order of introduction) yields the neutron transport equation:

\[
\frac{1}{v(E)} \frac{\partial}{\partial t} \psi(\vec{r}, \vec{\Omega}, E, t) + \vec{\Omega} \cdot \nabla \psi(\vec{r}, \vec{\Omega}, E, t) + \Sigma_t(r, E) \psi(\vec{r}, \vec{\Omega}, E, t) - \int_{E_G}^{E_0} \int_{4\pi} \Sigma_s(\vec{r}, \vec{\Omega}' \rightarrow \vec{\Omega}, E' \rightarrow E) \psi(\vec{r}, \vec{\Omega}', E', t) d\Omega' dE' = q(\vec{r}, \vec{\Omega}, E, t)
\]

(3.1)

where \(\Sigma_t, \Sigma_s,\) and \(\nu \Sigma_f\) are the macroscopic cross sections of collision, scattering, and (neutron production by) fission respectively. Further, \(\chi\) is the energy distribution of fission neutrons, normalized to unity, and \(q\) is an external source. Since this dissertation is not concerned with transient problems, we can immediately simplify by setting the rate of change equal to zero and removing the dependence on time \(t\),

\[
\vec{\Omega} \cdot \nabla \psi(\vec{r}, \vec{\Omega}, E) + \Sigma_t(r, E) \psi(\vec{r}, \vec{\Omega}, E) - \int_{E_G}^{E_0} \int_{4\pi} \Sigma_s(\vec{r}, \vec{\Omega}' \rightarrow \vec{\Omega}, E' \rightarrow E) \psi(\vec{r}, \vec{\Omega}', E', t) d\Omega' dE' = q(\vec{r}, \vec{\Omega}, E, t)
\]

(3.2)

As indicated by the limits of integration, the energy domain\(^4\) is the interval \([E_G, E_0]\), while that in angle is the surface of the unit sphere, denoted \(4\pi\). Likewise, the spatial domain is volume \(\mathcal{V}\), bounded by surface \(\partial \mathcal{V}\), upon which there is some incident flux described by boundary condition \(\psi_{\text{in}}\),

\[
\psi(\vec{r}, \vec{\Omega}, E) = \psi_{\text{in}}(\vec{r}, \vec{\Omega}, E) \quad \text{on} \quad \partial \mathcal{V}^- \equiv \left\{ \left(\vec{r}, \vec{\Omega} \right) : \vec{r} \in \partial \mathcal{V} \land \vec{\Omega} \cdot \vec{n}(\vec{r}) < 0 \right\},
\]

(3.3)

where \(\partial \mathcal{V}^-\) is the inflow boundary and \(\vec{n}\) is the surface normal. For the purposes of this dissertation, this boundary condition will either a vacuum (no incident flux), or specular

---

\(^4\)More precisely, the physical limits would be \([0, \infty)\), but restriction to a finite range is required for numerical applications.
reflection (symmetry),

\[
\psi_{in}(\vec{r}, \vec{\Omega}, E) \equiv \begin{cases} 
0 & \text{on } \partial \mathcal{V}_0^-, \\
\psi(\vec{r}, \vec{\Omega}_s, E) & \text{on } \partial \mathcal{V}_{\text{refl}}^-.
\end{cases}
\]  

(3.4)

on any particular patch of the boundary \( \partial \mathcal{V}^- = \partial \mathcal{V}_0^- \cup \partial \mathcal{V}_{\text{refl}}^- \) where

\[
\vec{\Omega}_s \equiv \vec{\Omega} - 2(\vec{n} \cdot \vec{\Omega})\vec{n}
\]  

(3.5)

is the reflected angle. For brevity, Equation 3.2 is often written in operator notation

\[
(L - S)\psi = q + XF\psi
\]  

(3.6)

where

\[
L\psi \equiv \left( \vec{\Omega} \cdot \nabla + \Sigma_t(\vec{r}, E) \right) \psi(\vec{r}, \vec{\Omega}, E),
\]  

(3.7)

\[
S\psi \equiv \int_{E_0}^{E_G} \int_{4\pi} \Sigma_s(\vec{r}, E' \rightarrow E, \vec{\Omega}' \rightarrow \vec{\Omega}) \psi(\vec{r}, \vec{\Omega}', E') d\Omega' dE',
\]  

(3.8)

\[
XF\psi \equiv \frac{\chi(\vec{r}, E)}{4\pi} \int_{E_0}^{E_G} \nu \Sigma_f(\vec{r}, E') \int_{4\pi} \psi(\vec{r}, \vec{\Omega}', E') d\Omega' dE'.
\]  

(3.9)

Upon discretization, these linear operators become matrices, allowing the numerical computation of (some approximation of) the flux \( \psi \). These matrices, in turn, form the basic components of the Aether software library, discussed in Chapter 7. In what follows, we present the conventional approximation of the scattering kernel, followed by the ubiquitous multigroup, discrete ordinates, and DGFEM discretizations in energy, angle, and space respectively.
3.2 Simplification and Numerical Discretization

3.2.1 Legendre Expansion of the Scattering Kernel

Typically, the double-differential scattering cross section $\Sigma_s$ is approximated as an expansion of Legendre polynomials $P_\ell$ truncated at order $L$,

$$\int_{E_G}^{E_0} \int_{4\pi} \Sigma_s(\vec{r}, E' \to E, \vec{\Omega}' \to \vec{\Omega}) \psi(\vec{r}, \vec{\Omega}', E') d\Omega' dE' \approx \sum_{\ell=0}^{L} \frac{2\ell + 1}{4\pi} \int_{E_G}^{E_0} \int_{4\pi} \Sigma_{s,\ell}(\vec{r}, E' \to E) P_\ell(\vec{\Omega} \cdot \vec{\Omega}') \psi(\vec{r}, \vec{\Omega}', E') d\Omega' dE'$$

(3.10)

where $\Sigma_{s,\ell}$ is the $\ell^{th}$-order scattering moment

$$\Sigma_{s,\ell}(\vec{r}, E' \to E) \equiv \int_{-1}^{1} \Sigma_s(\vec{r}, E' \to E, \mu_0) P_\ell(\mu_0) d\mu_0$$

(3.11)

and $\mu_0 \equiv \vec{\Omega}' \cdot \vec{\Omega}$ is the scattering angle [145]. By the addition theory for spherical harmonics

$$P_\ell(\vec{\Omega} \cdot \vec{\Omega}') = \sum_{m=-\ell}^{\ell} Y_{\ell,m}(\vec{\Omega}) Y_{\ell,m}(\vec{\Omega}') (3.12)$$

the scattering kernel can be expanded further as

$$\sum_{\ell=0}^{L} \frac{2\ell + 1}{4\pi} \int_{E_G}^{E_0} \Sigma_{s,\ell}(\vec{r}, E' \to E) \sum_{m=-\ell}^{\ell} Y_{\ell,m}(\vec{\Omega}) \int_{4\pi} Y_{\ell,m}(\vec{\Omega}') \psi(\vec{r}, \vec{\Omega}', E') d\Omega' dE'. \quad (3.13)$$

Or, letting $\phi_{\ell,m}$ represent the spherical harmonic moments of the flux,

$$\phi_{\ell,m}(\vec{r}, E') \equiv \int_{4\pi} Y_{\ell,m}(\vec{\Omega}') \psi(\vec{r}, \vec{\Omega}', E') d\Omega', \quad (3.14)$$

the kernel can be written more simply as

$$\sum_{\ell=0}^{L} \frac{2\ell + 1}{4\pi} \int_{E_G}^{E_0} \Sigma_{s,\ell}(\vec{r}, E' \to E) \sum_{m=-\ell}^{\ell} Y_{\ell,m}(\vec{\Omega}) \phi_{\ell,m}(\vec{r}, E') dE'. \quad (3.15)$$

Often, the source $q$ is likewise expanded by spherical harmonic moments. This is convenient when the source arises only from fission (emission being generally isotropic) and inscattering (containing only $L$ terms). However, in the ROMs of later chapters, modes are coupled in such a way that the source cannot usually be represented exactly by such an expansion of
size \( L \). As such, we here make no such expansion of source \( q \).

### 3.2.2 Energetic Discretization by Multigroup Approximation

Evidently, both the scattering and fission terms of Equation 3.2 contain Fredholm (or Volterra, in the former case, if there is no upscattering) integral operators in energy. Almost invariably, these are discretized by the multigroup approximation \[144\], which divides the energy scale into \( G \) groups with each group \( g \) spanning the interval \([E_g, E_{g-1}]\). Doing so, the integral operator simplifies to a summation

\[
\int_{E_g}^{E_0} \psi(\vec{r}, \vec{\Omega}, E)dE \approx \sum_{g=1}^{G} \psi_g(\vec{r}, \vec{\Omega})dE.
\]  

(3.16)
as in the analogous case of collocation methods. However, where in the latter it is sufficient to define the solution only at discrete collocation (quadrature) points, the same—that is, setting \( \psi_g \) to be the flux evaluated at the midpoint of group \( g \), for instance—is not suitable in energy, as the flux may vary substantially (by several orders-of-magnitude) across even a single energy group. As such, it is preferable to define \( \psi_g \) as the group-integrated flux

\[
\psi_g(\vec{r}, \vec{\Omega}) \equiv \int_{E_g}^{E_{g-1}} \psi(\vec{r}, \vec{\Omega}, E)dE
\]

(3.17)

which explains why Equation 3.16 does not contain a coefficient for the group width \( \Delta_g \) or some other similar quadrature weight.

Meanwhile, the group cross sections, \( \Sigma_{x,g} \) for reaction \( x \), are not defined in the same manner, as doing so,

\[
\Sigma_{x,g}(\vec{r}) \leftarrow \int_{E_g}^{E_{g-1}} \Sigma_x(\vec{r}, E)dE,
\]

(3.18)

would grossly misrepresent the group-integrated reaction rate \( \Sigma_{x,g}\psi_g \),

\[
\Sigma_{x,g}(\vec{r})\psi_g(\vec{r}, \vec{\Omega}) \neq \int_{E_g}^{E_{g-1}} \Sigma_x(\vec{r}, E)\psi(\vec{r}, \vec{\Omega}, E)dE.
\]

(3.19)

Clearly, Equation 3.19 would be satisfied if \( \Sigma_{x,g} \) were replaced with

\[
\tilde{\Sigma}_{x,g}(\vec{r}, \vec{\Omega}) \equiv \frac{\int_{E_g}^{E_{g-1}} \Sigma_x(\vec{r}, E)\psi(\vec{r}, \vec{\Omega}, E)dE}{\int_{E_g}^{E_{g-1}} \psi(\vec{r}, \vec{\Omega}, E)dE},
\]

(3.20)
but this introduces two difficulties. First, the flux $\psi$ is usually unavailable at the time of discretization (that is, when $\Sigma_{x,g}$ is computed), as it is the objective of the simulation. Second, this would render the cross section angularly-dependent in the same manner as the flux, while it is generally preferred $\Sigma_{x,g}$ be either isotropic or expressed as a Legendre expansion. For that matter, is usually also preferred that the spatial dependence of the cross section be piecewise-constant, as in

$$\Sigma_x(\vec{r}, E) = \Sigma_{x,j}(E), \quad \forall \vec{r} \in V_j,$$

within some non-overlapping regions $V_j$ that comprise the domain $\mathcal{V}$.

Given this, the usual compromise is to set

$$\Sigma_{x,j,g}(\vec{r}) \leftarrow \frac{\int_{E_g}^{E_g-1} \Sigma_{x,j}(E) \xi_j(E) dE}{\int_{E_g}^{E_g-1} \xi_j(E) dE},$$

where $\xi_j$ is the isotropic spectrum in volume $V_j$

$$\xi_j(E) \equiv \int_{V_j} \int_{4\pi} \tilde{\psi}(\vec{r}, \vec{\Omega}, E) d\Omega d\vec{r}$$

given some reference solution $\tilde{\psi} \approx \psi$ such that

$$\Sigma_{x,j,g} \int_{V_j} \int_{4\pi} \psi_g(\vec{r}, \vec{\Omega}) d\Omega dE \approx \int_{E_g}^{E_g-1} \int_{V_j} \int_{4\pi} \Sigma_{x,j}(E) \psi(\vec{r}, \vec{\Omega}, E) d\Omega d\vec{r} dE.$$

While this describes the procedure of cross section condensation (or collapsing) as it is commonly practiced, that is not to say it is always a good approximation—especially when comparing angular fluxes $\psi$ rather than scalar fluxes $\phi$. Indeed, recent studies suggest neglecting the angular dependence of multigroup cross sections incurs appreciable errors, even given an exact reference solution $\tilde{\psi} = \psi$. As it relates to our purposes, the ramifications of the present assumptions are discussed further in Section 4.2.1.

To implement the full-order model, and likewise to compute group-integrated fluxes and reaction rates, the preceding terminology suffices; however, in the PGD ROMs that follow, inner products of the flux must be computed, necessitating a differential value of the flux—that is, the flux per unit energy. This is obtained trivially by introducing the group-average flux $\bar{\psi}_g$

$$\bar{\psi}_g \equiv \psi_g/\Delta_g$$
where $\Delta_g \equiv E_{g-1} - E_g$. That said, the energy scales involved in neutron slowing down are inherently logarithmic, as the physics are that of a particle colliding many times and losing a fraction of its energy each time. As such, it is often convenient to introduce the lethargy $u \equiv \ln(E_0/E)$ [144]. Likewise, a more appropriate inner product for the PGD ROMs could be defined in terms of lethargy $u$ rather than energy $E$. This has no effect on the group-fluxes, and simply amounts to redefining the group-width $\Delta_g \leftarrow \ln(E_{g-1}/E_g)$, such that $\bar{\psi}_g$ becomes the average flux per unit lethargy, instead of unit energy. This change of units (or rather, deletion of units, lethargy being dimensionless), is applied to each of the PGD ROMs in Chapters 4 through 6.

Given these preliminaries, Equation 3.6 can be “semi-discretized” in energy, leaving angle and space for the following sections. In particular, the multigroup operators can naturally be expressed in terms of block rows $g$ and columns $g'$. Specifically, the loss (streaming plus collision) operator $L$ is block diagonal, $L \equiv \text{diag}(L_1, \ldots, L_g, \ldots, L_G)$. Meanwhile, the scattering operator may be full, but is often roughly lower triangular—as in, $S_{g' \to g} \approx 0$ for $g' > g$—as there tends to be little upscattering beyond the thermal energy range. In any case, each block $S_{g' \to g}$ can be subdivided into three,

$$S_{g' \to g} \psi_{g'} \equiv M_g \Sigma_{s,g' \to g} D_g \psi_{g'} = M_g \Sigma_{s,g' \to g} \phi_{g'},$$

specifically, the moment-to-direction (M), scattering cross section ($\Sigma_s$), and direction-to-
moment (D) operators defined as [146]

\[
D_g \psi_g' \equiv \left[ \int_{4\pi} Y_{\ell,m}(\tilde{\Omega}') \psi_g(r, \tilde{\Omega}') d\Omega' \right]_{\ell = 0}^{\ell} = \phi_g',
\]

(3.28)

\[
\Sigma_{s,g' \rightarrow g} \phi_g' \equiv \left[ \Sigma_{s,\ell,g' \rightarrow g} (\tilde{r}) [\phi_g', \ell, m(\tilde{r})]_{m = -\ell}^{\ell} \right]_{\ell = 0}^{L} ,
\]

(3.29)

\[
M_g \phi_g \equiv \sum_{\ell=0}^{L} \frac{2\ell + 1}{4\pi} \sum_{m=-\ell}^{\ell} Y_{\ell,m}(\tilde{\Omega}) \phi_{g,\ell,m}(\tilde{r}) \approx \psi_g ,
\]

(3.30)

in which subscript \( G \) denotes operators that apply to a single group but are equivalent for every \( g \). Equivalently, we can define the multigroup forms of these operators as \( S = M \Sigma D \), where \( M \equiv \text{diag}(M_G, \ldots, M_G) \), \( D \equiv \text{diag}(D_G, \ldots, D_G) \), and

\[
\Sigma_s \equiv \begin{bmatrix}
\Sigma_{s,1 \rightarrow 1} & \ldots & \Sigma_{s,g' \rightarrow 1} & \ldots & \Sigma_{s,G \rightarrow 1} \\
\vdots & \ddots & \vdots \\
\Sigma_{s,1 \rightarrow G} & \ldots & \Sigma_{s,g' \rightarrow G} & \ldots & \Sigma_{s,G \rightarrow G}
\end{bmatrix}.
\]

(3.31)

The fission operator is also full, but usually rank-deficient

\[
XF \equiv \begin{bmatrix}
X_1F_1 & \ldots & X_1F_{g'} & \ldots & X_1F_G \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
X_gF_1 & X_gF_{g'} & X_gF_G \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
X_GF_1 & X_GF_{g'} & \ldots & X_GF_G
\end{bmatrix} = X_g \begin{bmatrix}
F_1 \cdots F_{g'} \cdots F_G
\end{bmatrix}
\]

(3.32)

and can likewise be subdivided as

\[
X_gF_{g'} \equiv D_{g',0} \chi_g \nu \Sigma_{f,g} M_{G,0}.
\]

(3.33)

Or, in multigroup form, \( XF = D_0 \chi \nu \Sigma_f M_0 \), where \( \chi \equiv \begin{bmatrix} \chi_1 \cdots \chi_g \cdots \chi_G \end{bmatrix}^T \) and \( \nu \Sigma_f \equiv \begin{bmatrix} \nu \Sigma_{f,1} \cdots \nu \Sigma_{f,g} \cdots \nu \Sigma_{f,G} \end{bmatrix} \).
3.2.3 Angular Discretization by Discrete Ordinates

As was the case in energy, both scattering and fission are modeled by Fredholm integral operators in angle. While these could be discretized by a functional expansion—typically a series of spherical harmonics, as in the $P_N$ method or the scattering kernel of Section 3.2.1—we here adopt the discrete ordinates or $S_N$ method [144], [147]. This amounts to collocation, in that the integral is approximated by a numerical quadrature, with the caveat that the quadrature points are here ordinates, pointing in some direction on the unit sphere. In short, given ordinates $\{\widehat{\Omega}_n\}_{n=1}^N$ and weights $\{w_n\}_{n=1}^N$, and letting the flux along $\widehat{\Omega}_n$ be written $\psi_n(\vec{r}, E) \equiv \psi(\vec{r}, \widehat{\Omega}_n, E)$,

$$\int_{4\pi} \psi(\vec{r}, \widehat{\Omega}, E) d\Omega \approx \sum_{n=1}^N \psi_n(\vec{r}, E) w_n. \quad (3.34)$$

The specific choice of angular quadrature, meanwhile, is immaterial in all but the axial-polar PGD—one of the two 2D/1D PGD ROMs presented in Chapter 6. In that case, the polar and azimuthal angles $\mu$ and $\omega$ are separated, and so we find it convenient to adopt a quadrature defined as product of polar and azimuthal sub-quadratures: namely, a Gauss-Legendre polar quadrature and a midpoint (or Gauss-Chebyshev) azimuthal quadrature, as illustrated in Figure 3.3. Doing so, the full-order angular flux can easily be reconstructed from the axial-polar PGD. Beyond that, this quadrature has the useful properties of exactly integrating the spherical harmonics up to degree $2n - 1$, where $n$ is the order of the polar quadrature [148], and (like all product quadratures) allowing the polar and azimuthal resolutions to be adjusted individually. For these advantages, as well as convenience, this quadrature was likewise selected in all other numerical experiments.

3.2.4 Spatial Discretization by Discontinuous Galerkin Finite Elements

As enacting PGD entails the computation of several inner products (at least, in the continuous rather than discrete form), it is convenient to select a spatial discretization based on the weak form of the transport equation. This is no great burden, as the Finite Element Method (FEM) is often the spatial discretization of choice for neutron transport regardless—apart from perhaps the Method of Characteristics. Particularly, DGFEM [149]–[151] offers the capacity to model unstructured meshes (as in the FEM generally), avoids the “spurious oscillations” associated with non-upwinded discretizations of transport, and—in conjunction
Figure 3.3: A product Gauss-Legendre-Chebyshev angular quadrature (a) is formed by a Gauss-Legendre polar and midpoint azimuthal quadrature (b).

with the $S_N$ discretization in angle—lends itself to fast, memory-efficient, matrix-free block Gauss-Seidel iterations known as “transport sweeps,” described further in Section 3.3.2 and Appendix C.

The DGFEM discretization begins, as in all finite element methods, by taking the weak form of Equation 3.6 by multiplying both sides by an arbitrary test function $\psi^*$ and integrating in space over the domain $\mathcal{V}$. Having previously discretized in energy (by the multigroup approximation) and angle (by discrete ordinates) we can then expand the flux as a summation of basis functions indexed $b_j$

$$
\psi_{g,n}(\vec{r}) \approx \sum_j b_j(\vec{r}) \psi_{g,n,j} \quad (3.35)
$$

where $\psi_{g,n}$ is the block of vector $\psi$ containing the spatial degrees-of-freedom associated with group $g$ and ordinate $n$. Likewise, since we apply a Galerkin (rather than Petrov-Galerkin) FEM, the test function belongs to the same finite element space but without degrees-of-freedom

$$
\psi^*(\vec{r}) \equiv \sum_i b_i(\vec{r}) \quad (3.36)
$$

Doing so, the loss (streaming plus collision) operator $L_{g,n}$ can be written

$$
(\psi^*, \Omega_n \cdot \nabla \psi_{g,n} + \Sigma_t g \psi_{g,n})_V \equiv L_{g,n} \psi_{g,n} = \left( \psi^*, \Omega_n \cdot \nabla \psi_{g,n} \right)_V + (\psi^*, \Sigma_t g \psi_{g,n})_V \equiv (G_{g,n} + B_{g,n} \Sigma_t g) \psi_{g,n}. \quad (3.37)
$$
Note the unusual subscript of \( B_{G,N} \) is intended to denote it is the mass matrix for a single group and ordinate, but does not depend on \( g \) or \( n \); that is, each \( B_{g,n} = B_{G,N} \).

As detailed further in Appendix C, the discretization of \( L_{g,n} \) is completed by applying Gauss’s divergence theorem to the streaming term \( G_{G,n} \), imposing the upwind scheme, and selecting discontinuous shape functions. Meanwhile, the loss operator for all ordinates is \( L_g \) and will be block-diagonal, \( L_g \equiv \text{diag}(L_{g,1}, \ldots, L_{g,N}) \), if the incoming boundary fluxes are prescribed rather than reflected or white, which would introduce off-diagonal entries for ordinate-to-ordinate streaming. Furthermore, \( L_{g,n} \) will be block-lower-triangular—each block corresponding to an individual element—if a non-cyclic downstream ordering relative to \( \Omega_n \) through the mesh exists and is taken. Outside of such cases, we further subdivide the loss operator into lower and strictly-upper block-triangular components

\[
L_g = L_{g,l} + L_{g,u}
\]

such that \( L_{g,l} \) can easily be directly inverted, as explained in Section 3.3.2.

Meanwhile, recalling Equations 3.28–3.30, the remaining discrete operators (that is, matrices) can be written as

\[
D_g \psi_g \equiv \left[ \sum_{n'=1}^{N} Y_{\ell,m}(\Omega_n') \psi_{g,n'} w_{n'} \right]_{m=-\ell}^{\ell} = \phi_g,
\]

\[
\Sigma_{s,g'\rightarrow g} \phi_{g'} \equiv \left[ \Sigma_{s,\ell,g'\rightarrow g}(\vec{r}) \phi_{g',\ell,m} \right]_{m=-\ell}^{\ell} = \psi_{g'},
\]

\[
M_g \phi_g \equiv \left[ \sum_{\ell=0}^{L} \frac{2\ell + 1}{4\pi} \sum_{m=-\ell}^{\ell} Y_{\ell,m}(\Omega_n) \phi_{g,\ell,m} \right]_{n=1}^{N} \approx \psi_g.
\]

### 3.3 Iterative Solution

Given these preliminaries, Equation 3.6 can be approximated by the linear system of equations

\[
(L - BS) \psi = B (q + XF \psi)
\]

or, expanding the scattering, emission, and production operators,

\[
(L - BM\Sigma_s D) \psi = B (q + M_0 \chi \nu \Sigma_f D_0 \psi).
\]
That said, analysts are typically interested in solving one of two slightly modified forms of these equations: namely, either the fixed-source or criticality \((k\text{-eigenvalue})\) problem.

### 3.3.1 Fixed-Source Problems

The first of these scenarios, the fixed-source problem, describes a system in which there is no fission, neutrons being born only out of exogenous source (vector) \(q\), as in

\[
(L - BM\Sigma_sD) \psi = Bq. \tag{3.44}
\]

Of course, \(q\) may itself be some approximation of the fission source, such as

\[
q = XF\psi^{(i-1)} \tag{3.45}
\]

where \(\psi^{(i-1)}\) is the flux (vector) from a previous iteration or some initial guess. In what follows, the fixed-source problem is used as an example to explain two important iterative procedures in neutron transport: transport sweeps and block Gauss-Seidel iteration in energy. Subsequently, in Section 3.3.4, the \(k\)-eigenvalue problem is introduced.

### 3.3.2 Transport Sweeps

As explained in Sections 3.2.2 and 3.2.4, the multigroup loss operator \(L\) is block-diagonal, and the blocks \(L_g\) are block-lower-triangular, or nearly so. This structure can be exploited to directly invert \(L\) by block forward substitution (detailed further in Appendix C), as in \(L^{-1} = \text{diag} \left( L_1^{-1}, \ldots, L_g^{-1}, \ldots, L_G^{-1} \right) \). Applying this transport sweep \(L^{-1}\) to both sides of Equation 3.44 then yields

\[
\left( I + L^{-1} (L - BM\Sigma_sD) \right) \psi = L^{-1}Bq. \tag{3.46}
\]

Further, multiplying both sides by \(D\) achieves

\[
\left( I - L^{-1}BM\Sigma_s \right) \phi + L^{-1}L\psi = DL^{-1}Bq, \tag{3.47}
\]

where the only remaining angular flux \(\phi\) is the operand of \(L\). Since this operator is, by definition, zero except at special boundaries or in cyclic streaming paths, one need only store the few entries of \(\psi\) corresponding to nonzero columns of \(L\), the “significant angular flux,”
here denoted $\overline{\psi} = I \psi$ [152]. Using this vector and the transpose of our mapping, $I^\top$, one can rewrite Equation 3.47 as

$$\left( I - L^{-1}BM\Sigma_s \right) \phi + L^{-1}LI^\top\overline{\psi} = DL^{-1}Bq. \quad (3.48)$$

In this way, the full-order model is usually implemented such that $\psi$ is never stored in its entirety, only $\phi$ and $\overline{\psi}$. As will be seen, however, the PGD ROM introduces a coupling between modes that (typically) cannot be represented exactly as an expansion in the form of $\phi$, incurring a substantial, but not prohibitive, memory burden. That is, unless $\psi$ is recomputed as needed, at the cost of one transport sweep

$$\psi = L^{-1} \left( B \left( q + M\Sigma_s \phi \right) - LL^\top \overline{\psi} \right), \quad (3.49)$$

but, for ease of implementation and to avoid the accompanying expense, this option is not investigated further here. Mitigating this storage requirement could present an area for future research.

### 3.3.3 Block Gauss-Seidel Iteration in Energy

As discussed in Section 3.2.2, $\Sigma_s$ is typically almost block lower triangular. A natural approach is then block Gauss-Seidel in energy. Representing $\Sigma_s$ as lower and strictly upper triangular components

$$\Sigma_s = \underline{\Sigma}_s + \overline{\Sigma}_s \quad (3.50)$$

the block Gauss-Seidel iteration in energy is written

$$\left( I + L^{-1} \left( L - BM\Sigma_s D \right) \right) \psi^{(i)} = L^{-1}B \left( q + M\Sigma_s D \psi^{(i-1)} \right) \quad (3.51)$$

where the left-hand-side matrix can be inverted by block forward substitution

$$= L^{-1}_g B_g \left( q_g + M_g \left( \sum_{g' < g} \Sigma_{s,g' \rightarrow g} \phi^{(i)}_{g'} + \sum_{g' > g} \Sigma_{s,g' \rightarrow g} \phi^{(i-1)}_{g'} \right) \right) \quad (3.52)$$
proceeding sequentially from $g \leftarrow 1 \ldots G$. These diagonal or “within-group” blocks must each be solved (iteratively) to compute $\psi_g^{(i)}$. The simplest means of doing so is Richardson Iteration, historically referred to as “source iteration” in this context [153]. But, of course, any iterative linear solver for non-symmetric matrices would apply. Here, we find it advantageous to use a Krylov subspace method: namely, Generalized Minimal Residual (GMRES) [154]. Additionally, while preconditioning, such as by “Diffusion Synthetic Acceleration” (DSA) [153], would likely accelerate this iterative solution, none is implemented here for simplicity. The implications of this choice, as well as prospects for future preconditioners, are discussed further in Section 7.3.2.

3.3.4 Criticality Problems

Secondly, for a multiplying system at steady-state, with no external source,

$$\begin{align*}
(L - BS) \psi &= BXF \psi . \\
(L - BS) \psi &= \frac{1}{k} BXF \psi \\
(L - BS) \psi &= \frac{1}{k} BXF \psi
\end{align*}$$

However, this holds only when the nuclear reaction is exactly self-sustaining; to represent the case where there may be an imbalance (the reaction is either increasing or decreasing), one can scale the right-hand-side (the fission source) by $1/k$

$$\begin{align*}
(L - BS) \psi &= \frac{1}{k} BXF \psi \\
(L - BS) \psi &= \frac{1}{k} BXF \psi
\end{align*}$$

which is equal to Equation 3.53 if $k = 1$. As in Equation 3.46, both sides can be multiplied by $L^{-1}$

$$\begin{align*}
\left( I + L^{-1} (L - BS) \right) \psi &= \frac{1}{k} L^{-1} BXF \psi \\
\left( I + L^{-1} (L - BS) \right) \psi &= \frac{1}{k} L^{-1} BXF \psi
\end{align*}$$

Likewise, as in Equation 3.47, one could further multiply by $D$ so as to avoid storing $\psi$ in completeness, though we refrain from doing so here. More succinctly, gathering these matrices into the fixed-source and fission operators $B$ and $A$,

$$\begin{align*}
B \psi &= \frac{1}{k} A \psi , \\
B \psi &= \frac{1}{k} A \psi
\end{align*}$$

or equivalently, multiplying both sides by $k$ and rearranging,

$$\begin{align*}
A \psi &= k B \psi . \\
A \psi &= k B \psi
\end{align*}$$
Evidently, this is an eigenvalue problem with eigenvalues $k$ and eigenvectors $\psi$, the latter representing the degrees-of-freedom of eigenfunctions $\psi$. Typically, the eigenpair—as in, pair of $k$ and $\psi$—of interest is the dominant one (that with the largest eigenvalue) $k_0$ and $\psi_0$; this describes the asymptotic state of the “static” reactor. As such, we hereafter refer to this fundamental mode as simply $\psi$ (and likewise $k$, $\psi$) for sake of brevity. Physically, $k$—or the “$k$-eigenvalue”—represents the multiplication factor, the ratio of neutrons produced by fission to those lost by leakage or absorption. Naturally, $k > 1$ indicates supercriticality (the reaction is increasing), $k = 1$ indicates criticality (the reaction is self-sustaining), and $k < 1$ indicates subcriticality (the reaction is decreasing).

Regarding numerical solution, Equation 3.57 is a generalized eigenvalue problem and can be solved as such. Alternatively, the eigenproblem can be standardized by multiplying both sides by $B^{-1}$

$$B^{-1}A\psi = k\psi.$$  \hspace{1cm} (3.58)

where each application of $B^{-1}$ amounts to solving a fixed-source problem. To do so, one can conveniently use a block Gauss-Seidel solver or preconditioner in energy, as discussed in Section 3.3.3. Note that this eigenproblem cannot be standardized by multiplying by $A^{-1}$, as $A$ is not full-rank and therefore not invertible. In the present work, we elect to solve Equation 3.58 with a shift-of-origin and a Krylov-Schur eigensolver.

These numerical methods in hand, we are now equipped to solve the full-order model of multigroup, $S_N$, DGFEM radiation transport. Moreover, as all PGD “submodels” with the exception of that in energy can be interpreted as a degenerate form of this full-order model, these pose only minor difficulties, as will be demonstrated. As such, we proceed to our first application of PGD, model order reduction in energy.
CHAPTER 4
MODEL ORDER REDUCTION IN ENERGY

4.1 Introduction

Accurate simulation of neutron transport is challenged by the effect of nuclear resonances, dramatic variations in neutron reaction probabilities (cross sections) in an energy range between the speeds at which neutrons are typically emitted and absorbed to induce fission. Resolving this structure requires an energy mesh far finer than is practical for reactor physics applications, motivating a progressive, flux-weighted coarsening of this data, known as cross section condensation. In practice, this requires a series of neutron transport simulations of still-formidable energy fidelity and introduces irrecoverable error [7].

In this chapter, we propose a ROM of neutron transport separated in energy by PGD [3] as a means of mitigating (or obviating) the challenges of fine-group neutron transport and cross section condensation. This PGD ROM is subsequently extended to $k$-eigenvalue problems in Chapter 5. Specifically, we approximate the neutron flux $\psi$ as a summation of $M$ products of spatio-angular and energetic modes $\psi_m$ and $\xi_m$,

$$\psi(\vec{r},\vec{\Omega},E) \approx \sum_{m=1}^{M} \psi_m(\vec{r},\vec{\Omega})\xi_m(E),$$

(4.1)

for position $\vec{r}$, direction (angle) $\vec{\Omega}$, and energy $E$. The advantage is clear: assuming the number of spatio-angular and energetic unknowns are $N_{(r,\Omega)}$ and $G$ respectively, we have taken a problem of dimension $N_{(r,\Omega)} \times G$ and reduced it to size $M \times (N_{(r,\Omega)} + G)$. As either number grows large, the ROM becomes dramatically smaller than the full-order model, so
long as $M$ is reasonably low. Practically, the cost is particular to the means by which we find these modes (for example, progressive PGD with or without update), but it is this model order reduction which makes our approximation compelling—especially compared to cross section condensation, which only reduces $G$.

4.2 Motivation

The energetic dimension of neutron transport is especially challenging to discretize, owing to the extremely fine structure of cross sections in the so-called resonance range of the energy scale. Plotting\(^5\) the total (and fission, for fissile nuclides) cross sections of $^{239}$Pu, $^{238}$U, and $^{235}$U in Figure 4.1, one can immediately recognize these resonances and the difficulty posed to numerical approximations.

![Figure 4.1: Neutron cross sections of uranium and plutonium, including nuclear resonances.](image)

Accurately representing such a function requires an extremely fine energy mesh, typically composed of energy groups, within which the cross section is assumed to be constant. Knott and Yamamoto assert between 20,000 and 100,000 (“some tens of thousand to a hundred thousand,” p. 1032) of these groups are typically required to obviate any special treatment of resolved nuclear resonances [1]. This regime of fidelity, known as ultrafine-group, when applied to neutron transport over a nuclear reactor core, is intractable even on modern

\(^5\)Using Java-based Nuclear Information Software (JANIS) [157].
supercomputers, let alone those of decades past or a personal workstation, usually in both computational and memory requirements.

### 4.2.1 Cross Section Condensation

![Figure 4.2: The traditional domain of ultrafine-group calculations.](image)

As such, ultrafine-group analyses (if used at all) must typically be restricted to pin-cell calculations where a square nuclear fuel pin-cell is approximated as a cylinder. In polar coordinates, one can then eliminate all spatial variables except the radius $r$ by azimuthal symmetry, as depicted in Figure 4.2. This simplified simulation is used to calculate region-wise reaction rates for cross section condensation. Put simply, this is a process of computing constants which preserve these integral rates, but with a progressively fewer number of groups. For example, the total cross section $\Sigma_{t,j,g}$ for region $V_j$ and group $g$ may be condensed from (continuous-energy or finer-group) $\Sigma_t$ by writing

$$
\int_{V_j} \int_{4\pi} \int_{E_g}^{E_g-1} \Sigma_t(\vec{r}, E) \psi(\vec{r}, \vec{\Omega}, E) dEd\Omega dr = \Sigma_{t,j,g} \int_{V_j} \int_{4\pi} \psi_g(\vec{r}, \vec{\Omega}) d\Omega dr,
$$

where the group flux $\psi_g$ is defined as in Equation 3.17, leading to the identity

$$
\Sigma_{t,j,g} = \frac{\int_{V_j} \int_{4\pi} \int_{E_g}^{E_g-1} \Sigma_t(\vec{r}, E) \psi(\vec{r}, \vec{\Omega}, E) dEd\Omega dr}{\int_{V_j} \int_{4\pi} \int_{E_g}^{E_g-1} \psi(\vec{r}, \vec{\Omega}, E) dEd\Omega dr}.
$$

Clearly, if $\psi$ is known, the condensed cross sections will exactly verify Equation 4.2, recovering the integral reaction rates. In practice, however, only an estimate of the flux will be available, rendering this equality an approximation.

As the simplest, indivisible unit of a light water reactor’s geometry, this pin-cell calculation, if performed, is usually the highest level of a repeated, multi-scale condensation workflow.\(^6\) Following the pin-cell simulation, condensed cross sections are used to simulate a

\(^6\)Generally preceded only by the condensation of nuclear data assuming a homogeneous, infinite medium,
fuel assembly (an array of pin-cells), from which even coarser cross sections are condensed for the full-core model (an array of assemblies). We summarize such a procedure in Figure 4.3.

4.2.1.1 Spatio-angular Condensation Errors

While cross section condensation is the workhorse of the reactor physics community, its application can still lead to deleterious effects. For example, while condensation aims to retain region-wise integral reaction rates, it may not accurately represent the angular or spatial distributions within these regions.

This is especially evident in cases of fuel self-shielding, illustrated in Figure 4.4. If one imagines neutrons of an energy corresponding to a resonance peak, it is obvious many fewer will exit than enter, as they are sharply, exponentially attenuated. Yet, Equation 4.3 defines the condensed total cross section $\Sigma_{t,j,g}$ as isotropic. As demonstrated by Boyd et al. [7] and Nelson et al. [8] (and replicated here), this assumption—neglecting the intra-zone spatio-angular dependence of reaction rate—can lead to appreciable discrepancies in resonance-group fluxes. However, eliminating this error would require that cross sections possess the

---

with a preliminary correction for heterogeneity by “equivalence in dilution.” These steps can be omitted if the pin-cell calculation is ultrafine.
same spatio-angular dependence as the flux from which they are computed, limiting their applicability and imposing an extreme memory cost. As such, some irrecoverable error must be accepted, even if the reference solution is exact.

Figure 4.4: Spatio-angular disparities caused by resonance self-shielding. Dashed lines represent condensed cross sections. Percentages indicate difference relative to reference cross section (in black).

4.2.1.2 Validity of the Condensation Spectrum

These issues are compounded by the fact that the reference solution is almost never exact in practice. Rather, as is evident in Figure 4.3, the condensation problem is always simpler (less spatially detailed) than that in which the cross sections are to be used. Almost invariably, this simplification neglects to explicitly model the neighboring pin-cells or assemblies,\textsuperscript{7} instead modeling the unit-cell as an infinite lattice, and takes a two-dimensional plane to represent a zone of a three-dimensional geometry. Obviously, at the interfaces of unit-cells and axial zones, plus the core periphery, these assumptions break down—especially when neighbors are appreciably different, such as a fuel pin and a control rod, or UO\textsubscript{2} and Mixed Oxide (MOX) assemblies—introducing some indeterminate error. This could even be considered a lack of generality, in that analysts rely on the short neutron mean-free-paths of light water reactors to mitigate these interfacial disparities; when considering graphite-moderated \cite{161}.

\textsuperscript{7}Though their influence may be otherwise accounted for, such as by boundary albedos \cite{158} or (B\textsubscript{1} or P\textsubscript{1}) leakage corrections \cite{159}, \cite{160}.
or fast reactors [162]–[164], these discrepancies can become unacceptably severe.

4.2.2 Reduced-Order Modeling for Fine-Group Neutron Transport

Having just briefly discussed the state-of-the-art in deterministic neutron transport, it is clear the challenge stems from the necessity to condense cross sections and the computational difficulty in doing so. Naturally, an approximate model of neutron transport could relieve this necessity—allowing finer cross sections in the target application and therefore the omission of one or several condensation steps. Moreover, what preceding condensation problems remain could be approximated by the same model—either reducing the computational burden or allowing greater spatio-angular and energetic detail, alleviating the errors discussed in Section 4.2.1.2. Practically, as cross section condensation is itself an approximation, this model need not approach exactitude, but merely contribute marginal, or less, error compared to condensation alone. It is in this context that we propose a (projection-based) ROM by PGD.

4.3 Previous Work

The goal of precisely and deterministically computing the neutron flux (and multiplication factor $k$, see Chapter 5) in reactor physics has motivated a body of literature too expansive to be reviewed here. (See [1], [165] for broader accounts.) However, a brief list of notable or ongoing efforts to enhance cross section condensation and/or coarse-group neutron transport could include: Discontinuity Factors [53], [166]; both Generalized Energy Condensation [167] and Discrete Generalized Multigroup [168]; angularly-dependent cross sections [7], [8], Superhomogenization factors [169]–[171]; and transport corrected cross sections [147]. While these approaches generally seek to remedy some source of error in coarse-group simulations, our strategy is instead to devise a model in which the fine-group (or at least, a finer-group) simulation is tractable. In doing so, we aim to obviate, or at least reduce the scope of, this coarsening, and thereby eliminate or mitigate the corresponding error. Of course, there is a trade-off in that the ROM will incur some approximation error of its own—however, we hope to demonstrate that this error is comparatively small, or even negligible, such that it remains a preferable alternative.

In the domain of reduced-order modeling, Reed and Roberts [86] recently characterized the use of the POD [16], [17] to determine spectral (intra-group) basis sets for DGM, which was found to yield a more effective truncated expansion. For applications in RMMs, Reed
and Roberts [172] also employed POD to determine a basis set by which to expand the energetic dependence of the nodal boundary conditions. In a full-core, two-group nodal diffusion scheme, Gamarino et al. [173] used POD to compute a discrepancy spectrum which iteratively corrects cross section weighting spectra calculated assuming an infinite medium. These are in addition to the ROMs of Sections 2.2.2 (DMD) and 2.2.1 (POD), which separate dimensions or parameters besides energy.

Meanwhile, for PGD, Dominesey and Ji [142] demonstrated separation in energy using an analytical (isotropic) scattering kernel, coupled with one-dimensional neutron diffusion. Prince and Ragusa [143] also pursued a similar method, but with multigroup energy and simultaneous decomposition of neutron diffusion in two or three spatial dimensions. Other applications of PGD to radiation transport and diffusion (which do not pertain to energy) are discussed in Section 2.2.4.

However, we now seek to build upon prior work by applying energy-separated PGD to, and comparing it against, cross section condensation. Doing so conclusively requires coupling with transport (rather than diffusion) and on unstructured two-dimensional geometry, which cannot be further spatially decomposed by PGD, two aspects which complicate numerical implementation. In this light, we invoke similar methodology to previous work, but here employ state-of-the-art discretizations for neutron transport, discrete ordinates ($S_N$) in angle and DGFEM [150], [152] in space; we also explore the implications thereof, including regard to error indicators, caching, and storage of the angular flux. Doing so, we make possible and conduct numerical experiments on practical applications rather than academic examples. Specifically, we exploit this capability to characterize the effectiveness of PGD (separating in energy) for pin-cell problems with industry-standard energy meshes, both to approximate the flux and to condense cross sections, with reference to full-order solutions. These results bear directly on industry practice, where such problems are ubiquitous and necessary, providing a meaningful test of the promises of PGD.
4.4 Methodology

We begin with the traditional, continuous-energy neutron transport equation, reproduced below for convenience,

\[ \vec{\Omega} \cdot \nabla \psi(\vec{r}, \vec{\Omega}, E) + \Sigma_t(\vec{r}, E)\psi(\vec{r}, \vec{\Omega}, E) = \int_{E_0}^{E_G} \int_{4\pi} \Sigma_s(\vec{r}, \vec{\Omega}' \rightarrow \vec{\Omega}, E' \rightarrow E)\psi(\vec{r}, \vec{\Omega}', E')d\Omega'dE' + q(\vec{r}, \vec{\Omega}, E), \]  

(4.4)

where we will subsequently isolate the energetic dependence by PGD.

4.4.1 Separation of Coefficients

To impose our separated scheme, we must first separate the coefficients of each term: here, simply the macroscopic total and scattering cross sections \( \Sigma_t \) and \( \Sigma_s \). We can do so easily by defining our cross sections as a summation of \( J \) materials indexed \( j \), with spatially-dependent number densities \( N_j \) and microscopic cross sections \( \sigma_{t,j} \) and \( \sigma_{s,j} \). Further, to decouple the angular and energetic variation of \( \sigma_{s,j} \) we approximate this cross section as a Legendre series of finite size \( L \). These separations can then be written as

\[
\Sigma_t(\vec{r}, E) = \sum_{j=1}^{J} N_j(\vec{r})\sigma_{t,j}(E),
\]

(4.5)

\[
\Sigma_s(\vec{r}, \vec{\Omega}' \rightarrow \vec{\Omega}, E' \rightarrow E) = \sum_{j=1}^{J} N_j(\vec{r})\sigma_{s,j}(\vec{\Omega}' \rightarrow \vec{\Omega}, E' \rightarrow E)
\approx \sum_{j=1}^{J} N_j(\vec{r}) \sum_{\ell=0}^{L} \frac{2\ell + 1}{4\pi} P_\ell(\vec{\Omega}' \cdot \vec{\Omega})\sigma_{s,\ell,j}(E' \rightarrow E),
\]

(4.6)

where \( P_\ell \) represents the Legendre polynomial of degree \( \ell \).

Notably, our decomposition of the spatial dependence is exact (at least for microscopic, rather than multigroup collapsed, cross sections), and the expression of the scattering cross section by Legendre moments is ubiquitous even in full-order neutron transport. Therefore, we have yet to impose any special approximations for PGD.
4.4.2 Separated Representation

We can now apply our separated representation of the flux \( \psi \) as a summation of \( M \) modes indexed \( m \), as in Equation 4.1, and likewise the source \( q \) as \( I \) functions indexed \( i \),

\[
q(\vec{r}, \vec{\Omega}, E) = \sum_{i=1}^{I} q_i(\vec{r}, \vec{\Omega})\bar{\chi}_i(E),
\]

achieving the separated equation of neutron transport

\[
\sum_{m=1}^{M} \left( \vec{\Omega} \cdot \nabla \psi_m(\vec{r}, \vec{\Omega})\xi_m(E) + \sum_{j=1}^{J} N_j(\vec{r})\psi_m(\vec{r}, \vec{\Omega})\sigma_{t,j}(E)\xi_m(E) \right) \\
- \sum_{j=1}^{J} N_j(\vec{r}) \sum_{\ell=0}^{L} \frac{2\ell + 1}{4\pi} \int P_{\ell}(\vec{\Omega}' \cdot \vec{\Omega})\psi_m(\vec{r}, \vec{\Omega})d\Omega' \int_{E_0}^{E_G} \sigma_{s,\ell}(E' \rightarrow E)\xi_m(E')dE' \right) \\
= \sum_{i=1}^{I} q_i(\vec{r}, \vec{\Omega})\bar{\chi}_i(E)
\]

from which each of our projection-based ROMs will be derived.

