

AXIALLY QUADRATIC DISCONTINUOUS FINITE ELEMENTS FOR NEUTRON
TRANSPORT REACTOR SIMULATIONS

A Thesis

by

ROBERT FRANKLIN TURNER

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

Chair of Committee, Jean C. Ragusa
Committee Members, Marvin L. Adams
Nancy M. Amato
Head of Department, Michael Nastasi

August 2020

Major Subject: Nuclear Engineering

Copyright 2020 Robert Franklin Turner

ABSTRACT

Numerical methods are often used to approximate the solutions to partial differential equations. Each problem may warrant the use of multiple numerical methods to solve for each domain over which the problem is posed. For the form of the neutron transport equation developed in this work, a discontinuous finite element method (DFEM) is utilized to discretize and solve the transport problem over physical space. Furthermore, the level of spatial refinement implemented with the DFEM influences both the accuracy of the solution and the time needed to solve the problem. This work serves to extend the existing Linear Discontinuous (LD) finite element method in Texas A&M University's parallel deterministic transport code PDT to an axially quadratic DFEM deemed Linear Discontinuous Quadratic-in-Z (LDQZ) in the hope of both increasing the accuracy of the solution provided by the existing DFEM and reducing the time needed to arrive at the solution. This increase in performance is investigated on a neutronics slab problem, a criticality slab problem, a 3D C5G7 UO₂ fuel pin, and a 3D quarter assembly from CASL's VERA Core Physics Benchmark. For the simpler test problems modeled, the new method is shown to be more accurate via both axial flux profile comparisons and errors in the L_2 norm as well as faster for prescribed error thresholds. For the two more complex and realistic problems, LDQZ shows no significant improvement over LD when comparing axial flux and power profiles and Δk_{eff} values while also inspecting the times to solution.

ACKNOWLEDGMENTS

I am grateful to have had the opportunity to work with my advisor Dr. Jean Ragusa, whose support and direction aided in my development as both a student and engineer. I also want to thank my committee members Drs. Marvin Adams and Nancy Amato for their investment in and review of this research, as well as Milan Hanus, Daryl Hawkins, Jan Vermaak, and Yunhuang Zhang for their advice and knowledge. Finally, I want to thank my family, whose constant love encouraged me throughout my time spent with this undertaking.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis committee consisting of advisor Professor Jean Ragusa, Professor Marvin Adams of the Department of Nuclear Engineering, and Professor Nancy Amato of the Department of Computer Science and Engineering.

Jan Vermaak of Texas A&M provided a post-processing script used in part to generate figures in Sections 5.4, 5.5, and 5.6.

All other work conducted for the thesis was completed by the student independently.

Funding Sources

This material is based upon work supported under a Department of Energy, Office of Nuclear Energy, Integrated University Program Graduate Fellowship. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the author and do not necessarily reflect the views of the Department of Energy Office of Nuclear Energy.

NOMENCLATURE

LD	Linear Discontinuous
LDQZ	Linear Discontinuous Quadratic in Z
PDT	Parallel Deterministic Transport Code
TAMU	Texas A&M University
PDE	Partial Differential Equation
FEM	Finite Element Method
CFEM	Continuous Finite Element Method
DFEM	Discontinuous Finite Element Method
WRM	Weighted Residual Method
DOF	Degree of Freedom

TABLE OF CONTENTS

	Page
ABSTRACT	ii
ACKNOWLEDGMENTS	iii
CONTRIBUTORS AND FUNDING SOURCES	iv
NOMENCLATURE	v
TABLE OF CONTENTS	vi
LIST OF FIGURES	viii
LIST OF TABLES.....	x
1. INTRODUCTION.....	1
1.1 Objective.....	2
2. FINITE ELEMENT METHOD.....	4
2.0.1 Weighted residual method.....	4
2.0.2 Continuous finite element method	6
2.0.3 Discontinuous finite element method.....	7
3. NEUTRON TRANSPORT EQUATION	10
3.0.1 Solve Overview	12
3.0.2 Inscatter source treatment with spherical harmonics	14
3.0.3 Energy discretization	15
3.0.4 Angle discretization	18
3.0.5 Spatial discretization	20
4. LINEAR DISCONTINUOUS QUADRATIC IN Z.....	23
4.1 LD and LDQZ weak form and basis functions	23
4.2 LDQZ matrices.....	27
4.2.1 Advection term	28
4.2.2 Total interaction term.....	30
4.2.3 Vertical surface loss term	30
4.2.4 Horizontal surface loss term	32
4.2.5 External source term	34
4.2.6 Upwind surface source term	35

5. RESULTS.....	38
5.1 Method of manufactured solutions	38
5.2 Convergence in the L^2 norm.....	42
5.3 Regression tests	43
5.4 Axial 1D monoenergetic neutronics problem.....	45
5.5 Axial 1D 2 group criticality problem.....	49
5.6 3D C5G7 pin cell problem.....	52
5.7 CASL VERA benchmark quarter fuel assembly problem	59
6. CONCLUSIONS	66
REFERENCES	68
APPENDIX A.	71

LIST OF FIGURES

FIGURE	Page
2.1 Example 1D solution with quadratic finite elements. At its left boundary, element B takes the solution from the upwind cell A , where Ω is the direction of solution propagation.	8
3.1 General transport solve procedure [1].	13
4.1 Cell geometries not supported by LDQZ.	23
4.2 Shift in coordinate system for an example vertical cell surface.	31
5.1 Solution and input vectors for $\psi(\mathbf{r}) = c$	40
5.2 Solution and input vectors for $\psi(\mathbf{r}) = z$	41
5.3 Solution and input vectors for $\psi(\mathbf{r}) = z^2$	42
5.4 Convergence rates in the L^2 norm of the error for the axially 1D neutronics problem for the LD and LDQZ methods ¹	43
5.5 Radial slice of the Spiderweb mesh used in second regression test.	44
5.6 LD scalar flux solutions for 10, 20, 30, and 40 cell discretizations of the 1D neutronics problem with a 3000 cell PWLD reference solution.	47
5.7 LDQZ scalar flux solutions for 10, 20, 30, and 40 cell discretizations of the 1D neutronics problem with a 3000 cell PWLD reference solution.	48
5.8 LD thermal flux solutions for 21, 42, 63, and 84 cell discretizations of the 1D criticality problem with a 18900 cell LDQZ reference solution.	50
5.9 LDQZ thermal flux solutions for 21, 42, 63, and 84 cell discretizations of the 1D criticality problem with a 18900 cell LDQZ reference solution.	51
5.10 Axial and radial dimensions of the problem (left) and the radial mesh of the pin cell (right) for the C5G7 benchmark pin cell.	53
5.11 Radial location of the lineouts shown for the 3D C5G7 pin cell problem.	54
5.12 LD axial lineout of the 7th group scalar flux for an 11-layer axial refinement at the fuel edge with a 1980-layer LDQZ reference for the 3D C5G7 problem.	55

5.13	LDQZ axial lineout of the 7th group scalar flux for an 11-layer axial refinement at the fuel edge with a 1980-layer LDQZ reference for the 3D C5G7 problem.....	55
5.14	LD axial lineout of the 7th group scalar flux zoomed in near the water region for 11, 22, 33, and 44 total axial layers with a 1980-layer LDQZ reference for the 3D C5G7 problem.	56
5.15	LDQZ axial lineout of the 7th group scalar flux zoomed in near the water region for 11, 22, 33, and 44 total axial layers with a 1980-layer LDQZ reference for the 3D C5G7 problem.	57
5.16	Δk_{eff} for multiple LD and LDQZ axial refinements against a 1980-layer LDQZ reference for the 3D C5G7 problem.	58
5.17	Radial slice of the CASL Problem 3A quarter assembly through an axial region with a grid spacer.	61
5.18	Axial slice of the CASL Problem 3A quarter assembly showing a region in the active fuel with grid spacers.	62
5.19	49-layer integrated and volume-averaged axial power profiles for LD and LDQZ for the CASL VERA quarter assembly.	63
5.20	29-layer integrated and volume-averaged axial power profiles for LD and LDQZ for the CASL VERA quarter assembly.	64

LIST OF TABLES

TABLE	Page
5.1 Material ordering for the axial 1D neutronics problem.....	46
5.2 Total cross section, absorption cross section, and volumetric source data for the axial 1D neutronics problem.	46
5.3 L^2 error norms and total run times per processor for four axial discretizations of LD and LDQZ simulations for the 1D neutronics problem.	49
5.4 Δk_{eff} , L^2 error norms and total run times per processor for four axial discretizations of LD and LDQZ simulations for the 1D 2-group criticality problem run on 7 processors with an 18900-cell LDQZ reference; $k_{\text{ref}} = 1.015261$	52
5.5 Δk_{eff} and total run times per processor for axial refinements of LD and LDQZ simulations for the 3D C5G7 pin cell run on 32 processors with an 2200-layer LDQZ reference; $k_{\text{ref}} = 1.321264$	59
5.6 Δk_{eff} and total run times per processor for two axial discretizations of LD and LDQZ simulations for the CASL quarter assembly problem with a reference $k_{\text{eff}} = 1.175722$ provided by the benchmark.	65

1. INTRODUCTION

In an effort to ensure the safe operation of current nuclear reactors and to develop future reactor technologies, nuclear processes must be simulated computationally. Often in these simulations, this entails approximating the solution to the neutron transport equation to discern a specific type of neutron concentration in different regions within a reactor core. This is no simple task, however, as the most general form of the transport equation is dependent on seven independent variables: three in physical space, two in neutron angle, one in neutron energy, and one in time. When using deterministic solution methods, the sought-after neutron concentration, deemed the neutron flux in nuclear vernacular, is often approximated by discretizing all seven independent variables and finding the solution for each discrete portion of the problem space. Essentially, this results in a large linear system of equations, where the size of the system is approximately equal to the product of the number of all the discrete portions of each independent variable. This system quickly becomes too large even for powerful computers to solve in a realistic time frame.

The earliest computer codes used to solve the transport equation did so by drastically simplifying aspects of the problem or by neglecting certain independent variables of the transport equation entirely. One early but well-known code, DTF-IV, only simulated 1D slab geometries and neglected time for flux calculations [2]. Another more advanced code TRANZIT handled cylindrical 2D, time dependent problems [3]. Codes continued to increase in complexity and in their capabilities to approach reality, often faster than the improvement in computational resources needed to run them.

Current transport codes still face this issue. Either more subtle assumptions are implemented to hasten these simulations, or the full complexity of the problem is handled at the cost of large computational times. The DeCART code, which can model full 3D reactor cores in parallel and incorporates pin-wise thermal feedback, uses the method of characteristics to solve the transport problem radially but uses the diffusion approximation in the axial dimension [4]. This use of the diffusion approximation removes the angular dependence axially, thereby eliminating the two an-

gular variables for which the solution must be determined. Other codes make no assumptions at all. The MINARET code solves the 3D, multigroup transport equation without diffusion approximations, but some realistic simulations require solution times on the order of days [5].

However, codes that utilize finite element methods to discretize the spatial variable of the transport equation may be studied with somewhat of a juggling act to determine the means for an adequately accurate solution in a realistic time frame. As is detailed extensively in Chapter 2, finite element methods are used for the discretization or meshing of the spatial domain as well as for determining the order of the approximate solution. Both the number of cells in the mesh and the order of the function that comprises the approximate solution may be increased or decreased. Generally, increasing either the mesh refinement or the order of the approximate solution results in a more accurate solution at the expense of a longer time to solution. The opposite is also true. So, a more accurate and timely solution may arise from one combination of mesh and order refinement over another. As Texas A&M's PDT finds itself among these codes that both use finite element methods but do not rely on assumptions when solving the transport equation, this work serves to implement a higher order FEM solution approximation and to study the resulting effects on accuracy and solution time from multiple mesh and order refinement combinations for both simple and complex spatial geometries.

1.1 Objective

A new Linear Discontinuous Quadratic-in-Z (LDQZ) method is proposed with the goal of reducing the time to solution of 3D transport solves without sacrificing accuracy when compared to the Linear Discontinuous (LD) method that currently exists in PDT. The new LDQZ method is built solely for axially extruded meshes, where meshes of this type are simply 2D meshes extruded in the axial dimension. This does not prohibit material heterogeneity in the axial dimension so long as any heterogeneities are confined to separate cells. As is detailed in Chapter 4, the time reduction goal relies on the idea that LDQZ might arrive at an accurate solution with less axial refinement than LD. The use of fewer axial refinements results in fewer total spatial unknowns when evaluating the solution, which decreases the computational resources and time needed to solve the problem

assuming the axial coarsening can be handled accurately by LDQZ. While the order of the FEM solution approximation may be refined radially as well, the radial heterogeneity of most reactor geometries necessitates that the coarsest radial meshes are much more highly refined than along the axial dimension of the geometry. So, refining the order of the solution approximation on the already highly refined radial mesh may be redundant.

This work serves to detail the derivation of this method, implement the necessary changes in PDT, verify its correct implementation through simple test cases, and provide results regarding its efficiency in terms of both accuracy and total simulation run time. The remainder of this thesis is as follows: Chapter 2 develops the mathematical background of continuous and discontinuous finite element methods, Chapter 3 covers the discretization process of the steady-state neutron transport equation, Chapter 4 details the solution process when using the LDQZ method and the resultant matrices that make up the final linear system to be solved, and Chapter 5 compares the efficiency of the LDQZ method with that of the LD method for problems of varying complexity.

2. FINITE ELEMENT METHOD

Partial differential equations (PDEs) mathematically represent many physical processes of interest in engineering disciplines. The solution, or, more often, the approximation of the solution of PDEs for quantities like flows, fluxes, stresses, and others is of prominent importance in these disciplines. Numerous techniques have been developed to determine both analytical and numerical solutions of the quantity of interest; finite difference methods, Laplace and Fourier transforms, and the method of characteristics are a few. Moreover, if a certain technique can only produce an approximate solution to the PDE, the determination and quantification of the error introduced by the approximate solution is needed to assess the reliability of the solution. Finite element methods (FEMs) belong to these numerical techniques, where FEMs can be further cleaved into continuous (CFEM) and discontinuous (DFEM) methods. Because this work is the consequence of the utilization of a DFEM in PDT, the attributes and descriptions of FEMs in general will be the focus of the remainder of this chapter. A general CFEM is detailed first followed by a contrast of DFEMs with CFEMs.

2.0.1 Weighted residual method

At the core of FEMs is their capability to minimize the error produced by approximating the solution to the PDE in question. One facet of FEMs is to approximate a solution to a given PDE by minimizing the residual of the PDE with respect to the particular set of basis functions chosen by the analyst that comprise the approximate solution. In this way, FEMs utilize the weighted residual method (WRM) [6]. Consider some arbitrary PDE with an accompanying boundary condition:

$$\mathbf{A}u(x) = f(x) \quad \forall x \in \mathcal{V}, \quad (2.1)$$

$$u(x) = h(x) \quad \forall x \in \partial\mathcal{V}. \quad (2.2)$$

where \mathbf{A} is some arbitrary linear operator, $u(x)$ is the solution of the PDE, $f(x)$ is the forcing function, and $h(x)$ is the solution of the PDE at the boundary $\partial\mathcal{V}$ of the domain \mathcal{V} over which x is defined. If an approximation $\tilde{u}(x)$ to the solution is substituted into the PDE, the residual is defined as

$$R(x) \equiv f(x) - \mathbf{A}\tilde{u}(x). \quad (2.3)$$

With this definition of the residual, the WRM renders the residual orthogonal to some weight space \mathcal{W} with respect to the inner product

$$\int_x R(x)w_i(x)dx = 0 \quad \forall x \in \mathcal{V}, \quad (2.4)$$

where $i = 1, 2, \dots, N$ and $w_i(x)$ is the i^{th} weighting function in the set \mathcal{W} of size N chosen by the analyst. In other words, the WRM returns the best approximate solution to the PDE with respect to the function space defined by the set of weighting functions.

The approximation $\tilde{u}(x)$ is comprised of a linear combination of scalar unknowns and a set of basis functions of size N . This set of basis functions may either be the same set as the weighting function set or a different set. When the two sets are identical, the WRM is a Galerkin method. This is the approach used for LDQZ, so only the Galerkin method is detailed further. With this understood, the solution approximation is given as

$$\tilde{u}(x) = \sum_{i=1}^N U_i w_i(x), \quad (2.5)$$

where U is the sought after size- N vector of scalar unknowns. The value of N is subject to the features of the FEM and domain discretization chosen, so the conditions under which the value of N is determined are detailed in Sections 2.0.2 and 2.0.3.

As a general procedure when using a Galerkin FEM, the variational or so called weak form of the PDE is built by implementing the weighted residual method. Using the arbitrary PDE from Equation 2.1, the weak form is constructed as such:

Find $\tilde{u}(x) \in \mathcal{W}$ such that

$$\int_x \mathbf{A}\tilde{u}(x)w(x)dx = \int_x f(x)w(x)dx \quad \forall w(x) \in \mathcal{W}, \quad (2.6)$$

where \mathcal{W} symbolizes both the basis space and the weighting space, as the two are identical. If the weak form is represented with the solution approximation and the test function as a column vector and row vector respectively applied to the column vector U , this form can be solved as a system of linear equations. This assembly of the weak form is detailed comprehensively in Section 4.1 for LDQZ.

2.0.2 Continuous finite element method

Finite element methods are most often employed in engineering applications to handle the spatial dependence of the PDE in question. In this regard, an FEM is composed of three facets controlled by the analyst wielding the FEM. First, the spatial domain on which the PDE is defined must be divided into a mesh of finite elements. The chosen shape of the elements in the mesh as well as any local refinements are both highly dependent on the problem geometry. Mesh generation is a focus of research all its own, so it will not be developed here. Interested readers can refer to the freely available computational mesh generation software Triangle [7] and Gmsh [8] for further reading.

After a mesh is rendered that adequately discretizes the problem geometry, the order of the basis function set used to approximate the solution is selected. In the simplest formulations, linear polynomials in each spatial dimension are used. While it would seem that the order of this set of polynomials should be equal to the order of the solution in an attempt to generate the best approximate solution, this is often both impossible and unnecessary. Even so, higher order basis functions generally decrease the error in the FEM solution; the study of implementing these higher order basis functions is often referred to as p refinement [9] [10].

Once the order of the basis set is chosen, the last step in preparing the FEM consists of selecting basis functions that satisfy certain necessary or attractive criteria for the problem. For CFEMs, the

chosen basis functions are often associated with a node on the spatial mesh. These nodes most often are locations within each element where the associated basis function is unity and where all other elemental basis functions are zero. In a more concise representation,

$$w_i(x_j) = \delta_{ij} \quad \forall \quad i, j \in 1, 2, \dots, N, \quad (2.7)$$

where the Dirac delta function δ_{ij} is equal to zero if $i \neq j$ and equal to one if $i = j$. To satisfy this criterion, the chosen basis functions are usually piecewise yet continuous across element boundaries. The placement and amount of nodes within each element are both dependent on the order of these piecewise continuous functions. After the basis functions and their associated degrees of freedom are determined, the variational form of the PDE (Equation 2.6) can be solved for the vector of scalar multiples U with size equal to the number of nodes in the entirety of the spatial domain. Finally, the error of the solution approximation can be quantified typically using L^2 error norms if reference results are available. If the error is unacceptably high, the analyst may choose to implement higher order basis functions (p refinement), to increase the number of cells in the mesh (h refinement [11]), or to perform both (hp refinement [12]).

2.0.3 Discontinuous finite element method

The process for constructing DFEMs is identical to that of CFEMs with a significant exception regarding the solve process. With CFEMs, the solution can be determined in every cell simultaneously as a consequence of continuity across cell boundaries. In contrast, discontinuous basis functions are not required to satisfy continuity with the basis functions in adjacent cells, so multiple nonzero values of these basis functions may exist at cell interfaces. This possibility renders the use of nodes pointless, and the influence on the solution in one cell from adjacent cells that is mandated for physically realistic solutions is seemingly lost.

However, this issue is resolved by simply enforcing inflow and outflow conditions between cells during the DFEM solve process. Unlike with CFEMs, the discontinuous solution must be determined cell by cell via the enforced condition that guarantees an appropriate connection of the

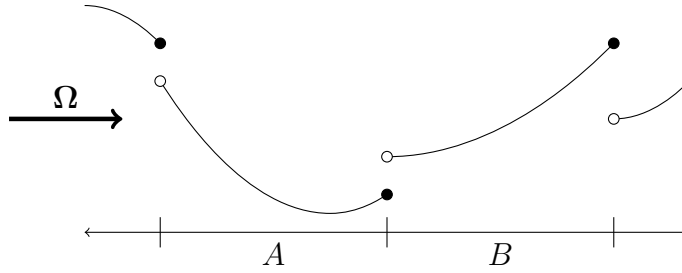


Figure 2.1: Example 1D solution with quadratic finite elements. At its left boundary, element B takes the solution from the upwind cell A , where Ω is the direction of solution propagation.

cells. In PDT, the popular "upwind condition" is utilized [6]. The upwind condition introduces the concept of a directional dependence of the solve process across the domain by communicating the solution in one cell to all adjacent cells in a particular direction. These particular adjacent cells are deemed "downwind" cells, and these cells select the solution communicated from the upwind cell as the solution at the boundary across which the information was passed. The upwind condition is illustrated in Fig. 2.1. A "sweep" across the domain with the cells is then performed, where the solution is determined cell-by-cell across the entire spatial domain. Because the solution to the variational problem is determined in each cell, a linear system is solved for each cell where the size of the cellular system is equal to the size of the set of basis functions. If the PDE has no directional dependence, one sweep is sufficient for determining the final global solution U , where the solution is of size *number of cells* \times *basis set size*.

