

ABSTRACT

DAWN, WILLIAM CHRISTOPHER. Simulation of Fast Reactors with the Finite Element Method and Multiphysics Models. (Under the direction of Scott P. Palmtag).

Renewed interest in advanced nuclear power reactors, such as the Versatile Test Reactor (VTR) at Idaho National Laboratory (INL), has encouraged enhanced modeling and simulation of fast nuclear power reactors. Since the inception of fast reactors in the early days of nuclear engineering, with reactors such as at the Experimental Breeder Reactor I (EBR-I) and Fermi 1, many new modeling techniques have been developed. This work seeks to introduce modern methods and multiphysics simulation of fast reactors.

In this work, the multigroup neutron diffusion equation is solved via the Finite Element Method (FEM), allowing for the use of unstructured and general meshes. By using an unstructured mesh, physical phenomena such as thermal expansion can be modeled and allowed to distort the mesh. Additionally, the FEM will allow for spatial refinement by means of both traditional mesh refinement and the use of higher-order methods without regenerating the mesh. Unique to this thesis is the use of pentahedral wedge elements in the FEM mesh for three-dimension simulations. Wedge elements are selected for their natural description of hexagonal geometry common to fast reactors.

Thermal feedback effects within a fast reactors are also modeled in this work. A simplified thermal hydraulic model is developed, modeling both axial heat convection and radial heat conduction. Temperatures from this model are used to calculate temperature dependent neutron cross-sections. In addition to the thermal hydraulic model, a thermal expansion model is developed. Thermal expansion effects significantly impact reactor behavior and contribute to the passive safety of fast reactor designs as demonstrated in the experiments performed at Experimental Breeder Reactor II (EBR-II) [Pla87].

Using the multiphysics models developed in this work, a typical fast reactor is simulated at operating conditions. The models as implemented demonstrate expected reactor behavior for a fast reactor. For the simulated reactor, reactivity feedback coefficients are calculated which would not be possible without a coupled multiphysics model. These results can be used to describe the passive safety features and feedback effects of such a nuclear reactor.

© Copyright 2020 by William Christopher Dawn

All Rights Reserved

Simulation of Fast Reactors with the Finite Element Method and Multiphysics Models

by
William Christopher Dawn

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Nuclear Engineering

Raleigh, North Carolina

2020

APPROVED BY:

Joseph M. Doster

Ralph C. Smith

Scott P. Palmtag
Chair of Advisory Committee

DEDICATION

To the future of clean electricity.

BIOGRAPHY

William Christopher Dawn was born and raised in Stafford, Virginia. He attended public schools there for his primary education, participating in the Commonwealth Governor's School in High School. William earned a Bachelor's of Science degree in Nuclear Engineering from North Carolina State University (NC State) in May 2017. After his Master's degree, William will remain at NC State to pursue a Ph.D. degree in Nuclear Engineering.

William is a fellow of the Integrated University Program (IUP) facilitated by U.S. Department of Energy Office of Nuclear Energy (DOE-NE). During his undergraduate and graduate careers, he has had the opportunity to work with GE-Hitachi Nuclear Energy Americas LLC (GEH) and Oak Ridge National Laboratory (ORNL). William has also made contributions to the Consortium for Advanced Simulation of LWRs (CASL).

ACKNOWLEDGEMENTS

This work would not have been possible without the help of friends and family. I would like to thank my Mom and Dad, Suzanne and Bill Dawn, for their patience, their listening, and their advice. Their support has helped to make this work a reality.

I would also like to thank my advisor, Dr. Scott Palmtag. We have both learned tremendously during this process. His consistency and desire to know more have kept me busy these last few months and I am grateful.

TABLE OF CONTENTS

List of Tables	viii
List of Figures	x
List of Acronyms	xii
Chapter 1 Introduction	1
1.1 Motivation	1
1.2 Geometry Description	2
1.3 Cross Section Treatment	5
1.4 Thesis Organization	7
Chapter 2 Finite Element Neutron Diffusion	9
2.1 Introduction	9
2.2 Multigroup Neutron Diffusion Equation	10
2.3 Formulation of Finite Element Equations	12
2.3.1 Derivation	12
2.3.2 Matrix Quantities	17
2.3.3 Quadratures	22
2.4 Power Iterations	24
2.4.1 Convergence of Power Iteration Method	25
2.4.2 Calculation of Source with Power Iterations	28
2.5 Implementation	28
2.5.1 Algorithm	29
2.5.2 Memory and Storage	31
2.5.3 Boundary Conditions	32
2.5.4 Linear System Solution	33
Chapter 3 Neutron Diffusion Results	36
3.1 Introduction	36
3.2 Error Analysis	37
3.3 Analytic Solutions	38
3.3.1 One-Dimension, One-Group, Fixed Source	38
3.3.2 One-Dimension, One-Group, Criticality	38
3.3.3 Two-Dimension, One-Group, Criticality	40
3.3.4 One-Dimension, Two-Group, Criticality	40
3.3.5 One-Dimension, One-Group, Two-Region, Criticality	40
3.3.6 Three-Dimension, One-Group, Finite Cylinder	40
3.4 Two-Dimensional Benchmark Solutions	43
3.4.1 VVER440	43
3.4.2 SNR	45
3.4.3 HWR	46
3.4.4 IAEA Hex	46

3.5	Three-Dimensional Benchmark Solutions	50
3.5.1	MONJU	50
3.5.2	KNK	50
Chapter 4	Thermal Hydraulics	52
4.1	Introduction	52
4.2	Material Properties	52
4.3	Power Normalization	54
4.4	Axial Convection Model	55
4.4.1	Geometric Model	55
4.4.2	Channel Mass Flow	57
4.4.3	Chunk Powers	57
4.4.4	Channel Enthalpy	57
4.5	Radial Conduction Model	58
4.5.1	Geometric Model	59
4.5.2	Surface Temperature and Centerline Temperatures	60
4.5.3	Average Temperatures	67
4.6	Cross Section Treatment	69
4.6.1	Coolant Cross Sections	71
4.6.2	Clad Cross Sections	71
4.6.3	Bond Cross Sections	71
4.6.4	Fuel Cross Sections	72
4.7	Thermal Hydraulic Results	72
4.7.1	Total Reactor Power	72
4.7.2	Radial Results	73
4.7.3	Axial Results	74
Chapter 5	Thermal Expansion	78
5.1	Necessity of Modeling	78
5.2	Material Properties	79
5.3	Model Details	80
5.3.1	Expansion of Finite Element Coordinates	81
5.3.2	Expansion of Area Fractions	83
5.3.3	Conservation of Material and Cross Section Effects	83
5.4	Results	85
Chapter 6	Coupled Multiphysics Results	87
6.1	Power Reactor Modeling	87
6.2	Advanced Burner Reactor – MET-1000	87
6.3	Reactivity Coefficients	90
6.3.1	Power Reactivity Coefficient	91
6.3.2	Thermal Expansion Reactivity Coefficient	91
6.3.3	Fuel Temperature Reactivity Coefficient	92
6.3.4	Coolant Temperature Coefficient	92
6.4	Results	93

Chapter 7	Summary, Conclusions, and Recommendations	96
7.1	Summary of Simulation Results	96
7.2	Conclusions	97
7.3	Recommendations for Future Research	97
7.3.1	Depletion Capabilities	98
7.3.2	Higher Order Finite Elements	98
7.3.3	Simplified P_N Solution	98
7.3.4	Encouraging Code Usage	98
Bibliography		100
Appendices		103
Appendix A	Analytic Solutions to the Neutron Diffusion Equation	104
A.1	Introduction	104
A.2	One-Dimension, One-Group, Fixed Source	106
A.3	One-Dimension, One-Group, Criticality	108
A.4	Two-Dimension, One-Group, Criticality	111
A.5	One-Dimension, Two-Group, Criticality	114
A.6	One-Dimension, One-Group, Two-Region, Criticality	118
A.7	Finite-Cylinder, One-Group, Criticality	124
Appendix B	Brief Compendium of Neutron Diffusion Benchmarks	130
B.1	Introduction	130
B.2	Two-Dimensional Benchmark Problems	131
B.2.1	VVER440	132
B.2.2	SNR	133
B.2.3	HWR	136
B.2.4	IAEA Hex	136
B.3	Three-Dimensional Benchmark Problems	140
B.3.1	MONJU	141
B.3.2	KNK	141

LIST OF TABLES

Table 1.1	Temperatures Selected for Cross Section Libraries.	6
Table 2.1	Quadrature Orders for FEM Quantities.	23
Table 2.2	Jacobi for Selected Elements.	23
Table 3.1	One-Dimension, One-Group, Fixed Source Convergence Study Results.	39
Table 3.2	One-Dimension, One-Group, Criticality Convergence Study Results.	39
Table 3.3	Two-Dimension, One-Group, Criticality Convergence Study Results.	40
Table 3.4	One-Dimension, Two-Group, Criticality Convergence Study Results.	41
Table 3.5	One-Dimension, One-Group, Two-Region, Criticality Convergence Study Results.	41
Table 3.6	Finite Cylinder Convergence Study Results.	43
Table 3.7	VVER440 Benchmark Convergence Study.	44
Table 3.8	SNR Benchmark Convergence Study.	46
Table 3.9	HWR Benchmark Convergence Study.	46
Table 3.10	IAEA Hex Benchmark Convergence Study. No Reflector. $\alpha = 0.125$	48
Table 3.11	IAEA Hex Benchmark Convergence Study. No Reflector. $\alpha = 0.500$	48
Table 3.12	IAEA Hex Benchmark Convergence Study. With Reflector. $\alpha = 0.125$	49
Table 3.13	IAEA Hex Benchmark Convergence Study. With Reflector. $\alpha = 0.500$	49
Table 3.14	MONJU Benchmark Rod Worth Results.	51
Table 3.15	KNK Benchmark Rod Worth Results.	51
Table 4.1	Default Constant Thermal Conductivity for Sodium and HT9.	53
Table 4.2	System Properties for Axial Model Verification.	74
Table 6.1	Advanced Burner Reactor Refinement Results.	88
Table 6.2	Multiphysics Contributions to Total Power Defect.	94
Table A.1	Case Matrix for Analytic Solutions.	105
Table A.2	One-Group Sample Cross Sections.	110
Table A.3	Two-Group VVER440 Material Constants.	118
Table A.4	Two-Region Material Constants.	123
Table A.5	Finite Cylinder Cross Sections.	128
Table B.1	Case Matrix for Benchmark Solutions.	131
Table B.2	VVER440 Cross Sections.	133
Table B.3	VVER440 Fission Spectrum.	133
Table B.4	SNR Cross Sections.	135
Table B.5	SNR Fission Spectrum.	136
Table B.6	HWR Cross Sections.	138
Table B.7	HWR Fission Spectrum.	138
Table B.8	IAEA Hex Effective Neutron Multiplication Factors.	138
Table B.9	IAEA Hex Cross Sections.	140
Table B.10	IAEA Hex Fission Spectrum.	140

Table B.11	MONJU Effective Neutron Multiplication Factors and Rod Worths.	141
Table B.12	MONJU Cross Sections.	143
Table B.13	MONJU Fission Spectrum.	143
Table B.14	KNK Effective Neutron Multiplication Factors and Rod Worths.	144
Table B.15	KNK Cross Sections (Part A).	147
Table B.16	KNK Cross Sections (Part B).	148
Table B.17	KNK Fission Spectrum.	149

LIST OF FIGURES

Figure 1.1	Example of Fast Reactor Materials based on MONJU.	3
Figure 1.2	Example of Fast Reactor Fuel Assembly Cross Section.	3
Figure 1.3	Dimensions of Thermal Hydraulic Rod Model (not to scale).	4
Figure 1.4	Dimensions of Hexagonal Can (not to scale).	5
Figure 2.1	Example of Rectangular Unstructured Mesh.	13
Figure 2.2	Description of Triangle Elements.	18
Figure 2.3	Description of Wedge Elements.	20
Figure 2.4	Description of Reference Wedge.	21
Figure 2.5	Typical Fast Flux Visualization for Two-Dimensional and Three-Dimensional Simulations.	29
Figure 2.6	Demonstration of RCM Matrix Ordering.	30
Figure 2.7	Example Unstructured Mesh.	32
Figure 3.1	Mesh Refinement of Curved Mesh.	42
Figure 3.2	VVER440 Benchmark Power Comparison for Most Refined Mesh.	44
Figure 3.3	SNR Benchmark Power Comparison for Most Refined Mesh.	45
Figure 3.4	HWR Benchmark Power Comparison for Most Refined Mesh.	47
Figure 4.1	Variable Thermal Conductivity in Fuel.	54
Figure 4.2	Progression of Element, to Chunk, to Channel.	56
Figure 4.3	One-Dimensional Axial Heat Convection Model Description.	56
Figure 4.4	Geometry Description of Radial Heat Conduction Model (not to scale).	59
Figure 4.5	Radial Temperatures for Typical Fuel Rod.	73
Figure 4.6	Difference Between Analytic and Modeled Axial Temperatures for 36 Axial Levels.	76
Figure 4.7	Average Axial Temperatures for Model Reactor Conditions.	77
Figure 5.1	Linear Expansion Factor for HT9 Steel and U10Zr Fuel.	80
Figure 5.2	Thermal Expansion of General Volume.	82
Figure 5.3	Effective Neutron Multiplication Factor as a Function of Thermal Expansion Temperature.	86
Figure 5.4	Reactivity as a Function of Thermal Expansion Temperature.	86
Figure 6.1	Materials in ABR.	89
Figure 6.2	Fast and Thermal Neutron Flux in ABR.	90
Figure 6.3	Feedback Effects on k_{eff}	94
Figure 6.4	Advanced Burner Reactor (ABR) Reactivity Coefficients.	95
Figure A.1	Fixed Source and Criticality Flux Shapes for One-Dimension, One-Group Problems.	108
Figure A.2	Two-Dimension Criticality Flux Shape.	113
Figure A.3	Example Two-Group Flux Plot.	117
Figure A.4	Geometry for Two-Region Problem.	119
Figure A.5	Two-Region Flux Shape.	123

Figure A.6	Example Finite Cylinder Flux Shape.	129
Figure A.7	Mesh Refinement of Curved Mesh.	129
Figure B.1	VVER440 Geometry.	132
Figure B.2	SNR Geometry.	134
Figure B.3	HWR Geometry.	137
Figure B.4	IAEA Hex Geometry.	139
Figure B.5	MONJU Geometry.	142
Figure B.6	MONJU Assembly Geometries.	142
Figure B.7	KNK Geometry.	145
Figure B.8	KNK Assembly Geometry.	145
Figure B.9	KNK Control Rod Geometry.	146

LIST OF ACRONYMS

ABR	Advanced Burner Reactor.
ANL	Argonne National Laboratory.
ATWS	Anticipated Transient Without Scram.
CASL	Consortium for Advanced Simulation of LWRs.
CG	Conjugate Gradient.
CRAM	Chebyshev Rational Approximation Method.
CTC	Coolant Temperature Coefficient.
DOE-NE	U.S. Department of Energy Office of Nuclear Energy.
EBR-I	Experimental Breeder Reactor I.
EBR-II	Experimental Breeder Reactor II.
FBR	Fast Breeder Reactor.
FEM	Finite Element Method.
GEH	GE-Hitachi Nuclear Energy Americas LLC.
HWR	Heavy Water Reactor.
INL	Idaho National Laboratory.
IUP	Integrated University Program.
LEF	Linear Expansion Factor.
LWR	Light Water Reactor.
MTC	Moderator Temperature Coefficient.
NEA	Nuclear Energy Agency.
OECD	Organisation for Economic Co-operation and Development.
ORNL	Oak Ridge National Laboratory.
PWR	Pressurized Water Reactor.
RCM	Reverse Cuthill-McKee.
RMS	Root-Mean-Squared.
SFR	Sodium-cooled Fast Reactor.
SOR	Successive Over-Relaxation.
SP_N	Simplified P_N .
SPD	Symmetric Positive Definite.
ULOF	Unprotected Loss-Of-Flow.
ULOHS	Unprotected Loss-Of-Heat-Sink.
VTR	Versatile Test Reactor.

This material is based upon work supported under an 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.

CHAPTER

1

INTRODUCTION

1.1 Motivation

Recent interest in advanced and next-generation nuclear power reactor designs has encouraged further development of modeling and simulation methods for these reactors. Fast reactors, a class of advanced reactors, operate with predominately high-energy (“fast”) neutrons in the fission reaction. Since early development of fast reactors, such as Experimental Breeder Reactor I (EBR-I) in 1951 and Fermi 1 in 1956, there have been significant innovations in both nuclear modeling and computational methods. As development of fast reactors is revisited in the form of the Versatile Test Reactor (VTR) at Idaho National Laboratory (INL), improvements in simulation can be used to simulate fast reactors with modern best practices.

Nuclear reactor simulations are inherently multiphysics simulations. For example, neutron reaction probabilities are described by cross sections. Neutron cross sections are dependent on material temperatures and densities, both of which vary over the operating range of a nuclear power reactor. As reactor power changes, material temperatures and densities change, therefore cross sections change and affect the reactor power. The multiphysics nature of the reactor necessitate a simulation of the power distribution within the reactor as well as all feedback effects which will be modeled.

In this thesis, models for simulating fast reactors will be developed and demonstrated. Reactor power distribution will be modeled according to the multigroup neutron diffusion equation as solved by the Finite Element Method (FEM) based on unstructured meshes with special attention to hexagonal geometries.

The multigroup neutron diffusion solution method is verified through comparison to both benchmark and analytic solutions. Multiphysics effects are modeled including thermal hydraulics and thermal expansion. Thermal hydraulic effects are modeled as axial heat convection and radial heat conduction. Thermal expansion is modeled using simplified linear expansion models. The methods developed in this work can easily be used for fast reactors with a variety of coolants including sodium, lead, or molten salt.

By employing a modern solution method to the neutron diffusion equation in the form of the FEM, the simulation can take advantage of developments in numerical methods including the solution of linear systems. Additionally, the simulation allows for the incorporation of generalized multiphysics effects whereas current state-of-the-art techniques (such as DIF3D) require data processing and manual iteration to simulate multiphysics effects. Ultimately, the simulation is designed to simulate an operating fast reactor and estimate feedback coefficients by coupling multiphysics models.

1.2 Geometry Description

The high-energy neutron spectra inherent to fast reactors results in relatively small neutron cross sections compared to larger cross sections in the thermal energy range. To compensate for this fact, fast reactors are typically designed with hexagonal, triangularly pitched, fuel assemblies to maximize fuel packing. An example of a fast reactor with hexagonal fuel assemblies is shown in Fig. 1.1.

A cross-sectional representation of a hexagonal fuel assembly is shown in Fig. 1.2. This geometry is used in the homogenization of neutron cross sections and is also used to describe coolant flow geometries. Dimensions of assemblies are measured at room temperature and will later be expanded according to the thermal expansion model in Chapter 5.

Note the individual rods in Fig. 1.2 are cylindrical and are arranged into a hexagonal assembly. The basic geometry is a metallic fuel material within stainless steel cladding. The gap between the fuel and cladding is filled by sodium bond to improve thermal conductivity across the gap. The rod is helically wrapped by a steel wire to ensure separation between rods that will allow for coolant flow. The wire wrap also serves to encourage the mixture of coolant within the assembly. (Note: wire wrap is omitted from Fig. 1.2.) Many rods are then assembled into an assembly and surrounded by a hexagonal can made of steel. This can aids in structural stability and prohibits cross-flow of coolant between assemblies.

The dimensions within a single rod are shown in Fig. 1.3 and the dimensions within a hexagonal assembly can are shown in Fig. 1.4. In Fig. 1.4, Th_{can} is the thickness of the assembly can, $F2F$ is the flat-to-flat measurement of the outside of the hexagonal can, and $Pitch$ is the distance between the center of two rods. Using the geometry described in these figures, the material cross-sectional areas are calculated according to the given formulae where N_{rod} is the number of rods in the assembly and AP is

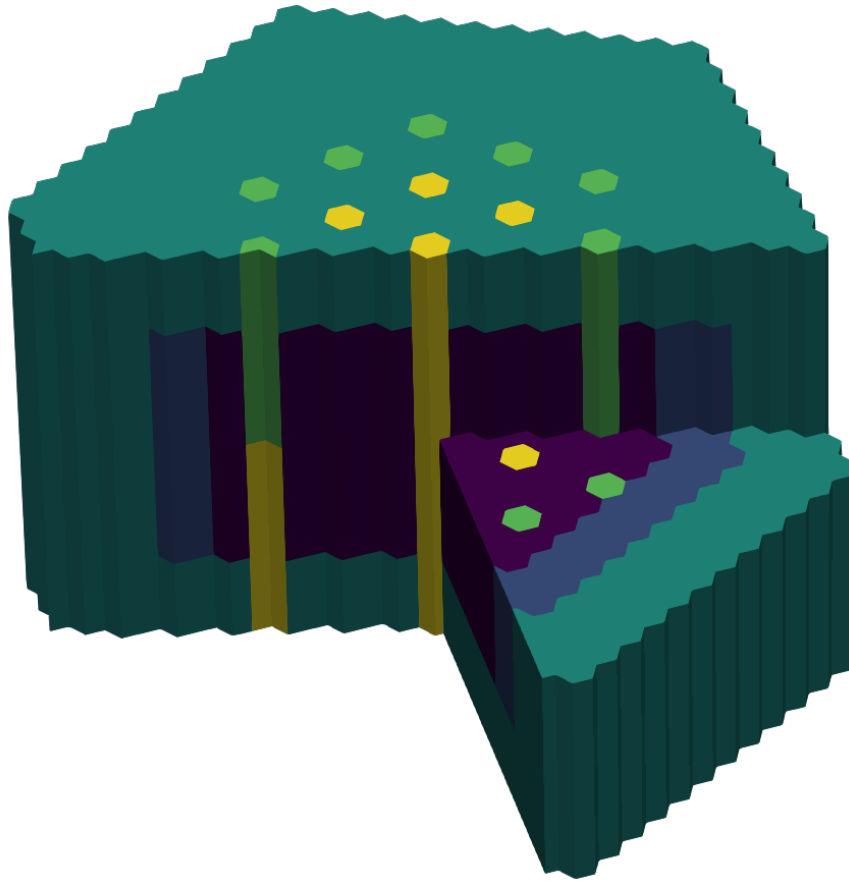


Figure 1.1 Example of Fast Reactor Materials based on MONJU.