Particularly, having separated our variables, PGD proceeds as follows (and as described further below). First, we take the Petrov-Galerkin projection of Equation 5 by integration against energetic and spatio-angular weight functions \( \tilde{\xi}_M \) and \( \tilde{\psi}_M \), yielding two low-order models detailed in Sections 5.4.2 and 5.4.3. Then—solving the former for \( \psi_M \) and the latter for \( \xi_M \)—we alternate between the two until the pair of modes converges. In the Minimax PGD (Section 4.4.5), two additional adjoint problems are subsequently introduced by which to compute \( \tilde{\psi}_M \) and \( \tilde{\xi}_M \). Alternatively, setting \( \tilde{\psi}_M \leftarrow \psi_M \) and \( \tilde{\xi}_M \leftarrow \xi_M \) simplifies the Petrov-Galerkin to the Galerkin PGD. Upon convergence, we can optionally solve an linear system of size \( MG \times MG \) to update (recompute) the energy modes \( \{\xi_m\}_{m=1}^{M} \), as described in Section 5.4.5, to achieve the Progressive with Update PGD. This procedure then repeats, incrementing \( M \) until the series converges (as measured by some error estimator) or a maximum number of modes is reached.

4.4.3 Spatio-angular Submodel

To derive the spatio-angular submodel intrinsic to this PGD ROM, we begin by operating on Equation 4.8 by

\[
\int_{E_G}^{E_0} \xi_M(E)dE \equiv \left( \xi_M, \cdot \right)_G.
\]
Omitting the functional arguments within inner products (for brevity, as they become scalars), this operation yields

\[
\sum_{m=1}^{M} \left( \langle \xi_M, \xi_m \rangle \right)_G \vec{\Omega} \cdot \nabla \psi_m(\vec{r}, \vec{\Omega}) + \sum_{j=1}^{J} N_j(\vec{r}) \langle \xi_M, \sigma_{t,j} \xi_m \rangle \psi_m(\vec{r}, \vec{\Omega})
\]

\[
- \sum_{j=1}^{J} N_j(\vec{r}) \sum_{\ell=0}^{L} \frac{2\ell + 1}{4\pi} \left( \langle \xi_M, \int_{E_0}^{E} \sigma_{s,\ell,j} \xi_m dE' \rangle \right)_G \int_{4\pi} P_{\ell}(\vec{\Omega}' \cdot \vec{\Omega}) \psi_m(\vec{r}, \vec{\Omega}) d\Omega'
\]

\[
= \sum_{i=1}^{I} \left( \xi_M, \bar{\chi}_i \right)_G q_i(\vec{r}, \vec{\Omega}),
\]

which is equivalent to the conventional, energy-independent (or one-group/in-group) transport equation. This can be more easily realized by rearranging the coefficients to resemble the typical macroscopic cross sections

\[
\bar{\Sigma}_{t,M,m}(\vec{r}) \equiv \sum_{j=1}^{J} N_j(\vec{r}) \left( \langle \xi_M, \sigma_{t,j} \xi_m \rangle \right)_G,
\]

\[
\bar{\Sigma}_{s,\ell,M,m}(\vec{r}) \equiv \sum_{j=1}^{J} N_j(\vec{r}) \left( \langle \xi_M, \int_{E_0}^{E} \sigma_{s,\ell,j} \xi_m dE' \rangle \right)_G,
\]

and likewise the source

\[
\bar{q}_M(\vec{r}, \vec{\Omega}) \equiv \sum_{i=1}^{I} \left( \xi_M, \bar{\chi}_i \right)_G q_i(\vec{r}, \vec{\Omega}).
\]

Doing so, the linear operator (for test and trial indices \( M \) and \( m \)) is defined as

\[
B^{(G)}_{M,m} \psi_m \equiv \left( \xi_M, \xi_m \right)_G \vec{\Omega} \cdot \nabla \psi_m(\vec{r}, \vec{\Omega}) + \bar{\Sigma}_{t,M,m}(\vec{r}) \psi_m(\vec{r}, \vec{\Omega})
\]

\[
- \sum_{\ell=0}^{L} \frac{2\ell + 1}{4\pi} \bar{\Sigma}_{s,\ell,M,m}(\vec{r}) \int_{4\pi} P_{\ell}(\vec{\Omega}' \cdot \vec{\Omega}) \psi_m(\vec{r}, \vec{\Omega}) d\Omega',
\]

such that Equation 4.10 can be more succinctly written as

\[
\sum_{m=1}^{M} B^{(G)}_{M,m} \psi_m = \bar{q}_M
\]
or, equivalently, moving the known summands \((m < M)\) to the right-hand-side and introducing a residual term \(r_M\),

\[
B_{M,M}^{(g)} \phi_M = \bar{q}_M - \sum_{m=1}^{M-1} B_{M,m}^{(g)} \phi_m = \bar{r}_M^{(g)}.
\] (4.16)

This equation can be made even more similar by dividing both sides by \((\xi_M,\xi_M)_G\), as by amending the definitions

\[
\bar{\Sigma}_{t,M,M}(\bar{r}) \leftarrow \bar{\Sigma}_{t,M,M}(\bar{r}) / (\xi_M,\xi_M)_G,
\]

\[
\bar{\Sigma}_{s,t,M,M}(\bar{r}) \leftarrow \bar{\Sigma}_{s,t,M,M}(\bar{r}) / (\xi_M,\xi_M)_G,
\]

\[
\bar{r}_M(\bar{r},\bar{\Omega}) \leftarrow \bar{r}_M(\bar{r},\bar{\Omega}) / (\xi_M,\xi_M)_G,
\] (4.17)

such that (reverting to the original, verbose notation) Equation 4.16 becomes

\[
\bar{\Omega} \cdot \nabla \phi_M(\bar{r},\bar{\Omega}) + \bar{\Sigma}_{t,M,M}(\bar{r}) \phi_M(\bar{r},\bar{\Omega})
- \sum_{\ell=0}^{L} \frac{2\ell + 1}{4\pi} \bar{\Sigma}_{s,t,M,M}(\bar{r}) \int_{4\pi} P_{\ell}(\bar{\Omega}' \cdot \bar{\Omega}) \phi_M(\bar{r},\bar{\Omega}') d\Omega = \bar{r}_M(\bar{r},\bar{\Omega}),
\] (4.18)

which differs from traditional neutron transport only by our redefinition of the cross sections and source. Linearizing this equation by granting that \(\bar{\xi}_M\) and \(\{\xi_m\}_{m=1}^M\) are known, \(\phi_M\) can then be computed using any discretization applicable to radiation transport, such as long characteristics (especially popular for lattice physics [1]), short characteristics, or finite differences in space, and likewise \(S_N\) or \(P_N\) in angle, and so on. Here, the selected discretizations are \(S_N\) and DGFEM, as discussed in Sections 3.2.3 and 3.2.4.

That said, the definition of \(\bar{r}_M\) introduces two slight peculiarities referenced only obliquely in these sections. Namely, while the source \(q\) often arises only from in-scattering and fission in the full-order model, and may therefore expressed as a spherical harmonics expansion, the same is not true of \(\bar{r}_M\) for \(M > 1\)—specifically, because the latter incorporates streaming and collision products of \(\phi_m\), where \(m < M\). This is perhaps inconvenient, but easily overcome; the more pressing implication is that one must store several modes of the angular flux (or products thereof) in order to compute \(\bar{r}_M\). As the (one-group) transport sweep \(D_g L_g^{-1} B_g M_g\) is usually implemented such that the angular flux is never stored in its
entirety—only the “significant” portion of this incurs the additional memory overhead of PGD alluded to in Section 3.3.2. Naturally, eliminating or mitigating this storage burden appears a fruitful direction for future research in PGD ROMs of neutron transport.

4.4.4 Energetic Submodel

Turning our attention to the PGD submodel in energy, we proceed likewise, obtaining the weak form of Equation 4.8 by

$$\int_{4\pi} \bullet \psi_m^*(\vec{r},\vec{\Omega})d\Omega \equiv (\psi_m^*(\vec{r},\vec{\Omega}),\bullet)_{\mathcal{V},4\pi},$$

though, with test function indexed $m^*$ rather than $M$, so that the update step can be formulated later. By this operation, again suppressing arguments $\vec{r}$ and $\vec{\Omega}$ within inner products, we find

$$\sum_{m=1}^{M} \left( \left( \psi_m^*,\vec{\Omega} \cdot \nabla \psi_m \right)_{\mathcal{V},4\pi} + \sum_{j=1}^{J} \left( \psi_m^*,N_j \psi_m \right)_{\mathcal{V},4\pi} \sigma_{t,j}(E) \right) \xi_m(E)$$

$$- \sum_{\ell=0}^{L} \frac{2\ell + 1}{4\pi} \sum_{j=1}^{J} \left( \psi_m^*,N_j \int_{4\pi} P_\ell \psi d\Omega' \right)_{\mathcal{V},4\pi} \int_{E_0}^{E_G} \sigma_{s,\ell}(E' \to E) \xi_m(E')dE'$$

$$= \sum_{i=1}^{I} \left( \psi_m^*,q_i \right) \bar{\chi}_i(E),$$

which amounts to a model of neutron slowing down in a homogeneous, infinite medium. Again, adjusting the notation to reveal the intuitive interpretation, let us write the effective cross sections and source

$$\bar{\Sigma}_{t,m^*,m}(E) \equiv \sum_{j=1}^{J} \left( \psi_m^*,N_j \psi_m \right)_{\mathcal{V},4\pi} \sigma_{t,j}(E) + \left( \psi_m^*,\vec{\Omega} \cdot \nabla \psi_m \right)_{\mathcal{V},4\pi},$$

$$\bar{\Sigma}_{s,\ell,m^*,m}(E' \to E) \equiv \sum_{j=1}^{J} \frac{2\ell + 1}{4\pi} \left( \psi_m^*,N_j \int_{4\pi} P_\ell \psi_m d\Omega' \right)_{\mathcal{V},4\pi} \sigma_{s,\ell,j}(E' \to E),$$

$$\bar{\chi}_{m^*}(E) \equiv \sum_{i=1}^{I} \left( \psi_m^*,q_i \right)_{\mathcal{V},4\pi} \bar{\chi}_i(E),$$

allowing the linear operator $B_{m^*,m}^{(\mathcal{V},4\pi)}$ to be defined as

$$B_{m^*,m}^{(\mathcal{V},4\pi)} \xi_m \equiv \bar{\Sigma}_{t,m^*,m}(E)\xi_m(E) - \sum_{\ell=0}^{L} \int_{E_0}^{E_G} \bar{\Sigma}_{s,\ell,m^*,m}(E' \to E)\xi_m(E')dE'.$$
such that Equation 4.20 may be concisely rewritten

$$\sum_{m=1}^{M} B_{m*}^{(V,\lambda)} \xi_m = \bar{\chi}_{m*}.$$  \hspace{1cm} (4.25)

Again, rearranging these terms to yield a linear system relating $\xi_M$ and a residual $\bar{r}_{M*}$,

$$B_{m*}^{(V,\lambda)} \xi_M = \bar{\chi}_{m*} - \sum_{m \neq m*}^{M} B_{m*}^{(V,\lambda)} \xi_m$$

$$= \bar{r}_{m*}^{(V,\lambda)},$$

or, more expressively,

$$\Sigma_{t,m*,m*}(E)\xi_m(E) - \sum_{\ell=0}^{L} \int_{E_0}^{E} \Sigma_{s,t,m*,m*}(E' \to E)\xi_m(E')dE' = \bar{r}_{m*}^{(V,\lambda)}(E),$$

which, as in the spatio-angular case, differs from the familiar equation only by the cross sections and source. In fact, one can even see the infinite medium problem as a degenerate case of this model, where one mode is given ($M = 1$) and the spatio-angular mode is uniform and isotropic, such that the gradient $\nabla \psi_{m=1}$ vanishes and the cross sections become volume-weighted rather than flux-weighted quantities. Further, this model becomes linear if $\bar{\psi}_{m*}$ and $\{\psi_m\}_{m=1}^{M}$ are fixed. As in the full-order model, the discretization of choice for this integral equation is the multigroup method; further particularities of the numerical implementation are reviewed in Section 4.4.8. At present, having achieved the spatio-angular and energetic submodels, Equations 4.18 and 4.27, we now turn to specifying the test functions—both for Galerkin and Minimax PGD—and elaborating the overall PGD algorithm.

### 4.4.5 Galerkin and Minimax Formulations

Given these equations for modes $\psi_M$ and $\xi_M$, the only remaining undefined terms are the test functions $\tilde{\psi}_M$ and $\tilde{\xi}_M$. In the Galerkin PGD these are simply defined as $\psi_M$ and $\xi_M$. In a Petrov-Galerkin PGD, these weight functions have some other definition; in the Minimax PGD specifically, we introduce additional (adjoint) equations to find these functions. As a complete description of the Minimax PGD is given by Nouy [9], it suffices for our purposes to note that this derivation proceeds by likening the test functions to Lagrange multipliers. In short, the PGD enrichment problem can be viewed as a constrained optimization problem:
maximize the $L^2$ norm of mode $M$ subject to the constraint

$$B \left( \tilde{\psi}_M \xi_M, \sum_{m=1}^{M} \psi_m \xi_m \right) = Q \left( \tilde{\psi}_M \tilde{\xi}_M \right)$$

(4.28)

where $B(\cdot, \cdot)$ and $Q(\cdot)$ are the bilinear and linear (weak) forms corresponding to Equation 4.8. This leads to the Lagrangian

$$\mathcal{L}_M \left( \tilde{\psi}_M \tilde{\xi}_M, \psi_M \xi_M \right) \equiv \frac{1}{2} \left( \psi_M, \psi_M \right)_{\mathcal{V}, \mathcal{A}_\pi} (\xi_M, \xi_M)_{\mathcal{G}}$$

$$+ Q \left( \tilde{\psi}_M \tilde{\xi}_M \right) - B \left( \tilde{\psi}_M \tilde{\xi}_M, \sum_{m=1}^{M} \psi_m \xi_m \right).$$

(4.29)

The stationary conditions of $\mathcal{L}_M$—those under which the partial derivative of $\mathcal{L}_M$ is zero—with respect to $\psi_M$ and $\xi_M$ are Equations 4.18 and 4.27. Likewise, enforcing the stationary conditions with respect to $\tilde{\psi}_M$ and $\tilde{\xi}_M$ yields

$$B \left( \psi^* \xi_M, \tilde{\psi}_M \tilde{\xi}_M \right) = \left( \psi^*, \psi_M \right)_{\mathcal{V}, \mathcal{A}_\pi} (\xi_M, \xi_M)_{\mathcal{G}}$$

(4.30)

$$B \left( \psi_M \xi^*, \tilde{\psi}_M \tilde{\xi}_M \right) = \left( \psi_M, \psi_M \right)_{\mathcal{V}, \mathcal{A}_\pi} (\xi^*, \xi_M)_{\mathcal{G}}.$$  

(4.31)

where $\psi^*$ and $\xi^*$ are arbitrary functions. Equivalently, reverting to the strong form and recalling our previously-defined operators,

$$\left( B_{M,M}^{(\mathcal{G})} \right)^\dagger \tilde{\psi}_M = (\xi_M, \xi_M)_{\mathcal{G}} \psi_M$$

(4.32)

$$\left( B_{M,M}^{(\mathcal{V}, \mathcal{A}_\pi)} \right)^\dagger \tilde{\xi}_M = (\psi_M, \psi_M)_{\mathcal{V}, \mathcal{A}_\pi} \xi_M$$

(4.33)

which we solve to find $\tilde{\psi}_M$ and $\tilde{\xi}_M$ respectively. Conveniently, the operators to be inverted here are the adjoints (denoted by the dagger $\dagger$) of those in the progressive problems, Equations 5.8 and 5.11. In fact, if the progressive problem is solved directly—as we do for Equation 5.11—rather than iteratively, one need not even factorize the adjoint matrix, since $\left( B^\top \right)^{-1} = (B^{-1})^\top$ for any invertible matrix $B$.

### 4.4.6 Progressive PGD

In progressive applications of PGD, the reduced basis is constructed one mode at a time. That is to say, one begins with zero modes, and progressively “enriches” the basis
(add additional modes) until convergence is reached. The latest mode $M$ is discovered by solving the nonlinear system of (two) equations formed by Equations 4.18 and 4.27. Here, as is typical in PGD literature, this system is resolved by fixed-point iteration—that is, solving each equation in sequence, iterating until convergence. (This ubiquitous procedure is also termed Alternating Directions, Power, or Picard Iteration in other PGD works.) We summarize this procedure of progressive PGD in Algorithm 1. Note that the residual norms referred to therein are more precisely defined as $\|\tilde{r}_{m+1}\|_{L^2}/\|\tilde{r}_m\|_{L^2}$ where $m$ is as stated in line 2, and likewise $\tilde{r}_m$ as in Section 4.4.3 or 4.4.4 as appropriate. The spatio-angular mode $\psi_M$ is normalized in line 5 for numerical stability.

**Algorithm 1: Galerkin or Minimax Progressive PGD**

1. $\psi \leftarrow 0$
2. for $m = 1 \ldots M$ do  // enrichment iteration
3.   $\psi_m, \bar{\psi}_m \leftarrow 1$
4. while residual $> \text{tolerance}$ do  // nonlinear iteration
5.   $\psi_m, \bar{\psi}_m \leftarrow \psi_m/\|\psi_m\|_{L^2}, \bar{\psi}_m/\|\psi_m\|_{L^2}$
6.   residual $\times_i$ $\leftarrow$ relative $\ell^2$ norm of residual of Equation 4.27
7.   $\xi_m \leftarrow$ solution of Equation 4.27
8.   if minimax then solution of Equation 4.33 else $\xi_m$
9.   residual $\psi$ $\leftarrow$ relative $\ell^2$ norm of residual of Equation 4.18
10.  $\psi_m \leftarrow$ solution of Equation 4.18
11.  if minimax then solution of Equation 4.32 else $\psi_m$
12. residual $\leftarrow \sqrt{\text{residual} \times_i^2 + \text{residual} \psi^2}$
13.  $\psi \leftarrow \psi + \psi_M \times \xi_M$

4.4.7 Progressive PGD with Update

While the progressive PGD is viable, it requires that each mode $M$ is solved only with knowledge of prior modes $m < M$. If instead, all modes were computed simultaneously, as by

$$\sum_{m=1}^{M} B_{m,m}^{(g)} \psi_m = \tilde{r}_m^{(g)} \quad \forall m^* = 1 \ldots M,$$

$$\sum_{m=1}^{M} B_{m,m}^{(H,4\pi)} \xi_m = \tilde{r}_m^{(H,4\pi)} \quad \forall m^* = 1 \ldots M,$$

a better decomposition may be possible, but the computational cost could be prohibitive.
However, this cost would be dominated almost exclusively by the spatio-angular Equation(s) 4.34 for reactor physics. For example, a full-core problem could easily contain millions of spatio-angular unknowns, but only a dozen to several hundred energy groups. Since solving for all modes effectively multiplies the size of each linear system by $M$, this suggests solving for all energetic modes simultaneously may be feasible, even if the same does not apply for the spatio-angular modes.

To exploit this, we amend progressive PGD to “update,” or recompute, all energetic modes—by solving Equation 4.35—following each enrichment step (the discovery of another pair of modes $M$). While we only consider energy, this update step could also be applied in any other dimension, time being the canonical example in literature [9], [174]. We detail the Progressive PGD with Update in Algorithm 2.

**Algorithm 2: Minimax or Galerkin Progressive PGD with Update**

1. for $m = 1 \ldots M$
   2. steps 2 to 11 of Algorithm 1
   3. $\{\xi_{m^*}\}_{m^*=1}^M \leftarrow$ solution of Equation 4.35
   4. $\psi \leftarrow \sum_{m=1}^M \psi_m \times \xi_m$

Notably, the update equation can be seen as a generalization of the progressive equation, as they are equivalent except that the former is repeated for all modes ($m^* = 1 \ldots M$) instead of only the last ($m^* = M$). Analogously, expressing the update problem (Equation 4.35) as a block system

$$\widetilde{B}\xi = \tilde{\chi}$$

where $\widetilde{B}$ is a a block operator and $\xi$ and $\tilde{\chi}$ are block vectors,

$$\widetilde{B} = \begin{bmatrix}
B_{1,1}^{(V,4\pi)} & \cdots & B_{1,M}^{(V,4\pi)} \\
\vdots & \ddots & \vdots \\
B_{M,1}^{(V,4\pi)} & \cdots & B_{M,M}^{(V,4\pi)}
\end{bmatrix}, \quad \xi \equiv \begin{bmatrix}
\xi_1 \\
\vdots \\
\xi_M
\end{bmatrix}, \quad \tilde{\chi} \equiv \begin{bmatrix}
\tilde{\chi}_1 \\
\vdots \\
\tilde{\chi}_M
\end{bmatrix}$$

makes clear the simple relationship between Equations 4.27 and 4.35: the former corresponds to the last block row of the latter. Further, although not applied in this chapter, Section 5.4.6 describes an efficient means of directly factorizing $\widetilde{B}$ by recursive LU decomposition.
4.4.8 Multigroup Discretization

Following the discussion of Section 4.4.4 we now discretize our energetic equation using the multigroup approximation. That is to say, we have group-wise unknowns

\[ \xi_{m,g} \equiv \int_{E_g}^{E_{g-1}} \xi_m(E) \, dE \]  

(4.38)

and multigroup cross sections

\[ \tilde{\Sigma}_{t,g,m^*,m} \equiv \sum_{j=1}^{J} \left( \tilde{\psi}_{m^*}, N_j \psi_m \right)_{\mathcal{V},4\pi} \sigma_{t,j,g} + \left( \tilde{\psi}_{m^*}, \vec{\Omega} \cdot \nabla \psi_m \right)_{\mathcal{V},4\pi}, \]  

(4.39)

\[ \tilde{\Sigma}_{s,t,g' \rightarrow g,m^*,m} \equiv \sum_{j=1}^{J} \frac{2\ell + 1}{4\pi} \left( \tilde{\psi}_{m^*}, N_j \int_{4\pi} P_{\ell} \psi \, d\Omega' \right)_{\mathcal{V},4\pi} \sigma_{s,j,t,g' \rightarrow g}, \]  

(4.40)

in each energy group \( g, [E_g, E_{g-1}] \). As is convention, we assume the spatial dependence of \( N_j \) is piecewise-constant and that the number density is actually multiplied by the cross section values in advance. Doing this, \( N_j \) becomes essentially an indicator function

\[ N_j(\vec{r}) \leftarrow \mathbf{1}_j(\vec{r}) \equiv \begin{cases} 
1, & \vec{r} \in \mathcal{V}_j, \\
0, & \text{otherwise},
\end{cases} \]  

(4.41)

where material regions \( \mathcal{V}_j \) are comprised of whole elements and do not overlap. Meanwhile, for the source, we can now modify our expansion to be slightly more intuitive in cases where a natural decomposition is not apparent. Specifically, let

\[ q(\vec{r}, \vec{\Omega}, E) = \sum_{i=1}^{I} q_i(\vec{r}, \vec{\Omega}) \tilde{\chi}_i(E) + \sum_{g'=1}^{G} q_{g'}(\vec{r}, \vec{\Omega}) \tilde{1}_{g'}(E), \]  

(4.42)

where

\[ \tilde{1}_{g'}(E) \equiv \begin{cases} 
1/(E_{g'-1} - E_{g'}), & E \in [E_{g'}, E_{g'-1}] \\
0, & \text{otherwise}
\end{cases} \]  

(4.43)

which integrates in energy over \( [E_g, E_{g-1}] \) to the Kronecker delta \( \delta_{g,g'} \) such that we recover the ordinary definition of the group sources

\[ q_g(\vec{r}, \vec{\Omega}) = \int_{E_g}^{E_{g-1}} q(\vec{r}, \vec{\Omega}, E) \, dE \]  

(4.44)
and can write
\[ \tilde{r}_{M,g} = \int_{E_g}^{E_{g-1}} \tilde{r}(E) dE \]
\[ = \left( \tilde{\psi}_M, q_g \right)_{\nu,4\pi} - \sum_{m=1}^{M-1} \left( \tilde{\Sigma}_{t,g,M,m} \xi_{m,g} - \sum_{\ell=0}^{L} \sum_{g'=1}^{G} \tilde{\Sigma}_{s,\ell,g'\rightarrow g,M,m} \xi_{m,g'} \right). \] 

(4.45)

Combining these terms, we arrive at a system of \( G \) coupled equations for each group \( g = 1 \ldots G \)
\[ \tilde{\Sigma}_{t,g,M,M} \xi_{M,g} - \sum_{\ell=0}^{L} \sum_{g'=1}^{G} \tilde{\Sigma}_{s,\ell,g'\rightarrow g,M,M} \xi_{M,g'} = \tilde{r}_{M,g} \] 

(4.46)

or, more succinctly
\[ \left( \Sigma_{t,M,m} - \tilde{\Sigma}_{s,M,m} \right) \xi_{M} = \tilde{r}_{M} \]

(4.47)

where \( \tilde{\Sigma}_{t,M,M} \equiv \text{diag}(\tilde{\Sigma}_{t,1,M,M}, \ldots, \tilde{\Sigma}_{t,g,M,M}, \ldots, \tilde{\Sigma}_{t,G,M,M}) \) is the collision matrix, \( \xi_{M} \equiv [\xi_{M,1} \ldots \xi_{M,g} \ldots \xi_{M,G}] \) is the solution vector (or spectrum), \( \tilde{r}_{M} \equiv [\tilde{r}_{M,1} \ldots \tilde{r}_{M,g} \ldots \tilde{r}_{M,G}] \) is the residual vector, and
\[ \tilde{\Sigma}_{s,M,M} \equiv \sum_{\ell=0}^{L} \left[ \begin{array}{cccc}
\tilde{\Sigma}_{s,\ell,1\rightarrow 1,M,M} & \ldots & \tilde{\Sigma}_{s,\ell,g'\rightarrow 1,M,M} & \ldots & \tilde{\Sigma}_{s,\ell,G\rightarrow 1,M,M} \\
\vdots & \ddots & \vdots & & \vdots \\
\tilde{\Sigma}_{s,\ell,1\rightarrow g,M,M} & \tilde{\Sigma}_{s,\ell,g'\rightarrow g,M,M} & \tilde{\Sigma}_{s,\ell,G\rightarrow g,M,M} & & \vdots \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
\tilde{\Sigma}_{s,\ell,1\rightarrow G,M,M} & \tilde{\Sigma}_{s,\ell,g'\rightarrow G,M,M} & \tilde{\Sigma}_{s,\ell,G\rightarrow G,M,M}
\end{array} \right] \]

(4.48)

is the scattering matrix.

### 4.4.9 Energetic Inner Products

Looking at the energetic inner products present in Equation 4.11, it is clear we must define a rule for computing terms of the form
\[ \left( \xi_{m*}, \bullet \right)_g. \]

(4.49)

While the multigroup discretization only defines integral degrees-of-freedom \( \xi_{m,g} \), it is obvious we can relate this to a differential (average) value \( \bar{\xi}_{m,g} \) as in Equation 3.25 by letting
\[ \xi_{m,g} = \bar{\xi}_{m,g} \Delta_g, \]

(4.50)
where \( \Delta_g \equiv \ln(E_{g-1}/E_g) \) is the lethargy width of group \( g \). It follows then, that we have coefficients

\[
(\xi_{m^*}, \xi_m) = \sum_{g=1}^{G} \Delta_g \bar{\xi}_{m^*,g} \bar{\xi}_{m,g} = \sum_{g=1}^{G} \Delta_g^{-1} \xi_{m^*,g} \xi_{m,g} \quad (4.51)
\]

and likewise

\[
(\xi_{m^*}, \sigma_{t,j} \xi_m) = \sum_{g=1}^{G} \Delta_g^{-1} \xi_{m^*,g} \sigma_{t,j,g} \xi_{m,g},
\]

\[
(\xi_{m^*}, \int_{E_G}^{E_0} \sigma_{s,t,j} \xi_m dE') = \sum_{g=1}^{G} \Delta_g^{-1} \xi_{m^*,g} \sum_{g'=1}^{G} \sigma_{s,t,j,g' g} \xi_{m,g'},
\]

\[
(\xi_{m^*}, q_i) = \sum_{g=1}^{G} \Delta_g^{-1} \xi_{m^*,g} q_{i,g}.
\]

### 4.4.10 Spatio-angular Inner Products and Caching

Let us consider now the spatio-angular inner products required by Equation 4.21. As a benefit of employing a Galerkin FEM in space and discrete ordinates in angle, one can compute these quantities using our established spatial operators—particularly, (omitting subscript \( G \) for brevity) streaming and mass matrices \( G \) and \( B \)—and angular quadrature. Summing up over each discontinuous element \( K \) (denoting, as a subscript, the corresponding block of a vector or matrix),

\[
(\bar{\psi}_{m^*}, \Omega \cdot \nabla \psi_m)_{V,4\pi} = \sum_{n=1}^{N} w_n \bar{\psi}_{m^*,n}^\top G_n \psi_{m,n},
\]

\[
(\bar{\psi}_{m^*}, N_j \psi_m)_{V,4\pi} = \sum_{n=1}^{N} w_n \sum_{K} N_{j,K} \bar{\psi}_{m^*,n,K}^\top B_{N,K} \psi_{m,n,K},
\]

\[
(\bar{\psi}_{m^*}, N_j \frac{2\ell + 1}{4\pi} \sum_{k=-\ell}^{\ell} Y_{\ell,k} \phi_{\ell,k})_{V,4\pi} = \sum_{n=1}^{N} w_n \sum_{K} N_{j,K} \bar{\psi}_{m^*,n,K}^\top B_{N,K} \left[ M_{\ell,\phi_{m,\ell}} \right]_{n,K},
\]

where \( N_{j,K} \) is the constant value of \( N_j \) over each cell. Clearly, applying these operators repeatedly would be expensive, but since, for progressive PGD without update in space-angle, \( \psi_m \) is fixed for \( m < M \), we can cache these matrix-vector products, leaving only a dot product, the left-multiplication by \( \psi_{m^*,} \), to evaluate.

As a compromise between computation and memory costs, we choose to store \( D \psi_m \equiv \Phi_m \) since this vector is small and physically meaningful and \( G \psi_m \) since the streaming operator is the most expensive to compute. Because these terms also appear in Equation
4.11 (and the separated residual, Equation 4.77, if used) we also avoid those evaluations by this pre-computation.

4.5 Numerical Results

In this section we assess the performance of the Galerkin PGD ROM on two problems: the C5G7 UO$_2$ pin-cell [175] and the un-condensed (Cathalau) UO$_2$ and MOX pin-cells [176], plus two lattices thereof. The former uses the specified multigroup cross sections while representative lattice-physics libraries were generated for the latter using OpenMC [177]. Additionally, Minimax results for the latter are provided in Appendix E.2. In all scenarios, we find the PGD ROM performs well.

4.5.1 Numerical Sundries

Each numerical experiment is performed with an $S^4_8$ product Gauss-Legendre-Chebyshev angular quadrature [178]: that is, with 4 polar and 8 azimuthal angles per quadrant, 128 angles in total. Unless otherwise noted, linear ($p = 1$), discontinuous Lagrangian finite elements are used, meaning each quadrilateral element has 4 spatial degrees-of-freedom. Both the spatio-angular equation for PGD and the within-group problems of the full-order model are solved using the Generalized Minimal Residual (GMRES) method [154]. The former is terminated when the residual—more precisely, the $\ell^2$ norm thereof—falls below $10^{-8}$ or after 25 iterations, while the latter is restarted after the Krylov subspace reaches a maximum size of 30 and terminated when the residual falls below $10^{-10}$ or $10^{-2}$ times the initial residual. The full-order model is solved by block Gauss-Seidel preconditioned Richardson iteration, terminated when the residual falls below $10^{-8}$. The progressive energetic equation is inverted directly by Gauss-Jordan elimination, while the update in energy is accomplished by block Gauss-Seidel preconditioned GMRES, where the diagonal ($G \times G$) blocks are directly inverted. This GMRES loop also restarts after the Krylov subspace is of size 30 and continues until the residual is below $10^{-6}$ or 1000 iterations have elapsed, after which the answer is accepted even if convergence is not reached. The tolerance for the nonlinear block Gauss-Seidel solver of PGD is set as $10^{-4}$ (or $10^{-8}$ in Section 4.5.3.1) and the maximum number of iterations as 50. Non-convergence—reaching the iteration limit without meeting the tolerance—is found to be not uncommon, which may degrade the efficacy of PGD. A more robust nonlinear solver than block Gauss-Seidel could therefore improve the following results and presents an
area for future research.

As detailed in Algorithm 1, the initial guess for $\psi_M$ is always isotropically and uniformly set to be of unit $L^2$ norm. Under these conditions, as per our comments in Section 4.4.4, the energetic equation effectively simplifies to that of a homogeneous, infinite medium, giving an intuitive (and, hopefully, advantageous) meaning to this initialization. Given the spatio-angular PGD equation is solved iteratively, the initial solution vector is set to be that from the previous nonlinear iteration. Likewise, the initial guess for the solution vector of the energy-update—which solves for all energy modes—is set to the current state of the modes. The implementation itself is written in object-oriented C++ using version 9.1 of the deal.II finite element library [179], [180].

4.5.2 Computational Costs

Given the relative number of spatio-angular (here, about 100,000) to energetic (less than 400) degrees-of-freedom, we expect the spatio-angular problems to dominate the computational cost in lattice physics applications like those we show here.\footnote{This may be challenged in some cases by the update step, which with 361 groups and 50 modes grows to 18,050 unknowns. That said, this cost could likely be abated if necessary—perhaps by updating less frequently or only updating the last twenty or so modes.} Since these problems are tantamount to the within-group problems of the block Gauss-Seidel in energy scheme—which is ubiquitously used to solve or precondition the full-order model—the question roughly becomes which model requires more solutions to these problems. For the full-order model, the required number is at least $G$, multiplied by the average number of iterations to account for upscattering $N_{up}$. We say average in case the energy scale is partitioned into larger blocks (comprised of several groups), so as to avoid iterating excessively on groups with little to no upscattering. Meanwhile, in the PGD ROM, $M$ spatio-angular problems must be solved, multiplied by the average number of nonlinear iterations per mode $N_{nl}$. One might expect $N_{nl} > N_{up}$ since there is little upscattering outside of thermal energies. This is an imperfect comparison, as it neglects all other costs and does not consider that each within-group problem may require a different number of iterations—depending, for instance, on the quality of the initial guess or the relative amount of within-group scattering. Nevertheless, the operative insight is that the advantage of the PGD ROM can be estimated roughly by $(GN_{up})/(MN_{nl})$. Moreover, improvements to the nonlinear solver, in so far as they reduce
That said, we do not attempt to measure the runtime of our models here, because neither has been optimized for performance. Additionally, we have set the tolerances to be tight and the iteration limit to be high in both the full-order model and the nonlinear loop of the ROM to avoid iterative error from influencing our results—in practice, such stringent requirements may be unnecessary. Finally, we have neglected practical considerations such as diffusion preconditioning (for both models), adaptive tolerances in the enrichment and nonlinear loops of the ROM, and perhaps a more sophisticated nonlinear solver (like a Newton method). Addressing such concerns before comparing the relative runtimes is advisable. As such, we aim only to empirically establish that the ROM converges and characterize a typical $M$ to achieve a given accuracy, with the expectation that the runtime will scale at least linearly with $M$.

### 4.5.3 C5G7 UO$_2$ Pin-Cell

We begin our inquiry with a well-known benchmark for neutron transport, the C5G7. In order to keep the simulations computationally inexpensive, and to reflect our intended application of pin-cell- and lattice-physics, we restrict the domain to a single quarter UO$_2$ pin with reflecting boundary conditions, sketched in Figure 4.5.

![Figure 4.5: C5G7 UO$_2$ pin-cell, 5:1 scale.](image)

The first objective is simply verifying both the full-order model and the PGD ROM are implemented correctly. To do so, we measure the spatial order of convergence with increasing mesh refinement. Having spatially discretized by DGFEM, we expect the order to be $p + 1$ where $p$ is the polynomial order of the elements. (This order is not guaranteed, but is frequently found in practice [151].) To do so, we here employ the Method of Manufactured Solutions (MMS).
4.5.3.1 Method of Manufactured Solutions

As we first wish only to verify the PGD implementation, not validate its performance, we will use MMS to construct a problem whose solution can be expressed exactly in a single mode and therefore can be obtained by a single enrichment iteration. So, we select the solution

\[ \psi_{g,\text{ref}}(x, y, \tilde{\Omega}) = \cos(ax) \cos(ay) \times g, \quad (4.58) \]

where \( a = \pi \times (1.26/2) \), which corresponds to the 1.26 cm pitch of the lattice (the side lengths of the square pin-cell). Doing so, we arrive at a separated source of

\[ q_g(x, y, \tilde{\Omega}) = q^\text{stream}_g(x, y, \tilde{\Omega}) + \sum_{j=1}^{J} \left( q^\text{collide}_{j,g}(x, y, \tilde{\Omega}) + q^\text{scatter}_{j,g}(x, y, \tilde{\Omega}) \right), \quad (4.59) \]

where

\[ q^\text{stream}_g(x, y, \tilde{\Omega}) \equiv \left( -\tilde{\Omega}_x a \sin(ax) \cos(ay) - \tilde{\Omega}_y a \cos(ax) \sin(ay) \right) \times g, \quad (4.60) \]

\[ q^\text{collide}_{j,g}(x, y, \tilde{\Omega}) \equiv N_j(x, y) \cos(ax) \cos(ay) \times \sigma_{t,j,g} g, \quad (4.61) \]

\[ q^\text{scatter}_{j,g}(x, y, \tilde{\Omega}) \equiv N_j(x, y) \cos(ax) \cos(ay) \times \sum_{g'=1}^{G} \sigma_{s,j,g' \rightarrow g} g'. \quad (4.62) \]

As an aside, the collision and scattering sources could actually be combined in this example because the spatio-angular dependence is the same. This will not be true if the solution is anisotropic. Regardless, we solve this problem with both the full-order and PGD ROM for \( p = 0, 1, 2 \) (constant, linear, and quadratic finite elements), the convergence of which is plotted in Figure 4.7. Both models demonstrate the expected \( p + 1 \) order convergence, verifying the implementation such that we can continue to problems without known solutions, for which we employ the twice-refined mesh containing 81,920 spatio-angular degrees-of-freedom.
4.5.3.2 Uniform Fission Source

While the C5G7 benchmark is defined as a criticality (eigenvalue) problem, our aim in developing the PGD ROM is first to assess the performance on a fixed-source problem. To create a comparable simulation, we fix our separable source as

\[ q(\vec{r}, \Omega, E) \leftarrow \chi(E)F(\vec{r}), \]

where \( \chi \) is the emission spectrum of fission neutrons and \( F \) is the rate of neutron production by fission. This emulates a criticality problem, differing only in the sense that, for criticality,

\[ F(\vec{r}) \leftarrow F(\vec{r}) \]

\[ \frac{1}{4\pi k} \int_{E_G}^{E_0} \nu \Sigma_f(\vec{r}, E') \int_{4\pi} \psi(\vec{r}, \Omega', E') d\Omega' dE', \]

where the flux \( \psi \) and eigenvalue \( k \) are often taken from a previous criticality iteration and \( \nu \Sigma_f \) is the neutron yield times the fission cross section, while we here set uniformly

\[ F(\vec{r}) \leftarrow \begin{cases} 1, & \vec{r} \in V_{\text{fuel}}, \\ 0, & \text{otherwise}, \end{cases} \]

where \( V_{\text{fuel}} \) is the fueled region, corresponding to a single material. In this sense, this fixed-source problem can effectively be thought of as the first iteration of a criticality calculation. This assumption is also commonly invoked for ultrafine-group simulations, where iterating
on a fission source could be prohibitive [165]. This source could be generalized to multiple fuel materials simply by writing

\[ q(\vec{r}, \vec{\Omega}, E) \leftarrow \sum_{j=1}^{J} \chi_j(E) F_j(\vec{r}) \],

(4.66)

where \( \chi_j \) will of course be zero if material \( j \) is not fissionable, allowing the term to be omitted, but there is no need in the present case.

### 4.5.3.3 Reduced-Order Error Indicators

We assess the convergence of our PGD ROM relative to the full-order model in the \( L^2 \) norm in Figure 4.8 by three measures. In each case, the square of the \( L^2 \) norm is defined and discretized as

\[ \|\psi\|_{L^2}^2 \equiv \int_{u_0}^{u_G} \int_{4\pi} \int_{V} \psi^2 drd\Omega du \left\langle \sum_{g=1}^{G} \Delta_g^{-1} \sum_{n=1}^{N} w_n \psi_{g,n}^T B_{g,N} \psi_{g,n} \right. \]

(4.67)

Firstly, the error describes the difference between the full-order model and PGD ROM. Given both models use the same discretization (mesh, quadrature, and group structure) this difference is due only to the PGD approximation (the truncation of modes beyond \( M \)). Since simulation practitioners will often be interested in the flux moments \( \phi \) rather than the angular flux \( \psi \), we also separately measure this error. Furthermore, as the SVD provides a provably optimal decomposition of a known solution, we also compare to the error of this ideal decomposition of the full-order flux in the \( L^2 \) norm, as per Appendix D, taking as many singular vectors as PGD modes. However, all these metrics require knowledge of the full-order solution, which will usually not be available to the PGD simulation, as it defeats the purpose.

Accordingly, we also measure the residual \( r \), which is achieved by substituting the solution back into Equation 4.8 (discretized as in Equation 3.44) and subtracting the left-hand-side from the right-hand-side. Written as matrices,

\[ Br \equiv Bq - (L - BM\Sigma_s D) \psi , \]

(4.68)

or solving for \( r \)

\[ r = q - \left( B^{-1}L - M\Sigma_s D \right) \psi , \]

(4.69)
However, this is not the usual definition of the residual, in that for full-order calculations one usually multiplies Equation 3.44 by $\mathbf{L}^{-1}$ before solving iteratively. This would yield the “swept residual”

$$\mathbf{L}^{-1}\mathbf{Br} = \mathbf{L}^{-1}\mathbf{Bq} - \left(\mathbf{I} + \mathbf{L}^{-1}\left(\mathbf{L} - \mathbf{BM}\Sigma_s\mathbf{D}\right)\right) \psi$$

(4.70)

This distinction is not without significance, since for any linear operator $\mathbf{A}$ where $\mathbf{Ax} = \mathbf{b}$, the residual and error are defined as $\mathbf{r} \equiv \mathbf{b} - \mathbf{Ax}$ and $\mathbf{e} \equiv \mathbf{A}^{-1}\mathbf{b} - \mathbf{x}$ respectively. From this, it is clear that $\mathbf{A}\mathbf{e} = \mathbf{r}$, or in our case,

$$(\mathbf{L} - \mathbf{BM}\Sigma_s\mathbf{D})\mathbf{e} = \mathbf{Br}$$

(4.71)

and likewise

$$\left(\mathbf{I} + \mathbf{L}^{-1}\left(\mathbf{L} - \mathbf{BM}\Sigma_s\mathbf{D}\right)\right)\mathbf{e} = \mathbf{L}^{-1}\mathbf{Br}$$

(4.72)

which can be rearranged

$$\mathbf{L}^{-1}\mathbf{Br} = \mathbf{e} + \mathbf{L}^{-1}\left(\mathbf{L} - \mathbf{BM}\Sigma_s\mathbf{D}\right)\mathbf{e}$$

$$\approx \mathbf{e}$$

(4.73)

from which we can infer that the “swept” residual may be a better approximation of the error than the residual itself, which is borne out by numerical experiments. But $\mathbf{L}$, being a summation of separated operators, is separable while the inverse $\mathbf{L}^{-1}$ is not and so cannot be evaluated in a PGD simulation without degrading the model order. As a compromise, let us invert only $\mathbf{G}$, the streaming, or sweeping without collision, operator, to achieve the “streamed residual”

$$\mathbf{G}^{-1}\mathbf{Br} = \mathbf{e} + \mathbf{G}^{-1}\left(\mathbf{G} + \mathbf{B}(\Sigma_t - \mathbf{M}\Sigma_s\mathbf{D})\right)\mathbf{e}$$

(4.74)

where, given that $\mathbf{I}$ is the $G \times G$ identity matrix and $\mathbf{GB} = \mathbf{I} \otimes \mathbf{G}_g\mathbf{B}_g$, it follows that $\mathbf{G}^{-1}\mathbf{B} = \mathbf{I} \otimes \mathbf{G}_g^{-1}\mathbf{B}_g$. Of course, given our discrete separated solution

$$\psi \equiv \sum_{m=1}^{M} \xi_m \otimes \psi_m$$

(4.75)
and source
\[ q \equiv \sum_{i=1}^{I} \bar{\chi}_i \otimes q_i \quad (4.76) \]
we find our separated residual—having inferred the notation of our discretized energy equation from Sections 4.4.4 and 4.4.8—as
\[ r = \sum_{i=1}^{I} \bar{\chi}_i \otimes q_i - \sum_{m=1}^{M} \left( \xi_m \otimes B_g^{-1} G_q \psi_m \right. \]
\[ + \sum_{j=1}^{J} \sigma_{t,j} \xi_m \otimes N_{g,j} \psi_m \left. + \sum_{j=1}^{J} \sum_{\ell=0}^{L} \sigma_{s,\ell} \xi_m \otimes N_{g,j} M_{G,\ell} D_{G,\ell} \psi_m \right) \quad (4.77) \]
from which it is clear we must distribute \( G^{-1} B \) to retain our separation
\[ G^{-1} B r = \sum_{i=1}^{I} \bar{\chi}_i \otimes G_g^{-1} B_q q_i \]
\[ - \sum_{m=1}^{M} \left( \xi_m \otimes G_g^{-1} G_q \psi_m \right. \]
\[ + \sum_{j=1}^{J} \sigma_{t,j} \xi_m \otimes G_g^{-1} B_q N_{g,j} \psi_m \left. + \sum_{j=1}^{J} \sum_{\ell=0}^{L} \sigma_{s,\ell} \xi_m \otimes G_g^{-1} B_q N_{g,j} M_{G,\ell} D_{G,\ell} \psi_m \right) \quad (4.78) \]
meaning the action of \( G_g^{-1} \) must be evaluated \( I + M \times (3 + L) \) times, assuming the spatio-angular modes \( \psi_m \) are not updated throughout enrichment and all multiplications by \( G_g^{-1} \) are cached. In the present implementation, since \( G_g \psi_m \) and \( D_g \psi_m \) are already cached (see Section 4.4.10) the residual is relatively cheap to compute, though applying \( G_g^{-1} \) would obviously incur an additional computational cost.\(^9\)

Lastly, we measure the \( L^2 \) norm of the most recent mode. This is an appropriate indicator because one can assume (provided the series is converging quickly enough)
\[ \sum_{m=1}^{M-1} \psi_m(\vec{r}, \vec{\Omega}) \xi_m(E) - \psi_{\text{ref}}(\vec{r}, \vec{\Omega}, E) = \sum_{m=M}^{\infty} \psi_m(\vec{r}, \vec{\Omega}) \xi_m(E) \]
\[ \approx \psi_M(\vec{r}, \vec{\Omega}) \xi_M(E), \quad (4.79) \]
so the \( L^2 \) norm of this latest pair of modes \( M \) should approximately equal the \( L^2 \) norm of the

\(^9\)For simplicity, we do not distribute \( G^{-1} B \) in our implementation, since we have yet to optimize for performance.
approximation error from truncating the series at $M - 1$ modes. Practically, one would use this metric in a “lagged” sense, to determine whether to truncate at $M$, not $M - 1$, modes, so it may be slightly conservative. The advantage of this indicator is of course that it is less costly to compute than the residual.