While the computational implementation of DFEMs can be more involved than that of finite difference or finite volume methods, DFEMs offer attractive features for reactor transport simulations specifically. First, they are very useful when applied to problems with highly irregular meshes with many heterogeneous materials. Meshes of reactor geometries contain adjacent cells with differing cross sections, and a discontinuous solution approximation may better capture certain consequences of these cross sections in the solution than those of a continuous approximation. DFEMs also allow localized hp -refinement for adaptive meshes [13]. This is useful for geometries with certain locations that require more attention than the remainder of the domain. Fine control over both the local mesh size and local order of the solution approximation in these areas of the

mesh also allows optimization of computational resources during the solution process.

3. NEUTRON TRANSPORT EQUATION

The fundamental equation that governs the movement and location of neutrons in the core of nuclear reactors is the neutron transport equation. A few forms of this equation are used in reactor analysis applications, but the integro-differential form is used in this study. This linear partial differential equation is simply a statement that balances the number of neutrons in a phase space. The phase space in the transport context consists of physical space (associated with the vector \mathbf{r} that defines the location of a neutron in the Cartesian frame), the range of possible energies of each neutron in the system (associated with the variable E that defines the energy that a neutron possesses), the direction of flight of a neutron (associated with the unit vector $\boldsymbol{\Omega}$), and the range of times in which the neutrons in the system are investigated (associated with the variable t). However, as this work focuses on solutions to the transport equation that are independent of time, the temporal dependence is neglected from further development of the transport equation. The time-independent or steady-state neutron transport equation reads

$$\begin{aligned} \boldsymbol{\Omega} \cdot \nabla \psi(\mathbf{r}, E, \boldsymbol{\Omega}) + \sigma_t(\mathbf{r}, E) \psi(\mathbf{r}, E, \boldsymbol{\Omega}) \\ = \int_{4\pi} \int_0^\infty \sigma_s(\mathbf{r}, E' \rightarrow E, \boldsymbol{\Omega}' \rightarrow \boldsymbol{\Omega}) \psi(\mathbf{r}, E', \boldsymbol{\Omega}') dE' d\boldsymbol{\Omega}' + Q(\mathbf{r}, E, \boldsymbol{\Omega}). \end{aligned} \quad (3.1)$$

This steady-state equation is used to solve first for the angular flux $\psi(\mathbf{r}, E, \boldsymbol{\Omega})$, which is a function that represents the number of neutrons that exist at the location \mathbf{r} with energy E with direction of flight $\boldsymbol{\Omega}$ per unit area, per unit energy, per unit time. Then, as the directional dependence of the solution is rarely of interest, the angular flux is integrated over the solid angle of the unit sphere surrounding the location \mathbf{r} :

$$\phi(\mathbf{r}, E) = \int_{4\pi} \psi(\mathbf{r}, E, \boldsymbol{\Omega}) d\boldsymbol{\Omega}. \quad (3.2)$$

However, without a given boundary condition that mandates a certain profile of the solution at the boundary of the phase space, this equation may have infinitely many solutions. While many forms

of boundary conditions can enforce the existence of a unique solution $\psi(\mathbf{r}, E, \boldsymbol{\Omega})$, the Dirichlet boundary condition is used here as an example:

$$\psi(\mathbf{r}, E, \boldsymbol{\Omega}) = h(\mathbf{r}, E, \boldsymbol{\Omega}), \quad \forall \boldsymbol{\Omega} \cdot \mathbf{n} < 0, \quad \mathbf{r} \in \partial\mathcal{V}. \quad (3.3)$$

This boundary condition states that the solution $\psi(\mathbf{r}, E, \boldsymbol{\Omega})$ is equal to some function $h(\mathbf{r}, E, \boldsymbol{\Omega})$ on the boundary of the spatial domain $\partial\mathcal{V}$ only for incoming directions $\boldsymbol{\Omega}$ with respect to the unit surface normal \mathbf{n} . In Equation 3.1, we have

- the advection term $\boldsymbol{\Omega} \cdot \nabla\psi(\mathbf{r}, E, \boldsymbol{\Omega})$. This loss term represents the expected net leakage of streaming neutrons per unit volume, energy, and time from the physical domain through the boundary $\partial\mathcal{V}$.
- the total interaction term $\sigma_t(\mathbf{r}, E)\psi(\mathbf{r}, E, \boldsymbol{\Omega})$. This loss term accounts for the expected number of neutrons per unit volume, energy, and time that either change in both energy and streaming direction through scattering interactions with nuclei of the host medium or are absorbed by the host medium nuclei. The total macroscopic cross section $\sigma_t(\mathbf{r}, E)$ is dependent only upon the atomic character of the host medium and the energy that the neutron possesses. The possibly misleading character σ is used here to denote macroscopic cross sections, as the use of the conventional Greek character Σ along with the many summations renders the eventual discrete form of the transport equation somewhat bloated.
- the inscattering term $\int_{4\pi} \int_0^\infty \sigma_s(\mathbf{r}, E' \rightarrow E, \boldsymbol{\Omega}' \rightarrow \boldsymbol{\Omega})\psi(\mathbf{r}, E', \boldsymbol{\Omega}')dE'd\boldsymbol{\Omega}'$. This gain term accounts for the neutrons with an energy and angle existing outside the current phase volume that scatter into the current phase volume per unit volume, energy, and time.
- the external source term $Q(\mathbf{r}, E, \boldsymbol{\Omega})$. This general gain term accounts any other sources of neutrons per unit volume, energy, and time like neutron emitters, fission, and others.

3.0.1 Solve Overview

To solve this equation for the angular flux $\psi(\mathbf{r}, E, \Omega)$ and subsequently the scalar flux $\phi(\mathbf{r}, E)$, PDT utilizes deterministic methods to appropriately discretize the six independent variables that describe neutronic character (three in space, one in energy, and two in angle). Then, generally, a system of N equations is formed to evaluate the solution at all N discrete values of all independent variables. A matrix system succinctly illustrates this system of equations:

$$Ax = b, \tag{3.4}$$

where x is the column vector of the N unknowns, A is an $N \times N$ square matrix of the coefficients applied to the unknowns, and b is the column vector of N forcing terms. For small systems, the matrix A is inverted and applied to b to determine x . However, linear systems associated with real transport problems can contain up to trillions of unknowns, and directly inverting a square matrix this large is computationally prohibitive for many reasons.

Very often in transport calculations, this impasse is evaded by utilizing the source iteration technique. Generally, the source iteration procedure begins with the decomposition of A into two matrices: one computationally inexpensive to invert and sometimes diagonal D and another still difficult to invert J . So, A is decomposed as

$$A = D + J. \tag{3.5}$$

Then, in order to avoid inversion, J is moved to the right hand side of Eq. 3.4 and iteration indices are introduced:

$$Dx^{(i+1)} = b - Jx^{(i)} \quad \text{for } i = 0, 1, 2, \dots \tag{3.6}$$

The iteration process begins with a guess for $x^{(0)}$, and the system is solved for $x^{(i+1)}$. As the final step in each iteration, $x^{(i)}$ and $x^{(i+1)}$ are compared in order to determine if some problem-dependent convergence criteria are met. A typical convergence criterion is met when the maximum

relative difference of all values in $x^{(i)}$ with each corresponding value in $x^{(i+1)}$ is less than some tolerance value. If convergence criteria are not met, the process continues until convergence, where $x^{(i+1)}$ of the converged iterate is taken as the final solution.

In typical computational transport solve procedures and in PDT, the energy, angle, and space variables are discretized into groups, discrete angles, and cells, respectively. Each subset can be envisioned to represent a loop in the solve process. The source iteration occurs within each energy group, but the resultant solution vector is used in the nested angle and cell loops. Figure 3.1 depicts the nested structure of the transport solve, and the specifics of the process are detailed in the following sections.

```
while group to group upscattering is not converged do  
  for each energy group from highest to lowest do  
    while within group scattering is not converged do  
      for each discrete angle in the set in this energy group do  
        for each cell in the domain in the order determined by this angle do  
          Solve the transport equation for this angle and group  
        end for  
      end for  
    end while  
  end for  
end while
```

Figure 3.1: General transport solve procedure [1].

3.0.2 Inscatter source treatment with spherical harmonics

Before any discretizations are performed, the angular dependence of the inscattering cross section is reformed with the Legendre polynomials $P_l(\boldsymbol{\Omega}' \cdot \boldsymbol{\Omega})$ and their associated scattering moments:

$$\sigma_s(\mathbf{r}, E' \rightarrow E, \boldsymbol{\Omega}' \cdot \boldsymbol{\Omega}) = \sum_{l=0}^{\infty} \frac{2l+1}{4\pi} \sigma_l(\mathbf{r}, E' \rightarrow E) P_l(\boldsymbol{\Omega}' \cdot \boldsymbol{\Omega}) \quad (3.7)$$

where $l = 0, 1, 2, \dots$ is the degree of scattering anisotropy, and $\sigma_l(\mathbf{r}, E' \rightarrow E)$ is the Legendre scattering moment. Notice the use of $\boldsymbol{\Omega}' \cdot \boldsymbol{\Omega}$ instead of $\boldsymbol{\Omega}' \rightarrow \boldsymbol{\Omega}$ in the inscattering cross section; the inscattering cross section in actuality only depends on the angle subtended by the incident and resultant directions and not the directions themselves.

While this representation of the cross section function is still exact, the infinite series can be truncated to a finite number of scattering moments L and still adequately account for all scattering anisotropy of the incident neutrons:

$$\sigma_s(\mathbf{r}, E' \rightarrow E, \boldsymbol{\Omega}' \cdot \boldsymbol{\Omega}) \approx \sum_{l=0}^L \frac{2l+1}{4\pi} \sigma_l(\mathbf{r}, E' \rightarrow E) P_l(\boldsymbol{\Omega}' \cdot \boldsymbol{\Omega}). \quad (3.8)$$

The Legendre polynomial can be further expanded into spherical harmonics functions evaluated at the incoming and resultant neutron angles:

$$P_l(\boldsymbol{\Omega}' \cdot \boldsymbol{\Omega}) = \frac{4\pi}{2l+1} \sum_{m=-l}^l Y_{lm}(\boldsymbol{\Omega}') Y_{lm}(\boldsymbol{\Omega}). \quad (3.9)$$

After substitution, the cross section function is then

$$\sigma_s(\mathbf{r}, E' \rightarrow E, \boldsymbol{\Omega}' \cdot \boldsymbol{\Omega}) \approx \sum_{l=0}^L \sigma_l(\mathbf{r}, E' \rightarrow E) \sum_{m=-l}^l Y_{lm}(\boldsymbol{\Omega}') Y_{lm}(\boldsymbol{\Omega}), \quad (3.10)$$

where the scattering moments σ_l are provided by the appropriate nuclear data libraries like ENDF [14]. Although the evaluation of the spherical harmonics equations is not complicated, their definitions are superfluous to this text and can be investigated elsewhere [15] [16].

This splitting of the energy and angle dependence of the inscatter cross section function is performed to allow a better matrix representation of the inscatter term after discretization of the energy and angle variables. Now, the neutron transport equation reads

$$\begin{aligned} & \boldsymbol{\Omega} \cdot \nabla \psi(\mathbf{r}, E, \boldsymbol{\Omega}) + \sigma_t(\mathbf{r}, E)\psi(\mathbf{r}, E, \boldsymbol{\Omega}) \\ &= \sum_{l=0}^L \sum_{m=-l}^l Y_{lm}(\boldsymbol{\Omega}) \int_0^\infty \sigma_l(\mathbf{r}, E' \rightarrow E) \int_{4\pi} Y_{lm}(\boldsymbol{\Omega}') \psi(\mathbf{r}, E', \boldsymbol{\Omega}') d\boldsymbol{\Omega}' dE' + Q(\mathbf{r}, E, \boldsymbol{\Omega}) \end{aligned} \quad (3.11)$$

with the unaltered boundary condition

$$\psi(\mathbf{r}, E, \boldsymbol{\Omega}) = h(\mathbf{r}, E, \boldsymbol{\Omega}), \quad \forall \boldsymbol{\Omega} \cdot \mathbf{n} < 0, \quad \mathbf{r} \in \partial\mathcal{V}. \quad (3.12)$$

While the left and right hand sides are only approximately equal due to the truncation of the scattering anisotropy, a true equality is used as to not draw undue attention to the minor truncation effect on the transport equation in the ensuing development. To further ease the development of the transport solve, the angular integration of the angular flux can be represented with angular flux moments:

$$\phi_{lm}(\mathbf{r}, E) \equiv \int_{4\pi} Y_{lm}(\boldsymbol{\Omega}) \psi(\mathbf{r}, E, \boldsymbol{\Omega}) d\boldsymbol{\Omega}. \quad (3.13)$$

The transport equation becomes

$$\begin{aligned} & \boldsymbol{\Omega} \cdot \nabla \psi(\mathbf{r}, E, \boldsymbol{\Omega}) + \sigma_t(\mathbf{r}, E)\psi(\mathbf{r}, E, \boldsymbol{\Omega}) \\ &= \sum_{l=0}^L \sum_{m=-l}^l Y_{lm}(\boldsymbol{\Omega}) \int_0^\infty \sigma_l(\mathbf{r}, E' \rightarrow E) \phi_{lm}(\mathbf{r}, E') dE' + Q(\mathbf{r}, E, \boldsymbol{\Omega}). \end{aligned} \quad (3.14)$$

At this point, the transport equation is ready to be discretized in energy, angle, and space.

3.0.3 Energy discretization

Conventionally, the energy variable is discretized first when developing the transport solve procedure. Energy discretization in PDT is accomplished by means of the popular multigroup method.

With this method, the energy domain is divided into G energy groups. The degree of refinement of the energy discretization is problem dependent; some simple problem geometries only require a crude 2 group division to correctly return an adequate solution, while more complicated problems may require 200 groups. Applying the multigroup approximation necessitates the definition of the group angular flux:

$$\psi_g(\mathbf{r}, \boldsymbol{\Omega}) \equiv \int_g \psi(\mathbf{r}, E, \boldsymbol{\Omega}) dE. \quad (3.15)$$

This group angular flux for group g is simply the integral of the energy-dependent flux over the energy bounds of group g where $g = 1, 2, \dots, G$. All other terms must also be integrated over energy; for example, the group angular moments are given as

$$\phi_{lm,g}(\mathbf{r}) \equiv \int_g \phi_{lm}(\mathbf{r}, E) dE. \quad (3.16)$$

After performing the necessary integrations, the transport equation can be written as

$$\boldsymbol{\Omega} \cdot \nabla \psi_g(\mathbf{r}, \boldsymbol{\Omega}) + \sigma_{t,g}(\mathbf{r}) \psi_g(\mathbf{r}, \boldsymbol{\Omega}) = \sum_{g'=1}^G \sum_{l=0}^L \sigma_{l,g' \rightarrow g}(\mathbf{r}) \sum_{m=-l}^l Y_{lm}(\boldsymbol{\Omega}) \phi_{lm,g'}(\mathbf{r}) + Q_g(\mathbf{r}, \boldsymbol{\Omega}) \quad (3.17)$$

with the similarly discretized boundary condition

$$\psi_g(\mathbf{r}, \boldsymbol{\Omega}) = h_g(\mathbf{r}, \boldsymbol{\Omega}), \quad \forall \boldsymbol{\Omega} \cdot \mathbf{n} < 0, \quad \mathbf{r} \in \partial\mathcal{V}. \quad (3.18)$$

Now, the cross sections have become flux-averaged over each group, and $\sigma_{l,g' \rightarrow g}(\mathbf{r})$ is the scattering moment for neutrons arriving in group g after a scattering event in group g' . These scattering cross sections and moments are commonly calculated using available nuclear data, but their generation procedure is beyond the scope of this text.

Note here the lack of restriction on g' ; at this step in the development, g' and g may represent the same energy group. As a result, the inscatter term accounts for both the neutrons that scatter into group g from all other groups and the neutrons whose incident and resultant energies from the scattering event are both within the bounds of group g . If these two fractions of the total

in-scattering contribution are separated, the transport equation can be written for a single energy group. Here, the true in-scattering term (in-scattering from groups g' to g where $g' \neq g$) and the external source term Q_g are coalesced into a single total source term $Q_{t,g}$:

$$\boldsymbol{\Omega} \cdot \nabla \psi_g(\mathbf{r}, \boldsymbol{\Omega}) + \sigma_{t,g}(\mathbf{r})\psi_g(\mathbf{r}, \boldsymbol{\Omega}) = \sum_{l=0}^L \sigma_{l,g \rightarrow g}(\mathbf{r}) \sum_{m=-l}^l Y_{lm}(\boldsymbol{\Omega}) \phi_{lm,g}(\mathbf{r}) + Q_{t,g}(\mathbf{r}, \boldsymbol{\Omega}); \quad (3.19)$$

$$\psi_g(\mathbf{r}, \boldsymbol{\Omega}) = h_g(\mathbf{r}, \boldsymbol{\Omega}), \quad \forall \boldsymbol{\Omega} \cdot \mathbf{n} < 0, \quad \mathbf{r} \in \partial\mathcal{V}, \quad (3.20)$$

At this point, the transport equation is split into G equations with $\gg G$ unknowns. These equations are coupled through the total source term $Q_{t,g}$, which implies the energy-discretized flux solutions in every other group are known. Evaluating the flux solution in each group is accomplished computationally by looping through the energy groups from highest to lowest and solving the transport equation in each group with the source iteration technique detailed in Section 3.0.1. This ordering of the groups is intentional; for many physical problems, neutron up-scattering from lower groups to higher groups can be neglected without consequence. Utilizing this approximation can reduce computation time significantly, as the outer iteration needed to determine up-scattering contributions in Figure 3.1 can be neglected. However, if up-scattering is present in the problem, the flux solutions used to evaluate scattering from lower groups to higher groups are not converged solutions via source iterations. Consequently, these lower-to-higher contributions are merely estimates, and the outer up-scattering loop is necessary for convergence in each group.

While source iteration is performed for each energy group, the transport equation solved for each iteration must be discretized twice more in order to achieve a discrete flux solution for the remainder of the phase space (angle and physical space). As this further development of the solve process can be restricted to a single energy group, the subscript g is dropped from the transport equation and the boundary condition.