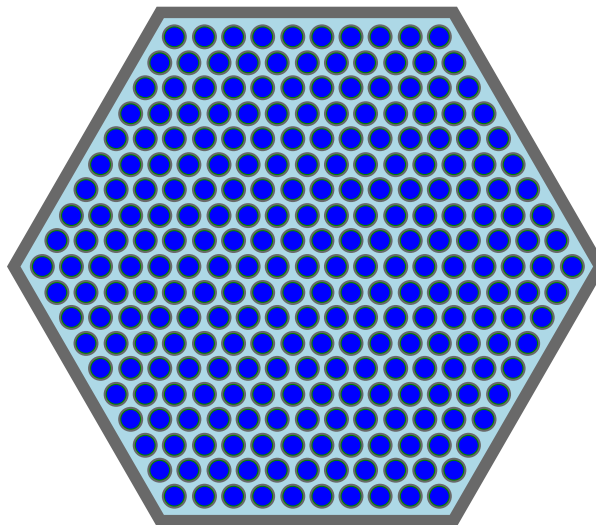


Figure 1.2 Example of Fast Reactor Fuel Assembly Cross Section.

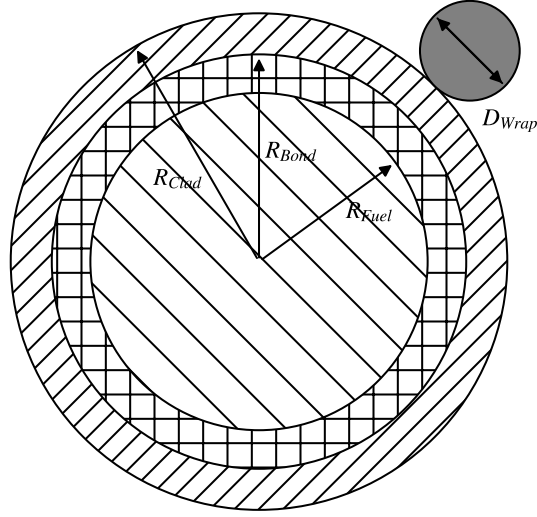


Figure 1.3 Dimensions of Thermal Hydraulic Rod Model (not to scale).

the assembly pitch. $AP > F2F$ to account for inter-assembly sodium gaps (see “Gap” in Fig. 1.4).

$$A_{total} = \frac{\sqrt{3}}{2} AP^2 \quad (1.1)$$

$$A_{box} = \frac{\sqrt{3}}{2} \left(F2F^2 - (F2F - 2Th_{can})^2 \right) \quad (1.2)$$

$$A_{wrap} = N_{rod} \frac{\pi}{4} D_{wrap}^2 \quad (1.3)$$

$$A_{clad} = N_{rod} \pi (R_C^2 - R_B^2) \quad (1.4)$$

$$A_{bond} = N_{rod} \pi (R_B^2 - R_F^2) \quad (1.5)$$

$$A_{fuel} = N_{rod} \pi R_F^2 \quad (1.6)$$

$$A_{cool} = A_{total} - A_{box} - A_{wrap} - A_{clad} - A_{bond} - A_{fuel} \quad (1.7)$$

$$A_{struct} = A_{box} + A_{wrap} + A_{clad} \quad (1.8)$$

Calculating the areas as above allows for calculation of cross-sectional area fractions. Assuming constant dimensions in the axial direction, these area fractions are equivalent to volume fractions and are useful for neutron cross section homogenization. Additionally, these formulae allow for thermal expansion as the liquid sodium in the bond and the liquid coolant are allowed to vary to allow for the expansion of other materials.

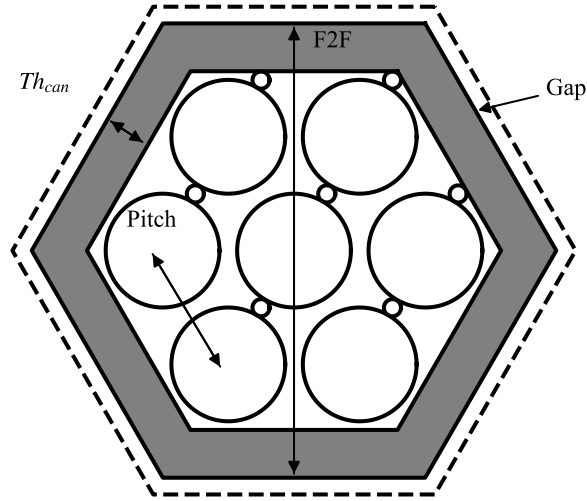


Figure 1.4 Dimensions of Hexagonal Can (not to scale).

1.3 Cross Section Treatment

Reactor materials are “smeared” into homogeneous regions. This treatment is common to fast reactors because of the relatively large neutron mean-free-paths compared to the scale of material dimensions. Additionally, the neutron distribution will be modeled using the neutron diffusion equation which cannot accurately resolve small geometric details. The natural choice for these homogeneous regions are the hexagonal assemblies themselves. Materials are permitted to be heterogeneous axially. For this work, four distinct regions are modeled within a hexagonal assembly: fuel, bond, coolant, and steel. Steel material includes cladding, wire wrap, and assembly can. These four regions are then homogenized into a hexagonal assembly.

For simplified analytic and benchmark problems, cross sections are specified by the problem. For realistic simulations, multigroup microscopic cross sections are generated using the computer program MC**2 [Lee12]. The cross section generator uses 2,082 fine energy groups to collapse down to an arbitrary number of coarse energy groups. For this simulation, the recommended and default 33-group energy structure is used. However, the methods in this work are implemented generally and are not dependent on a particular energy group structure. MC**2 solves the infinite-homogeneous (zero-dimension) neutron transport equation for isotopic number densities as input by the user. Cross sections for each assembly type are generated separately to accurately simulate the neutron energy spectrum within the assembly. This procedure results in a unique material cross sections for each assembly type. For example, each assembly type contains steel; therefore, there will be a separate steel cross section for each assembly type in the cross section library. Within MC**2, the neutron energy spectrum for fissile media is generated by

Table 1.1 Temperatures Selected for Cross Section Libraries.

Library	T_{cool} [K]	T_{clad} [K]	T_{fuel} [K]
1	628.15	628.15	628.15
2	708.65	757.50	807.15
3	896.87	920.47	961.46
4	1072.81	1114.83	1183.14

the media's fission spectrum. Non-fissile homogenized mixtures, such as control assemblies or reflector assemblies, the default ^{238}U fission spectrum is assumed.

Cross section libraries are generated for several different temperatures to capture temperature-dependent cross section effects. These libraries are then used during the simulation to calculate cross sections as a function of material temperatures. The fuel, clad, and coolant temperatures in a simulated reactor can be calculated with a thermal hydraulic model (see Chapter 4). However, the temperatures calculated in the thermal hydraulic model are functions of reactor power and coolant mass flow rate. These parameters are not known before the simulation for a general reactor. Instead, a simplified one-dimensional, single-channel model is used to estimate temperatures for cross section library generation. This model is based on the axial convection and radial conduction models in Chapter 4. The simulation model can use general cross section library temperatures and a general number of temperature libraries. A user may select the number of cross section libraries and specify the temperatures for which the libraries apply in a general manner. Typical cross section library temperatures for simulating liquid metal cooled, metal fueled, fast reactors are given in Table 1.1.

Note that the maximum temperatures in Table 1.1 are greater than temperatures observed at typical reactor operating conditions. This is necessary so that even at perturbed reactor conditions (e.g. 110% full power), the peak core temperatures can still be interpolated within the libraries.

Cross sections are homogenized within each hexagonal assembly using isotopic microscopic cross sections from MC**2 and user input number densities. Homogenization is performed in two steps: first, isotopic homogenization and second, volumetric homogenization. Isotopic homogenization is performed by summing microscopic cross sections and associated number densities. Let $\sigma_{i,j,x,g}$ represent the microscopic cross section for isotope i , in region j , for reaction type x , and energy group g as output by MC**2. $\sigma_{i,j,x,g}$ has units of area. Then, let $N_{i,j}$ represent the atom number density for isotope i in region j as input by the user. $N_{i,j}$ has units of inverse volume. The macroscopic cross section can then be defined

$$\Sigma_{j,x,g} = \sum_{i=1}^{N_{iso}} N_{i,j} \sigma_{i,j,x,g} \quad (1.9)$$

where $\Sigma_{j,x,g}$ is the macroscopic cross section in region j , for reaction type x , and energy group g . In Eq. (1.9), N_{iso} represents the number of isotopes in region j . Note that $\Sigma_{j,x,g}$ will have units of inverse length.

Next, volumetric cross section homogenization is performed using volume fractions. Assuming dimensions do not change axially within a hexagonal assembly, the areas calculated in Eq. (1.1) through Eq. (1.8) can be used to calculate area fractions. These area fractions can subsequently be treated as volume fractions. Using the definition of macroscopic cross sections from Eq. (1.9), the volumetrically homogenized macroscopic cross section is

$$\Sigma_{x,g} = \frac{\sum_{j=1}^{N_{reg}} \Sigma_{j,x,g} V_j}{\sum_{j=1}^{N_{reg}} V_j} \quad (1.10)$$

where V_j is the volume or area occupied by region j in the hexagonal assembly and N_{reg} is the number of regions in the hexagonal assembly. Typically, $N_{reg} = 4$ with unique regions for fuel, sodium bond, sodium coolant, and steel structural material. After homogenizing cross sections isotopically and volumetrically, the diffusion coefficient for energy group g , D_g , can be calculated as

$$D_g = \frac{1}{3\Sigma_{tr,g}} \quad (1.11)$$

where $\Sigma_{tr,g}$ represents the macroscopic transport cross section for energy group g . $\Sigma_{tr,g}$ has been homogenized according to Eq. (1.9) and Eq. (1.10). Note that D_g will have units of length.

1.4 Thesis Organization

In Chapter 2, the derivation of the FEM solution to the multigroup neutron diffusion equation is presented. Special attention is paid to triangular and wedge elements. The resulting eigenvalue problem is solved using the Power Method. Results from the diffusion solution are verified in two-dimensional and three-dimensional problems with both analytic and benchmark solutions. These verification problems for the neutron diffusion equation are presented in Chapter 3.

Chapter 4 presents the formulation of axial heat convection and radial heat conduction models for a typical fast reactor. These models are used to calculate material temperatures and update cross sections for the simulation. Results of the numerical model are compared to analytical models and example material temperatures are shown.

In Chapter 5 a simplified thermal expansion model is presented. The model assumes linear thermal expansion for given material properties and user-specified thermal expansion temperatures. A simple demonstration of the effects of thermal expansion on reactivity are presented.

The combination of all of these models allows for the realistic simulation of a fast reactor. In Chapter 6, the multiphysics models are coupled and investigated for a benchmark reactor problem. Using this benchmark reactor and the models described, multiphysics reactivity feedback coefficients are estimated.

Finally, Chapter 7 presents a summary and the conclusions of this research. Additionally, recommendations for further research are provided.

CHAPTER

2

FINITE ELEMENT NEUTRON DIFFUSION

2.1 Introduction

This chapter will describe the solution of the multigroup neutron diffusion equations for general geometry via the Finite Element Method (FEM). The solution method and derivations here are general to the multigroup neutron diffusion equation and its application to any standard reactor geometry is straightforward. For typical fast reactor applications, diffusion theory approximates the neutron distribution within the reactor well. The diffusion approximation is a standard assumption for fast reactors because neutron mean-free-paths within the reactor are large relative to material dimensions.

Spatial discretization will be done with the FEM. This spatial discretization method is selected for several reasons. It allows for easily increasing the spatial convergence order of the method by increasing the order of the elements without refining the mesh. For example, with a given mesh, quadratic elements instead of linear elements could be used to spatially refine the solution. Additionally, coordinates of nodes and elements can be easily updated to reflect physical phenomena, such as thermal expansion (see Chapter 5). Finally, material properties are calculated on an element basis allowing for detailed updates to the material properties during the calculation.

2.2 Multigroup Neutron Diffusion Equation

In the multigroup neutron diffusion equation, an energy structure is described by the set of energies $\{E_g\}$ for $g = 1, 2, \dots, G$. By convention, the energy groups are arranged in order of decreasing energy.

$$0 < E_G < E_{G-1} < \dots < E_2 < E_1 \quad (2.1)$$

Multigroup neutron cross sections can be calculated using this energy group structure from energy dependent cross sections, and a representative flux spectrum. Generation of multigroup neutron cross sections is performed using MC**2 and is described in §1.3.

In conventional notation, the multigroup neutron diffusion equation can be written as

$$-\nabla \cdot (D_g(\mathbf{r}) \nabla \phi_g(\mathbf{r})) + \Sigma_{t,g}(\mathbf{r}) \phi_g(\mathbf{r}) = \frac{\widetilde{\chi}_g(\mathbf{r})}{k_{eff}} \sum_{g'=1}^G \nu \Sigma_{f,g'}(\mathbf{r}) \phi_{g'}(\mathbf{r}) + \sum_{g'=1}^G \Sigma_{s,g' \rightarrow g}(\mathbf{r}) \phi_{g'}(\mathbf{r}) \quad (2.2)$$

where

- \mathbf{r} = spatial position vector,
- $D_g(\mathbf{r})$ = diffusion coefficient for energy group g [cm],
- $\phi_g(\mathbf{r})$ = scalar neutron flux for energy group g $\left[\frac{1}{\text{cm}^2 \text{ s}}\right]$,
- $\Sigma_{t,g}(\mathbf{r})$ = macroscopic total cross section for energy group g $\left[\frac{1}{\text{cm}}\right]$,
- $\widetilde{\chi}_g(\mathbf{r})$ = effective fission spectrum for energy group g ,
- k_{eff} = effective neutron multiplication factor,
- $\nu \Sigma_{f,g}(\mathbf{r})$ = number of fission neutrons times macroscopic fission cross section in energy group g $\left[\frac{1}{\text{cm}}\right]$,
- $\Sigma_{s,g' \rightarrow g}(\mathbf{r})$ = macroscopic scatter cross section from energy group g' to energy group g $\left[\frac{1}{\text{cm}}\right]$,
- G = total number of energy groups.

The total neutron cross section includes the contribution due to within-group scattering; that is, due to $\Sigma_{s,g \rightarrow g}$. This can be subtracted from both sides of Eq. (2.2) for simplicity and numeric efficiency. Rewriting Eq. (2.2) with this modification yields

$$-\nabla \cdot (D_g(\mathbf{r}) \nabla \phi_g(\mathbf{r})) + \Sigma_{r,g}(\mathbf{r}) \phi_g(\mathbf{r}) = \frac{\widetilde{\chi}_g(\mathbf{r})}{k_{eff}} \sum_{g'=1}^G \nu \Sigma_{f,g'}(\mathbf{r}) \phi_{g'}(\mathbf{r}) + \sum_{\substack{g'=1 \\ g' \neq g}}^G \Sigma_{s,g' \rightarrow g}(\mathbf{r}) \phi_{g'}(\mathbf{r}) \quad (2.3)$$

where $\Sigma_{r,g}$ is the removal cross section defined as $\Sigma_{r,g}(\mathbf{r}) = \Sigma_{t,g}(\mathbf{r}) - \Sigma_{s,g \rightarrow g}(\mathbf{r})$. The removal cross section now describes the removal of neutrons from the element of phase space due to nuclear interactions.

For simplicity, the neutron sources in Eq. (2.3) can be combined into a single term as

$$-\nabla \cdot (D_g(\mathbf{r})\nabla\phi_g(\mathbf{r})) + \Sigma_{r,g}(\mathbf{r})\phi_g(\mathbf{r}) = q_g(\mathbf{r}) \quad (2.4)$$

where $q_g(\mathbf{r})$ is the combined neutron source at position \mathbf{r} for energy group g and is expressed as

$$q_g(\mathbf{r}) = q_{fiss,g}(\mathbf{r}) + q_{up,g}(\mathbf{r}) + q_{down,g}(\mathbf{r}) \quad (2.5)$$

with contributing terms

$$q_{fiss,g}(\mathbf{r}) = \frac{\widetilde{\chi}_g(\mathbf{r})}{k_{eff}} \sum_{g'=1}^G \nu\Sigma_{f,g'}(\mathbf{r})\phi_{g'}(\mathbf{r}), \quad (2.6)$$

$$q_{up,g}(\mathbf{r}) = \sum_{g'=g+1}^G \Sigma_{s,g' \rightarrow g}(\mathbf{r})\phi_{g'}(\mathbf{r}), \quad (2.7)$$

$$q_{down,g}(\mathbf{r}) = \sum_{g'=1}^{g-1} \Sigma_{s,g' \rightarrow g}(\mathbf{r})\phi_{g'}(\mathbf{r}), \quad (2.8)$$

where the difference between q_{up} and q_{down} are the limits of the summation. q_{up} represents the neutron source due to scattering from lower energy groups (up-scattering) and q_{down} represents the neutron source due to scattering from higher energy groups (down-scattering). This form allows for operator splitting of the neutron source term. In an iterative scheme, it will be necessary for fission and up-scatter sources to use a different flux iterate than down-scatter so this separation of sources will prove useful (see §2.4.2).

The combined source form is useful for solving the multigroup neutron diffusion problem for an arbitrary number of groups. Eq. (2.4) is solved for each energy group and interaction between groups is described in the source term, $q_g(\mathbf{r})$. In other literature, the multigroup equation may be solved for all groups simultaneously by treating interaction between groups explicitly. By solving each group independently (as done here) the method remains general. Additionally, for many-group energy structures, as common to fast reactor applications, solving each group independently is typically more computationally efficient as such linear systems have favorable conditioning and are dimensionally smaller. Finally, fast reactors are also dominated by down-scatter as opposed to thermal reactors which experience significant up-scatter at thermal energies. This fact implies that the term $q_{up,g}(\mathbf{r})$ will not have to be updated frequently in fast reactor simulations and the one group at-a-time solution method will benefit.

Typically, reactor materials are described isotopically and χ may be specified isotopically. However, for calculating the fission neutron source $q_{fiss,g}(\mathbf{r})$ as in Eq. (2.6), an effective $\widetilde{\chi}$ is needed. From an

isotopic description, $q_{fiss,g}(\mathbf{r})$ is given as

$$q_{fiss,g}(\mathbf{r}) = \sum_{i=1}^{N_{iso}} \chi_{i,g}(\mathbf{r}) \left(\sum_{g'=1}^G \nu \Sigma_{f,i,g'}(\mathbf{r}) \phi_{g'}(\mathbf{r}) \right) \quad (2.9)$$

where $\chi_{i,g}(\mathbf{r})$ is the isotopic fission spectrum and N_{iso} is the number of isotopes at position \mathbf{r} . Next, require $q_{fiss,g}(\mathbf{r})$ to have the form of Eq. (2.10).

$$q_{fiss,g}(\mathbf{r}) = \widetilde{\chi}_g(\mathbf{r}) \sum_{i=1}^{N_{iso}} \sum_{g'=1}^G \nu \Sigma_{f,i,g'}(\mathbf{r}) \phi_{g'}(\mathbf{r}) \quad (2.10)$$

Setting Eq. (2.9) equal to Eq. (2.10) yields the expression for $\widetilde{\chi}_g(\mathbf{r})$ based on isotopic data.

$$\widetilde{\chi}_g(\mathbf{r}) = \frac{\sum_{i=1}^{N_{iso}} \chi_{i,g}(\mathbf{r}) \left(\sum_{g'=1}^G \nu \Sigma_{f,i,g'}(\mathbf{r}) \phi_{g'}(\mathbf{r}) \right)}{\sum_{i=1}^{N_{iso}} \sum_{g'=1}^G \nu \Sigma_{f,i,g'}(\mathbf{r}) \phi_{g'}(\mathbf{r})} \quad (2.11)$$

Note that Eq. (2.11) requires the solution $\phi_g(\mathbf{r})$. Ultimately, the flux is unknown but will be solved in an iterative manner. Eq. (2.11) implies that $\widetilde{\chi}_g(\mathbf{r})$ must be updated for each iteration of the solution (see Step 9 in Algorithm 2.2).

2.3 Formulation of Finite Element Equations

This section presents the derivation of the spatial discretization of the multigroup neutron diffusion equation based on the FEM. The method results in a linear system of equations for a fixed source iteration. Details are also provided on constructing the finite element matrix for use with triangular and wedge elements.

2.3.1 Derivation

The remaining continuous variable in the problem to be discretized in Eq. (2.4) is the spatial variable \mathbf{r} . This will be discretized according to the FEM. The problem is solved in a finite domain $\mathbf{r} \in \Omega$ where $\partial\Omega$ represents the boundary of the domain and some boundary condition is specified. Boundary condition options include:

1. Mirror. $\nabla \phi_g(\mathbf{r}) \cdot \hat{\mathbf{n}} = 0$ for $\mathbf{r} \in \partial\Omega$.
2. Albedo. $D_g(\mathbf{r}) \nabla \phi_g(\mathbf{r}) \cdot \hat{\mathbf{n}} + \alpha \phi_g(\mathbf{r}) = 0$ for $\mathbf{r} \in \partial\Omega$, where $\alpha \in \mathbb{R}$ is a scalar constant specified by the user. For non-reentrant boundary condition, $\alpha = \frac{1}{2}$.
3. Zero Flux. $\phi_g(\mathbf{r}) = 0$ for $\mathbf{r} \in \partial\Omega$.

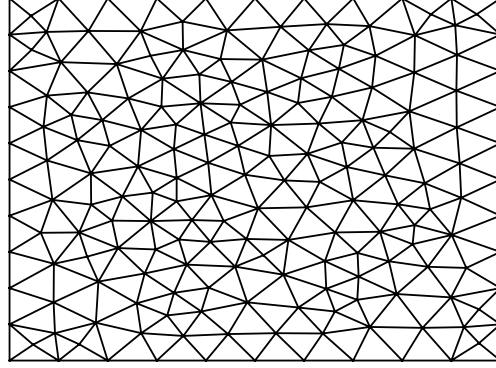


Figure 2.1 Example of Rectangular Unstructured Mesh.

$\hat{\mathbf{n}}$ represents the unit outward normal vector at the boundary $\partial\Omega$. (Note: the order of the above list corresponds to the order of boundary condition precedent in code with the greater the integer, the greater the precedent.)