**Figure 4.8: Convergence of progressive Proper Generalized Decomposition with and without update, C5G7 UO$_2$ pin-cell.**

From Figure 4.8, we see the swept or streamed residual and norm of the latest mode tend to agree with the true error for the C5G7 UO$_2$ pin-cell. The residual itself, however, is consistently higher than the error. Compared to the SVD results, it is evident neither PGD with or without update converges exactly in seven modes—or at all—despite that the full-order solution admits such a decomposition. However, it is also clear the PGD with update converges much more rapidly, quickly outperforming the non-updated case by over a decade, suggesting this update step to be effective. In any case, we find both models converge reasonably quickly, attaining relative $L^2$ errors less than $10^{-4}$ in 14 and 23 modes, with and without update respectively.

### 4.5.4 Cathalau UO$_2$ & MOX Pin-Cells

While these examples are instructive because of how well-known the C5G7 benchmark is, seven groups is far fewer than would be expected to make this PGD ROM attractive. Noting this, we now see the true power of this approximation: one can increase the energy fidelity with no influence on the spatio-angular cost. If the latter is dominant (or near-dominant) this energy fidelity is achieved practically “for free.” This is, of course, another way to say
we have achieved model order reduction with respect to energy. However, we must qualify this previous statement; if problems with finer groups require more modes to resolve, the spatio-angular cost and energy fidelity will still scale together, albeit at a reduced order. So, our next task is quantifying this relationship (or lack thereof) by numerical experimentation on a model problem.

![Figure 4.9: Cathalau pin-cell, 5:1 scale.](image)

Specifically, we select the infinite lattice UO$_2$ and 4.3% enriched MOX pin-cell problems, sketched in Figure 4.9, from which the seven-group cross sections of the C5G7 were condensed, referred to here as the Cathalau pin-cells. While the authors of the C5G7 employed a WIMS-69 group library for this calculation [175], we wish to characterize the performance with multiple energy fidelities, and so select the CASMO-70 [181], XMAS-172 [182], [183], SHEM-361 [184], [185] structures, specifically designed for light water reactor analysis. Cross sections in these structures, shown in Figure 4.10, were condensed by modeling the continuous-energy criticality problem in OpenMC [177], [186]. As before, the fission source is assumed uniform within the fuel region. The mesh is displayed in Figure 4.9 and contains 224 elements, and therefore 896 spatial and 114,688 spatio-angular degrees-of-freedom in total.

From Figures 4.11 and 4.12 and Table 4.1, we see first and foremost that PGD is convergent for each problem, achieving a normalized $L^2$ error less than 0.36% with 50 modes even in the worst case, SHEM-361 without update. Moreover, the update step in energy is again seen to be effective, reducing the final error by about a factor of ten in many cases. Comparing UO$_2$ and MOX fuels, we find these errors similar for the SHEM-361 structure (3.84e-04 and 5.02e-04 respectively, with update) while somewhat lower for UO$_2$ in the CASMO-70 and XMAS-172 problems (3.03e-05 and 9.28e-05 versus 1.18e-04 and 2.34e-04

---

10Cross sections with transport-corrected $P_0$ scattering were tallied using ENDF/B-VIII.0 nuclear data [187], given 100,000 particles per cycle, 50 inactive cycles, and 3950 active cycles. Neutron multiplication by $(n,xn)$ reactions is neglected.
Figure 4.10: Fine-group Cathalau cross sections by fuel and group structure.

in MOX). The same holds for the SVD results, the UO\(_2\) errors in increasing fidelity being 1.41e-07, 3.88e-06, and 3.36e-05, compared to those of MOX, 1.68e-06, 1.12e-05, and 4.48e-05. This illustrates the problem-specific performance of PGD, in that the rate of convergence depends on the inherent separability of the full-order solution, which can be quantified by the SVD. The spatial and energy distributions of these PGD with update and SVD errors are discussed in Appendix E.3. Additionally, these Galerkin results are compared to those of the Minimax PGD in Appendix E.2.

Examining across energy meshes, we find the MOX errors with update predictably do increase somewhat with fidelity, from 1.18e-04, 2.34e-04, to 5.02e-04. Inspecting the number of groups, a 2.46\(\times\) increase in resolution, from CASMO-70 to XMAS-172, is accompanied by a 1.98\(\times\) increase in error. Likewise, from XMAS-172 to SHEM-361, the corresponding factors are 2.10 and 2.15. This is not to claim one can predict the PGD error based on the energy mesh, but—given the primary benefit of this approximation is the model order reduction with respect to energy—it behooves us to investigate whether finer solutions (at least within the narrow range of resolution typically invoked for lattice physics) might require more modes,
degrading this advantage. An important area of future work will be investigating the same for very- to ultra-fine resolutions of 1,000 to 100,000 groups. Within the limited scope of our pin-cell analysis, we find this effect to be evident, but slight enough that SHEM-361 analyses remain eminently feasible.

Finally, we observe that the error of the scalar flux does tend to be slightly lower than that of the angular flux due to cancellation of error and that this effect appears more pronounced in the results with update. This is to be expected, as the error in energy is mitigated by this update, such that the error is driven in greater proportion by spatio-angular discrepancies. Accordingly, the solution benefits to a greater extent from cancellation of error

Figure 4.11: Convergence of Proper Generalized Decomposition with and without update, by group structure, Cathalau UO$_2$ pin-cell.
when integrated in the angular domain.

4.5.4.1 Cross Section Generation

Given the utility of our PGD ROM for cross section generation, we now turn to measuring the truncation error in relation to these quantities specifically. Having demonstrated the efficacy of the update step, we exclusively consider the PGD with update in the following studies. From the discussion of [188] (originally derived in [147]), but with generalized geometry and the notation of [165], the transport-corrected equation for coarse groups $\bar{g} \in [1, \ldots, \bar{G}]$
Table 4.1: Normalized $L^2$ errors of progressive PGD with and without update, by fuel and group structure, Cathalau pin-cells.

<table>
<thead>
<tr>
<th>Fuel</th>
<th>Structure</th>
<th>Flux</th>
<th>Algorithm</th>
<th>PGD, $M =$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>CASMO-70</td>
<td>Angular</td>
<td>Prog.</td>
<td>1.06e-02</td>
<td>3.04e-03</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Update</td>
<td>1.27e-03</td>
<td>2.86e-04</td>
</tr>
<tr>
<td></td>
<td>Scalar</td>
<td>Prog.</td>
<td>8.41e-03</td>
<td>2.51e-03</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Update</td>
<td>5.44e-04</td>
<td>1.18e-04</td>
</tr>
<tr>
<td>UO2</td>
<td>Angular</td>
<td>Prog.</td>
<td>1.24e-02</td>
<td>3.91e-03</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Update</td>
<td>2.20e-03</td>
<td>6.48e-04</td>
</tr>
<tr>
<td></td>
<td>Scalar</td>
<td>Prog.</td>
<td>9.85e-03</td>
<td>3.35e-03</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Update</td>
<td>6.88e-04</td>
<td>1.79e-04</td>
</tr>
<tr>
<td>XMAS-172</td>
<td>Angular</td>
<td>Prog.</td>
<td>2.60e-02</td>
<td>9.46e-03</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Update</td>
<td>7.31e-03</td>
<td>1.96e-03</td>
</tr>
<tr>
<td></td>
<td>Scalar</td>
<td>Prog.</td>
<td>1.92e-02</td>
<td>6.65e-03</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Update</td>
<td>4.09e-03</td>
<td>5.03e-04</td>
</tr>
<tr>
<td>SHEM-361</td>
<td>Angular</td>
<td>Prog.</td>
<td>1.15e-02</td>
<td>4.63e-03</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Update</td>
<td>3.98e-03</td>
<td>9.73e-04</td>
</tr>
<tr>
<td></td>
<td>Scalar</td>
<td>Prog.</td>
<td>6.23e-03</td>
<td>3.66e-03</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Update</td>
<td>2.05e-03</td>
<td>1.85e-04</td>
</tr>
<tr>
<td>CASMO-70</td>
<td>Angular</td>
<td>Prog.</td>
<td>1.37e-02</td>
<td>5.47e-03</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Update</td>
<td>5.34e-03</td>
<td>9.73e-04</td>
</tr>
<tr>
<td></td>
<td>Scalar</td>
<td>Prog.</td>
<td>8.48e-03</td>
<td>3.69e-03</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Update</td>
<td>2.42e-03</td>
<td>6.67e-04</td>
</tr>
<tr>
<td>MOX</td>
<td>Angular</td>
<td>Prog.</td>
<td>2.61e-02</td>
<td>8.96e-03</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Update</td>
<td>6.38e-03</td>
<td>2.19e-03</td>
</tr>
<tr>
<td></td>
<td>Scalar</td>
<td>Prog.</td>
<td>1.86e-02</td>
<td>6.18e-03</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Update</td>
<td>2.32e-03</td>
<td>6.67e-04</td>
</tr>
</tbody>
</table>

where $\tilde{G} < G$ is written as

\[
\tilde{\Omega} \cdot \nabla \psi_\tilde{g}(\vec{r}, \tilde{\Omega}) + \hat{\Sigma}_{t,\tilde{g}}(\vec{r}) \psi_\tilde{g}(\vec{r}, \tilde{\Omega}) \\
= \sum_{\ell=0}^{L} \frac{2\ell + 1}{4\pi} \sum_{k=-\ell}^{\ell} Y_{\ell,k}(\tilde{\Omega}) \sum_{\tilde{g}'} 1 \left[ \Sigma_{s,\tilde{g}' \rightarrow \tilde{g},\ell}(\vec{r}) - \left( \hat{\Sigma}_{t,\tilde{g},\ell}(\vec{r}) - \hat{\Sigma}_{t,\tilde{g}}(\vec{r}) \right) \delta_{\tilde{g},\tilde{g}'} \right] \phi_{g',\ell,k}(\vec{r})
\]
where, given group $\bar{g}$ is comprised of some sequence of fine-groups $g$ and the relationship between $\Sigma$ and $\sigma$ is as in Equations 4.5 and 4.6,

$$\sigma_{t,j,\bar{g},\ell} \equiv \frac{\int_{E_{\bar{g}}}^{E_{\bar{g}}-1} \sigma_{t,j}(E) \tilde{\xi}_{j,\ell}(E) dE}{\int_{E_{\bar{g}}}^{E_{\bar{g}}-1} \tilde{\xi}_{j,\ell}(E) dE}$$

$$= \frac{\sum_{g \in \bar{g}} \sigma_{t,j,g} \tilde{\xi}_{j,\ell,g}}{\sum_{g \in \bar{g}} \tilde{\xi}_{j,\ell,g}} \tag{4.81}$$

and likewise

$$\sigma_{s,j,\bar{g} \to \bar{g}',\ell} \equiv \frac{\int_{E_{\bar{g}}}^{E_{\bar{g}}-1} \int_{E'_{\bar{g} \to \bar{g}'}} \sigma_{s,j,\ell}(E' \to E) \tilde{\xi}_{j,\ell}(E') dE' dE}{\int_{E_{\bar{g}}}^{E_{\bar{g}}-1} \tilde{\xi}_{j,\ell}(E') dE'}$$

$$= \frac{\sum_{g \in \bar{g}} \sum_{g' \in \bar{g}'} \sigma_{s,j,g' \to g,\ell,\tilde{\xi}_{j,\ell,g'}}}{\sum_{g' \in \bar{g}'} \tilde{\xi}_{j,\ell,g'}} \tag{4.82}$$

which require a weighting spectrum $\tilde{\xi}_{j,\ell}$, here taken from a fine(r) multigroup problem. The zeroth-order moment, $\ell = 0$, is described by

$$\tilde{\xi}_{j,0,g} \equiv \int_{V_j} \int_{4\pi} \psi_g(\vec{r},\vec{\Omega}) d\Omega d\vec{r}, \tag{4.83}$$

and, supposing the reference solution $\psi_g$ does not depend on the azimuthal angle $\omega$, as in one-dimensional Cartesian or spherical geometry,

$$\tilde{\xi}_{j,\ell,g} \equiv \int_{V_j} \int_{-1}^{1} P_{\ell}(\mu) \psi_g(x,\mu) d\mu d\vec{r} = \int_{V_j} \phi_{g,\ell,0}(x) dx. \tag{4.84}$$

In general geometry, the meaning of $\tilde{\xi}_{j,\ell,g}$ is ambiguous for $\ell > 0$, since each Legendre polynomial of degree $\ell$ is expanded as a summation of $\ell + 1$ and $2\ell + 1$ nonzero terms in two- and three-dimensional Cartesian space, respectively. However, rendering cross sections dependent on this azimuthal order $k (= -\ell \ldots \ell)$ is considered cumbersome and memory-intensive, and so has historically been avoided. We here follow the approach of Stamm’ler and Abbate [159], which prescribes the $\ell = 1$, or current-weighted, spectrum to be that of the net current. Given that $\phi_{g,1,1}$, $\phi_{g,1,-1}$, and $\phi_{g,1,0}$ are the $x$-, $y$-, and $z$-components of neutron current, this dictates

$$\tilde{\xi}_{j,1,g} \equiv \int_{V_j} \sqrt{\sum_{k=-1}^{1} \phi_{g,1,k}^2(\vec{r}) dr}. \tag{4.85}$$
Finally, $\tilde{\Sigma}_{t,j,g}$, the transport cross section, is a free parameter whose value can be set to better account for anisotropic collision and scattering in the fine-group solution—that is, those truncated moments above $L$. In short, there is the Consistent-P approximation:

$$\tilde{\sigma}_{t,j,\tilde{g}} \equiv \sigma_{t,j,\tilde{g},0}, \quad (4.86)$$

the Inconsistent-P approximation:

$$\tilde{\sigma}_{t,j,\tilde{g}} \equiv \sigma_{t,j,\tilde{g},L+1}, \quad (4.87)$$

the Diagonal approximation:

$$\tilde{\sigma}_{t,j,\tilde{g}} \equiv \sigma_{t,j,\tilde{g},L+1} - \sigma_{s,\tilde{g} \rightarrow \tilde{g},L+1}, \quad (4.88)$$

the Bell-Hansen-Sandmeier (BHS) approximation:

$$\tilde{\sigma}_{t,j,\tilde{g}} \equiv \sigma_{t,j,\tilde{g},L+1} - \sum_{\tilde{g}'=1} \sigma_{s,j,\tilde{g}' \rightarrow \tilde{g},L+1}, \quad (4.89)$$

and the Inflow transport approximation (not shown here). Since the latter three differ only in their use of $L + 1$ order fine-group scattering cross sections, which are zero in this example, they each simplify to the Inconsistent-P, which we will compare against the Consistent-P in the following results. Also, because here $L = 0$, the Consistent-P transport-correction is equivalent to condensation by the scalar flux: effectively, there is no transport-correction.

As a comparison, let us imagine we wish to approximate the multigroup spectrum $\bar{\xi}_{j,g}$ without solving for the spatio-angular fluxes $\psi_{j,g}$. An intuitive means would be to homogenize a unit-cell of the infinite lattice to define an infinite medium. In short, assuming $N_j$ is constant in $V_j$ and each region has volume (or area, length) $V_j$, the cross sections become

$$\Sigma_{t,g} \equiv \sum_{j=1}^{J} V_j \Sigma_{t,j,g}, \quad (4.90)$$

$$\Sigma_{s,0,g' \rightarrow g} \equiv \sum_{j=1}^{J} V_j \Sigma_{s,j,0,g' \rightarrow g}, \quad (4.91)$$
and one finds the spectrum $\xi_g$ (with no subscript $j$, since all regions are homogenized) as

$$
\Sigma_{t,g}\xi_g - \sum_{g'=1}^{G} \Sigma_{s,0,g'\rightarrow g} \xi_{g'} = q_g, \quad \forall g = 1 \ldots G, \tag{4.92}
$$

a linear system of size $G \times G$, assuming the group sources $q_g$ are isotropic. For a fair comparison, we will assess the efficacy of the PGD ROM by its improvement over this infinite medium collapsing in the case of Consistent-P as well as the error compared to the full-order cross sections. We omit this comparison for the Inconsistent-P correction, because the infinite medium model cannot meaningfully estimate anisotropic moments of the flux induced here solely by streaming.

![Figure 4.13: Number of SHEM-361 groups per CASMO-SH-70 group. Highlight denotes resonance range, groups 21 to 27.](image)

Given this reference, one expects the most salient difference between the homogeneous and heterogeneous problems to be fuel self-shielding. To this end, we condense from the (fine) SHEM-361 to the (coarse) CASMO-SH-70 structure and compare the error with respect to the full-order model of resonance group cross sections collapsed by the infinite medium and PGD models.$^{11}$ As seen in Figure 4.13, such groups are selectively and considerably coarsened by this condensation. This, in effect, quantifies how well PGD accounts for self-shielding: if not at all, the error will be identical to that of the infinite medium; if effectively, the error will quickly approach zero as the number of PGD modes increases.

From Figure 4.16 we see the results of this study for CASMO-SH-70 groups 21 to 27,

---

$^{11}$Since the group boundaries of these libraries are not exactly aligned, we simply round each (coarse) CASMO-70 bound up or down to the nearest whole (fine) SHEM-361 group. Doing so, our coarse structure is an ad hoc analogue, CASMO-SH-70, defined in Appendix E.1.
while the reference fine-group cross sections and spectra are plotted in Figures 4.14 and 4.15. By neglecting heterogeneous self-shielding, the homogenized (infinite medium) problem overestimates the fuel’s coarse-group resonance cross sections by 14–17% in groups 24, 25, and 27. Initially, the PGD ROM achieves nearly identical errors, but converges further as modes are added, with all zeroth-order errors falling below 1% by the 11th mode and 0.1% by
the 34\textsuperscript{th} and 40\textsuperscript{th} for UO\textsubscript{2} or MOX respectively. Accordingly, this scenario demonstrates that the progressive PGD can be useful for cross section condensation and that the performance with few—here one to three—modes is comparable to that of an infinite medium model.

![Relative Errors of Total Cross Section Moments](image)

**Figure 4.16**: Relative errors of total cross section moments (Consistent-P and Inconsistent-P transport cross sections) by modes, Cathalau pin-cells, collapsed from SHEM-361 to CASMO-SH-70 by PGD. Errors for each $M$ are computed prior to subsequent updates.

However, a more thorough analysis would seek to characterize the error contributed by these approximated cross sections. To do so, we take the fine- and coarse-group problems to be identical, save for the specification of cross sections, and define the “coarsened flux”

$$\tilde{\psi}_g(\vec{r}, \vec{\Omega}) = \sum_{g \in \tilde{g}} \psi_g(\vec{r}, \vec{\Omega}),$$

(4.93)

where $\psi_g$ is the fine-group solution from which cross sections were condensed. This allows us to compare the “condensation error” of the flux, either $|\tilde{\psi}_g - \psi_g|$ or $|\tilde{\phi}_g - \phi_g|$, where $\psi_{\tilde{g}}$ and $\phi_g$ are the coarse-group solutions. One may wonder, should not the coarse-group flux be exact, since the cross sections were condensed using the reference solution? However, we
made two approximations during condensation: truncating the angular dependence of the total cross section and setting all cross sections constant in each volume $V_j$. As discussed in Section 4.2.1.1 and demonstrated at length by Boyd et al. [7], these assumptions can lead to measurable discrepancies, which we plot for the Cathalau pin-cells in Figure 4.17.

![Graph showing relative group-wise condensation errors, Cathalau pin-cells. Cross sections are collapsed from SHEM-361 to CASMO-SH-70 by PGD (colored) or the full-order model (black), each with a Consistent-P (solid) or Inconsistent-P (dotted) correction.]

Figure 4.17: Relative group-wise condensation errors, Cathalau pin-cells. Cross sections are collapsed from SHEM-361 to CASMO-SH-70 by PGD (colored) or the full-order model (black), each with a Consistent-P (solid) or Inconsistent-P (dotted) correction.

Clearly, we see the one-mode cross sections—essentially identical to those of the infinite medium—lead to appreciable errors of roughly 5–6%, especially in the resonance and thermal ranges. However, the 10–20 mode errors are marginal$^{12}$ and, by 30 modes, it is evident the dominant source of error is not truncation of PGD modes but rather the condensation assumptions. As such, this example illustrates the practical limit of precision for cross section condensation, beyond which additional accuracy of the transport solution is inconsequential.

$^{12}$From Figure 4.16, we can see that at 11 modes, the maximum (zeroth-order) cross section error for UO$_2$ drops from roughly 2% to 0.4%; accordingly, we can expect the corresponding 11-mode error is appreciably less than the 10-mode error plotted in Figure 4.17.
Moreover, this limit seems well in grasp of PGD approximations, even with as few as 30 modes for a 361-group problem, suggesting considerable computational savings may be possible.

4.5.5 Comparison to Coarse-Group Model

While reduced-order modeling by PGD and cross section condensation may appear unalike, the goal of each is an approximate model of neutron transport. In this sense, we can consider PGD not only as a tool for cross section generation, but as an alternative to it. As a basis of comparison, we define the coarse PGD error, $|\hat{\psi}_g - \hat{\psi}^{\text{PGD}}_g|$ or $|\hat{\phi}_g - \hat{\phi}^{\text{PGD}}_g|$, where the fine-group PGD fluxes have been coarsened analogously to $\hat{\psi}_g$ (respectively $\hat{\phi}_g$) in Equation 4.93. This workflow is visualized in Figure 4.18.

Figure 4.18: Workflow for comparing flux errors between fine-group PGD ROM and coarse-group full-order model. Despite being depicted as single boxes, each solution consists of two fluxes (angular and scalar). Likewise, the “coarse full-order model” refers to both that with Consistent- and Inconsistent-P transport corrections.

4.5.5.1 Exact Coarse-Group Cross Sections

To begin, we seek to characterize the error of a coarse-group model versus the PGD ROM on a problem identical to the one from which cross sections were condensed. To this end, we again invoke our analyses of Cathalau UO$_2$ and MOX pin-cells with SHEM-361 and CASMO-SH-70 group-structures. From Figures 4.19 and 4.20,$^{13}$ we see the PGD

$^{13}$Note that the total errors plotted in Figures 4.19, 4.20, 4.22, and 4.23 are unitless, despite being plotted alongside the errors per unit lethargy.
truncation error with 10 to 30 modes is well-below the threshold of condensation error in most resonance groups. However, one should note the normalized $L^2$ error is a more apt measure of convergence for PGD than the relative group-wise errors. By this metric, the condensation errors are almost exclusively dominated by resonance groups, while the PGD errors tend to be more evenly distributed in energy. From Table 4.2 it is evident that the total condensation errors are comparable to that achieved by PGD with ten modes, where the latter decreases by over an order of magnitude as additional modes are computed.

Figure 4.19: Group-wise condensation (black and gray) and PGD (colored) errors, Cathalau UO$_2$ pin-cell, SHEM-361 to CASMO-SH-70.

4.5.5.2 Approximate Coarse-Group Cross Sections

We now wish to introduce some deviation from the fine-group reference problem used for condensation. Again inspired by the C5G7, we choose to model two infinite lattices: a checkerboard and alternating stripes of UO$_2$ and (low-enriched) MOX pins, shown in Figure 4.21, in an attempt to emulate the interface between UO$_2$ and MOX assemblies. Doing so, we now compound the condensation errors from earlier with a “dissimilarity” error arising from the differences between the fine- and coarse-group problems. Our aim, as before, is to
Figure 4.20: Group-wise condensation (black and gray) and PGD (colored) errors, Cathalau MOX pin-cell, SHEM-361 to CASMO-SH-70.

compare the error of the coarse-group models with that of the progressive with update PGD approximation.

Figure 4.21: Cathalau UO$_2$/MOX checkerboard and stripes, 4:1 scale.

To separately quantify this dissimilarity error, we run two coarse-group models: one with “similar” cross sections, condensed from the checkered or striped fine-group problem, and one with the “dissimilar” cross sections, condensed from infinite lattices of only UO$_2$ or MOX. For simplicity, the water and cladding in each pin-cell are considered separate materials, although their cross sections are nearly identical. To account for the differing fission rates in each pin (without rendering this case a criticality problem), we first run a homogenized, $k$-eigenvalue version of this problem, from which we find the multiplication
factor $k = 1.084$ and the ratio of MOX to UO$_2$ emissions to be roughly 1.85. On this basis, we again set the source to be uniform across each fuel pin, but with a proportionally greater intensity in the MOX pin.

![Transport Correction and PGD Errors](image)

**Figure 4.22**: Group-wise condensation (plus dissimilarity if dotted, black and gray) and PGD (colored) errors, Cathalau UO$_2$/MOX checkerboard, SHEM-361 to CASMO-SH-70.

From Figures 4.22 and 4.23 we find the coarse-group model with true cross sections performs similarly as in our previous study. However, using the dissimilar (isolated) cross sections clearly incurs additional error, especially in the thermal range. Meanwhile, PGD incurs slightly greater errors than on the single pin-cell problems, yet still attains an $L^2$ error less than that of the similar and dissimilar coarse-group models by twenty modes. As before, the total $L^2$ errors of the coarse-group models are dominated almost exclusively by resonance groups.

### 4.6 Conclusions

In summary, we have derived and demonstrated the use of PGD to create a ROM of neutron transport separated in energy, as applied to nuclear reactor physics. Effectively, while
Figure 4.23: Group-wise condensation (plus dissimilarity if dotted, black and gray) and PGD (colored) errors, Cathalau UO\textsubscript{2}/MOX stripes, SHEM-361 to CASMO-SH-70.

Table 4.2: Normalized $L^2$ condensation and coarsened PGD errors, Cathalau pin-cell(s), SHEM-361 to CASMO-SH-70. Condensation errors are labeled CP or IP to indicate either a Consistent- or Inconsistent-P transport correction.

<table>
<thead>
<tr>
<th>Lattice</th>
<th>UO\textsubscript{2}</th>
<th>MOX</th>
<th>Checkerboard</th>
<th>Stripes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Angular</td>
<td>Scalar</td>
<td>Angular</td>
<td>Scalar</td>
</tr>
<tr>
<td>CP</td>
<td>Similar</td>
<td>3.91e-03</td>
<td>1.67e-03</td>
<td>4.01e-03</td>
</tr>
<tr>
<td></td>
<td>Dissim.</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>IP</td>
<td>Similar</td>
<td>4.13e-03</td>
<td>1.77e-03</td>
<td>4.01e-03</td>
</tr>
<tr>
<td></td>
<td>Dissim.</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>PGD, $M =$</td>
<td>10</td>
<td>4.82e-03</td>
<td>2.49e-03</td>
<td>4.95e-03</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>1.14e-03</td>
<td>3.80e-04</td>
<td>1.61e-03</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>5.72e-04</td>
<td>1.88e-04</td>
<td>8.87e-04</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>2.10e-04</td>
<td>8.54e-05</td>
<td>2.68e-04</td>
</tr>
</tbody>
</table>

the spatio-angular and energetic costs scale multiplicatively in the full-order model—one must solve for the angular flux in all groups, $\psi_g$—these efforts scale only additively in the
ROM—only $\psi_m$ and $\xi_m$ must be computed, where $m = 1 \ldots M$. If the number of spatio-angular or energetic unknowns grows large, as both often do for reactor physics, the ROM may become dramatically easier to solve. This is especially true if few modes $M$ are required, and so quantifying a sufficient number of modes to estimate the flux or cross sections is a key result of the numerical experiments presented here.

First, we characterized our PGD ROM on a C5G7 UO$_2$ pin-cell and both UO$_2$ and MOX Cathalau pin-cells, using the CASMO-70, XMAS-172, and SHEM-361 energy group-structures in the latter case. We also took this opportunity to characterize PGD error indicators based on the norms of the most recent modes and the residual. Notably, we found both indicators were effective estimates of the true error, though the residual could be made more representative by the action (vector-multiplication) of a separable linear operator—specifically, the streaming term. Through this comparison, we found the ROM does converge towards the full-order solution, with normalized $L^2$ errors of the angular flux less than 0.36% by 50 modes even in the worst scenario, SHEM-361 without update. Additionally, we observed that the update step improves final errors by about a factor of 10 and that the error of the scalar flux is often appreciably less than that of the angular flux—more than 3.5 times so for SHEM-361 with update. These results suggest the PGD ROM, especially with update, converges quickly—achieving sufficient accuracy within 20 to 50 modes—and may prove an effective tool for general-purpose reactor analysis.

Next, we assessed the applicability of the PGD ROM to cross section generation. Specifically, we defined our example as the condensation of material-wise cross sections for both the UO$_2$ and MOX Cathalau pin-cells from the SHEM-361 to the CASMO-SH-70 (our approximation to CASMO-70 which aligns with SHEM-361) group-structures. For these examples, few modes (1 to 3) yielded comparable errors to a homogenized, infinite medium model, while slightly more modes, between 10 and 20, were sufficient to compute the coarse-group fluxes with comparable error to the solution with exact cross sections. By 30 modes, the PGD error became nearly imperceptible, being outweighed by the errors of condensation. This suggests one could substitute the ROM for the full-order model and save computational effort with practically no detriment to accuracy. Moreover, these savings may be substantial, considering a 361-group solution can be effectively approximated with only 30 modes.

Finally, beyond this application to cross section generation, we have also shown the PGD ROM may obviate (some steps of) the condensation workflow itself, given the ten-mode
PGD achieved comparable $L^2$ errors to condensed models with seventy coarse-groups for single pin-cell analyses, even when the cross sections were generated on the problem of interest, providing an exact reference solution. Arguably, this is even more notable in that the PGD ROM had no reference solution and actually computed the fine-group (SHEM-361) flux—unavailable to the coarse-group model—which was only coarsened during post-processing for comparison. If these results prove similar in larger problems (both fixed-source and criticality), this may suggest one can employ cross sections typically reserved for pin-cell- and lattice-physics—of several hundred groups, like SHEM-361 or CASMO-586 [181]—directly in assembly or core calculations by relying on PGD, not condensation, for approximation. Doing so, deterministic models, affordable enough for routine analysis, may approach levels of accuracy currently enjoyed only by continuous-energy Monte Carlo calculations, typically reserved for benchmarking due to their computational cost. While these claims must be substantiated with numerical experiments, the promise is sufficiently great that we consider PGD ROMs of neutron transport, separated in energy, to merit further investigation and view this study as an encouraging advancement in the development and application thereof.

That said, while these comparisons are instructive, the $L^2$ error of the flux is not the only quantity of interest for nuclear engineers; often, the multiplication factor $k$ is another decisive global parameter. Noting this, the following chapter extends this ROM to the $k$-eigenvalue problem by means of an original algorithm, Progressive PGD with Eigenvalue Update. Validating this ROM by similar numerical experiments yields encouraging results, comparable to those presented here for fixed-source problems.
CHAPTER 5
MODEL ORDER REDUCTION IN ENERGY FOR
EIGENVALUE PROBLEMS

5.1 Introduction

In the previous chapter, we demonstrate an \textit{a priori} ROM of neutron transport separated in energy by PGD in which the computational cost (assuming spatio-angular costs dominate) scales linearly with the number of “modes” required to resolve the solution rather than the number of energy groups—as it would in the full-order model in the best case. As a result, this ROM may make it practical to refine the energy mesh enough to mitigate the errors induced by condensation, or even obviate one or more condensation steps altogether. However, the former work was limited to fixed-source neutron transport problems, rather than the $k$-eigenvalue formulation used to describe the asymptotic state of multiplying systems, such as reactors. Necessarily, this limitation also precluded empirical results regarding the ROM’s accuracy in estimating the $k$-eigenvalue, or multiplication factor, one of the most important quantities-of-interest in reactor physics.

The purpose of this chapter, then, is to reformulate the PGD ROM so as to support these ubiquitous criticality problems. Moreover, we perform numerical experiments to validate the model’s precision for the same suite of representative LWR (Cathalau [176]) pin-cell benchmarks. In support of these goals, we also devise a novel extension of Ammar and Chinesta’s [189] approach to solving eigenvalue problems with PGD ROMs and an efficient implementation of PGD’s update step based on recursive LU factorization. Altogether, we find this ROM (both the Galerkin and Minimax versions) achieves satisfactory precision within a tractable number of modes—roughly, ten to twenty—and has a computational cost comparable to that of the fixed-source ROM previously presented.

5.2 Motivation

While the previous chapter demonstrated the use of Progressive PGD, separated in energy, for fixed-source neutron transport problems, analysts are often concerned with the

\footnote{Portions of this chapter have been submitted to: K. A. Dominesey and W. Ji, “Reduced-order modeling of neutron transport eigenvalue problems separated in energy by Proper Generalized Decomposition,” \textit{J. Comput. Phys.}, 2022, under review.}
multiplication factor, \( k \), of a particular system. This factor represents the ratio of neutrons produced by fission to those lost by leakage or absorption, and therefore quantifies whether the nuclear chain reaction is decreasing (subcritical), self-sustaining (critical), or increasing (supercritical). Practically, control of a nuclear reactor requires the multiplication factor to be less than \( 1/(1 - \beta_{\text{eff}}) \), where \( \beta_{\text{eff}} \) is the effective delayed neutron fraction, which depends on the reactor in question, but is typically around 0.7% for LWRs [144]. Beyond this point, a reactor is said to be prompt supercritical, or supercritical on prompt neutrons alone, and the reaction will increase too rapidly to be controlled by engineered safety systems. Therefore, \( \beta_{\text{eff}} \) usually dictates the margin \( 1/(1 - \beta_{\text{eff}}) - 1 \approx 0.00705 \) between normal operation and an uncontrollable power excursion. Clearly then, most useful calculations of \( k \) must be precise to an even tighter margin, usually measured in units of percent mille (pcm), denoting one-thousandth of a percent, \( 10^{-5} \).

Unfortunately, meeting these exacting standards remains a challenge in deterministic neutron transport, not least due to the use of multigroup-collapsed cross sections. Recent studies have demonstrated that this collapse can introduce discrepancies on the order of hundreds of pcm, even with a perfect reference solution (weighting flux) and an identical discretization of space and angle [7], [8]. And, of course, even if an accurate solution is possible, it will often demand high-fidelity discretizations (in space, angle, and energy), rendering the simulation resource-intensive and time-consuming.

In this chapter, we propose a Progressive PGD ROM of neutron transport, separated in energy, as a means of addressing both issues—that is, enhancing precision and/or reducing computational burden. To the latter end, we first assess the PGD ROM as a potential replacement for an otherwise identical full-order model. Second, and to both ends, we compare a fine-group ROM against a coarse-group full-order model, to explore whether PGD may offer a preferable, alternative means of approximation to cross section condensation. Notably, for fixed-source problems, similar applications are investigated in [13]; however, this article is the first to explore these topics in \( k \)-eigenvalue neutron transport.

5.3 Previous Work

The solution of eigenvalue problems by PGD ROMs remains an evolving practice in literature; in specific, several such ROMs have been demonstrated, but only in a small number of articles as compared to the large volume of work on PGD for linear, or nonlinear (but
Moreover, among these few papers, five distinct algorithms have been proposed, of varying suitability to our application of interest. The first of these was authored by Ammar and Chinesta [189] and applied to the Schrödinger equation, in which the “projection” step, which re-weights all modes, is cast as a (reduced-order) eigenvalue problem. That aside, the methodology is as usual for Progressive PGD. This algorithm, plus two original methods—based on the minimization of either the Rayleigh quotient or residual—is investigated further by Cancès et al. [190], who provide both numerical experiments and convergence results for elliptic eigenproblems. Later, González-Pintor et al. [131] applied PGD to a neutron diffusion eigenproblem. There, the Progressive PGD algorithm is modified; following an initialization step (which iterates between two coupled eigenproblems), a Newton method is used to compute eigenvalue and eigenvector corrections (the latter being comprised of multiple modes). Additionally, a Rayleigh-Ritz procedure (written as an eigenproblem) is used to both re-weight the current iterate and Newton correction (before they are summed) and re-compute the eigenvalue. More recently, in Senecal’s thesis [136] and an article by Prince and Ragusa [137], a fifth PGD eigenvalue method is presented. Specifically, both authors cast their eigenvalue solver (Power Iteration, possibly shifted in the latter work) as an outer loop, with PGD enrichment comprising an inner loop. Prince and Ragusa also introduce additional steps, such as solving the projection step as an eigenproblem and compressing the PGD eigenvector following every Power Iteration.

For our PGD method, we expect the cost to be dominated by solving the spatio-angular (rather than energetic) equations, as reactor physics applications typically involve comparatively few energy groups (dozens to hundreds) but millions to billions of spatio-angular degrees-of-freedom. Therefore, we choose to nest the eigenvalue iteration as an inner loop—such that it does not involve this expensive operation—and to recompute the eigenvalue as frequently as practical. Given these considerations, a natural approach is to solve the update in energy—which recomputes all energy modes, see Section 5.4.3—as an eigenproblem for $k$; this algorithm is essentially identical to that proposed by Ammar and Chinesta [189], except that the reduced-order eigenproblem is the update rather than projection step. Herein, we demonstrate this approach is effective for pin-cell problems like those encountered in LWRs.
5.4 Methodology

We begin with the generalized eigenvalue problem for the neutron flux $\psi$ as a function of position $\vec{r}$, direction (angle) $\vec{\Omega}$, and energy $E$, reproduced below for convenience

$$\vec{\Omega} \cdot \nabla \psi(\vec{r}, \vec{\Omega}, E) + \Sigma_t(\vec{r}, E)\psi(\vec{r}, \vec{\Omega}, E) - \int_{E_0}^{E_G} \int_{4\pi} \Sigma_s(\vec{r}, \vec{\Omega}' \rightarrow \vec{\Omega}, E' \rightarrow E)\psi(\vec{r}, \vec{\Omega}, E)d\Omega'dE' = \frac{1}{k} \chi(\vec{r}, E) \int_{E_0}^{E_G} \int_{4\pi} \nu \Sigma_f(\vec{r}, E')\psi(\vec{r}, \vec{\Omega}', E')d\Omega'dE'. \quad (5.1)$$

Equivalently, multiplying both sides by $k$ and gathering the left- and right-hand-sides into the fixed-source and fission operators, $B$ and $A$ respectively, we have more succinctly

$$A\psi = kB\psi. \quad (5.2)$$

While this equation generally admits multiple eigenpairs, consisting of an eigenvector $\psi$ and eigenvalue $k$, at present we will consider only the dominant eigenpair (that with the largest eigenvalue), which describes the asymptotic state of the idealized, “static” reactor.\(^{14}\)

5.4.1 Separated Representation

To apply PGD, we must first separate the coefficients of Equation 5.1. To do so, we recall the separated total and scattering cross sections of Equations 4.5 and 4.6 and apply a similar decomposition of the emission spectra $\chi$ and neutron production cross section $\nu \Sigma_f$,

$$\chi(\vec{r}, E)\nu \Sigma_f(\vec{r}, E') = \sum_{j=1}^{J} N_j(\vec{r})\chi_j(E)\nu \sigma_{f,j}(E'). \quad (5.3)$$

As before, this separation of coefficients is conventional even in full-order neutron transport, rather than a unique assumption of our PGD ROM. Next, we substitute in the separable approximation of the solution, Equation 4.1, to arrive at the separated neutron transport

\(^{14}\)Finding multiple eigenpairs poses no methodological difficulties, but the dominant pair is often the primary or exclusive interest of a reactor physicist.
equation for $k$-eigenvalue problems

$$
\sum_{m=1}^{M} \left( \vec{\Omega} \cdot \nabla \psi_m(\vec{r}, \vec{\Omega}) \xi_m(E) + \sum_{j=1}^{J} N_j(\vec{r}) \psi_m(\vec{r}, \vec{\Omega}) \sigma_{t,j}(E) \xi_m(E) \right) - \sum_{j=1}^{J} N_j(\vec{r}) \sum_{\ell=0}^{L} \frac{2\ell + 1}{4\pi} \int_{4\pi} P_\ell(\vec{\Omega} \cdot \vec{\Omega}') \psi_m(\vec{r}, \vec{\Omega}) d\Omega' \int_{E_0}^{E_g} \sigma_{a,\ell,j}(E' \rightarrow E) \xi_m(E') dE' \right) \right) \right) (5.4)

\right.

which is identical to that for fixed-source problems, except that the fixed source has been replaced by a fission source, scaled by the reciprocal of the $k$-eigenvalue.

Having again achieved a separable model, PGD proceeds as in the previous chapter, with two caveats. First, a special initialization procedure is invoked for $\psi_1$ and $\xi_1$, in which the former is set as uniform and isotropic and the latter as (what amounts to) the fundamental eigenfunction of a homogenized, infinite medium approximation; likewise, $k$ is initialized as the corresponding eigenvalue. Secondly, the update of energy modes $\{\xi_m\}_{m=1}^{M}$, as in Equation 4.35, is recast as an eigenproblem by which $k$ is also recomputed (as summarized in Section 5.4.5). For all other purposes, $k$ is simply considered to be a fixed constant. This original algorithm is referred to herein as Progressive PGD with Eigenvalue Update for sake of clarity.

### 5.4.2 Spatio-angular Submodel

To derive an equation for the last spatio-angular mode $\psi_M$, we take the weak form of Equation 5.4 by integration against the test function $\tilde{\xi}_M$, as in Equation 4.9. For convenience, we then collect the coefficients into forms typical of macroscopic cross sections, as in Equations 4.5 and 4.6 and likewise

$$
\tilde{\nu} \Sigma_{f,M,M}(\vec{r}) \equiv \sum_{j=1}^{J} N_j(\vec{r}) \left( \tilde{\xi}_M, \chi_j \int_{E_0}^{E_g} \nu \sigma_{f,j} \xi_m dE' \right) \right) \right) \right) \right).

Next, by defining the linear operator

$$
A_{M,m}^{(g)} \psi_m = \frac{\tilde{\nu} \Sigma_{f,M,M}(\vec{r})}{4\pi} \int_{4\pi} \psi_m(\vec{r}, \vec{\Omega}') d\Omega'

(5.6)
and recalling that of $B_{M,m}^{(G)}$ from Equation 4.14, the resulting equation can be written as

$$
\sum_{m=1}^{M} \left( A_{M,m}^{(G)} - k B_{M,m}^{(G)} \right) \psi_m = 0.
$$

Equivalently, collecting all the summands $m < M$ into a residual term $r_{M}^{(G)}$, we have

$$
\left( A_{M,M}^{(G)} - k B_{M,M}^{(G)} \right) \psi_M = - \sum_{m=1}^{M-1} \left( A_{M,m}^{(G)} - k B_{M,m}^{(G)} \right) \psi_m = r_{M}^{(G)}.
$$

Granting that terms $\bar{\xi}_M$, $\{\xi_m\}_{m=1}^{M}$, $\{\psi_m\}_{m=1}^{M-1}$, and $k$ are known (from a previous iteration), this is a linear equation describing energy-independent, fixed-source neutron transport which can be solved for $\psi_M$; notably, we do not treat this as an eigenvalue problem and, in fact, cannot do so for $M > 1$ because of the residual term $r_{M}^{(G)}$.

### 5.4.3 Energetic Submodel

For the energy equation, we proceed analogously, but allow the test function to have index $m^*$ (rather than assuming $m^* = M$), as in Equation 4.19. Again rearranging our coefficients to fit the familiar notation of cross sections, as in Equations 4.21 and 4.22 and

$$
\bar{\nu} \Sigma_{f,j,m^*,m}(E') \equiv \frac{1}{4\pi} \left( \bar{\psi}_{m^*}, N_j \int_{4\pi} \psi_m d\Omega' \right)_{V,4\pi} \nu \sigma_{f,j}(E'),
$$

we achieve the linear operator

$$
A_{m^*,m}^{(V,4\pi)} \xi_m \equiv \sum_{j=1}^{J} \chi_j(E) \int_{E_0}^{E_G} \bar{\nu} \Sigma_{f,j,m^*,m}(E') \xi_m(E') dE'.
$$

Given these preliminaries, plus $B_{m^*,m}^{(V,4\pi)}$ as previously defined in Equation 4.25, we can succinctly write the progressive equation—that which is solved for $\xi_M$—as

$$
\left( A_{M,M}^{(V,4\pi)} - k B_{M,M}^{(V,4\pi)} \right) \xi_M = - \sum_{m=1}^{M-1} \left( A_{M,m}^{(V,4\pi)} - k B_{M,m}^{(V,4\pi)} \right) \xi_m = r_{M}^{(V,4\pi)}.
$$
which can be solved given \( \bar{\psi}_M, \{ \psi_m \}_{m=1}^M, \{ \xi_m \}_{m=1}^{M-1} \) and \( k \). Further, we define the update equation(s)—those which are solved for \( \{ \xi_m \}_{m=1}^M \)—as

\[
\sum_{m=1}^{M} \left( A_{m^*,m}^{(V,4\pi)} - k B_{m^*,m}^{(V,4\pi)} \right) \xi_m = 0, \quad \forall m^* = 1 \ldots M. \tag{5.12}
\]

Granting that \( \{ \bar{\psi}_{m^*} \}_{m^*=1}^M \) and \( \{ \psi_m \}_{m=1}^M \) are known and \( k \) is unknown, the latter can be written as an eigenvalue problem

\[
\widetilde{A} \xi = k \widetilde{B} \xi \tag{5.13}
\]

where \( \widetilde{B} \) and \( \xi \) are as in Equation 4.37 and likewise

\[
\widetilde{A} \equiv \begin{bmatrix}
A_{1,1}^{(V,4\pi)} & \cdots & A_{1,M}^{(V,4\pi)} \\
\vdots & \ddots & \vdots \\
A_{M,1}^{(V,4\pi)} & \cdots & A_{M,M}^{(V,4\pi)}
\end{bmatrix} \tag{5.14}
\]

As before, this notation illustrates that the progressive problem, Equation 5.11, corresponds to the last block row of the update problem, Equation 5.12.

### 5.4.4 Galerkin and Minimax Formulations

As before, the test functions of Galerkin PGD are equal to the trial functions—that is, \( \bar{\psi}_{m^*} \leftarrow \psi_{m^*} \) and \( \bar{\xi}_{m^*} \leftarrow \xi_{m^*} \)—by definition. Meanwhile, by an identical procedure as in Section 4.4.5, one here finds the adjoint problems

\[
\begin{align*}
\left( A_{M,M}^{(r,\Omega)} - k B_{M,M}^{(r,\Omega)} \right)^\dagger \bar{\psi}_M &= (\xi_M, \xi_M)_G \psi_M, \tag{5.15} \\
\left( A_{M,M}^{(E)} - k B_{M,M}^{(E)} \right)^\dagger \bar{\xi}_M &= (\psi_M, \psi_M)_{V,4\pi} \xi_M. \tag{5.16}
\end{align*}
\]

As in Equations 5.8 and 5.11, the \( k \)-eigenvalue as it appears above is treated as a fixed constant. Doing so, one can solve these equations for \( \bar{\psi}_M \) and \( \bar{\xi}_M \) respectively given some pair of modes \( \psi_M \) and \( \xi_M \).
5.4.5 Application to Eigenvalue Problems

Given these preliminaries, we can now outline our PGD algorithm. As in [13], we select the Progressive PGD, which begins with zero modes (no prior knowledge of the solution) and incrementally adds more modes (“enriches” the reduced basis) until convergence, as measured by some error estimator. This only requires solving the nonlinearly-coupled Equations 5.8 and 5.11 for $\psi_M$ and $\xi_M$ respectively. However, for $M > 1$, neither of these equations are eigenvalue problems, despite both containing $k$, because of the presence of a residual term. Therefore, it is not immediately clear how best to apply PGD to what was originally a straightforward, linear eigenvalue problem, and several different algorithms have been proposed, as discussed in Section 5.3.