3.0.4 Angle discretization

Although the angular approximation of the transport equation is commonly performed with any one of a number of methods, the discrete ordinates or S_N approximation is developed here as PDT is built with this approximation [16] [17]. The S_N approximation serves to select a finite number of directions N from the unit sphere along which the angular flux is to be solved. The discrete ordinate angular flux is approximated as

$$\psi(\mathbf{r}, \boldsymbol{\Omega}) \approx \begin{cases} \psi(\mathbf{r}, \boldsymbol{\Omega}_n), & \forall \boldsymbol{\Omega}_n \in \Theta \\ 0, & \text{elsewhere.} \end{cases} \quad (3.21)$$

where $\boldsymbol{\Omega}_n$ is the n^{th} discrete ordinate in the set of discrete ordinates Θ , and $n = 1, 2, \dots, N$. For simplicity, the discrete angular flux is also defined as

$$\psi_n(\mathbf{r}) \equiv \psi(\mathbf{r}, \boldsymbol{\Omega}_n). \quad (3.22)$$

Because the S_N method evaluates the flux along discrete directions, quadrature methods are best suited to evaluate the angular in-scattering source term. Quadrature methods use a linear combination of a finite set of weight values with an equally-sized finite set of points on the domain of interest to approximate function integrations. While selection of these sets of weights and domain points is problem dependent, the angular flux moments can be represented generally as

$$\phi_{lm}(\mathbf{r}) \approx \sum_{n=1}^N w_n Y_{lm}(\boldsymbol{\Omega}_n) \psi_n(\mathbf{r}). \quad (3.23)$$

This approximation may even be exact if the number of weights and points and their values are appropriate for the order of the function they integrate. After implementation of both the discrete angular flux and the quadrature method for the angular flux moments, the transport equation with

its boundary condition for a single discrete direction reads

$$\boldsymbol{\Omega}_n \cdot \nabla \psi_n(\mathbf{r}) + \sigma_t(\mathbf{r})\psi_n(\mathbf{r}) = \sum_{l=0}^L \sigma_l(\mathbf{r}) \sum_{m=-l}^l Y_{lm}(\boldsymbol{\Omega}_n)\phi_{lm}(\mathbf{r}) + Q_{t,n}(\mathbf{r}); \quad (3.24)$$

$$\psi(\mathbf{r}, \boldsymbol{\Omega}_n) = h(\mathbf{r}, \boldsymbol{\Omega}_n), \quad \forall \boldsymbol{\Omega}_n \cdot \mathbf{n} < 0, \quad \mathbf{r} \in \partial\mathcal{V}, \quad (3.25)$$

At this point, the operators and sums are more concisely represented in operator notation:

$$\mathbf{A}\Psi(\mathbf{r}) + \sigma_t(\mathbf{r})I\Psi(\mathbf{r}) = \mathbf{S}\Psi(\mathbf{r}) + q, \quad (3.26)$$

where

- $\Psi(\mathbf{r})$ is the $N \times 1$ vector of discrete angular flux values $\psi_n(\mathbf{r})$,
- \mathbf{A} is the diagonal advection operator $\text{diag}\{\boldsymbol{\Omega}_1 \cdot \nabla, \boldsymbol{\Omega}_2 \cdot \nabla, \dots, \boldsymbol{\Omega}_N \cdot \nabla\}$ applied to $\Psi(\mathbf{r})$,
- $\sigma_t(\mathbf{r})I$ is the diagonal outscattering matrix and I is the $N \times N$ identity matrix,
- \mathbf{S} is the inscattering matrix composed of the matrix product MCD , where

D , the discrete-to-moments matrix, =

$$\begin{bmatrix} w_1 Y_{00}(\boldsymbol{\Omega}_1) & w_2 Y_{00}(\boldsymbol{\Omega}_2) & \dots & w_N Y_{00}(\boldsymbol{\Omega}_N) \\ w_1 \sum_{m=-1}^1 Y_{1m}(\boldsymbol{\Omega}_1) & w_2 \sum_{m=-1}^1 Y_{1m}(\boldsymbol{\Omega}_2) & \dots & w_N \sum_{m=-1}^1 Y_{1m}(\boldsymbol{\Omega}_N) \\ \vdots & \vdots & \ddots & \vdots \\ w_1 \sum_{m=-L}^L Y_{Lm}(\boldsymbol{\Omega}_1) & w_2 \sum_{m=-L}^L Y_{Lm}(\boldsymbol{\Omega}_2) & \dots & w_N \sum_{m=-L}^L Y_{Lm}(\boldsymbol{\Omega}_N) \end{bmatrix},$$

C is the $L \times L$ diagonal matrix of the scattering moments $\sigma_l(\mathbf{r})$,

M , the moments-to-discrete matrix, = $W^{-1}D^T$, where

W is the $N \times N$ diagonal matrix of the weighting values w_n , and

q is the $N \times 1$ column vector of contributions from all energy groups to each discrete direction.

Notice that, with source iteration, the terms on the right hand side are known, while the matrix operators on the left hand side are diagonal. This allows each of the N equations to be solved independently; consequentially, in PDT, the transport equation can be solved along each discrete direction at the same time in parallel if enough processors are available for the solve. As was done with the energy subscript g , the angular subscript n is abandoned from the following short introduction to the spatial discretization of the transport equation along a single discrete direction.

3.0.5 Spatial discretization

If all terms on the right hand side are combined to form a single source term $q(\mathbf{r})$, the transport equation for a single discrete direction within a single energy group is given as

$$\boldsymbol{\Omega} \cdot \nabla \psi(\mathbf{r}) + \sigma_t(\mathbf{r})\psi(\mathbf{r}) = q(\mathbf{r}); \quad (3.27)$$

$$\psi(\mathbf{r}) = h(\mathbf{r}), \quad \forall \mathbf{r} \in \partial\mathcal{V}. \quad (3.28)$$

In PDT, DFEMs are used to both discretize the spatial domain and to solve for the discrete angular fluxes in each cell and for each group. If a general DFEM is applied to the within-angle transport equation, the equation for each cell becomes

$$\int_{\mathcal{K}} (\boldsymbol{\Omega} \cdot \nabla \psi(\mathbf{r}) + \sigma_t(\mathbf{r})\psi(\mathbf{r})) v(\mathbf{r}) d^3\mathbf{r} = \int_{\mathcal{K}} q(\mathbf{r})v(\mathbf{r})d^3\mathbf{r} \quad \forall \mathcal{K} \in \mathcal{V}, \quad (3.29)$$

where \mathcal{K} symbolizes a single cell in the spatial domain \mathcal{V} and $v(\mathbf{r})$ is one of the weight functions applied to the transport equation. The boundary conditions for each cell are determined using the upwind condition mentioned in Section 2.0.3.

While this cursory depiction of the spatial transport solve is detailed comprehensively in the next section, this general DFEM results in a system of equations small enough for solution via direct inversion for each cell in the spatial mesh. The solution itself is a linear combination of the

set of basis functions and an associated vector of scalar multiples; this vector of scalar multiples is the solution of the spatial transport solve in each cell. Once this procedure has been performed for all cells and discrete directions for the energy group of interest, a convergence check is performed against the angular flux moments determined in the previous iterate. If the user-defined tolerance is met, the solve process is repeated for the next lowest energy group. Finally, if upscattering is not present in the problem, the solve is complete after the lowest energy group reaches convergence.

The previous development has detailed the procedure for calculating the angular and scalar flux for a neutronics problem with a general external source. For criticality problems involving fissile material, the same nested loop procedure is followed with the addition of another outer loop that serves to calculate the multiplication factor k of neutrons in the problem domain introduced along with the fission source. This multiplication factor is simply the ratio of the number neutrons produced in one generation to the number of neutrons produced in the previous generation. Criticality problems also assume fission to be the only source of neutrons, so Eq. 3.26 can be rewritten as

$$\mathbf{A}\Psi(\mathbf{r}) + \sigma_t(\mathbf{r})I\Psi(\mathbf{r}) = \frac{1}{k}\mathbf{S}\Psi(\mathbf{r}). \quad (3.30)$$

At this point, an eigenvalue problem results with eigenvalue k and eigenvector $\Psi(\mathbf{r})$. The eigenvector is updated with each inner iteration as denoted previously, but the eigenvalue must be updated at the conclusion of these inner iterations. Many update choices exist, but PDT updates the eigenvalue according directly to the definition of neutron multiplication given above:

$$k^{(n+1)} = \frac{k^{(n)} \int_{\mathcal{V}} \mathbf{S}\Psi^{(n+1)}(\mathbf{r})d^3\mathbf{r}}{\int_{\mathcal{V}} \mathbf{S}\Psi^{(n)}(\mathbf{r})d^3\mathbf{r}}, \quad (3.31)$$

where n is the iteration index and \mathcal{V} denotes the entire spatial domain. The entire iteration procedure concludes when both the eigenvalue and associated eigenvector reach specified convergence tolerances.

It should be noted that while the solve process detailed in this chapter follows the traditional order followed when solving neutronics and criticality problems, PDT offers a means to swap the

order in which the discrete energy, angle, and space variables are solved via groupsets, anglesets, and cellsets, respectively. With these, the user may wish to, for example, solve the problem for every energy group and angle on a certain order of cellsets instead of solving the problem for every cell in the domain at once. While these features allow the user fine control over the solution process, they are beyond the scope of this work and will not be detailed further.

4. LINEAR DISCONTINUOUS QUADRATIC IN Z

This work seeks to extend the fully linear LD spatial discretization method in PDT with a fifth axially quadratic basis function. This entails developing the mass, gradient, and surface matrices for the new LDQZ method and implementing new functions in PDT that use the LDQZ matrices. What follows is the application of the LDQZ method to the inner cell solve for the discrete angular flux with particular focus on the transfer of solution information across cell boundaries in the spatial mesh. First, the weak form of the cellular transport problem is scrutinized, followed by the definition of the LD and LDQZ basis functions.

4.1 LD and LDQZ weak form and basis functions

Before the LDQZ discretization method is detailed, a note on the spatial meshing of the domain in PDT is warranted. Other spatial discretizations, including LD, used by PDT are compatible with cells with face normals in any convex direction with respect to a given face's adjacent face normals. However, the LDQZ method is only developed to function for simple 3D extruded polygons; examples of cells incompatible with LDQZ can be seen in Fig. 4.1. While this limits the capability of the LDQZ method, extruded polygons are widely used in PDT applications. The extension of LDQZ capability for any cell shape supported by PDT may be a developmental focus in the future.

The LDQZ derivation begins with the weak formulation of the transport equation in a single

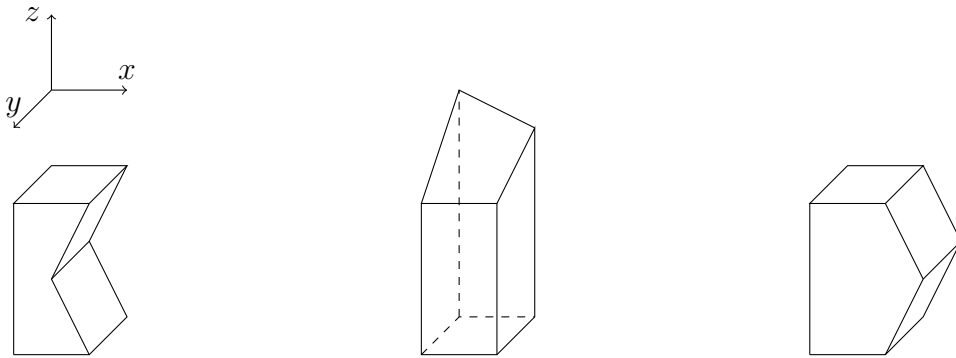


Figure 4.1: Cell geometries not supported by LDQZ.

cell, which is just a slight alteration of Eq. 3.29:

Find $\tilde{\psi}(\mathbf{r}) \in \mathcal{W}$ such that

$$\int_{\mathcal{K}} \left(\boldsymbol{\Omega} \cdot \nabla \tilde{\psi}(\mathbf{r}) + \sigma_i(\mathbf{r}) \tilde{\psi}(\mathbf{r}) \right) \varphi_i(\mathbf{r}) d^3\mathbf{r} = \int_{\mathcal{K}} q(\mathbf{r}) \varphi_i(\mathbf{r}) d^3\mathbf{r}, \quad \forall \varphi_i(\mathbf{r}) \in \mathcal{W}, \quad (4.1)$$

where $\varphi_i(\mathbf{r})$ is a single basis function in the set of basis functions, \mathcal{K} denotes a single cell, and \mathcal{W} is the function space inhabited by the basis functions. Here, the approximate solution is

$$\tilde{\psi}(\mathbf{r}) = \sum_{i=1}^N U_i \varphi_i(\mathbf{r}), \quad (4.2)$$

where where N is the size of the basis function set. This linear combination approximates the spatially varying discrete angular flux over the cell in question. Due to the discontinuity of the solution at cell interfaces, the boundary condition for each cell is determined by the solution in the upwind cell. This upwind solution $\tilde{\psi}_u(\mathbf{r})$ is introduced when the advection term is integrated by parts with help from the Divergence Theorem:

$$\int_{\mathcal{K}} \varphi_i(\mathbf{r}) \boldsymbol{\Omega} \cdot \nabla \tilde{\psi}(\mathbf{r}) d^3\mathbf{r} \approx - \int_{\mathcal{K}} \tilde{\psi}(\mathbf{r}) \boldsymbol{\Omega} \cdot \nabla \varphi_i(\mathbf{r}) d^3\mathbf{r} + \int_{\partial\mathcal{K}} \boldsymbol{\Omega} \cdot \mathbf{n} \varphi_i(\mathbf{r}) \tilde{\psi}_u(\mathbf{r}) d^2\mathbf{r}, \quad (4.3)$$

where \mathbf{n} is the unit normal vector to the surface of the cell $\partial\mathcal{K}$. This upwind flux can take on one of two functions for each surface of the cell. On incoming cell surfaces, or cell faces where $\boldsymbol{\Omega} \cdot \mathbf{n}_f < 0$ if $\boldsymbol{\Omega}$ is the current discrete angle and \mathbf{n}_f is the outward normal of the planar face f , the upwind condition mandates that the solution on the surface be the solution on the surface from the neighboring upwind cell and is denoted with $\tilde{\psi}_s(\mathbf{r})$. On outgoing cell surfaces, or cell faces where $\boldsymbol{\Omega} \cdot \mathbf{n}_f > 0$, the upwind solution is simply the solution from the current cell, so its notation is left untouched. So, Equation 4.3 can be rewritten to show the difference in treatment between

the incoming and outgoing cell surfaces:

$$\int_{\mathcal{K}} \varphi_i(\mathbf{r}) \boldsymbol{\Omega} \cdot \nabla \tilde{\psi}(\mathbf{r}) d^3 \mathbf{r} \approx - \int_{\mathcal{K}} \tilde{\psi}(\mathbf{r}) \boldsymbol{\Omega} \cdot \nabla \varphi_i(\mathbf{r}) d^3 \mathbf{r} + \sum_{f \in \mathcal{F}^-} \int_f \boldsymbol{\Omega} \cdot \mathbf{n} \varphi_i(\mathbf{r}) \tilde{\psi}_s(\mathbf{r}) d^2 \mathbf{r} + \sum_{f \in \mathcal{F}^+} \int_f \boldsymbol{\Omega} \cdot \mathbf{n} \varphi_i(\mathbf{r}) \tilde{\psi}(\mathbf{r}) d^2 \mathbf{r}, \quad (4.4)$$

where f is a single face on the cell and \mathcal{F}^- and \mathcal{F}^+ are the sets of all incoming and outgoing faces on the cell, respectively.

While integration by parts is typically only performed once for similar problems, integration by parts is performed a second time for purposes outside the scope of this work. This time, however, the upwind condition is not enforced on incoming cell surfaces:

$$\int_{\mathcal{K}} \tilde{\psi}(\mathbf{r}) \boldsymbol{\Omega} \cdot \nabla \varphi_i(\mathbf{r}) d^3 \mathbf{r} = - \int_{\mathcal{K}} \varphi_i(\mathbf{r}) \boldsymbol{\Omega} \cdot \nabla \tilde{\psi}(\mathbf{r}) d^3 \mathbf{r} + \sum_{f \in \mathcal{F}^-} \int_f \boldsymbol{\Omega} \cdot \mathbf{n} \varphi_i(\mathbf{r}) \tilde{\psi}(\mathbf{r}) d^2 \mathbf{r} + \sum_{f \in \mathcal{F}^+} \int_f \boldsymbol{\Omega} \cdot \mathbf{n} \varphi_i(\mathbf{r}) \tilde{\psi}(\mathbf{r}) d^2 \mathbf{r}, \quad (4.5)$$

Substituting Equation 4.5 into Equation 4.4 results in

$$\int_{\mathcal{K}} \varphi_i(\mathbf{r}) \boldsymbol{\Omega} \cdot \nabla \tilde{\psi}(\mathbf{r}) d^3 \mathbf{r} \approx \int_{\mathcal{K}} \varphi_i(\mathbf{r}) \boldsymbol{\Omega} \cdot \nabla \tilde{\psi}(\mathbf{r}) d^3 \mathbf{r} + \sum_{f \in \mathcal{F}^-} \int_f \boldsymbol{\Omega} \cdot \mathbf{n} \varphi_i(\mathbf{r}) (\tilde{\psi}_s(\mathbf{r}) - \tilde{\psi}(\mathbf{r})) d^2 \mathbf{r}. \quad (4.6)$$

Notice that only integrals over the incoming cell faces contribute to the surface term introduced through integration by parts of the leakage term. Also notice that the surface term that integrates the product of the upwind surface solution $\tilde{\psi}_s(\mathbf{r})$ and the basis functions is a source term belonging on the right hand side of the equation, while the surface term that integrates the product of the current cell solution $\tilde{\psi}(\mathbf{r})$ and the basis functions is an unknown belonging on the left hand side. Finally, Equation 4.1 can be rewritten as

Find $\tilde{\psi}(\mathbf{r}) \in \mathcal{W}$ such that

$$\begin{aligned} \int_{\mathcal{K}} \left(\boldsymbol{\Omega} \cdot \nabla \tilde{\psi}(\mathbf{r}) + \sigma_t(\mathbf{r}) \tilde{\psi}(\mathbf{r}) \right) \varphi_i(\mathbf{r}) d^3\mathbf{r} - \sum_{f \in \mathcal{F}^-} \boldsymbol{\Omega} \cdot \mathbf{n}_f \int_f \varphi_i(\mathbf{r}) \tilde{\psi}(\mathbf{r}) d^2\mathbf{r} \\ \approx \int_{\mathcal{K}} q(\mathbf{r}) \varphi_i(\mathbf{r}) d^3\mathbf{r} - \sum_{f \in \mathcal{F}^-} \boldsymbol{\Omega} \cdot \mathbf{n}_f \int_f \varphi_i(\mathbf{r}) \tilde{\psi}_s(\mathbf{r}) d^2\mathbf{r}, \quad \forall \varphi_i(\mathbf{r}) \in \mathcal{W}. \end{aligned} \quad (4.7)$$

In the following section, the weighting function $\varphi_i(\mathbf{r})$ and approximate solution $\tilde{\psi}(\mathbf{r})$ are replaced with vectors of the basis functions Φ and the desired vector of unknowns U in order to form a system of equations. To perform the integrations, the LD and LDQZ basis functions must be defined. The basis function set in the LD method is given as

$$\Phi(x, y, z) = \begin{bmatrix} 1 \\ x - \bar{x} \\ y - \bar{y} \\ z - \bar{z} \end{bmatrix},$$

where the bar denotes the respective coordinate of the average of the vertices of the cell. Formally,

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i, \quad (4.8)$$

where N is the number of vertices of the cell, and x_i is the x coordinate of the i^{th} vertex. The same definition holds respectively for the y and z coordinates of the cell average. To form the linear system with solution vector U , the basis function set and weighting function set, which are identical in this development, are multiplied to form a square matrix. Take the mass matrix as an example, where the dependent variable \mathbf{r} has been omitted for brevity:

$$M = \int_{\mathcal{K}} \Phi \Phi^T d^3\mathbf{r}. \quad (4.9)$$

If these basis and weighting function sets are those from the LD method, then the resulting vector

multiplication results in a 4×4 matrix. However, if this mass matrix belongs to the LDQZ method, then the system has size five, and the inner 4×4 matrix elements are identical to the LD matrix regardless of the choice of the fifth LDQZ basis function. This eases the implementation of LDQZ in PDT, as the code used to build the LD matrices can be reused to build the inner 4×4 elements of the LDQZ matrices. This leaves only the selection of a fifth LDQZ basis function, as well as the determination of the fifth row and fifth column of the LDQZ matrices.

As this fifth LDQZ basis function must be multiplied by every other basis function in the set and integrated over the cell volume and each incoming cell surface, a wise selection exploits orthogonality with the other basis functions in these inner products to maximize the number of zero entries in the fifth rows and fifth columns. A modified second order Legendre polynomial fits this bill:

$$\Phi_5(z) = 3 \left(\frac{z - \bar{z}}{\Delta z/2} \right)^2 - 1,$$

where Δz is the axial height of the cell. Not only does this orthogonality simplify LDQZ implementation, computational resources are also spared by the reduction of nonzero terms that must be stored. With the LDQZ basis functions defined, the solution vector can also be defined:

$$U \equiv \left[\bar{u} \quad u_x \quad u_y \quad u_z \quad u_{z^2} \right]^T.$$

The elements here are simply scalar multiples that are applied to the basis functions in a linear combination as detailed in Eq. 4.2. While the last four elements have no physical significance alone, the first element is denoted as \bar{u} as this element represents the flux solution at the average location of the vertices of the cell. This detail is expounded upon at the conclusion of this chapter, but the following section articulates the full linear system in the cell and highlights the many inner products where the fifth basis function is orthogonal to the preceding basis functions.

4.2 LDQZ matrices

The matrix representation of the within-cell transport equation is quite cumbersome, so it will be introduced gradually term by term. Because the LDQZ basis set is identical to the LD basis set

with the exception of the fifth term, the inner 4×4 matrix in the LDQZ matrices are identical to the respective LD matrices. As such, only the elements in the fifth row and fifth column of the LDQZ matrices are evaluated here. A summary of the LDQZ matrices in their matrix forms is provided in Appendix A.