Begin by partitioning the spatial domain Ω into a set of finite elements.

$$\Omega = \Omega_1 \cup \Omega_2 \cup \Omega_3 \cup \dots \cup \Omega_{N_E} \quad (2.12)$$

such that $\Omega = \{\Omega_e\}$ for $e = 1, 2, \dots, N_E$ is a set of non-overlapping elements

$$\Omega_i \cap \Omega_j = \emptyset, \quad \text{for } i \neq j, \quad (2.13)$$

and N_E is the total number of elements. Elements are in an unstructured mesh and can be generated by any number of mesh generation methods (e.g. Delaunay triangulation) to describe the geometry of the problem. An example of an unstructured mesh generated for a rectangular domain is shown in Fig. 2.1.

Proceeding with the Galerkin FEM, Eq. (2.4) is multiplied by a testing function $v(\mathbf{r}) \in H_1(\Omega)$ where H is the Sobolev space.

$$-\nabla \cdot (D_g(\mathbf{r}) \nabla \phi_g(\mathbf{r})) v(\mathbf{r}) + \Sigma_{r,g}(\mathbf{r}) \phi_g(\mathbf{r}) v(\mathbf{r}) = q_g(\mathbf{r}) v(\mathbf{r}) \quad (2.14)$$

Then, Eq. (2.14) is integrated over the problem domain. This integration yields the Weak Form or Variational Form of the problem.

$$-\int_{\Omega} \nabla \cdot (D_g(\mathbf{r}) \nabla \phi_g(\mathbf{r})) v(\mathbf{r}) \, d\mathbf{r} + \int_{\Omega} \Sigma_{r,g}(\mathbf{r}) \phi_g(\mathbf{r}) v(\mathbf{r}) \, d\mathbf{r} = \int_{\Omega} q_g(\mathbf{r}) v(\mathbf{r}) \, d\mathbf{r} \quad (2.15)$$

For the purposes of this work, material cross sections and the neutron source, $q_{g,e}$, are assumed to be constant within an element. In the future, q_g could be considered discrete at each node rather than

each element. To calculate a constant neutron source within an element, Eq. (2.5) is used to calculate the average neutron source, $q_{g,e}$, in an element.

$$q_{g,e} = q_{fiss,g,e} + q_{up,g,e} + q_{down,g,e} \quad (2.16)$$

$$\widetilde{\chi}_{g,e} = \frac{\sum_{i=1}^{N_{iso}} \chi_{i,g,e} \left(\sum_{g'=1}^G \nu \Sigma_{f,i,g',e} \bar{\phi}_{g',e} \right)}{\sum_{g'=1}^G \sum_{i=1}^{N_{iso}} \nu \Sigma_{f,i,g',e} \bar{\phi}_{g',e}} \quad (2.17)$$

$$q_{fiss,g,e} = \frac{\widetilde{\chi}_{g,e}}{k_{eff}} \sum_{g'=1}^G \nu \Sigma_{f,g',e} \bar{\phi}_{g',e} \quad (2.18)$$

$$q_{up,g,e} = \sum_{g'=g+1}^G \Sigma_{s,g' \rightarrow g,e} \bar{\phi}_{g',e} \quad (2.19)$$

$$q_{down,g,e} = \sum_{g'=1}^{g-1} \Sigma_{s,g' \rightarrow g,e} \bar{\phi}_{g',e} \quad (2.20)$$

Note that in Eq. (2.17), $\widetilde{\chi}_{g,e}$ must now be calculated for each finite element. For first-order, linear implementations of the FEM, the element-average flux $\bar{\phi}_{g,e}$ is

$$\bar{\phi}_{g,e} = \frac{1}{N_p} \sum_{i \in \Omega_e}^{N_p} \phi_{i,g} \quad (2.21)$$

where N_p is the number of nodes on the element and $i \in \Omega_e$ is the summation over all nodes in element Ω_e . For example, a triangle has $N_p = 3$ and a wedge has $N_p = 6$.

Given constant material properties and constant neutron source over the element, the integrals in Eq. (2.15) can be partitioned into a sum of integrals over the elements in the domain assuming the non-overlapping set of elements from Eq. (2.12) and Eq. (2.13).

$$- \sum_{e=1}^{N_E} D_{g,e} \int_{\Omega_e} \nabla \cdot \nabla \phi_g(\mathbf{r}) v(\mathbf{r}) d\mathbf{r} + \sum_{e=1}^{N_E} \Sigma_{r,g,e} \int_{\Omega_e} \phi_g(\mathbf{r}) v(\mathbf{r}) d\mathbf{r} = \sum_{e=1}^{N_E} q_{g,e} \int_{\Omega_e} v(\mathbf{r}) d\mathbf{r} \quad (2.22)$$

The Second Green's Theorem is used to rewrite the integral in the first term. A proof invoking the Second Green's Theorem has been published by Li et al. [Li18]. The Second Green's Theorem is

$$- \int_{\Omega_e} \nabla \cdot \nabla \phi_g(\mathbf{r}) v(\mathbf{r}) d\mathbf{r} = - \int_{\partial\Omega_e} (\nabla \phi_g(\mathbf{r}) \cdot \hat{\mathbf{n}}) v(\mathbf{r}) ds + \int_{\Omega_e} \nabla \phi_g(\mathbf{r}) \cdot \nabla v(\mathbf{r}) d\mathbf{r} \quad (2.23)$$

where $\nabla \phi_g(\mathbf{r}) \cdot \hat{\mathbf{n}}$ is the outward normal derivative and the integral $\int_{\partial\Omega} \cdot ds$ is a line integral in two dimensions or a surface integral in three dimensions. Recognizing that this quantity will only be relevant on the boundary of the problem, the value of the outward normal derivative will be specified as a boundary

condition. Specifically, the albedo boundary condition which has the form

$$D_g(\mathbf{r})\nabla\phi_g(\mathbf{r}) \cdot \hat{\mathbf{n}} + \alpha\phi_g(\mathbf{r}) = 0 \quad (2.24)$$

$$D_g(\mathbf{r})\nabla\phi_g(\mathbf{r}) \cdot \hat{\mathbf{n}} = -\alpha\phi_g(\mathbf{r}) \quad (2.25)$$

for $\mathbf{r} \in \partial\Omega$. Note that all allowed boundary conditions (mirror, albedo, and zero-flux) can be specified as an albedo condition. For mirror boundaries, $\alpha = 0$ and for zero-flux boundaries, $\alpha \rightarrow \infty$. Substituting Eq. (2.23) into Eq. (2.22).

$$\begin{aligned} - \sum_{e=1}^{N_E} D_{g,e} \int_{\partial\Omega_e} v(\mathbf{r}) \nabla\phi_g(\mathbf{r}) \cdot \hat{\mathbf{n}} \, ds + \sum_{e=1}^{N_E} D_{g,e} \int_{\Omega_e} \nabla\phi_g(\mathbf{r}) \cdot \nabla v(\mathbf{r}) \, d\mathbf{r} + \\ \sum_{e=1}^{N_E} \Sigma_{r,g,e} \int_{\Omega_e} \phi_g(\mathbf{r}) v(\mathbf{r}) \, d\mathbf{r} = \sum_{e=1}^{N_E} q_{g,e} \int_{\Omega_e} v(\mathbf{r}) \, d\mathbf{r} \end{aligned} \quad (2.26)$$

Assuming the outward normal derivative is specified in the form of an albedo boundary condition with α constant throughout the problem boundary as in Eq. (2.25).

$$\begin{aligned} \sum_{e=1}^{N_E} \alpha \int_{\partial\Omega_e} v(\mathbf{r}) \phi_g(\mathbf{r}) \, ds + \sum_{e=1}^{N_E} D_{g,e} \int_{\Omega_e} \nabla\phi_g(\mathbf{r}) \cdot \nabla v(\mathbf{r}) \, d\mathbf{r} + \\ \sum_{e=1}^{N_E} \Sigma_{r,g,e} \int_{\Omega_e} \phi_g(\mathbf{r}) v(\mathbf{r}) \, d\mathbf{r} = \sum_{e=1}^{N_E} q_{g,e} \int_{\Omega_e} v(\mathbf{r}) \, d\mathbf{r} \end{aligned} \quad (2.27)$$

Next, for the Galerkin formulation of the FEM, the function of interest $\phi_g(\mathbf{r})$ is assumed to be a linear combination of chosen basis functions, N_i , as

$$\phi_g(\mathbf{r}) = \sum_{i=1}^{DOF} v_{i,g} N_i(\mathbf{r}) \quad (2.28)$$

where coefficients $\mathbf{u}_g = \{v_{i,g}\}$ are unknown and will be determined and DOF is the total number degrees of freedom of the problem. Typically DOF is the number of nodes less any nodes for which the flux is fixed (e.g. zero-flux nodes).

Typically, basis functions have unit magnitude and are centered at the node points so the coefficients $v_{i,g}$ are the FEM solution at the nodes. It is also convenient for basis functions to have compact support. That is, basis functions are created such that they are zero almost everywhere except some minimal region. Compact support in this implementation is chosen such that basis functions have unit value on a single mesh node and are zero on all other mesh nodes. Basis functions are typically piecewise continuous polynomials of arbitrary degree. For selected elements, basis functions will be defined explicitly in

§2.3.2. Linear and quadratic polynomials are common; but, for the work presented here, only linear basis functions are explored.

The test function, $v(\mathbf{r}) \in H_1(\Omega)$, is also chosen as a linear combination of the same basis functions.

$$v(\mathbf{r}) = \sum_{j=1}^{DOF} N_j(\mathbf{r}) \quad (2.29)$$

The magnitude of the testing function is arbitrary so the magnitude is set to unity.

Eq. (2.28) and Eq. (2.29) are inserted into Eq. (2.27).

$$\begin{aligned} \sum_{e=1}^{N_E} \alpha \sum_{i=1}^{DOF} v_{i,g} \int_{\partial\Omega_e} N_i(\mathbf{r}) N_j(\mathbf{r}) ds + \sum_{e=1}^{N_E} D_{g,e} \sum_{i=1}^{DOF} v_{i,g} \int_{\Omega_e} \nabla N_i(\mathbf{r}) \cdot \nabla N_j(\mathbf{r}) d\mathbf{r} + \\ \sum_{e=1}^{N_E} \Sigma_{r,g,e} \sum_{i=1}^{DOF} v_{i,g} \int_{\Omega_e} N_i(\mathbf{r}) N_j(\mathbf{r}) d\mathbf{r} = \sum_{e=1}^{N_E} q_{g,e} \sum_{i=1}^{DOF} \int_{\Omega_e} N_i(\mathbf{r}) d\mathbf{r} \end{aligned} \quad (2.30)$$

Eq. (2.30) can be rearranged as a linear system of equations.

$$\begin{aligned} \sum_{i=1}^{DOF} v_{i,g} \sum_{j=1}^{DOF} \left(\sum_{e=1}^{N_E} \alpha \int_{\partial\Omega_e} N_i(\mathbf{r}) N_j(\mathbf{r}) ds + \sum_{e=1}^{N_E} D_{g,e} \int_{\Omega_e} \nabla N_i(\mathbf{r}) \cdot \nabla N_j(\mathbf{r}) d\mathbf{r} + \right. \\ \left. \sum_{e=1}^{N_E} \Sigma_{r,g,e} \int_{\Omega_e} N_i(\mathbf{r}) N_j(\mathbf{r}) d\mathbf{r} \right) = \sum_{i=1}^{DOF} \left(\sum_{e=1}^{N_E} q_{g,e} \int_{\Omega_e} N_i(\mathbf{r}) d\mathbf{r} \right) \end{aligned} \quad (2.31)$$

Which can be written in the notation common to the mathematical discussions of the FEM

$$a_g(N_i, N_j) = f_g(N_i) \quad (2.32)$$

where $a_g(N_i, N_j)$ is the bilinear form of the FEM for group g and $f_g(N_i)$ is the linear form of the FEM for group g . Eq. (2.31) can also be written in matrix format as

$$\mathbf{A}_g \mathbf{u}_g = \mathbf{f}_g \quad (2.33)$$

where $\mathbf{u}_g = \{v_{i,g}\}$.

The diffusion coefficient $D_g(\mathbf{r})$ is non-zero and bounded and the removal cross section $\Sigma_{r,g}(\mathbf{r})$ is bounded. Given these conditions, the Lax-Milgram Lemma implies the solution to the FEM equations as derived here is both unique and bounded [Li18]. This is not the entire solution description as the source function $q_g(\mathbf{r})$ is updated on each power iteration (see §2.4). What the satisfaction of the Lax-Milgram Lemma does imply is, for a fixed source problem in a given power iteration, a unique and bounded solution exists. The multigroup neutron diffusion problem remains an eigenvalue problem.

In the matrix notation of Eq. (2.33), matrix \mathbf{A}_g is described by integral quantities, vector \mathbf{u}_g is the unknown magnitudes of the basis functions $\{\nu_{i,g}\}$, and vector \mathbf{f}_g is described by source integral quantities. Inspecting the finite elements matrix \mathbf{A}_g and the vector \mathbf{f}_g for element e reveals

$$A_{i,j,g,e} = \alpha \int_{\partial\Omega_e} N_i(\mathbf{r})N_j(\mathbf{r}) \, ds + D_{g,e} \int_{\Omega_e} \nabla N_i(\mathbf{r}) \cdot \nabla N_j(\mathbf{r}) \, d\Omega_e + \Sigma_{r,g,e} \int_{\Omega_e} N_i(\mathbf{r})N_j(\mathbf{r}) \, d\Omega_e, \quad (2.34)$$

$$f_{i,g,e} = q_{g,e} \int_{\Omega_e} N_i(\mathbf{r}) \, d\Omega_e. \quad (2.35)$$

Then, all element data can be combined as

$$A_{i,j,g} = \sum_{e=1}^{N_E} A_{i,j,g,e}, \quad (2.36)$$

$$f_{i,g} = \sum_{e=1}^{N_E} f_{i,g,e}, \quad (2.37)$$

which leads to the natural population of the matrix \mathbf{A}_g in an element-by-element procedure. Matrix \mathbf{A}_g is assembled by looping through all elements and summing their contribution to the matrix. Note that the contribution due to the surface integral will be zero in elements not on the problem boundary and may also be zero for problems with select boundary conditions. See §2.5.3 for boundary condition discussion. The population of the vector \mathbf{f}_g is done similarly in an element-by-element fashion. Then, the matrix \mathbf{A}_g and the vector \mathbf{f}_g are known for each energy group. The equations are solved for one energy group at a time and ϕ_g is calculated and stored.

The above derivation reduces to a linear system of equations. These equations are constructed from the integral quantities specified by the FEM and the coefficients given by the cross sections. The integral quantities themselves are expressed explicitly in the §2.3.2.

2.3.2 Matrix Quantities

For selected simple elements, the integral quantities described in Eq. (2.34) and Eq. (2.35) have exact analytic forms. For this work, linear triangles and linear wedges are investigated and many of the integrals have exact expressions. If these quantities cannot be expressed exactly, or doing so would be computationally inefficient, numerical quadratures are used. Given proper selection of the quadrature set, a quadrature rule can express the integrals exactly. This will be discussed in §2.3.3.

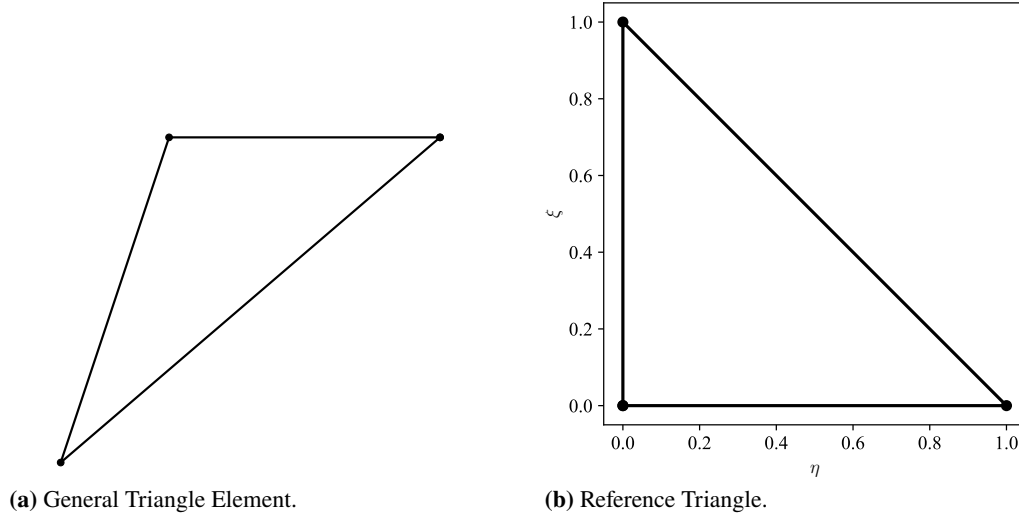


Figure 2.2 Description of Triangle Elements.

2.3.2.1 Linear Triangles

Linear triangles are common to two-dimensional FEMs and have been investigated in many methods for the solution of the few-group neutron diffusion equation [Hos17; Hos13; Hos15]. The linear triangle element is a triangle defined by three corner coordinates with basis functions located on each corner.

It is difficult to analytically calculate the desired integral quantities for an arbitrary triangle. Instead, a simplified reference element is created and quantities are calculated for the reference element and then translated to the arbitrary element using a Jacobian. The reference triangle T_{ref} is located in $\xi \in [0, 1]$ and $\eta \in [0, 1 - \xi]$. General and reference triangles are shown in Fig. 2.2. The basis functions are zero outside of the reference triangle. Within the reference triangle, basis functions for the triangle are provided in the natural coordinates of T_{ref} .

$$N_i(\xi, \eta) = 0 \quad \forall (\xi, \eta) \notin T_{ref} \quad (2.38)$$

$$N_1(\xi, \eta) = \xi \quad (2.39)$$

$$N_2(\xi, \eta) = \eta \quad (2.40)$$

$$N_3(\xi, \eta) = 1 - \xi - \eta \quad (2.41)$$

For linear triangles, simple expressions for the integral quantities for an arbitrary triangle can be derived [Whi85]. The expression for the line integral for the arbitrary element can also be derived [Joh09].

For a general triangle with corners (x_i, y_i) with $i = 1, 2, 3$

$$\int_{\Omega_e} \nabla N_i(\mathbf{r}) \cdot \nabla N_j(\mathbf{r}) d\mathbf{r} = \frac{1}{4A_e} ((x_{i+1} - x_{i+2})(x_{j+1} - x_{j+2}) + (y_{i+1} - y_{i+2})(y_{j+1} - y_{j+2})), \quad (2.42)$$

$$\int_{\Omega_e} N_i(\mathbf{r}) N_j(\mathbf{r}) d\mathbf{r} = \frac{A_e}{12} (1 + \delta_{ij}), \quad (2.43)$$

$$\int_{\Omega_e} N_i(\mathbf{r}) d\mathbf{r} = \frac{A_e}{3}, \quad (2.44)$$

$$\int_{\partial\Omega_e} N_i(\mathbf{r}) N_j(\mathbf{r}) ds = \frac{L_e}{6} (1 + \delta_{ij}), \quad (2.45)$$

where A_e is the area of the triangular element, L_e is the length of the edge between node i and node j , and δ_{ij} is the Kronecker delta. The Kronecker delta is defined as

$$\delta_{ij} = \begin{cases} 0 & \text{if } i \neq j, \\ 1 & \text{if } i = j. \end{cases} \quad (2.46)$$

The area of a triangle in three dimensions is calculated for a triangle with corner coordinates $\mathbf{c}_i = (x_i, y_i, z_i)$ with $i = 1, 2, 3$. That is, \mathbf{c}_i is the coordinates of corner i . Calculation of the area of a general triangle is then given by the vector operations

$$\mathbf{a} = \mathbf{c}_2 - \mathbf{c}_1, \quad (2.47)$$

$$\mathbf{b} = \mathbf{c}_3 - \mathbf{c}_1, \quad (2.48)$$

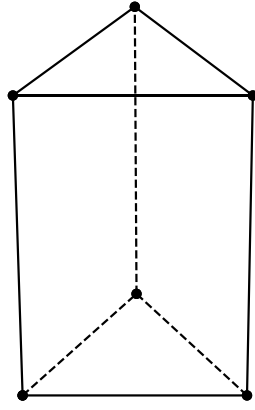
$$A_e = \frac{1}{2} |\mathbf{a} \times \mathbf{b}|, \quad (2.49)$$

$$A_e = \frac{1}{2} \sqrt{(a_2b_3 - a_3b_2)^2 + (a_3b_1 - a_1b_3)^2 + (a_1b_2 - a_2b_1)^2}, \quad (2.50)$$

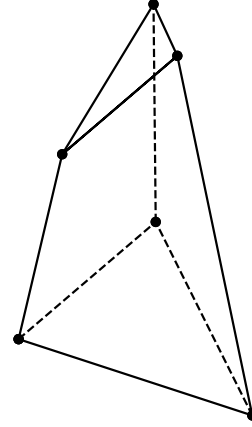
where a_i is the i^{th} component of vector \mathbf{a} and b_i is the i^{th} component of vector \mathbf{b} . For higher order triangular elements (e.g. quadratic or cubic elements), it may be necessary to employ a quadrature to calculate the necessary integral quantities.

2.3.2.2 Linear Wedges

A wedge element is a pentahedron with six corner nodes, and is sometimes referred to as a triangular prism. A simple example of a wedge is an extruded triangle. However, unlike an extruded triangle, the exact geometric relation of corner nodes in a wedge is not fixed and the nodes are free to expand and distort. An example of typical and distorted wedge elements are shown in Fig. 2.3. These elements are unique because there are two different types of faces. Three faces are quadrilateral and two are triangular.



(a) General Wedge Element.



(b) Distorted Wedge Element.

Figure 2.3 Description of Wedge Elements.

Fast reactors typically have hexagonal-z geometry so wedge elements are a natural choice for this coordinate system. Reactor geometries are also typically described in lattices so the wedge element allows for easily “stacking” lattices on top of each other.