For our application, we expect the spatio-angular (rather than energetic) equations will be much more burdensome to solve. Following this line of reasoning, a straightforward and seemingly economical approach is to re-compute the eigenvalue during the update step, since it can be solved as a generalized eigenvalue problem. This excludes the spatio-angular operators entirely from the eigenvalue iteration and yields a simple, linear eigenproblem which can be readily solved by familiar algorithms (Power Iteration, Krylov-Schur, Davidson Methods, and so on). The complete details of this approach are listed in Algorithm 3. Note that the residual norms referred to therein are, like in Algorithm 1, more specifically as defined $\|\bar{r}_{m+1}\|_2 / \|\bar{r}_{m}\|_2$ where $m$ is as stated in line 3, and likewise $\bar{r}_{m}^{(*)}$ as in Section 5.4.2 or 5.4.3 as appropriate.

5.4.6 Recursive LU Factorization

To solve the eigenproblem Equation 5.12, we apply a shift-of-origin (with shift $\sigma$) and select Krylov-Schur as the iterative solver. At each iteration, this requires one matrix-vector multiplication with operator $\bar{B}^{-1}\bar{A} - \sigma I$. While one may be inclined to solve $\bar{B}$ iteratively (since $\bar{B}$ could become large), this iterative process must be repeated at every Krylov-Schur iteration—in contrast with a direct factorization, which need only be computed once. That said, a direct factorization has time complexity $O(N^3)$ where $N$ is the dimension of a square matrix (here $M \times G$) and so may become impractically expensive.\(^{15}\)

\(^{15}\)To be precise, LU decomposition has the same asymptotic complexity as matrix multiplication [191], and so slightly faster algorithms are possible [192]. These considerations are largely irrelevant to the present discussion, however, so we presently assume LU factorization and matrix multiplication both have complexity
Algorithm 3: Galerkin or Minimax Progressive PGD, with Eigenvalue Update

\begin{algorithm}
\begin{align*}
1 & \psi_1, \tilde{\psi}_1 \leftarrow 1/\|1\|_{L^2} \quad // \text{uniform, isotropic, normalized guess} \\
2 & k, \xi_1 \leftarrow \text{solution of Equation 5.12} \quad // \text{corresponding spectrum and } k \\
3 & \textbf{for } m = 2 \ldots M \textbf{ do} \quad // \text{enrichment iteration} \\
4 & \psi_m, \tilde{\psi}_m \leftarrow 1/\|\psi_m\|_{L^2}, \tilde{\psi}_m/\|\tilde{\psi}_m\|_{L^2} \\
5 & \textbf{while residual} > \text{tolerance} \textbf{ do} \quad // \text{nonlinear iteration} \\
6 & \psi_m, \tilde{\psi}_m \leftarrow \psi_m/\|\psi_m\|_{L^2}, \tilde{\psi}_m/\|\tilde{\psi}_m\|_{L^2} \\
7 & \text{residual}_\text{xi} \leftarrow \text{relative } \ell^2 \text{ norm of residual of Equation 5.11} \\
8 & \xi_m \leftarrow \text{solution of Equation 5.11} \\
9 & \text{if minimax then solution of Equation 5.16 else } \xi_m \\
10 & \text{residual}_\text{psi} \leftarrow \text{relative } \ell^2 \text{ norm of residual of Equation 5.8} \\
11 & \psi_m \leftarrow \text{solution of Equation 5.8} \\
12 & \tilde{\psi}_m \leftarrow \text{if minimax then solution of Equation 5.15 else } \psi_m \\
13 & \text{residual} \leftarrow \sqrt{\text{residual}_\text{xi}^2 + \text{residual}_\text{psi}^2} \\
14 & k, \{\xi_{m^*}\}_{m^* = 1}^m \leftarrow \text{solution of Equation 5.12} \\
15 & \psi \leftarrow \sum_{m=1}^M \psi_m \times \xi_m
\end{align*}
\end{algorithm}

However, one can exploit the structure of matrix $\tilde{B}$ to speed up the factorization. Namely, the blocks $B_{m^*,m}^{(V,4\pi)}$ depend only on mode $\psi_m$ and test function $\tilde{\psi}_{m^*}$, which do not change for $m$ and $m^* < M$—that is, they are never updated. Supposing we superscript matrices $\tilde{B}^{(M)}$ according to the number of modes $M$, it is clear that

$$
\tilde{B}^{(M)} \equiv \begin{bmatrix}
\tilde{B}^{(M-1)} & B_{1,M}^{(V,4\pi)} \\
B_{M,1}^{(V,4\pi)} & \cdots & B_{M,M}^{(V,4\pi)}
\end{bmatrix}
$$

(5.17)

for $M > 1$ while $\tilde{B}^{(1)} \equiv B_{1,1}^{(V,4\pi)}$. This implies the LU factorization of $\tilde{B}^{(M)}$ with $M > 1$ can be computed from that of $\tilde{B}^{(M-1)}$ and the Schur complement $\tilde{B}^{(M)}/B_{M,M}^{(V,4\pi)}$—the former of which is available from the previous enrichment iteration. This is effectively equivalent to factorizing a matrix by recursive partitioning, which has been explored in great detail elsewhere [193], [194]. At present, it suffices to provide the high-level description in Function 4. To be clear, our function amounts to a single, recursive factorization of $\tilde{B}^{(M)}$ for the final value of $M$, though the cost of this one factorization is amortized across every enrichment iteration (with the marginal cost for mode $m + 1$ being roughly $2m^2G^2$). Moreover, as one

$O(N^3)$. 

may wonder if any additional cost is incurred by not partitioning the blocks equally (as in [194]), we prove in Appendix F.3 that to leading order, the number of floating point operations (FLOPs), $\frac{2}{3}M^3G^3$, is unchanged.

Function 4: RecursiveLU($A \in \mathbb{R}^{(M+1)G \times (M+1)G}$)

// Partition the matrix $A$ into four submatrices.
1 $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \in \mathbb{R}^{(M+1)G \times (M+1)G}$

where $A_{11} \in \mathbb{R}^{MG \times MG}$, $A_{12} \in \mathbb{R}^{MG \times G}$, $A_{21} \in \mathbb{R}^{G \times MG}$, $A_{22} \in \mathbb{R}^{G \times G}$

// Assume $A_{11}$ has already been factorized such that $P_{11}A_{11} = L_{11}U_{11}$
// where $L_{11}$ is unit lower triangular and $U_{11}$ is upper triangular.
2 $A_{12} \leftarrow P_{11}A_{12}$  // pivot $A_{12}$, equivalent to LAPACK xLASWP

// Perform upper triangular solve, equivalent to Level 3 BLAS xTRSM.
3 Find $X_{21}$ such that: $X_{21}U_{11} = A_{21}$

$A_{21} \leftarrow X_{21}$

// Perform lower triangular solve, equivalent to Level 3 BLAS xTRSM.
4 Find $X_{12}$ such that: $L_{11}X_{12} = A_{12}$

$A_{12} \leftarrow X_{12}$

// Compute Schur complement, equivalent to matrix-matrix multiply
// performed by Level 3 BLAS xGEMM.
5 $A_{22} \leftarrow A_{22} - A_{21}A_{12}$

// Factorize Schur complement, equivalent to Level 3 BLAS xGETRF.
6 Factorize $A_{22}$ such that: $P_{22}A_{22} = L_{22}U_{22}$

7 $A_{21} \leftarrow P_{22}A_{21}$  // pivot $A_{21}$, equivalent to LAPACK xLASWP

8 Offset and append pivot indices from $P_{22}$ to $P_{11}$ to create permutation matrix $P$

It should also be understood that this recursive LU factorization is equally applicable to the traditional PGD update step (rather than our formulation as an eigenproblem). In fact, $\tilde{B}$ is the same matrix as must be inverted in the fixed-source energy update, and so this recursive LU factorization in that case provides a direct method with no iteration.

5.5 Numerical Results

Next, we seek to validate our PGD ROM by numerical experiments on LWR pin-cell benchmarks. First, we compare the ROM to an equivalent full-order model to quantify the error (in the flux, fission source, and eigenvalue) incurred by PGD. Second, we consider a fine-group ROM compared to a coarse-group full-order model, judging the error of both with respect to a fine-group full-order model. This effectively quantifies the relative merits of cross section condensation and the PGD ROM as alternative means of model approximation.
5.5.1 Numerical Sundries

Most numerical methods, iteration limits, and tolerances used are identical to those of Section 4.5.1. Briefly, the key features include: a discrete ordinates ($S_N$) discretization of angle with a $S_8^4$ product Gauss-Legendre-Chebyshev angular quadrature\footnote{That is, 4 polar and 8 azimuthal angles per quadrant, 128 angles in total}; matrix-free transport sweeps; and linear ($p = 1$), discontinuous Lagrangian finite elements over an unstructured quadrilateral mesh. The exceptions are those settings pertaining to the update step—which we now solve as an eigenproblem—and the full-order eigensolver. For the former, we use a Krylov-Schur solver (as implemented in SLEPc\cite{195}) with a tolerance of $10^{-6}$ and a shift-of-origin, where the shift $\sigma$ is set to be the eigenvalue from the previous enrichment iteration (or the initial Rayleigh quotient, if $M = 1$). The current modes are always supplied as the initial guess of the eigenvector. As explained in Section 5.4.6, the operator $B_E$ is directly inverted using recursive LU factorization. The latter eigensolver, meanwhile, is chosen to be Power Iteration with a tolerance of $10^{-10}$ on the absolute difference between successive eigenvalue estimates. As before, models are implemented in C++ using version 9.1 of the deal.II finite element library\cite{179}, \cite{180}.

5.5.2 Computational Cost

The computational cost of the eigenvalue PGD ROM—at least, per mode—is comparable to that of the fixed-source ROM presented in the previous chapter (see Section 4.5.2). This is because the eigenvalue loop is nested so as to involve only energetic (not spatio-angular) operators, which are generally fast to apply/solve compared to the latter. While this expectation is not guaranteed, many reactor physics applications involve a million or more spatio-angular degrees-of-freedom yet a few hundred energy groups at most, in which case it appears a safe assumption.

Should the update step become inconveniently time-consuming regardless, some straightforward means to mitigate this cost seem apparent. For instance, the recursive LU factorization explained in Section 5.4.6 was found to be effective. Alternatively, the update could be performed less frequently: perhaps only every two or five modes or, better yet, only when convergence stagnates (as in the Arnoldi-like method of\cite{196}). However, since these latter considerations have proved unnecessary so far, we defer them to future work.

Of course, even if the spatio-angular costs remain dominant, the eigenvalue PGD
ROM may converge somewhat slower with $M$, because the eigenvalue $k$ is being recomputed throughout enrichment. If so, the fixed-source PGD ROM may be slightly faster in achieving a given precision. However, comparing our numerical results with those of [13], we find this effect appears, at worst, marginal in our pin-cell benchmarks.

5.5.3 Comparison to Fine-Group Model

Our first goal is to quantify the error of the PGD ROM compared to the full-order model with an identical energy mesh. To do so, we measure the $L^2$ norms of the error in three distributions—the angular flux $\psi$, scalar flux $\phi$, and fission source $q$—and the error of the $k$-eigenvalue (a scalar). Each of these quantities-of-interest is in fact a function of $\psi$; specifically,

$$\phi(\vec{r}, E) \equiv \int_{4\pi} \psi(\vec{r}, \vec{\Omega}, E) d\Omega,$$  \hspace{1cm} (5.18)

$$q(\vec{r}) \equiv \int_{E_0}^{E_G} \nu \Sigma_f(\vec{r}, E) \phi(\vec{r}, E) dE,$$  \hspace{1cm} (5.19)

and $k$ corresponds to eigenfunction $\psi$. However, each will converge at different rates: $\phi$ faster than $\psi$ due to cancellation of error; $q$ possibly faster than $\phi$ for the same reason, though not necessarily since the integral is weighted by $\nu \Sigma_f$; and $k$ lacking a straightforward relationship to $\psi$. For context, we also plot the errors in distributions $\psi$ and $\phi$ achieved by the SVD of the full-order angular flux. Since the SVD provides the provably optimal decomposition in the $L^2$ norm, this essentially serves as a lower bound which no other decomposition (including PGD) can surpass. That said, each of these metrics requires knowledge of the full-order angular flux, which would not be available in practice. Accordingly, we also measure the $L^2$ norm of the latest modes—which should approximately equal the truncation error if the series is converging quickly—and the residual $A\psi - kB\psi$, which admits a separated representation. More precisely, we measure both this angular residual and the scalar residual, which is equivalent but integrated in angle. Each of these errors and error estimators—normalized by the $L^2$ norm of the full-order angular flux (except the difference in $k$)—is plotted for both the Galerkin and Minimax PGD in Figures 5.1 and 5.2 for the UO$_2$ and (low-enriched) MOX

\footnote{At least in the full-order model; technically, in the ROM, $k$ corresponds to eigenvector $\xi$.}

\footnote{More precisely, SVD yields the best decomposition in the $\ell^2$ norm, but one can use this result to find the same in the $L^2$ norm by straightforward pre- and post-processing, see Appendix D.}
pin-cells respectively.

Figure 5.1: Convergence of Galerkin and Minimax PGD (with eigenvalue update), by group structure, UO$_2$ pin-cell.

We can see from these figures first and foremost that each of these ROMs converges, though erratically so in some cases—as evidenced by large modes corresponding with sudden drops in error. These features could perhaps be mitigated or eliminated by improvements in the nonlinear solver, which does not always converge. Despite this, the $L^2$ error in $\psi$ decreases to less than 0.1% within 50 modes in every scenario. Additionally, as expected, the relative errors of $\phi$ and (usually) $q$ are observed to be less than that of $\psi$, often by an appreciable margin. We also find that, by 50 modes, all of these metrics are slightly lower in the Minimax as compared to the Galerkin PGD. At the same number of modes,
Figure 5.2: Convergence of Galerkin and Minimax PGD (with eigenvalue update), by group structure, MOX pin-cell.

the $k$-eigenvalue error is less than $10 \text{ pcm} \left(10^{-4}\right)$ in every case except for MOX, SHEM-361, with Galerkin PGD, in which case it is still less than 20 pcm. These errors tend to be far less, in several cases by roughly a decade, in the Minimax (instead of Galerkin) PGD, which achieves an error less than 2 pcm with $M = 50$ in all scenarios.

While these results appear promising, one cannot meaningfully determine what level of error is acceptable in practice—and therefore, how many modes are required—without first assessing the precision of the full-order model which the PGD ROM is approximating. Some cursory answers can be gleaned by comparing each of the three full-order models (CASMO-70, XMAS-172, or SHEM-361) for the two fuels considered (UO$_2$ and MOX) both with each
Table 5.1: Reference $k$-eigenvalues from OpenMC and multigroup full-order models. The error represents the difference from OpenMC.

<table>
<thead>
<tr>
<th></th>
<th>UO$_2$</th>
<th>MOX</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$k$</td>
<td>Error [pcm]</td>
</tr>
<tr>
<td>OpenMC</td>
<td>1.326655</td>
<td>-</td>
</tr>
<tr>
<td>CASMO-70</td>
<td>1.323035</td>
<td>−362.0</td>
</tr>
<tr>
<td>XMAS-172</td>
<td>1.321269</td>
<td>−538.6</td>
</tr>
<tr>
<td>SHEM-361</td>
<td>1.328544</td>
<td>+188.9</td>
</tr>
</tbody>
</table>

From this table, we see that the eigenvalues of the full-order model differ from those of OpenMC on the order of hundreds of pcm. Moreover, the biases are not consistent between fuels or energy meshes—across which they differ by at least 200 pcm and 600 pcm respectively. Inspecting the plots, meanwhile, reveals that the additional error incurred

---

These errors are due not only to cross section condensation, but also the discretization of space and angle and the assumption of transport-corrected isotropic (rather than anisotropic) scattering in the deterministic model. However, Boyd et al. [7] and Nelson et al. [8] eliminate the former and latter considerations respectively and find appreciable discrepancies remain regardless.
by PGD vanishes quickly within the first ten modes; by twenty, the detriment to accuracy appears to be marginal or negligible in the Galerkin or Minimax case respectively. Further, by virtue of plotting all our $k$-eigenvalues on a common scale, some interesting trends become apparent. First, we observe that in all but two cases (UO$_2$ fuel with CASMO-70 or XMAS-172 energy mesh) the Galerkin PGD appreciably underpredicts the multiplication factor with few (roughly, one to five) modes. This is consistent with the findings of [13]: namely, that the Galerkin PGD ROM with one mode ($M = 1$) achieves an energy spectrum similar to that produced by the “homogeneous infinite medium” assumption, and therefore fails to represent the effects of resonance self-shielding. Neglecting this phenomenon leads to an overprediction of coarse-group resonance cross sections (as seen in [13]) and an underprediction of the $k$-eigenvalue. This effect also explains the two exceptions—they are the only cases without resolved resonance cross sections (see Figure 4.10)—and why the underpredictions are more severe in the MOX cases—the plutonium in the fuel contributes additional resonances.

5.5.4 Comparison to Coarse-Group Model

The foregoing discussion assesses the prospect of replacing a full-order model with an otherwise identical PGD ROM. However, if one aims to obviate cross section condensation, the more pertinent comparison is between the fine-group PGD ROM and a coarse-group full-order model. As each approximates the fine-group full-order model—the former by seeking a low-rank approximation and the latter by coarsening the energy mesh—one can judge their relative merits by the corresponding errors in coarse-group fluxes, fission source, and $k$-eigenvalue. For this task, our benchmark problems are Cathalau lattices with SHEM-361 and CASMO-SH-70 [13] as the fine and coarse energy meshes, respectively. Crucially, as illustrated in Figure 4.18, both the PGD ROM and the fine-group full-order model are solved on the SHEM-361 energy mesh; their solutions are only coarsened to the CASMO-SH-70 structure during post-processing. By contrast, the SHEM-361 flux from the latter model is used to condense cross sections for the coarse-group model, such that its solution is on the CASMO-SH-70 mesh. This isolates the condensation error, as the fine- and coarse-group models are otherwise identical, and allows a fair comparison of CASMO-SH-70 fluxes—though not SHEM-361 fluxes, since this cannot be computed from the coarse-group model. As such, this simple condensation study provides a meaningful baseline against which to benchmark the precision of the PGD ROM with respect to the coarse-group angular and scalar fluxes,
\(\psi\) and \(\phi\), the fission source \(q\), and the multiplication factor \(k\). Results for each of the two fuels (UO\(_2\) and MOX), two coarse-group models (those with Consistent- and Inconsistent-P transport-corrected cross sections [13], [147]), and two formulations of PGD (Galerkin and Minimix) are presented in Figure 5.4.

Figure 5.4: Convergence of CASMO-SH-70 fluxes, fission source, and \(k\) by SHEM-361 PGD, versus CASMO-SH-70 full-order model.

From this plot, we observe that the error in \(\psi\), \(\phi\), and \(k\) achieved by the coarse-group models is comparable to that attained by PGD between roughly ten and twenty modes. From ten to fifty modes, each of these errors decreases by slightly more than an order-of-magnitude. Regarding the \(L^2\) error of the fission source, the PGD ROM surpasses the accuracy of the coarse-group models by twenty-six modes. These results are especially notable because the PGD ROM, unlike the coarse-group model, required no reference solution and produced a fine-group solution (only coarsened here for sake of comparison) rather than only the coarse-group solution.
5.6 Conclusions

In summary, we have derived and demonstrated a PGD ROM of neutron transport, separated in energy, as applied to \( k \)-eigenvalue problems. This is an extension of previous work which considered PGD for fixed-source neutron transport (the preceding chapter) and \( k \)-eigenvalue neutron diffusion [131], [136], [137], but not \( k \)-eigenvalue neutron transport. Moreover, we have validated this model on a suite of two pin-cell benchmarks (with three energy meshes) representative of LWRs. Underpinning these efforts, we also developed a novel extension of Ammar and Chinesta’s algorithm for PGD eigenproblems [189], in which the eigenvalue loop is nested so as to avoid involving costly spatio-angular solves. Further, we explained how the update step, which we solve as an eigenproblem in our algorithm, can be solved more efficiently using a recursive LU factorization.

Through this investigation, we found both our ROM converges reasonably quickly, with the \( L^2 \) norm of the angular flux error (with 50 modes) being less than 0.1\% in all cases. Given the same number of modes, the Galerkin and Minimax ROMs achieved \( k \)-eigenvalue errors less than 20 pcm and 2 pcm respectively (often much less). Compared to the eigenvalue estimates of the full-order models—which disagree with the OpenMC reference (and each other) on the order of hundreds of pcm—only between ten and twenty modes are required to reduce the ROM’s error to the point where it is essentially negligible relative to that incurred by cross section condensation. Moreover, by comparing a fine-group (SHEM-361) PGD ROM to a coarse-group (CASMO-SH-70) full-order model, we found the former surpasses the latter’s accuracy—as measured by the error in \( \psi, \phi, \) and \( k \)—between roughly ten and twenty modes. Given these results, we expect this PGD ROM may achieve considerable computational savings in fine-group neutron transport simulations. Additionally, PGD model order reduction may prove preferable to cross section condensation, delivering comparable or superior precision—at a comparable or lesser runtime—without a loss of resolution or a required reference solution.

5.6.1 Future Work

Regarding both the fixed-source and eigenvalue PGD ROMs of the previous and present chapter, future research could seek to enhance performance by improvements to the nonlinear solver, adaptive tolerances [197], and diffusion preconditioning for the spatio-angular operators. Additionally, if a suitable guess for the energy modes were available—such as from the
SVD of a reference solution like those employed in cross section condensation—this could dramatically reduce the number of nonlinear iterations required. Moreover, if this guess is sufficiently accurate, one could even obviate the nonlinear iteration entirely and render the computation of the energy modes an exclusively “offline” cost—perhaps enabling the use of even finer energy meshes. Such \textit{a posteriori} strategies would presumably be incompatible with Progressive PGD, but not closely related ROMs like Subspace PGD [9] and POD.

Furthermore, we have so far limited our numerical experiments to LWR pin-cells with fresh fuel and 70 to 361 energy groups. Expanding the scope to more complicated problems—involving assemblies or cores, depleted fuel, burnable absorbers, finer energy meshes, and so on—would empirically indicate whether a similar convergence with $M$ can be expected or, if not, what new and/or larger-scale features the PGD ROM struggles to resolve. Undertaking this work seems conceptually straightforward, but would at some point necessitate an optimized, parallel implementation of the ROM (and an available computing cluster).

Finally, this application of PGD is but one of many possibilities. Future researchers could, in addition to energy, separate out further physical (spatial and/or angular) or parametric (geometry, composition, and so on) dimensions. Doing so may yield an even greater improvement over the full-order model, so long as the solution admits a low-rank tensor decomposition compatible with one’s chosen model.
6.1 Motivation

Given that even 2D neutron transport calculations can require multiple hours on dozens of processors, the computational burden may well be overwhelming in 3D. This is especially true for nuclear reactors, which tend to be tall, though geometrically simple, in the axial, or \( z \), dimension. To exploit this, as well as the short neutron mean-free-paths observed in LWRs, many 2D/1D methods have been proposed in the reactor physics community, notably those applied within the CRX [10], DeCART [11], and MPACT [2] transport solvers. Each relies on various engineering assumptions—generally, some homogenization of 2D unit-cells and a spatio-angular approximation of transverse leakage—to decouple the radial and axial dimensions, apart from some iterative procedure. While doing so renders 3D full-core transport tractable, the trade-off is some indeterminate error, numerical instability in certain circumstances,\(^{20}\) and limited applicability in the presence of large flux gradients [12].

Inspired by, but methodologically distinct from, these 2D/1D approaches, we propose two analogous ROMs of neutron transport by Proper Generalized Decomposition (PGD) as a means of decoupling radial and axial dimensions of neutron transport. First, we consider axial PGD, where the neutron flux \( \psi \) is separated as an expansion of \( M \) radial and axial modes, \( R_m^{(\mu)} \) and \( Z_m \) respectively,

\[
\psi(\vec{r}, z, \vartheta, \mu, \omega, E) \approx \sum_{m=1}^{M} R_m^{(\mu)}(\vec{r}, \mu, \omega, E)Z_m(z, \vartheta, E),
\]

and second, axial-polar PGD, which is identical except that the axial mode \( Z_m^{(\mu)} \) is now dependent on polar angle \( \mu \) (and the radial mode \( R_m \) is not)

\[
\psi(\vec{r}, z, \mu, \omega, E) \approx \sum_{m=1}^{M} R_m(\vec{r}, \omega, E)Z_m^{(\mu)}(z, \mu, E),
\]

for radial and axial positions \( \vec{r} = (x, y) \) and \( z \), azimuthal angle \( \omega \), and energy \( E \). Lastly,\(^{20}\) Though Collins et al. [2] show this can be rectified by a combination of Coarse Mesh Finite Difference (CMFD) acceleration and adaptive, energy dependent relaxation factors, as implemented in MPACT.

\[^{20}\text{Though Collins et al. [2] show this can be rectified by a combination of Coarse Mesh Finite Difference (CMFD) acceleration and adaptive, energy dependent relaxation factors, as implemented in MPACT.} \]
\( \vartheta \in \{-1, +1\} \) denotes the axial direction, up or down. This is usually implied by the sign of \( \mu \) (positive or negative) where \( \mu \in [-1, +1] \), but we here find it necessary to restrict \( \mu \) to \([0, 1]\) and introduce the extra coordinate \( \vartheta \) in the axial (but not polar-axial) PGD, as is explained in Section 6.2.2. Additionally, while Equations 6.1 and 6.2 represent “group-wise” 2D/1D decompositions, in that both axial and radial modes are functions of energy, we also consider the alternative that \( E \) could be solely ascribed to either of the two. As such, we present six candidate 2D/1D PGD ROMs: namely, axial or axial-polar PGD, wherein energy is either shared between the axial and radial modes or else is separated out from one of the two. These first two categories, axial- and axial-polar, are synonymously denoted as 2D(\( \mu \))/1D and 2D/1D(\( \mu \)) where convenient.

Doing so, we have decoupled the radial and axial dimensions, albeit apart from the nonlinear (fixed-point) iteration outlined in Section 6.2.7. Importantly, we have done so with only a single assumption: that \( \psi \) can be adequately approximated with a tractable number of modes \( M \). This implies, also, that as \( M \) grows large our approximation should converge to the true 3D solution.\(^{21}\) This is not the case for 2D/1D methods, meanwhile, in that adding additional 2D “planes,” for instance, will not guarantee convergence to the 3D flux, since other approximations remain. Likewise, as we do not rely on any properties of LWRs—save for the assumption of extruded geometry—our method is equally applicable to other reactors (graphite-moderated, fast, and so on) and applications outside of reactor physics.

### 6.2 Methodology

As in other chapters, we begin with the fixed-source neutron transport equation, reproduced below for convenience. For brevity, we assume the scattering operator is already

\[^{21}\text{This, of course, supposes that the approximation does converge, which is not theoretically guaranteed (see [198]), but is often found in practice, including in the numerical results of Section 6.3.}\]
expanded by spherical harmonics $Y_{\ell,m}$ up to degree $L$, yielding

$$
\sqrt{1 - \mu^2} \left( \cos(\omega) \frac{\partial}{\partial x} + \sin(\omega) \frac{\partial}{\partial y} \right) \psi(\vec{r}, z, \vec{\Omega}, E)
+ \mu \frac{\partial}{\partial z} \psi(\vec{r}, z, \vec{\Omega}, E) + \Sigma_t(\vec{r}, z, E) \psi(\vec{r}, z, \vec{\Omega}, E)
- \sum_{\ell=0}^{L} \frac{2\ell + 1}{4\pi} \sum_{k=-\ell}^{\ell} Y_{\ell,k}(\vec{\Omega}) \int_{E_0}^{E} \Sigma_{s,\ell}(\vec{r}, z, \vec{\Omega}', E' \rightarrow E) \int_{4\pi} Y_{\ell,k}(\vec{\Omega}') \psi(\vec{r}, z, \vec{\Omega}', E') d\Omega' dE'
= q(\vec{r}, z, \vec{\Omega}, E)
$$

where again $\Sigma_t$ and $\Sigma_s$ are macroscopic total and scattering cross sections, $q$ is a neutron source, and $\vec{\Omega}$ is a direction on the unit sphere defined by angular coordinates $(\mu, \omega)$. The procedure of deriving separated models (to be solved for radial and axial modes) then begins by expanding each coefficient and the source analogously to $\psi$ in Equation 6.1 or 6.2, thereby rendering Equation 6.3 separable in the sense required by either axial- or axial-polar PGD.

### 6.2.1 Separation of Coefficients

Let us divide the 3D geometry into $J$ axial layers $H_j$ and $I$ radial areas $A_i$. Naturally, our cross sections may be written

$$
\Sigma_t(\vec{r}, z, E) = \sum_{j=1}^{J} \sum_{i=1}^{I} 1_{A_i}(\vec{r}) 1_{H_j}(z) \Sigma_{t,i,j}(E)
$$

$$
\Sigma_s(\vec{r}, z, E' \rightarrow E) = \sum_{j=1}^{J} \sum_{i=1}^{I} 1_{A_i}(\vec{r}) 1_{H_j}(z) \Sigma_{s,\ell,i,j}(E' \rightarrow E)
$$

where $1$ is simply an indicator function

$$
1_{A_i}(\vec{r}) = \begin{cases} 
1, & \vec{r} \in A_i \\
0, & \text{otherwise}
\end{cases}, \quad 1_{H_j}(z) = \begin{cases} 
1, & z \in H_j \\
0, & \text{otherwise}
\end{cases}
$$

In other words, the cross sections in axial layer $j$ and radial area $i$ are constant, as visualized in Figure 6.1. While this assumption does limit the complexity of geometry which can be considered by our PGD method, many reactor physics problems can be expressed in this form, and it is commonly encountered in other neutron transport software (for example, PROTEUS-MOC [199] and MPACT [2]).
Further, let us define the real spherical harmonics

$$Y_{\ell,m}(\mu, \omega) \equiv P_{\ell}^{m} |m| \mu W_{\ell,m}(\omega)$$

(6.7)

where $P_{\ell}^{m}$ is the associated Legendre polynomial of degree $\ell$ and order $|m|$ and

$$W_{\ell,m}(\omega) \equiv \begin{cases} 
(-1)^m \sqrt{2} \sqrt{\frac{2\ell+1}{4\pi}} \frac{(\ell-|m|)!}{(|m|)!} \sin(|m|\omega) & m < 0 \\
\sqrt{\frac{2\ell+1}{4\pi}} & m = 0 \\
(-1)^m \sqrt{2} \sqrt{\frac{2\ell+1}{4\pi}} \frac{(\ell-m)!}{(\ell+m)!} \cos(m\omega) & m > 0
\end{cases}$$

(6.8)

from which it is clear one can easily separate the polar and azimuthal dependencies (as required for axial-polar PGD). An additional step is needed for axial PGD, however, in that we have substituted $\vartheta \mu$ for $\mu$, but require $\vartheta$ and $\mu$ to be separated. This is overcome easily by noting

$$P_{\ell}^{m}(\vartheta \mu) = \vartheta^{\ell+m} P_{\ell}^{m} |\mu|, \quad \forall \mu \in [0, 1], \vartheta = \pm 1$$

(6.9)

which effectively accounts for the parity of $P_{\ell}^{m}$ (either even or odd).
6.2.2 Separation of Axial Space from Polar Angle

Although the axial position $z$ and polar angle $\mu$ appear to be separable in Equation 6.3, attempting to separate these variables (without introducing $\vartheta$) would render it impossible to impose the proper boundary conditions on the axial mode, $Z_m$. The reason is that neutron transport is essentially an advective equation, and so the boundary conditions require some prescription of the incoming flux on the problem domain. However, at the axial (that is, top or bottom) boundaries, the polar angle dictates whether the flux is incoming (known) or outgoing (unknown); specifically, this is determined by the sign of $\mu$, either positive or negative. Plainly, without this information, there appears no means of computing $Z_m$. The straightforward resolution is to substitute $\mu \leftarrow \vartheta \mu$, where $\vartheta = \pm 1$ is the axial direction, and to restrict $\mu$ from the original domain $[-1, +1]$ to $[0, 1]$. Doing so, we arrive at a slightly modified equation of neutron transport

\begin{equation}
\sqrt{1 - \mu^2} \left( \cos(\omega) \frac{\partial}{\partial x} \psi(\vec{r}, \vec{\Omega}, E) + \sin(\omega) \frac{\partial}{\partial y} \psi(\vec{r}, \vec{\Omega}, E) \right) + \vartheta \mu \frac{\partial}{\partial z} \psi(\vec{r}, \vec{\Omega}, E) \\
+ \sum_{\ell=0}^{L} \frac{2\ell + 1}{4\pi} \sum_{k=-\ell}^{\ell} \vartheta^{\ell+k} Y_{\ell,k}(\vec{\Omega}) \int_{E_0}^{E_G} \Sigma_{\ell}(\vec{r}, E') dE' = q(\vec{r}, \vec{\Omega}, E)
\end{equation}

which is amenable to separation by axial PGD. Meanwhile, this issue does not arise in axial-polar PGD since in that case the axial and polar dimensions are not separated. These two different problem domains, including the requisite change of coordinates, are visualized in Figure 6.2, where

\begin{equation}
Z_m(z, \vartheta) \equiv \begin{cases} 
    Z_+^m(z), & \vartheta = +1 \\
    Z_-^m(z), & \vartheta = -1,
\end{cases}
\end{equation}

while Figure 6.3 illustrates the corresponding angular quadratures.

Notably, this approach was first devised by Chinesta et al. [139] and subsequently applied to neutron transport with collision and scattering by Domesney and Ji [140], as described in Appendix I (including verification against an analytical benchmark). As the present case of 2D($\mu$)/1D PGD is similar, except that we no longer simplify out the azimuthal angle and 2D space, we can proceed with confidence to this more general scenario.
The derivation continues by substituting the separated representation of the flux—either that of axial or axial-polar PGD, Equation 6.1 or 6.2—and a similarly separated expansion of the source, either

\[ q(\vec{r}, z, \mu, \omega, E) = \sum_{f=1}^{F} q_{f}^{2D(\mu)}(\vec{r}, \mu, \omega, E)q_{f}^{1D}(z, \vartheta, E) \]  
(6.12)

or

\[ q(\vec{r}, z, \vartheta, \omega, E) = \sum_{f=1}^{F} q_{f}^{2D}(\vec{r}, \vartheta, \omega, E)q_{f}^{1D(\mu)}(z, \mu, E) \]  
(6.13)

respectively, into Equation 6.10 or 6.3. This, along with the separated cross sections and spherical harmonics, yields a statement of neutron transport which is compatible with 2D/1D
PGD. Namely, that for the axial, 2D(µ)/1D, case,

\[
\sum_{m=1}^{M} \left[ \sqrt{1 - \mu^2} Z_m(z, \vartheta, E) \times \left( \cos(\omega) \frac{\partial}{\partial x} R_{m}(\mu)(\vec{r}, \vec{\Omega}, E) + \sin(\omega) \frac{\partial}{\partial y} R_{m}(\mu)(\vec{r}, \vec{\Omega}, E) \right) + \mu \frac{\partial}{\partial z} Z_m(z, \vartheta, E) \times R_{m}(\mu)(\vec{r}, \vec{\Omega}, E) \right.
\]

\[
+ \sum_{j=1}^{J} \sum_{l=1}^{L} \Sigma_{j,l,i}(E) \times 1_{H_{j}}(z) Z_m(z, \vartheta, E) \times 1_{A_{i}}(\vec{r}) R_{m}(\vec{r}, \vec{\Omega}, E) \times 1_{H_{j}}(z) Z_m(z, \vartheta, E) \times 1_{A_{i}}(\vec{r}) R_{m}(\vec{r}, \vec{\Omega}, E) \times 1_{H_{j}}(z) Z_m(z, \vartheta, E) \times 1_{A_{i}}(\vec{r}) R_{m}(\vec{r}, \vec{\Omega}, E)
\]

\[
- \sum_{\ell=0}^{L} \frac{2\ell + 1}{4\pi} \sum_{k=-\ell}^{\ell} \sum_{j=1}^{J} \sum_{i=1}^{I} \int_{E_0}^{E'_0} dE' \Sigma_{s,\ell,i,j}(E' \rightarrow E) R_{m}(\mu)(\vec{r}, \vec{\Omega}', E') d\Omega'
\]

\[
= \sum_{j=1}^{F} q_{j}^{2D(\mu)}(\vec{r}, \vec{\Omega}, E) q_{j}^{1D}(z, \vartheta, E),
\]

and likewise in the axial-polar, 2D/1D(µ), case

\[
\sum_{m=1}^{M} \left[ \sqrt{1 - \mu^2} Z_{m}^{(\mu)}(z, \mu, E) \times \left( \cos(\omega) \frac{\partial}{\partial x} R_{m}(\mu)(\vec{r}, \omega, E) + \sin(\omega) \frac{\partial}{\partial y} R_{m}(\mu)(\vec{r}, \omega, E) \right) + \mu \frac{\partial}{\partial z} Z_{m}^{(\mu)}(z, \mu, E) \times R_{m}(\mu)(\vec{r}, \omega, E) \right.
\]

\[
+ \sum_{j=1}^{J} \sum_{l=1}^{L} \Sigma_{j,l,i}(E) \times 1_{H_{j}}(z) Z_{m}^{(\mu)}(z, \mu, E) \times 1_{A_{i}}(\vec{r}) R_{m}(\vec{r}, \chi, E) \times 1_{H_{j}}(z) Z_{m}^{(\mu)}(z, \mu, E) \times 1_{A_{i}}(\vec{r}) R_{m}(\vec{r}, \chi, E)
\]

\[
- \sum_{\ell=0}^{L} \frac{2\ell + 1}{4\pi} \sum_{k=-\ell}^{\ell} \sum_{j=1}^{J} \sum_{i=1}^{I} \int_{E_0}^{E'_0} dE' \Sigma_{s,\ell,i,j}(E' \rightarrow E) R_{m}(\mu)(\vec{r}, \chi, E') d\Omega'
\]

\[
= \sum_{j=1}^{F} q_{j}^{2D(\mu)}(\vec{r}, \chi, E) q_{j}^{1D(\mu)}(z, \mu, E).
\]

Despite the change of notation, these equations remain identical to the original formulation in Equation 6.3 except for the approximation of the solution. That said, the source \( q \) may be approximated as well for convenience, depending on whether it is analytically available in a separated form or, if not, whether it is impractical to retain the full-rank decomposition (as
computed by SVD or other methods). From this starting point, the PGD “sub-”models—the 2D(µ), 2D, 1D, and 1D(µ) equations which are solved for individual radial or axial modes—are determined by a Galerkin projection, as detailed in the following sections. In short, this refers to multiplying these equations by a particular test function and integrating over all variables of which the mode is not a function. Having found and discretized such sub-models, PGD proceeds by a two-level iterative procedure—described in Section 6.2.7—consisting of an outer loop which “enriches” the reduced basis by computing the next term of the series, and an inner loop which repeatedly solves each of the (nonlinearly-coupled) sub-models in sequence until converging on a corresponding pair of modes. Or, to be precise, this describes the typical Progressive Galerkin PGD enacted here, of which various generalizations and enhancements can be found elsewhere, including Chapters 4 and 5.

6.2.4 Axial-Polar Decomposition, 2D/1D(µ)

For simplicity, though without loss of generality, we here assume the multigroup discretization is applied in energy. Moreover, we detail only the most general case in full, where both sub-models are multigroup; simplifying either to the energy-independent case is straightforward, as demonstrated in Section 6.2.6. As will be seen, the resultant equations—in either the 2D/1D(µ) or 2D(µ)/1D case—can easily be recognized as typical formulas of neutron transport in 1D and 2D geometry, notwithstanding certain peculiarities of the cross sections, source, and angular domains. The following sections discuss the 2D and 1D(µ) sub-models of the axial-polar PGD; those of 2D(µ)/1D PGD are then outlined in Section 6.2.5.

6.2.4.1 Radial Submodel, 2D

Consider the separable neutron transport model, Equation 6.3, acted on by

$$\int_{0}^{h} \int_{-1}^{+1} \bullet Z_{m^*,g}^{(µ)}(z, µ) dµ dz \equiv \left( Z_{m^*,g}^{(µ)}, \bullet \right)_{H,µ}. \quad (6.16)$$

The cross sections in each radial area $A_{i}$ can then be written as

$$\bar{\Sigma}^{(m^*,m)}_{t,g,i} \equiv \sum_{j=1}^{J} \left( Z_{m^*,g}^{(µ)}, Z_{m,g}^{(µ)} \right)_{H,µ} \Sigma_{t,g,i,j} \quad (6.17)$$

$$\bar{\Sigma}^{(m^*,m)}_{s,t,k,g'\rightarrow g,i} \equiv \frac{1}{2} \left( Z_{m^*,g}^{(µ)}, P_{k}^{[kl]} \int_{-1}^{+1} P_{k}^{[kl]} Z_{m,g'}^{(µ)} dµ' \right)_{H,µ} \Sigma_{s,t,k,g'\rightarrow g,i,j}. \quad (6.18)$$
which plainly amount to flux-weighted sums over all layers $H_j$. Naturally, these constants appear in the separated expansion of the radial cross sections

$$
\bar{\Sigma}_{t,g}(\vec{r}) \equiv \sum_{i=1}^{I} A_i(\vec{r}) \bar{\Sigma}_{t,g,i}^{(m^*,m)} + \left( Z_{m^*,g}^{(\mu)} \mu \frac{\partial}{\partial z} Z_{m,g}^{(\mu)} \right)_{H,\mu}, \quad (6.19)
$$

$$
\bar{\Sigma}_{s,t,k,g' \rightarrow g}(\vec{r}) \equiv \sum_{i=1}^{I} A_i(\vec{r}) \bar{\Sigma}_{s,t,k,g' \rightarrow g}^{(m^*,m)}, \quad (6.20)
$$

which, in effect, simply serve to select the appropriate cross-section $i$, with an additional term in the total cross-section $\bar{\Sigma}_{t,g}^{(m^*,m)}$ to account for transverse (axial) leakage. For brevity, let us introduce the coplanar streaming coefficient

$$
s_{2D}^{(m^*,m)} \equiv \left( Z_{m^*,g}^{(\mu)} \sqrt{1 - \mu^2 Z_{m,g}^{(\mu)}} \right)_{H,\mu} \quad (6.21)
$$

such that we can write our radial operator as

$$
B_{2D,g}^{(m^*,m)} R_{m,g} \equiv
s_{2D}^{(m^*,m)} \left( \sin(\omega) \frac{\partial}{\partial x} R_{m,g}(\vec{r},\omega) + \cos(\omega) \frac{\partial}{\partial y} R_{m,g}(\vec{r},\omega) \right) + \bar{\Sigma}_{t,g}^{(m^*,m)}(\vec{r}) R_{m,g}(\vec{r},\omega)
$$

$$
- \sum_{\ell=0}^{L} \frac{2\ell + 1}{2\pi} \sum_{k=-\ell}^{k} W_{\ell,k}(\omega) \sum_{g'=1}^{G} \bar{\Sigma}_{s,t,k,g' \rightarrow g}^{(m^*,m)}(\vec{r}) \int_{0}^{2\pi} W_{\ell,k}(\omega') R_{m,g'}(\vec{r},\omega') d\omega'. \quad (6.22)
$$

Meanwhile, we introduce a term for the effective source

$$
q_{2D,g}^{(m^*)}(\vec{r},\omega) \equiv \sum_{f=1}^{F} \left( Z_{m^*,g}^{(\mu)} q_{f,g}^{1D(\mu)} \right)_{H,\mu} q_{f,g}^{2D}(\vec{r},\omega). \quad (6.23)
$$

Doing so, we achieve a reduced, 2D version of the original transport equation which may be solved for radial mode $R_{m^*,g}$

$$
B_{2D,g}^{(m^*,m^*)} R_{m^*,g} = q_{2D,g}^{(m^*)} - \sum_{m \neq m^*}^{M} B_{2D,g}^{(m^*,m)} R_{m,g} = \bar{r}_{2D,g}^{(m^*)} \quad (6.24)
$$

where $\bar{r}_{2D,g}^{(m^*)}$ is a residual term—that is, the source minus the contribution of other modes. For convenience, the coefficient of the streaming term can be canceled by amending our
such that Equation 6.24 becomes identical to a conventional 2D fixed-source neutron transport problem save for three details: first, the cross sections and source are redefined; second, the polar angle is absent; and third, the atypical scattering kernel (for $\ell > 0$). These peculiarities aside, this equation can be implemented and solved numerically via the same discretizations and algorithms used in the full-order model.