4.2.1 Advection term

When the basis function sets are represented as vectors and substituted into Equation 4.7 appropriately, a linear system $AU = b$ is formed, where A is the sum of the left hand contributions that act on the solution vector U , and b is the column vector representing sources. The advection contribution to A , also known as the gradient contribution, is represented as

$$\left. \begin{array}{l} \int_{\mathcal{K}} \varphi_1(\mathbf{r}) \boldsymbol{\Omega} \cdot \nabla \tilde{\psi}(\mathbf{r}) d^3 \mathbf{r} \\ \vdots \\ \int_{\mathcal{K}} \varphi_5(\mathbf{r}) \boldsymbol{\Omega} \cdot \nabla \tilde{\psi}(\mathbf{r}) d^3 \mathbf{r} \end{array} \right\} \Rightarrow GU, \quad (4.10)$$

where

$$G \equiv \int_{\mathcal{K}} \mu \Phi(\mathbf{r}) \partial_x \Phi^T(\mathbf{r}) + \eta \Phi(\mathbf{r}) \partial_y \Phi^T(\mathbf{r}) + \xi \Phi(\mathbf{r}) \partial_z \Phi^T(\mathbf{r}) d^3 \mathbf{r}. \quad (4.11)$$

Here, μ , η , and ξ are the components $\boldsymbol{\Omega}$, ∂ is the partial derivative with respect to the subscripted dimension, and the independent variable \mathbf{r} is used instead of x, y, z for ease of notation. The first element of the fifth column of G is evaluated as follows:

$$G_{15} = \int_{\mathcal{K}} \mu \Phi_1(\mathbf{r}) \partial_x \Phi_5(\mathbf{r}) + \eta \Phi_1(\mathbf{r}) \partial_y \Phi_5(\mathbf{r}) + \xi \Phi_1(\mathbf{r}) \partial_z \Phi_5(\mathbf{r}) d^3 \mathbf{r} \quad (4.12)$$

$$= \xi \int_x \int_y \int_z (1) \left(\frac{24(z - \bar{z})}{\Delta z^2} \right) dz dy dx \quad (4.13)$$

$$= 0, \quad (4.14)$$

where $\Phi_i(\mathbf{r})$ is the i^{th} basis function in the set of LDQZ basis functions. The illustrations of the evaluation of the remainder of the elements in the fifth column of G are detailed less rigorously:

$$\begin{aligned}
G_{25} &= \xi \int_x \int_y \int_z (x - \bar{x}) \left(\frac{24(z - \bar{z})}{\Delta z^2} \right) dz dy dx = 0, \\
G_{35} &= \xi \int_x \int_y \int_z (y - \bar{y}) \left(\frac{24(z - \bar{z})}{\Delta z^2} \right) dz dy dx = 0, \\
G_{45} &= \xi \int_x \int_y \int_z (z - \bar{z}) \left(\frac{24(z - \bar{z})}{\Delta z^2} \right) dz dy dx = 2\xi V, \\
G_{55} &= \xi \int_x \int_y \int_z \left(3 \left(\frac{z - \bar{z}}{\Delta z/2} \right)^2 - 1 \right) \left(\frac{24(z - \bar{z})}{\Delta z^2} \right) dz dy dx = 0,
\end{aligned}$$

where V is the volume of the cell. The first four elements in the fifth row of G are 0, where again the terms that include derivatives that evaluate to 0 are omitted:

$$\begin{aligned}
G_{51} &= 0, \\
G_{52} &= \mu \int_{\mathcal{K}} \Phi_5(\mathbf{r}) \partial_x \Phi_2(\mathbf{r}) d^3 \mathbf{r} = 0, \\
G_{53} &= \eta \int_{\mathcal{K}} \Phi_5(\mathbf{r}) \partial_y \Phi_3(\mathbf{r}) d^3 \mathbf{r} = 0, \\
G_{54} &= \xi \int_{\mathcal{K}} \Phi_5(\mathbf{r}) \partial_z \Phi_4(\mathbf{r}) d^3 \mathbf{r} = 0.
\end{aligned}$$

In order to make implementation of this matrix in PDT easier, the G_{45} term can be manipulated to utilize a matrix element that exists in the LD gradient matrix. Again, G_{45} is given as

$$G_{45} = \xi \int_{\mathcal{K}} \Phi_4(\mathbf{r}) \partial_z \Phi_5(\mathbf{r}) d^3 \mathbf{r}. \quad (4.15)$$

If G_{14} is given as

$$G_{14} = \xi \int_{\mathcal{K}} \Phi_1(\mathbf{r}) \partial_z \Phi_4(\mathbf{r}) d^3 \mathbf{r} = \xi V, \quad (4.16)$$

then Equation 4.15 can be rewritten as

$$G_{45} = \frac{G_{14}}{\Delta z} \int_z \Phi_4(\mathbf{r}) \partial_z \Phi_5(\mathbf{r}) dz = 2\xi V = 2G_{14}. \quad (4.17)$$

So, in PDT, the gradient portion of A for LDQZ is simply the LD gradient matrix with a row and column of zeros appended with the exception of G_{45} .

4.2.2 Total interaction term

The total interaction term introduces the contribution from the mass matrix M to A :

$$\left. \begin{array}{l} \int_{\mathcal{K}} \sigma_t(\mathbf{r}) \varphi_1(\mathbf{r}) \tilde{\psi}(\mathbf{r}) d^3\mathbf{r} \\ \vdots \\ \int_{\mathcal{K}} \sigma_t(\mathbf{r}) \varphi_5(\mathbf{r}) \tilde{\psi}(\mathbf{r}) d^3\mathbf{r} \end{array} \right\} \Rightarrow \sigma_{t,\mathcal{K}} M U, \quad (4.18)$$

where

$$M \equiv \int_{\mathcal{K}} \Phi(\mathbf{r}) \Phi^T(\mathbf{r}) d^3\mathbf{r} \quad (4.19)$$

and $\sigma_{t,\mathcal{K}}$ is the constant total cross section within the cell. Because the fifth LDQZ basis function is orthogonal to all of the other basis functions when integrating over the volume of the cell, the first four elements in the fifth column of M are 0. This is true for the fifth row as well, as M is symmetric. This leaves the final element in the fifth row and fifth column:

$$M_{55} = \int_{\mathcal{K}} \Phi_5^2(\mathbf{r}) d^3\mathbf{r} = \frac{4}{5} V. \quad (4.20)$$

Using a similar manipulation to that of the gradient matrix element G_{45} , M_{55} can also be written as

$$M_{55} = \frac{4}{5} M_{11}. \quad (4.21)$$

4.2.3 Vertical surface loss term

This surface term K includes the contribution to A from all vertical incoming cell faces. These vertical cell faces are all simple rectangles due to the extruded nature of the cell, which in turn simplifies the surface integrals on these faces. However, the basis functions must be parameterized radially in order to comfortably integrate a 2D plane in a 3D space. This can be done for the x

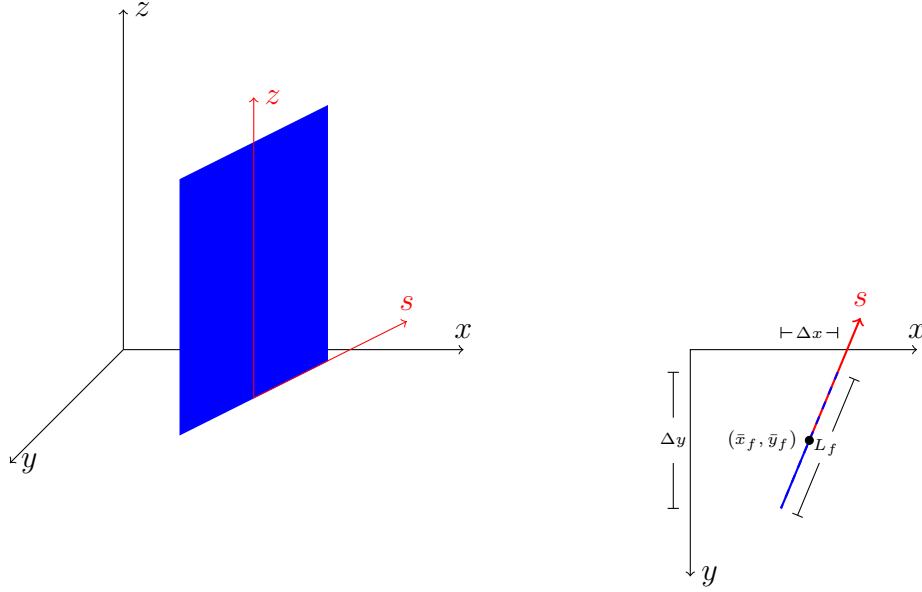


Figure 4.2: Shift in coordinate system for an example vertical cell surface.

variable as

$$x = \bar{x}_f + s \frac{\Delta x}{L_f}, \quad (4.22)$$

where \bar{x}_f is the x coordinate of the center of the vertical face, L_f is the radial length of the face, s is the new independent variable defined from $-L_f/2$ to $L_f/2$ along the radial dimension of the face, and Δx is the change in the x dimension from one radial end of the face to the other. The same parameterization is performed for with the appropriate changes for the y variable.

These substitutions are made in the LDQZ basis functions as

$$\Phi(s, z) = \begin{bmatrix} 1 \\ \bar{x}_f - \bar{x} + s \frac{\Delta x}{L_f} \\ \bar{y}_f - \bar{y} + s \frac{\Delta y}{L_f} \\ z - \bar{z} \\ 3 \left(\frac{z - \bar{z}}{\Delta z/2} \right)^2 - 1 \end{bmatrix}.$$

The contribution to A from these incoming vertical cell faces can now be defined with the parameterized basis function set:

$$- \sum_{f_v \in \mathcal{F}} \left. \begin{array}{c} \Omega \cdot \mathbf{n}_{f_v} \\ \int_{f_v} \varphi_1(\mathbf{r}) \tilde{\psi}(\mathbf{r}) d^2 \mathbf{r} \\ \vdots \\ \int_{f_v} \varphi_5(\mathbf{r}) \tilde{\psi}(\mathbf{r}) d^2 \mathbf{r} \end{array} \right\} \Rightarrow - \left(\sum_{f_v \in \mathcal{F}} \Omega \cdot \mathbf{n}_{f_v} K_{f_v} \right) U, \quad (4.23)$$

where

$$K_{f_v} \equiv \int_{f_v} \Phi(s, z) \Phi^T(s, z) d^2 \mathbf{r}. \quad (4.24)$$

By nature of this term appearing on the left hand side of the weak transport equation, the resultant contribution to A is positive due to the negative evaluation of the dot product $\Omega \cdot \mathbf{n}_{f_v}$ for incoming faces. The first four elements in the fifth column of K_{f_v} are again zero due to the orthogonality of the fifth basis function with the other basis functions. These matrices are also symmetric, so same is true of the fifth row. Only the fifth element in the fifth row and fifth column is nonzero:

$$K_{f_v,55} = \int_{-\frac{L_f}{2}}^{\frac{L_f}{2}} \int_{z_b}^{z_t} \Phi_5^2(s, z) dz ds = \frac{4}{5} A_{f_v}, \quad (4.25)$$

where $K_{f_v,55}$ is the element at the intersection of the fifth row and fifth column of the surface matrix corresponding to the f_v^{th} face, z_t and z_b are the top and bottom coordinates of the cell, respectively, and A_{f_v} is the area of the f_v^{th} face. Again, this element can be rewritten as

$$K_{f_v,55} = \frac{4}{5} K_{f_v,11}. \quad (4.26)$$

4.2.4 Horizontal surface loss term

The final contribution to A on the left hand side occurs from either one of the two horizontal faces of the cell or neither horizontal face if the discrete angle is perpendicular to the horizontal face normals. Unlike the vertical faces, the radial basis functions need not be parameterized, as the horizontal faces exist only in the radial dimensions accounted for in the basis functions. This

horizontal face term can be defined with the matrix H :

$$\left. \begin{array}{c} \int_{f_h} \varphi_1(\mathbf{r})\tilde{\psi}(\mathbf{r})d^2\mathbf{r} \\ -\boldsymbol{\Omega} \cdot \mathbf{n}_{f_h} \quad \quad \quad \vdots \\ \int_{f_h} \varphi_5(\mathbf{r})\tilde{\psi}(\mathbf{r})d^2\mathbf{r} \end{array} \right\} \Rightarrow -\boldsymbol{\Omega} \cdot \mathbf{n}_{f_h} H U, \quad (4.27)$$

where

$$H \equiv \int_{f_h} \Phi(\mathbf{r})\Phi^T(\mathbf{r})d^2\mathbf{r}, \quad (4.28)$$

and where f_h denotes the incoming horizontal cell face. Only one incoming face of the two horizontal faces on the cell can exist, as their normals are antiparallel. Also, the horizontal faces cannot be subdivided into subsets of surfaces, as the PDT meshing process prevents this.

Because the fourth and fifth basis functions are evaluated at either $z = z_b$ or $z = z_t$, they can be removed from the integrand as constants. With this in mind, the first, fourth, and fifth elements in the fifth column of H are:

$$H_{15} = \left(3 \left(\frac{z_{f_h} - \bar{z}}{\Delta z/2} \right)^2 - 1 \right) \int_x \int_y \Phi_1(x, y) dy dx = 2A_{f_h}, \quad (4.29)$$

$$H_{45} = (z_{f_h} - \bar{z}) \left(3 \left(\frac{z_{f_h} - \bar{z}}{\Delta z/2} \right)^2 - 1 \right) \int_x \int_y dy dx = 2(\pm \Delta z) A_{f_h}, \quad (4.30)$$

$$H_{55} = \left(3 \left(\frac{z_{f_h} - \bar{z}}{\Delta z/2} \right)^2 - 1 \right)^2 \int_x \int_y dy dx = 4A_{f_h}, \quad (4.31)$$

where H_{45} is positive if the horizontal face normal of focus has the same orientation as the positive z direction and is negative if the surface normal has the opposite orientation with respect to the positive z direction. Again, because this matrix is symmetric, the fifth row of H is equal to the transpose of the fifth column.

However, the second and third elements of the fifth column have been omitted, as the integrations needed to evaluate these elements are not simple. To skirt this issue, manipulations similar to

those used for the elements in the previous matrices are used. To see this, H_{25} is defined as

$$H_{25} \equiv \int_x \int_y \Phi_2(x, y) \Phi_5(z = \bar{z}_f) dy dx. \quad (4.32)$$

The following is also true:

$$H_{25} = \Phi_5(z = \bar{z}_f) \int_x \int_y \Phi_2(x, y) \Phi_1(x, y) dy dx. \quad (4.33)$$

Since

$$H_{21} \equiv \int_x \int_y \Phi_2(x, y) \Phi_1(x, y) dy dx,$$

Equation 4.33 can be rewritten as

$$H_{25} = \Phi_5(z = \bar{z}_f) H_{21}, \quad (4.34)$$

where H_{21} is known. The same substitution must be made for H_{35} , and can be used for H_{15} , H_{45} , and H_{55} as well:

$$H_{35} = \Phi_5(z = \bar{z}_f) H_{31}; \quad (4.35)$$

$$H_{15} = \Phi_5(z = \bar{z}_f) H_{11}, \quad (4.36)$$

$$H_{45} = \Phi_5(z = \bar{z}_f) H_{41}, \quad (4.37)$$

$$H_{55} = \Phi_5(z = \bar{z}_f) H_{51}. \quad (4.38)$$

Due again to the symmetric nature of this matrix, the fifth row is the transpose of the fifth column.

This concludes the contributions to the left hand side matrix A .

4.2.5 External source term

The external source term includes contributions to the right hand side vector b from inscattering from other discrete angles and energy groups. The mass matrix appears again in this term with the

inscattering sources given in $q(\mathbf{r})$:

$$\left. \begin{array}{c} \int_{\mathcal{K}} q(\mathbf{r})\varphi_1(\mathbf{r})d^3\mathbf{r} \\ \vdots \\ \int_{\mathcal{K}} q(\mathbf{r})\varphi_5(\mathbf{r})d^3\mathbf{r} \end{array} \right\} \Rightarrow MQ, \quad (4.39)$$

where

$$q(\mathbf{r}) \equiv \sum_{i=1}^5 Q_i \varphi_i(\mathbf{r}), \quad (4.40)$$

$$Q \equiv \left[\bar{q} \quad q_x \quad q_y \quad q_z \quad q_{z^2} \right]^T,$$

and each element of Q is determined from the sources of previous iterations in the source iteration process. The elements of the mass matrix are detailed in Section 4.2.2, so they are not detailed again here.

4.2.6 Upwind surface source term

Finally, the last contribution to the total source term on the right hand side of the weak transport formulation is given by the solution from the upwind cell. This upwinding scheme enforces influence on the solution in one cell from the solution in the upwind cell, which is necessary for a physically realistic solution. To minimize the amount of information communicated between cells, the solution from the upwind cell is converted to a form dependent only on the center of the shared face. As both cells share this face center, the solution at the upwind cell average need not be passed to the current cell. Due to this, the surface matrices K_{f_v} and H built for the current cell in Sections 4.2.3 and 4.2.4 may be reused:

$$\sum_{f \in \mathcal{F}^-} \left. \begin{array}{c} \int_f \varphi_1(\mathbf{r})\tilde{\psi}_u(\mathbf{r})d^2\mathbf{r} \\ \vdots \\ \int_f \varphi_5(\mathbf{r})\tilde{\psi}_u(\mathbf{r})d^2\mathbf{r} \end{array} \right\} \Rightarrow -\boldsymbol{\Omega} \cdot \mathbf{n}_{f_h} H \Psi_{f_h} - \sum_{f_v \in \mathcal{F}^-} \boldsymbol{\Omega} \cdot \mathbf{n}_{f_v} K_{f_v} \Psi_{f_v}. \quad (4.41)$$

Here, Ψ_{f_h} and Ψ_{f_v} are the upwind cell solutions on the horizontal face and the vertical faces, respectively:

$$\Psi = \begin{bmatrix} \bar{\psi}_f & \psi_x & \psi_y & \psi_z & \psi_{z^2} \end{bmatrix},$$

where $\bar{\psi}_f$ is the known upwind discrete angular flux at the face center.

Now that all terms have been developed, the weak formulation can be written with vector notation and solved for U . If S represents the left-hand surface matrices and F represents the right-hand surface matrices, the weak form is recast as

$$(G + \sigma_{t,\mathcal{K}}M + S)U = MQ + F. \quad (4.42)$$

To review, the left hand side representing neutron loss from the cell is composed of

- the advection matrix $G \equiv \int_{\mathcal{K}} \mu \Phi(\mathbf{r}) \partial_x \Phi^T(\mathbf{r}) + \eta \Phi(\mathbf{r}) \partial_y \Phi^T(\mathbf{r}) + \xi \Phi(\mathbf{r}) \partial_z \Phi^T(\mathbf{r}) d^3 \mathbf{r}$,
- the total interaction matrix $\sigma_{t,\mathcal{K}}M \equiv \int_{\mathcal{K}} \sigma_t(\mathbf{r}) \Phi(\mathbf{r}) \Phi^T(\mathbf{r}) d^3 \mathbf{r}$,
- the surface matrix $S \equiv -\Omega \cdot \mathbf{n}_{f_h} H - \sum_{f_v \in \mathcal{F}^-} \Omega \cdot \mathbf{n}_{f_v} K_{f_v}$, which is itself composed of
the horizontal surface matrix $H \equiv \int_{f_h} \Phi(\mathbf{r}) \Phi^T(\mathbf{r}) d^2 \mathbf{r}$
and the vertical surface matrices $K_{f_v} \equiv \int_{f_v} \Phi(s, z) \Phi^T(s, z) d^2 \mathbf{r}$,
- and the unknown cellular solution vector $U \equiv \begin{bmatrix} \bar{u} & u_x & u_y & u_z & u_{z^2} \end{bmatrix}^T$.

The right hand side that represents neutron gain in the cell is composed of

- the external source matrix MQ , where
the mass matrix M is given above in the total interaction matrix
and the external source vector is $Q \equiv \begin{bmatrix} \bar{q} & q_x & q_y & q_z & q_{z^2} \end{bmatrix}^T$,
- the upwind surface source matrix $F \equiv -\Omega \cdot \mathbf{n}_{f_h} H \Psi_{f_h} - \sum_{f_v \in \mathcal{F}^-} \Omega \cdot \mathbf{n}_{f_v} K_{f_v} \Psi_{f_v}$, where

the known upwind surface solution vector on a face is $\Psi_f \equiv \left[\bar{\psi}_f \quad \psi_x \quad \psi_y \quad \psi_z \quad \psi_{z^2} \right]^T$.

This assembly and solution process is then repeated for all cells in the domain in an order determined by the appropriate discrete neutron direction. After the spatial domain has been swept in every discrete direction, the first element \bar{u} in the solution set for each cell and each direction is converted to angular flux moments in each cell by means of the discrete-to-moments operator defined in Section 3.0.4. For problems with isotropic scattering, the lone angular flux moment built with the first element in the cellular solution vector for all discrete angles is the scalar flux at the cell average. Because of this, only the angular flux moments built with the first element are used when testing for convergence during the source iteration process. When convergence is reached, PDT returns all angular flux moments in each cell and for each energy group.