The reference wedge W_{ref} is located in $\xi \in [0, 1]$, $\eta \in [0, 1 - \xi]$, and $\zeta \in [-1, 1]$. The coordinate system of the reference wedge is shown in Fig. 2.4. The basis functions are zero outside of the reference wedge and are provided within the reference wedge.

$$N_i(\xi, \eta, \zeta) = 0 \quad \forall (\xi, \eta, \zeta) \notin W_{ref} \quad (2.51)$$

$$N_1(\xi, \eta, \zeta) = \frac{1}{2}(1 - \zeta)(1 - \xi - \eta) \quad (2.52)$$

$$N_2(\xi, \eta, \zeta) = \frac{1}{2}(1 - \zeta)\xi \quad (2.53)$$

$$N_3(\xi, \eta, \zeta) = \frac{1}{2}(1 - \zeta)\eta \quad (2.54)$$

$$N_4(\xi, \eta, \zeta) = \frac{1}{2}(1 + \zeta)(1 - \xi - \eta) \quad (2.55)$$

$$N_5(\xi, \eta, \zeta) = \frac{1}{2}(1 + \zeta)\xi \quad (2.56)$$

$$N_6(\xi, \eta, \zeta) = \frac{1}{2}(1 + \zeta)\eta \quad (2.57)$$

The integrals of the basis function over the element are given in Eq. (2.58) through Eq. (2.60). The values presented herein are not found in literature and have been calculated by the author. For a general

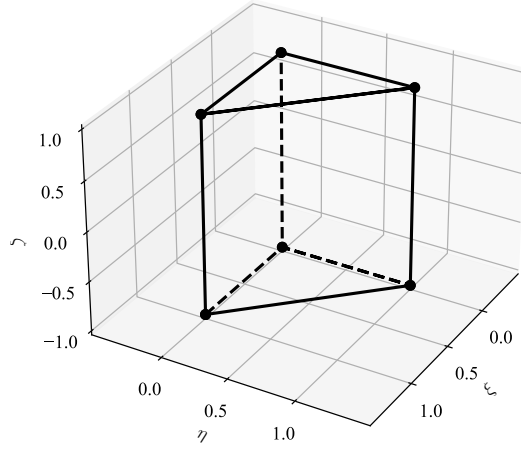


Figure 2.4 Description of Reference Wedge.

wedge, the integral quantities are

$$\int_{\Omega_e} N_i(\mathbf{r}) N_j(\mathbf{r}) d\mathbf{r} = \frac{V_e}{144} \begin{pmatrix} 4 & 2 & 2 & 2 & 1 & 1 \\ 2 & 4 & 2 & 1 & 2 & 1 \\ 2 & 2 & 4 & 1 & 1 & 2 \\ 2 & 1 & 1 & 4 & 2 & 2 \\ 1 & 2 & 1 & 2 & 4 & 2 \\ 1 & 1 & 2 & 2 & 2 & 4 \end{pmatrix}, \quad (2.58)$$

$$\int_{\Omega_e} N_i(\mathbf{r}) d\mathbf{r} = \frac{V_e}{12}, \quad (2.59)$$

$$\int_{\partial\Omega_e} N_i(\mathbf{r}) N_j(\mathbf{r}) ds = \begin{cases} \frac{A_\Delta}{12} (1 + \delta_{ij}) & \text{if triangular surface,} \\ \frac{A_\square}{36} (1 + \delta_{ij}) (1 - \frac{1}{2} \delta_{i,(5-j)}) & \text{if quadrilateral surface,} \end{cases} \quad (2.60)$$

where V_e is the volume of the element, A_Δ is the area of the triangular surface, and A_\square is the area of the quadrilateral surface. The matrix in Eq. (2.58) is indexed \mathbf{M}_{ij} where $i, j = 1, 2, \dots, 6$. The indices, ij , are the indices of the six basis functions corresponding to $N_i(\mathbf{r})$ and $N_j(\mathbf{r})$ respectively.

Notice the integral containing the gradient operator has been omitted because it is computed using a quadrature. If it could be computed analytically, it would be less computationally efficient than using a quadrature.

A_Δ is computed according to Eq. (2.50) and A_\square is computed as the sum of the area of two triangles, employing the same formula. For a simple extruded triangle, the volume calculation is straightforward and is the product of triangular area and linear height. However, allowing the nodes to move with respect

to each other makes the volume of the element difficult to calculate. Therefore, the Jacobian is used to calculate V_e . For more detail, see §2.3.3, especially Table 2.2.

2.3.3 Quadratures

Quadratures are sets of coordinates and weights which are used to approximate an integral. For a given set of weights, $\{w_i\}$, and a set of coordinates, $\{\mathbf{x}_i\}$, an integral can be represented as the sum

$$\int_{\Omega} f(\mathbf{x}) d\Omega \approx \sum_{i=1}^N f(\mathbf{x}_i) w_i \quad (2.61)$$

where Ω is an arbitrary domain described by $\{\mathbf{x}_i\}$. A quadrature, such as the one in Eq. (2.61), can be designed to exactly integrate a polynomial of order n . It is not necessarily true that the number of quadrature points, N , will equal the order of the polynomial exactly integrated, n . Generally, $n \neq N$.

For one-dimensional integrals, the Gaussian quadrature is common and the most compact quadrature. The Gaussian quadrature exactly integrates a polynomial of order n using exactly $N = n$ points. For the one-dimensional Gaussian quadrature, the number of points in the quadrature is the same as the order of the quadrature and $n = N$.

Two-dimensional and three-dimensional quadratures are necessary for the FEM. Triangular quadratures are not as simple to derive as line quadratures and the number of points need not equal the order of the polynomial integrated. The triangular quadrature as implemented here is symmetric and open. That is, there are no points on the boundary of the triangle [Dun85]. Any triangular quadrature will suffice that exactly integrates polynomials of a given order. There is no fixed relationship between n and N and for this quadrature, $n \neq N$.

Quadrilateral quadrature sets are simply tensor products of two line Gaussian quadratures. For an order n polynomial, now $N^2 = n$ points are required.

Wedge quadrature sets are simply tensor products of a line Gaussian quadrature and a triangular quadrature. Again, there is no fixed relationship between n and N .

Though only linear functions are used here, basis functions are generally polynomials of first, second, or third order; that is, linear, quadratic, or cubic functions. The quadratures described are capable of exactly integrating polynomial functions of given order so there exists a quadrature order that will exactly integrate the finite element quantities to numeric precision. The table of the order required for exact integration are provided in Table 2.1.

All of the quadratures described here are tabulated for a reference element be it a line, triangle, quadrilateral, or wedge. Integration in the FEM is performed on an arbitrary element in space. Therefore, it is necessary to perform a coordinate transform when using a quadrature set. Integration with transformed

Table 2.1 Quadrature Orders for FEM Quantities.

Quantity	Linear	Quadratic	Cubic
$\int_{\Omega} N_i(\mathbf{r}) d\Omega$	1	2	4 [†]
$\int_{\Omega} N_i(\mathbf{r})N_j(\mathbf{r}) d\Omega$	2	4	6
$\int_{\Omega} \nabla N_i(\mathbf{r}) \cdot \nabla N_j(\mathbf{r}) d\Omega$	2	3	5

[†] A third-order quadrature would be exact but the triangular quadrature would have negative weights so a fourth order quadrature is selected.

Table 2.2 Jacobi for Selected Elements.

Element	J
Triangle	A_e
Quadrilateral	$\frac{1}{4}A_e$
Wedge	$\frac{1}{2}V_e$

coordinates from domain Ω to the reference domain Ω_{ref} can be written

$$\int_{\Omega} f(\mathbf{x}) d\Omega = \int_{\Omega_{ref}} f(\mathbf{x}) |\mathbf{J}| d\Omega_{ref} \approx \sum_{i=1}^N w_i f(\mathbf{x}_i) |\mathbf{J}_i| \quad (2.62)$$

where \mathbf{J} is the Jacobian matrix, \mathbf{J}_i is the Jacobian matrix at quadrature coordinate \mathbf{x}_i , and $|\cdot|$ represents the matrix determinant. Notationally, $J = |\mathbf{J}|$ and is termed the Jacobi.

Isoparametric elements are elements in which shape functions can be used to relate global coordinates, (x, y, z) , to local coordinates (ξ, ζ, η) . Generally, non-curved elements are isoparametric. For isoparametric elements, such as triangles and wedges, the Jacobi is constant over the element and can be pre-calculated. Pre-calculating the Jacobi avoids populating and evaluating the determinant of a matrix for each integration point. For the elements of concern, these values are presented in Table 2.2 [Fel04].

For the general element, the Jacobian matrix, \mathbf{J}_i , is calculated at each point of the quadrature (ξ_i, ζ_i, η_i) as described in Eq. (2.63).

$$\mathbf{J}_i = \begin{pmatrix} \sum_{k=1}^{N_P} \frac{\partial N_k}{\partial \xi} \bigg|_{(\xi_i, \zeta_i, \eta_i)} x_k & \sum_{k=1}^{N_P} \frac{\partial N_k}{\partial \zeta} \bigg|_{(\xi_i, \zeta_i, \eta_i)} x_k & \sum_{k=1}^{N_P} \frac{\partial N_k}{\partial \eta} \bigg|_{(\xi_i, \zeta_i, \eta_i)} x_k \\ \sum_{k=1}^{N_P} \frac{\partial N_k}{\partial \xi} \bigg|_{(\xi_i, \zeta_i, \eta_i)} y_k & \sum_{k=1}^{N_P} \frac{\partial N_k}{\partial \zeta} \bigg|_{(\xi_i, \zeta_i, \eta_i)} y_k & \sum_{k=1}^{N_P} \frac{\partial N_k}{\partial \eta} \bigg|_{(\xi_i, \zeta_i, \eta_i)} y_k \\ \sum_{k=1}^{N_P} \frac{\partial N_k}{\partial \xi} \bigg|_{(\xi_i, \zeta_i, \eta_i)} z_k & \sum_{k=1}^{N_P} \frac{\partial N_k}{\partial \zeta} \bigg|_{(\xi_i, \zeta_i, \eta_i)} z_k & \sum_{k=1}^{N_P} \frac{\partial N_k}{\partial \eta} \bigg|_{(\xi_i, \zeta_i, \eta_i)} z_k \end{pmatrix} \quad (2.63)$$

In Eq. (2.63), N_P is the number of points in the element, $N_k(\mathbf{r})$ is the basis function centered at the k^{th} corner, and for corner coordinates (x_k, y_k, z_k) for $k = 1, 2, \dots, N_P$.

With the Jacobian populated as Eq. (2.63), $J = |\mathbf{J}|$ is simply the matrix determinant. For the quadrature integration of derivative quantities as necessary in the FEM, the derivatives must also be translated from the reference element to the spatial element. This is performed according to Algorithm 2.1. The notation is dense as the method requires two sets of coordinates. First, the coordinate in the reference element (ξ, ζ, η) and second, the coordinate in Cartesian space (x, y, z) .

The vector $\mathbf{d}_{i, (\xi, \zeta, \eta)}$ is the gradient vector for N_i with respect to the reference coordinates (ξ, ζ, η) .

$$\mathbf{d}_{i, (\xi, \zeta, \eta)} = \nabla_{(\xi, \zeta, \eta)} N_i(\mathbf{r}) \quad (2.64)$$

Vector $\mathbf{d}_{i, (x, y, z)}$ is the gradient vector for N_i with respect to the Cartesian coordinates (x, y, z) .

$$\mathbf{d}_{i, (x, y, z)} = \nabla_{(x, y, z)} N_i(\mathbf{r}) \quad (2.65)$$

In Algorithm 2.1, the quadrature has points $\{\mathbf{x}_p\}$ and weights $\{w_p\}$ for $p = 1, 2, \dots, N$ and the value of the integral, v , is represented by Eq. (2.66).

$$v = \int_{\Omega_e} \nabla_{(x, y, z)} N_i(\mathbf{r}) \cdot \nabla_{(x, y, z)} N_j(\mathbf{r}) d\Omega_e \quad (2.66)$$

Algorithm 2.1 Integral of Derivative with Jacobian Method.

- 1: $v = 0$
 - 2: **for** $p = 1, N_P$ **do**
 - 3: Calculate the Jacobian \mathbf{J} as in Eq. (2.63).
 - 4: Calculate the vector $\mathbf{d}_{i, (\xi, \zeta, \eta)}$ at quadrature point \mathbf{x}_p .
 - 5: Calculate the vector $\mathbf{d}_{j, (\xi, \zeta, \eta)}$ at quadrature point \mathbf{x}_p .
 - 6: Invert and store the Jacobian \mathbf{J}^{-1} .
 - 7: Calculate the vector $\mathbf{d}_{i, (x, y, z)} = \mathbf{d}_{i, (\xi, \zeta, \eta)} \mathbf{J}^{-1}$.
 - 8: Calculate the vector $\mathbf{d}_{j, (x, y, z)} = \mathbf{d}_{j, (\xi, \zeta, \eta)} \mathbf{J}^{-1}$.
 - 9: $v = v + \left(\mathbf{d}_{i, (x, y, z)}^T \mathbf{d}_{j, (x, y, z)} \right) w_p |\mathbf{J}|$
-

2.4 Power Iterations

The FEM is used to solve a fixed source problem for a given source distribution $q_g(\mathbf{r})$. However, for multigroup problems with a fission source, the problem is an eigenvalue problem and the source is not

known. For eigenvalue problems, the problem does not have a fixed source and the system has many solutions. The method of Power Iterations allows eigenvalue problems to be solved iteratively for the fundamental eigenvalue and eigenvector.

2.4.1 Convergence of Power Iteration Method

Noting the FEM equations can be written as matrix form Eq. (2.33), the discretized multigroup neutron diffusion equation can be rewritten as

$$\mathbf{B}(\Phi, k_{eff}) \Phi = \frac{1}{k_{eff}} \mathbf{M} \Phi \quad (2.67)$$

where Φ is the vector of the flux containing all energy groups, matrix \mathbf{B} contains the diffusion, removal, and all scattering terms, and \mathbf{M} includes all fission generation and operates on Φ and k_{eff} . \mathbf{B} is an S-matrix and its inverse, \mathbf{B}^{-1} , exists and has all positive elements [Nak77]. Therefore, Eq. (2.67) can be rewritten as

$$\Phi = \frac{1}{k_{eff}} \mathbf{R} \Phi \quad (2.68)$$

where

$$\mathbf{R} = \mathbf{B}^{-1} \mathbf{M}. \quad (2.69)$$

Matrix \mathbf{M} is non-symmetric and is non-negative and \mathbf{B} is an S-matrix; therefore, \mathbf{R} is a non-symmetric, non-negative matrix.

In the solution of Eq. (2.2), the largest eigenvalue, k_{eff} , is desired along with its associated eigenvalue, ϕ_g . The solution can be found using the method of power iterations which can be written as

$$\Phi^{(s+1)} = \frac{1}{k_{eff}^{(s)}} \mathbf{R} \Phi^{(s)}, \quad (2.70)$$

$$k_{eff}^{(s+1)} = k_{eff}^{(s)} \frac{\langle \mathbf{w}, \Phi^{(s+1)} \rangle}{\langle \mathbf{w}, \Phi^{(s)} \rangle}, \quad s = 1, 2, \dots, \infty, \quad (2.71)$$

where s is the iteration counter, \mathbf{w} is a weighting vector and $\langle \mathbf{w}, \Phi \rangle$ is an inner-product. According to the Perron-Frobenius theorem, a matrix with the properties of \mathbf{R} has a unique, positive eigenvalue, greater in magnitude than the modulus of all other eigenvalues of the matrix [Geh92; Nak77]. The weighting vector \mathbf{w} is arbitrary but does affect convergence rate. For this work, $\mathbf{w} = \{\nu \Sigma_f\}$ such that the inner product $\langle \{\nu \Sigma_f\}, \Phi \rangle$ represents the summation of the fission neutron production rate throughout all energy groups and all elements.

It can then be shown in that the method of power iterations described in Eq. (2.70) and Eq. (2.71) converges to the largest eigenvalue, k_{eff} , and unique positive eigenvector Φ [Nak77]. For notation and

enumeration, allow the eigenvalue to be rewritten as

$$\mu = \frac{1}{k_{eff}}. \quad (2.72)$$

The eigenvectors \mathbf{u}_n and corresponding eigenvalues, μ_n , of \mathbf{R} are defined by

$$\mathbf{u}_n = \mu_n \mathbf{R} \mathbf{u}_n. \quad (2.73)$$

It may be proved that all eigenvalues, μ , are real, positive, and distinct. The eigenvalues are then numbered in the sequence

$$\mu_0 < \mu_1 < \mu_2 < \dots < \mu_N \quad (2.74)$$

where N is the rank of the problem. The eigenvectors have the orthogonality relations

$$\mathbf{u}_n^T \mathbf{u}_m = 0 \quad \text{for } n \neq m \quad (2.75)$$

where $\mathbf{u}^T \mathbf{u}$ is the vector inner-product. Assume eigenvectors are normalized such that

$$\mathbf{u}_n^T \mathbf{u}_n = 1 \quad \text{for } n = 1, 2, \dots, N. \quad (2.76)$$

The initial vector, $\Phi^{(0)}$, may be expressed as a projection onto the eigenvectors using a linear superposition of eigenvectors of the form

$$\Phi^{(0)} = \sum_{n=1}^N c_n^{(0)} \mathbf{u}_n \quad (2.77)$$

where $c_n^{(0)}$ is a coefficient given by the orthogonality relationship from Eq. (2.75) such that

$$c_n^{(0)} = \mathbf{u}_n^T \Phi^{(0)}. \quad (2.78)$$

Using this eigenmode projection, Eq. (2.70) can be rewritten as

$$\Phi^{(s+1)} = \mu^{(s)} \mu^{(s-1)} \mu^{(s-2)} \dots \mu^{(0)} \mathbf{R} \Phi^{(0)}. \quad (2.79)$$

Then, Eq. (2.77) can be inserted into Eq. (2.79).

$$\Phi^{(s+1)} = \mu^{(s)} \mu^{(s-1)} \dots \mu^{(0)} \sum_{n=1}^N c_n^{(0)} \mathbf{R} \mathbf{u}_n \quad (2.80)$$

$$= \left(\prod_{p=0}^s \mu^{(p)} \right) \sum_{n=1}^N c_n^{(0)} \mathbf{R} \mathbf{u}_n \quad (2.81)$$

Recalling the relationship Eq. (2.77).

$$\Phi^{(s+1)} = \left(\prod_{p=0}^s \mu^{(p)} \right) \sum_{n=1}^N c_n^{(0)} \frac{1}{\mu_n^{(s+1)}} \mathbf{u}_n \quad (2.82)$$

Eq. (2.82) can be rewritten by dividing and multiplying by μ_0 and dividing and multiplying by c_0 .

$$\Phi^{(s+1)} = \left(\prod_{p=0}^s \frac{\mu^{(p)}}{\mu_0} \right) c_0 \left(\mathbf{u}_0 \sum_n^N \frac{c_n^{(0)}}{c_0^{(0)}} \frac{\mu_0}{\mu_n^{(s+1)}} \mathbf{u}_n \right) \quad (2.83)$$

$$\Phi^{(s+1)} \approx \text{const.} \cdot \left(\mathbf{u}_0 \sum_n^N \frac{c_n^{(0)}}{c_0^{(0)}} \frac{\mu_0}{\mu_n^{(s+1)}} \mathbf{u}_n \right) \quad (2.84)$$

The ordering of unique eigenvalues required by Eq. (2.74) requires $\mu_0/\mu_n < 1$ and the problem is convergent. The convergence rate is determined by the dominance ratio, d , where

$$d \equiv \max_{n=1,2,\dots,N} \frac{\mu_0}{\mu_n} = \frac{\mu_0}{\mu_1} \quad (2.85)$$

and it can be seen that for problems with small dominance ratio, the power iteration method will converge more quickly [Nak77]. Eq. (2.85) can be rewritten in terms of the eigenvalues of the multigroup neutron diffusion equation $k_{eff} = k_{eff,0}$ and $k_{eff,1}$.

$$d \equiv \frac{k_{eff,1}}{k_{eff,0}} \quad (2.86)$$

As the dominance ratio approaches unity, $d \rightarrow 1$, the power iteration method will be slower to converge.

Convergence criteria are then specified as an absolute tolerance in the sense of the eigenvalue

$$\varepsilon_{k_{eff}} > |k_{eff}^{(s+1)} - k_{eff}^{(s)}| \quad (2.87)$$

and as a relative tolerance in the sense of the eigenvector

$$\varepsilon_{\Phi} > \max_i \left| \frac{\Phi_i^{(s+1)} - \Phi_i^{(s)}}{\Phi_i^{(s)}} \right|. \quad (2.88)$$

It is important to note that all of the analysis in this section assumed the matrix \mathbf{R} does not change between iterations. For simple multigroup criticality calculations, this assumption is correct. However, in multiphysics nuclear power reactor simulations, the cross sections of the problem may be considered functions of material temperature or may have some variable number density. For these problems, the matrix \mathbf{R} is necessarily updated between power iterations in a nonlinear manner. For nonlinear power

iteration updates, convergence is no longer guaranteed. The argument for convergence with nonlinear power iterations in this implementation falls back to the stability of the physical system.

The proof of the power iteration as presented is valid for general solution methods to the multigroup neutron diffusion equation. However, the exact procedure of this proof is computationally inefficient and not used in practice. The matrix \mathbf{B} is not inverted to construct \mathbf{R} as in Eq. (2.69). Instead, Eq. (2.67) is solved using an iterative solution method. In this work, the FEM is used to solve Eq. (2.67) as described in §2.3. In the FEM representation, the quantity $\frac{1}{k_{eff}}\mathbf{M}\Phi$ in Eq. (2.67) is equivalent to the combined source representation, $q_{g,e}$, as described in Eq. (2.16). While the notation from the FEM could be rewritten for this proof, the notation from Eq. (2.67) is more intuitive and compact.