6.2.4.2 Axial-Polar Submodel, 1D($\mu$)

The derivation for the axial-polar equation proceeds in much the same manner. One applies

$$
\int_{\mathcal{A}} \int_{2\pi} \mathbf{R}_{m^*}(\vec{r}, \omega, E)d\omega d\mathbf{r} dE \equiv (\mathbf{R}_{m^*}, \bullet)_{\Delta, \omega}
$$

(6.26)

to Equation 6.3, rendering the cross sections for each layer $j = 1 \ldots J$ equal to

$$
\tilde{\Sigma}_{t,g,j}^{(m^*,m^*)} = \sum_{i=1}^{I} (\mathbf{R}_{m^*,g}, \mathbf{R}_{m,g})_{\Delta,\omega} \tilde{\Sigma}_{t,i,j}^{(m^*,m^*)},
$$

(6.27)

$$
\tilde{\Sigma}_{s,m^*,\ell,k,g'\rightarrow g,j}^{(m^*,m^*)} = \sum_{i=1}^{I} \frac{1}{2\pi} \left( \mathbf{R}_{m^*,g}, W_{\ell,k} \int_{0}^{2\pi} W_{\ell,k} R_{m,g'} d\omega' dE' \right)_{\Delta,\omega} \tilde{\Sigma}_{s,i,g'\rightarrow g,i,j}.
$$

(6.28)

These constants are incorporated into $z$-dependent cross sections

$$
\tilde{\Sigma}_{t,g,j}^{(m^*,m^*)}(z, \mu) \equiv \sum_{j=1}^{J} \mathbf{1}_{H_j}(z) \tilde{\Sigma}_{t,g,j}^{(m^*,m^*)}
$$

(6.29)

$$
\tilde{\Sigma}_{s,m^*,\ell,k,g'\rightarrow g,j}^{(m^*,m^*)}(z) \equiv \sum_{j=1}^{J} \mathbf{1}_{H_j}(z) \tilde{\Sigma}_{s,m^*,\ell,k,g'\rightarrow g,j}^{(m^*,m^*)},
$$

(6.30)

where we find the effective total cross section $\tilde{\Sigma}_{t,g,j}^{(m^*,m^*)}$ is necessarily anisotropic, by virtue of the fact that the transverse (radial) leakage term contained within depends on polar angle. This is unusual, but poses no great practical difficulties. Then, given the coaxial streaming
coefficient,
\[ s_{1D(\mu),g}^{(m^*,m)} \equiv (R_{g,m^*}, R_{g,m^*})_{A,\omega}, \]  
(6.31)
the axial-polar operator can be written
\[ B_{1D(\mu),g}^{(m^*,m)} Z_{m^*,g}^{(\mu)} \equiv s_{1D(\mu),g}^{(m^*,m)} \frac{\partial}{\partial z} Z_{m^*,g}^{(\mu)}(z,\mu) + \tilde{\Sigma}_t,g^{(m^*,m)}(z,\mu) Z_{m^*,g}^{(\mu)}(z,\mu) \]
\[ - \sum_{\ell=0}^{L} \frac{2\ell + 1}{2} \sum_{k=0}^{\ell} P^{(k)}_{\ell}((\mu)) \sum_{g'} G_{s,t,k,g'\rightarrow g}^{(m^*,m)}(z) \int_{-1}^{1} P^{(k')}_{\ell}(\mu') Z_{m^*,g'}^{(\mu)}(z,\mu') d\mu'. \]
(6.32)
Having done so, we now find a reduced, 1D neutron transport equation for the axial-polar mode \( Z_{m^*,g}^{(\mu)} \)
\[ B_{1D(\mu),g}^{(m^*,m)} Z_{m^*,g}^{(\mu)} = \tilde{q}_{1D(\mu),g}^{(m^*)} - \sum_{m \neq m^*} B_{1D(\mu),g}^{(m^*,m)} Z_{m,g}^{(\mu)} \]
\[ = \tilde{r}_{1D(\mu),g}^{(m^*)} \]
(6.33)
where the effective source is given by
\[ \tilde{q}_{1D(\mu),g}^{(m^*)}(z,\mu) \equiv \sum_{f=1}^{F} \left( R_{m^*,g}, q_{f,g}^{2D} \right)_{A,\omega} q_{1D(\mu),g}^{(m^*)}(z,\mu). \]
(6.34)
As before, the streaming coefficient can be canceled by substituting
\[ \tilde{\Sigma}_t,g^{(m^*,m)}(z,\mu) \leftarrow \tilde{\Sigma}_t,g^{(m^*,m)}(z,\mu)/s_{1D(\mu),g}^{(m^*,m)}; \]
\[ \tilde{\Sigma}_{s,t,k,g'\rightarrow g}^{(m^*,m)}(z) \leftarrow \tilde{\Sigma}_{s,t,k,g'\rightarrow g}^{(m^*,m)}(z)/s_{1D(\mu),g}^{(m^*,m)}; \]
\[ \tilde{r}_{1D(\mu),g}^{(m^*)}(z,\mu) \leftarrow \tilde{r}_{1D(\mu),g}^{(m^*)}(z,\mu)/s_{1D(\mu),g}^{(m^*,m)}; \]
(6.35)
to render Equation 6.33 a 1D fixed-source neutron transport problem which, besides the specification of cross sections and source, is unusual only in the presence of an anisotropic transverse leakage term and the inclusion of spherical harmonic moments other than \( k = 0 \). Specifically, the latter detail is unusual because the 1D solution is typically (by definition) azimuthally constant, such that these moments equal zero when integrated in \( \omega \) over [0,2\pi].
6.2.5 Axial Decomposition, 2D(\(\mu\))/1D

Likewise, to achieve the axial (rather than axial-polar) decomposition, we start from Equation 6.10, in which we have restricted \(\mu\) from \([-1, +1]\) to \([0, 1]\) and introduced an additional coordinate \(\vartheta \in \{-1, +1\}\) to represent the axial streaming direction (up or down). As explained in Section 6.2.2, this is necessary to separate \(z\) and \(\mu\) by PGD, and so is required if \(\mu\) is to be assigned to the 2D sub-model, which is the defining feature of the axial decomposition. Apart from this distinction, the derivation proceeds similarly as in the preceding case of 2D/1D(\(\mu\)) decomposition.

6.2.5.1 Radial-Polar Submodel, 2D(\(\mu\))

To achieve the radial-polar, or 2D(\(\mu\)), sub-model, we act on Equation 6.10 by

\[
\sum_{\vartheta = \pm 1} \int_{0}^{h} \bullet Z_{m^*,g}(z, \vartheta) dz \equiv (Z_{m^*,g}, \bullet)_{H, \pm}
\]  

(6.36)

and subsequently define the axially-integrated, flux-weighted cross sections for each radial area \(A_i\)

\[
\Sigma_{t,g,i}^{(m^*,m)} \equiv \sum_{j=1}^{J} (Z_{m^*,g}, Z_{m,g})_{H, \pm} \Sigma_{t,i,j} (E),
\]  

(6.37)

\[
\Sigma_{s,l,k,g' \rightarrow g}^{(m^*,m)} \equiv \sum_{j=1}^{J} \frac{1}{2} \left( Z_{m^*,g}, \vartheta^\ell+k \sum_{\vartheta' = \pm 1} \vartheta'^{\ell+k} Z_{m,g'} \right)_{H, \pm} \Sigma_{s,l,g' \rightarrow g,i,j}.
\]  

(6.38)

Further, by evaluating the summations over \(\vartheta\) and \(\vartheta'\), the latter inner product can be simplified as

\[
\left( Z_{m^*,g}, \vartheta^\ell+k \sum_{\vartheta' = \pm 1} \vartheta'^{\ell+k} Z_{m,g'} \right)_{H, \pm} \rightarrow \left( Z_{m^*,g}^p, Z_{m,g'}^p \right)_{H_j}
\]  

(6.39)

where \(Z_{m,g}^p\) is the even- or odd-parity axial flux,

\[
Z_{m,g}^p(z) \equiv \begin{cases} Z_{m,g}^+(z) \equiv Z_{m,g}^+(z) + Z_{m,g}^-(z) & \ell + k \text{ is even} \\ Z_{m,g}^o(z) \equiv Z_{m,g}^+(z) - Z_{m,g}^-(z) & \ell + k \text{ is odd} \end{cases}
\]  

(6.40)
Again, to cancel the streaming coefficient, we can further redefine

$$
\bar{\Sigma}_{t,g}(\vec{r}, \mu) \equiv \sum_{i=1}^{I} 1_{A_i}(\vec{r}) \Sigma^{(m^*,m)}_{t,g,i}(\vec{r}, \mu) + \mu \left( Z_{m^*,g}, \vartheta \frac{\partial}{\partial \vartheta} Z_{m,g} \right)_{H,\pm},
$$

(6.41)

$$
\bar{\Sigma}_{s,t,k,g'\rightarrow g}(\vec{r}) \equiv \sum_{i=1}^{I} 1_{A_i}(\vec{r}) \Sigma^{(m^*,m)}_{s,t,k,g'\rightarrow g,i},
$$

(6.42)

where, as in the axial-polar case, the total cross section \(\bar{\Sigma}^{(m^*,m)}_{n,g}\) is a function of \(\mu\) because the transverse (axial) leakage is dependent on polar angle. Using these terms, plus a coplanar streaming coefficient,

$$
s^{(m^*,m)}_{2D(\mu),g} \equiv (Z_{m^*,g}, Z_{m,g})_{H,\pm}
$$

(6.43)

we define \(B^{(m^*,m)}_{2D(\mu)}\), the radial-polar operator

$$
B^{(m^*,m)}_{2D(\mu),g} R^{(\mu)}_{m^*,g} = s^{(m^*,m)}_{2D(\mu),g} \vec{\Omega} \cdot \nabla R^{(\mu)}_{m^*,g}(\vec{r}, \vec{\Omega}) + \bar{\Sigma}^{(m^*,m)}_{t,g}(\vec{r}, \mu, E) R^{(\mu)}_{m^*,g}(\vec{r}, \vec{\Omega})
$$

$$
- \sum_{\ell=0}^{L} \frac{2\ell + 1}{2\pi} \sum_{k=-\ell}^{\ell} Y_{\ell,k}(\vec{\Omega}) \sum_{g'g} \bar{\Sigma}^{(m^*,m)}_{s,t,k,g'\rightarrow g}(\vec{r}) \int_{2\pi} Y_{\ell,k}(\vec{\Omega}') R^{(\mu)}_{m^*,g'}(\vec{r}, \vec{\Omega}') d\Omega'
$$

(6.44)

which allows us to succinctly express our equation for \(R^{(\mu)}_{m^*,g}\)

$$
B^{(m^*,m)}_{2D(\mu),g} R^{(\mu)}_{m^*,g} = \vec{q}^{(m^*)}_{2D(\mu),g} - \sum_{m \neq m^*} B^{(m^*,m)}_{2D(\mu),g} R^{(\mu)}_{m^*,g}
$$

$$
= \vec{r}^{(m^*)}_{2D(\mu),g}
$$

(6.45)

where

$$
\vec{q}^{(m^*)}_{2D(\mu),g}(\vec{r}, \vec{\Omega}) \equiv \sum_{f=1}^{F} \left( Z_{m^*,g}, q^{1D}_{f,g} \right)_{H,\pm} \vec{q}^{2D(\mu)}_{f,g}(\vec{r}, \vec{\Omega}).
$$

(6.46)

Again, to cancel the streaming coefficient, we can further redefine

$$
\bar{\Sigma}^{(m^*,m^*)}_{t,g}(\vec{r}, \mu) \leftarrow \bar{\Sigma}^{(m^*,m^*)}_{t,g}(\vec{r}, \mu) / s^{(m^*,m)}_{2D(\mu),g},
$$

$$
\bar{\Sigma}^{(m^*,m)}_{s,t,k,g'\rightarrow g}(\vec{r}) \leftarrow \bar{\Sigma}^{(m^*,m)}_{s,t,k,g'\rightarrow g}(\vec{r}) / s^{(m^*,m)}_{2D(\mu),g},
$$

$$
\vec{r}^{(m^*)}_{2D(\mu),g}(\vec{r}, \vec{\Omega}) \leftarrow \vec{r}^{(m^*)}_{2D(\mu),g}(\vec{r}, \vec{\Omega}) / s^{(m^*,m)}_{2D(\mu),g}
$$

(6.47)

yielding an equation nearly identical with the usual statement of 2D fixed-source neutron transport. Specifically, it differs—beyond the redefinition of cross sections and source—in that the total cross section is anisotropic, and in that those spherical harmonics moments
which are odd functions of $\mu$ (that is, where $\ell + k$ is odd) do not integrate to zero. Of course, these moments vanish regardless if the corresponding cross sections becomes zero, such as when there is no net axial current, $\int_0^h Z_{m^*,g}^a(z)dz$ or $\int_0^h Z_{m,g'}^a(z)dz = 0$. Further, while one may expect that solving for only half the range of polar angles—those in $[0,1]$—is unusual, the case is in fact the opposite: since the solution of any 2D problem will be symmetric with respect to $\mu$ (so long as the source is as well), there is rarely a need to solve for both (north and south) hemispheres. As such, this particular aspect of Equation 6.45 should not pose any implementation difficulties.

6.2.5.2 Axial Submodel, 1D

Finding the corresponding axial equation, meanwhile, begins by operating on Equation 6.10 by

$$\int_A \int_{2\pi} \bullet R_{m^*}(\vec{r},\vec{\Omega})d\vec{r}d\Omega \equiv \left(R_{m^*}(\bullet)\right)_{A,2\pi}$$

(6.48)

to yield radially-integrated cross sections for each layer $H_j$

$$\Sigma^{(m^*,m)}_{t,g,j} \equiv \sum_{i=1}^I (R_{m^*,g}, R_{m,g})_{A_i,2\pi} \Sigma_{t,g,i,j},$$

(6.49)

$$\Sigma^{(m^*,m)}_{s,m,\ell,k,g' \rightarrow g,j} \equiv \sum_{i=1}^I \frac{1}{2\pi} \left(R_{m^*,g}, Y_{\ell,k} \int_{2\pi} Y_{\ell,k} R_{m,g'}d\Omega'\right)_{A_i,2\pi} \Sigma_{s,\ell,k,g' \rightarrow g,j},$$

(6.50)

which appear in the $z$-dependent cross sections

$$\bar{\Sigma}^{(m^*,m)}_{t,g}(z) \equiv \sum_{j=1}^J H_t(z) \bar{\Sigma}^{(m^*,m)}_{t,g,j}$$

$$+ \left(R_{m^*,g}, \sqrt{1 - \mu^2} \left(\sin(\omega) \frac{\partial}{\partial x} R_{m,g} + \cos(\omega) \frac{\partial}{\partial y} R_{m,g} \right)\right)_{A,2\pi},$$

(6.51)

$$\bar{\Sigma}^{(m^*,m)}_{s,\ell,k,g' \rightarrow g}(z) \equiv \sum_{j=1}^J H_t(z) \bar{\Sigma}^{(m^*,m)}_{s,\ell,k,g' \rightarrow g,j}.$$  

(6.52)

Introducing yet another (coaxial) streaming coefficient for convenience,

$$s^{(m^*,m)}_{1D,g} \equiv (R_{m^*,g}, \mu R_{m,g})_{A,2\pi}$$

(6.53)
the axial operator becomes succinctly

$$B^{(m^*,m)}_{1D,g} Z_{m^*,g} = s^{(m^*,m)}_{1D,g} \frac{\partial}{\partial z} Z_{g,m^*,}(z, \vartheta) + \tilde{\Sigma}_{t,g}^{(m^*,m)}(z) Z_{m^*,g}(z, \vartheta)$$

$$- \sum_{\ell=0}^{L} \frac{2\ell + 1}{2} \sum_{k=-\ell}^{\ell} \mathcal{G} \sum_{g' = 1}^{G} \tilde{\Sigma}_{s,\ell,k,g'\rightarrow g}^{(m^*,m)}(z) \sum_{\vartheta' = \pm 1}^{1} \vartheta^{\ell+k} Z_{m^*,g'}(z, \vartheta'),$$

from which it follows that the reduced, 1D equation the axial mode for $Z_{m^*,g}$ is

$$B^{(m^*,m)}_{1D,g} Z_{m^*,g} = q^{(m^*)}_{1D,g} - \sum_{m \neq m^*}^{M} B^{(m^*,m)}_{1D,g} Z_{m,g}$$

$$= \tilde{r}^{(m^*)}_{1D,g}$$

in which

$$\tilde{q}^{(m^*)}_{1D,g}(z, \vartheta) = \sum_{f=1}^{F} \left( R^{(m^*,m^*)}_{f,g} \tilde{q}^{2D(m^*)}_{f,g} \right) A_{2\pi} q^{1D}_{f,g}(z, \vartheta).$$

As in previous sections, the streaming coefficient can be canceled by substituting

$$\tilde{\Sigma}_{t,g}^{(m^*,m)}(z) \leftarrow \tilde{\Sigma}_{t,g}^{(m^*,m)}(z)/s^{(m^*,m^*)}_{1D,g},$$

$$\tilde{\Sigma}_{s,\ell,k,g'\rightarrow g}^{(m^*,m)}(z) \leftarrow \tilde{\Sigma}_{s,\ell,k,g'\rightarrow g}^{(m^*,m^*)}(z)/s^{(m^*,m^*)}_{1D,g},$$

$$\tilde{r}^{(m^*)}_{1D,g}(z) \leftarrow \tilde{r}^{(m^*)}_{1D,g}(z)/s^{(m^*,m^*)}_{1D,g}.$$
alternatively be assigned to either the radial or axial modes and separated out of the other. Especially as the number of energy groups increases, this may reduce the computational cost of the ROM substantially. Moreover, as computing the radial modes will generally be much more burdensome than the same for the axial modes, the savings may be even greater in the case where the radial equations are rendered energy-independent.

Of course, these gains will be hampered somewhat in that the 2D/1D decomposition may be appreciably poorer when not localized to each individual energy group—in turn, requiring more modes to be computed to meet a given precision, thereby counteracting, and perhaps outweighing, the aforementioned computational savings. Conceivably, these two factors—the resolution of the energy mesh and the degradation of the decomposition—will be interrelated in reactor physics, as neutrons of similar energies tend to have similar distributions. Accordingly, fluxes in coarse group structures—as in the extreme of two groups, fast and thermal—may have starkly dissimilar features that impede a low-rank 2D/1D decomposition which separates in energy. By contrast, fine structures may contain many groups dominated by the same phenomena (leakage, resonance absorption, and so on), so as to yield fluxes that can be readily approximated by such a factorization, at a rank far lower than that of the full-order solution. As such, one might expect group-wise 2D/1D decomposition to be an economical proposition for few- or coarse-group problems, but increasingly less so beyond some particular level of fidelity. These expectations should be investigated more thoroughly in future work, however, as the numerical experiments presented here each contain only two groups.

In any case, deriving the energy independent equations from their multigroup counterparts is straightforward, especially as the former can be considered a degenerate case of the latter where \( G = 1 \) (that is, there is only one group). The coefficients for this single group, denoted \( \mathcal{G} \) for clarity, can be found by integrating each energy dependent term over \([E_G, E_0]\). In the multigroup approximation, this amounts to a summation, weighted by lethargy widths
Δg as per Section 3.2.2, yielding one-group coefficients

\[
\Delta_g^{(m^*,m)} = \sum_{g=1}^{G} \Delta_g^{-1} s_{ND,G}^{(m^*,m)},
\]

(6.58)

\[
\Sigma_{t,G}^{(m^*,m)} = \sum_{g=1}^{G} \Delta_g^{-1} \Sigma_t^{(m^*,m)},
\]

(6.59)

\[
\Sigma_{s,l,k,G}^{(m^*,m)} = \sum_{g=1}^{G} \Delta_g^{-1} \sum_{g'=1}^{G} \Sigma_{s,l,k,g' \rightarrow g},
\]

(6.60)

and likewise the one-group source,

\[
q_{ND,G}^{(m^*)} = \sum_{g=1}^{G} \Delta_g^{-1} q_{ND,G}^{(m^*)},
\]

(6.61)

where the subscript ND refers to the sub-model from which the energy variable is being separated, either 2D, 1D(µ), 2D(µ), or 1D as in Sections 6.2.4 and 6.2.5. Beyond this, the corresponding mode \((R_m, Z_m^\mu, R_m^\mu, Z_m)\) is redefined as independent of \(g\), as are the source components \(q_{ND}^f\). With these modifications, the multigroup models—Equations 6.24, 6.33, 6.33, and 6.45—simplify to one-group, but otherwise retain all aforementioned particularities, such as the absence of µ or polar anisotropy of the total cross section.

### 6.2.7 Proper Generalized Decomposition Algorithm

Given these 2D and 1D sub-models, where either or both (but not neither) is multigroup, the overall iterative procedure of PGD can be enacted simply. To review, Progressive PGD begins with zero modes, \(M = 0\), and employs a greedy algorithm to find the first pair. To do, some nonlinear iteration is required between the axial and radial sub-models for \(m^* = M = 1\); here, as in most PGD literature, fixed-point (sometimes termed “Alternating Directions”) iteration is chosen for its simplicity and demonstrated efficacy. Upon nonlinear convergence, this process then repeats, “enriching” the reduced basis—as in, incrementing \(M\)—until some numerical tolerance or other termination criterion is met. This procedure is analogous to that detailed in Algorithm 1 in the case of Galerkin PGD.

### 6.2.8 Similarity to Conventional 2D/1D Methods

Despite their independent origin, these 2D/1D PGD models share some similarities to existing 2D/1D methods. Namely, taking the Galerkin projection with respect to the
transverse spatial dimension(s)—that is, applying Equations 6.16, 6.26, 6.36, or 6.48—is essentially equivalent to homogenization by transverse integration as practiced in many 2D/1D methods, except that we introduce a (non-uniform) test function [2]. Likewise, anisotropic leakage terms in the total cross section also appear in MPACT’s 2D/1D method, through a procedure termed “transverse leakage splitting” [2], [12]. Finally, although we do not solve it as such here, the 1D equation of axial PGD is equivalent to a diffusion or \( P_1 \) model (see Appendix G.1), which is the conventional axial discretization in many 2D/1D methods [200].

The main differences, meanwhile, are first that we do not approximate the angular dependence of the flux in either the transverse leakage terms or the other inner products that appear in the cross sections. This is commonly done in conventional 2D/1D methods, where the transverse leakage is assumed to be isotropic (or approximated as a truncated series) and the cross section “homogenization” is applied using the scalar rather than angular flux (which Jarrett [12] and Faure et al. [163] identify as an appreciable source of error). Secondly, PGD seeks multiple modes, which allows it to converge to the 3D solution (hypothetically, if not in practice), while other 2D/1D methods seek only one—in effect, approximating the solution as rank-one separable. Thirdly, and relatedly, conventional 2D/1D methods subdivide the domain into several planes and unit-cells (commonly pin-cells) in order to localize the 2D/1D decomposition. This could be done in PGD as well, but is not strictly necessary since the number of modes can simply be increased until the solution is adequately resolved throughout the domain, as measured by some error indicator. Fourthly, most 2D/1D methods only apply transverse integration in space (\( r \) and \( z \)) and not angle (\( \mu \) and \( \omega \)). Consequently, both the axial and radial fluxes remain functions of \( \mu \) and \( \omega \). This could also be considered in PGD (analogously to the case where both axial and radial modes are multigroup), but we find it more natural to relegate the azimuthal angle \( \omega \) to the 2D domain and the polar angle \( \mu \) to either the 1D or 2D domain, but not both. Finally, we here use the same discretization in the 2D and 1D sub-models—such that we can easily reconstruct the corresponding 3D solution—whereas many 2D/1D models employ the Method of Characteristics in the 2D planes and nodal diffusion in the 1D axes. That said, this is not an essential feature of either method: one could conceivably apply different discretizations to the 2D and 1D sub-models of PGD just as easily as one could use the same discretization for both in traditional 2D/1D methods (see Jarrett’s “1D/1D \( S_N \)” method [12], for example).
6.3 Numerical Results

To characterize performance, we begin by solving the first Takeda benchmark [6], a small homogenized LWR core. Two cases are presented, with or without control rod. Performance is primarily measured by the $L^2$ norm of the error of the angular and scalar fluxes, as compared to a full-order reference solution. Moreover, the ideal decomposition in the $L^2$ norm given by the SVD of the reference solution (see Appendix D for details) is shown for comparison. Results are given both as a function of rank $M$ and, for PGD, runtime—that is, the real or “wall-clock” time elapsed since computation began, not including any requisite pre- and post-processing. The same is not provided for the full-order (3D) calculation, as performance-critical optimizations—most notably, diffusion preconditioning and parallelism—have yet to be implemented in either the full- or reduced-order models, rendering any quantitative comparison of the two premature and perhaps misleading. (That said, the potential for large computational savings is evident by the definition of the ROM and was qualitatively observed in practice.) However, as each of the six PGD ROMs employ the same iterative procedures, numerical tolerances, and overall algorithm, comparison among them appears apt, given any future enhancements—such as PGD’s “update” step, Minimax projection, or a Newton or quasi-Newton nonlinear solver—stand to benefit each. Accordingly, while we do not expect the reported runtimes to be representative in absolute terms (except as an upper bound to be improved upon), the performance of each relative to the others may hold (at least roughly) fixed even as the algorithm and underlying implementation are improved. As such, these results are practically useful, especially in informing the direction of future research.

6.3.1 Takeda Light Water Reactor

The first of the Takeda benchmarks [6]—a small LWR with octant symmetry representing the Kyoto University Critical Assembly (KUCA)—is depicted in Figure 6.1a, where the red, blue, and white volumes correspond to the homogenized core, reflector (water), and control rod channel respectively. This rod channel is filled with either a neutron absorber or void in the first (rodded) and second (unrodded) cases respectively. As per benchmark specifications, energy and spatial discretizations are given by two-group cross sections and a Cartesian grid of uniform 1 cm spacing. In angle, meanwhile, a $4 \times 4$ (that is, 4 polar by 4 azimuthal angles per octant) product Gauss-Legendre-Chebyshev quadrature is chosen,
for the useful property that product quadratures—those defined as tensor products of polar and azimuthal quadratures—permit the angular flux to be reconstructed from the 2D/1D($\mu$) PGD. Likewise, finite elements formed by (spatial) tensor products, like the linear ($p = 1$) discontinuous Lagrangian elements selected here, readily enable reconstruction of the 3D flux from 2D and 1D modes. Because the Takeda benchmarks are specified as criticality (that is, $k$-eigenvalue) problems rather than fixed-source problems, a uniform source is assumed throughout the core; this is typically a crude assumption of the true fission source, but a reasonable guess by which to initialize some iterative eigenvalue procedure. In this sense, we consider the present case sufficiently representative of a prototypical use-case for 2D/1D methods, and proceed to numerical experiments.

In an illustration of the curse of dimensionality (discussed previously in Section 1.1), one finds that this apparently simple benchmark—consisting of 2 energy groups, 128 angles, and $50^3$ finite elements with 8 degrees-of-freedom each—is not trivial to solve numerically, containing 32 million unknowns. This simultaneously highlights the utility of PGD, which reduces this figure to between 40,800 and 320,000 unknowns per mode, bounded by the 2D/1D($\mu, E$) and 2D($\mu, E$)/1D($E$) ROMs. Of course, this advantage is contingent on how readily the 3D solution(s)—shown here in Figure 6.4—admits a low-rank approximation and, secondly, the efficacy of PGD in obtaining the same.

To investigate these questions, we plot the convergence of both SVD and PGD for each of the six decompositions in Figure 6.5. From this, we first observe that a low-rank approximation is possible and that PGD does indeed converge. More specifically, by thirty modes the optimal decomposition of SVD is in all scenarios seen to achieve $L^2$ errors of the angular flux below $2 \times 10^{-3}$. Moreover, the convergence of the SVD is slightly improved in the rodded versus unrodded case, as is that of PGD. This suggests, plausibly, that the features induced by the voided channel—notably a streaming path crossing the axial interface between core and reflector—require more modes to resolve than the local flux depression induced by the rod. In any case, we find the PGD, while not as rapidly convergent as the SVD, does yield an angular flux with an $L^2$ error below 2% in each case at $M = 30$. These results are somewhat superior in the group-wise PGD, decreasing to 0.1% and 0.7% for the rodded and unrodded configurations respectively.

However, while the convergence of each PGD scheme with increasing $M$ is comparable, the same does not hold with respect to runtime. Namely, the 2D($\mu$)/1D PGDs—those
which assign the polar angle to the 2D sub-model—take substantially longer to compute. This aligns with the expectation that the computational cost of 2D/1D PGD will often be dominated by that of calculating the 2D modes. As such, removing $\mu$ from this computation—in this case reducing the number of angles in the 2D plane by a factor of four—yields an appreciable speedup, yet appears to have little adverse effect on the quality of the decomposition. Meanwhile, a second, less severe, discrepancy is observed between the three treatments of energy. As to be expected, for both 2D($\mu$)/1D and 2D/1D($\mu$) schemes, the fastest-running model (per mode) is that where only the 1D modes are multigroup, then that for 2D modes, and finally the group-wise case, where both are multigroup. However, unlike
Figure 6.5: Convergence of 2D/1D Proper Generalized Decomposition (opaque lines) and Singular Value Decomposition (faint lines) for the Takeda Light Water Reactor.

the previous case, this additional runtime is offset by a commensurate increase in accuracy, such that the group-wise PGD generally achieves the lowest error for a given amount of runtime. That said, as noted in Section 6.2.6, group-wise PGD may be most advantageous when there are few energy groups with markedly different features arising in each (as clearly observed in Figure 6.4); given this, the net benefit demonstrated here may be diminished or reversed in finer-group problems. In any case, these numerical results confirm each of the six 2D/1D ROMs to be feasible, convergent, and effective for this prototypical LWR benchmark.

6.4 Conclusion

In this chapter, we devised and implemented six PGD ROMs for 2D/1D decomposition. These are methodologically distinct, but practically analogous to, the conventional 2D/1D methods already popular in reactor physics. However, while the latter rests on engineering assumptions which sometimes impair the accuracy and reliability of the solution, the former
assumes only that a sufficiently precise approximation is achievable in a tractable number of modes $M$. Accordingly, the 2D/1D PGD is hypothetically able to converge to the 3D solution within an arbitrarily small precision (though in practice the convergence may be prohibitively slow), which is not the case in conventional 2D/1D methods.

Moreover, beyond the hallmark features of PGD—namely, progressive construction of a rank-$M$ decomposition by solving a series of low-order, nonlinear systems—five of these six ROMs introduce further novelties not found in previous 2D/1D methods. First, the latter assume that both the 2D and 1D components are functions of polar angle; at least in the case where the axial solver is diffusion, this is analogous to the Axial PGD shown here. However, the Axial-Polar PGD demonstrates that $\mu$ can be separated from $\vec{\Omega}$ analogously to how 2D/1D methods separate $z$ from 3D position $\vec{r}$. Moreover, numerical experiments on the first Takeda benchmark (a small LWR) show this to be practically advantageous, yielding markedly superior precision for an equal amount of wall time. Second, 2D/1D methods traditionally grant that both 2D and 1D equations are energy-dependent (multigroup). Presently, we consider the same, plus the two additional possibilities that $E$ is assigned uniquely to either the axial or radial domain. In this case, numerical performances on the Takeda LWR are comparable, with the “group-wise” decomposition often being the most economical (albeit by a small margin). Nevertheless, it is foreseeable that separating the energy dependence may be preferable for simulations with finer group-structures (the Takeda LWR having only two energy groups).

Given these unique advantages, one might expect 2D/1D PGD ROMs to compare favorably to existing methods. Alternatively, as these latter two principles seem equally applicable to established 2D/1D methods—in that they apparently entail only the extension of “transverse integration” to include dimension other than axial or radial space—future research could seek to incorporate these concepts into more traditional 2D/1D approaches. As either outcome stands to substantially advance the state-of-the-art in 2D/1D modeling of neutron transport, we anticipate further investigation of 2D/1D PGD to be a fruitful direction of research, and view the present results as an encouraging first step in the development and application thereof.

\[22\text{More precisely, none of the PGD ROMs presented here grant that both the radial and axial modes depend on $\mu$. Nevertheless, the 1D—though not 1D($\mu$)—submodel is tantamount to a diffusion, or $P_1$, model, as per Appendix G.1.}\]
6.4.1 Future Work

A clear opportunity for future work would be to enhance the performance of 2D/1D PGD by the same means already demonstrated in Chapters 4 and 5, Minimax PGD and the update step (foreseeably, here of the axial modes). (Other considerations could include applying Subspace PGD [9] or the Arnoldi-like PGD of [196].) Likewise, extending this ROM to $k$-eigenvalue problems—perhaps as in Chapter 5—would allow for much wider applicability, as well as more meaningful validation. These matters aside, the prospects for future research are much as in Section 5.6.1: namely, incorporating prior knowledge of the solution (perhaps here the axial flux or power profile) and implementing the performance-critical optimizations (especially preconditioning and parallelism) necessary to apply these ROMs to HPC applications and perform rigorous performance benchmarks.

Another direction could be to investigate whether features developed for conventional 2D/1D methods could be applied to enhance PGD. These could include 3D CMFD preconditioners, localization of the decomposition (perhaps to pin-cells, assemblies, and/or axial layers), and energy-dependent relaxation factors, for instance [2]. Likewise, direct performance comparisons between the two approaches would be illuminating. That said, computing the requisite inner products for PGD may prove more complicated in the Method of Characteristics, rather than DGFEM, discretization of space. As such, applying and modifying 2D/1D PGD to the former may constitute a separate research objective in and of itself.
CHAPTER 7
NUMERICAL IMPLEMENTATION IN AETHER

7.1 Motivation

To implement the aforementioned PGD ROMs, it was first necessary to obtain a code capable of solving both the ROM and full-order model, for comparison, and which affords enough features to be considered “state-of-the-art.” In particular, the geometric capabilities must extend (at least) to unstructured 2D meshes; otherwise, cylindrical fuel pins could not be accurately represented. Likewise, since many inner products are required for PGD, spatial discretization by FEM was deemed essential, ruling out finite differences and the Method of Characteristics. Moreover, as the PGD ROM may have differing efficacy when considering the first-order (hyperbolic) or second-order (even-parity, among others) formulations of neutron transport, it was decided to focus exclusively on the more conventional first-order form. Discrete ordinates ($S_N$) was preferred (but not required) over spherical harmonics ($P_N$) for the same reason: it is more common in reactor physics applications.

Unfortunately, the prospect of adopting an existing open-source neutron transport code did not prove fruitful under these constraints. The four well-established projects in this category (at time of writing) are OpenMC [177], SCONE [201], OpenMOC [202], and DRAGON [203], which, as two Monte Carlo and two Method of Characteristics codes respectively, would be difficult to adapt for our purposes. Even considering software developed internally by Department of Energy national laboratories, no package exists that fits this niche exactly\footnote{Capsaicin [204] and Teton [205] do offer the correct discretization, but are intended for thermal radiative (not neutron) transport.}, though Idaho National Laboratory’s Rattlesnake [206] and Argonne National Laboratory’s PROTEUS-SN [207] come close in that both support $S_N$ in angle and FEM in space, but consider either predominantly\footnote{Rattlesnake’s transport sweep is described as “precluding deployment for production calculations” due to “current grind time” (p. 66) [208], though this is foreseeably subject to change.} or exclusively second-order transport. In light of these alternatives, the best choice became to develop an original (multigroup, $S_N$, DGFEM) neutron transport code specifically for the present research, titled Aether.

Moreover, noting this gap, the objective in developing Aether was not only to research PGD but also to establish an open-source radiation transport library that can be adapted to
other ends. Particular research applications (including those herein) can then implemented as modifications of the general, full-order model. To support this goal, the library is designed to be modular and object-oriented, such that new features can easily be added and implementation details are well-encapsulated behind public interfaces. Additionally, since this software is written in C++ using the deal.II finite element library [179], many valuable features—parallelism, hp refinement, multigrid methods, and so on—are already available or supported. Moreover, Aether could foreseeably be made equally useful for end-users (analysts) by the implementation of a just a few outstanding features: notably, parallelism and preconditioning. This is beyond the scope of the present research, but notable nonetheless, especially as there is no mature, open-source software available for deterministic neutron transport on unstructured mesh geometry (rather than Constructive Solid Geometry) at time of writing. Given these considerations, we anticipate Aether will prove a valuable contribution to the neutron transport community and may be welcomed by fellow developers.

7.2 Methodology

As represented by the linear systems below, Aether solves the multigroup, fixed-source, neutron transport equation

\[ \mathbb{B} \psi = L^{-1} \mathbf{Bq}, \]  
(7.1)

as well as the generalized,

\[ A \psi = k \mathbb{B} \psi, \]  
(7.2)

or standardized,

\[ \mathbb{B}^{-1} A \psi = k \psi, \]  
(7.3)

\(k\)-eigenvalue problem. As discussed in Section 3.2.2, the multigroup fixed-source operator \(\mathbb{B}\) is naturally expressed as a block matrix

\[ \mathbb{B} \psi \equiv \begin{bmatrix} W_1 & \cdots & S_{g'\to1} & \cdots & S_{G\to1} \\ \vdots & \ddots & \vdots \\ S_{1\to g} & W_g & S_{G\to g} \\ \vdots & \ddots & \vdots \\ S_{1\to G} & \cdots & S_{g'\to G} & \cdots & W_G \end{bmatrix} \begin{bmatrix} \psi_1 \\ \vdots \\ \psi_g \\ \vdots \\ \psi_G \end{bmatrix} \]  
(7.4)
where $W_g$ denotes a within-group block,

$$W_g \equiv I + L_g^{-1}(L_g + S_{g \rightarrow g}), \quad (7.5)$$

and $S_{g' \rightarrow g}$ a group-to-group scattering block,

$$S_{g' \rightarrow g} \equiv -B_g M_g \Sigma_{s,g' \rightarrow g} D_g, \quad (7.6)$$

which are themselves composed of four basic operators, $L_g$, $M_g$, $\Sigma_{s,g' \rightarrow g}$, and $D_g$. (The mass matrix $B_g$ is not considered a distinct operator because it is also contained in $L_g$ and need only be evaluated at the same time as $L_g^{-1}$.) As will be demonstrated, the organization of Aether closely corresponds to this discussion.

### 7.2.1 Fixed-Source Problems

To render our software modular and easily comprehensible, we adopt an object-oriented approach where each matrix is represented by an object, and its operation (vector-multiplication) by a method $\text{vmult}$, or sometimes $\text{step}$. Likewise, the adjoint operations are termed $\text{Tvmult}$ or $\text{Tstep}$. Despite this abstraction, none of our classes are internally implemented as matrices; rather, each merely emulates one’s action. The basic building blocks are:

- **TransportBlock**: $\text{vmult}(x,b)$ which emulates one block Gauss-Seidel step of $L_g$, $x \leftarrow L_g^{-1}(B_g b + L_g x)$
- **ScatteringBlock** which represents $\Sigma_{s,g' \rightarrow g}$
- **DiscreteToMoment** which represents $D_g$
- **MomentToDiscrete** which represents $M_g$

Because each **TransportBlock** and **ScatteringBlock** differ only by their specification of cross sections (and, for the former, boundary conditions) we implement these classes as thin wrappers which simply pass these parameters to:

- **Transport** which emulates one block Gauss-Seidel step of $L_{\bullet}$
- **Scattering** which emulates $\Sigma_{s,\bullet \rightarrow \bullet}$
This is especially important for Transport because it must pre-assemble cell matrices to be used in the transport sweep; repeating this calculation for each group \( g \) would be a waste of time and memory (provided the same mesh is used throughout). These are combined into more complicated operators:

- **WithinGroup** which represents \( W_g \) (see Equation 7.5)
- **FixedSource** which represents \( B \) (see Equation 7.4)
- **FixedSource\( \text{GS} \)** which represents the inverse of the lower block-triangular component of \( B, B^{-1} \)
- **FixedSource\( \text{GS} \)::step(\( x,b \))** which emulates one block Gauss-Seidel step of \( B, x \leftarrow B^{-1}(Bb + Bx) \)

To invert any of these linear operators—notably **FixedSourceGS** must invert each diagonal (**WithinGroup**) block and **FixedSource** must be inverted to solve the fixed-source problem—one simply constructs a linear solver, implementing Richardson, GMRES, FGMRES, BiCGSTAB, or relaxation (for example, block Gauss-Seidel) iterations. Then, given solver \( \text{solver} \), operator \( A \), solution \( x \), source \( b \), and preconditioner \( P \) (in the degenerate case, being \( I \)), one calls a method such as \( \text{solver.solve}(A, x, b, P) \). The exception is the relaxation solver, where \( P \) implements the relaxation stepping operator and \( A \) is used solely to compute residuals (that is, to check convergence). We expect this organization will be flexible to different research objectives and intuitive to other engineers and developers.

### 7.2.2 Criticality Problems

Additionally, in \( k \)-eigenvalue problems the fission operator is written

\[
A = L^{-1} X F
\]

(7.7)

where \( F \) represents the production of neutrons by fission (encapsulated as Production)

\[
F = \nu \Sigma_f D_0 = \begin{bmatrix} \nu \Sigma_{f,0} D_{g,0} & \cdots & \nu \Sigma_{f,g} D_{g,0} & \cdots & \nu \Sigma_{f,G} D_{g,0} \end{bmatrix}
\]

(7.8)
and \( X \) describes the emission of neutrons (encapsulated as \textbf{Emission})

\[
X \equiv M_0 \chi \equiv \begin{bmatrix} M_{g,0} \chi_0 & \cdots & M_{g,0} \chi_g & \cdots & M_{g,0} \chi_G \end{bmatrix}^T
\]

(7.9)

which are used in the operators:

- \textbf{Fission} which represents \( \hat{A} \)
- \textbf{FissionSource} which represents \( \mathbb{B}^{-1} \hat{A} \)

The eigenvalue problem may then be solved in one of two ways. The simpler choice is to use \textit{deal.II}'s native solvers, which implement the Power and (Shifted) Inverse Power solvers. The second, more versatile, approach is to interface with SLEPc, an extension to PETSc for eigenproblems, which offers more advanced solvers, such as Arnoldi, Krylov-Schur, Generalized Davidson, and Jacobi Davidson, among others. Moreover, SLEPc provides spectral transformations such as shift-of-origin and shift-and-invert. In either case, given operators \( \mathbb{B} \) and \( \hat{A} \), and solver \textit{eigensolver}, the generalized problem can typically be solved similarly to \textit{eigensolver.solver}(\( \mathbb{A}, \mathbb{B}, \mathbb{k}, \mathbb{x} \)) or, given the standardized operator \( \mathbb{B}_{\text{inv}} \hat{A} \), \textit{eigensolver.solver}(\( \mathbb{B}_{\text{inv}} \hat{A}, \mathbb{k}, \mathbb{x} \)) where \( \mathbb{k} \) and \( \mathbb{x} \) are containers for the eigenvalues and eigenvectors respectively. Preconditioners and spectral transformations, meanwhile, are specified through methods of \textit{eigensolver}.

### 7.2.3 Multigroup Cross Sections

In Aether, multigroup cross sections are represented by the \texttt{Mgxs} class, which simply stores the values of the total, scattering, and neutron production by fission cross sections, as well as emission spectra, indexed by energy group \( g \) and material ID \( j \). Each group is then assigned to the relevant operator—an instance of \texttt{TransportBlock} or \texttt{ScatteringBlock}—which selects the appropriate value according to each element’s material ID. Cross section libraries may be imported from or exported as HDF5 files, formatted as per their specification in OpenMC. Rudimentary functions are provided for computing weighting spectra from multigroup fluxes and, subsequently, coarsening a cross section library.

### 7.2.4 Angular Quadratures

To impose the \( S_N \) approximation, an angular quadrature must be specified, as represented by an instance of \texttt{QAngle}. This class inherits from \textit{deal.II}’s (spatial) \texttt{Quadrature},
which associates quadrature points on a reference domain—[0, 1] in each dimension up to \( \text{dim} \) (a template parameter)—with specific weights. In \texttt{QAngle}, parameterized by \( q\text{dim} \in \{1, 2\} \) and \( \text{dim} \in \{1, 2, 3\} \), each point is also associated with a polar and (for \( q\text{dim} = 2 \)) azimuthal angle and an ordinate (an instance of \texttt{dealii::Tensor<1, \text{dim}>}). Subclasses of \texttt{QAngle} may support reflecting boundaries by overriding a method which accepts a quadrature index and surface normal and returns the index of the reflected ordinate (or throws an exception). At time of writing, the only quadrature implemented is the Product Gauss-Legendre-Chebyshev (\texttt{QPglc})—that is, the product of a polar Gauss-Legendre and an azimuthal midpoint quadrature.

\section*{7.3 Future Work}

\subsection*{7.3.1 Parallelism}

Parallelism is a practical necessity for many problems of interest. Both shared- and distributed-memory parallelism are supported in \texttt{deal.II}, through the Threading Building Blocks (TBB) library and Message Passing Interface (MPI) respectively. Ideally, we would then use distributed-memory parallelism across nodes and shared-memory parallelism within each node. In practice, this would entail partitioning and distributing the spatial domain across several MPI processes and, within each subdomain, sharing the angles among all available threads. However, while the latter task (shared-memory parallelism along the ordinates for \( L_{g}^{-1} \)) is simple enough, the former is more complicated, because the (block Gauss-Seidel) transport sweep must march in the downstream direction. This necessitates some “sweep scheduling” algorithm to partition and schedule the problem in such a way that upstream information is always available while keeping all processors (simultaneously) as busy as possible. While such algorithms exist [209], [210], implementing them may prove difficult and distract from more pertinent research. Given this, we have chosen to focus first on small—though not trivial—problems which can be run in serial. Implementing shared- and distributed-memory parallelism then remains a high priority for future work, which stands to substantially enhance Aether’s utility as a research tool.

\subsection*{7.3.2 Preconditioning}

Since transport sweeps are computationally expensive, accelerating convergence by preconditioning is a crucial step to keeping simulations fast. There are two predominant
forms used in neutron transport, Diffusion Synthetic Acceleration (DSA) and Nonlinear Diffusion Acceleration (NDA) [153], the most common variant of the latter being Coarse Mesh Finite Difference (CMFD). As evident by the names, both preconditioners are based on diffusion. This suggests another clear advantage of our use of deal.II, in that it would allow easy implementation of diffusion operators for preconditioning—especially over unstructured meshes and/or by matrix-free methods. Transport Synthetic Acceleration (TSA) [153], perhaps by $P_N$ or simplified $P_N$ ($SP_N$) methods, could also be a productive avenue for research, and undoubtedly the $P_N$ operators would be useful in and of themselves. Given that preconditioners are easily interchangeable in Aether, these options are not mutually exclusive, and in fact implementing several should allow for illuminating comparative analyses. Despite this, the effort involved is non-trivial and arguably not essential to our primary research direction of PGD ROMs. Preconditioning is then left as an important area of future work.

7.3.3 Anisotropic Scattering

Anisotropic scattering is yet another desirable feature. Arguably, considering first-order (or even second-order) anisotropic scattering would make the energetic PGD comparisons in Chapters 4 and 5 more representative of the state-of-the-art in reactor physics. Given the effects of Consistent- and Inconsistent-P transport corrections are also assessed therein, including higher-order scattering would also affect those results—though perhaps not dramatically, because the reference solution is always specified to have the same scattering order as in the coarse-group problem. The effort required is not prohibitive and entails only the generalization of $D_g$, $M_g$, and $\Sigma_{s,g'\to g}$ operators.
CHAPTER 8
CONCLUSION

In conclusion, we have reduced the inherently high dimensionality of neutron transport by several novel PGD ROMs. Our first application is to separate the energy dependence thereof, as detailed in Chapter 4, which we find successful and even competitive to cross section condensation with as few as ten to twenty modes, based on the $L^2$ coarse-group error. Likewise, to condense the cross sections themselves, we find PGD benefits from the inherent errors of condensation, such that it can achieve comparable results to the full-order model in only 10 to 20 modes. Moreover, the PGD ROM performs no worse than a homogenized, infinite medium model with even just a single mode. For general-purpose reactor physics, we find the ROM (both Galerkin and Minimax) with update is consistently able to achieve $L^2$ errors of the angular flux below 0.1% by about 30 modes or fewer in our Cathalau pin-cell analyses with 70 to 361 groups. Extending this ROM to criticality problems (on the same benchmark) by a novel PGD algorithm described in Chapter 5, the $k$-eigenvalue errors are found to be $2 \times 10^{-4}$ or less given fifty modes. Further, the criticality ROM again surpasses the precision of the CASMO-SH-70 coarse-group model by ten to twenty modes.

Second, in Chapter 6 we invoke PGD to separate the axial (among other) dimensions of neutron transport, rendering 3D reactor physics simulations much cheaper to compute. Compared to existing 2D/1D methods, we expect our approach may be both more general and accurate—especially as it converges to the 3D solution, while established methods cannot because of engineering assumptions. Further, we consider separation in axial space and optionally polar angle, yielding two distinct ROMs. This number is brought to six by granting the energy dependence may be assigned to either the axial or radial modes, or both. Encouraging numerical results are demonstrated on the full-core (assembly-homogenized) Takeda LWR benchmark, with each ROM converging to an $L^2$ error less than 2\% given 30 modes.