5. RESULTS

This chapter serves to investigate the profitability of the LDQZ method for both simple test cases and real complex reactor problems. First, evidence of the verification of the LDQZ method is presented through manufactured solutions, test cases included in the PDT regression test suite, and the return of expected convergence rates in the L^2 norm. Next, two simple axial 1D problems are modeled with both LD and LDQZ to showcase the higher accuracy obtained with LDQZ for problems with axial heterogeneity. Finally, problems from the C5G7 [18] and CASL VERA [19] benchmarks are modeled in order to compare LD to LDQZ for more complex and realistic simulations.

5.1 Method of manufactured solutions

To ensure the correct implementation of the LDQZ method in PDT, an accompanying Matlab code was written to solve the transport equation within a single right rectangular cell with both the LD and LDQZ methods. The LD solutions between Matlab and PDT were first compared to garner trust in the Matlab code, which was then used to check the manufactured LDQZ solutions in PDT. Three manufactured solutions are used to verify LDQZ in PDT: $\psi(\mathbf{r}) = c$, $\psi(\mathbf{r}) = z$, and $\psi(\mathbf{r}) = z^2$, where c is a constant and z is the axial independent variable. The vectors below correspond to those used in Equation 4.42, where the solution vector U is the expected vector of scalars resulting from the manufactured solution $\psi(\mathbf{r})$ and the inputs Ω , Q , and Ψ . This simplified solve is performed for the single direction given, and the upwind solution Ψ is applied to each upwind face.

To determine the solution vector resulting from the selection of the constant solution $\psi(\mathbf{r}) = c$ for a transport problem with one direction, energy group, and spatial cell, the constant solution is simply set equal to the FEM approximate solution and solved for the unknowns:

$$\psi(\mathbf{r}) = c = \bar{u}(1) + u_x(x - \bar{x}) + u_y(y - \bar{y}) + u_z(z - \bar{z}) + u_{z^2} \left(3 \left(\frac{z - \bar{z}}{\Delta z/2} \right)^2 - 1 \right). \quad (5.1)$$

Every unknown but the first must be zero to return the constant solution, where the first unknown is simply the constant value:

$$U = \begin{bmatrix} c & 0 & 0 & 0 & 0 \end{bmatrix}^T.$$

Any incoming direction can be chosen due to the constant solution within the cell, so the direction $\Omega = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$ before normalization is chosen. This direction indicates that the three adjacent faces about the corner of the cell at the origin are incoming faces, and the solution on each incoming face is equivalent to the FEM solution:

$$\Psi_f = \begin{bmatrix} c & 0 & 0 & 0 & 0 \end{bmatrix}^T \quad \forall f \in \mathcal{F}^-.$$

Finally, the volumetric source vector Q must be determined for the interior of the cell. First, the simplified transport equation that these manufactured solutions solve is introduced:

$$\Omega \cdot \nabla \psi(\mathbf{r}) + \sigma_t \psi(\mathbf{r}) = q(\mathbf{r}). \quad (5.2)$$

Then, if $q(\mathbf{r})$ is treated as a linear combination like the FEM solution, the unknowns of the linear combination can be determined if the chosen analytical solution for the flux is substituted:

$$\Omega \cdot \nabla c + \sigma_t c = \bar{q}(1) + q_x(x - \bar{x}) + q_y(y - \bar{y}) + q_z(z - \bar{z}) + q_{z^2} \left(3 \left(\frac{z - \bar{z}}{\Delta z / 2} \right)^2 - 1 \right), \quad (5.3)$$

$$Q = \begin{bmatrix} \sigma_t c & 0 & 0 & 0 & 0 \end{bmatrix}^T.$$

By solving Equation 4.42 for U with the vectors listed in Figure 5.1, the Matlab solve and the PDT solve with LDQZ both return the correct solution U .

For the next test case, the solution $\psi(\mathbf{r}) = z$ is chosen:

$$\psi(\mathbf{r}) = z = \bar{u}(1) + u_x(x - \bar{x}) + u_y(y - \bar{y}) + u_z(z - \bar{z}) + u_{z^2} \left(3 \left(\frac{z - \bar{z}}{\Delta z / 2} \right)^2 - 1 \right). \quad (5.4)$$

$$\begin{aligned}
\Omega &= [1 \ 1 \ 1] \\
Q &= [\sigma_t c \ 0 \ 0 \ 0 \ 0]^T \\
\Psi_f &= [c \ 0 \ 0 \ 0 \ 0]^T \\
U &= [c \ 0 \ 0 \ 0 \ 0]^T
\end{aligned}$$

Figure 5.1: Solution and input vectors for $\psi(\mathbf{r}) = c$.

To satisfy Eq. 5.4, \bar{u} and u_z must be unity and nonzero, respectively, with all others equal to zero:

$$\bar{u} = \bar{z}. \quad (5.5)$$

The resultant solution vector is

$$U = \begin{bmatrix} \bar{z} & 0 & 0 & 1 & 0 \end{bmatrix}^T.$$

To simplify the manufactured solutions process, an incoming direction is chosen such that only one face on the cell is deemed incoming:

$$\Omega = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}.$$

To find the solution on the only incoming face, where the incoming face has normal $\begin{bmatrix} 0 & 0 & -1 \end{bmatrix}$, the location of the face is substituted into the chosen solution and set equal to the solution from the incoming cell:

$$\bar{z} - \frac{\Delta z}{2} = \Phi^T \Psi, \quad (5.6)$$

where Φ is the vector of LDQZ basis functions. Only the first unknown is nonzero:

$$\bar{\psi}_f = \bar{z} - \frac{\Delta z}{2}. \quad (5.7)$$

For the incoming face, the final solution vector is

$$\Psi_f = \left[\bar{z} - \frac{\Delta z}{2} \quad 0 \quad 0 \quad 0 \quad 0 \right]^T \quad \forall f \in \mathcal{F}^-.$$

Next, the solution is substituted into the transport equation to determine the elements in the source vector Q :

$$\mathbf{\Omega} \cdot \nabla \psi(\mathbf{r}) + \sigma_t \psi(\mathbf{r}) = \Phi^T Q. \quad (5.8)$$

Again, all unknowns but the first and fourth must be zero:

$$\xi + \sigma_t z = \bar{q} + q_z(z - \bar{z}). \quad (5.9)$$

If $q_z = \sigma_t$, the final unknown is determined:

$$\bar{q} = \xi + \sigma_t \bar{z}.$$

The full source vector is

$$Q = \left[\xi + \sigma_t \bar{z} \quad 0 \quad 0 \quad \sigma_t \quad 0 \right]^T.$$

The Matlab code and PDT again agree for this test case and return the solution vector in Figure 5.2.

$$\begin{aligned} \mathbf{\Omega} &= [0 \quad 0 \quad 1] \\ Q &= [\xi + \sigma_t \bar{z} \quad 0 \quad 0 \quad \sigma_t \quad 0]^T \\ \Psi_f &= [\bar{z} - \frac{\Delta z}{2} \quad 0 \quad 0 \quad 0 \quad 0]^T \\ U &= [\bar{z} \quad 0 \quad 0 \quad 1 \quad 0]^T \end{aligned}$$

Figure 5.2: Solution and input vectors for $\psi(\mathbf{r}) = z$.

Because the algebra is a bit more cumbersome for the third test case with $\psi(\mathbf{r}) = z^2$, the development is omitted. However, this test case is summarized in Figure 5.3, where both Matlab and PDT return identical solution vectors.

$$\begin{aligned}
\Omega &= [0 \ 0 \ 1] \\
Q &= [2(\xi + \sigma_t \bar{z}) \bar{z} - \sigma_t \bar{z}^2 + \sigma_t (\Delta z)^2 / 12 \quad 0 \quad 0 \quad 2(\xi + \sigma_t \bar{z}) \quad \sigma_t (\Delta z)^2 / 12]^T \\
\Psi_f &= \left[\left(\bar{z} - \frac{\Delta z}{2} \right)^2 \quad 0 \quad 0 \quad 0 \quad 0 \right]^T \\
U &= [(\Delta z)^2 / 12 + \bar{z}^2 \quad 0 \quad 0 \quad 2\bar{z} \quad (\Delta z)^2 / 12]^T
\end{aligned}$$

Figure 5.3: Solution and input vectors for $\psi(\mathbf{r}) = z^2$.

5.2 Convergence in the L^2 norm

This convergence study utilizes a test case that mimics a 1D monoenergetic neutronics problem and uses as the reference solution a highly refined LDQZ solve. While merely an overview of the problem is given for this convergence study, a full description of the parameters used here is detailed in Section 5.4 when demonstrating the superior fit of the LDQZ solution to the reference solution. This test case mimics a 1D problem in that the geometry is composed of a single $1\text{E}6 \times 1\text{E}6 \text{ cm}^2$ radial cell with opposing radial reflective boundary conditions and is extruded in the axial dimension 300 cm with ten materially heterogeneous divisions. So, the material composition of the domain only varies in the axial dimension. The two axial faces of this stack of cells are given vacuum boundary conditions, and the geometry of the refined LDQZ solution contains 2187 cells per material region. If this refined LDQZ solution is taken to be the reference scalar flux solution $\phi_{\text{ref}}(z)$ with the trial solution $\tilde{\phi}(z)$, the L^2 norm of the error is expressed as

$$\|\tilde{\phi}(z) - \phi_{\text{ref}}(z)\|_{L^2} \equiv \sqrt{\int_{z_b}^{z_t} \left(\tilde{\phi}(z) - \phi_{\text{ref}}(z) \right)^2 dz}, \quad (5.10)$$

where z_b and z_t are the axial locations of the bottom and top of the domain, respectively. The L^2 norms for six axial discretizations are plotted in Figure 5.4 for both the LD and LDQZ methods.

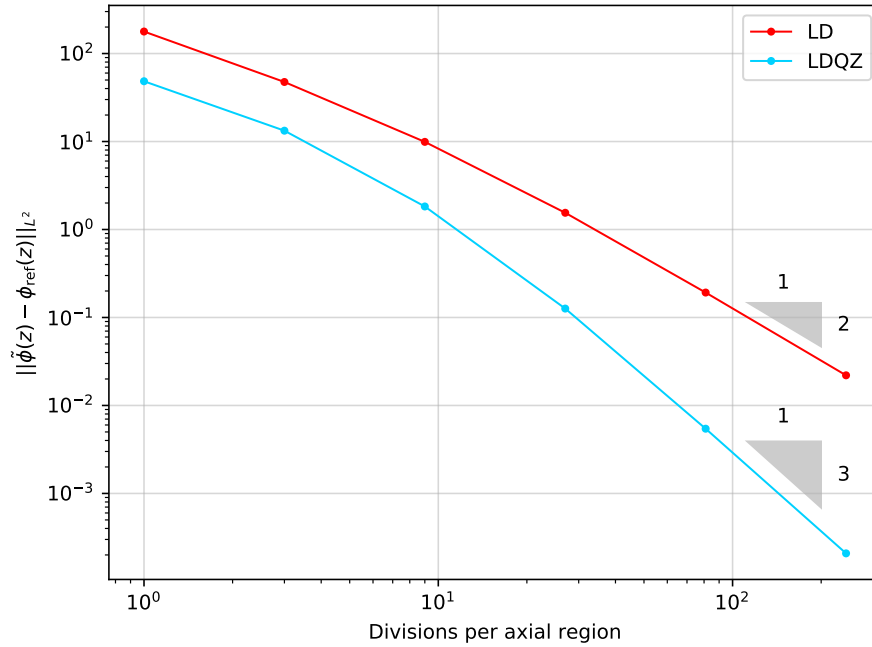


Figure 5.4: Convergence rates in the L^2 norm of the error for the axially 1D neutronics problem for the LD and LDQZ methods¹.

It has been proven that for discontinuous Galerkin methods that utilize spatial domains made of cells with unique outflow faces and basis functions composed of polynomials of degree k , convergence rates of $k + 1$ may be achieved [20]. Because the LDQZ basis function set contains an axially quadratic polynomial, the method returns third order convergence for this 1D problem in the axial dimension, as seen in Figure 5.4.

5.3 Regression tests

In order to insure source code against damage or alteration by new modifications to the code, PDT houses a regression test suite of input files and their generated output files. To protect the new

¹Credit Tony S. Yu and his `matplotlib` Matplotlib toolkit for the slope markers seen in Figure 5.4.

LDQZ method specifically, four new regression tests are included in this suite. The first regression test aims to produce the constant solution within a single unit cube cell for a single energy group. This cube is filled with a purely absorbing material with a total macroscopic cross section of 0.2 cm^{-1} and a volumetric source of $3 \text{ cm}^{-3} \text{ s}^{-1}$. With these parameters, the constant solution is

$$\phi(\mathbf{r}) = \frac{q(\mathbf{r})}{\sigma_t(\mathbf{r})} = 15 \text{ cm}^{-2} \text{ s}^{-1}.$$

With an incident isotropic source of $15 \text{ cm}^{-2} \text{ s}^{-1}$ on every face of the cube, the constant solution of $15 \text{ cm}^{-2} \text{ s}^{-1}$ is returned with LDQZ.

The second regression test verifies LDQZ for multicell geometries where the vertex averages of the cells in the mesh are not guaranteed to be equal to the cellular centroids. This test utilizes PDT's Spiderweb meshing routine to generate a $1 \times 1 \times 100 \text{ cm}^3$ dummy pin cell with a single material, and a 2D view of the mesh used can be seen in Figure 5.5.

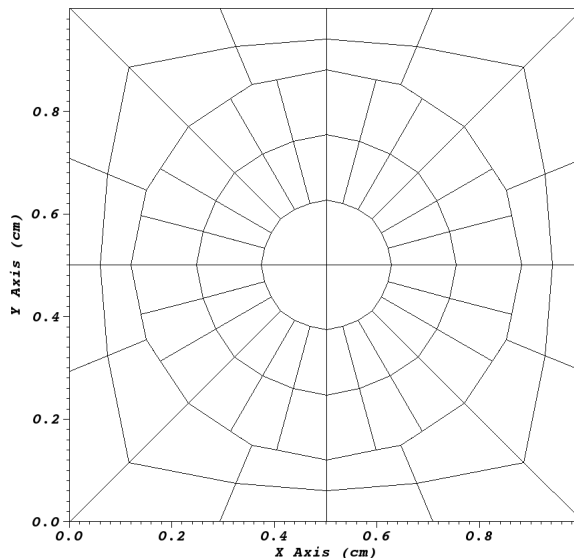


Figure 5.5: Radial slice of the Spiderweb mesh used in second regression test.

This problem uses the same material and source specifications as the first regression test and similarly returns the correct constant solution of $15 \text{ cm}^{-2} \text{ s}^{-1}$ with the LDQZ method for every cell in the domain.

The third input is a modified copy of an LD test that is included to test LDQZ for a multigroup, multicell problem for both serial and parallel runs. This test is a 138 group, 64 cell high-density polyethylene infinite cube neutronics problem with a spatially varying source. Because the convergence rate of the L^2 norm of the error for LDQZ for the axially 1D problem is correct for quadratic DFEMs and the verified LD solutions to this test case and new LDQZ solutions for each cell and energy group are identical out to the first seven digits, the solutions in the output file generated by the LDQZ solve are considered correct.

The fourth and final regression test serves to test LDQZ for a multigroup criticality problem. This test case models an 8-cell infinite cube of Pu-239 with two discrete energy groups and with cross sections given in a Los Alamos benchmark to verify criticality code [21]. A second linear DFEM housed in PDT, the Piece-Wise Linear Discontinuous (PWLD) method, is used this time as a reference solution. The scalar flux results between LDQZ and PWLD are this time identical to at least eight digits for each energy and cell, and the relative difference between the calculated k -eigenvalues is $3.93\text{E-}10$. So, the LDQZ solution is considered correct, and the output file is included in the regression suite.

5.4 Axial 1D monoenergetic neutronics problem

As a first attempt to visualize the increase in accuracy offered by LDQZ over LD without a sacrifice in time, a simple monoenergetic problem with axial heterogeneity is posed that mimics a 1D axial domain. As mentioned in Section 5.2, this problem is defined on a $1\text{E}6 \times 1\text{E}6 \times 300 \text{ cm}^3$ spatial domain with 10 axial material regions. The regions and their compositions are given in Tables 5.1 and 5.2, respectively. By applying reflective conditions to the radial boundaries and vacuum conditions to the axial boundaries, the solution is made to be dependent only on the axial location in the spatial domain. Figures 5.6 and 5.7 are the scalar flux solutions when solving this monoenergetic neutronics problem with Product Gauss Legendre Chebyshev quadrature with four

polar angles and one azimuthal angle per octant with the LD and LDQZ methods, respectively.

Table 5.1: Material ordering for the axial 1D neutronics problem.

Region	1	2	3	4	5	6	7	8	9	10
Material	A	C	C	B	C	D	B	C	D	E

Table 5.2: Total cross section, absorption cross section, and volumetric source data for the axial 1D neutronics problem.

Material	σ_t (cm ⁻¹)	σ_a (cm ⁻¹)	S (cm ⁻³ s ⁻¹)
A	0.167	0.020	0
B	0.303	0.070	10
C	0.278	0.065	3
D	0.222	0.085	13
E	0.167	0.015	0

With a single axial discretization per material region, the top left plot of Figure 5.6 shows the failure of LD in capturing the flux peaks and valleys present in alternating regions of neutron sources. At the same axial refinement, however, the top left plot of Figure 5.7 shows the capability of LDQZ to capture these flux peaks and dips. While LDQZ performs well in approximating the flux profile for this problem with just a single axial refinement per region, the top right plot in Figure 5.7 shows the consequence of using two cells in regions where the flux profile is approximately quadratic. Here, either two slight false peaks and a slight false valley or vice versa are plotted near the true peak or valley. On the other hand, these errors are minor and disappear with further refinement. The total error in the L^2 norm to which this effect contributes can be seen in Table 5.3 along with the associated total run times.

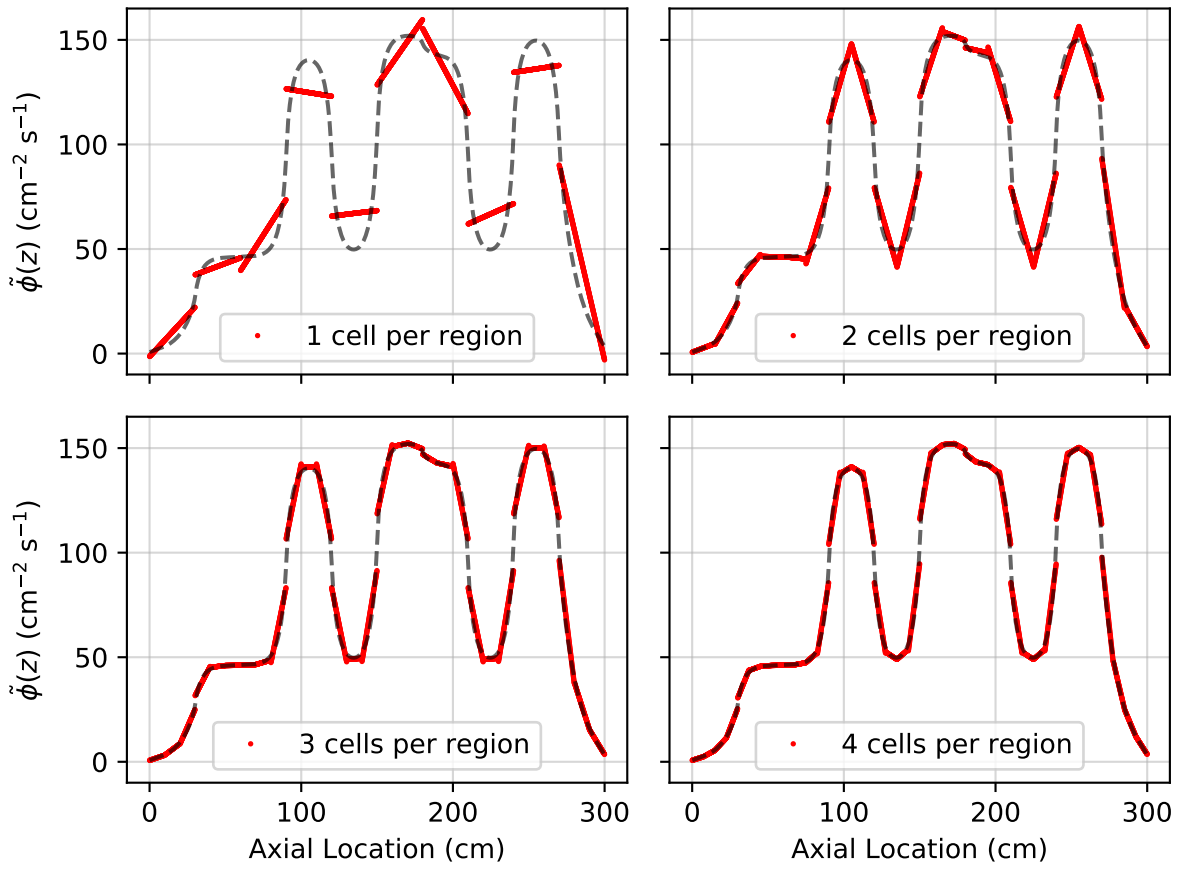


Figure 5.6: LD scalar flux solutions for 10, 20, 30, and 40 cell discretizations of the 1D neutronics problem with a 3000 cell PWLD reference solution.