2.4.2 Calculation of Source with Power Iterations

The source term in Eq. (2.67), $\frac{1}{k_{eff}}\mathbf{M}\Phi$, is representative of the combined multigroup neutron source $q_g(\mathbf{r})$. The multigroup neutron diffusion equation from Eq. (2.4) can be written with the source term, $q_g(\mathbf{r})$, expanded into its component parts.

$$-\nabla \cdot (D_g(\mathbf{r})\nabla\phi_g^{(s)}(\mathbf{r})) + \Sigma_{r,g}(\mathbf{r})\phi_g^{(s)}(\mathbf{r}) = q_{fiss,g}(\mathbf{r}) + q_{up,g}(\mathbf{r}) + q_{down,g}(\mathbf{r}) \quad (2.89)$$

Recall from the definitions of the source components that their calculation requires the flux $\phi_g(\mathbf{r})$. The source components each require different energy groups of the flux distribution to be known. The fission component $q_{fiss,g}(\mathbf{r})$ requires all groups. The up-scatter component $q_{up,g}(\mathbf{r})$ requires lower energy groups (i.e. $g' > g$). The down-scatter component $q_{down,g}(\mathbf{r})$ requires higher energy groups (i.e. $g' < g$). Based on these requirements, these source components can be calculated based on different iterations within the power iteration method. This is described in Algorithm 2.2. Eq. (2.89) is then more explicitly written.

$$-\nabla \cdot (D_g(\mathbf{r})\nabla\phi_g^{(s+1)}(\mathbf{r})) + \Sigma_{r,g}(\mathbf{r})\phi_g^{(s+1)}(\mathbf{r}) = q_{fiss,g}^{(s)}(\mathbf{r}) + q_{up,g}^{(s)}(\mathbf{r}) + q_{down,g}^{(s+1)}(\mathbf{r}) \quad (2.90)$$

Therefore, the matrix \mathbf{M} is not computed but instead, the combined multigroup neutron source $q_g(\mathbf{r})$ is updated using an appropriate scalar flux iterate.

2.5 Implementation

A FEM neutron diffusion solution has been developed using the above formulae. The program begins by reading a geometry description specified in a plain text VTK file [Sch06]. The VTK format is chosen because it is a standard that can be used with visualization tools such as ParaView [Aya15] and VisIt [Law05]. Additionally, open-source C and Python packages exist for easy manipulation of the format. Cross sections are specified in either a plain text user format or the ISOTXS format as common to fast

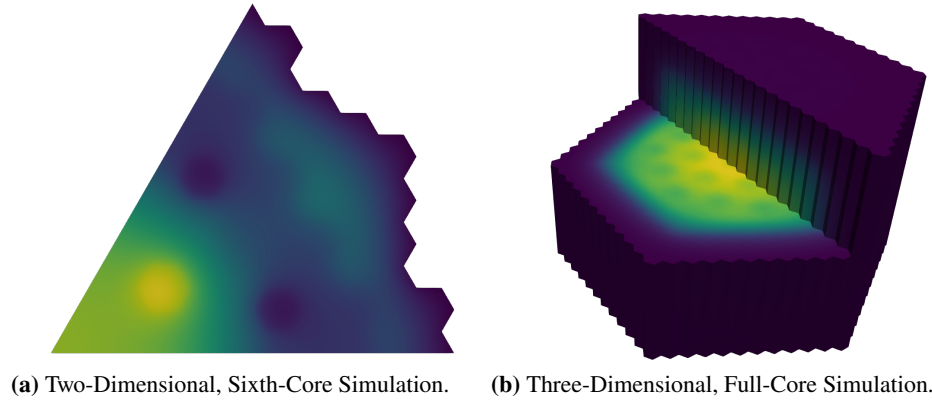


Figure 2.5 Typical Fast Flux Visualization for Two-Dimensional and Three-Dimensional Simulations.

reactor applications and the multigroup cross section generator MC**2 [Lee12]. The multigroup neutron diffusion equation is solved according to Eq. (2.4). The resulting effective neutron multiplication factor, k_{eff} , is written to an output file. The multigroup neutron flux is written to a different results VTK file for easy visualization. In a multiphysics simulation, material temperatures and thermal expansion dimensions are also written to the results VTK. Examples of fast neutron flux visualized in the results VTK are shown in Fig. 2.5 for two-dimensional and three-dimensional reactor simulations.

2.5.1 Algorithm

The algorithm for the solution to the diffusion equation is similar to most implementations of the power iteration method to solve the multigroup neutron diffusion equation. The algorithm itself is presented in Algorithm 2.2. The steps unique to the FEM are steps 5 and 13. These require the quantities previously derived and form the linear system described by the FEM.

In step 3 the matrix is reordered. Mathematically this has no effect on the result as the linear system represented is equivalent. This choice to reorder the system is made to improve computational efficiency. Indexing nodes that are physically proximate with proximate indices causes rows in the finite element matrix, \mathbf{A}_g , to be closely coupled to nearby rows. In a general unstructured and unordered mesh, rows may be coupled to other random rows in the matrix. This step of reordering the matrix \mathbf{A}_g seeks to decrease the bandwidth of the matrix and encourage cache hits when accessing coupled values in the linear system. The ordering chosen is the Reverse Cuthill-McKee (RCM) method and common to sparse linear systems [Cut69]. An example of the bandwidth reduction provided by the RCM method is shown in Fig. 2.6. The plots show the sparsity pattern of the matrix before and after reordering. After reordering, the matrix bandwidth is reduced from 2,326 to 81.

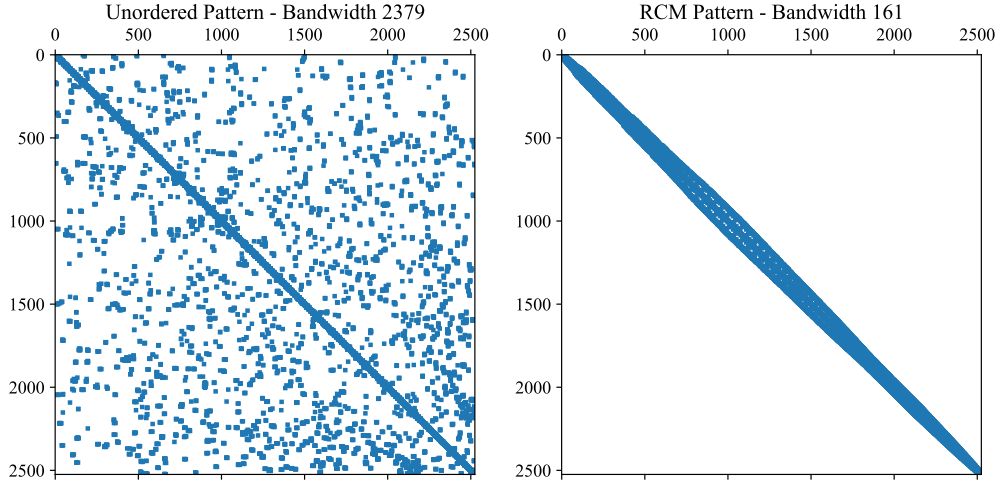


Figure 2.6 Demonstration of RCM Matrix Ordering.

Algorithm 2.2 General Iteration Scheme

- 1: Read mesh from VTK.
 - 2: Initialize $\bar{\phi}_g^{(0)}$.
 - 3: Order the nodes of the mesh into RCM order.
 - 4: Calculate $\Sigma_{s,g' \rightarrow g}$, $\Sigma_{r,g}$, and $\nu \Sigma_{f,g}$ for each element.
 - 5: Calculate finite element matrix \mathbf{A}_g for each group. Store this.
 - 6: **while** Power Iteration **do**
 - 7: Update the iteration counter. $s = s + 1$
 - 8: Update $q_{fiss,g}$ and $q_{up,g}$ for all groups from previous data $\bar{\phi}^{(s-1)}$.
 - 9: Update $\widetilde{\chi}_g$ in each element using previous data.
 - 10: **for** $g = 1, G$ **do**
 - 11: Update $q_{down,g}$ from current data $\bar{\phi}_g^{(s)}$
 - 12: Calculate total source in each element.
 - 13: Update finite element Vector \mathbf{f}_g with new source.
 - 14: Solve $\mathbf{A}_g \mathbf{u}_g = \mathbf{f}_g$ using an iterative technique (See §2.5.4).
 - 15: Parse \mathbf{u}_g for ϕ_g solution on nodes.
 - 16: Calculate element-average $\bar{\phi}_g$.
 - 17: Update k_{eff} .
 - 18: Check convergence.
 - 19: Perform non-linear update if necessary and update \mathbf{A}_g .
-

Step 19 in Algorithm 2.2 incorporates the possibility of non-linear updates into the power iteration method. This invalidates the proof of convergence from §2.4 but is performed to simulate thermal hydraulic and thermal expansion effects as described in Chapter 4 and Chapter 5 respectively. This procedure is commonly used in practice and no convergence problems have been observed. This update also requires recalculating the finite element matrix, \mathbf{A}_g , for each group.

A benefit of this implementation is that the finite element vector \mathbf{f}_g must be updated on each power iteration of the solution whereas the matrix \mathbf{A}_g is described entirely by geometry and the material cross sections. For this reason, in problems with no multiphysics simulations, \mathbf{A}_g can be generated once at the beginning of the problem and stored for the duration of the calculation.

2.5.2 Memory and Storage

The finite element matrix, \mathbf{A}_g , is large and sparse so a sparse storage and sparse solution to the linear system are required. Many sparse matrix implementations have been described and implemented in the past including triplet, reduced column, and reduced row storage [Duf02]. For this work a TwoTable method is chosen which was uniquely developed. TwoTable storage is chosen for its simplicity and implementation with the FEM. Future work may include a reduced row implementation but there will be a trade-off between memory minimization and computational efficiency.

The TwoTable method is composed of two separate matrices in memory. An integer index table, IDX , and an IEEE double precision value table VAL . Table IDX is dimension $DOF \times D$ where DOF is the number of degrees of freedom of the linear system and D is the maximum number of nodes that a node shares including itself. D is the degree of the matrix plus one. This must be determined at the beginning of the problem based on the input geometry. Table VAL is dimension $DOF \times D \times G$ where G is the number of energy groups as the matrix \mathbf{A}_g is group dependent.

IDX is initialized to -1 and VAL is initialized to 0.0 such that an index of -1 corresponds to a null entry in the matrix. IDX is then populated with a modified adjacency graph. Values in IDX indicate the column in which the VAL entry occurs. An example unstructured mesh is given in Fig. 2.7. Then, for node 5, the table IDX may resemble Eq. (2.91).

$$\text{IDX}(:, 5) = (8, 200, 48, 96, 5) \quad (2.91)$$

Eq. (2.91) is only an example because the order of the nodes in row 5 is arbitrary. Similarly, a numeric example is provided in Eq. (2.92) (note that the value “3” is an arbitrary node number).

$$\left. \begin{array}{l} \text{IDX}(7, 3) = 8 \\ \text{VAL}(7, 3, g) = 12.1 \end{array} \right\} \Rightarrow A_{7,8,g} = 12.1 \quad (2.92)$$

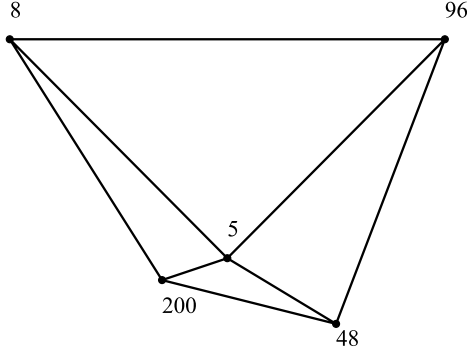


Figure 2.7 Example Unstructured Mesh.

Eq. (2.92) indicates that the value of matrix \mathbf{A}_g for group g in the seventh row in the eighth column is 12.1. This will allow for simple row operations and efficient matrix-vector multiplication which will be necessary in the solution of the linear system.

2.5.3 Boundary Conditions

Boundary conditions merit brief consideration. There are many ways to implement boundary conditions and all result in mathematically the same solution. The choices made for this work are presented below. Mirror boundary conditions require $\nabla\phi_g(\mathbf{r}) = 0$ for $\mathbf{r} \in \partial\Omega$. These are termed “natural” boundary conditions because the finite element matrix, \mathbf{A}_g , requires no additional treatment and this condition is natural. In the albedo representation, this is equivalent to $\alpha = 0$.

Albedo boundary conditions are treated with an additional contribution to the finite element matrix, \mathbf{A}_g . These contributions represent a line integral in two dimensions and a surface integral in three dimensions. These values are found in Eq. (2.34) and the quantities are expressed in §2.3.2 or by quadratures §2.3.3.

Zero-flux boundary conditions require $\phi_g(\mathbf{r}) = 0$ for $\mathbf{r} \in \partial\Omega$. These are treated by removing these entries from the finite element matrix, \mathbf{A}_g . Entries are removed by using an index vector. In a problem with only mirror and albedo boundary conditions, each node corresponds to an index row/column of the matrix. With entries removed, node number and index number may not exactly agree. A vector ID is introduced to track node numbers and their location in the finite element matrix, \mathbf{A}_g [Joh09]. Nodes with non-zero flux boundary conditions are set to a sequential positive integer. Nodes with zero-flux boundary conditions are set to a negative integer (-1) and are omitted in the actual solution of the system. Other strategies have been proposed such as the penalty approach [Hug87] and forcing the solution of the linear system [Li18]. This method is chosen because it decreases the degree of freedom of the linear system while encouraging a well conditioned matrix. Now, the degree of freedom of the matrix is equal to the number of nodes with non-zero flux boundary conditions. Alternatively, zero-flux boundary conditions can be represented as $\alpha \rightarrow \infty$.

2.5.4 Linear System Solution

For a non-singular linear system $\mathbf{A}_g \mathbf{u}_g = \mathbf{f}_g$, where \mathbf{A}_g is a square matrix, there exists a unique solution. Many strategies have been proposed to solve this system in an efficient manner. Options are restricted in this work because the solution must operate with a sparsely stored linear system and result in no fill-in. This immediately demands an iterative method. The linear system described by the FEM can then be exploited for its unique properties to select a favorable solution method.

The finite element matrix \mathbf{A}_g for the problem described in Eq. (2.4) is Symmetric Positive Definite (SPD) if the multigroup equations are solved one group at a time as they are in this work. Note, the matrix will not have this especially useful property if all the groups were instead solved simultaneously. The symmetry condition is straight-forward and is observed in the elemental matrix description Eq. (2.34). Briefly, $\mathbf{A}_g^T = \mathbf{A}_g$ because the matrix elements $A_{i,j,g,e} = A_{j,i,g,e}$. Positive definiteness is a particularly useful condition but is often difficult to prove. \mathbf{A}_g is typically diagonally dominant for structured meshes but, generally, the matrix is not diagonally dominant.

A matrix $\mathbf{A}_h \in \mathbb{R}^{n \times n}$ is positive definite if

$$\mathbf{x}^T \mathbf{A}_g \mathbf{x} > 0 \quad \forall \mathbf{x} \in \mathbb{R}^n, \mathbf{x} \neq 0. \quad (2.93)$$

Let $\mathbf{x} = \{x_i\}$ for $i = 1, 2, \dots, N$. Then the vector-matrix and matrix-vector products can be rewritten as summations [Hug87].

$$\mathbf{x}^T \mathbf{A}_g \mathbf{x} = \sum_{i=1}^N \sum_{j=1}^N x_i A_{ij} x_j \quad (2.94)$$

By the definition of $A_{i,j,g}$ in Eq. (2.34) and Eq. (2.32).

$$\mathbf{x}^T \mathbf{A}_g \mathbf{x} = \sum_{i=1}^N \sum_{j=1}^N x_i a_g(N_i, N_j) x_j \quad (2.95)$$

Noting the property that $a_g(\cdot, \cdot)$ is a bilinear operator (i.e. linear in both arguments) [Li18].

$$\mathbf{x}^T \mathbf{A}_g \mathbf{x} = a_g \left(\sum_{i=1}^N x_i N_i, \sum_{j=1}^N x_j N_j \right) \quad (2.96)$$

By construction of the FEM, $w(\mathbf{r}) = \sum_{i=1}^N x_i N_i$ where $w(\mathbf{r})$ is a piecewise continuous polynomial of arbitrary order.

$$\mathbf{x}^T \mathbf{A}_g \mathbf{x} = a_g(w(\mathbf{r}), w(\mathbf{r})) \quad (2.97)$$

$a(\cdot, \cdot)$ can be shown to form a norm $\|\cdot\|_a$ [Li18] satisfying the positive definite condition Eq. (2.93).

$$\mathbf{x}^T \mathbf{A}_g \mathbf{x} > 0 \quad \forall \mathbf{x} \in \mathbb{R}^n, \mathbf{x} \neq 0 \quad (2.98)$$

A given matrix can be verified as positive definite by one of two methods. First, all eigenvalues of the matrix have positive real components. Second, the matrix has a Cholesky decomposition such that $\mathbf{A} = \mathbf{L} \mathbf{L}^*$ where \mathbf{L} is a lower triangular matrix with positive diagonal entries and \mathbf{L}^* is the conjugate transpose of matrix \mathbf{L} [Ips09]. For real valued matrices, the conjugate transpose is equivalent to the conventional transpose. Though an eigenvalue decomposition or a Cholesky factorization may be useful for debugging or numerical analysis purposes, these operations are computationally expensive. Instead, this is verified here for the general matrix and not tested during simulations.

Conventional methods used to solve a linear system described by an SPD matrix include Gauss-Seidel iteration with Successive Over-Relaxation (SOR) and the Conjugate Gradient (CG) Krylov subspace method. SOR requires *a priori* knowledge of the optimized over-relaxation factor, ω_{opt} , for optimal performance. In practice, calculation of ω_{opt} is performed analytically for contrived solutions or in a modified guess-and-check method. The CG method is chosen because it requires no *a priori* knowledge and produced a solution to the same tolerance in a reduced wall-time without the need for guess-and-check iterations.

A simple recipe for the CG method is replicated in Algorithm 2.3 [Kel95]. In the k^{th} iteration of the CG method, the residual

$$\mathbf{r}_k = \mathbf{x}^* - \mathbf{x}_k \quad (2.99)$$

is minimized where \mathbf{x}^* is the exact solution $\mathbf{x}^* = \mathbf{A}^{-1} \mathbf{b}$. The k^{th} Krylov subspace, \mathcal{K}_k , is

$$\mathcal{K}_k = \text{span} \left(\mathbf{r}_0, \mathbf{A} \mathbf{r}_0, \dots, \mathbf{A}^{k-1} \mathbf{r}_0 \right) \quad (2.100)$$

for $k \geq 1$. Eventually, the Krylov subspace from Eq. (2.100) can be guaranteed to contain the exact solution \mathbf{x}^* and the CG method will converge to the exact solution. It can be shown that, for a rank N linear system, a maximum of N CG iterations are required to converge to the exact solution [Kel95]. In practice, CG is terminated after some small relative residual, ε , is obtained. This termination condition occurs well before the exact solution is obtained.

In Algorithm 2.3, ε is a tolerance set by the user and the square of the two-norm is most efficiently replaced by the vector inner-product as

$$\|\mathbf{r}\|_2^2 = \mathbf{r}^T \mathbf{r}. \quad (2.101)$$

It is noted that this method requires little storage with only four vectors required (\mathbf{x} , \mathbf{w} , \mathbf{p} , and \mathbf{r}). Additionally, each iteration requires only two scalar products and a single matrix-vector product. The matrix-vector product proves to be the most computationally expensive [Kel95]. As most of the computational time of the diffusion solutions is spent in the linear system solution, it is crucial that this process be efficient.

Algorithm 2.3 Conjugate Gradient Method [Kel95].

```
1:  $k = 0$ 
2:  $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$ 
3:  $\rho_k = \|\mathbf{r}\|_2^2$ 
4:  $k = k + 1$ 
5: while  $\sqrt{\rho_{k-1}} > \varepsilon \|\mathbf{b}\|_2$  do
6:   if  $k = 1$  then
7:      $\mathbf{p} = \mathbf{r}$ 
8:   else
9:      $\beta = \rho_{k-1} / \rho_k - 2$ 
10:     $\mathbf{p} = \mathbf{r} + \beta \mathbf{p}$ 
11:     $\mathbf{w} = \mathbf{A}\mathbf{p}$ 
12:     $\beta = \rho_{k-1} / \mathbf{p}^T \mathbf{w}$ 
13:     $\mathbf{x} = \mathbf{x} + \beta \mathbf{p}$ 
14:     $\mathbf{r} = \mathbf{r} - \beta \mathbf{w}$ 
15:     $\rho_k = \|\mathbf{r}\|_2^2$ 
16:     $k = k + 1$ 
```

With the solution of the linear system, the implementation of the FEM to solve the multigroup neutron diffusion problem is completed. Results in the form of analytic and benchmark verification problems are presented in Chapter 3.

CHAPTER

3

NEUTRON DIFFUSION RESULTS

3.1 Introduction

The implementation of the FEM in this work has been compared against standard benchmark problems as well as analytic solutions to the multigroup neutron diffusion equation. Comparison against benchmark problems demonstrates an ability to solve problems for which this work was intended. Comparison against analytic solutions allows for more detailed error and convergence analysis as not only the system k_{eff} but also the flux solution is known exactly.

These comparisons serve as a verification of this implementation of the FEM. The verification strategy employed in this work is developed by Oberkampf & Trucano [Obe07]. The first step is “code verification.” Code verification compares computational results to exact analytic or manufactured results. Code verification serves to demonstrate that the code itself is solving the equations correctly as designed and with quantified numerical errors. This will be demonstrated with convergence to the analytic answer at the expected rate. The second step is “solution verification.” Solution verification compares computational results to benchmark results for the intended application of the solver. These benchmarks may have been calculated computationally via another method or may come from experimental data. Whereas analytic solutions are known exactly, the data of benchmarks is not exact. Typically, the benchmark solution has been verified by others previously.

3.2 Error Analysis

For all benchmark and analytic problems solved, a convergence study is presented. An error vector is defined as $\mathbf{e} = \phi(\mathbf{r}) - \phi_{FEM}$ where ϕ_{FEM} is the solution to the FEM system of equations. The error is then considered in terms of both Root-Mean-Squared (RMS) error calculated as

$$\text{RMS}(\mathbf{e}) = \sqrt{\frac{1}{N} \sum_{i=1}^N e_i^2}, \quad (3.1)$$

and error is considered in the maximal norm calculated as

$$\|\mathbf{e}\|_{\infty} = \max_{i=1,2,\dots,N} |e_i|. \quad (3.2)$$

It has been shown that for a bounded second spatial derivative within the problem domain, the FEM with linear elements as derived in Chapter 2 is second-order convergent in space [Li18]. This implies the error in the maximal norm defined in Eq. (3.2) is bounded as

$$\|\mathbf{e}\|_{\infty} \leq ch^2 \|\nabla^2 \phi(\mathbf{r})\|_{\infty} \quad (3.3)$$

where \mathbf{e} is the error vector, h is the characteristic mesh size, and c is a constant. Eq. (3.3) implies that if the characteristic mesh size is halved, the error is quartered. This relationship is useful as a proper implementation of the FEM must converge to the correct answer and do so at the correct rate.