Finally, we have developed Aether, a multigroup, $S_N$, DGFEM neutron transport library (outlined in Chapter 7) in which we implement both our PGD ROMs and the full-order model, for comparison. Despite the specialized nature of the above research, Aether is intended to be as general as possible, such that other researchers and users may therefore adopt our implementation (rather than writing their own). This is valuable even in light
of the many deterministic neutronics codes already established: first, because we intend to release our library as permissively open-source, eliminating the usual logistical and legal hurdles to software access and adoption; and second, because these tools do not offer first-order (hyperbolic) transport with a DGFEM discretization over unstructured meshes (with the arguable exceptions of Rattlesnake, Capsaicin, and Teton, see Section 7.1). Based on this gap, plus the design decisions outlined in Chapter 7 (notably, modularity and use of deal.II), we feel Aether constitutes a valuable contribution to the transport community.

8.1 Future Work

8.1.1 Algorithmic Enhancements

Although several practical applications of PGD have been demonstrated herein, the necessity of developing both the full- and reduced-order models from scratch—albeit, capably aided by deal.II—left little time for performance optimizations, most notably by preconditioning and parallelization. As such, this work makes only circumspect predictions about the time and memory costs of the PGD ROMs compared to the full-order models they are intended to replace. Implementing such features would allow for meaningful numerical experiments that conclusively answer such lingering questions.

Concretely, these experiments should investigate whether the parallel performance (that is, the scaling) of the PGD ROMs differs from that of the full-order model. Foreseeably, this could arise if some stage of the PGD calculation cannot saturate the available computing resources, leaving wasted idle time. This could be true of solving the energetic or axial (/polar) subproblems in the Energetic or 2D/1D PGD respectively. However, as these computations are not expected to be the dominant cost, such a disadvantage may be of little consequence unless it also holds true for the opposite subproblem.

Likewise, preconditioning both the full- and reduced-order models may skew the relative performances. This seems especially relevant in the case of multigrid preconditioners—such as the CMFD schemes of [211] and [212]—which simultaneously coarsen dimensions that are separated in the PGD ROMs and so may offer an advantage unique to the full-order model. Both NDA and DSA are popular preconditioners in reactor physics, and so a comprehensive analysis would ideally investigate this topic in both contexts. Additionally, while the most straightforward method of preconditioning would be application thereof to the individual, linear subproblems, other possibilities are conceivable. For instance, several conventional
2D/1D methods employ 3D CMFD preconditioners to great success. One might reasonably expect that CMFD, or another 3D preconditioner, could then be incorporated into 2D/1D PGD to comparable effect. In a similar fashion, the broad, open-ended investigation of PGD preconditioners in reactor physics may yield several effective schemes that further enhance the appeal of these ROMs.

Finally, the straightforward but crucial work of “tuning” the ROM—for instance, by adjusting tolerances, selecting appropriate solvers, adapting preconditioners as needed, and so on—stands to substantially affect the numerical performance. As such, this appears yet another key aspect of meaningful comparisons. Specifically, some form of adaptive tolerances in the nonlinear loop (mitigating “over-solving” [197]) seems desirable. Likewise, the same should be investigated for the outer enrichment loop, determining the ideal trade-off between many, loosely-converged modes or fewer, tightly-converged modes. Lastly, as the fixed-point (also termed Alternating Directions, Picard) iteration occasionally fails to converge, a Newton or quasi-Newton solver may present an appealing alternative.

8.1.2 Further Validation

Likewise, conclusive performance testing of the PGD ROM is also hampered by the limited scale of the benchmarks considered. This is, of course, another consequence of the as-yet unimplemented performance enhancements of the previous section, as these are practically necessary to perform the intensive, HPC simulations in which these ROMs are most useful. In specific, one application of interest could be extending the studies from Chapters 4 and 5 from a single pin-cell to an entire fuel assembly (or perhaps a “colorset” comprised of several assemblies). Additionally, incorporating depleted fuel, burnable absorbers, and/or finer energy meshes may further complicate the in-core spectra, perhaps providing a more challenging scenario in which to validate PGD in energy.

Regarding the Takeda LWR used to demonstrate 2D/1D PGD, a more practical case could include finer energy groups, heterogeneous resolution of individual pin-cells, and a taller reactor. The BEAVRS benchmark [213] and VERA progression problems [214] in particular seem to be apt test cases. That said, despite the small size and core-wide homogeneity of the Takeda LWR, several works [200], [215], [216] describe this benchmark (especially the unrodded configuration) as a challenge to traditional 2D/1D methods due to “stark material
heterogeneities" [200] at the interfaces between the core, reflector, and control rod. As such, the numerical results of 6.3.1 may already represent a more strenuous test case than the simple geometry would seem to suggest.

8.1.3 Incorporating Prior Information

Additionally, although PGD can be performed without any prior knowledge of the solution, it is often the case in reactor physics that a reasonable—though not exact—prediction is available, either from a previous simulation or an asymptotic limit. One may, for instance, expect that: the neutron spectra follows a $1/E$ curve in the slowing-down range and a Maxwell-Boltzmann distribution in the thermal range; the spectra can be approximated by equivalence in distribution; the axial power distribution of a reactor roughly resembles a cosine; the flux obeys Fick’s law in the diffusive regime; the flux at the present timestep is comparable to that of the previous; and so on.

As such, the ideal application of PGD may not be to construct a strictly a priori ROM, but rather one that can exploit—though does not require (nor is limited to)—prior knowledge, as from reference solutions, historical models, expert judgment, and the like. In fact, exploiting this information was the primary motivation of flux synthesis, as explained by Price and Duderstadt [51]:

There are [...] problems in which significant details of the flux distribution can be predicted in advance, and in these situations the full calculation of the discretized flux generates (at great expense) large amounts of redundant information. Synthesis techniques [...] are designed to treat just such situations by allowing the construction of approximations which incorporate any known features and require solution for only the remaining unknown parts.

Any number of means seem feasible to achieve this goal, which are not necessarily mutually exclusive. Perhaps the most straightforward is to “seed” the PGD algorithm with

\[25\] That said, one of three studies [215] finds much improved results by approximating the azimuthal anisotropy of transverse leakage as a Fourier series rather than assuming it to be isotropic. However, this scheme is subsequently found to be less accurate (as measured by $k$-eigenvalues) than the isotropic assumption in a heterogeneous (pin-resolved) benchmark. This puzzling result leads the authors to conclude that, in the former case, there is a cancellation of error between the spatial (flat) and angular (isotropic) approximations of the transverse leakage; accordingly, mitigating the angular (but not spatial) error counter-intuitively yields a less accurate $k$-eigenvalue.
some number of modes, either from a previous PGD simulation or the SVD (or other decomposition) of a reference solution. Subsequently, this guess can be refined in the usual manner: enriching the reduced bases by progressively computing rank-one corrections. Alternatively, one could adapt these modes to the present problem by “updates” or Subspace PGD as in Nouy [9]. This could conceivably be applied even if a guess is only available for one set of dimensions—for instance, of the energetic but not spatio-angular modes—to compute the modes in all other dimensions. Additionally, many sophisticated algorithms for \textit{a posteriori} model order reduction are available in literature: among them, POD, DMD, DLRA, RB, and the like. These ROMs, or their operating principles, could conceivably be applied to, incorporated in, or implemented alongside PGD to construct even more effective and performant ROMs. This is, of course, a broad topic applicable far beyond the realm of reactor physics, and so our understanding of such techniques should be greatly enriched by a continuing discourse with the larger PGD community.

\subsection*{8.1.4 Complementary Techniques}

Finally, this dissertation investigates PGD ROMs as a “standalone” solver—that is, an alternative to the full-order model and other ROMs. But this need not be the case. An obvious example would be to precondition the full-order model by way of the PGD ROM. This would afford the precision of the full-order model, but perhaps at a reduced cost and/or number of iterations. Likewise, the ROM could be used as a cheap approximation in multi-physics iterations, uncertainty quantification, quasi-static methods, and similar applications in which the full-order model should be evaluated as infrequently as practical.

Additionally, as hinted at in the previous section, PGD could be employed in conjunction with other ROMs. For example, a POD or RB method could be used to construct an initial ROM from reference solutions, which is adapted to the current problem by PGD. Similarly, a low-rank approximation achieved by PGD could be evolved by DLRA, either across iterations of some solver, the short time-scales of reactor transients, or the longer time-scales of fuel depletion. Researchers may combine and adapt ROMs in this manner to better suit particular applications, exploiting the unique advantages of each—including but not limited to PGD—to their greatest effect.
REFERENCES


APPENDIX A
COMPUTATIONAL COMPLEXITY

A.1 Introduction

Quantifying the computational complexity of the full- and reduced-order models is essential to predict their asymptotic performance and to precisely describe the model order reduction for which the latter are named. As such, this appendix presents these results for the full-order model and PGD in energy as defined in Chapters 3 and 4 respectively.\textsuperscript{26} Because the complexity of Progressive PGD amounts to that of repeatedly constructing and solving two or more low-dimensional submodels—each formed by projecting the full-order model onto one or more reduced bases—the analyses of the full- and reduced-order models are seen to be intimately related. More specifically, each submodel can be interpreted as some degenerate case of the full-order model, such that the complexity of solving the former can be easily deduced from that of the latter. The costs of constructing the submodel, meanwhile, are unique to the PGD ROMs and therefore require special consideration, as detailed in Sections A.3.2.2 and A.3.2.3. However, it is instructive to first consider a more abstract example, as in Section A.2.1, such that the nuances of neutron transport do not obscure broader insights into PGD.

Through these analyses, one finds that in Progressive PGD the number of matrix solves and matrix-vector products scales at $O(M)$. Meanwhile, the amount of vector arithmetic scales at $O(M^2)$. As such, while the latter may dominate for large $M$—supposing the former calculations are also $O(N)$ for $N$ degrees-of-freedom—it is foreseeable the matrix solves will dominate for low- to moderate-rank approximations, such that the overall runtime is roughly proportional to $M$ in this pre-asymptotic regime. Higher-order (as in $\geq O(M^3)$) terms are incurred in Progressive PGD with Update, though this update step is optional and typically enabled only when one expects these costs to be marginal regardless. Moreover, Appendix F.3 demonstrates that exploiting the “progressive” structure of PGD allows direct solution of the update problems at cost $O(M^3)$ rather than the naive $O(M^4)$. This proves especially useful in the Progressive PGD with Eigenvalue Update (see Chapter 5) given a standardized eigenproblem $B^{-1}Ax = kx$, as it avoids the need to solve $B$ repeatedly (that is, once per

\textsuperscript{26}Note that the 2D/1D ROMs of Chapter 6 are not discussed, as each 2D/1D submodel is equivalent to the spatio-angular submodel presented here, with the possible generalization to multigroup.
eigenvalue iteration).

A.2 General Example in Finite-Dimensional Tensor Space

A.2.1 Full-Order Model

To illustrate the model order reduction achieved by PGD, we begin by considering the separable matrix \( C \in \mathbb{R}^{AB \times AB} \) of Kronecker rank \( R \)

\[
C \equiv \sum_{r=1}^{R} A_r \otimes B_r ,
\]  

(A.1)

where \( \otimes \) is the Kronecker product, \( A_r \in \mathbb{R}^{A \times A} \), and \( B_r \in \mathbb{R}^{B \times B} \). Further, the problem of interest is that of solving the linear system

\[
Cz = f
\]

(A.2)

for \( z \in \mathbb{R}^{AB} \) given \( f \in \mathbb{R}^{AB} \). More generally, operator \( C \) and element \( f \) need only correspond to appropriate bilinear and linear forms on some inner product space, but a real matrix and vector employing the usual dot product are here chosen for sake of exposition. In the particular case of \( R = 1 \), solving Equation A.1 becomes a simple matter, as \( C^{-1} = A_1^{-1} \otimes B_1^{-1} \) and the product \( z = C^{-1}f \) can be efficiently computed as \( z = \text{vec}(B_1^{-1}FA_1^{-1,T}) \) where \( \text{vec}(F) = f \). Plainly, this reduces the complexity immensely, from \( \mathcal{O}(A^3B^3) \) to \( \mathcal{O}(A^3 + B^3) \) if solved directly. For \( R > 1 \), meanwhile, such a straightforward approach is not available, and only special forms of \( C \)—notably, Sylvester equations and their tensor generalization by Grasedyck [217]—permit similarly impressive results in general. One attempt to recover such savings is to approximate \( z \) by successive, rank-one corrections, as in the applications of Galerkin and Minimax PGD shown in Chapters 4–6. At present, however, we review the “full-order” case for comparison—specifically, that of block Gauss-Seidel iteration. This method is ubiquitously applied in energy-dependent neutron transport (exploiting the relative lack of upscattering above thermal energies) for \( C \leftarrow B \) as in Equation A.45.

Specifically, Equation A.2 may be solved by partitioning \( C = L + U \) into lower and strictly upper triangles (respectively) and repeatedly applying \( L^{-1}U \), as in

\[
Lz^{(k)} = f - Uz^{(k-1)}.
\]

(A.3)
Algorithm 5: Block Gauss-Seidel

```
for k = 1 \ldots \alpha do
  for i = 1 \ldots A do
    \( z_i^{(k)} \leftarrow \text{solve } C_{i,i} z_i^{(k)} = f_i - \sum_{j=1}^{i-1} C_{i,j} z_j^{(k)} - \sum_{j=i+1}^{A} C_{i,j} z_j^{(k-1)} \)
```

and detailed further in Algorithm 5. Doing so, the problem becomes that of solving each diagonal block \( C_{i,i} \) in sequence from \( i = 1 \) to \( A \), where

\[
C_{i,j} = \sum_{r=1}^{R} a_{i,j}^{(r)} B_r.
\]

Like \( C \) itself, these blocks are generally too large for direct solution, such than an inner iteration on the diagonals must be repeated at each outer iteration \( k \leftarrow k + 1 \), the required number of which (denoted \( \alpha \)) depends on the spectral radius of \( L^{-1}U \).

Notably, apart from the as-yet unspecified entries of \( A_r \) and \( B_r \), this represents the same scheme applied in energy to the multigroup full-order model of Section 4.4.4. Likewise, for the degenerate case of \( U = 0 \) such that \( \alpha = 1 \), this could be interpreted as some time-stepping iteration. In any case, as \( C \) is defined as a sum of Kronecker products, the structure of matrices \( A_r \) determines the block structure of \( C \), while the structure of matrices \( B_r \) determines the structure of each block \( C_{i,j} \), where structure refers to matrix properties which can be numerically exploited—sparsity, symmetry, triangularity, and so on. More formally, one can define the set of matrices \( \mathcal{A} \equiv \{ \sum_{r=1}^{R} b_r A_r : b_r \in \mathbb{R} \} \) (likewise \( \mathcal{B} \)) such that \( C_{i,j} \in \mathcal{B} \) and in the degenerate case of \( B = 1 \), \( C \in \mathcal{A} \). More generally, \( C \in \mathcal{C} \equiv \{ \sum_{r=1}^{R} F_r \otimes G_r : F_r \in \mathcal{A}, G_r \in \mathcal{B} \} \), by definition. Therefore, establishing some properties of \( \mathcal{A} \) and \( \mathcal{B} \)—for instance, the “worst” sparsity pattern of any matrix in \( \mathcal{A} \) is that of the bitmask \( \bigcup_{r=1}^{R} A_r \) where each entry of \( A_1 \cup A_2 \) is defined as \( a_{i,j}^{(1)} \neq 0 \lor a_{i,j}^{(2)} \neq 0 \)—this structure can be exploited to reduce the complexity relative to the general case \( \mathcal{A} = \mathbb{R}^{A \times A} \) and \( \mathcal{B} = \mathbb{R}^{B \times B} \). Specifically, let us denote the complexity (asymptotic number of operations) of a single matrix-vector or matrix-inverse-vector product as \( \mathcal{M}_A \) and \( \mathcal{S}_A \) respectively, assuming \( \mathcal{M}_A \leq \mathcal{O}(A^2) \) and \( \mathcal{M}_A \leq \mathcal{S}_A \leq \mathcal{O}(A^3) \), and likewise for \( \mathcal{B} \) and \( \mathcal{S} \). (For brevity, the latter operation is elsewhere in this thesis termed the solution of a matrix.) Given these
preliminaries, the complexity of Algorithm 5 can be described generally as

\[ S_C^{GS} = \alpha (A S_B + M_A M_B) . \]  \( \text{(A.5)} \)

where \( M_C = M_A M_B \)—that is, the number of block multiplications times the number of operations per block multiplication. For instance, assuming \( C_{i,j} \) is directly solved and no assumptions are made of \( A \) and \( B \), then \( S_B = O(B^3) \) and \( M_C = O(A^2 B^2) \) such that \( S_C^{GS} = \alpha O(AB^3 + A^2 B^2) \). At the other extreme, if all matrices in \( A \) and \( B \) are diagonal, then \( \alpha = 1 \) and \( S_C^{GS} = O(AB) \). These results will be revisited at the end of the following section, so as to quantify the model order reduction by PGD, which is characterized specifically for the original ROMs of Chapters 4–5 in Section A.3.

A.2.2 Progressive Proper Generalized Decomposition

As reviewed above, exploiting this Kronecker structure to solve the linear system \( Cz = f \) is trivial (and extremely effective) for \( R = 1 \), but not so otherwise. To avoid the burden of solving the full-order matrix \( C \) when \( R > 1 \), as is always the case in radiation transport, let us seek an approximate rank-\( M \) solution

\[ z \approx \sum_{m=1}^{M} x_m \otimes y_m \]  \( \text{(A.6)} \)

given a rank-\( N \) separable source

\[ f = \sum_{n=1}^{N} p_n \otimes q_n \]  \( \text{(A.7)} \)

by Progressive Galerkin PGD. This amounts to a a greedy algorithm where, starting at \( m = 1 \) and holding \( m' < m \) fixed, one seeks the best rank-one approximation \( x_m \otimes y_m \), defined as that which minimizes

\[ \left( \sum_{r=1}^{R} A_r \otimes B_r \right) (x_m \otimes y_m) = r_m \]  \( \text{(A.8)} \)

in a particular norm, where

\[ r_m \equiv \sum_{n=1}^{N} p_n \otimes q_n - \left( \sum_{r=1}^{R} A_r \otimes B_r \right) \left( \sum_{m'=1}^{m-1} x_{m'} \otimes y_{m'} \right) \]  \( \text{(A.9)} \)
is the residual given \( m - 1 \) modes. In practice, this is ubiquitously accomplished by some fixed-point algorithm (find \( x_m \) given \( y_m \), find \( y_m \) given \( x_m \), repeat) typically named for its alternating nature (Alternating Directions [5], nonlinear block Gauss-Seidel [218], Block Coordinated Gradient Descent [219], [220], Alternating Minimization [221], Alternating Least Squares [219], [222]) or connection to eigendecomposition (High-Order Power Method [223] or simply Power Iteration [9]). Specifically, in the conventional case of Galerkin PGD one initializes either mode, as in \( y_m \leftarrow y_m^{(0)} \), and then multiplies by \( I_{A \times A} \otimes y_m^\top \), reducing the above equation to the low-dimensional linear system

\[
A_m^{(0)} x_m^{(1)} = p_m^{(0)}
\]

where \( A_m^{(0)} \in \mathbb{R}^{A \times A} \), which can be solved for \( x_m^{(1)} \). More specifically, this is the Galerkin projection of Equation A.8 where residual \( r_m^{(1)} \) is orthogonal to \( \text{span}\{y_m^{(0)}\} \). Proceeding likewise given \( x_m \leftarrow x_m^{(1)} \) yields

\[
B_m^{(1)} y_m^{(1)} = q_m^{(1)}
\]

where \( B_m^{(1)} \in \mathbb{R}^{B \times B} \), which can be solved for \( y_m^{(1)} \). This nonlinear iteration repeats until \( x_m \) and \( y_m \) converge, at which point one may compute another rank-one correction \( m \leftarrow m + 1 \) or terminate the algorithm (the final rank being denoted \( m = M \)). This procedure is outlined in Algorithm 6, which reveals its similarity to the previous block Gauss-Seidel algorithm. In particular, Progressive PGD can be interpreted as the blockwise solution of two (or more) nonlinearly coupled, lower triangular matrices, the blocks of which are \( A_{m,m'} \) and \( B_{m,m'} \); these can be written—including the requisite inner products \( a_{m,m'}^{(r)} \) and \( b_{m,m'}^{(r)} \) as

\begin{algorithm}
\begin{enumerate}
\item for \( m = 1 \ldots M \) do
\item \hspace{1em} for \( k = 1 \ldots \gamma_m \) do
\item \hspace{2em} \( x_m^{(k)} \leftarrow \text{solve } A_{m,m}^{(k-1)} x_m^{(k)} = \left(p_m^{(k-1)} - \sum_{m'=1}^{m-1} A_{m,m'}^{(k-1)} x_{m'}\right) \)
\item \hspace{2em} \( y_m^{(k)} \leftarrow \text{solve } B_{m,m}^{(k)} y_m^{(k)} = \left(q_m^{(k)} - \sum_{m'=1}^{m-1} B_{m,m'}^{(k)} y_{m'}\right) \)
\end{enumerate}
\end{algorithm}
\[ \mathbf{B}_{m,m'}^{(k)} \equiv \sum_{r=1}^{R} a_{m,m'}^{(r,k)} \mathbf{B}_r, \quad a_{m,m'}^{(r,k)} \equiv \mathbf{x}_m^{(r,k)} \mathbf{x}_{m'}^{(r,k)}, \quad (A.12) \]

\[ \mathbf{A}_{m,m'}^{(k)} \equiv \sum_{r=1}^{R} b_{m,m'}^{(r,k)} \mathbf{A}_r, \quad b_{m,m'}^{(r,k)} \equiv \mathbf{y}_m^{(r,k)} \mathbf{y}_{m'}^{(r,k)}, \quad (A.13) \]

while the projection of the right-hand-side is given by

\[ \hat{\mathbf{p}}_m^{(k)} \equiv \sum_{n=1}^{N} q_{m,n}^{(k)} \mathbf{p}_n, \quad q_{m,n}^{(k)} \equiv \mathbf{x}_m^{(k)} \mathbf{q}_n, \quad (A.14) \]

\[ \hat{\mathbf{q}}_m^{(k)} \equiv \sum_{n=1}^{N} p_{m,n}^{(k)} \mathbf{q}_n, \quad p_{m,n}^{(k)} \equiv \mathbf{y}_m^{(k)} \mathbf{P}_n. \quad (A.15) \]

Given these definitions, one can account for a single step of \( \mathbf{x}_m \) in detail (omitting index \( k \) for brevity) as:

1. Scale and add vectors, \( \hat{\mathbf{p}}_m \leftarrow \sum_{n=1}^{N} q_{m,n} \mathbf{p}_n - \sum_{r=1}^{R} \sum_{m'=1}^{m-1} b_{m,m'}^{(r,k)} \mathbf{\alpha}_m^{(r)} \)
2. Solve for \( \mathbf{x}_m \leftarrow \mathbf{A}_m^{-1} \hat{\mathbf{p}}_m \), after assembling \( \mathbf{A}_{m,m} \leftarrow \sum_{r=1}^{R} b_{m,m'}^{(r)} \mathbf{A}_r \) if necessary
3. Compute matrix-vector products \( \mathbf{\alpha}_m^{(r)} \leftarrow \mathbf{A}_r \mathbf{x}_m \) for \( r = 1 \ldots R \)
4. Compute dot products:
   i. \( a_{m,m'}^{(r)} \leftarrow \mathbf{x}_m^{\top} \mathbf{\alpha}_m^{(r)} \) and for \( r = 1 \ldots R \) and \( m' = 1 \ldots m \)
   ii. \( p_{m,n} \leftarrow \mathbf{x}_m^{\top} \mathbf{P}_n \) for \( n = 1 \ldots N \)

Alternatively, if \( \mathbf{A}_r^{\top} \) (or, more generally, the adjoint operator \( \mathbf{A}_r^{\dagger} \) in the relevant inner product space) is available, one could evaluate the inner products by first multiplying from the left instead of the right,

\[ a_{m,m'}^{(r)} = \mathbf{x}_m^{\top} \mathbf{\alpha}_m^{(r)} = \mathbf{\hat{\alpha}}_m^{(r)} \mathbf{x}_{m'}, \quad (A.16) \]

where \( \mathbf{\hat{\alpha}}_m^{(r)} \leftarrow \mathbf{A}_r^{\top} \mathbf{x}_m \). Provided one also factors out \( \mathbf{A}_r \) from the double summation in \( r \) and \( m' \) required to compute the projected residual \( \hat{\mathbf{p}}_m \), as in

\[ \sum_{r=1}^{R} \sum_{m'=1}^{m-1} b_{m,m'}^{(r,k)} \mathbf{\alpha}_m^{(r)} = \sum_{r=1}^{R} \mathbf{A}_r \sum_{m'=1}^{m-1} b_{m,m'}^{(r,k)} \mathbf{\alpha}_m^{(r)} \left( \mathbf{\alpha}_m^{(r)} \right), \quad (A.17) \]
this obviates the need to store $\mathbf{\alpha}_{m}^{(r)}$ ($mR$ vectors), requiring only $\hat{\mathbf{\alpha}}_{m}^{(r)}$ ($R$ vectors)—in addition, of course, to each of the $m$ modes $\mathbf{x}_{m'}$—at the cost of doubling the required number of matrix-vector products per iteration. This may often be preferable, but were here focus exclusively on the former choice, for three reasons: first, both yield the same asymptotic complexity; second, implementing the adjoint $A_{r}^{\dagger}$ may be inconvenient, especially if $A_{r}$ is implemented as a matrix-free operator\textsuperscript{27} or the memory layout and/or access pattern must be reconfigured to preserve cache efficiency; and third, in spatially-dependent neutron transport, each term $r$ typically corresponds to a single material region $j$, with no overlap, such that this memory requirement reduces to $m(L+3)$ vectors rather than $mR$ where $R = (L+2)J+1$ in the case of $L$\textsuperscript{th} order scattering, $J$ materials, and discontinuous elements. That said, the choice of left- or right-multiplication can be made independently for each term $r$, and so this trade-off between memory and computation can be particularized to each operator $A_{r}$.

From this description, it is clear the blocks of the full-order model $C_{i,j}$ and those of the submodel $B_{m,m'}$ differ only by the (scalar) coefficients of the expansion and so are both in $B \subseteq \mathbb{R}^{B \times B}$. Likewise the structure of $A_{m,m'}$ is determined by the block structure of $C$, such that the former is a degenerate case of the latter, $C = A_{m,m'}$ where $B = 1$ and $B_{1} = [b_{m,m'}^{(r)}]$.

As such, the complexity of Progressive PGD can be quantified in terms of the same basic operations—matrix-vector multiplications and solves ($M_{\bullet}$ and $S_{\bullet}$) of matrices in either $A$ or $B$—as in Section A.2.1, facilitating a more general comparison.

Particularly, inspecting the complexity of each fixed-point iteration $k$—that is, one step of $\mathbf{x}_{m}$ then $\mathbf{y}_{m}$ as described above—it follows that

\[
S_{C}^{PGD} = \sum_{m=1}^{M} \gamma_{m} O \left( (S_{A} + S_{B}) + R (M_{A} + M_{B}) + (Rm + N) (A + B) \right) = M \bar{\gamma} O \left( (S_{A} + S_{B}) + R (M_{A} + M_{B}) + (RM + N) (A + B) \right)
\]

(A.18)

assuming that the complexity of all vector arithmetic is either $O(A)$ or $O(B)$. The coefficients of $R$ above neglect any improvements that can be made based on the structure of the expansion in $r$; for instance, as in the following section, that operators $A_{r}$ (likewise vectors $\mathbf{\alpha}_{m}^{(r)} = A_{r}\mathbf{x}_{m'}$) correspond to disjoint spatial domains $j$. Additionally, assuming matrices in $A$ are solved by point Gauss-Seidel (analogously to the block Gauss-Seidel solution of

\textsuperscript{27}Either of the kind in Appendix C, where the matrix exists but is not stored, or as in the Method of Characteristics or Monte Carlo methods, where the linear operator is not represented as a matrix at all.
matrices in $\mathcal{C}$, $\mathcal{S}^{GS}_{\mathcal{C}}$ as per Equation A.5)

$$
\mathcal{S}^{GS}_{\mathcal{A}} = \alpha \mathcal{M}_{\mathcal{A}}.
$$

(A.19)

where, again, $\alpha$ is not necessarily an independent parameter, and should be further investigated for particular cases of $\mathcal{A}$ and $\mathcal{C}$. Two illustrative examples of $\mathcal{S}^{PGD}_{\mathcal{C}}$ and $\mathcal{S}^{GS}_{\mathcal{C}}$ are those where $A \ll B$ and $A = B$. In the former, $\mathcal{S}_{\mathcal{A}}$, $\mathcal{M}_{\mathcal{A}}$, and $\mathcal{A}$ become negligible. This does not affect $\mathcal{S}^{GS}_{\mathcal{C}}$, in which these terms appear only as coefficients. Meanwhile,

$$
\mathcal{S}^{PGD}_{\mathcal{C}, A \ll B} = M\bar{\gamma} \mathcal{O}\left(\mathcal{S}_{\mathcal{B}} + RM_{\mathcal{B}} + (RM + N)B\right).
$$

(A.20)

In the latter case ($A = B$), assuming further that $\mathcal{A} = \mathcal{B}$ (that is, matrices in $\mathcal{A}$ and $\mathcal{B}$ have the same structure), the complexity of the full-order model becomes

$$
\mathcal{S}^{GS}_{\mathcal{C}, A = B} = \alpha \mathcal{O}(BS_{\mathcal{B}} + M^2_{\mathcal{B}})
$$

(A.21)

while, in PGD, the complexity is actually equivalent to the previous case,

$$
\mathcal{S}^{PGD}_{\mathcal{C}, A = B} = \mathcal{S}^{PGD}_{\mathcal{C}, A \ll B}
$$

(A.22)

as even though the latter requires roughly twice the number of floating point operations (assuming all other parameters do not change), this does not affect the order of complexity. Granting that matrices in $\mathcal{B}$ are solved iteratively $\mathcal{S}_{\mathcal{B}} = \beta \mathcal{M}_{\mathcal{B}}$ where each matrix-vector product $\mathcal{M}_{\mathcal{B}} = \mathcal{O}(B)$, this yields $\mathcal{S}^{GS}_{\mathcal{C}, A = B} = \mathcal{O}(\alpha B^2)$ and $\mathcal{S}^{GS}_{\mathcal{C}, A = B} = M\bar{\gamma} \mathcal{O}((RM + N)B)$. As expected, separating a $B^2 \times B^2$ matrix into two $B \times B$ matrices reduces the model order by a factor of $B$, all other variables being held constant. While the actual cost will depend on the values of each other parameter (especially $M$), the immense potential for computational savings is evident, especially when generalized to higher-order tensor decompositions. In what follows, we will now particularize this discussion to the full- and reduced-order models of radiation transport.
A.3 Neutron Transport

A.3.1 Full-Order Model

As a baseline for comparison, we must first estimate the computational complexity of the full-order model of multigroup, $S_N$, DGFEM radiation transport. Let us suppose this discretization includes $G$ energy groups, $N$ ordinates, $C$ spatial degrees-of-freedom, and $L$\textsuperscript{th} order scattering, such that the inscattered source can be exactly expressed as a spherical harmonics expansion of $\Upsilon$ terms where

$$\Upsilon = \begin{cases} 
L + 1, & d = 1, \\
(L + 1)(L + 2)/2, & d = 2, \\
(L + 1)^2, & d = 3.
\end{cases}$$

in $d$-dimensional geometry. The full-order fixed-source problem is then given by

$$(L + B M \Sigma_s D) \psi = B q$$

where we will assume the source $q$ can be expressed as a spherical harmonics expansion $Q$ such that

$$q = M Q$$

analogously to $\psi$ and $\phi$. As usual in reactor physics, this linear system will be solved by first multiplying by $D L^{-1}$, which is tantamount to preconditioning. Assuming, for sake of exposition, the sweep contains no cycles, such that $L = L$,

$$(I + D L^{-1} B M \Sigma_s) \phi = D L^{-1} B M Q$$

or

$$(I + T \Sigma_s) \phi = T Q$$

where the multigroup transport sweep is

$$T \equiv D L^{-1} B M$$
which is block diagonal, the within-group transport sweeps being defined as

$$T_g \equiv D_g L_g^{-1} B_g M_g.$$  \hfill (A.29)

Applying a block Gauss-Seidel scheme in energy, the problem becomes that of solving the within-group (diagonal) blocks

$$W_g \phi_g^{(i)} = T_g \left( Q_g + \sum_{g' < g} \Sigma_{s,g' \rightarrow g} \phi_g^{(i)} + \sum_{g' > g} \Sigma_{s,g' \rightarrow g} \phi_g^{(i-1)} \right)$$ \hfill (A.30)

in which

$$W_g \equiv (I - T_g \Sigma_{s,g \rightarrow g})$$ \hfill (A.31)

for $\phi_g^{(i)}$, iterating sequentially from $g = 1$ to $G$. Starting from $\phi^{(0)}$, this procedure then repeats ($i \leftarrow i + 1$) until convergence. Doing so, it is natural to express the computational cost as a summation of two components: multiplication by strictly upper and lower triangles (up- and downscattering) $M_S$ and solving the diagonal (within-group) blocks $S_{W,g}$

$$S_{g}^{GS} = \alpha O \left( \sum_{g=1}^{G} S_{W,g} + M_S \right)$$ \hfill (A.32)

$$= \alpha O \left( G S_{W} + M_S \right)$$ \hfill (A.33)

multiplied by the number of Gauss-Seidel iterations $\alpha$ (where $S_{W}$ is the group-wise average of $S_{W,g}$).

A.3.1.1 Up- and Downscattering

The up- and downscattering terms require only multiplication by the diagonal matrices $\Sigma_{s,g' \rightarrow g}$, as in

$$Q_g^\downarrow \equiv \sum_{g' < g} \Sigma_{s,g' \rightarrow g} \phi_{g'}^{(i)},$$ \hfill (A.34)

$$Q_g^\uparrow \equiv \sum_{g' > g} \Sigma_{s,g' \rightarrow g} \phi_{g'}^{(i-1)},$$ \hfill (A.35)
and consequently, if either triangle of $\Sigma_s$ is dense, the complexity is

$$\mathcal{M}_s \leftarrow \mathcal{O}(G^2YC)$$  \hspace{1cm} (A.36)

In general, the upper triangle (upscattering) will be dense in the thermal energy range, but banded or zero at higher energies. An important exception is when a shift-and-invert spectral transformation (or “Wielandt” shift) is applied, in which case this operator is modified to include contributions of both scattering and fission (the latter being necessarily isotropic), such that the upper triangle will be dense. That said, the correlation between incoming and outgoing energies ($E'$ and $E$) from fission is typically slight, such that these terms may be low-rank, often rank-one, as in

$$Q_f^g \equiv \chi_g f_g^{(i)}$$  \hspace{1cm} (A.37)

where the fission rate $f_g^{(i)}$ is a vector of length $NC$ and can be updated once per group, as in $f_1^{(0)} \equiv 0$ and

$$f_g^{(i)} \equiv f_{g-1}^{(i)} + \nu \Sigma f_{g-1} \left( \phi_{g-1,\ell=0}^{(i)} - \phi_{g-1,\ell=0}^{(i-1)} \right)$$  \hspace{1cm} (A.38)

for $g > 1$; exploiting this reduces the complexity of computing the fission source from $O(G^2C)$ to $O(GC)$. The lower triangle, meanwhile, will be dense in the presence of a hydrogenous ($^1$H) moderator, such as light water, but banded otherwise (according to the maximum energy loss a neutron can suffer by a single scatter). In either case, it is typical that $L = 0$ or 1, such that $Y \ll N$. This mitigates the cost of the $G^2$ term, particularly if $N/G \gtrsim Y$.

Regardless, for applications with “ultrafine” energy grids, this quadratic scaling still becomes burdensome or dominant, motivating several schemes which recover $O(G)$ complexity. In CENTRM [224]—which models the actual kinematics of elastic and inelastic scattering (rather than relying solely on tabulated constants)—this is achieved by “submoment expansion” of Legendre polynomial $P_\ell$. Specifically, as the correlation function $\kappa_j$ between $u'$ and $u$ for material $j$ is a an argument of $P_\ell$, it is advantageous to expand each moment $\ell$ as

$$\int_{u-\epsilon_j}^u P_\ell(\kappa_j(u',u))du' = \sum_{m=-\ell}^{\ell} y_{\ell,m,j}^{\text{out}}(u) \int_{u-\epsilon_j}^u y_{\ell,m,j}^{\text{in}}(u')du'$$  \hspace{1cm} (A.39)

where $y_{\ell,m,j}^{\text{in/out}}$ are the incoming/outgoing components and $\epsilon_j$ is the maximum lethargy gain (energy loss) per scatter, which may be equal to $u$. As in the discrete (rank-one) expansion
of Equations A.37 and A.38, this separable form allows evaluation in $O(G)$. Notably, for isotropic, elastic scattering a rank-one expansion of this form is trivially available, as $P_0$ is constant.

Alternatively, in RAZOR [225], the typical approach of computing group-to-group contributions while marching down the energy grid is maintained; however, the $O(G^2)$ cost of ordinary Gauss-Seidel is avoided by use of two overlapping grids, one fine and one coarse. Fine-group to fine-group contributions are only calculated for some band of energy below group (or point) $g$. Denoting the size of the coarse mesh as $G_c$—each containing a fine mesh of size $G_f$ (such that $G = G_c G_f$)—and the bandwidth as $b$ fine groups, it is clear the cost reduces to $O(G(b + G_c)) = O(G_c G_f b + G_c^2 G_f)$, which remains quadratic in $G_c$ but not $G_f$. Accordingly, refining the fine (though not coarse) mesh has only an $O(G)$ effect on runtime.

A.3.1.2 Within-Group Blocks

Meanwhile, diagonal blocks $W_g$ of Equation A.31 are solved iteratively by GMRES, with matrix-free transport sweeps. Since $\Sigma_{s,g\rightarrow g}$ is a diagonal matrix, applying it to $\phi^{(i)}_g$ requires only $\Upsilon C$ multiplications. As such, the cost of solving Equation A.30 is dominated almost exclusively by that of repeatedly applying the transport sweep $T_g$, as well as the preconditioner $P_g$ if used.

As evident by its definition in Equation A.29, a transport sweep consists of three stages: converting the spherical harmonics series into a source along each ordinate (by $M_G$); computing the uncollided flux along each ordinate arising from this source (by $L_g^{-1} B_G$); and integrating these directional fluxes back into moments of the spherical harmonics (by $D_G$). Although $T_g$ is usually implemented so as to avoid storing the entire vector of angular fluxes (of size $NC$), this affects only the sequence, not the quantity, of computations. As such, one may observe that multiplying by $M_G$ and $D_G$ both require $O(N \Upsilon C)$ operations, leaving only $L_g^{-1} B_G$ to consider.

Assuming one applies matrix-free transport sweeps as in Appendix C, the cost of a transport sweep is that of assembling and then solving the local matrices. Assuming, for simplicity a $d$-dimensional mesh of $K$ cells where each element has $C_K$ degrees-of-freedom ($C = K C_K$) and $F$ faces, the cost of assembly would be $(d(F + 1) + 1) \times O(C_K^2)$. Computing the right-hand-side then requires one matrix-vector product per inflow boundary, plus the product $\bar{q} \equiv B_K q_K$, each of which being $O(C_K^2)$. Directly factorizing $L_K$ at cost $O(C_K^3)$,
the local solution $\psi_K = L^{-1}_K \hat{q}$ is computed at cost $O(C_K^2)$. If each element is formed by a tensor product of 1D shape functions with $\hat{C}_K$ degrees-of-freedom each, one can further write $C_K = \hat{C}_K^{dK}$. Likewise, in the quadrilateral and hexahedral meshes employed in this thesis $F = 2^d$. Repeating this procedure for every element and ordinate, this local cost is multiplied by a factor of $NK$. As such the cost of one transport sweep can be written

$$ S_T \leftarrow O(NC(C_K^2 + dFC_K + T)). \quad (A.40) $$

For completeness, we also define the cost of the preconditioner, $S_P$, though if $P$ is the identity matrix, $S_P = O(1)$. Assuming each diagonal block can be solved—or, more specifically, that the residual of the linear system can be reduced by $\tau$ orders of magnitude—in $\beta_g$ iterations, the total cost is

$$ S_{W,g} \leftarrow \beta_g (S_T + S_P). \quad (A.41) $$

Given this definition of convergence, $\beta_g$ can be estimated as in $10^{-\tau} \approx \rho_g^{\beta_g}$ or equivalently $\beta_g \approx -\tau / \log(\rho_g)$ for spectral radius $\rho_g$. From the Fourier analysis [153] of an energy-independent, discrete (diamond-differenced) $S_N$ method in a spatially-periodic medium, solved using unpreconditioned Richardson iteration with transport sweeps, one may reasonably expect $\rho_g$ to be bounded from above by the maximum scattering ratio $c_{g\rightarrow g,j^*}$,

$$ \rho_g \leq c_{g\rightarrow g,j^*} = \frac{\Sigma_{s,g\rightarrow g,j^*}}{\Sigma_{t,g,j^*}}. \quad (A.42) $$

Notably, this estimate is independent of $N$ or $C$; however, one would expect $c_{g\rightarrow g,j^*}$ to be inversely correlated with $G$ in that the ratio of within-group scattering will decrease as the groups become finer, with the particular effect being contingent on the differential scattering cross section and method of cross section generation.

That said, unpreconditioned Richardson iteration will rarely be employed in practice. Instead, some form of diffusion or transport preconditioner $P$ is ubiquitously employed. Notably, Transport Synthetic Acceleration (TSA) or Consistent Synthetic Acceleration (DSA) can be shown to be unconditionally stable, achieving the same reduction in spectral radius in both the continuous and discretized Fourier analyses [153]. Meanwhile, Inconsistent DSA introduces a dependence on the optical mesh thickness $\Sigma_t h$, achieving the efficacy of the Consistent version only as $\Sigma_t h \rightarrow 0$. In the opposite direction, as $\Sigma_t h \rightarrow \infty$, the spectral
radius either approaches that of the unpreconditioned iteration or (depending on one’s specific choice of discretization) grows beyond unity such that the iterations diverge.

In any case, letting $\bar{\beta}$ be the average number of within-group iterations, substituting in the above values of $O_T$ and $O_P$ and simplifying yields a final value of

$$S_{B}^{GS} = \alpha \left( G\bar{\beta} \left( S_T + \bar{S}_P \right) + M_S \right)$$

(A.43)

This estimate explains why in Chapter 5, the computational cost of the full-order model is described as scaling “at best” linearly with $G$: although the quadratic ($\propto G^2$, arising if $\Sigma_s$ is dense) scattering term will dominate for an asymptotically fine energy grid, many reactor physics analyses are posed in the pre-asymptotic regime where the linear ($\propto G$) within-group term constitutes the dominant cost—often, because many within-group iterations are required ($\bar{\beta}$ is large), there are few moments relative to ordinates ($Y \ll N$), or both.

### A.3.2 Progressive PGD in Energy

Having determined the complexity of the full-order model, we now proceed to the same of the PGD ROM presented in Chapters 4 and 5—as separated in space-angle, $\vec{r}$ and $\vec{\Omega}$, and energy, $E$—beginning with the separated fixed-source problem,

$$\mathbb{B}\psi = (I \otimes B) q$$

(A.44)

where $\mathbb{B}$ is the “leakage plus absorption” operator

$$\mathbb{B} \equiv I \otimes G + \sum_{j=1}^{J} \sigma_{t,j} \otimes N_j B - \sum_{\ell=0}^{L} \sum_{j=1}^{J} \sigma_{s,j,\ell} \otimes N_j B M_{\ell} D_{\ell}$$

(A.45)

and $G$, $B$, $M_{\ell}$, and $D_{\ell}$ are sparse matrices—specifically, the streaming, mass, moment-to-direction, and direction-to-moment matrices—defined in Equations 3.39–3.41. Additionally, indicator matrix $N_j$ is diagonal, with entries of 1 for cells $K \in \mathcal{V}_j$ and 0 otherwise. Further,
I is the $G \times G$ identity matrix, $\sigma_{t,j} \equiv \text{diag}(\sigma_{t,j,1}, \ldots, \sigma_{t,j,g}, \ldots, \sigma_{t,j,G})$, and

$$
\sigma_{s,j,\ell} \equiv \begin{bmatrix}
\sigma_{s,j,\ell,1 \to 1} & \ldots & \sigma_{s,j,\ell,g' \to 1} & \ldots & \sigma_{s,j,\ell,G \to 1} \\
\vdots & \ddots & \vdots \\
\sigma_{s,j,\ell,1 \to g} & \sigma_{s,j,\ell,g' \to g} & \sigma_{s,j,\ell,G \to g} & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
\sigma_{s,j,\ell,1 \to G} & \ldots & \sigma_{s,j,\ell,g' \to G} & \ldots & \sigma_{s,j,\ell,G \to G}
\end{bmatrix}.
$$

(A.46)

Similarly, the separated source is defined as

$$
q \equiv \sum_{i=1}^{I} \bar{\chi}_i \otimes Bq_i,
$$

(A.47)

as well as the approximate solution

$$
\psi \approx \sum_{m=1}^{M} \xi_m \otimes \psi_m.
$$

(A.48)

Given these preliminaries, the computational complexity of Progressive PGD (without update) can be determined as the total of that of each rank-one correction.