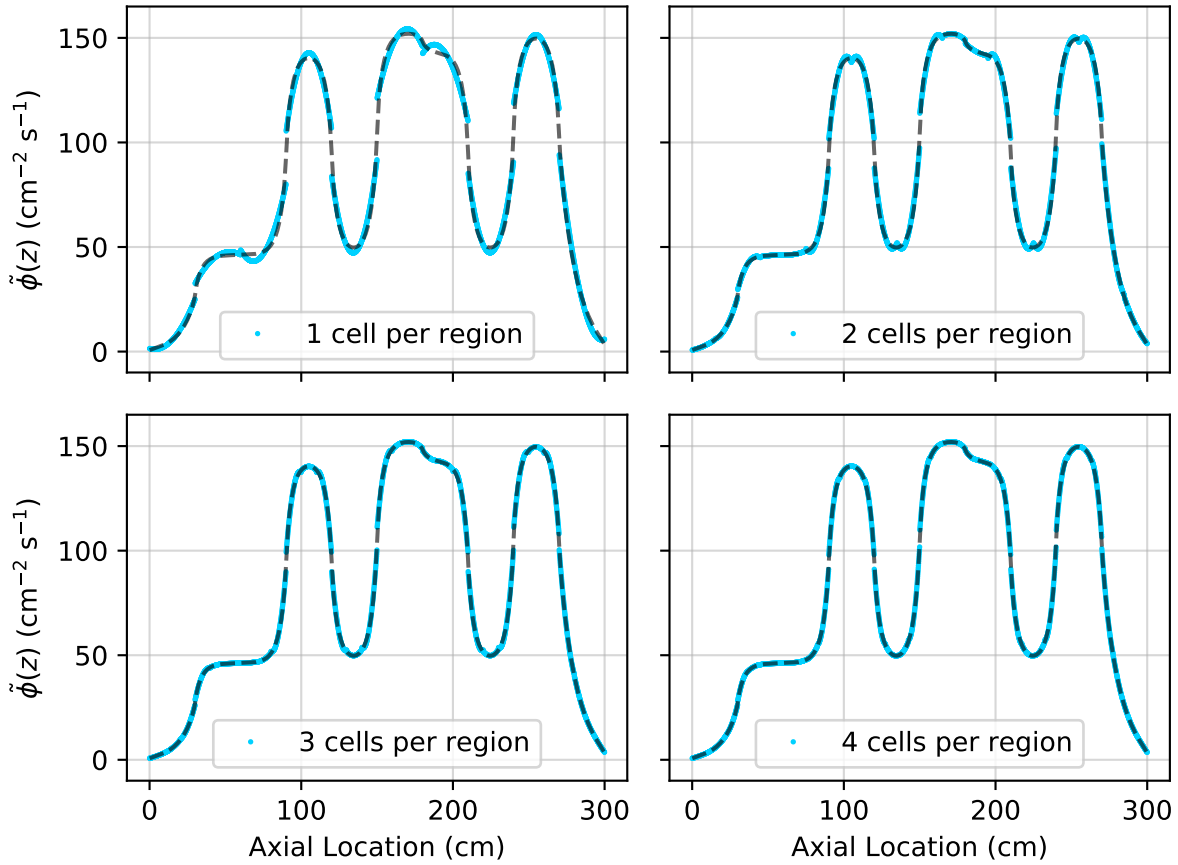


Figure 5.7: LDQZ scalar flux solutions for 10, 20, 30, and 40 cell discretizations of the 1D neutronics problem with a 3000 cell PWLD reference solution.

In Table 5.3, the LDQZ method in certain cases is shown to perform better than LD when performance is viewed as a function of both the error in the L^2 norm and the time to solution. When utilizing LDQZ with just one axial discretization per region, the error is about 35% less than when using LD with two cells per region and takes about two thirds of the time to run. If, for instance, the desired error is less than $4.00\text{E}1$, both a 4-cell LD run and a 2-cell LDQZ run offer this level of accuracy. However, the 2-cell LDQZ run returns a solution 23% more accurate than the 4-cell LD run in the L^2 norm with a 38% reduction in the time to solution. Ultimately, while the errors shown are large and the run times short, Table 5.3 shows that, for certain comparisons of axial discretizations, LDQZ has the ability to return a more accurate solution in less time than LD

for this 1D neutronics problem.

Table 5.3: L^2 error norms and total run times per processor for four axial discretizations of LD and LDQZ simulations for the 1D neutronics problem.

Cells per Region	LD		LDQZ		$\frac{\text{LDQZ Time}}{\text{LD Time}}$
	$\ \epsilon\ _{L^2}$	Run Time (ms)	$\ \epsilon\ _{L^2}$	Run Time (ms)	
1	1.78E2	54	4.84E1	73	1.35
2	7.48E1	108	2.49E1	127	1.18
3	4.75E1	151	1.33E1	173	1.15
4	3.23E1	205	8.23E0	237	1.16

5.5 Axial 1D 2 group criticality problem

In a step towards showing the usefulness of LDQZ for a more physically realistic problem, a 2-group criticality problem is devised with the same imitation of an axially one-dimensional geometry as the monoenergetic neutronics problem. Here, the radial domain is a $1\text{E}8 \times 1\text{E}8 \text{ cm}^2$ square, while the axial domain is of length 420 cm with 19 alternating regions of a fuel and a moderator and a reflector region on the top and bottom of the axial domain. Furthermore, the axial domain is bifurcated by the innermost axial region composed of a mixture of the fuel and an absorber, which has a noticeable effect on the flux profiles. While reflective boundary conditions are set on all four radial surfaces of the domain, vacuum conditions are set on both axial faces of the domain. Last, this problem assumes that all fission neutrons are born in the fast group and that neutrons may only downscatter. The LD and LDQZ thermal flux profiles for the first four axial refinements can be seen in Figures 5.8 and 5.9, respectively, while Table 5.4 provides details regarding accuracy and time to solution for each refinement.

Like the neutronics problem, Figure 5.9 shows the capability of LDQZ in effectively capturing highly-alternating flux profiles with few axial discretizations. LD, on the other hand, requires more

axial refinements to approximate domains with high axial heterogeneity, and too few refinements may result in very inaccurate flux approximations, as seen in the sharp peaks and dips in the top right plot of Figure 5.8. Also, while both methods are indeed discontinuous across cells, LDQZ offers flux solutions with less drastic jumps across material regions than LD. This is particularly consequential when many regions of axially alternating flux exist in a domain, as two very different flux values exist at these region interfaces. This problem in particular highlights this effect.

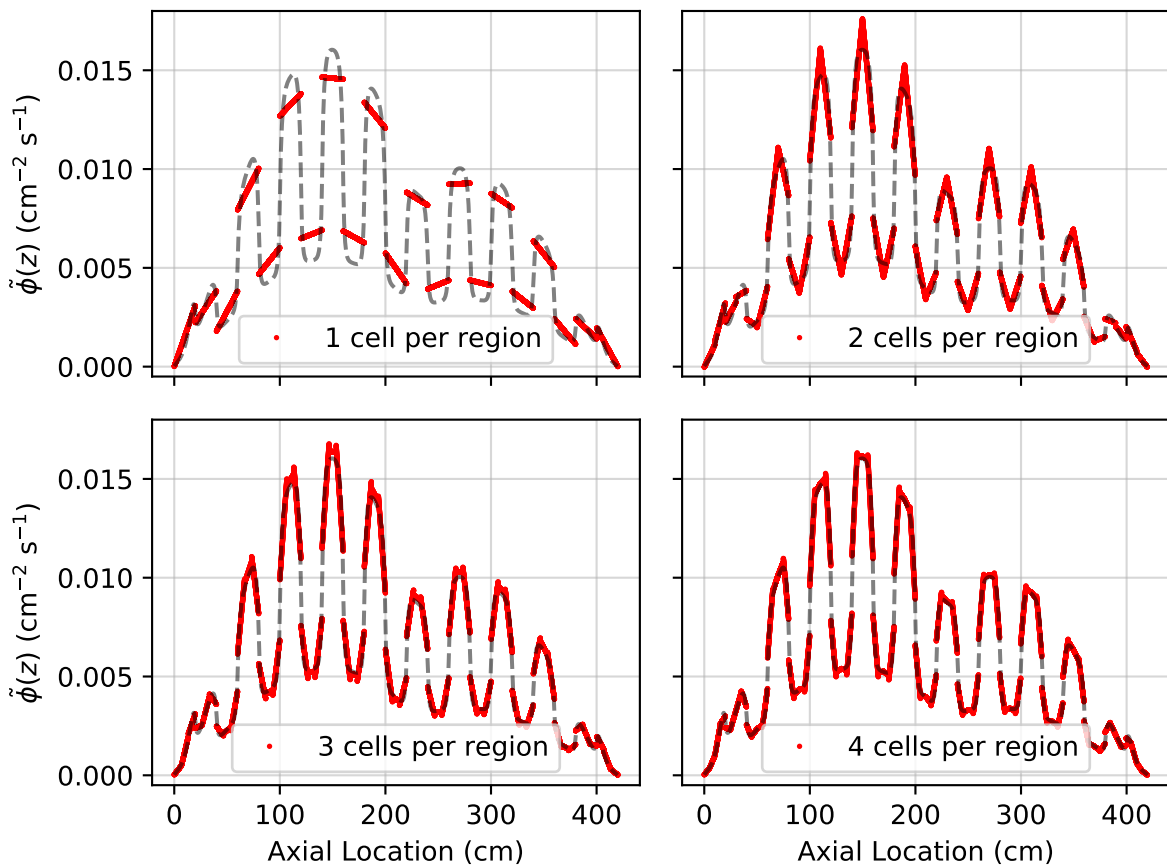


Figure 5.8: LD thermal flux solutions for 21, 42, 63, and 84 cell discretizations of the 1D criticality problem with a 18900 cell LDQZ reference solution.

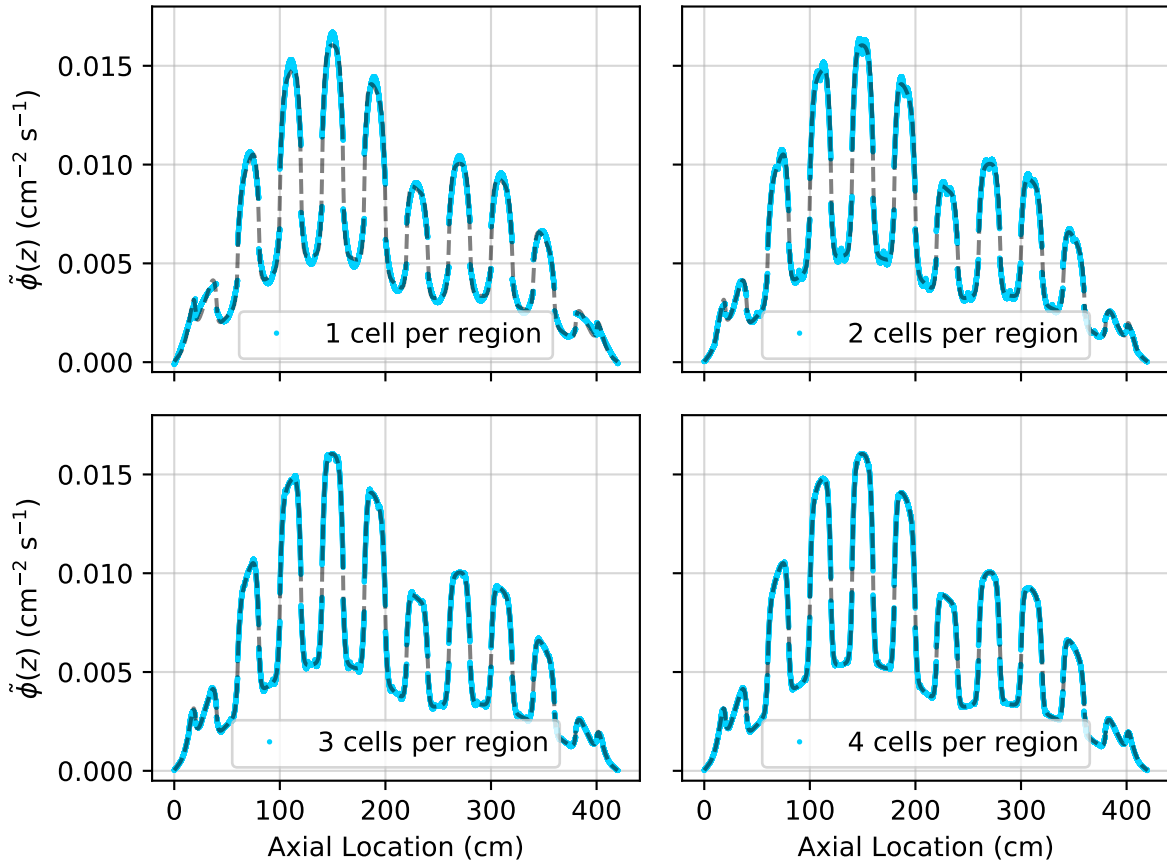


Figure 5.9: LDQZ thermal flux solutions for 21, 42, 63, and 84 cell discretizations of the 1D criticality problem with a 18900 cell LDQZ reference solution.

As an added benefit of this multigroup criticality problem, Table 5.4 depicts the profitability of LDQZ over LD in terms of accuracy as a function of both the error in the L^2 norm and the absolute difference in k_{eff} from the reference k_{eff} . Like the monoenergetic neutronics problem, improvements in accuracy in the L^2 error norm sometimes also are met with decreased run times. For instance, if an error maximum of 0.8 is warranted, the only LD option prescribed in Table 5.4 with a lesser error is a 4-cell-per-region simulation. However, LDQZ offers a 2-cell run with reductions in error and run time per processor of about 15% and 38%, respectively. On the other hand, if an error maximum of 1.5 is set, the 2-cell LD and 1-cell LDQZ runs have similar accuracy, but LDQZ returns a solution in half the time needed for LD.

Table 5.4: Δk_{eff} , L^2 error norms and total run times per processor for four axial discretizations of LD and LDQZ simulations for the 1D 2-group criticality problem run on 7 processors with an 18900-cell LDQZ reference; $k_{\text{ref}} = 1.015261$.

Cells per Region	LD			LDQZ			$\frac{\text{LDQZ Time}}{\text{LD Time}}$
	Δk_{eff} (pcm)	$\ \epsilon\ _{L^2}$	Run Time (min)	Δk_{eff} (pcm)	$\ \epsilon\ _{L^2}$	Run Time (min)	
1	530	3.55E0	3.5	30	1.44E0	4.7	1.34
2	146	1.42E0	9.2	36	6.03E-1	12.2	1.33
3	95	9.57E-1	15.9	15	3.78E-1	18.6	1.17
4	60	7.10E-1	19.6	7	2.58E-1	20.4	1.04

So far, simulation results have only been provided for the thermal flux data of this multigroup problem. However, Table 5.4 provides k -eigenvalue data, which gives somewhat of an overview of the accuracy of these two methods for all energy groups of the problem. The table shows that the eigenvalue results for all axial refinements of the LDQZ method are more accurate than any LD result. So, in cases of the methods' accuracy measured by either the L^2 norm of the scalar flux or Δk_{eff} , LDQZ is proven to outperform LD for this criticality problem with high axial heterogeneity.

5.6 3D C5G7 pin cell problem

Next, a 3D C5G7 UO₂ pin cell is modeled to demonstrate LDQZ on a real 3D benchmark problem [18]. This test case consists of a single pin with UO₂ fuel surrounded by water with a water reflector region above the pin, as detailed in Figure 5.10. Reflective boundary conditions are imposed on the four radial boundaries and on the bottom boundary, while a vacuum condition is imposed on the top.

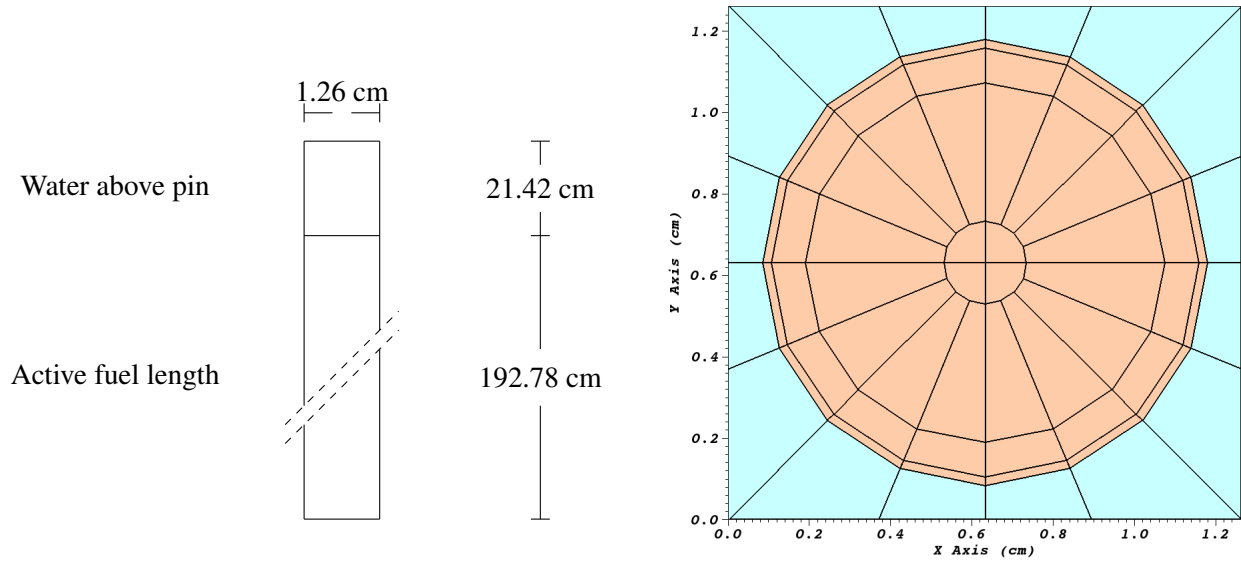


Figure 5.10: Axial and radial dimensions of the problem (left) and the radial mesh of the pin cell (right) for the C5G7 benchmark pin cell.

To generate Figures 5.12 and 5.13 which detail the axial LD and LDQZ flux in the pin cell respectively, a post-processing script was used that plots axial lineouts through a user-defined radial location using the DOF data output by PDT. The location of the lineouts shown is depicted in Figure 5.11. Multiple radial locations were selected to investigate effects of spatial self-shielding of the fuel on the axial lineout, but the differences seen were negligible.

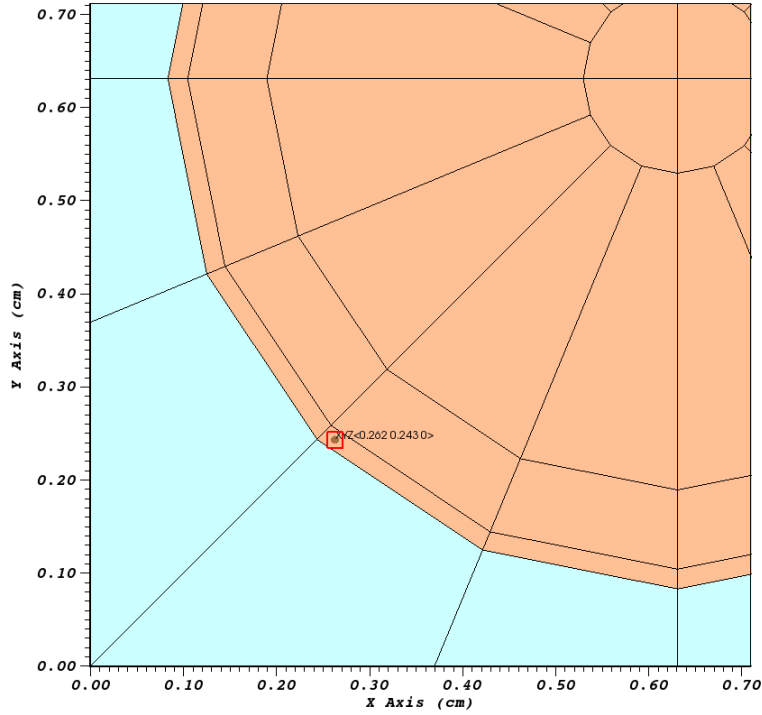


Figure 5.11: Radial location of the lineouts shown for the 3D C5G7 pin cell problem.

Figures 5.12 and 5.13 show that, even at a coarse refinement of only 11 axial layers, both LD and LDQZ approximate the thermal flux in the active fuel region accurately. However, this is not the case for either in the water reflector region where the flux spikes due to the reflected thermalized neutrons. Figures 5.14 and 5.15 focus on the effects of axial refinements in this water region where the coarse flux approximations for both methods perform poorly.