Mesh refinement studies are presented herein. For each refinement, h is halved by introducing new elements and placing new nodes at the midway point between existing nodes. Let i be the refinement index and the refinement ratio for a second-order convergent method is

$$4 = \frac{e^{(i-1)}}{e^{(i)}} \quad (3.4)$$

such that for some error quantity e , the error should theoretically decrease by a factor of four for each refinement. As these are numerical solutions, the ratio rarely equals exactly four. It is observed that a few refinements are often necessary before the convergence reaches the asymptotic regime and the ratio approaches the expected value. This is especially the case when the second derivative is not bounded in problems with heterogeneous materials.

For analytic solutions, the error of the function $\phi(\mathbf{r})$ itself can be analyzed because the solution is known exactly. The derivations of the exact solutions are presented in Appendix A. Both RMS errors and maximal norm errors are presented. It is observed that the RMS error refinement ratio approaches the expected value of four before the maximal error refinement ratio because the RMS error is an integral quantity over the problem domain whereas the maximal norm error is a point-wise quantity.

For analytic criticality problems, k_{eff} is also presented and the ratio between refinements should assume the expected rate. Though the convergence is defined in Eq. (3.3) is for the flux itself, k_{eff} is expected to converge at the same rate because k_{eff} is an integral quantity of the flux. k_{eff} error is calculated with Eq. (3.5) in units of percent-mille [pcm].

$$k_{eff} \text{ error [pcm]} = (k_{ref} - k_{eff}) \times 10^5 \quad (3.5)$$

A summary of benchmark problems used in this chapter is presented in Appendix B. For benchmark problems, only k_{eff} is analyzed and expected to converge to the reference solution. The convergence rate of benchmark problems is not analyzed because the convergence rate is sensitive to the precision of the benchmark and can vary greatly between benchmarks. When assembly powers were available, these are also presented graphically.

3.3 Analytic Solutions

As a demonstration of the proper implementation of and solution to the neutron diffusion equation, analytic solutions are derived and then computed numerically. These are one-dimensional, two-dimensional, and three-dimensional problems. These problems exercise both triangle and wedge elements. For one-dimensional problems, a two-dimensional rectangular domain is used and the top and bottom edges, $y = 0$ and $y = L_x$, are treated as mirror boundary conditions to reduce the dimension of the problem. To verify there was no error obscured by this process, the results were reproduced for a rotated problem with the left and right edges ($x = 0$ and $x = L_x$) set to mirror boundary conditions as well.

3.3.1 One-Dimension, One-Group, Fixed Source

Arguably the simplest solution to the neutron diffusion equation, this problem consists of a fixed unit source throughout the problem. The exact solution is derived in §A.2 and is presented in Eq. (A.22). Results from the convergence study are presented in Table 3.1.

3.3.2 One-Dimension, One-Group, Criticality

This problem tests the calculation of the source term and the general power iteration implementation in the solution method. The exact solution is derived in §A.3 and presented in Eq. (A.39). Results from the convergence study are presented in Table 3.2. The exact value for the effective multiplication factor is $k_{ref} = 1.998028$.

Table 3.1 One-Dimension, One-Group, Fixed Source Convergence Study Results.

Refine	RMS	RMS ratio	$\ \mathbf{e}\ _\infty$	$\ \mathbf{e}\ _\infty$ ratio
0	1.44E-04	4.32	4.69E-04	2.67
1	3.33E-05	4.24	1.76E-04	3.04
2	7.85E-06	4.11	5.77E-05	3.25
3	1.91E-06	4.04	1.77E-05	3.38
4	4.72E-07	4.01	5.26E-06	3.46
5	1.18E-07	4.01	1.52E-06	3.51
6	2.94E-08		4.33E-07	

Table 3.2 One-Dimension, One-Group, Criticality Convergence Study Results.

Refine	k_{eff}	k_{eff} error [pcm]	k_{eff} ratio	RMS	RMS ratio	$\ \mathbf{e}\ _\infty$	$\ \mathbf{e}\ _\infty$ ratio
0	1.991780	624.77	4.02	3.29E-02	1.77	9.18E-02	1.70
1	1.996474	155.37	3.96	1.86E-02	2.67	5.41E-02	2.40
2	1.997635	39.27	3.98	6.96E-03	3.53	2.26E-02	3.42
3	1.997929	9.88	3.99	1.97E-03	3.84	6.60E-03	3.84
4	1.998003	2.47	4.00	5.13E-04	3.98	1.72E-03	3.96
5	1.998022	0.62	4.00	1.29E-04	3.98	4.35E-04	3.99
6	1.998026	0.15		3.24E-05		1.09E-04	
Ref.	1.998028						

Table 3.3 Two-Dimension, One-Group, Criticality Convergence Study Results.

Refine	k_{eff}	k_{eff} error [pcm]	k_{eff} ratio	RMS	RMS ratio	$\ \mathbf{e}\ _\infty$	$\ \mathbf{e}\ _\infty$ ratio
0	1.983243	1281.65	4.03	1.90E-02	1.66	6.63E-02	1.49
1	1.992884	317.64	3.96	1.15E-02	2.65	4.45E-02	2.59
2	1.995258	80.16	3.98	4.32E-03	3.43	1.72E-02	3.41
3	1.995858	20.15	3.99	1.26E-03	3.88	5.04E-03	3.87
4	1.996009	5.05	4.00	3.25E-04	3.96	1.30E-03	3.96
5	1.996047	1.26	4.00	8.20E-05	3.93	3.28E-04	3.93
6	1.996057	0.32		2.09E-05		8.34E-05	
Ref.	1.996060						

3.3.3 Two-Dimension, One-Group, Criticality

This problem tests the ability to solve truly two-dimensional problems. The exact solution is derived in §A.4 and presented in Eq. (A.65). Results from the convergence study are presented in Table 3.3. The exact value for the effective multiplication factor is $k_{ref} = 1.996060$.

3.3.4 One-Dimension, Two-Group, Criticality

This problem tests the solution of multigroup problems. The results presented are the convergence of k_{eff} and the convergence of the ratio of relative magnitude of thermal to fast flux ϕ_2/ϕ_1 . A proof is not provided that the relative magnitude should observe the expected convergence rate. However, the method is second-order convergent and the expected convergence rate is, in fact, observed.

The exact solutions are derived in §A.5 and the solutions are presented in Eq. (A.70) and Eq. (A.71). Results from the convergence study are presented in Table 3.4. The exact value for the effective multiplication factor is $k_{ref} = 0.892349$ and the exact value for the relative flux ratio is $(\phi_2/\phi_1)_{ref} = 0.261324$.

3.3.5 One-Dimension, One-Group, Two-Region, Criticality

This problem tests the mapping of materials to regions within the problem. The exact solution is derived in §A.6 and presented in Eq. (A.135). Results from the convergence study are presented in Table 3.5. The exact value for the effective multiplication factor is $k_{ref} = 0.982622$.

3.3.6 Three-Dimension, One-Group, Finite Cylinder

This problem is a finite cylinder composed of fissile material. The solution with the FEM uses a three-dimensional solution and wedge finite elements. The exact solution is derived in §A.7 and the

Table 3.4 One-Dimension, Two-Group, Criticality Convergence Study Results.

Refine	k_{eff}	k_{eff} error [pcm]	k_{eff} ratio	ϕ_2/ϕ_1	ϕ_2/ϕ_1 error	ϕ_2/ϕ_1 ratio
0	0.886982	536.70	4.00	0.260236	1.09E-03	4.65
1	0.891008	134.10	4.00	0.261090	2.34E-04	3.63
2	0.892013	33.56	4.00	0.261260	6.45E-05	3.62
3	0.892265	8.39	4.00	0.261306	1.78E-05	4.27
4	0.892328	2.10	4.00	0.261320	4.18E-06	3.78
5	0.892344	0.52	4.00	0.261323	1.11E-06	4.02
6	0.892348	0.13		0.261324	2.75E-07	
Ref.	0.892349			0.261324		

Table 3.5 One-Dimension, One-Group, Two-Region, Criticality Convergence Study Results.

Refine	k_{eff}	k_{eff} error [pcm]	k_{eff} ratio	RMS	RMS ratio	$\ \mathbf{e}\ _\infty$	$\ \mathbf{e}\ _\infty$ ratio
0	0.977882	473.94	3.89	1.70E-02	3.75	3.86E-02	3.74
1	0.981404	121.79	3.97	4.53E-03	3.91	1.03E-02	3.94
2	0.982315	30.70	3.99	1.16E-03	3.95	2.62E-03	4.00
3	0.982545	7.69	4.00	2.93E-04	3.96	6.55E-04	4.02
4	0.982602	1.92	4.00	7.40E-05	3.96	1.63E-04	4.02
5	0.982617	0.48		1.87E-05		4.05E-05	
Ref.	0.982622						

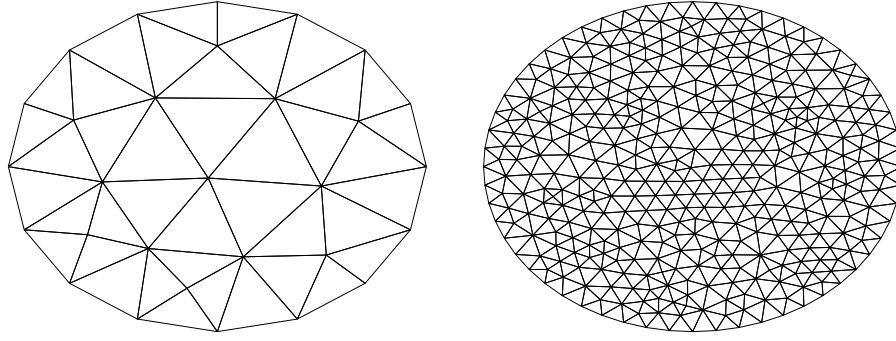


Figure 3.1 Mesh Refinement of Curved Mesh.

solution is presented in Eq. (A.180). Results from the convergence study are presented in Table 3.6. The exact value for the effective multiplication factor is $k_{ref} = 0.996711$.

It is observed in Table 3.6 that the convergence rate of the flux appears to fluctuate about the expected value of four. This is attributed to the process of refining circular meshes. Previously, the mesh was refined by simply introducing additional nodes at the midpoint between existing nodes. With a circular (or curved) boundary, this is not acceptable. It is necessary to totally regenerate the mesh for each refinement so that the problem boundary may be better approximated. For example, if only six nodes were on the problem boundary in the zeroth refinement and the traditional refinement procedure were used, the cylinder would be approximated as a hexagon throughout the mesh refinement study. Therefore, each mesh for the refinement study in Table 3.6 is independently generated. This mesh regeneration necessity means that the nodes are moving throughout the domain during refinement process. For an illustration of the nodes moving, see Fig. 3.1.

Table 3.6 shows that for refinement two, the refinement ratio is especially poor. However, for refinement three, the refinement ratio is roughly twice the expected rate. Therefore, on average, the flux appears to be converging at the correct rate despite difficulties with mesh regeneration.

Table 3.6 Finite Cylinder Convergence Study Results.

Refine	k_{eff}	k_{eff} error [pcm]	k_{eff} ratio	RMS	RMS ratio	$\ \mathbf{e}\ _\infty$	$\ \mathbf{e}\ _\infty$ ratio
0	0.895108	10160.26	4.18	5.34E-02	2.57	2.12E-01	1.62
1	0.972412	2429.90	4.16	2.07E-02	3.19	1.31E-01	4.65
2 [†]	0.990870	584.06	3.90	6.50E-03	1.85	2.81E-02	1.79
3	0.995215	149.61	3.99	3.51E-03	9.22	1.57E-02	8.28
4	0.996336	37.48		3.81E-04		1.90E-03	
Ref.	0.996711						

[†] Refinement ratio ≈ 1 but next case ≈ 8 .

This is due to the movement of mesh nodes in the process of circular mesh regeneration.

3.4 Two-Dimensional Benchmark Solutions

The work presented in this section was designed simulate nuclear power reactors with special attention to fast reactor applications. Though the target application is fast reactors, the general solution of the diffusion equation is applicable to any reactor type to which the diffusion approximation applies. For example, VVER reactors are thermal reactors but have a hexagonal geometry.

In this section, two-dimension nuclear reactor benchmark problems have been examined. These problems come from existing benchmarks based on hexagonal geometry. Initial refinement is based on six triangles per hexagon and these triangles are split with each refinement. This collection of benchmarks represents various energy group structures, geometries, assembly sizes, boundary conditions, as well as other properties. Data used in these problems is concisely presented in Appendix B. Mesh refinement studies are provided and convergence is observed relative to the reference k_{eff} value similar to analytic problems.

3.4.1 VVER440

Proposed by Chao & Shatilla [Cha95] and described in §B.2.1, this benchmark is a two-dimensional hexagonal problem based on a VVER-440 reactor. The VVER-440 is a Light Water Reactor (LWR) and, as such, operates principally with thermal neutron spectrum. Cross sections are provided for a two-group energy structure.

Power comparison between the most refined mesh and the reference solution are presented graphically in Fig. 3.2. Numerical mesh convergence study for the quantity k_{eff} is presented in Table 3.7.

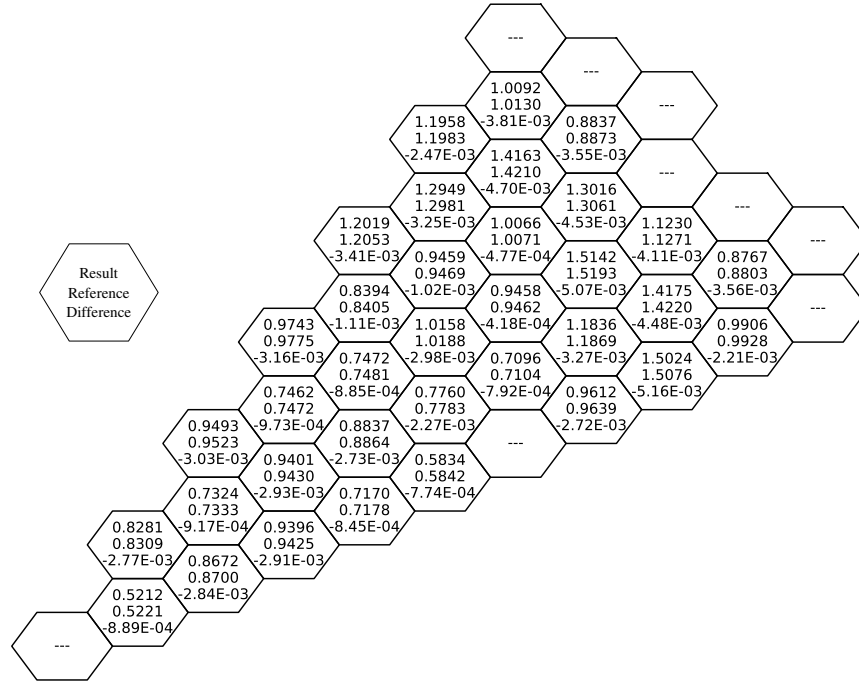


Figure 3.2 VVER440 Benchmark Power Comparison for Most Refined Mesh.

Table 3.7 VVER440 Benchmark Convergence Study.

Refine	k_{eff}	k_{eff} error [pcm]
0	1.005932	376.80
1	1.008980	72.00
2	1.009572	12.82
3	1.009666	3.35
4	1.009692	0.76
5	1.009698	0.22
Ref. [†]	1.009700	

[†] See [Cha95].

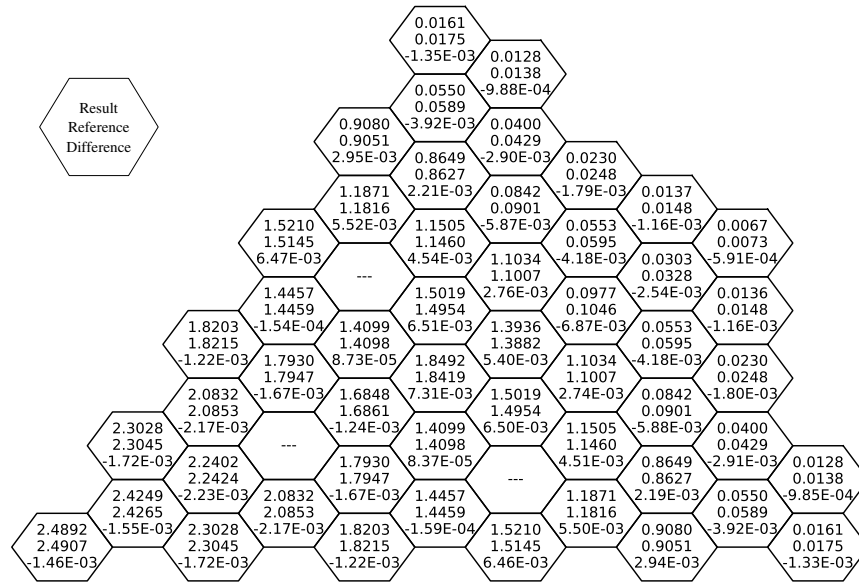


Figure 3.3 SNR Benchmark Power Comparison for Most Refined Mesh.

3.4.2 SNR

Proposed in the Argonne Code Center Benchmark Problem Book [Ame77] and described in §B.2.2, this benchmark is a two-dimensional problem based on the SNR reactor. The SNR is a Sodium-cooled Fast Reactor (SFR) and operates principally with fast neutrons. Cross sections are provided for a four-group energy structure.

Power comparison between the most refined mesh and numerical solution given by DIF3D are presented graphically in Fig. 3.3 (the benchmark does not specify a power distribution). A numerical mesh convergence study for the quantity k_{eff} is presented in Table 3.8.

Table 3.8 SNR Benchmark Convergence Study.

Refine	k_{eff}	k_{eff} error [pcm]
0	1.117613	638.71
1	1.122053	194.71
2	1.123376	62.42
3	1.123620	37.99
4	1.123703	29.69
5	1.123724	27.65
Ref. [†]	1.124	

[†] See [Ame77].

Table 3.9 HWR Benchmark Convergence Study.

Refine	k_{eff}	k_{eff} error [pcm]
0	0.989780	218.47
1	0.991303	66.16
2	0.991687	27.80
3	0.991783	18.19
4	0.991807	15.79
Ref. [†]	0.991965	

[†] See [Cha95].

3.4.3 HWR

Proposed by Chao & Shatilla [Cha95] and described in §B.2.3, this benchmark is a two-dimensional problem based on a large Heavy Water Reactor (HWR). This is a heavy water moderated reactor and operates principally with thermal neutrons. Cross sections are provided for a two-group energy structure.

Power comparison between the most refined mesh and the reference solution are presented graphically in Fig. 3.4. A numerical mesh convergence study for the quantity k_{eff} is presented in Table 3.9.

3.4.4 IAEA Hex

Proposed by Chao & Shatilla [Cha95] and described in §B.2.4, this two-dimensional problem was originally based on a two-dimensional Pressurized Water Reactor (PWR) with a Cartesian grid. The benchmark was converted to hexagonal geometry to represent a VVER reactor [Cha95]. As it is originally based on a PWR design, the reactor operates principally with a thermal neutron spectrum. Cross sections are provided for a two-group energy structure.



Figure 3.4 HWR Benchmark Power Comparison for Most Refined Mesh.

Table 3.10 IAEA Hex Benchmark Convergence Study. No Reflector. $\alpha = 0.125$.

Refine	k_{eff}	k_{eff} error [pcm]	k_{eff} ratio
0	0.973229	1814.91	
1	0.985766	561.18	
2	0.989840	153.83	
3	0.990767	61.13	
4	0.991127	25.15	
5	0.991262	11.64	
6	0.991321	5.73	
Ref. [†]	0.991378		

[†] See [Cha95].

Table 3.11 IAEA Hex Benchmark Convergence Study. No Reflector. $\alpha = 0.500$.

Refine	k_{eff}	k_{eff} error [pcm]
0	0.959683	1839.38
1	0.972744	533.29
2	0.976884	119.33
3	0.977698	37.91
4	0.977968	10.88
5	0.978043	3.44
6	0.978064	1.27
Ref. [†]	0.978077	

[†] See [Cha95].

The IAEA Hex reactor is presented in four scenarios; both with and without reflective assemblies as well as with albedo boundary condition $\alpha = 0.125$ and $\alpha = 0.5$. Numerical mesh convergence studies are presented for the quantity k_{eff} for each case in each of Table 3.10, Table 3.11, Table 3.12, and Table 3.13.

Table 3.12 IAEA Hex Benchmark Convergence Study. With Reflector. $\alpha = 0.125$.

Refine	k_{eff}	k_{eff} error [pcm]
0	1.000350	627.98
1	1.005640	99.05
2	1.006487	14.29
3	1.006607	2.31
4	1.006644	-1.44
5	1.006657	-2.75
6	1.006663	-3.29
Ref. [†]	1.006630	

[†] See [Cha95].

Table 3.13 IAEA Hex Benchmark Convergence Study. With Reflector. $\alpha = 0.500$.

Refine	k_{eff}	k_{eff} error [pcm]	k_{eff} ratio
0	0.996237	927.02	
1	1.003760	174.74	
2	1.005185	32.16	
3	1.005404	10.30	
4	1.005472	3.47	
5	1.005494	1.25	
6	1.005503	0.42	
Ref. [†]	1.005507		

[†] See [Cha95].

3.5 Three-Dimensional Benchmark Solutions

For two-dimensional benchmarks, mesh refinement studies are provided and convergence is observed relative to the reference k_{eff} value similar to analytic problems. For three-dimensional benchmarks, control rod worth measurement is used to observe agreement of the FEM neutron diffusion solution to the benchmark solution. To measure control rod worth, three cases are simulated, $\{A, B, C\}$, with control rods fully removed in case A , partially inserted in case B , and fully inserted in case C . Rod worth is presented in units $[\Delta k]$ and calculated as

$$\text{Rod Worth}_x [\Delta k] = \frac{k_{eff,A} - k_{eff,x}}{k_{eff,A} k_{eff,x}} \quad (3.6)$$

for $x = \{B, C\}$. That is, rod worth is always compared to the case with control rods fully removed, case A . Additionally, Rod Difference is presented in units $[\% \Delta k]$ as

$$\text{Rod Difference}_x [\% \Delta k] = (k_{eff,A} - k_{eff,x}) \times 100\% \quad (3.7)$$

for $x = \{B, C\}$.