A.3.2.1 Fixed-Point Iteration

Applying the ubiquitous fixed-point (Alternating Directions) iteration yields the following two linear subproblems

$$
\mathbb{B}^{G}_{m,m} \psi_m = r^{G}_m
$$

(A.49)

$$
\mathbb{B}^{(V,4\pi)}_{m,m} \xi_m = r^{(V,4\pi)}_m
$$

(A.50)

which equate the weighted residual with respect to test functions $\tilde{\xi}_m$ and $\tilde{\psi}_m$

$$
r^{G}_m \equiv \sum_{i=1}^{I} \left( \tilde{\xi}_m^T \Delta \tilde{\chi}_i \right) Bq_i - \sum_{m'=1}^{m-1} \mathbb{B}^{G}_{m,m'} \psi_{m'}
$$

(A.51)

$$
r^{(V,4\pi)}_m \equiv \sum_{i=1}^{I} \left( \tilde{\psi}_m^T Bq_i \right) \tilde{\chi}_i - \sum_{m'=1}^{m-1} \mathbb{B}^{(V,4\pi)}_{m,m'} \xi_{m'}
$$

(A.52)
and, given the same test functions, the projection of $B$ onto reduced bases $\{\xi_{m'}\}_{m'=1}^m$ and $\{\psi_{m'}\}_{m'=1}^m$, alternately,

\[
\mathbb{B}_{m,m'}^{(g)} = \left( \tilde{\xi}_m^\top \Delta \xi_{m'} \right) G + \sum_{j=1}^J \left( \tilde{\xi}_m^\top \Delta \sigma_{t,j} \xi_{m'} \right) N_j B - \sum_{\ell=0}^L \sum_{j=1}^J \left( \tilde{\xi}_m^\top \Delta \sigma_{s,j,\ell} \xi_{m'} \right) N_j B M_{\ell} D_{\ell},
\]

\[
\mathbb{B}_{m,m'}^{(V,4\pi)} = \left( \tilde{\psi}_m^\top Y G \psi_{m'} \right) I + \sum_{j=1}^J \left( \tilde{\psi}_m^\top Y N_j B \psi_{m'} \right) \sigma_{t,j} - \sum_{\ell=0}^L \sum_{j=1}^J \left( \tilde{\psi}_m^\top Y N_j B M_{\ell} D_{\ell} \psi_{m'} \right) \sigma_{s,j,\ell},
\]

where $\Delta \equiv \text{diag}(\Delta_1^{-1}, \ldots, \Delta_g^{-1}, \ldots, \Delta_G^{-1})$ and $Y \equiv \text{diag}(w_1, \ldots, w_n, \ldots, w_N) \otimes I_{C \times C}$ are diagonal matrices. Evaluating the summation over $j$ simplifies the notation somewhat, allowing these submodels to be more readily recognized as those of energy-independent radiation transport (an integro-differential equation) and neutron slowing down in an infinite, homogeneous medium (a Fredholm integral equation),

\[
\mathbb{B}_{m,m'}^{(g)} = \left( \tilde{\xi}_m^\top \Delta \xi_{m'} \right) G + \tilde{\Sigma}_{t,m,m'}^{(g)} B - B D \tilde{\Sigma}_{s,m,m'}^{(g)} M, \quad (A.55)
\]

\[
\mathbb{B}_{m,m'}^{(V,4\pi)} = \left( \tilde{\psi}_m^\top Y G \psi_{m'} \right) I + \tilde{\Sigma}_{t,m,m'}^{(V,4\pi)} - \tilde{\Sigma}_{s,m,m'}^{(V,4\pi)}. \quad (A.56)
\]

Naturally, the cost (and therein complexity) of a single fixed-point iteration consists of that required to construct and solve Equations A.55 and A.56, denoted $\mathcal{F}_\star$, yielding

\[
\mathcal{S}_{B}^{\text{PGD}} = \sum_{m=1}^M \gamma_m \left( \mathcal{F}_\psi + \mathcal{F}_\xi \right) \quad (A.57)
\]

given $\gamma_m$ fixed-point iterations. This analysis could be continued by estimating the order of $\gamma_m$, whether as a function of discretization parameters $G$, $N$, $L$, and/or $C$; physical properties such as cross sections; or both. Addressing this by Fourier analyses of the discretized submodels would foreseeably yield many actionable insights into means of enhancing or ensuring the convergence of this iteration, as well as useful comparisons with similar methods from literature (such as Adams and Morel’s two-grid acceleration in energy [226] or Kelley and Larsen’s analysis of a novel 2D/1D method [227]). Therefore, such analyses seem a high priority for future work. At present, however, we will instead turn to analyzing $\mathcal{F}_\psi$ and $\mathcal{F}_\xi$. 
A.3.2.2 Spatio-angular Calculations

For clarity, the complexity associated with nonlinear iterate $\psi_m$ can be divided into that of solving the spatio-angular submodel (for $\psi_m$) plus that of computing both the residual $r_m^{(G)}$ and the requisite inner products to construct the energetic submodel. The latter task can be further subdivided into matrix-vector products (as in $y = A\psi_m$) and dot products (as in $\langle \psi_m, y \rangle$). The complexity of each stage ($S_{\psi}$, $M_{\psi}$, and $V_{\psi}$ respectively) is detailed in what follows.

Beginning with matrix-vector products, the required terms are $G\psi_m$, $B\psi_m$, and $B M_\ell D_\ell \psi_m$. As explained in Section 4.4.10, because $\psi_{m'}$ is fixed for $m' < m$, products of those modes need not be recomputed, though one may choose to do so to avoid the memory burden. Additionally, as the submodel transport sweep is implemented as $L^{-1}_{m,m} B$ (see Section 7.2) it is convenient to multiply by $B^{-1}$ such that each term of $r_m^{(G)}$ is an operand of $B$. The cost is not prohibitive, since $B \equiv I_{N\times N} \otimes B_N$ where $B_N$ is block-diagonal, such that $B$ can be directly inverted at cost $O(CC_K^2)$ (once, during the pre-calculation stage of Section C.1) and stored with memory footprint $O(CC_K)$, the same as that of $B$. Doing so, the required terms become $B^{-1} G\psi_m$ and $M_\ell D_\ell \psi_m$, which can be computed at complexities $O(d F N C K)$ and $O(NYC)$, yielding

$$M_{\psi} = O(d F N C K + NYC) \quad (A.58)$$

From these vectors, multiplication by diagonal matrices $Y$ and $Y N_j$ and the dot product by $\tilde{\psi}_m^T$ can be accomplished in $O(NC)$ operations. Notably, there is no coefficient of $J$, because the material regions $j$ are assumed not to overlap, such that $\sum_{j=1}^{J'} N_j = I_{N C \times N C}$. Applying such operations to each source term $i$ and mode $m'$ (each of the latter consisting of a streaming, collision, and $L + 1$ scattering terms), yields a total of $I + (3 + L) M$ products, so that

$$V_{\psi} = O((I + ML')NC) \quad (A.59)$$

where $L' = L + 1$ is the number of Legendre moments. Regarding the spatio-angular residual term, $r_M^{(G)}$, the same matrix-vector products of $\psi_m$ as above (where $m' < m$) must be read from memory or recomputed, though, in the latter case, operators can be factored out for efficiency, as in

$$\sum_{m'=1}^{m-1} \left( \xi_m^T \Delta \xi_{m'} \right) B^{-1} G \psi_{m'} = B^{-1} G \sum_{m'=1}^{m-1} \left( \xi_m^T \Delta \xi_{m'} \right) \psi_{m'} \quad (A.60)$$
and likewise for $M\ell D\ell$. In any case, scaling and summing each of these spatio-angular vectors is also of complexity $O((I + ML').NC)$ and so does not affect the previous estimate.

The only remaining cost is that of solving Equation A.55, the spatio-angular submodel, for $\psi_m$. As discussed in Section 4.4.3, this is tantamount to a solution of a single diagonal block of the full-order model, $W_g$, and so has the same computational complexity,

$$S_\psi = \tilde{\beta}_m O(S_T + S_P) \quad \text{(A.61)}$$

where $S_T$ is as is Equation A.40. Likewise,

$$\tilde{\beta}_m = \text{ceil}(-\tau/\log(\tilde{\rho}_m)) \quad \text{(A.62)}$$

where again for unpreconditioned Richardson iteration, $S_P = O(1)$ and

$$\tilde{\rho}_m \leq \tilde{c}_{m,j} \equiv \frac{\tilde{\Sigma}^{(g)}(t,m,m,j^*)}{\tilde{\Sigma}^{(g)}(s,m,m,j^*)} \quad \text{(A.63)}$$

which is analogous to Equation A.42, except that (as the submodel is effectively one-group) all scattering is, by definition, within-group, and so $\tilde{\beta}_m$ does not depend on $G$. Meanwhile, the discussion of preconditioners (notably TSA and DSA) of Section A.3.1.2 also holds, with the simplification that $G = 1$.

Summing these three distinct components, the complexity of each fixed-point iteration $F_\psi$ is found to be

$$F_\psi = O(S_\psi + M_\psi + V_\psi) = O(\tilde{\beta}_m(C^2_K + dFC_K + T)NC + (I + ML').NC), \quad \text{(A.64)}$$

which reveals some important details about the spatio-angular submodel. First, the spatio-angular solve is tantamount to the within-group solve of the full-order model; additionally, this cost (asymptotically) subsumes that of computing the required matrix-vector products for PGD. Secondly, the only cost which scales with $M$ is that of computing the necessary dot products and residual vector, which is linear with respect to $M$, $L$, $N$ and $C$. This may pose difficulties for large values of $M$, but for as long as the costs of the spatio-angular solve (that is, transport sweeps $S_T$ and preconditioning $S_P$) dominate, there will be a pre-asymptotic
range in which the performance of the ROM is roughly independent of $M$.

Should this scaling become burdensome regardless, one could seek to reduce $M$ (while retaining precision) by occasionally compressing and truncating the PGD bases. Alternatively, in $S_N$ radiation transport, one could seek to reformulate PGD such that operands are expressed as a spherical harmonics expansion (like $\phi_m$) rather than along discrete ordinates (like $\psi_m$). This would not remove the dependence on $M$, but would make these costs linear with respect to $\mathcal{Y}$ rather than $N$, extending this pre-asymptotic regime and mitigating the memory requirements of the ROM (as discussed in Section 3.3.2). That said, it is unclear if and how this could be effectively accomplished. In any case, as this cost is an inherent feature of Progressive PGD, future developments in the broader community could alleviate this issue if it arises in practice.

A.3.2.3 Energetic Calculations

As in the previous section, we now categorize the costs associated with (one nonlinear iterate of) $\xi_m$ as $S_\xi$, $M_\xi$, and $V_\xi$. In what follows, we assume that each scattering matrix $\sigma_{s,j,\ell}$ is full, such that a matrix-vector product costs $\mathcal{O}(G^2)$, though this conservative assumption is subject to all the same exceptions discussed in Section A.3.1.1. Noting the required matrix-vector products are $\sigma_{s,j,\ell} \xi_m$ and $\sigma_{i,j} \xi_m$, it follows that

$$M_\xi = \mathcal{O}(JL'G^2).$$

(A.65)

The vector operations required to compute $r_{m}^{(\mathcal{Y},4\pi)}$ and the necessary inner and dot products, meanwhile, are of order

$$V_\xi = \mathcal{O}((I + mJL')G)$$

(A.66)

Further, the energetic submodel (omitting the complexity of assembly, which is equivalent to $M_\xi$) may be solved by Gauss-Seidel iteration, analogously to Equation A.19,

$$S_\xi^{GS} = \mathcal{O}(\tilde{\alpha}_m G^2)$$

(A.67)

or directly (as by LU decomposition)

$$S_\xi^{LU} = \mathcal{O}(G^3)$$

(A.68)
yielding two cases of the total complexity per nonliner iteration $\mathcal{F}_\xi$

$$\mathcal{F}_\xi := \begin{cases} 
\mathcal{O}(\bar{\alpha}_m + JL')G^2 + (I + mJL')G), & \text{iterative,} \\
\mathcal{O}(G^3 + JL'G^2 + (I + mJL')G), & \text{direct.}
\end{cases} \quad (A.69)$$

As expected, $\mathcal{F}_\xi$ necessarily contains a term of order $M$ and $G$ associated with computing the required inner and dot products. However, for $M < G$ those terms containing $G^2$ and $G^3$ will foreseeably dominate the computational cost. Moreover, as $G \ll NC$ except perhaps in extreme ultrafine-group fidelity, $G$ usually represents the full rank, such that at $M \approx G$ the theoretical (though not necessarily numerical) maximum relative error in the Frobenius norm approaches zero, $1 - \sqrt{M/G} \approx 0$. Accordingly, in most practical uses of the ROM, one would expect $M$ to be substantially less than $G$ (hence, a low-rank approximation). In any case, this completes the complexity analysis of the energetic submodel apart from the “update” and “eigenvalue update” steps, of which a thorough treatment can be found in Chapter 5 and Appendix 5.4.6.
APPENDIX B
CONVERGENCE PROOFS AND BOUNDS

B.1 Introduction

As a divergent or unreliable algorithm is of little to no use, we briefly review what convergence properties of Progressive PGD have been proven to date. Two notable gaps are found to be applications to eigenvalue problems and non-symmetric operators, both of which figure prominently in this thesis. That being said, limited results are available of each, and what theoretical questions remain have generally been found not to preclude numerical computation. Following this, upper and lower convergence bounds for PGD are provided and practical concerns about their application are discussed.

B.2 Convergence Results of Proper Generalized Decomposition

To begin, Ammar et al. [219] establish the convergence of least squares PGD for solving a linear system such as $Cz = f$, under only the assumption that matrix $C$ is invertible; moreover, a rate of convergence is given for the norm of the residual in terms of a sequence of angles, and the Alternating Directions (or Block Coordinated Descent) iteration is shown (in that case) to be equivalent to Alternating Least Squares. Another approach is that of Cancès et al. [228], who prove the convergence of a greedy algorithm based on minimizing a (strongly convex) energy functional associated with a linear or nonlinear problem, provided its gradient is Lipschitz continuous; further, it is proven that in finite-dimensional vector spaces the algorithm converges exponentially fast. This extends an earlier result by Le Bris et al. [229] who provide a similar proof in the prototypical case of the Poisson equation. Clarifying the theoretical connection between SVD and PGD, Falcó and Nouy [230] establish the convergence of the latter—in the Galerkin and least-squares cases for symmetric and non-symmetric operators respectively—by generalizing the Eckart-Young theorem from finite-dimensional matrix spaces to tensor product Hilbert spaces. Meanwhile, the convergence of greedy algorithms for eigenvalue problems (including two novel methods) is analyzed by Cancès et al. [190], of obvious import to the original research on the same of Chapter 5.

A common feature of the above proofs is that each considers either symmetric (typically also coercive and continuous) operators exclusively or the least-squares (rather than Galerkin) PGD. As reviewed thoroughly by Cancès [198], this is by necessity; convergence results of
PGD for non-symmetric operators have yet to be established (to the author’s knowledge) and, although many alternative PGDs have been proposed, only the least-squares PGD is well-understood (because simply the renders the problem symmetric). Despite this, Galerkin PGD is often found to converge in practice for non-symmetric problems such as the Fokker-Planck equation [3], [127], advection-diffusion-reaction systems [9], and the neutron transport problems explored here (among others). Even so, in the pursuit of a more reliable method, a popular resolution is provided by the least-squares (also called minimal residual) PGD [9]. This amounts to multiplying by $C^\top$ (more generally, the adjoint operator $C^\dagger$ in the appropriate inner product space) to transform the non-symmetric problem $Cz = f$ into an equivalent symmetric positive-definite system

$$C^\top Cz = C^\top f$$

(B.1)

often termed the normal equations. More specifically, this is typically accomplished by redefining the inner products formulating the problem from $\langle \bullet, C\bullet \rangle = \langle \bullet, f \rangle$ to

$$\langle C\bullet, C\bullet \rangle = \langle C\bullet, f \rangle.$$  

(B.2)

As this renders the problem symmetric, the better-understood properties of PGD in that case apply immediately. In practice, however, this condition number of $C^\top C$ is the square of that of $C$, foreseeably leading to numerical difficulties. Indeed, several authors have found that while least-squares PGD does yield a convergent decomposition, the convergence (with increasing rank $m$) is slower than observed in the Galerkin PGD [5], [9], [231] (even for non-symmetric operators, provided it converges). A means of avoiding this difficulty may lie in the ideal minimal residual PGD [232], which aims to retain the benefits of least-squares but to simultaneously enhance the decomposition by introducing an operator which transforms the residual norm into an ideal one (generally, some approximation of the error norm).

An alternative approach explored in Chapters 4 and 5 is that of the Minimax PGD [9]; this method also modifies the inner product, but specifically by defining the test functions as the solution of auxiliary adjoint problems (see Section 4.4.5 for more details). Although to the author’s knowledge, a theoretical proof for the optimality of Minimax PGD has only been established for rank-one ($R = 1$) separable operators (symmetric or otherwise) [9], numerical experiments—including those of this thesis—suggest the achieved decomposition is often
superior to that of the Galerkin PGD. In any case, it is evident that the theoretical analysis of PGD for non-symmetric operators remains an evolving field, albeit with some encouraging early results.

B.3 Convergence Bounds of Low-Rank Tensor Approximations

Meanwhile, we next turn to estimating the best- and worst-case convergence rates one can expect of a low-rank approximation. Specifically, in the finite-dimensional case, if $C$ is a tensor of order two (that is, a matrix) the convergence of PGD can be tightly bounded from below by the SVD (or an equivalent eigendecomposition) of the solution, calculated either in the Frobenius norm or a weighted norm, and from above by simply assuming the singular values of this optimal decomposition are constant. That said, these two bounds are of limited practical utility. On the one hand, computing the SVD of $Z$ (where $\text{vec}(Z) = z$) requires the solution to the full-order model, $z = C^{-1}f$, and is typically overly optimistic for Progressive PGDs (though not Subspace PGDs [9]). Naturally, overcoming this difficulty requires a means of estimating the singular values of $Z$, without knowledge of $z$, at a speed competitive with simply computing the decomposition itself—that is, $Z$ and its SVD or a PGD approximation of the same. Accomplishing this appears challenging, but may be possible using state-of-the-art techniques in spectrum approximation [233] and/or matrix sketching [234] (especially if the Kronecker structure of $C$ is exploited). On the other hand, an upper bound follows trivially from the observation that if $m^* = \min(A, B)$, then the best rank-$m < m^*$ approximation $Z_m$ of $Z$ in the Frobenius norm $\|\bullet\|_F$ satisfies

$$\|Z - Z_m\|_F^2 = \sum_{m' = m+1}^{m^*} \sigma_{m'}^2$$ (B.3)

and so, as the singular values cannot increase ($\sigma_m \geq \sigma_{m+1} \geq 0$), the worst-case scenario is that each is equal, $\sigma_1 = \sigma_2 = \ldots = \sigma_{m^*} \equiv \sigma_*$. In this case, even the best rank-$m$ approximation admits an error of

$$\|Z - Z_m\|_F^2 = (m^* - m)\sigma_*^2$$ (B.4)

---

28 Or for higher-order tensors, a generalization of the same such as Canonical Polyadic Decomposition [222], Tucker Decomposition, or even PGD itself [5] (though these are known to be sub-optimal).
or, relative to $\|Z\|_F^2 = m^* \sigma_*^2$:

$$\frac{\|Z - Z_m\|_F}{\|Z\|_F} = 1 - \sqrt{m/m^*}. \quad (B.5)$$

A similar bound for $m = 1$ in a finite-dimensional tensor space of order $D$ such as $\mathbb{D} \equiv \bigotimes_{d=1}^D \mathbb{R}^{N_d}$ where $2 \leq N_1 \leq \ldots \leq N_D$ is proven by Qi [235] to be

$$\frac{\|Z - Z_1\|_F}{\|Z\|_F} \leq 1 - \varepsilon \quad (B.6)$$

based on the best rank-one approximation ratio $\varepsilon$, as in

$$\frac{1}{\sqrt{N_1 \times \ldots \times N_{d-1}}} \leq \varepsilon \leq 1. \quad (B.7)$$

For $D = 2$ in specific, $\varepsilon$ equals this lower bound, coinciding with the result of Equation B.5. Further estimates of this ratio are provided for more restrictive spaces of symmetric or biquadratic tensors. Naturally, these estimates establish an upper bound for the convergence rate of algorithms based on successive rank-one corrections, such as Progressive PGD. In contrast to the best-case estimate by decomposing $Z$, these results are extremely general—based only on the dimensions and properties of the tensor space—and do not require knowledge of $z$ or $C^{-1}$ (or even $C$ and $f$) or any other quantities which are non-trivial to compute. That being said, two practical issues are evident. First, this establishes only that an algorithm converges given the best rank-one correction, not an iterative approximation of the same computed in floating point arithmetic. In practice, it has been repeatedly observed [9], [231] that Progressive PGD may fail to obtain an exact decomposition even when one is available (as when $m > m^*$), including in the first example of this thesis where $m^* = 7$. In so far as any such greedy algorithm amounts to successively deflated Power Iteration on some unknown matrix (or tensor) $Z^T Z$ or $ZZ^T$, it is plausible this is the analogue of the well-known failure of the classic method (that is, on an known matrix $A$) when computed numerically: namely, that accumulating round-off errors can (unless carefully corrected) quickly deteriorate the theoretical convergence rate. A mathematical analysis which accounts for this may therefore be less optimistic. The second, and in some sense, opposite difficulty is that these estimates may (in the interest of generality) be overly pessimistic, in that assuming the singular values are uniform (each equal to $\sigma_*$) precludes the existence of a low-rank approximation, which is
typically the hypothesis that motivates these methods. Given this, in perhaps the most useful applications of these methods, the solution will never approach the hypothetical maximum rank, in which case this bound appears inappropriately conservative.

Altogether, these analyses suggest that while sound means of estimating the convergence of PGD and related algorithms exist, the judicious choice of applications—guided by analytical, physical, and/or experiential (that is, numerical) evidence of a suitably accurate low-rank approximation—is still practically essential. Indeed, demonstrating the existence (by SVD) and practical computation (by PGD) of such approximations was an overarching objective of this thesis. Moving forward, further investigation into the accumulation of numerical errors seems prudent; moreover, if these can be rectified, it is conceivable Progressive PGD could approach the convergence rates of Subspace PGD, yielding an immensely more accurate and economical ROM.
APPENDIX C
MATRIX-FREE TRANSPORT SWEEPS

It is worthwhile to explain the matrix-free sweeps ubiquitously exploited in radiation transport, as these affect the above work in multiple, subtle ways—most notably, by obviating storage of $\psi$ in the full-order model, but apparently not the PGD ROM. As will be seen, apart from the nuances of the matrix-free approach, these sweeps amount to a block Gauss-Seidel iteration on $L_g$, where each block corresponds to a single element.

C.1 Solving the Local System

In particular, for a given group $g$ and ordinate $n$, on each element $K$, the statement of neutron conservation is

$$\left(\psi^*, \tilde{\Omega} \cdot \nabla \psi\right)_K + \left(\psi^*, \Sigma_t \psi\right)_K = \left(\psi^*, q\right)_K \quad (C.1)$$

where $q$ here accounts for the contributions of both an arbitrary source and in-scattering. Further, we have suppressed the subscripts $n$ and/or $g$ of $\psi$, $q$, $\tilde{\Omega}$, and $\Sigma_t$ for brevity. Applying the divergence theorem yields

$$-\left(\tilde{\Omega}_n \cdot \nabla \psi^*, \psi\right)_K + \left(\tilde{\Omega}_n \cdot \bar{n} \psi^*, \psi\right)_{\partial K} + \left(\psi^*, \Sigma_t \psi\right)_K = \left(\psi^*, q\right)_K . \quad (C.2)$$

However, the surface integral over $\partial K$ is ambiguous, as the flux $\psi$ is discontinuous—and therefore undefined—on the interface between elements. The ubiquitous resolution is to “upwind” the flux, by selecting $\psi$ to be the upwind flux at the interface

$$\psi(\vec{r} \in \partial K) \equiv \begin{cases} 
\psi_K(\vec{r}) & \vec{r} \in \partial K^+ \\
\psi_K'(\vec{r}) & \vec{r} \in \partial K^-_{int} \equiv \partial K^- \setminus \partial \mathcal{V} \\
\psi_{in}(\vec{r}) & \vec{r} \in \partial K^-_{ext} \equiv \partial K^- \cap \partial \mathcal{V}
\end{cases} \quad (C.3)$$

where the outflow/inflow surfaces are defined as

$$\partial K^\pm \equiv \left\{ \vec{r} \in \partial K : \tilde{\Omega} \cdot \bar{n}(\vec{r}) \gtrless 0 \right\} \quad (C.4)$$
and $\psi_K$ denotes the flux interior to element $K$ (and likewise the source $q_K$). Substituting Equation C.3 into Equation C.2, the surface integral is split into three terms representing outflow, inflow from neighbor, and inflow from boundary. That is,

$$- \left( \vec{\Omega} \cdot \nabla \psi^*, \psi_K \right)_K + \left( \vec{\Omega} \cdot \vec{n} \psi^*, \psi_K \right)_{\partial K^+} + (\psi^*, \Sigma t \psi_K)_K$$

$$= (\psi^*, q)_K - \left( \vec{\Omega} \cdot \vec{n} \psi^*, \psi_K \right)_{\partial K_{\text{int}}} - \left( \vec{\Omega} \cdot \vec{n} \psi^*, \psi_{\text{in}} \right)_{\partial K_{\text{ext}}}$$ \hspace{1cm} (C.5)

where $K'$ is the upwind element neighboring $K$. Substituting in the basis functions (and enumerating the faces $f$ comprising $\partial K$) yields the “local” system of equations,

$$\begin{pmatrix}
G_K + \Sigma t B_K + \sum_{f \in \partial K^+} H^+_{K,f}
\end{pmatrix} \psi_K = B_K q_K - \sum_{f \in \partial K_{\text{int}}} H^+_{K,f} \psi_{K'} - \sum_{f \in \partial K_{\text{ext}}} H^+_{K,f} \psi_{\text{in}}$$ \hspace{1cm} (C.6)

where $K'_f$ is the upwind element neighboring $K$ on face $f$. Equivalently, gathering the left- and right-hand-side terms into a single matrix and vector respectively,

$$L_K \psi_K = \bar{q}_K.$$ \hspace{1cm} (C.7)

Note that cross section $\Sigma_t$ is constant in $K$, such the mass matrix $B_K$ does not depend on group $g$. Rather, the elements

$$B_{K,i,j} \equiv \int_K b_i b_j dr$$ \hspace{1cm} (C.8)

depend only on cell $K$ and basis functions $b(\bullet)$. Likewise, the streaming and interfacial matrices $G_K, H^+_{K,f}$, and $H^-_{K,f}$ have entries

$$G_{n,K,i,j} \equiv \vec{\Omega}_n \cdot \vec{G}_{K,i,j},$$ \hspace{1cm} (C.9)

$$H^+_{n,K,i,j} \equiv \vec{\Omega}_n \cdot \vec{H}^+_{K,f,i,j},$$ \hspace{1cm} (C.10)

$$H^-_{n,K,i,j} \equiv \vec{\Omega}_n \cdot \vec{H}^-_{K,f,i,j}.$$ \hspace{1cm} (C.11)
which can evidently be expressed in terms of vectors

\[ \vec{G}_{K,i,j} \equiv \int_K \nabla b_i b_j dr \quad (C.12) \]
\[ \vec{H}^+_{K,f,i,j} \equiv \int_{\partial K_f} \vec{n} b_i b_j ds \quad (C.13) \]
\[ \vec{H}^-_{K,f,i,j} \equiv \int_{\partial K_f} \vec{n} b_i b_{K'_{f},j} ds \quad (C.14) \]

which do not depend on ordinate \( n \) (nor group \( g \)). As such, \( \mathbf{B}_K \) and Equations C.12–C.14 comprise the “elementary matrices” which are pre-computed and stored throughout the simulation [152]. Meanwhile, those matrices appearing in Equation C.6—which do depend on group \( g \) and/or ordinate \( n \)—are constructed as needed from these elementary terms and, after the equation has been solved for \( \Phi_K \) (directly, as by Gaussian Elimination), overwritten [152]. As such, the global matrix is never stored in its entirety, hence the descriptor “matrix-free.” These nuances aside, the above procedure simply represents the solution of one diagonal block within the Gauss-Seidel procedure.

C.2 Downstream Ordering

Inverting the whole lower triangle is then accomplished by block forward substitution—essentially, solving each diagonal block in sequence. That said, instead of actually renumbering the degrees-of-freedom such that \( \mathbf{L}_g \) is (almost) lower block-triangular, it is convenient to simply visit each block in the correct (downstream) ordering [150]. This pre-computed sweep order is naturally expressed as a Directed Acyclic Graph (DAG). Such an ordering is not always possible, as cycles can be induced by misshapen elements and reflecting boundaries; in this case, cyclic dependencies must be broken by lagging either cell, instead reading(/writing) its degrees-of-freedom from(/into) the significant angular flux \( \overline{\Phi} \) and creating a nonzero block in the upper triangle \( \mathbf{L}_g \) [151]. Apart from those degrees-of-freedom, the only angular unknowns which need be stored are those along the propagating “sweep front”; at least, this is so in the full-order model, but not the PGD ROM. As such, the latter benefits from the memory savings of the matrix-free sweep application (not storing the global matrix) but not the savings of discarding the angular flux \( \Phi \) (unless one chooses to recompute them as needed by an extra transport sweep, see Equation 3.49). Recovering this useful property of the full-order model, perhaps by modifying the PGD procedure, appears then to be a
valuable objective for future research.
APPENDIX D
SINGULAR VALUE DECOMPOSITION IN THE $L^2$ NORM

While not itself a ROM, the Singular Value Decomposition (SVD) achieves a provably optimal rank $M$ approximation—that with the minimum error in the Frobenius norm [236]—in two dimensions of a given matrix. In our case, we decompose the solution $\psi$, expressed as a rectangular matrix, and subsequently rearrange the result back into a vector. Doing so, this guarantee in the Frobenius norm is equivalent to one of the resultant vector in the $\ell^2$ norm. As such, SVD provides an ideal separation against which one can benchmark the performance of PGD. As an aside, the SVD is also invoked to decompose the snapshot matrix, a compilation of reference solutions, to find modes for POD. Accordingly, the SVD results should also be similar to those realizable by POD, at least in the verification case where the problem of interest is identical to that from which the snapshots were generated.

However, the $L^2$ norm is a more natural choice to measure the performance of our decomposition. Conveniently, following the discussion of [242], for some function $u(x)$ where $x \in X$, one can relate the $L^2$ to the $\ell^2$ norm by

$$\|u\|_{L^2(X)}^2 = \int_X u^2 dx = (u, u)_X \approx u^\top B_X u = \|\sqrt{B_X} u\|_{\ell^2}^2,$$  \hspace{1cm} (D.1)

where $u$ is a vector of degrees-of-freedom and matrix $B_X$ implements this inner product, consistent with the discretization of $u$—in finite element methods, being the “mass matrix.” Lastly, $\sqrt{\cdot}$ here denotes the Choleksy factorization. Analogously, for some variable $y$ with domain $Y$, one can relate the $L^2(Y)$ and $\ell^2$ norms using $B_Y$. Supposing we have a matrix $A$ which represents a two-dimensional function $u(x, y)$, such as by

$$A = \begin{bmatrix} u(x_1, y_1) & \cdots & u(x_1, y_{N_y}) \\
\vdots & \ddots & \vdots \\
u(x_N, y_1) & \cdots & u(x_N, y_{N_y}) \end{bmatrix},$$ \hspace{1cm} (D.2)


Separation into three or more dimensions requires generalization, as provided by the High-Order- (HO-) SVD [237], [238], Canonical Polyadic Decomposition [239]–[241], or even PGD [5] among others. Unlike SVD, these algorithms are not necessarily optimal.
the SVD of $A$ yields the optimal decomposition in the Frobenius norm, while that of

$$\sqrt{B_x} A \sqrt{B_y} = U \Sigma V^\top$$  \hfill (D.3)

yields that in the $L^2(\mathcal{X} \times \mathcal{Y})$ norm, $X \Sigma Y^\top$, where

$$X = (\sqrt{B_x})^{-1} U,$$  \hfill (D.4)

$$Y = \left(\left(\sqrt{B_y}\right)^{-1}\right)^\top V.$$  \hfill (D.5)

Similarly, we apply this procedure to decompose the flux $\psi$ in the $L^2$ norm, where the analogues of $x$ and $y$ are space-angle and energy in Chapters 4 and 5 and some sets of axial and radial variables in Chapter 6. While we do not repeat this explanation specific to our selected discretizations, the corresponding matrices to $B_x$ and $B_y$ can be readily inferred from the discussions in Sections 4.4.10, 4.4.9, and 6.2.
APPENDIX E
ADDENDUM TO CHAPTER 4

E.1 Definition of the CASMO-SH-70 Energy Mesh

Since the group boundaries of CASMO-70 and SHEM-361 are not exactly aligned, it is advantageous to construct an analogous structure, CASMO-SH-70, so as to avoid splitting groups, which would require assumptions of the intra-group flux distribution. To do so, we use a straightforward procedure, explained as follows. First, we set the first and last coarse bounds to coincide with those of the fine mesh. Then, we round every other coarse bound, starting with the second highest energy, either up or down to the nearer fine bound—unless rounding up would result in a repeated coarse bound, in which case we round down. Having done so, the result is a coarse mesh with approximates the reference mesh, but coincides with the fine bounds such that each coarse-group is a summation of fine-groups. For our case, we find the resultant CASMO-SH-70 mesh, defined in Table E.1, to be an acceptable analogue for CASMO-70 based on the comparison in Figure E.1.

![Figure E.1: Visual comparison of CASMO-70, CASMO-SH-70, and SHEM-361 structures.](image)


Table E.1: CASMO-SH-70 structure as inclusive ranges of SHEM-361 groups.

<table>
<thead>
<tr>
<th>Coarse</th>
<th>Fine</th>
<th>Coarse</th>
<th>Fine</th>
<th>Coarse</th>
<th>Fine</th>
<th>Coarse</th>
<th>Fine</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1–9</td>
<td>19</td>
<td>69–72</td>
<td>37</td>
<td>315</td>
<td>55</td>
<td>341</td>
</tr>
<tr>
<td>2</td>
<td>10–12</td>
<td>20</td>
<td>73–86</td>
<td>38</td>
<td>316–317</td>
<td>56</td>
<td>342–343</td>
</tr>
<tr>
<td>3</td>
<td>13–14</td>
<td>21</td>
<td>87–113</td>
<td>39</td>
<td>318</td>
<td>57</td>
<td>344–345</td>
</tr>
<tr>
<td>4</td>
<td>15–18</td>
<td>22</td>
<td>114–134</td>
<td>40</td>
<td>319</td>
<td>58</td>
<td>346–347</td>
</tr>
<tr>
<td>5</td>
<td>19–23</td>
<td>23</td>
<td>135–153</td>
<td>41</td>
<td>320–321</td>
<td>59</td>
<td>348</td>
</tr>
<tr>
<td>6</td>
<td>24–26</td>
<td>24</td>
<td>154–171</td>
<td>42</td>
<td>322</td>
<td>60</td>
<td>349</td>
</tr>
<tr>
<td>7</td>
<td>27–30</td>
<td>25</td>
<td>172–201</td>
<td>43</td>
<td>323</td>
<td>61</td>
<td>350</td>
</tr>
<tr>
<td>8</td>
<td>31–33</td>
<td>26</td>
<td>202–223</td>
<td>44</td>
<td>324</td>
<td>62</td>
<td>351</td>
</tr>
<tr>
<td>9</td>
<td>34–37</td>
<td>27</td>
<td>224–276</td>
<td>45</td>
<td>325</td>
<td>63</td>
<td>352</td>
</tr>
<tr>
<td>10</td>
<td>38–40</td>
<td>28</td>
<td>277–280</td>
<td>46</td>
<td>326–327</td>
<td>64</td>
<td>353</td>
</tr>
<tr>
<td>11</td>
<td>41–43</td>
<td>29</td>
<td>281–288</td>
<td>47</td>
<td>328–330</td>
<td>65</td>
<td>354</td>
</tr>
<tr>
<td>12</td>
<td>44–49</td>
<td>30</td>
<td>289–295</td>
<td>48</td>
<td>331–333</td>
<td>66</td>
<td>355</td>
</tr>
<tr>
<td>13</td>
<td>50–53</td>
<td>31</td>
<td>296–297</td>
<td>49</td>
<td>334</td>
<td>67</td>
<td>356–357</td>
</tr>
<tr>
<td>14</td>
<td>54–56</td>
<td>32</td>
<td>298–301</td>
<td>50</td>
<td>335–336</td>
<td>68</td>
<td>358</td>
</tr>
<tr>
<td>15</td>
<td>57–59</td>
<td>33</td>
<td>302–306</td>
<td>51</td>
<td>337</td>
<td>69</td>
<td>359</td>
</tr>
<tr>
<td>16</td>
<td>60–61</td>
<td>34</td>
<td>307–310</td>
<td>52</td>
<td>338</td>
<td>70</td>
<td>360–361</td>
</tr>
<tr>
<td>17</td>
<td>62–65</td>
<td>35</td>
<td>311–312</td>
<td>53</td>
<td>339</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>66–68</td>
<td>36</td>
<td>313–314</td>
<td>54</td>
<td>340</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

E.2 Comparison of Minimax and Galerkin PGD

Although Section 4.4.5 explained the methodologies of both Galerkin and Minimax PGD, numerical results of the former were only obtained for one of the several analyses detailed in Section 4.5—namely, comparison against the full-order model on the UO$_2$ and MOX Cathalau pin-cells. To avoid confusion, the latter section details only the Galerkin results, with the comparison between the two being instead provided here. In particular, from Figures E.2 and E.3, one can see that the Minimax PGD is indeed more accurate than the Galerkin, both with and without update. In the latter case, we observe the benefit of using Minimax PGD is pronounced, while it is somewhat slighter for the PGD with update. To quantify the relative gains in accuracy from using Minimax PGD we plot in Figure E.4 the ratio between the Galerkin and Minimax errors. From this figure we see that in most cases the $L^2$ error incurred by Minimax PGD with 50 modes is half that or less than that of Galerkin PGD.$^{30}$ Moreover, from Figure E.5 we see that these gains in accuracy can be translated into a reduction in the number of required modes to meet a given accuracy.

---

$^{30}$The exceptions are the PGD with update in the CASMO-70 and SHEM-361 MOX pin-cells, where the ratios of Minimax to Galerkin angular errors are slightly higher at 0.60 and 0.55 respectively.
For example, given $M = 30$, the angular $L^2$ error of Minimax PGD is less than that of Galerkin PGD with 35, 38, or 40 modes for the CASMO-70, XMAS-172, and SHEM-361 cases respectively. Altogether, these results illustrate that both the update step and the Minimax formulation appreciably improve the decomposition. That said, there is still room for future improvement, as no PGD algorithm yet matches the convergence of the the SVD.

### E.3 Distribution of PGD and SVD Errors in Cathalau Pin-Cells

The spectral distribution of PGD and SVD errors in the Cathalau pin-cells reveals which energy ranges contribute the bulk of the error to the totals already discussed in Section 4.5.4. Figures E.6–E.9 show such trends for the PGD with update and SVD cases. Note that the average error in these plots excludes the last (slowest) group because its lethargy width is unusually large for XMAS-172 and undefined for CASMO-70 and SHEM-361, where
Figure E.3: Convergence of progressive Galerkin and Minimax PGD, with and without update, Cathalau MOX pin-cell.

$E_G = 0$. Given this, the average equals the total $L^2$ error divided by the square root of the lethargy width: that is, for the angular flux

$$
\sqrt{\frac{1}{\ln(E_0/E_{G-1})} \int_{u_0}^{u_{G-1}} \int_{4\pi} \int_V \left( \epsilon(\bar{r}, \bar{\Omega}, u) \right)^2 drd\Omega du},
$$

(E.1)

where the error $\epsilon$ has already been normalized by $\|\psi\|_{L^2}$. From these figures, we observe that errors are particularly high in groups corresponding to resonance peaks resolved by the problem’s group structure. Moreover, the error in the thermal groups is pronounced in some PGD cases, especially for $M = 10$. This could be related to the influence of downscattering in the PGD ROM, which will propagate errors to lower energies. Finally, we see that the SVD errors tend to be more uniform in energy and, of course, lower, since the SVD decomposition is optimal in the $L^2$ norm.
Figure E.4: Ratio of $L^2$ flux errors, Galerkin to Minimax PGD, with and without update.

Meanwhile, the spatial distribution indicates where in the pin-cell the errors are concentrated. As this distribution is different in each energy group, we plot in Figures E.10 and E.11 the spatially-dependent angular and scalar $L^1$ errors for the PGD with update, defined respectively as

$$\varepsilon_{L_1}^\psi(\vec{r}) \equiv \int_{u_0}^{u_G} \int_{4\pi} \epsilon(\vec{r}, \Omega, u) d\Omega du$$

(E.2)

and

$$\varepsilon_{L_1}^\phi(\vec{r}) \equiv \int_{u_0}^{u_G} \left| \int_{4\pi} \epsilon(\vec{r}, \Omega, u) d\Omega \right| du$$

(E.3)

which are both normalized by $\|\phi\|_{L^1}$ for ease of comparison. Note that $\|\phi\|_{L^1} = \|\psi\|_{L^1}$ if $\psi$ is nonnegative, which is physically necessary (a flux of less than zero is nonsensical), though negative values can still appear as numerical artifacts. From these figures, we see that the angular $L^1$ error is in most cases highest in the outer cladding, near the interface with the
Figure E.5: Number of additional modes required for Galerkin PGD to achieve an $L^2$ flux error less than or equal to that of Minimax PGD, with and without update. Missing points indicate the Galerkin PGD did not achieve an equal or lesser error by 50 modes.

moderator. The scalar error, however, often peaks in the moderator itself, in the upper right corner of the pin-cell where the thermal flux is concentrated. This is consistent with the energy distributions in Figures E.6 and E.7, which show that the error in the scalar flux is in many cases dominated by the thermal energies.
Figure E.6: Group-wise PGD with update errors, by group structure, Cathalau UO$_2$ pin-cell.
Figure E.7: Group-wise PGD with update errors, by group structure, Cathalau MOX pin-cell.
Figure E.8: Group-wise SVD errors, by group structure, Cathalau UO$_2$ pin-cell.
Figure E.9: Group-wise SVD errors, by group structure, Cathalau MOX pin-cell.
Figure E.10: Normalized $L^1$ error by group structure, PGD with update, Cathalau UO$_2$ pin-cell.
Figure E.11: Normalized $L^1$ error by group structure, PGD with update, Cathalau MOX pin-cell.
F.1 Fission Source Errors in Cathalau Pin-Cells

In Section 5.5.3, we illustrate convergence of the fission source \( q \) in the \( L^2 \) norm for the Galerkin and Minimax PGD on the Cathalau pin-cell benchmarks. However, analysts may be concerned with the spatial distribution of this error, rather than the norm alone—especially if a consistent spatial bias is found. To investigate this topic, we begin by plotting the reference fission sources achieved by the full-order models in Figure F.1. Notably, each fission source is normalized such that a uniform \( q \) would yield a flat profile of unit magnitude—accordingly, these plots can reveal differences in shape but not magnitude.

![Figure F.1: Normalized fission sources by fuel and group structure, Cathalau pin-cells.](image)

From these figures we observe that, for a given fuel, the distribution of the fission source appears largely independent of the group structure. That said, the gradient across the pin for MOX fuel is sharper than that for \( \text{UO}_2 \), which can plausibly be ascribed primarily to the effect of epithermal resonances in plutonium.

---

The corresponding errors achieved by Galerkin and Minimax PGD (and normalized by the same factors) are displayed in Figures F.2 and F.3 for UO$_2$ and MOX fuel respectively. From these plots, we observe that, given fifty modes, the fission sources are generally well-converged, with discrepancies less than $5 \times 10^{-4}$ in all but two of the twelve scenarios. This convergence can be erratic, however, as observed in Section 5.5.3 and evidenced here by abrupt drops in error—for instance, between thirty and forty modes in the case of CASMO-70 UO$_2$ with Galerkin PGD. As to a consistent spatial bias, no stark pattern is readily apparent in these results. That said, these questions should be revisited later for assembly and core problems in which there will be both an inter- and intra-pin distribution of the fission source.

Finally, if any readers are likewise interested in the spatial distribution of the flux errors, we refer them to [13], in which analogous results for a similar fixed-source problem (UO$_2$ and MOX Cathalau pin-cells with a uniform fission source) are presented.

F.2 Comparison to Cathalau Benchmark Participants

In Section 5.5.3, we roughly estimate the precision of the multigroup full-order model by comparing the calculated $k$-eigenvalues against those of the original, continuous-energy reference solution. This provides useful context for interpreting the relative magnitude of the errors incurred by PGD. However, these six simulations—two lattices modeled in three energy meshes—are a relatively small set from which to draw conclusions about the precision of such full-order models. Given this, it would be helpful to demonstrate that the precision we observe is typical of, or better than, that which can be expected in practice.

Conveniently, since the Cathalau pin-cells are a benchmark exercise, we can compare our results to those of the benchmark participants$^{32}$ [243], summarized in Table F.1 and illustrated as Gaussian Kernel Density Estimates (KDEs, with bandwidth selected by Scott’s rule [244]) in Figure F.4. By inspecting this plot, we can see that our OpenMC references appear reasonable compared to those reported by benchmark participants. Second, our three deterministic approximations—that is, our three multigroup models condensed in different energy meshes—based on each OpenMC reference appear to be clustered relatively tightly compared to the spread seen across the participants’ results: each differs from OpenMC by less than $2.2 \times 10^{-3}$.

$^{31}$Namely, SHEM-361 with Galerkin PGD for both UO$_2$ and MOX, in which case the errors are still less than $2.2 \times 10^{-3}$.

$^{32}$Results from SCK CEN have been excluded as outliers.
than one standard deviation of the Cathalau distribution for the corresponding fuel. As such, this brief study confers some additional confidence about the veracity of our Monte Carlo and deterministic full-order models and suggests the precision we find among our full-order models is indeed comparable or superior to that which would be anticipated in practice—especially when considering other factors like differing spatio-angular discretizations and nuclear data.

Table F.1: Cathalau benchmark statistics for pin-cell $k$-eigenvalues.

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Std. Dev.</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>UO₂</td>
<td>1.323301</td>
<td>545.8 pcm</td>
<td>1780.0 pcm</td>
</tr>
<tr>
<td>MOX</td>
<td>1.135881</td>
<td>468.3 pcm</td>
<td>1503.0 pcm</td>
</tr>
</tbody>
</table>
Figure F.3: Normalized fission source error, PGD with eigenvalue update, Cathalau MOX pin-cell.

Figure F.4: Cathalau benchmark results (black and red) and original results (green). Curves represent Gaussian Kernel Density Estimates (KDEs) and highlights denote standard deviations from the mean.
F.3 Computational Cost of Recursive LU Factorization

Recursive LU factorization has been extensively studied in other work. The algorithm used in Section 5.4.6 here is nearly equivalent to the “recursive pivot-free LU factorization” presented in [245], which performs $\frac{2}{3}N^3$ floating point operations (FLOPs) to leading order. The only potentially significant difference is that our algorithm does not partition the blocks equally (or almost equally for odd-sized blocks); instead, they are always partitioned along the last block column and row, where the blocks are of size $G$. For completeness, we briefly prove this distinction does not affect the required number of FLOPs to leading order.

Theorem 1. The recursive LU factorization of a $(M + 1)G \times (M + 1)G$ matrix presented in Algorithm 4 performs $\frac{2}{3}(M + 1)^3G^3$ FLOPs to leading order.

Proof. Let $f((M + 1)G)$ denote the number of FLOPs performed in the recursive LU factorization of a $(M + 1)G \times (M + 1)G$ as per Algorithm 4. At each recursion level $M > 0$, the following operations are performed at the indicated line numbers:

Line 1. $A_{11}$ has been factorized recursively: $f(MG)$ FLOPs

Line 3. $\text{xTRSM}$ is called to perform a triangular solve on $A_{21}$: $M^2G^3$ FLOPs

Line 4. $\text{xTRSM}$ is called to perform a triangular solve on $A_{12}$: $M^2G^3$ FLOPs

Line 5. $\text{xGEMM}$ is called to compute the Schur complement: $2MG^3$ FLOPs

Line 6. $\text{xGETRF}$ is called to factorize the Schur complement: $\frac{2}{3}G^3$ FLOPs.