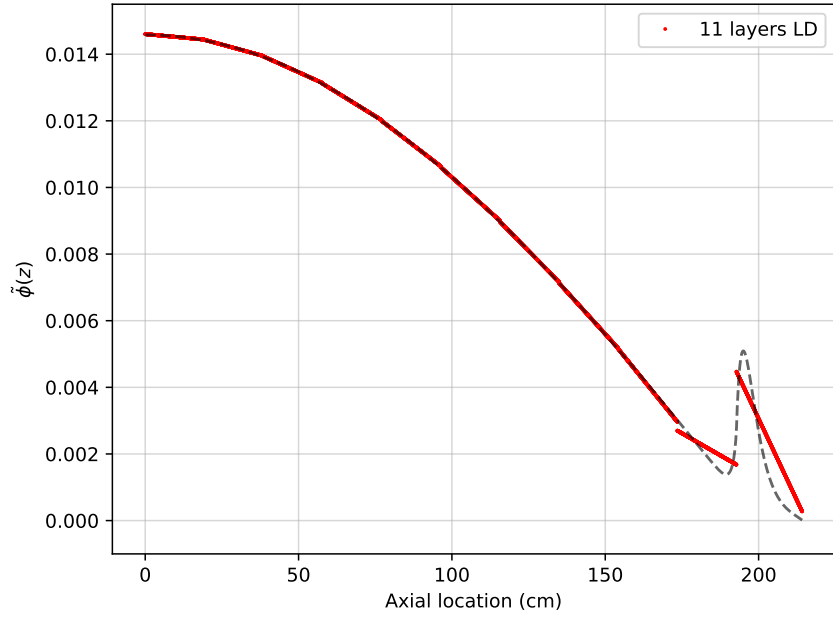


Figure 5.12: LD axial lineout of the 7th group scalar flux for an 11-layer axial refinement at the fuel edge with a 1980-layer LDQZ reference for the 3D C5G7 problem.

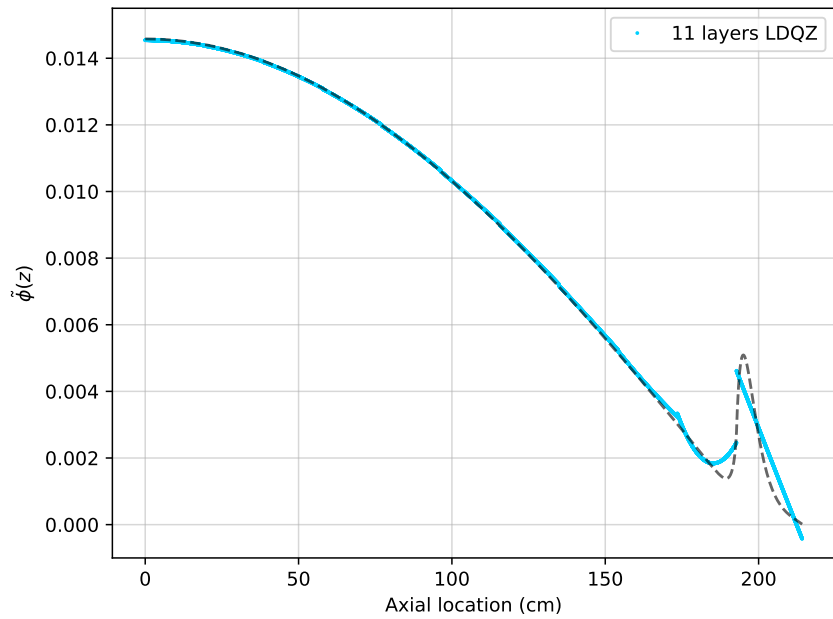


Figure 5.13: LDQZ axial lineout of the 7th group scalar flux for an 11-layer axial refinement at the fuel edge with a 1980-layer LDQZ reference for the 3D C5G7 problem.

Inspecting Figure 5.14 first, the lineouts show that LD struggles to approximate the thermal flux peak in the water reflector even at the highest refinement shown. The top of the pin cell, or the rightmost section of each of the lineouts, is the first region of the reflector that LD begins to approximate closest to the reference, while the peak seems to take shape last. This suggests that a number of refinements are still needed for LD to adequately capture the peak in the reflector. While this pattern of correctly approximating the flux nearest the top boundary first followed by the peak occurs when using LDQZ as well, Figure 5.15 shows that the LDQZ flux approximation is nearly identical to the reference at a 44-axial-layer refinement. Even at a 33-layer refinement, LDQZ returns a similar magnitude to that of the reference peak with only a slight error in the peak's axial location.

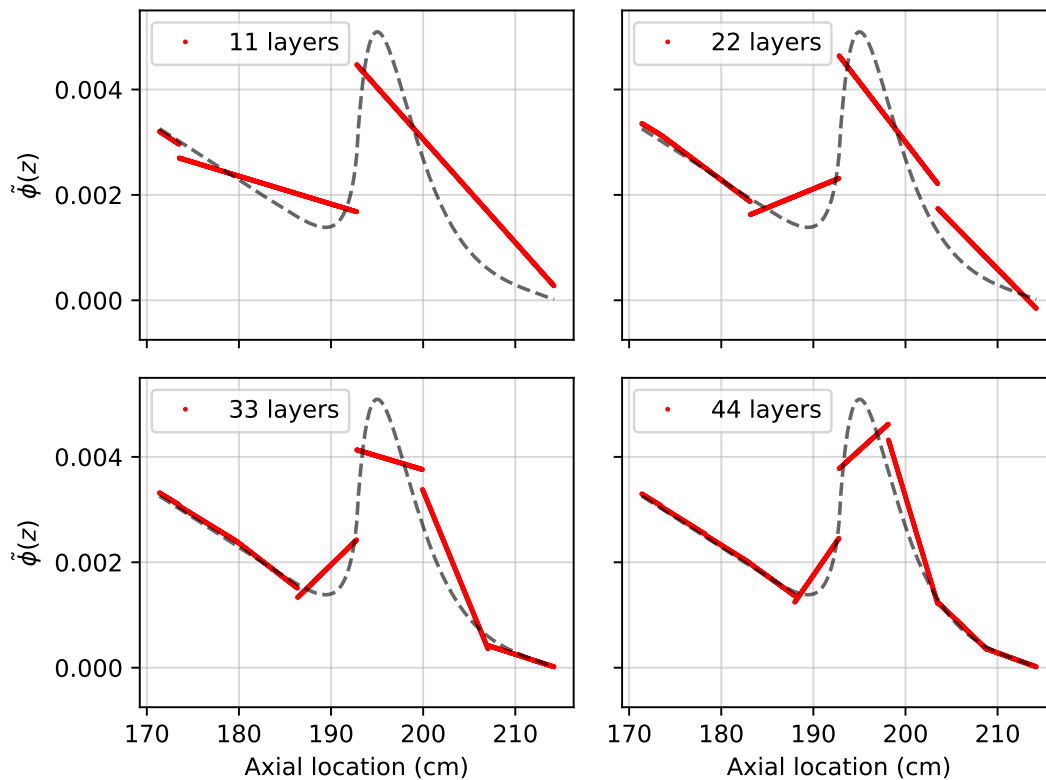


Figure 5.14: LD axial lineout of the 7th group scalar flux zoomed in near the water region for 11, 22, 33, and 44 total axial layers with a 1980-layer LDQZ reference for the 3D C5G7 problem.

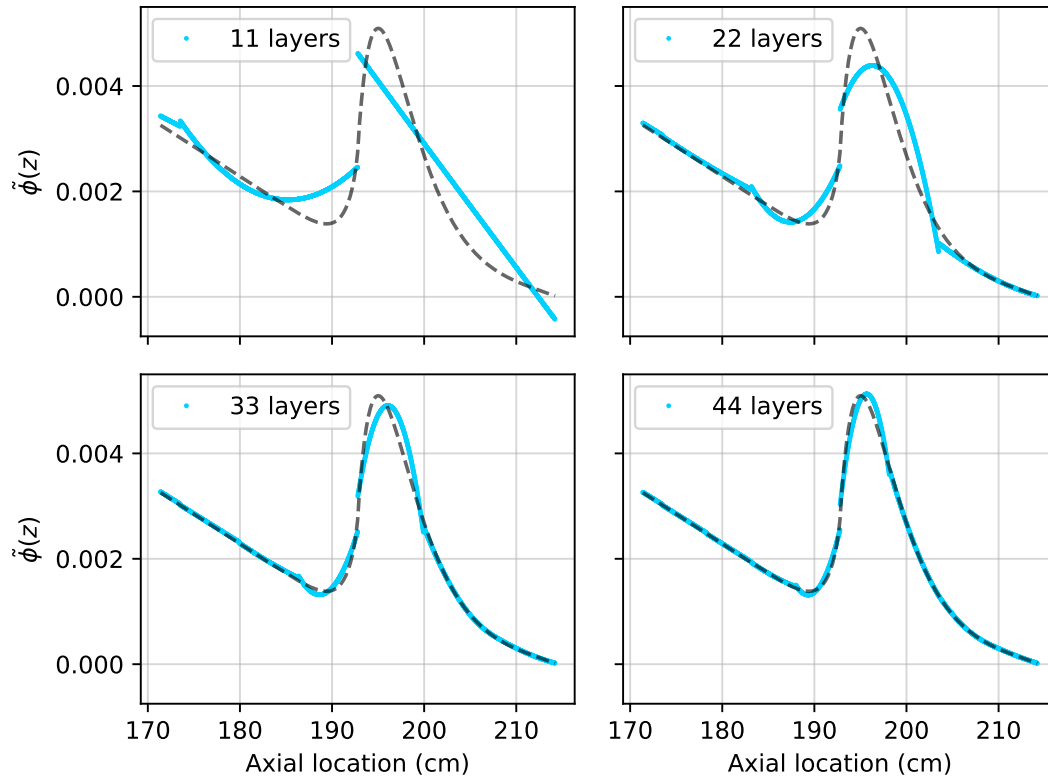


Figure 5.15: LDQZ axial lineout of the 7th group scalar flux zoomed in near the water region for 11, 22, 33, and 44 total axial layers with a 1980-layer LDQZ reference for the 3D C5G7 problem.

In an effort to quantify the error of these simulations, Figure 5.16 plots Δk_{eff} for seven axial refinements against a reference k_{eff} , while Table 5.5 details the same data in addition to the run times of each simulation per processor. Figure 5.16 serves to show the convergence-like behavior of Δk_{eff} for both LD and LDQZ with axial refinement, where LDQZ is more quickly approaching the reference k -eigenvalue.

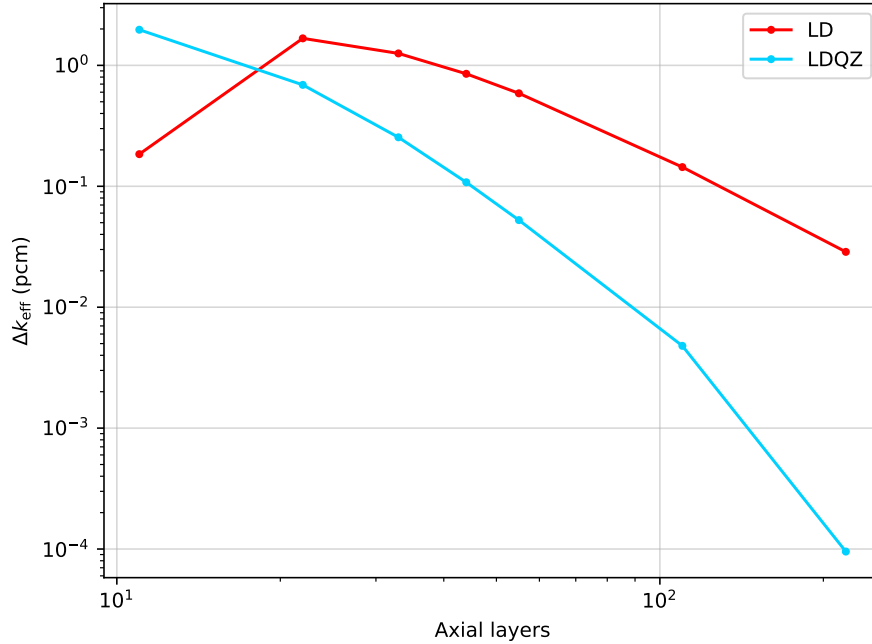


Figure 5.16: Δk_{eff} for multiple LD and LDQZ axial refinements against a 1980-layer LDQZ reference for the 3D C5G7 problem.

Even though the Δk_{eff} values gleaned from Table 5.5 show that even the coarsest refinements of both LD and LDQZ are quite accurate (< 2 pcm), the trend of LDQZ performing faster and more accurately than LD for a given error threshold is still present. This feature is most impressively realized when comparing a 110-axial-layer LD discretization to a 44-layer LDQZ simulation, where LDQZ returns a k -eigenvalue 25% more accurate in nearly half of the time per processor. As Figure 5.16 implies, this trend only improves with axial refinement as long as the increase in time to solution from LD to LDQZ at identical discretizations does not also increase. So, for this 3D C5G7 pin cell problem, LDQZ again is proven to perform faster and more accurately than LD at certain Δk_{eff} thresholds if accuracy is measured in terms of Δk_{eff} .

Table 5.5: Δk_{eff} and total run times per processor for axial refinements of LD and LDQZ simulations for the 3D C5G7 pin cell run on 32 processors with an 2200-layer LDQZ reference; $k_{\text{ref}} = 1.321264$.

Axial Layers	LD		LDQZ		$\frac{\text{LDQZ Time}}{\text{LD Time}}$
	Δk_{eff} (pcm)	Run Time (min)	Δk_{eff} (pcm)	Run Time (min)	
11	-1.84E-1	16	1.97E0	23	1.44
22	1.67E0	30	6.90E-1	38	1.27
33	1.26E0	45	2.54E-1	54	1.20
44	8.52E-1	54	1.08E-1	68	1.26
55	5.87E-1	64	5.25E-2	82	1.28
110	1.44E-1	132	4.79E-3	176	1.33
220	2.87E-2	298	9.53E-5	386	1.30

5.7 CASL VERA benchmark quarter fuel assembly problem

As the final investigation into the efficiency of LDQZ when compared with LD, the CASL VERA benchmark [19] Problem 3A 3D quarter fuel assembly is modeled. This simulation is both the most complex and most realistic of the four test problems used in this work, as this is a reactor benchmark problem modeled after a general fuel assembly existing in Watts Bar Unit 1 and used to validate new reactor physics codes. This quarter assembly problem is selected as it is a fully 3D benchmark problem that contains axial heterogeneity by way of the spacer grids located along the axial dimension of the assembly, as seen in Figure 5.18. A radial slice through a spacer grid region is shown in Figure 5.17. This quarter assembly mimics a full assembly with the imposition of reflective boundary conditions on each radial face and vacuum conditions on the axial faces.

Like the first three test problems, axial solution profiles are again presented against a reference solution for both LD and LDQZ. For this quarter assembly problem, however, the total integrated and volume-averaged power profiles are presented. Also, the reference solution is provided by

the benchmark itself and is not calculated with a highly refined LDQZ solve. As this problem is simulated in PDT by discretizing the energy variable into 93 groups, the resultant power accounts for the influence from the flux in each of the 93 groups. These power profiles are built by evaluating the integral power in each axial layer in the quarter fuel assembly, then normalizing by the volume of each layer. Figure 5.19 shows the power profiles for LD and LDQZ simulations with 49 axial layers in the active fuel region to match the number of layers in the benchmark, while Figure 5.20 shows the power profiles for LD and LDQZ with fewer axial layers to analyze any resultant changes in the shape of the power profiles. Finally, Table 5.6 details the temporal cost incurred by each simulation as well as the accuracy in terms of Δk_{eff} . It should be noted that neither LD nor LDQZ is capable of acceleration at the discrete energy group level; the long run times shown are partly a consequence of this. However, the comparisons between LD and LDQZ are still legitimate due to the absence of acceleration for both methods.

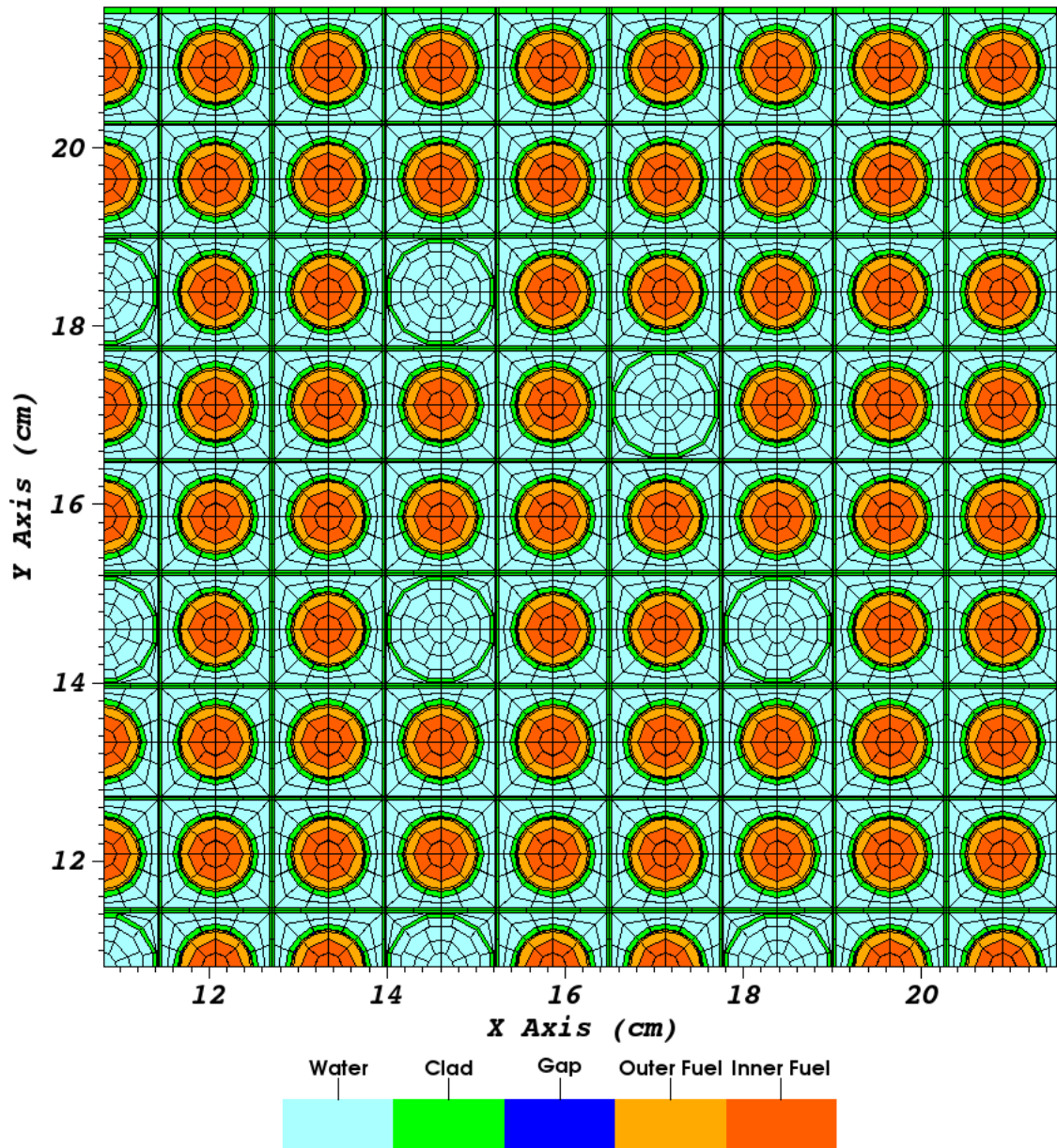


Figure 5.17: Radial slice of the CASL Problem 3A quarter assembly through an axial region with a grid spacer.