3.5.1 MONJU

Proposed by Komano et al. [Kom78] and described in §B.3.1, this three-dimension fast reactor benchmark problem is based on an SFR operating principally with fast neutrons. Cross sections are provided for a three-group energy structure. However, the fission spectrum, χ , is not provided in the specifications. Instead, a fission spectrum is selected for benchmark agreement. This assumed fission spectrum is presented in Table B.13. The rod worth measurements appear insensitive to the choice of fission spectrum.

Reference data can be found in Table B.11. Results from the FEM implementation are presented in Table 3.14. The error (difference between reference and calculated results) are presented in parenthesis next to the quantities. Rod worth is presented as calculated in Eq. (3.6). Rod difference is presented as calculated in Eq. (3.7).

3.5.2 KNK

Proposed by Takeda & Ikeda [Tak91] and described in §B.3.2, this three-dimension benchmark problem is based on an SFR and is a model of the KNK-II core. Cross sections are provided in a four-group energy structure. There are many materials specified in the problem so it also aids in testing material mapping in the code. The reference data is for a solution to the neutron *transport* equation. In this work, the cross sections from the transport problem are used to solve the neutron diffusion equation. This explains some of the numerical differences in the results but general trends are reflected.

Table 3.14 MONJU Benchmark Rod Worth Results.

Pattern	k_{eff}	Rod Worth [Δk]	Rod Difference [$\% \Delta k$]
A	1.056816		
B	1.031623	0.023 (2.51E-5) [†]	2.52 (-0.07)
C	1.006519	0.047 (1.77E-3)	5.03 (0.04)

[†] Value in parentheses is difference to reference value from [Kom78] as presented in §B.3.1.

Table 3.15 KNK Benchmark Rod Worth Results.

Pattern	k_{eff}	Rod Worth [Δk]	Rod Difference [$\% \Delta k$]
A	1.061752		
B	0.942404	0.119 (1.55E-2) [†]	11.93 (0.75)
C	0.829829	0.263 (3.99E-2)	23.19 (1.67)

[†] Value in parentheses is difference to reference value from [Tak91] as presented in §B.3.2.

Reference data can be found in Table B.14. Results from the FEM neutron diffusion solution are presented in Table 3.15. The error (difference between reference and calculated results) are presented in parenthesis next to the quantities. Rod worth is presented as calculated in Eq. (3.6). Rod difference is presented as calculated in Eq. (3.7).

CHAPTER

4

THERMAL HYDRAULICS

4.1 Introduction

Material temperatures and densities are a pivotal part of the multiphysics problem of simulating a fast reactor. Temperatures are necessary for calculating temperature-dependent neutron cross sections for an accurate power distribution. Thermal hydraulic models are developed for standard fast reactor geometry as described in §1.2. Recall, the standard assembly is a hexagonal can and filled with many cylindrical rods.

In this simulation, two thermal hydraulic models are developed. The first is a steady-state, one-dimensional, axial heat convection model to calculate bulk coolant temperatures and densities as the coolant flows vertically through a channel. This model assumes no cross-flow between channels and perfect fluid mixing within the flow channel. These assumptions are valid for simulating typical fast reactors with hexagonal assemblies. Hexagonal assemblies, as described in §1.2, satisfy these assumptions with canned assemblies to prevent cross-flow between channels and mixing encouraged by wire wrapping. The second thermal hydraulic model is a steady-state, one-dimensional, radial heat conduction model within a cylindrical rod to calculate average cladding, bond, and fuel temperatures.

4.2 Material Properties

Thermodynamic properties of reactor materials are required for thermal hydraulic models. The coolant properties required are density, enthalpy, thermal conductivity, dynamic viscosity, and heat capacity

Table 4.1 Default Constant Thermal Conductivity for Sodium and HT9.

Material	$k \left[\frac{\text{W}}{\text{m K}} \right]$
Sodium	64.33
HT9	25.81

required. In this thesis, only sodium coolant is considered [Fin95]. However, these thermal hydraulic properties can be easily changed to allow for the simulation of fast reactors with other coolants such as lead or molten salt. Thermal conductivity values are also required for the cladding and fuel material. Typical cladding for fast reactor designs is HT9 stainless-steel [Lei88]. Both sodium and steel have relatively high thermal conductivities and their thermal conductivities do not change significantly over the operating temperatures of sodium cooled fast reactors. Therefore, the thermal conductivity of sodium within the bond [Fin95] and HT9 [Lei88] are assumed constant at the average reactor operating value. The user can input thermal conductivity constants and the default values are given in Table 4.1.

Fuel composition is assumed to be metallic fuel of the form UXZr where X is the weight fraction of Zr in the fuel. A typical value for X is 10% and this is assumed for this work. Fuel thermal conductivity is given as a function of material temperature and zirconium weight fraction. A plot of thermal conductivity for U10Zr as a function of temperature is given in Fig. 4.1. The functional form of the thermal conductivity for U-XZr is given in Eq. (4.1) through Eq. (4.4) [Kim14].

$$k_U = 21.73 + 1.591 \times 10^{-2}T + 5.907 \times 10^{-6}T^2 \quad (4.1)$$

$$k_{Zr} = 8.853 + 7.082 \times 10^{-3}T + 2.533 \times 10^{-6}T^2 + 2.992 \times 10^3T^{-1} \quad (4.2)$$

$$k_{c,Zr} = -102.0 + 200.1x_{Zr} - 109.2x_{Zr}^2 + 9.435 \times 10^{-3}T + 3.459 \times 10^{-5}T^2 - 0.02093x_{Zr}T \quad (4.3)$$

$$k_{U-Zr} = \left(1 - \sqrt{1 - x_{Zr}}\right) k_{Zr} + \sqrt{1 - x_{Zr}} \left((1 - x_{Zr}) k_U + x_{Zr} k_{c,Zr}\right) \quad (4.4)$$

In the expressions for fuel thermal conductivity, x_{Zr} represents the zirconium weight fraction (e.g. $x_{Zr} = 0.1$), all temperatures are in units [K] and thermal conductivities are in units $\left[\frac{\text{W}}{\text{m K}} \right]$. For the given expression of fuel thermal conductivity, the integral of thermal conductivity is unbounded as $T \rightarrow 0$ so thermal conductivity is assumed constant below 300 [K] which is below the melting point of most fast reactor coolants, including sodium, so this assumption is valid for sodium-cooled reactor applications.

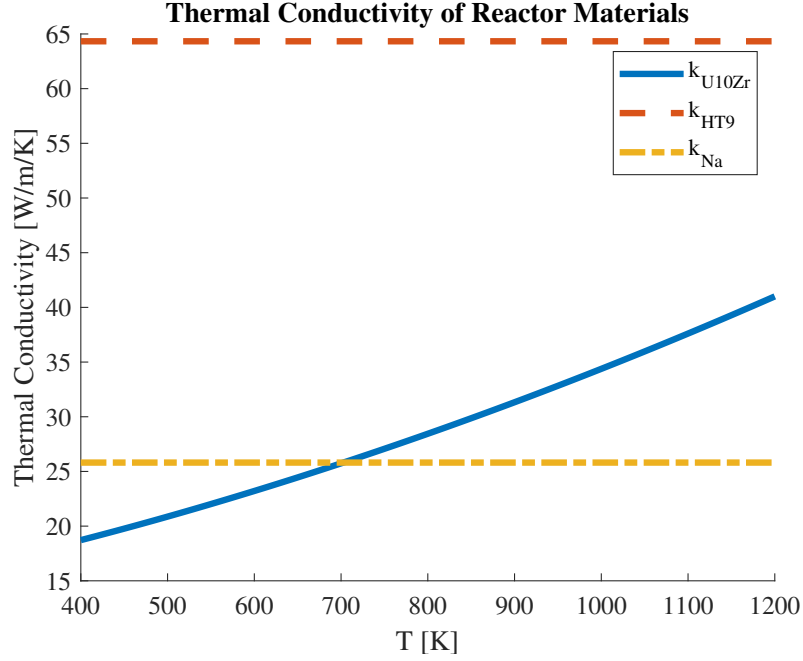


Figure 4.1 Variable Thermal Conductivity in Fuel.

4.3 Power Normalization

Recall the multigroup neutron diffusion equation solved via the power iteration method returns the largest eigenvalue k_{eff} and unique positive eigenvector (see §2.4). The neutron flux calculated according to this method is an eigenvector and can be normalized to an arbitrary constant. The flux is normalized to a given, user input reactor power Q_{Rx} . The flux is normalized such that

$$Q_{Rx} = \sum_{e=1}^{N_E} \sum_{g=1}^G \kappa \Sigma_{f,g,e} \bar{\phi}_{g,e} V_e \quad (4.5)$$

where N_E is the number of elements in the reactor and V_e is the volume of element e . In Eq. (4.5), the quantity κ represents the reclaimable (non-neutrino) energy produced per fission such that the quantity $\kappa \Sigma_f \bar{\phi}$ represents the volumetric heat generation rate. The total power generated in each element is

$$q_e = \sum_g^G \kappa \Sigma_{f,g,e} \bar{\phi}_{g,e} V_e. \quad (4.6)$$

The quantity q_e has units [W]. Assuming all heat generated in the element is generated within the fuel, the volumetric heat generation rate within the fuel is

$$q_e''' = \frac{q_e}{V_{fuel,e}} \quad (4.7)$$

where $V_{fuel,e}$ is the volume of fuel in element e (the volume of fuel is summed over all rods). The quantity q_e''' has units $\left[\frac{\text{W}}{\text{cm}^3} \right]$ and will be necessary for the radial conduction model. This assumption that all heat generation occurs in the fuel is discussed further in §4.5. Briefly, the reactor power Q_{Rx} is decreased by 2% to account for heat generated elsewhere in the reactor.

4.4 Axial Convection Model

In this model, the temperature and density of the coolant is calculated as it flows upward through a single hexagonal assembly. The axial thermal hydraulic model assumes one-dimensional, steady-state flow in each channel.

4.4.1 Geometric Model

Used in association with an unstructured mesh, the thermal hydraulic model requires mapping mesh elements into physical reactor assemblies. In the input geometry file, the user must specify to which hexagonal assembly each element belongs. Flow channels will have multiple finite elements in the radial direction. Only assembly-average thermal properties are calculated so each hexagonal assembly is equivalent to a one-dimensional flow channel. Subsequently, each assembly is also referred to as a “channel.” In the following discussion, a channel index is subscripted i for $i = 1, 2, \dots, N_{chan}$ where N_{chan} is the number of hexagonal assemblies in the simulated reactor.

The concept of “chunks” is also introduced for use in the discretization of the axial heat convection model. A chunk is the set of all elements in a channel with a unique axial elevation. For example, in a fast reactor hexagonal assembly, the assembly has a unique channel index, i , and contains a number of chunks equal to the number of axial elevations in the discretized model. Additionally, each chunk is required to have unique material composition. The relationship between elements, chunks, and channels is shown in Fig. 4.2. Elevations are indexed $k = 1, 2, \dots, N_z$ where N_z is the total number of axial elevation. Then, each chunk has a unique $\{i, k\}$ combination and the total number of chunks is the product $N_{chunk} = N_{chan} N_z$. The alignment of chunks in a one-dimensional flow channel is shown in Fig. 4.3.

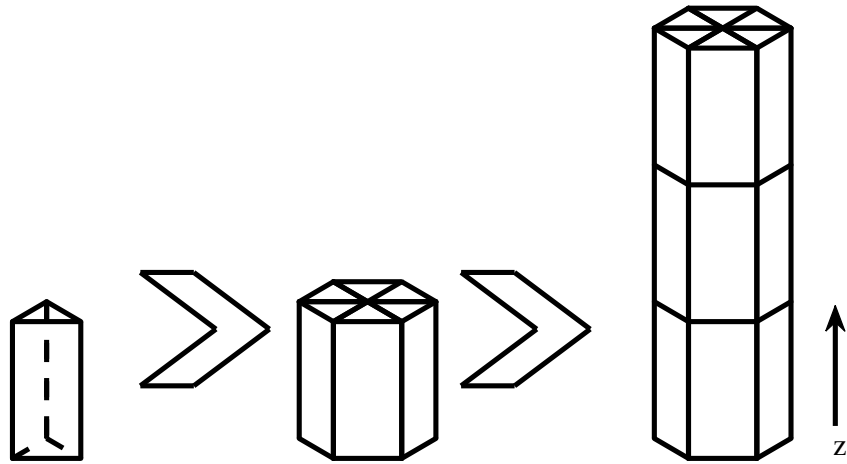


Figure 4.2 Progression of Element, to Chunk, to Channel.

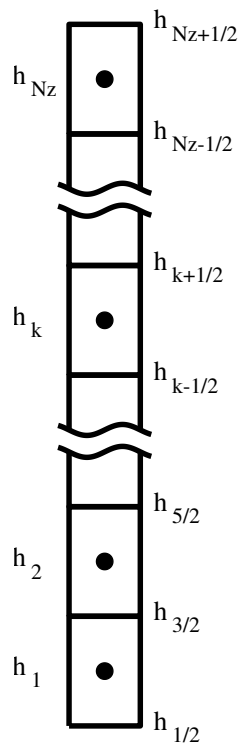


Figure 4.3 One-Dimensional Axial Heat Convection Model Description.

4.4.2 Channel Mass Flow

The channel mass flow, \dot{m}_i , is calculated given a user specified total reactor inlet mass flow rate \dot{m}_{Rx} . Mass flow is partitioned into each channel assuming uniform mass flux at each channel inlet according to

$$\dot{m}_i = \dot{m}_{Rx} \frac{A_{cool,i}}{A_{cool,Rx}} \quad (4.8)$$

where $A_{cool,i}$ is the coolant flow area for channel i and $A_{cool,Rx}$ is the total coolant flow area for the reactor. $A_{cool,i}$ includes the area between flow channels and Eq. (4.8) accounts for assembly bypass flow. The mass flow per unit area is assumed to be uniform at the reactor inlet and the mass flow in a channel is the product of the reactor average mass flux and the channel flow area. This partitioning of mass flow rate is chosen to allow for assemblies with different flow areas or the modeling of fractional assemblies in fractional core models.

4.4.3 Chunk Powers

For the one-dimensional axial heat convection model in the axial direction, heat generation quantities are needed for chunks. Heat generation quantities for elements are calculated in Eq. (4.6) and Eq. (4.7). The total heat generated in a chunk, $q_{i,k}$, and the average volumetric heat generation rate in the fuel within the chunk, $q'''_{i,k}$ are required. These relationships are given in Eq. (4.9) and Eq. (4.10) respectively. The notation $e \in \{i, k\}$ implies the summation over all elements e within the chunk in channel i at elevation k .

$$q_{i,k} = \sum_{e \in \{i,k\}} q_e \quad (4.9)$$

$$q'''_{i,k} = \frac{\sum_{e \in \{i,k\}} q'''_e V_{fuel,e}}{\sum_{e \in \{i,k\}} V_{fuel,e}} \quad (4.10)$$

4.4.4 Channel Enthalpy

It is assumed that all heat generated within the chunk is deposited in the coolant as it flows through the chunk. This occurs at steady-state with the assumption that axial heat conduction in the solids can be neglected. Then, the steady-state coolant enthalpy for an axial location z within the channel is expressed by a simple energy balance relation as

$$h_i(z) = h_{in} + \frac{1}{\dot{m}_i} \int_0^z q'_i(z') dz' \quad (4.11)$$

where $h_i(z)$ is the specific enthalpy at axial position z in channel i , h_{in} is the channel inlet enthalpy, \dot{m}_i is the mass flow rate within the channel given in Eq. (4.8), and $q'_i(z)$ is the linear heat generation rate

for channel i at elevation z . h_{in} is related to a user specified coolant inlet temperature, T_{inlet} , by a state relationship for the coolant $h_{in} = h(T_{inlet})$ [Fin95].

The integral in Eq. (4.11) can be discretized along the channel and converted to a summation. This is similar to a Riemann summation approximation to the integral. In this discretization, the model in Fig. 4.3 is used where $h_{i,k+1/2}$ is the enthalpy at the upper edge of the chunk and $h_{i,k}$ is the average enthalpy within the chunk. Then, the discretization of the integral Eq. (4.11) is

$$h_{i,k+1/2} = h_{in} + \frac{1}{\dot{m}_i} \sum_{k'=1}^k q'_{i,k'} \Delta z_{k'} \quad (4.12)$$

where $q'_{i,k}$ is the linear heat generation rate in the chunk in channel i at elevation k and Δz_k is the height of the node at elevation k such that $\Delta z_k = z_{k+1/2} - z_{k-1/2}$. Recognizing the quantity $q_{i,k} = q'_{i,k} \Delta z_{i,k}$ is the total heat generated in chunk $\{i, k\}$, then Eq. (4.12) can be rewritten.

$$h_{i,k+1/2} = h_{in} + \frac{1}{\dot{m}_i} \sum_{k'=1}^k q_{i,k'} \quad (4.13)$$

The quantity $h_{i,k+1/2}$ represents the enthalpy in channel i at the upper coordinate of the one-dimensional heat convection chunk (see Fig. 4.3). The node-average enthalpy is desired instead of the enthalpy at the upper elevation of the node. To first-order approximation, then

$$h_{i,k} = \frac{1}{2} (h_{i,k-1/2} + h_{i,k+1/2}) \quad (4.14)$$

where $h_{i,k}$ is the average enthalpy in the chunk located in channel i at axial level k . The final result of this model is $h_{i,k}$, the bulk coolant enthalpy in each chunk. Given, $h_{i,k}$, the average bulk coolant temperature in the chunk $T_{\infty,i,k} = T(h_{i,k})$ can be calculated using a state relationship [Fin95]. Bulk coolant temperature will be used in the radial heat conduction model and to update coolant density and cross sections.

4.5 Radial Conduction Model

The radial heat conduction model calculates the steady-state heat conduction from its generation in the fuel to the coolant. This model represents an average fuel rod in each fuel assembly. The model explicitly treats fuel, sodium bond, and cladding regions. In the radial heat conduction model, it is assumed that all heat generated in the reactor is due to fission as described by the volumetric heat generation rate $q''' = \kappa \Sigma_f \phi$. This assumption asserts that all deposited energy is described by the coefficient κ . However, roughly 2% of power in a typical fast reactor is transported via gamma radiation and will be deposited

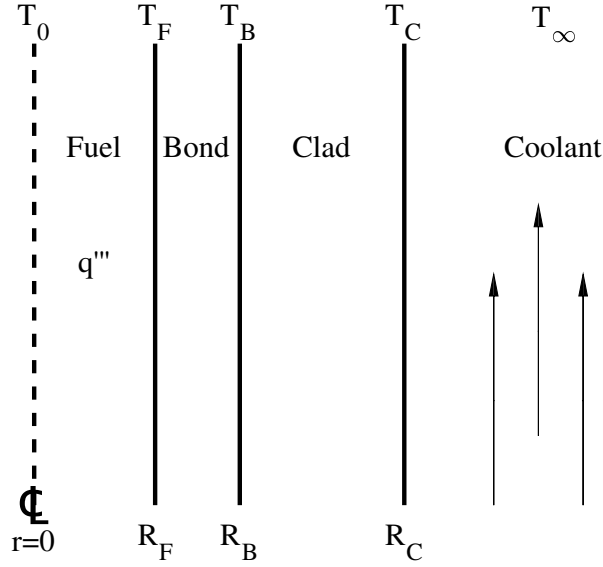


Figure 4.4 Geometry Description of Radial Heat Conduction Model (not to scale).

directly in the coolant and does not need to be conducted. Effectively, q''' is reduced by this fraction in the radial heat conduction model [Wal11].

For this model, it is assumed that the bulk coolant temperature, $T_{\infty,i,k}$, is known within the chunk as calculated in §4.4. Then, the model begins by calculating temperatures at key locations. Material temperature is calculated at the fuel centerline, fuel surface, bond surface, and clad surface. The derivation of these quantities are presented in §4.5.2. Similar results have been obtained by others [Wal11]. With the temperatures at these locations calculated, the average temperature within each material is calculated. These average temperatures are essential for the accurate simulation of the reactor as they influence multigroup neutron cross sections.

4.5.1 Geometric Model

For the purposes of the radial model, the geometry is shown in Fig. 4.4. This model represents a cylindrical fuel pellet, surrounded by sodium bond, enclosed in steel cladding, with sodium coolant flowing in the axial direction. The center of the fuel rod is located at $r = 0$ where r is the radial coordinate. The fuel pellet has radius R_F and fuel is located in $r \in (0, R_F)$. Then, bond is located in $r \in (R_F, R_B)$ and clad is located in $r \in (R_B, R_C)$. The fuel centerline temperature is T_0 and fuel surface temperature is T_F . Bond surface temperature is T_B and clad surface temperature is T_C . T_{∞} represents the bulk coolant temperature. In this model, heat is generated exclusively in the fuel with volumetric heat generation rate $q'''_{i,k}$.

4.5.2 Surface Temperature and Centerline Temperatures

Temperatures at selected locations are calculated using the steady-state heat conduction equation with material properties described in §4.2. The process of the derivation moves outward from the fuel centerline toward the coolant flow, always assuming that the next surface temperature is known. The derivation begins with the fuel centerline temperature, assuming the fuel surface temperature is known. Then, the derivation moves to the fuel surface temperature assuming the bond surface temperature is known. This pattern is terminated and the relationship is closed at the cladding surface where, for a given $T_{\infty,i,k}$, the clad surface temperature is described by Newton's law of cooling and a convective heat transfer correlation.

With the derivation complete, the temperatures of interest are calculated. While the derivation moved from fuel centerline outward to clad surface, the calculation moves from clad surface inward to fuel centerline. This is necessary because the quantities are all unknown with the exception of $T_{\infty,i,k}$ so the calculation begins there. Surface temperatures are all described explicitly due to the assumption of constant thermal conductivity within bond and clad regions. Fuel centerline temperature must be solved implicitly due to the treatment of variable thermal conductivity. A numerical solution method is used to solve for the fuel centerline temperature.

4.5.2.1 Temperature in Fuel

The fueled region is the only region modeled with non-zero volumetric heat generation. In this region, $q'''_{i,k}$ as specified by Eq. (4.10) is assumed constant within the fuel. Additionally, the thermal conductivity in the fuel is assumed to have the general, functional form of $k_F(T)$ [Kim14]. For the assumed fuel composition of U10Zr, the thermal conductivity is plotted in Fig. 4.1.