Accordingly, the function $f$ can be defined recursively as

$$f((M + 1)G) = f(MG) + 2M^2G^3 + 2MG^3 + \frac{2}{3}G^3.$$  \hspace{1cm} (F.1)

To leading order, $f((M + 1)G)$ is of the form $\alpha(M + 1)^3G^3$. Then, by substitution

$$\alpha(M + 1)^3G^3 = \alpha M^3G^3 + 2M^2G^3 + 2MG^3 + \frac{2}{3}G^3$$  \hspace{1cm} (F.2)

Or, simplifying

$$\alpha(M + 1)^3 = \alpha M^3 + 2M^2 + 2M + \frac{2}{3}$$  \hspace{1cm} (F.3)

which can be solved to find $\alpha = \frac{2}{3}$.
For reference, a naive implementation which factorizes every matrix in sequence without exploiting the structure would cost

\[
\frac{2}{3} G^3 \sum_{m=1}^{M} m^3 = \frac{1}{6} G^3 M^2(M + 1)^2
\]  

(F.4)

or to leading order, \(\frac{1}{6} M^4 G^3\). By these estimates, the recursive algorithm requires roughly \(\frac{1}{4} M\) times fewer FLOPs for large \(M\).
G.1 Relating the Axial and $P_1$ Equations

The 1D submodel, Equation 6.55, of the Axial or 2D($\mu$)/1D PGD amounts to neutron transport with a two-point angular quadrature at $\mu = \pm 1$. While this is straightforward to solve numerically, it may be useful in some cases to recast this equation to an equivalent form resembling the familiar $P_1$ equations; this can be accomplished as follows. For convenience, a more typical nomenclature is recovered by renaming $Z_{m^*,g}^{\pm}$ to the partial currents $J_{m^*,g}^{\pm}$, and likewise the even and odd axial fluxes $Z_{m^*,g}^{e}$ and $Z_{m^*,g}^{o}$ to the scalar flux and current $\phi_{m^*,g}$ and $J_{m^*,g}$. Doing so, Equation 6.55 can be rewritten as

$$\sum_{m=1}^{M} \left[ \pm s_{1D,g}^{(m^*,m)} \frac{\partial}{\partial z} J_{m^*,g}^{\pm}(z) + \sum_{\ell,g \neq g'}^{G} \Sigma_{\ell,k,g' \rightarrow g}(z) \phi_{m^*,g'}^{\ell}(z) \right] = Q_{g}^{\pm}(z)$$

where

$$\phi_{g}' \equiv \begin{cases} \phi_{g}(z) \equiv J_{g}^{+}(z) + J_{g}^{-}(z) & \ell + k \text{ is even} \\ J_{g}(z) \equiv J_{g}^{+}(z) - J_{g}^{-}(z) & \ell + k \text{ is odd} \end{cases}$$

Evidently, these two equations relating $J_{g}^{+}$ and $J_{g}^{-}$ (or, for brevity, Equation G.1$^{\pm}$) can be transformed to those relating $\phi_{g}$ and $J_{g}$ by adding Equations G.1$^{+}$ and G.1$^{-}$ to yield

$$\sum_{m=1}^{M} \left[ s_{1D,g}^{(m^*,m)} \frac{\partial}{\partial z} J_{m^*,g}(z) + \sum_{\ell,g \neq g'}^{G} \Sigma_{\ell,k,g' \rightarrow g}(z) \phi_{m,g'}^{\ell}(z) \right] = Q_{g}^{0}(z)$$

198
while likewise Equation G.1\(^{+}\) minus Equation G.1\(^{-}\) yields

\[
\sum_{m=1}^{M} \left[ \sum_{l=0}^{L} (2l + 1) \sum_{k=-\ell}^{\ell} \sum_{g=1}^{G} \sum_{g'=1}^{G} \Sigma_{s,\ell,k,g' \to g}(z) J_{m,g}(z) \right] = Q_{\ell}^{g}(z)
\]

in which \(\Sigma_{a,e,g}(m^{*},m)(z)\) and \(\Sigma_{a,o,g}(m^{*},m)(z)\) are even and odd-parity absorption cross sections, as in, that of collision minus the corresponding moments of within-group scattering,

\[
\Sigma_{a,e,g}(m^{*},m)(z) \equiv \tilde{\Sigma}(m^{*},m)(z) - \sum_{\ell=0}^{L} (2\ell + 1) \sum_{k=-\ell}^{\ell} \frac{\Sigma_{s,\ell,k,g' \to g}(z)}{\text{even } \ell+k}
\]

\[
\Sigma_{a,o,g}(m^{*},m)(z) \equiv \tilde{\Sigma}(m^{*},m)(z) - \sum_{\ell=0}^{L} (2\ell + 1) \sum_{k=-\ell}^{\ell} \frac{\Sigma_{s,\ell,k,g' \to g}(z)}{\text{odd } \ell+k}
\]

and \(Q_{0}^{g}\) and \(Q_{1}^{g}\) are the zeroth and first-order—synonymously, even and odd parity—moments of the source

\[
Q_{0}^{g}(z) \equiv Q_{g}^{+}(z) + Q_{g}^{-}(z),
\]

\[
Q_{1}^{g}(z) \equiv Q_{g}^{+}(z) - Q_{g}^{-}(z).
\]

Equations G.3 and G.4 can readily be recognized as a system of equations similar to the common \(P_{1}\) equations in reactor physics. One could solve them in this form, or symbolically solve Equation G.4 for \(J_{m^{*},g}\)

\[
J_{m^{*},g}(z) =
\]

\[
\frac{1}{\Sigma_{a,o,g}(m^{*},m^{*})(z)} \left[ -\frac{\partial}{\partial z} \phi_{m^{*},g}(z) + \sum_{l=0}^{L} (2\ell + 1) \sum_{k=-\ell}^{\ell} \sum_{g'=1}^{G} \Sigma_{s,\ell,k,g' \to g}(z) J_{m^{*},g'}(z) + R_{m^{*},g}^{o}(z) \right]
\]
where $R_{m^*,g}^o$ is an odd residual term

$$R_{m^*,g}^o(z) \equiv Q_g^0(z) - \sum_{m \neq m^*}^M \left[ s_{1D,g}^{(m^*,m)} \frac{\partial}{\partial z} \phi_{m,g}(z) + \Sigma_{a,o,g}^{(m^*,m)}(z) J_{m,g}(z) \right] - \sum_{\ell=0}^L (2\ell + 1) \sum_{m=-\ell}^{\ell} \sum_{g' = 1}^G \Sigma_{s,\ell,k,g' \rightarrow g}^{(m^*,m)}(z) J_{m,g}(z).$$  \hfill (G.9)

This expression of $J_{m^*,g}$ can then be substituted into Equation G.3 to find

$$-\frac{\partial}{\partial z} \frac{1}{\Sigma_{a,o,g}^{(m^*,m)}}(z) \frac{\partial}{\partial z} \phi_{m^*,g}(z) + \Sigma_{a,e,g}^{(m^*,m)}(z) \phi_{m^*,g}(z) - \sum_{\ell=0}^L (2\ell + 1) \sum_{k=-\ell}^{\ell} \sum_{g' = \ell \neq g}^G \Sigma_{s,\ell,k,g' \rightarrow g}^{(m^*,m)}(z) S_{m^*}^p(z) = R_{m^*,g}^e(z) - \frac{\partial}{\partial z} \frac{1}{\Sigma_{a,o,g}^{(m^*,m)}} R_{m^*,g}^o(z)$$ \hfill (G.10)

where

$$S_{m^*}^p(z) \equiv \begin{cases} \phi_{m^*,g'}(z), & \ell + k \text{ is even}, \\ -\frac{\partial}{\partial z} \frac{1}{\Sigma_{a,o,g}^{(m^*,m)}}(z) J_{m^*,g'}(z) & \ell + k \text{ is odd}. \end{cases} \hfill (G.11)$$

and $R_{m^*,g}^e$ is the even residual term

$$R_{m^*,g}^e(z) \equiv Q_g^0(z) - \sum_{m \neq m^*}^M \left[ s_{1D,g}^{(m^*,m)} \frac{\partial}{\partial z} J_{m,g}(z) + \Sigma_{a,e,g}^{(m^*,m)}(z) \phi_{m,g}(z) \right] - \sum_{\ell=0}^L (2\ell + 1) \sum_{m=-\ell}^{\ell} \sum_{g' = 1}^G \Sigma_{s,\ell,k,g' \rightarrow g}^{(m^*,m)}(z) \phi_{m,g}(z).$$ \hfill (G.12)

In the case of isotropic scattering ($L = 0$), one can further simplify to

$$-\frac{\partial}{\partial z} \frac{1}{\Sigma_{t,g}^{(m^*,m)}}(z) \frac{\partial}{\partial z} \phi_{m^*,g}(z) + \Sigma_{a,g}^{(m^*,m)}(z) \phi_{m^*,g}(z) - \sum_{g' = \ell \neq g}^G \Sigma_{s,0,0,g' \rightarrow g}^{(m^*,m)}(z) \phi_{m^*,g'}(z)$$

$$= R_g^o(z) - \frac{\partial}{\partial z} \frac{1}{\Sigma_{t,g}^{(m^*,m)}}(z) R_g^o(z)$$ \hfill (G.13)

where

$$\Sigma_{a,g}^{(m^*,m)}(z) = \Sigma_{t,g}^{(m^*,m)}(z) - \Sigma_{s,0,0,g \rightarrow g}^{(m^*,m)}(z)$$ \hfill (G.14)
As such, this procedure demonstrates that Equation G.1 (likewise, Equation 6.55) can be equivalently stated in a form like that of the $P_1$ or diffusion equations. This may be useful in future PGD implementations or 2D/1D methods, especially as the diffusion approximation is generally the preferred 1D model in the latter case.
APPENDIX H
MODEL ORDER REDUCTION IN ENERGY FOR 1D NEUTRON DIFFUSION

H.1 Introduction

Herein, we present a PGD ROM of one-dimensional neutron diffusion separated in space and energy, with an isotropic continuous slowing-down scattering kernel. Numerical convergence with increasing grid refinement and solution enrichment is demonstrated using the Method of Manufactured Solutions (MMS). Lastly, the method is applied to a prototypical problem regarding the slowing down of fission neutrons, where monotonic convergence with enrichment is observed by global and local relative numerical indicators. This study serves as a proof-of-concept for the more elaborate models of Chapters 4 and 5.

H.2 Methodology

For the problem at hand, we seek the scalar flux $\phi$ in terms of spatial modes $X_m$ and lethargy modes $U_m$

$$\phi(x, u) \approx \sum_{m=1}^{M} X_m(x) U_m(u).$$  \hspace{1cm} (H.1)

H.2.1 Application to Neutron Diffusion

We begin with the steady-state, energy-dependent neutron diffusion equation in slab geometry with isotropic scattering and an isotropic source

$$- \nabla \cdot D(x, u) \nabla \phi(x, u) + \Sigma_t(x, u) \phi(x, u) - \int_{u_0}^{u_2} \Sigma_s(x, u' \rightarrow u) \phi(x, u') du' = Q(x, u)$$  \hspace{1cm} (H.2)

and decompose the total and differential scattering cross-sections $\Sigma_t(x, u)$ and $\Sigma_s(x, u' \rightarrow u)$ as

$$\Sigma_t(x, u) \equiv \sum_{j=1}^{J} N_j(x) \sigma_{t,j}(u), \quad (H.3)$$

$$\Sigma_s(x, u' \rightarrow u) \equiv \sum_{j=1}^{J} N_j(x) \sigma_{s,j}(u') P_j(u' \rightarrow u), \quad (H.4)$$

for $J$ unique nuclides with number densities $N_j$ and microscopic total and scattering cross-sections $\sigma_{t,j}$ and $\sigma_{s,j}$. Moreover, $P_j(u' \rightarrow u)$ is the probability a neutron scattering off nuclide $j$ will be transferred from lethargy $u'$ to $u$. Assuming, without loss of generality, the target is at rest (that is, neglecting thermal motion) relative to the neutron, one finds

$$P_j(u' \rightarrow u) = \begin{cases} e^{u'-u} \frac{1}{1 - \alpha_j}, & u' \in [u - \ln(1/\alpha_j), u] \\ 0, & \text{otherwise} \end{cases}, \quad (H.5)$$

with

$$\alpha_j \equiv \left( \frac{A_j - 1}{A_j + 1} \right)^2 \quad (H.6)$$

where $A_j$ is the mass number of nuclide $j$.

Note that one could also resort to a multigroup-collapsed formulation by dividing the lethargy scale into groups $g$ and writing

$$\Sigma_s(x, u' \rightarrow u) \equiv \sum_{h=1}^{H} \Pi_h(x) \frac{\sum_{g'=g'}^{g} \Pi_h(x) \Sigma_{s,h,g' \rightarrow g}}{\Pi_h(x)}, \quad u' \in [u_{g'-1}, u_{g'}] \land u \in [u_{g-1}, u_g], \quad (H.7)$$

$$\Pi_h(x) \equiv \begin{cases} 1, & x \in L_h \\ 0, & \text{otherwise} \end{cases}, \quad (H.8)$$

for $H$ unique materials $h$ each located on segment(s) $L_h$. (A similar form would likewise be assumed for the total cross-section, $\Sigma_t$.) However, this requires a priori knowledge of the flux $\phi$ to compute flux-averaged multigroup cross-sections $\Sigma_{s,h,g' \rightarrow g}$.
In a similar fashion to the cross-sections, we now take a separable form of source $Q$,

$$Q(x, u) = \sum_{i=1}^{I} Q_i(x) \chi_i(u). \quad \text{(H.9)}$$

Furthermore, assuming a homogeneous medium of a single nuclide $j$ with unit number density, Equation H.2 (omitting the subscript $j = 1$) simplifies to

$$-D(u) \frac{\partial^2}{\partial x^2} \phi(x, u) + \sigma_t(u) \phi(x, u) - \int_{u_s}^{u} \frac{\sigma_s(u') e^{u'-u}}{1 - \alpha} \phi(x, u') du' = \sum_{i=1}^{I} Q_i(x) \chi_i(u) \quad \text{(H.10)}$$

where

$$u_s = \max(u - \ln(1/\alpha), u_0). \quad \text{(H.11)}$$

Substituting Equation H.1 into Equation H.10 then yields the separable form of the full-order model. Subsequently, the spatial and lethargic submodels—those defining the last modes $X_M$ and $U_M$—are found by integrating this separable model in lethargy or space against weight function $U_M$ or $X_M$ respectively. From these submodels, an arbitrarily high-rank decomposition (as in Equation H.1) can be progressively constructed by starting at $M = 1$, computing the last modes, incrementing $M$, then repeating the latter two steps until convergence.

**H.2.2 Spatial Submodel**

Denoting the weak form in lethargy as

$$\int_{u_0}^{u_G} \mathbf{u} \cdot U_M du = (U_M, \mathbf{u})_G \quad \text{(H.12)}$$

the spatial submodel defining $X_M$ is found to be

$$\sum_{m=1}^{M} \left[ -\hat{D}_{M,m} \frac{d^2}{dx^2} X_m(x) + \hat{\Sigma}_{a,M,m} X_m(x) \right] = \hat{Q}_M(x) \quad \text{(H.13)}$$
where

\[
\tilde{D}_{M,m} = \left( U_M(u), D(u)U_m(u) \right)_G,
\]

(H.14)

\[
\tilde{\Sigma}_{a,M,m} = \left( U_M(u), \sigma_t(u)U(u) \right)_G - \left( U_M(u), \int_{u_s}^u \frac{\sigma_s(u')e^{u'-u}}{1 - \alpha} U_m(u')du' \right)_G,
\]

(H.15)

\[
\tilde{Q}_M(x) = \sum_{i=1}^I \left( U_M(u), \bar{\chi}_n(u) \right)_G Q_i(x).
\]

(H.16)

In the Progressive PGD, one assumes that modes \( m < M \) are known (previously computed) from a previous iteration; as such, Equation H.13 could be simplified by defining a residual incorporating the source minus the known summands. However, the equation is still nonlinear, as it contains products of unknowns \( X_M \) and \( U_M \). This is resolved here by simple fixed-point iteration, as described in Section H.2.4. Doing so, Equation H.13 becomes a linear diffusion-reaction equation, akin to traditional one-group neutron diffusion. At present, we discretize this submodel using centered finite-differences.

**H.2.3 Lethargic Submodel**

Likewise, integrating along the length of the slab \( \mathcal{L} \equiv [x_L, x_R] \) against test function \( X_M \),

\[
\int_{x_L}^{x_R} \cdot X_M dx \equiv \left( X_M, \cdot \right)_\mathcal{L},
\]

(H.17)

yields the lethargic submodel

\[
\sum_{m=1}^M \left[ \tilde{\Sigma}_{t,M,m}(u)U_m(u) - \tilde{N}_{M,m} \int_{u_s}^u \frac{\sigma_s(u')e^{u'-u}}{1 - \alpha} U_m(u')du' \right] = \tilde{\chi}(u)
\]

(H.18)

in which

\[
\tilde{\Sigma}_{t,M,m}(u) \equiv \tilde{N}_{M,m}\sigma_t(u) - \left( X_M(x), \frac{d^2}{dx^2} X_m(x) \right)_\mathcal{L} D(u),
\]

(H.19)

\[
\tilde{N}_{M,m} \equiv \left( X_M(x), X_m(x) \right)_\mathcal{L},
\]

(H.20)

\[
\tilde{\chi}_M(u) \equiv \sum_{i=1}^I \left( X_M(x), Q_i(x) \right)_\mathcal{L} \bar{\chi}_i(u).
\]

(H.21)
Together with Equation H.13, this forms a nonlinear system of two equations. Linearizing as in the following section, Equation H.18 becomes a linear Volterra integral equation of the second kind, tantamount to a model of neutron slowing-down in an infinite medium. Discretization proceeds by a trapezoidal expansion method, expanding $U_m$, $\sigma_t$, and $\sigma_s$ as nodal values interpolated by first-order Lagrange polynomials.

**H.2.4 Fixed-Point Iteration**

Here, fixed-point iteration is selected for the resolution of our nonlinear system. Specifically, we initialize $U_M^{(0)}$ as a uniform distribution (in lethargy), then solve Equation H.13 for $X_M^{(1)}$, treating $U_M \leftarrow U_M^{(0)}$ as a known. This alternating iteration is repeated until convergence, as measured by a suitable indicator, here taken to be a relative $\ell^2$ metric

$$
\varepsilon^{(k)} = \frac{\|X_M^{(k)} U_M^{(k)} - X_M^{(k-1)} U_M^{(k-1)}\|_{\ell^2}}{\|X_M^{(k-1)} U_M^{(k-1)}\|_{\ell^2}}, \quad (H.22)
$$
on which we require $\varepsilon^{(k)} < 10^{-6}$ for convergence in the numerical experiments that follow.

Additionally, as the solution of our nonlinear system is truly some product of the constituent modes $X_M \times U_M$, we find it prudent to normalize one mode within each Picard iteration

$$
U_M^{(k)} \leftarrow \frac{U_M^{(k)}}{\|U_M^{(k)}\|_{\ell^2}} \quad (H.23)
$$
to avoid “drifting” of modes between iterations, where one mode shrinks while the other grows. We summarize our PGD algorithm, separated in space and lethargy, in Algorithm 7.

**H.3 Numerical Results**

**H.3.1 Method of Manufactured Solutions**

We begin by applying the Method of Manufactured Solutions (MMS) for an exactly separable function

$$
\phi(x, u) = \sigma_s^{-1}(u) \sin(\pi x) \sin \left( \frac{3\pi (10 \ln(1/\alpha) - u)}{10 \ln(1/\alpha)} \right) \quad (H.24)
$$

---

33 That is, assuming target-at-rest kinematics; upscattering yields a Fredholm rather than Volterra equation. Moreover, steady-state fission would introduce a nonlinear (eigenvalue) Fredholm operator.
Algorithm 7: Space-Lethargy PGD

// Initialize \( \phi \), satisfying left/right boundary conditions
1 \( \phi \leftarrow X_L U_L + X_R U_R \) // mode(s) \( L \) and/or \( R \) omitted if zero
2 for \( m \leftarrow 1, 2, \ldots M \) do Enrichment Iteration
3 Initialize \( U_m \) as guess // here, uniform
4 while \( \varepsilon > \) tolerance do Fixed-Point Iteration // tolerance here is \( 10^{-6} \)
5 \( X_m \leftarrow \) solution of Equation H.13 given \( U_m \)
6 \( U_m \leftarrow \) solution of Equation H.18 given \( X_m \)
7 \( U_m \leftarrow U_m / \| U_m \|_\ell^2 \) // normalization
8 \( \varepsilon \leftarrow \) compute by Equation H.22
9 \( \phi \leftarrow \phi + X_m U_m \)

which can be represented in a single mode (as in, \( M = 1 \)). Selecting our cross-sections and diffusion coefficient

\[
\begin{align*}
\sigma_t & \leftarrow 1/E = e^u/E_0, \\
\sigma_s & \leftarrow \sigma_t, \\
D & \leftarrow \sigma_s/(3\sigma_t^2),
\end{align*}
\]

(H.25)  (H.26)  (H.27)

we solve for a slab \( L \leftarrow [0, 1] \) with atomic mass\(^{34} \) \( A = 12 \) over lethargy interval \( G \leftarrow [0, 10 \ln(1/\alpha)] \) given homogenous Dirichlet boundary conditions—that is, \( \phi(x_L, u) = \phi(x_R, u) = 0 \). We find the PGD implementation converges to the numerical solution within one mode, such that all subsequent modes are of negligible magnitude. Comparing the numerical error of our method with increasing grid refinement, we achieve Figure H.1 where the \( \ell^2 \) error is defined as

\[
\ell^2 \text{ Error} \equiv \frac{\| \phi - \phi_{\text{exact}} \|_{\ell^2}}{\| \phi_{\text{exact}} \|_{\ell^2}}
\]

(H.28)

which suggests we have achieved second-order, or \( O(h^2) \), accuracy (where \( h \) is conventionally the “mesh width” and is inversely proportional to the number of grid points \( N - 1 \)).

Satisfied with the numerical accuracy of our method, we next move to characterize a manufactured solution which requires infinitely many modes to represent exactly

\[
\phi(x, u) = \sin \left( \pi \left( 1 - x^2 \right) \left( 1 - u^2 \right) \right)
\]

(H.29)

\(^{34}\)This value was chosen so as to represent a graphite moderator of \(^{12}\)C.
and show the error with increasing number of modes $M$ in Figure H.2. For this problem, we redefine the total cross section as

$$\sigma_t \leftarrow E = E_0 e^{-u}$$  \hspace{1cm} \text{(H.30)}$$

where $\sigma_s$ and $D$ remain as in Equations H.26 and H.27, though containing the above value of $\sigma_t$. Further, the slab domain is set as $\mathcal{L} \leftarrow [-1, 1]$ (on which homogeneous Dirichlet boundaries are imposed) while that in lethargy is $\mathcal{G} \leftarrow [-1, 1]$; meanwhile, the atomic mass is $A = 2$ and the number of grid points $N = 500$. As the manufactured source will also be of infinite rank, we decompose the discrete source via SVD to obtain a form compatible with Equation H.9.

From Figure H.2, we observe near optimal separation up to the first three modes, almost matching the ideal decomposition of the exact solution given by SVD. However, the error then stagnates, indicating we have reached the discretization error incurred by numerical evaluation of our differential (spatial diffusion) and integral (lethargic scattering) operators.
H.3.2 Slowing Down of Fission Neutrons

We move now to a more physically meaningful problem. Specifically, let us simulate the flux arising from a source

\[ Q(x, u) \leftarrow \Pi(x)\chi(u) \]  \hspace{1cm} (H.31)

where \( \Pi \) represents a uniform source in the center of the slab

\[ \Pi(x) \equiv \begin{cases} 
1, & -1 \text{ cm} \leq x \leq 1 \text{ cm} \\
0, & \text{otherwise} 
\end{cases} \]  \hspace{1cm} (H.32)

and

\[ \chi(u) \equiv 0.453e^{-1.036E^*e^{-u}} \text{ sinh} \left( \sqrt{2.29E^*e^{-u}} \right) \]  \hspace{1cm} (H.33)

is an expression for the prompt neutron fission spectrum of \(^{235}\text{U}\) (in MeV) given by [144]. We take our system to be a 10 cm thick slab of a fictitious material with the mass of deuterium \((A = 2)\). Discretizing over \( N = 201 \) nodes in both dimensions, we assume our cross-sections and diffusion coefficient to be

\[ \sigma_t \leftarrow \min\left(10^3e^u/E_0, 1\right), \]  \hspace{1cm} (H.34)

\[ \sigma_s \leftarrow 0.9\sigma_t, \]  \hspace{1cm} (H.35)

\[ D \leftarrow \sigma_s/(3\sigma_t^2). \]  \hspace{1cm} (H.36)

and compute the flux over the lethargy interval \( \mathcal{G} \leftarrow [0, 23.03] \), corresponding to energy \( E \in [10 \text{ MeV}, 1 \text{ meV}] \). Figure H.3 displays the first five modes in each dimension.

In Figure H.4 we estimate the convergence of our numerical problem. Unlike the previous two studies, we have no known solution against which to compare, and so instead measure the relative change between iterations. Specifically, we employ global \( \ell^2 \) and local \( \ell^\infty \) indicators—\( ||\mathcal{X}_m||_{\ell^2} \) and \( ||\mathcal{X}_m||_{\ell^\infty} ||\mathcal{U}_m||_{\ell^\infty} \)—normalized to the contribution of the first mode, \( m = 1 \). Each metric reports monotonic convergence (with agreement between global and local criteria), and is reduced by over seven decades given \( M = 15 \) modes. Moreover, we expect a tolerance on either measure would provide an appropriate termination criterion for
H.4 Conclusions

In summary, we have derived and demonstrated a method for neutron diffusion separated in space and energy via PGD. Using MMS, we have demonstrated $O(h^2)$ numerical accuracy and detailed a test case in which we obtain a near optimal decomposition—compared to that of SVD—before converging upon the discretization error. Lastly, we characterize convergence with enrichment by global and local error indicators for a prototypical fission slowing-down problem. This demonstration provides a proof-of-concept for the more practical applications pursued in Chapters 4 and 5.
Figure H.4: Estimated convergence with modal enrichment for the slowing down of fission neutrons.
APPENDIX I
MODEL ORDER REDUCTION IN ANGLE FOR 1D NEUTRON TRANSPORT

I.1 Introduction

Herein, we present an \textit{a priori} ROM of neutron transport in slab geometry, separated in space and angle using PGD. This method expands on the work of [246] by introducing bidirectional transport and that of [139] by considering collision and scattering operators particular to neutron transport. Doing so, it provides a proof-of-concept for the separation of axial space $z$ (here renamed $x$) and polar angle $\mu$ underpinning the Axial PGD demonstrated in Chapter 6. The expected order of accuracy is verified by MMS while convergence with enrichment is verified using a semi-analytical benchmark by Ganapol [247]. Finally, numerical results over variable scattering ratios are presented for a homogeneous problem with asymmetric, isotropic boundary conditions.

I.2 Methodology

At present, we seek the angular flux $\psi$ in terms of spatial and angular modes, $X_m$ and $U_m$, such that

$$\psi(x, \mu) \approx \sum_{m=1}^{M} X_m(x) U_m(\mu).$$

(I.1)

which is taken to be the solution to the monoenergetic neutron transport equation in slab geometry with isotropic scattering

$$\mu \frac{\partial}{\partial x} \psi(x, \mu) + \Sigma_t(x) \psi(x, \mu) - \frac{1}{2} \Sigma_s(x) \int_{-1}^{+1} \psi(x, \mu') d\mu' = Q(x, \mu)$$

(I.2)

subject to boundary conditions

\[
\psi(x, \mu) = \begin{cases} 
\psi_\ell(\mu), & x = 0, \mu > 0 \\
\psi_r(\mu), & x = L, \mu < 0 
\end{cases}
\] (I.3)

where \(L\) is the length of the slab, and \(\psi_\ell\) and \(\psi_r\) are known functions representing the incident flux on the left and right edges of the slab. Further, the separated source is defined as

\[
Q(x, \mu) \equiv \sum_{i=1}^{I} Q_i(x) \Theta_i(\mu),
\] (I.4)

completing the description of the (seemingly) separable full-order model.

### I.2.1 Bidirectional Separation

However, we now observe that enforcing our boundary conditions would be infeasible over \(\mu \in [-1, +1]\) so long as \(X_m\) is independent of \(\mu\), as there would be no means to distinguish between incoming and outgoing fluxes at the boundaries. As such, we restrict \(\mu\) to \((0, 1]\) and introduce an auxiliary coordinate \(\vartheta\) to represent the direction of particle motion—for brevity, streaming direction—where \(\vartheta \in \{-1, +1\}\), as in [139]. Doing so, the spatial modes can be redefined in terms of forward- and backward-traveling components

\[
X_m(x, \vartheta) \equiv \begin{cases} 
X_m^+(x), & \vartheta = +1 \\
X_m^-(x), & \vartheta = -1 
\end{cases}
\] (I.5)

allowing one to write an alternative but equivalent statement of particle conservation

\[
\vartheta \mu \frac{\partial}{\partial x} \psi(x, \mu, \vartheta) + \Sigma_t(x) \psi(x, \mu, \vartheta) - \frac{1}{2} \Sigma_s(x) \sum_{\vartheta' = \pm 1} \int_0^1 \psi(x, \mu', \vartheta') d\mu' = Q(x, \mu, \vartheta)
\] (I.6)

or, perhaps more expressively,

\[
\pm \mu \frac{\partial}{\partial x} \psi^\pm(x, \mu) + \Sigma_t(x) \psi^\pm(x, \mu) - \frac{1}{2} \Sigma_s(x) \int_0^1 \psi^e(x, \mu') d\mu' = Q^\pm(x, \mu)
\] (I.7)

where

\[
\psi^e(x, \mu) \equiv \psi^+(x, \mu) + \psi^-(x, \mu)
\] (I.8)
on which boundary conditions can be feasibly enforced in the separated model, as visualized in Figure 6.2.

I.2.2 Spatial Submodel

Taking the Galerkin projection of Equation I.7 by integrating in angle against test function $U_M$, as in

$$\int_0^1 \bullet U_M(\mu)d\mu = \left(U_M, \bullet \right)_\mu$$

one finds the spatial submodel

$$\sum_{m=1}^M \left[ \pm \tilde{\mu}_{M,m} \frac{d}{dx} X_{m}^\pm(x) + \tilde{\Sigma}_{t,M,m}(x)X_{m}^\pm(x) - \frac{1}{2} \tilde{\Sigma}_{s,M,m}(x)X_{m}^e(x) \right] = \tilde{Q}_M(x)$$  (I.10)

with coefficients incorporating several angular inner products

$$\tilde{\mu}_{M,m} \equiv \left(U_M(\mu), \mu U_m(\mu) \right)_\mu$$  (I.11)

$$\tilde{\Sigma}_{t,M,m}(x) \equiv \left(U_M(\mu), U_m(\mu) \right)_\mu \Sigma_t(x)$$  (I.12)

$$\tilde{\Sigma}_{s,M,m}(x) \equiv \left(U_M(\mu), \int_0^1 U_m(\mu')d\mu' \right)_\mu \Sigma_s(x)$$  (I.13)

$$\tilde{Q}_M(x) \equiv \sum_{i=1}^I \left(U_M(\mu), \Theta_i(\mu) \right)_\mu Q_i(x)$$  (I.14)

and where $X_{m}^e$ represents the even-parity spatial mode

$$X_{m}^e(x) \equiv X_{m}^+(x) + X_{m}^-(x).$$  (I.15)

As in Progressive PGD generally, modes $m < M$ are assumed to be known (previously computed) from a previous enrichment iteration and so could be subsumed into a residual term $\tilde{R}_M$. Even so, Equation I.10 is still nonlinear, in that it contains products of the unknown modes $X_M$ and $U_M$. Resolving this nonlinearity, as by the fixed-point iteration of Section I.2.4, this submodel becomes a linear equation essentially describing neutron transport with a two-point quadrature at $\pm \mu_{M,m}$ for each mode $X_{m}^\pm$.

Further, this equation is identical to the axial or 1D submodel, Equation 6.55, of the 2D($\mu$)/1D PGD as in Section 6, apart from the convenient assumptions of isotropic scattering and a single energy group. As before, this submodel can be solved for $X_M$ either as a coupled
system of two advection equations—streaming in opposite directions—or recast to a form like the $P_1$ or diffusion equations as in Appendix G.1. Here, the former strategy is selected. As such, any discretization suitable for advective equations would apply (such as DGFEM), though upwind finite differences—that is, backward Euler for $X_M^+$ and forward Euler for $X_M^-$—is chosen here for simplicity. Correspondingly, the scheme is only $O(h)$, or first-order accurate, with respect to mesh width $h$. Additionally, while $X_M^+$ and $X_M^-$ are computed directly (by LU factorization) the coupling between the two is resolved iteratively, effectively amounting to block Jacobi iteration, as interleaved with the nonlinear fixed-point iterations. This loop is detailed by lines 5–10 of Algorithm 8, in which steps 6 and 7 represent the solution of this submodel. Note this is unlike the approach for 1D and 1D($\mu$) submodels in Chapter 6, where the linear solution (by GMRES iteration, with transport sweeps) is converged completely before advancing to the next stage of the nonlinear iteration.

### I.2.3 Angular Submodel

Similarly, taking the weak form by integrating in space $x$ and streaming direction $\vartheta$ against test function $X_M$

$$\sum_{\vartheta=\pm 1} \int_0^L X_M(x, \vartheta) dx = (X_M, \bullet)_{x, \leftrightarrow}$$

(I.16)

yields the angular submodel

$$\left( \hat{\vartheta}_{M,m} + \hat{\Sigma}_{t,M,m} \right) U_m(\mu) - \hat{\Sigma}_{s,M,m} \int_0^1 U_m(\mu') d\mu' = \hat{\Theta}_M(\mu)$$

(I.17)

where the coefficients are now dependent on spatial modes $X_M$ and $X_m$

$$\hat{\vartheta}_{M,m} \equiv \left( X_M(x, \vartheta), \partial_x X_m(x, \vartheta) \right)_{x, \leftrightarrow}$$

(I.18)

$$\hat{\Sigma}_{t,M,m} \equiv \left( X_M(x, \vartheta), \Sigma_t X_m(x, \vartheta) \right)_{x, \leftrightarrow}$$

(I.19)

$$\hat{\Sigma}_{s,M,m} \equiv \left( X_M(x, \vartheta), \Sigma_s X_m^e(x) \right)_{x, \leftrightarrow}$$

(I.20)

$$\hat{\Theta}_M(\mu) \equiv \sum_{i=1}^I \left( X_M(x, \vartheta), Q_i(x, \vartheta) \right)_{x, \leftrightarrow} \Theta_i(\mu)$$

(I.21)
of which the scattering inner product can be simplified

$$\left( X_M(x, \vartheta), \Sigma_s(x)X_m^e(x) \right) \leftrightarrow \left( X_M^e(x), \Sigma_s(s)X_m^e(x) \right)_x.$$ (I.22)

As in the previous section, we grant all modes \( m < M \) are known and linearize by fixed-point iteration. Doing so, Equation I.17 becomes a linear, inhomogeneous Fredholm integral equation of the second kind. To solve this submodel, a collocation method over a trapezoidal quadrature (effectively, an \( S_N \) method) with \( O(h^2) \) accuracy is selected. That said, other quadratures or a Legendre expansion, as in \( P_N \) methods, would be equally applicable.

Further, as suggested by the connection between the preceding spatial and the axial submodels, this angular submodel can be considered a simplification of the radial-polar, or \( 2D(\mu) \), submodel. Namely, the simplification is that of slab geometry (here, \( y \) and \( z \) extending to infinity, with no material interfaces) with a radially-uniform and azimuthally-symmetric flux. In this case, the dependence on radial variables \( \vec{r} \equiv (y, z) \) can be omitted and integration over azimuthal angle \( \omega \) amounts to multiplication by a factor of \( 2\pi \). Under these circumstances, plus the simplifying assumptions of energy-independence and isotropic scattering (granted here for sake of exposition), the two models become equivalent.

### I.2.4 Fixed-Point Iteration

Given these preliminaries, Equations I.10 and I.17 comprise a nonlinear system of equations which can be solved to obtain the last modes \( X_M \) and \( U_M \). As throughout this dissertation, this nonlinearity is resolved by fixed-point (or Picard) iteration, though here with the caveat that the linear (block Jacobi) iterations between \( X_M^+ \) and \( X_M^- \) are interleaved therein. Particularly, this procedure begins by guessing \( U_M^{(0)}, X_M^{+, (0)}, \) and \( X_M^{-, (0)} \), subsequently solving for \( X_M^{+, (1)} \) and \( X_M^{-, (1)} \), and then solving for \( U_M^{(1)} \). This alternating iteration (omitting the initial guess) is repeated until some error indicator, here taken to be a relative \( \ell^2 \) measure

$$\varepsilon^{(k)} \equiv \frac{\left\| X_M^{e,(k)} U_M^{(k)} - X_M^{e,(k-1)} U_M^{(k-1)} \right\|_{\ell^2}}{\left\| X_M^{e,(k-1)} U_M^{(k-1)} \right\|_{\ell^2}}$$ (I.23)
falls below a user-prescribed tolerance, where $k$ denotes the current fixed-point iteration. Note also that $X_M^{+(k)}$ and $X_M^{-(k)}$ are normalized within each fixed-point iteration as

$$X_M^{\pm(k)} \leftarrow X_M^{\pm(k)} / \left( \|X_M^{+(k)}\|_2 + \|X_M^{-(k)}\|_2 \right) \quad \text{(I.24)}$$

where $\|\cdot\|_2$ represents the vector $\ell^2$ norm. This normalization mitigates the “drifting” of modes between iterations, wherein one mode (either $X_M$ or $U_M$) grows while the other shrinks. This iteration, as implemented within the larger procedure of PGD, is described more completely in Algorithm 7, with slight modification of nomenclature.

---

**Algorithm 8: Space-Angle PGD**

1. $\psi^+ \leftarrow X^+_l U_l$  // initialize to satisfy left/right boundary conditions
2. $\psi^- \leftarrow X^-_r U_r$  // modes $\ell/r$ can be omitted as problem allows
3. for $m \leftarrow 1, 2, \ldots M$ do Enrichment Iteration
   4. Initialize $X^{+, (0)}_m, X^{-, (0)}_m,$ and $U^{(0)}_m$ as guesses // here, uniform
   5. for $k \leftarrow 1, 2, \ldots$ while $\varepsilon < \text{tolerance}$ do Fixed-Point Iteration
   6. $X^{+, (k)}_m \leftarrow$ solution of Equation I.10 given $X^{-(k-1)}_m$ and $U^{(k-1)}_m$
   7. $X^{-(k)}_m \leftarrow$ solution of Equation I.10 given $X^{+, (k-1)}_m$ and $U^{(k-1)}_m$
   8. $X^{\pm(k)}_m \leftarrow$ normalize as in Equation I.24
   9. $U^{(k)}_m \leftarrow$ solution of Equation I.17 given $X^{+, (k)}_m$ and $X^{-(k)}_m$
   10. $\varepsilon \leftarrow$ compute by Equation I.23
   11. $\psi^+ \leftarrow \psi^+ + X^{+, (k)}_m U^{(k)}_m$
   12. $\psi^- \leftarrow \psi^- + X^{-(k)}_m U^{(k)}_m$

---

### I.3 Numerical Results

#### I.3.1 Verification by Method of Manufactured Solutions

To begin, we verify the implementation achieves the expected order of accuracy—$O(h)$, or first-order, being limited by the spatial discretization—by MMS. Specifically, let us simulate a problem in a two centimeter slab with a known solution

$$\psi(x, \mu, \vartheta = +1) = \sin(\pi x/2) (1 - \cos(\pi \mu))$$
$$\psi(x, \mu, \vartheta = -1) = \sin(\pi x/2) (1 - \cos(\pi \mu)) \times 0.5. \quad \text{(I.25)}$$
and cross-sections $\Sigma_t = 1$, $\Sigma_s = 0.9$ using $N$ mesh nodes in both $x$ and $\mu$. Measuring the convergence of the method by a relative $\ell^2$ error indicator

$$\ell^2 \text{ Error } = \frac{\|\psi - \psi_{\text{exact}}\|_{\ell^2}}{\|\psi_{\text{exact}}\|_{\ell^2}}$$

and truncating the PGD approximation at $M = 1$ we achieve Figure I.1 which demonstrates the expected $O(h)$ numerical accuracy (where $h$ is inversely proportional to $N - 1$).

Figure I.1: Verification of first-order convergence with grid refinement (b) to the rank-one manufactured solution (a).

### I.3.2 Verification by Analytical Benchmark

Having demonstrated the expected order of accuracy using MMS, we next seek to verify our method with increasing enrichment (that is, number of modes $M$) against an analytical benchmark of prototypical importance to neutron transport. Specifically, we compare against the semi-analytic benchmark of Ganapol [247] which describes a slab impinged on by an incident flux at the left and with a vacuum boundary, $\psi_r(\mu) = 0$, on the right, with isotropic scattering and no loss of energy.

More specifically, we take the incident flux $\psi_\ell(\mu)$ to be a monodirectional unit source normal to the slab (nonzero only for $\mu = 1$) and the slab to be one mean-free-path thick with a scattering ratio $c$ of 0.9 where $c = \Sigma_s/\Sigma_t$. However, as our implementation is discrete, a true normal beam cannot be represented exactly. Instead, we impose a vacuum boundary condition on the left and represent the normal beam as an effective (once-collided) scattering source $Q_1^\pm$, namely

$$Q_1^\pm(x, \mu) \equiv \frac{1}{2} \Sigma_s(x) \int_0^1 \psi_0(x, \mu^\prime) d\mu^\prime$$

where $\psi_0(x, \mu^\prime)$ is the uncollided flux arising from the left boundary, $\psi_0(0, \mu^\prime) = \psi_\ell(\mu^\prime)$. 


Analytically, by the Beer–Lambert law,

$$\psi_0(x, \mu') = \psi_\ell(\mu') \exp \left( -\int_0^s \Sigma_t(s'ds') \right)$$

(I.28)

where $s$ is the path length, $s \equiv x/\mu'$. Substituting $\psi_\ell(\mu') \leftarrow \delta(\mu' - 1)$ then yields

$$Q^{\pm}_1(x, \mu) = Q^{\pm}_1(x) = \frac{1}{2} \Sigma_s(x) \exp \left( -\int_0^x \Sigma_t(x')dx' \right)$$

(I.29)

such that the (once or more) collided flux for an increasing number of modes $M = 5, 10, 15,$ and 20 can be computed as in Figure I.2, as discretized using 300 nodes in space and angle.

Figure I.2: Verification of transmitted (a) and reflected (b) collided fluxes by semi-analytical benchmark.

From these plots we generally observe considerable agreement between the analytical and numerical PGD fluxes for $M \geq 10$. However, the solution near $\mu = 0$ for both the transmitted and reflected fluxes is apparently deficient. One may hypothesize this to be a consequence of the discontinuity at $\psi(0,0)$, as $\lim_{x \to 0^+} \psi(x, \mu) = \psi_\ell(\mu)$ except as $\mu \to 0^+$. Naturally, this jump discontinuity may render the solution difficult to approximate in a separated fashion along $\mu = 0$.

I.3.3 Study Over Scattering Ratio

We next move to simulate a problem lacking a known, analytical solution. For ease of demonstration, we employ isotropic Dirichlet boundary conditions of unit strength on the left edge of the slab and one-half strength on the right.

To illustrate the effects of scattering, we simulate transport through two mean-free-paths of a homogeneous slab with scattering ratios of either $c = 0.5$, $0.9$, or $1.0$ and $\Sigma_t(x) = 1.0$.
cm\(^{-1}\). Further, there is internal source \(Q \leftarrow 0\). Numerically, 20 modes are computed, using 200 nodes in both the spatial and angular grids and a nonlinear tolerance of \(10^{-4}\). To enforce boundary conditions, the first mode is assumed to be

\[
X_1^+ = \begin{cases} 
1, & x = 0, \\
0, & x > 0, 
\end{cases} \quad X_1^- = \begin{cases} 
0.5, & x = 2, \\
0, & x < 2, \end{cases} \quad U_1(\mu) = 1, \tag{I.30}
\]

and all subsequent modes are required to vanish on the boundaries, as is typical for PGD methods.

As nonlinear convergence is not guaranteed, a maximum number of 99 fixed-point iterations is imposed. This limit is intentionally high, as we aim to characterize the convergence of the PGD enrichment without complications of fixed-point error. To this end, the modes \(m\) in which this threshold is reached are marked as stars (\(\star\)) in Figure I.3. In practice, coarse nonlinear tolerances and low iteration limits may be advisable, as any remaining fixed-point error can be corrected by later modes or minimized by “projection” or “update” of the PGD reduced basis.

In measuring the convergence with enrichment (increasing \(M\)), Figure I.3 employs three indicators. Firstly, the relative \(\ell^2\) norm, \(\|U_m\|_{\ell^2}/\|U_2\|_{\ell^2}\) describes the importance of a given angular mode \(U_m\) compared to the first numerically computed mode \(U_2\). (As both spatial modes \(X_m^+\) and \(X_m^-\) are normalized, it is sufficient to measure only the angular modes.) A more physically intuitive metric is provided by the maximum change in angular flux \(\psi\) induced by mode \(m\), \(\max(\Delta\psi_m)\) or \(\|X_m\|_{\infty} \times \|X_m\|_{\infty}\). Likewise, \(\max(\Delta\phi_m)\) or \(\|X_m^+ + X_m^-\|_{\infty} \times \int_0^1 U_m d\mu\) describes the same for the scalar flux. For reference, scalar fluxes are displayed in Figure I.4.

From these plots, one can observe that the relative \(\ell^2\) significance of the last angular modes is approximately \(2 \times 10^{-3}\) or less that of the first computed mode, \(U_2\). The maximum absolute local changes in angular flux, meanwhile, are on the order of \(10^{-2}\) [cm\(^{-2}\)s\(^{-1}\)] or less, while that of the scalar flux are found to be in the vicinity of \(10^{-3}\) [cm\(^{-2}\)s\(^{-1}\)]. Varying the scattering ratio, meanwhile, does not appear to have any severe adverse effect on the PGD approximation. Moreover, despite restricting \(\mu\) to \((0, 1]\) to compute a directionally-independent \(U(\mu)\), the present method appears capable of adequately representing fluxes with dissimilar directional angular distributions—including, as in Figure I.3c, effectively opposite distributions.
Figure I.3: Angular fluxes computed by Proper Generalized Decomposition for scattering ratios $c$ of 0.5, 0.9, and 1.0 (from top to bottom). Stars $(\star)$ denote nonlinear non-convergence.

Figure I.4: Scalar fluxes computed by Proper Generalized Decomposition for varying scattering ratios $c$.

I.4 Conclusions

In conclusion, we have derived and demonstrated a ROM of neutron transport in slab geometry where the spatial and angular dimensions are separated by PGD. Discretization proceeds via upwind finite differences in space and a trapezoidal quadrature rule in angle. $O(h)$ numerical accuracy is verified by MMS while a semi-analytical benchmark is employed
for verification with increasing enrichment. A parametric study over scattering ratio is presented for a homogeneous problem with asymmetric, isotropic boundary conditions. As such, this investigation demonstrates the feasibility of separating axial space and polar angle by halving the angular range and introducing an auxiliary variable $\vartheta$ for the streaming direction. This principle is later taken to its logical conclusion in the 2D($\mu$)/1D ROMs of Chapter 6, which are based on this work.
APPENDIX J
SUPPLEMENTAL FILES

J.1 Permissions for Chapter 4 and Appendices E, H, and I

This file contains license details and terms and conditions for the reproduction used in Chapter 4 and Appendices E, H, and I of this dissertation.

File name: Dominesey2022-ANS-Permissions.pdf

File type: Portable Document Format (PDF)

File size: 127.2 kB

Required application software: Adobe Acrobat or any standard PDF viewer

Special hardware requirements: None