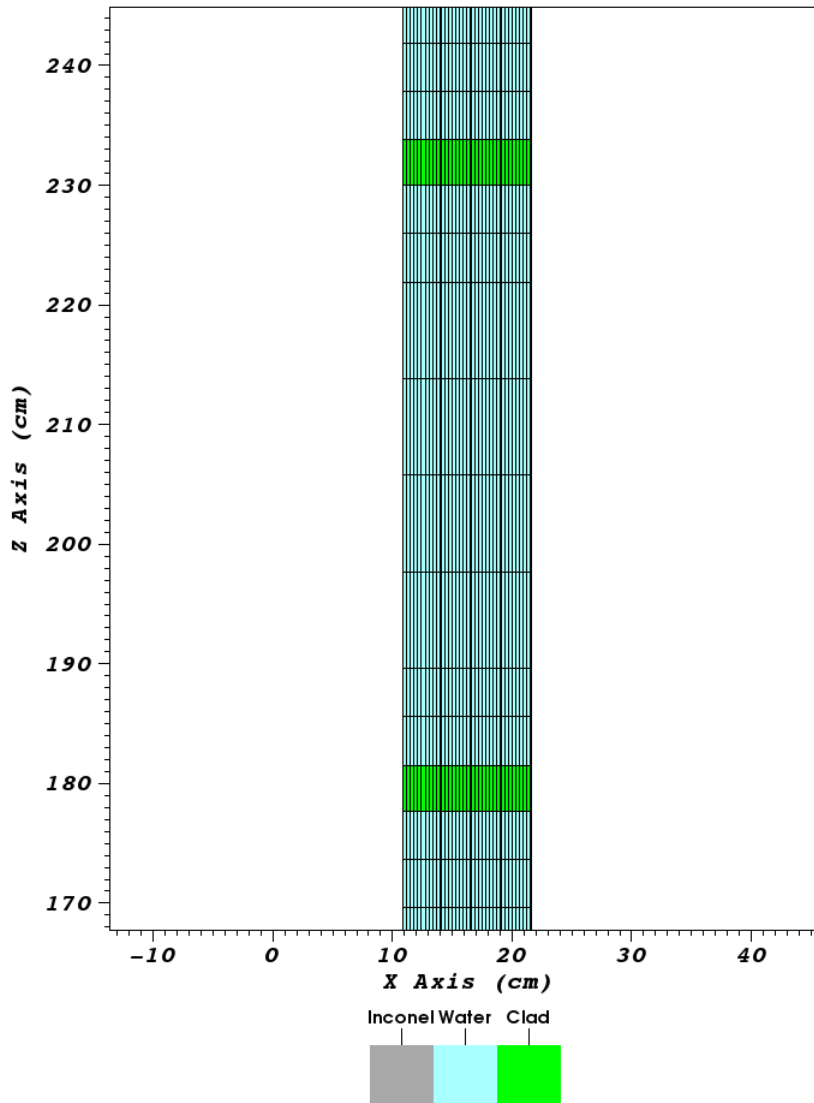


Figure 5.18: Axial slice of the CASL Problem 3A quarter assembly showing a region in the active fuel with grid spacers.

Figure 5.19 shows that, even at the coarsest axial refinement of one axial cell per layer, both LD and LDQZ return eigenvectors with nearly the same shape as that of the reference power profile. However, the LD solution more closely approximates the reference solution in many axial layers. This suggests that the CASL reference solution produced with the stochastic, continuous energy transport code KENO-VI [22] may not be identical to a highly-axially refined solution produced in

PDT with the 93-group cross sections. Even so, the difference between the LD and LDQZ power profiles are hardly visually differentiable.

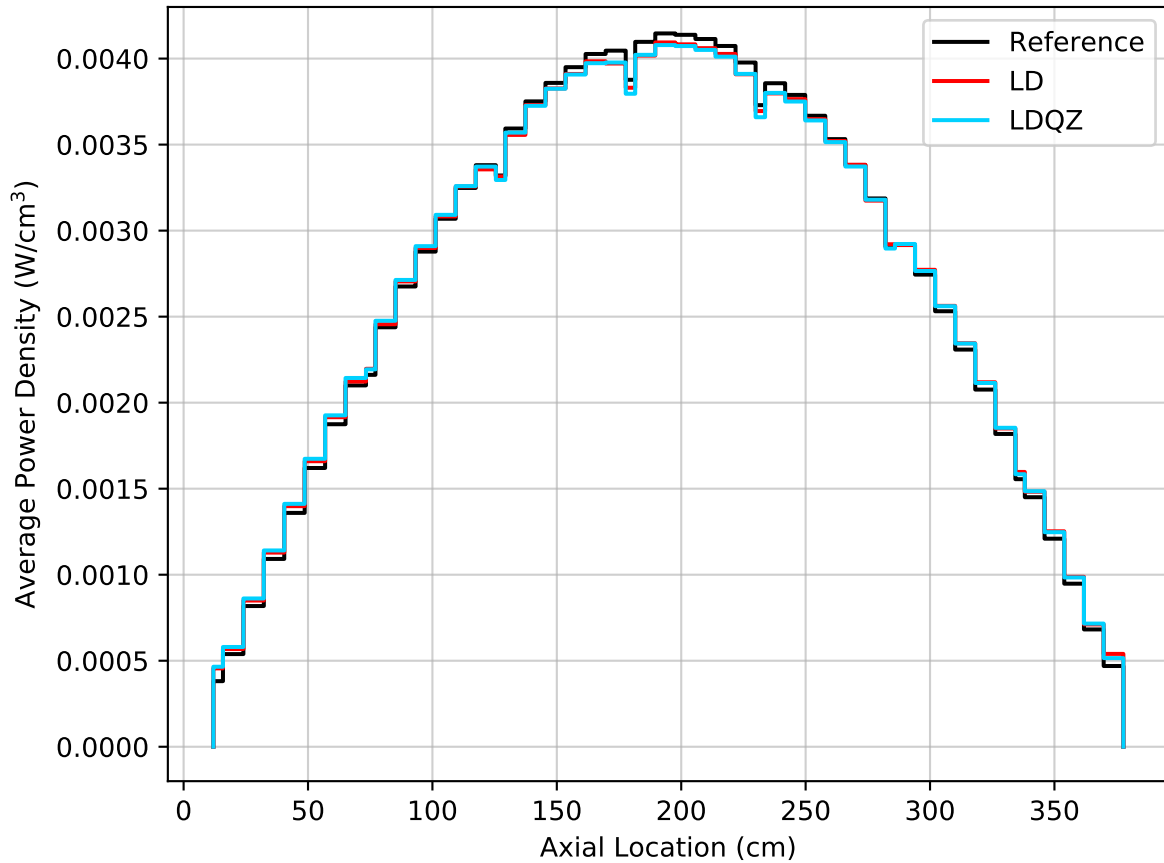


Figure 5.19: 49-layer integrated and volume-averaged axial power profiles for LD and LDQZ for the CASL VERA quarter assembly.

In an effort to widen the gap in accuracy between LD and LDQZ, the same simulation is run again, but with only 29 axial fuel layers instead of 49. Only the layers between grid spacer regions are heightened; the axial locations of the spacer grids are left untouched. The reference solution provided by the benchmark is also coarsened by simply averaging the appropriate layers in order to match the new LD and LDQZ layers. While this averaging of the reference solution is exact, it no longer represents the true power profile that might be returned by the code used in the benchmark

with the new mesh. This is of little consequence, however, as Figure 5.20 only serves as a visual comparison of results.

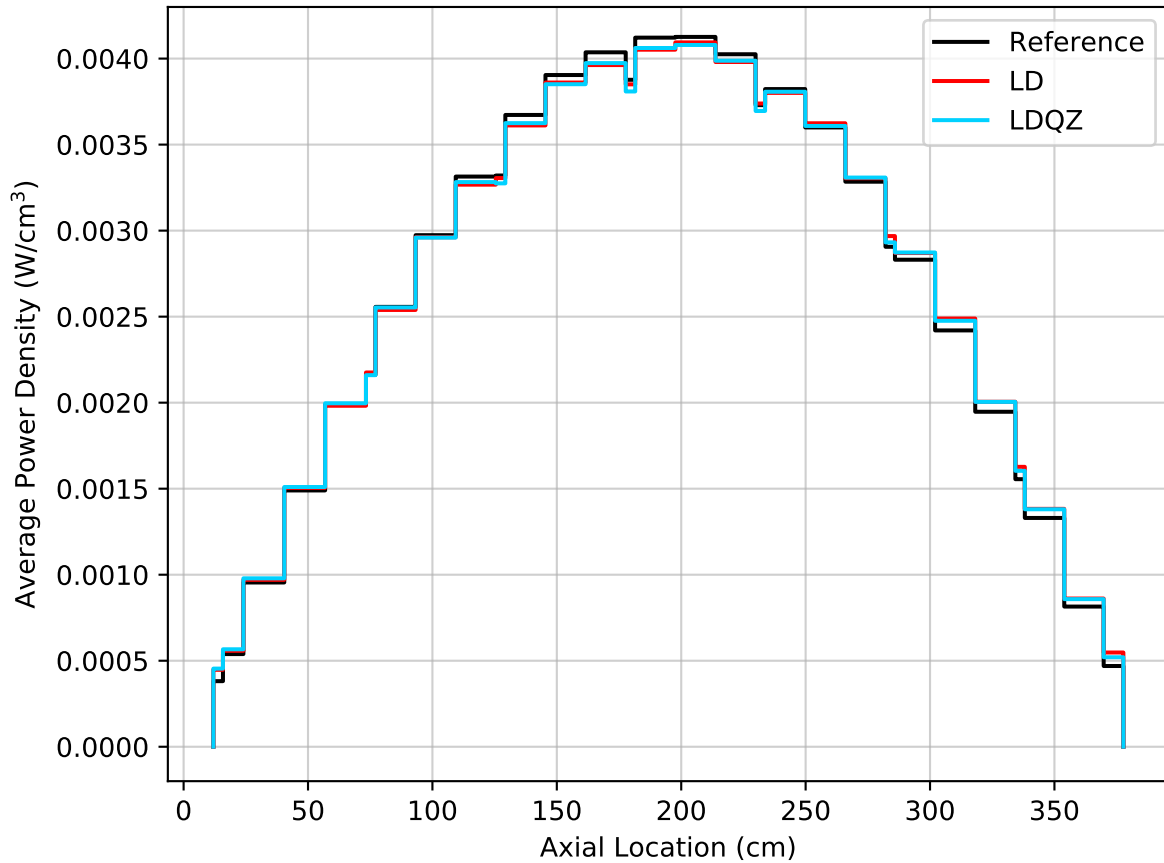


Figure 5.20: 29-layer integrated and volume-averaged axial power profiles for LD and LDQZ for the CASL VERA quarter assembly.

This result is both encouraging and discouraging at the same time. On one hand, both the LD and LDQZ simulations provide a high level of accuracy when compared to the averaged reference solution. On the other hand, LDQZ in no way offers a more accurate solution than LD even for this coarser refinement. This is true even if the 93-group cross sections used in the PDT simulations produce a reference solution that differs from the benchmark solution, as the LD and LDQZ power profiles are effectively the same.

Finally, Table 5.6 shows quantitatively the similarity of the LD and LDQZ simulations in terms of their differences in k_{eff} with the benchmark reference. These results corroborate the similarities in the power profiles for this problem, as the Δk_{eff} values within the range of -10 to -20 pcm indicate the LD and LDQZ eigenvalues are similar enough to the reference to be considered equal. At this error magnitude, the error is dominated by the use of the 93-group cross sections when compared to the reference k_{eff} obtained with continuous energy cross sections. So, any further comparison of the LD and LDQZ eigenvalues to each other is meaningless. In conclusion, Figure 5.20 and Table 5.6 show that the use of LDQZ does not produce a more accurate result than LD for a coarsely refined Problem 3A quarter assembly when comparing either the axial power profiles or Δk_{eff} to the benchmark reference solution, and the LDQZ simulation time is about 33% to 50% longer as well.

Table 5.6: Δk_{eff} and total run times per processor for two axial discretizations of LD and LDQZ simulations for the CASL quarter assembly problem with a reference $k_{\text{eff}} = 1.175722$ provided by the benchmark.

Layers in Fuel Length	Total Layers	LD			LDQZ			$\frac{\text{LDQZ Time}}{\text{LD Time}}$
		k_{eff}	Δk_{eff} (pcm)	Run Time (hrs)	k_{eff}	Δk_{eff} (pcm)	Run Time (hrs)	
29	44	1.175529	-14	49.8	1.175511	-15	74.4	1.49
49	64	1.175508	-16	74.4	1.175474	-18	99.5	1.34

6. CONCLUSIONS

This work serves to implement an axially higher order DFEM deemed LDQZ in TAMU's deterministic transport code PDT. Essentially, LDQZ is a p -refined LD method, but only in the axial dimension. This selective p -refinement is performed only axially due to the disparity between radial and axial spatial refinement present in most reactor geometries. As reactor geometries are usually highly radially refined due to the high material heterogeneity across pin cells and fuel assemblies, radial p -refinement is excessive and possibly harmful as a result of the added temporal cost in evaluating the solution.

As mentioned, LDQZ is an extension of LD by way of the fifth LDQZ basis function in the set. This is the only basis function in the set that differs from those in the LD set. Since this is the only difference between LD and LDQZ in the character of the solution approximation, attention can be focused on the consequences of the fifth basis function on the solution when comparing LD and LDQZ. Additionally, the axially quadratic basis function preserves attractive qualities possessed by the LD method, as this function integrates to zero when multiplied by any one of the other basis functions when integrating over the volume of the cell. LDQZ does possess limitations though, as LDQZ is only capable of handling axially extruded meshes and cannot be accelerated at the energy group level solve. The former is essentially a non-issue, as most reactor meshes are axially extruded. The lack of acceleration is more impactful, but LD is similarly hindered. Implementing within-group acceleration for both of these methods is possible future work.

The goal of this work is to show that the LDQZ method can be utilized to return a more accurate solution in the same or less time than LD for particular prescribed error thresholds. This goal is realized for the simpler test problems, namely the 1D neutronics and criticality problems, where nontrivial reductions in error are accompanied by significant reductions in run time when comparing LD to LDQZ. This is not the case, however, for the C5G7 and CASL VERA benchmark problems. The least accurate LD Δk_{eff} value for the C5G7 problem is less than 2 pcm; simulations are rarely required to be this accurate. While the LDQZ results are more accurate for all but the

first discretization, the added time cost renders LDQZ the less effective method for this problem. The LD method also performs as well as LDQZ for the CASL benchmark problem even for coarse axial refinements.

Overall, in the context of the test problems presented in this work, LDQZ does not show the improvement in efficiency over LD needed to warrant its use in reactor simulations. LDQZ does show more promise when compared to LD for simple problems with very high axial heterogeneity, but these problems are either neutronics problems with no fission sources or criticality problems with only one or two discrete energy groups. However, more realistic problems like those presented in the C5G7 and CASL VERA benchmarks do not exhibit the degree of axial heterogeneity present in the 1D problems shown in Sections 5.4 and 5.5. As the results show, LD is capable of handling the minor axial heterogeneity present in the C5G7 and CASL VERA benchmark problems without the added time required by LDQZ during the solve process. The LD method is therefore sufficient for solving complex reactor problems.

REFERENCES

- [1] PDT Developers, *PDT User Manual*. TAMU.
- [2] Lathrop, K. D., U.S. Atomic Energy Commission, Los Alamos Scientific Laboratory, “DTF-IV: a FORTRAN-IV program for solving the multigroup transport equation with anisotropic scattering,” 1965.
- [3] K. Lathrop, R. Anderson, and F. Brinkley, “TRANZIT: A program for multigroup time-dependent transport in (ρ, z) cylindrical geometry,” tech. rep., Los Alamos Scientific Lab., N. Mex., 1970.
- [4] H. G. Joo, J. Y. Cho, K. S. Kim, C. C. Lee, and S. Q. Zee, “Methods and performance of a three-dimensional whole-core transport code DeCART,” 2004.
- [5] J.-J. Lautard and J.-Y. Moller, “MINARET, a deterministic neutron transport solver for nuclear core calculations,” 2010.
- [6] J. Morel, “Lecture notes for Neutron Transport Theory,” February 2018.
- [7] J. R. Shewchuk, “Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator,” in *Applied Computational Geometry: Towards Geometric Engineering* (M. C. Lin and D. Manocha, eds.), vol. 1148 of *Lecture Notes in Computer Science*, pp. 203–222, Springer-Verlag, May 1996. From the First ACM Workshop on Applied Computational Geometry.
- [8] C. Geuzaine and J.-F. Remacle, “Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities,” *International journal for numerical methods in engineering*, vol. 79, no. 11, pp. 1309–1331, 2009.
- [9] Y. Wang, *Adaptive mesh refinement solution techniques for the multigroup S_N transport equation using a higher-order discontinuous finite element method*. Texas A&M University, 2009.

- [10] J. S. Warsa and A. K. Prinja, “ p -adaptive numerical methods for particle transport,” *Transport Theory and Statistical Physics*, vol. 28, no. 3, pp. 229–270, 1999.
- [11] A. M. Mirza, S. Iqbal, and F. Rahman, “A spatially adaptive grid-refinement approach for the finite element solution of the even-parity Boltzmann transport equation,” *Annals of Nuclear Energy*, vol. 34, no. 7, pp. 600 – 613, 2007.
- [12] D. Fournier, R. Herbin, and R. Tellier, “Discontinuous galerkin discretization and hp -refinement for the resolution of the neutron transport equation,” *SIAM Journal on Scientific Computing*, vol. 35, no. 2, pp. A936–A956, 2013.
- [13] B. Cockburn, “Discontinuous galerkin methods,” *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 83, no. 11, pp. 731–754, 2003.
- [14] M. Herman, C. S. E. W. Group, *et al.*, “ENDF-6 Formats Manual Data Formats and Procedures for the Evaluated Nuclear Data File ENDF/B-VI and ENDF/B-VII,” tech. rep., Brookhaven National Laboratory (BNL) National Nuclear Data Center, 2009.
- [15] W. C. Fan, C. R. Drumm, and J. L. Powell, “Discrete ordinates approximations to the first- and second-order radiation transport equations,” *NASA STI/Recon Technical Report N*, vol. 3, 2002.
- [16] M. Hanuš, *Mathematical Modeling of Neutron Transport*. PhD thesis, University of West Bohemia, 2014.
- [17] J. J. Duderstadt and L. J. Hamilton, *Nuclear Reactor Analysis*. Wiley & Sons, 1976.
- [18] E. Lewis, M. Smith, N. Tsoulfanidis, G. Palmiotti, T. Taiwo, and R. Blomquist, “Benchmark specification for deterministic 2-D/3-D MOX fuel assembly transport calculations without spatial homogenization (C5G7 MOX),” *NEA/NSC*, vol. 280, 2001.
- [19] A. T. Godfrey, “VERA Core Physics Benchmark Progression Problem Specifications,” tech. rep., Oak Ridge National Laboratory, 2014.

- [20] B. Cockburn, B. Dong, and J. Guzmán, “Optimal convergence of the original DG method for the transport-reaction equation on special meshes,” *SIAM Journal on Numerical Analysis*, vol. 46, no. 3, pp. 1250–1265, 2008.
- [21] A. Sood, R. A. Forster, and D. K. Parsons, “Analytical benchmark test set for criticality code verification,” *Progress in nuclear energy*, vol. 42, no. 1, pp. 55–106, 2003.
- [22] D. Hollenbach, L. Petrie, and N. Landers, “KENO-VI: A General Quadratic Version of the KENO Program,” tech. rep., ORNL/NUREG/CSD, 2007.

APPENDIX A

This appendix provides a summary of the LDQZ matrices developed in Section 4.2. These are

- the gradient matrix G :

$$G_{LDQZ} = \begin{bmatrix} & & & & 0 \\ & & & & 0 \\ & & & & 0 \\ & G_{LD} & & & 0 \\ & & & & 2\xi V \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{or} \quad G_{LDQZ} = \begin{bmatrix} & & & & 0 \\ & & & & 0 \\ & & & & 0 \\ & G_{LD} & & & 0 \\ & & & & 2G_{14} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

- the mass matrix M :

$$M_{LDQZ} = \begin{bmatrix} & & & & 0 \\ & & & & 0 \\ & & & & 0 \\ & M_{LD} & & & 0 \\ & & & & 0 \\ 0 & 0 & 0 & 0 & \frac{4}{5}V \end{bmatrix} \quad \text{or} \quad M_{LDQZ} = \begin{bmatrix} & & & & 0 \\ & & & & 0 \\ & & & & 0 \\ & M_{LD} & & & 0 \\ & & & & 0 \\ 0 & 0 & 0 & 0 & \frac{4}{5}M_{11} \end{bmatrix},$$

- the vertical surface matrix for each vertical face K_{f_v} :

$$K_{f_v,LDQZ} = \begin{bmatrix} & & & & 0 \\ & & & & 0 \\ & & & & 0 \\ & K_{f_v,LD} & & & 0 \\ & & & & 0 \\ 0 & 0 & 0 & 0 & \frac{4}{5}A_{f_v} \end{bmatrix} \quad \text{or} \quad K_{f_v,LDQZ} = \begin{bmatrix} & & & & 0 \\ & & & & 0 \\ & & & & 0 \\ & K_{f_v,LD} & & & 0 \\ & & & & 0 \\ 0 & 0 & 0 & 0 & \frac{4}{5}K_{f_v,11} \end{bmatrix},$$

- and the horizontal surface matrix H :

$$H_{LDQZ} = \begin{bmatrix} & & & & & 2H_{11} \\ & & & & & 2H_{21} \\ & & H_{LD} & & & 2H_{31} \\ & & & & & 2H_{41} \\ & & & & & 2H_{51} \\ \text{sym} & & & & & \end{bmatrix} .$$