The steady-state heat conduction equation for the fuel temperature $T_F(r)$, with constant volumetric heat generation rate $q'''_{i,k}$, and variable thermal conductivity $k_F(T)$ can be written.

$$\nabla \cdot (k_F(T_F(r)) \nabla T_F(r)) + q'''_{i,k} = 0 \quad (4.15)$$

Noting the gradient operator in one-dimensional cylindrical geometry, Eq. (4.15) is rewritten with derivatives.

$$\frac{1}{r} \frac{d}{dr} \left(r k_F(T_F(r)) \frac{dT_F}{dr} \right) + q'''_{i,k} = 0 \quad (4.16)$$

Begin solving Eq. (4.16) by multiplying the equation by radial coordinate r .

$$\frac{d}{dr} \left(r k_F(T_F(r)) \frac{dT_F}{dr} \right) + q'''_{i,k} r = 0 \quad (4.17)$$

Integrate the equation for $r \in [0, r']$ where r' is an arbitrary location $r' \in [0, R_F]$.

$$\int_0^{r'} \left(\frac{d}{dr} \left(r k_F(T_F(r)) \frac{dT_F}{dr} \right) + q_{i,k}''' r \right) dr = 0 \quad (4.18)$$

$$\left[r k_F(T_F(r)) \frac{dT_F}{dr} \right]_{r=0}^{r=r'} + \frac{q_{i,k}'''}{2} r^2 \Big|_{r=0}^{r=r'} = 0 \quad (4.19)$$

Using the boundary condition that the temperature of the fuel at the centerline is required to be finite,

$$\frac{dT_F}{dr} \Big|_{r=0} \text{ is finite} \quad (4.20)$$

the evaluations at $r = 0$ are zero and Eq. (4.19) can be evaluated.

$$r' k_F(T_F(r')) \frac{dT_F(r')}{dr} + \frac{q_{i,k}'''}{2} r'^2 = 0 \quad (4.21)$$

Divide by the location r' .

$$k_F(T_F(r')) \frac{dT_F(r')}{dr} + \frac{q_{i,k}'''}{2} r' = 0 \quad (4.22)$$

Next, integrate $r' \in [0, r]$.

$$\int_0^r \left(k_F(T_F(r')) \frac{dT_F(r')}{dr} + \frac{q_{i,k}'''}{2} r' \right) dr' = 0 \quad (4.23)$$

$$\int_0^r k_F(T_F(r')) \frac{dT_F(r')}{dr} dr' + \frac{q_{i,k}'''}{4} r^2 = 0 \quad (4.24)$$

The fundamental theorem of calculus allows for the expression of the first integral as

$$\int_{T_F(0)}^{T_F(r)} k_F(T_F(r)) dT + \frac{q_{i,k}'''}{4} r^2 = 0. \quad (4.25)$$

In this derivation, $k_F(T)$ is allowed to be a generic function. Therefore, Eq. (4.25) does not have a simple forward expression. Solving for $T_F(0)$ will require a non-linear search, such as the bisection method. Define the conductivity integral of the fuel conductivity in the form as

$$K_F(T) = \int_0^T k_F(T') dT' \quad (4.26)$$

where $k_F(T)$ is the thermal conductivity of the fuel material as plotted in Fig. 4.1. Then, Eq. (4.25) can be rewritten

$$K_F(T_F(r)) - K_F(T_F(0)) + \frac{q_{i,k}'''}{4}r^2 = 0, \quad (4.27)$$

$$K_F(T_F(0)) = K_F(T_F(r)) + \frac{q_{i,k}'''}{4}r^2, \quad (4.28)$$

$$K_F(T_F(0)) = K_F(T_F) + \frac{q_{i,k}'''}{4}R_F^2, \quad (4.29)$$

where $T_F(0)$ is the fuel centerline temperature, $T_F = T_F(R_F)$ is the fuel surface temperature, and R_F is the radius of the fuel. The fuel centerline temperature can be calculated using Eq. (4.29) given a functional form of the conductivity integral $K_F(T)$ and by employing a nonlinear search. In this work, the thermal conductivity of U10Zr given in Eq. (4.1) through Eq. (4.4) is analytically integrated using a symbolic math computer program and the fuel conductivity integral, $K_F(T)$, is stored in the code. Then, Eq. (4.29) is solved using the bisection method.

If, instead, $k_F(T)$ were constant such that $k_F(T) = \overline{k_F}$; then, Eq. (4.29) would be equivalent to

$$T_F(r) = T_F(0) - \frac{q_{i,k}'''}{4\overline{k_F}}r^2, \quad (4.30)$$

$$T_F(0) = T_F + \frac{q_{i,k}'''}{4\overline{k_F}}R_F^2, \quad (4.31)$$

where $\overline{k_F}$ is the assumed constant thermal conductivity in the fuel region.

4.5.2.2 Temperature in Sodium Bond

The fuel surface temperature is calculated by considering the heat conduction equation in the sodium bond region. There is no heat generation in the bond region. The thermal conductivity in the bond is assumed constant such that $k_B(T) = k_B$. The default value for k_B is $64.33 \left[\frac{\text{W}}{\text{m K}} \right]$ as specified in Table 4.1. The steady-state heat conduction equation with no heat generation is written as

$$\nabla \cdot (k_B \nabla T_B(r)) = 0 \quad (4.32)$$

where $T_B(r)$ is the temperature within the bond region. In this region, good thermal contact between the fuel and bond is assumed such that

$$T_F(R_F) = T_B(R_F). \quad (4.33)$$

That is, the temperature is continuous at the material discontinuity. Additionally, constant heat flux is assumed at the material discontinuity such that

$$k_F(T_F(R_F)) \left. \frac{dT_F}{dr} \right|_{r=R_F} = k_B \left. \frac{dT_B}{dr} \right|_{r=R_F}. \quad (4.34)$$

Recognizing one-dimensional cylindrical geometry, Eq. (4.32) can be rewritten.

$$\frac{1}{r} \frac{d}{dr} \left(r k_B \frac{dT_B}{dr} \right) = 0 \quad (4.35)$$

Begin solving Eq. (4.35) by multiplying the equation by the radial coordinate r .

$$\frac{d}{dr} \left(r k_B \frac{dT_B}{dr} \right) = 0 \quad (4.36)$$

Integrate the equation for $r \in [R_F, r']$ where r' is an arbitrary location $r' \in [R_F, R_B]$.

$$\int_{R_F}^{r'} \frac{d}{dr} \left(r k_B \frac{dT_B}{dr} \right) dr = 0 \quad (4.37)$$

$$r' k_B \frac{dT_B(r')}{dr} - R_F k_B \left. \frac{dT_B}{dr} \right|_{r=R_F} = 0 \quad (4.38)$$

The assumption of constant heat flux at the material boundary given in Eq. (4.34) allows for the treatment of the spatial derivative at R_F . Recall from the derivation within the fuel region in §4.5.2.1, the expression Eq. (4.21) is exploited. The expression is valid for any $r' \in [0, R_F]$ so allow $r' = R_F$.

$$R_F k_F(T_F(R_F)) \left. \frac{dT_F}{dr} \right|_{r=R_F} + \frac{q_{i,k}'''}{2} R_F^2 = 0 \quad (4.39)$$

$$R_F k_F(T_F(R_F)) \left. \frac{dT_F}{dr} \right|_{r=R_F} = -\frac{q_{i,k}'''}{2} R_F^2 \quad (4.40)$$

The quantity $q_{i,k}'''$ is the average volumetric heat generation rate within the fuel in chunk $\{i, k\}$ specified in Eq. (4.10). Eq. (4.40) is substituted into Eq. (4.38).

$$r' k_B \frac{dT_B(r')}{dr} + \frac{q_{i,k}'''}{2} R_F^2 = 0 \quad (4.41)$$

Divide Eq. (4.38) by the radial coordinate r' . This is valid because $r' \neq 0$ in this region.

$$k_B \frac{dT_B}{dr} + \frac{q_{i,k}'''}{2} \frac{R_F^2}{r'} = 0 \quad (4.42)$$

Integrate the equation for $r' \in [R_F, r]$ where $r \in [R_F, r']$.

$$\int_{R_F}^r \left(k_B \frac{dT_B(r')}{dr} + \frac{q_{i,k}'''}{2} \frac{R_F^2}{r'} \right) dr' = 0 \quad (4.43)$$

$$k_B \int_{R_F}^r \frac{dT_B(r')}{dr} dr' + \frac{q_{i,k}'''}{2} R_F^2 \ln(r') \Big|_{r'=R_F}^{r'=r} = 0 \quad (4.44)$$

$$k_B \int_{R_F}^r \frac{dT_B(r')}{dr} dr' + \frac{q_{i,k}'''}{2} R_F^2 \ln \left(\frac{r}{R_F} \right) = 0 \quad (4.45)$$

Again, the remaining integral can be rewritten by employing the fundamental theorem of calculus.

$$k_B (T_B(r) - T_B(R_F)) + \frac{q_{i,k}'''}{2} R_F^2 \ln \left(\frac{r}{R_F} \right) = 0 \quad (4.46)$$

Then, solving for $T_B(R_F)$,

$$T_B(r) = T_B(R_F) - \frac{q_{i,k}'''}{2k_B} R_F^2 \ln \left(\frac{r}{R_F} \right). \quad (4.47)$$

Evaluate at R_B and rearrange,

$$T_B(R_F) = T_B(R_B) + \frac{q_{i,k}'''}{2k_B} R_F^2 \ln \left(\frac{R_B}{R_F} \right). \quad (4.48)$$

The fuel surface temperature can be calculated using Eq. (4.48) for constant thermal conductivity k_B . In Eq. (4.48), the bond surface temperature must be given to calculate the fuel surface temperature.

4.5.2.3 Temperature in Cladding

Consider the heat conduction equation in the cladding region. Derivation of the bond surface temperature $T_B = T_B(R_B)$ is similar to the fuel surface temperature because both consider a heat conduction equation with no heat generation and constant thermal conductivity. The thermal conductivity in the clad is assumed constant such that $k_C(T) = k_C$. The default value for k_C is $25.81 \left[\frac{\text{W}}{\text{m K}} \right]$ as specified in Table 4.1. Using the expression for temperature within the bond, Eq. (4.48), the temperature within the cladding can be expressed similarly by changing material subscripts.

$$T_C(r) = T_C(R_B) - \frac{q_{i,k}'''}{2k_C} R_F^2 \ln \left(\frac{r}{R_B} \right) \quad (4.49)$$

$$T_C(R_B) = T_C(R_C) + \frac{q_{i,k}'''}{2k_C} R_F^2 \ln \left(\frac{R_C}{R_B} \right) \quad (4.50)$$

The bond surface temperature can be calculated using Eq. (4.50) for constant thermal conductivity k_C . In Eq. (4.50), the clad surface temperature must be given to calculate the bond surface temperature.

4.5.2.4 Clad Surface Temperature

Clad surface temperature T_C is given by Newton's Law of Cooling with convective heat transfer coefficient H_c specified according to the Subbotin-Ushakov correlation [Pfr07]. Newton's Law of Cooling may be written as

$$q''_{clad} = H_c(T_C - T_\infty) \quad (4.51)$$

where q''_{clad} is the heat flux at the clad surface R_C , H_c is the convective heat transfer coefficient, T_C is the clad surface temperature, and T_∞ is the bulk coolant temperature. Using the axial heat convection model in §4.4, $T_{\infty,i,k}$ is given by the state relationship $T_{\infty,i,k} = T(h_{i,k})$ [Fin95]. The heat flux at the clad surface is related to the volumetric heat generation rate in the fuel according to

$$q''_{clad} = q'''_{i,k} \frac{R_F^2}{2R_C} \quad (4.52)$$

where $q'''_{i,k}$ is the chunk-average volumetric heat generation rate in the fuel according to Eq. (4.10).

The coefficient H_c must be calculated via an applicable correlation. The Subbotin-Ushakov correlation is valid for coolant in heavy liquid metal cooled reactor designs with triangular pitch assemblies. This correlation is selected because it is validated by experimental data and is applicable for a wide range of operating conditions common to metal cooled fast reactors [Pfr07]. The correlation relates the Péclet number, Pe , to the Nusselt number, Nu . The Péclet number is the product of the Prandlt number, Pr , and the Reynolds number, Re .

$$Pe = Re Pr \quad (4.53)$$

The Subbotin-Ushankov correlation is given as

$$Nu = 7.55 \frac{S}{D} - 20 \left(\frac{S}{D} \right)^{-13} + \frac{3.67}{90 \left(\frac{S}{D} \right)^2} Pe^{(0.56+0.19 \frac{S}{D})} \quad (4.54)$$

where S is the rod-pitch within the assembly and D is the outer diameter of the cladding. The Subbotin-Ushankov correlation is valid for $1 < Pe < 4,000$ and $1.2 \leq S/D \leq 2.0$ [Pfr07].

Calculating the nondimensional numbers begins with the calculation of cross-sectional flow area and wetted perimeter. As all properties calculated are the channel average properties, the flow area and wetted perimeters are calculated for the average fuel rod [Wal11]. Then, the cross-sectional flow area is

$$A_x = \left(\frac{\sqrt{3}}{4} S^2 - \frac{\pi R_C^2}{2} - \frac{\pi D_{wrap}^2}{8} \right) N_{rod} \quad (4.55)$$

where N_{rod} is the number of rods in the assembly and D_{wrap} is the diameter of the wire wrap around the cylindrical rod. The wetted perimeter is

$$P_w = \left(\pi R_C + \frac{\pi D_{wrap}}{2} \right) N_{rod}. \quad (4.56)$$

Then, the effective flow diameter is defined as

$$D_e = \frac{4 A_x}{P_w} \quad (4.57)$$

and the mass-flux in the flow channel is

$$G_i = \frac{\dot{m}_i}{A_x}. \quad (4.58)$$

The Reynolds number is defined as

$$Re = \frac{G D_e}{A_x \mu} \quad (4.59)$$

where \dot{m}_i is the assembly mass flow rate, μ is the fluid's dynamic viscosity given by a state relationship [Fin95], and D_e and A_x are defined according to Eq. (4.57) and Eq. (4.55) respectively. The Prandtl number is defined according to

$$Pr = \frac{c_p \mu}{k} \quad (4.60)$$

where c_p is the fluid specific heat capacity at constant pressure, μ is the fluid dynamic viscosity, and k is the fluid thermal conductivity. Note that each of these quantities are given by state relationships. As such, the Prandtl number can itself be correlated for a given fluid as a state relationship.

With the dimensionless quantities expressed in Eq. (4.59) and Eq. (4.60), the Subotin-Ushakov correlation is used to calculate the Nusselt number. Given a Nusselt number, the convective heat transfer coefficient is defined as

$$H_c = \frac{Nu k}{D_e} \quad (4.61)$$

where Nu is the Nusselt number from the Subotin-Ushakov correlation Eq. (4.54), k is the fluid thermal conductivity given by a state relationship, and D_e is the effective flow diameter given in Eq. (4.57). Finally, the clad surface temperature is expressed.

$$T_C = \frac{q''_{clad}}{H_c} + T_{\infty,i,k} \quad (4.62)$$

$$T_C = q'''_{i,k} \frac{R_F^2}{2 R_c H_c} + T_{\infty,i,k} \quad (4.63)$$

With Eq. (4.63), all surface temperatures and the fuel centerline temperature have been related to $T_{\infty,i,k}$ and the calculation of these quantities may begin, moving from the coolant toward the fuel centerline.

4.5.3 Average Temperatures

Average material temperatures are used to evaluate material properties such as densities, temperature-dependent neutron cross sections, and thermal expansion within the different material regions. The derivations in this section require the fuel centerline and the various surface temperatures from §4.5.2. To calculate the average temperature in a region, an temperature function $T(r)$ is required. For the bond and clad regions, this is given in Eq. (4.47) and Eq. (4.49) respectively.

No such temperature function exists for the fuel region due to the variable thermal conductivity. In the fuel region, an effective constant thermal conductivity is calculated. First, the fuel centerline temperature is calculated assuming variable thermal conductivity. Then, the fuel surface and fuel centerline temperatures are used to calculate the effective constant thermal conductivity. Finally, $T_F(r)$ can be expressed by holding the fuel thermal conductivity constant at the effective value and using Eq. (4.31). In a test case at reactor conditions, this method of effective constant thermal conductivity resulted in a 10 [K] error in the average fuel temperature compared to a numerical integration with 1,000 points. For the purposes of calculating average values to be used in interpolation, this approximation is acceptable.

4.5.3.1 Average Fuel Temperature

In the fuel region, an effective constant thermal conductivity $\overline{k_F}$ is first calculated to satisfy Eq. (4.31) such that

$$\overline{k_F} = \frac{q'''_{i,k} R_F^2}{4(T_0 - T_F)} \quad (4.64)$$

where $T_0 = T_F(0)$ is the fuel centerline temperature and $T_F = T_F(R_F)$ is the fuel surface temperature. $T_F(0)$ and $T_F(R_F)$ have been calculated with variable thermal conductivity. Then, the temperature in the region, $T_F(r)$ for $r \in [0, R_F]$, can be expressed as Eq. (4.30). The average temperature in the fuel region is then

$$\overline{T_F} = \frac{\int_0^{R_F} T_F(r) r \, dr}{\int_0^{R_F} r \, dr} \quad (4.65)$$

$$\overline{T_F} = \frac{\int_0^{R_F} \left(T_0 - \frac{q'''_{i,k}}{4\overline{k_F}} r^2 \right) r \, dr}{\int_0^{R_F} r \, dr} \quad (4.66)$$

noting the additional r multiplication term due to cylindrical coordinates and a factor 2π has been canceled. Begin with evaluating the integral in the denominator of Eq. (4.66).

$$\int_0^{R_F} r \, dr = \frac{r^2}{2} \Big|_{r=0}^{r=R_F} = \frac{R_F^2}{2} \quad (4.67)$$

Next, the integral in the numerator of Eq. (4.66) is evaluated.

$$\int_0^{R_F} \left(T_0 - \frac{q'''_{i,k}}{4k_F} r^2 \right) r \, dr = T_0 \frac{R_F^2}{2} - \frac{q'''_{i,k}}{16k_F} R_F^4 \quad (4.68)$$

Evaluating Eq. (4.66) by dividing Eq. (4.68) by Eq. (4.67) yields an expression for the average fuel temperature in the fuel region.

$$\overline{T}_F = T_0 - \frac{q'''_{i,k}}{8k_F} R_F^2 \quad (4.69)$$

Eq. (4.69) then yields the average temperature in the fuel region assuming the thermal conductivity is constant at the effective value k_F . \overline{T}_F will be used to calculate fuel cross sections.

To calculate more accurate fuel cross sections, an effective fuel temperature could be used which weights the fuel surface temperature higher than the average value. Due to self-shielding in the fuel, the temperature corresponding to the average neutron flux within the fuel rod is nearer to the surface of the fuel as neutron flux decreases significantly within the fuel material itself. For the purposes of this model, the use of volume averaged fuel temperatures is acceptably accurate as rod powers cannot be resolved in this simplified geometry due to the smearing of cross sections within a hexagonal fuel assembly. In addition, metal fuel has a lower change in temperature between surface and centerline temperatures when compared to oxide fuels and rod dimensions are smaller than typical LWRs.

4.5.3.2 Average Bond Temperature

The value of the bond temperature at position $r \in [R_F, R_B]$ is given in Eq. (4.47). The average temperature within the bond, \overline{T}_B , is

$$\overline{T}_B = \frac{\int_{R_F}^{R_B} T_B(r) r \, dr}{\int_{R_F}^{R_B} r \, dr} = \frac{\int_{R_F}^{R_B} \left(T_F - \frac{q'''_{i,k}}{2k_B} R_F^2 \ln \left(\frac{r}{R_F} \right) \right) r \, dr}{\int_{R_F}^{R_B} r \, dr}, \quad (4.70)$$

with the weighting due to radial coordinates included. In Eq. (4.70), $T_F = T_F(R_F)$ is the fuel surface temperature and k_B is the constant thermal conductivity in the clad as specified in Table 4.1. Evaluating the denominator of Eq. (4.70).

$$\int_{R_F}^{R_B} r \, dr = \frac{r^2}{2} \Big|_{r=R_F}^{r=R_B} = \frac{R_B^2 - R_F^2}{2} \quad (4.71)$$

The integral in the numerator of Eq. (4.70) is more than a simple polynomial. The integral $\int \ln(r/\alpha) r dr$ is given in a table of integrals. Then, the numerator of Eq. (4.70) can be evaluated.

$$\int_{R_F}^{R_B} \left(T_F - \frac{q_{i,k}'''}{2k_B} R_F^2 \ln \left(\frac{r}{R_F} \right) \right) r dr = \int_{R_F}^{R_B} \left(T_F r - \frac{q_{i,k}'''}{2k_B} R_F^2 \ln \left(\frac{r}{R_F} \right) r \right) dr \quad (4.72)$$

$$= T_F \frac{r^2}{2} \Big|_{r=R_F}^{r=R_B} - \frac{q_{i,k}'''}{2k_B} R_F^2 \left(\frac{r^2}{2} \ln \left(\frac{r}{R_F} \right) - \frac{r^2}{4} \right) \Big|_{r=R_F}^{r=R_B} \quad (4.73)$$

$$= T_F \frac{R_B^2 - R_F^2}{2} - \frac{q_{i,k}'''}{2k_B} R_F^2 \left(\frac{R_F^2 - R_B^2}{4} + \frac{R_B^2}{2} \ln \left(\frac{R_B}{R_F} \right) \right) \quad (4.74)$$

With these expressions, Eq. (4.70) can be evaluated by dividing Eq. (4.74) by Eq. (4.71) and simplifying.

$$\overline{T_B} = T_F - \frac{q_{i,k}'''}{4k_B} R_F^2 \left(\frac{R_F^2 - R_B^2 + 2 R_B^2 \ln \left(\frac{R_B}{R_F} \right)}{R_B^2 - R_F^2} \right) \quad (4.75)$$

Eq. (4.75) then yields the average temperature in the fuel region and $\overline{T_B}$.

4.5.3.3 Average Clad Temperature

The calculation of the average clad temperature is similar to the calculation of the average bond temperature because both of these regions have no heat generation and are assumed to have constant thermal conductivity. A derivation procedure similar to §4.5.3.2 and the result will be similar. Therefore, the average value of the conductivity integral in the cladding is

$$\overline{T_C} = T_B - \frac{q_{i,k}'''}{4k_C} R_F^2 \left(\frac{2 R_C^2 \ln \left(\frac{R_C}{R_B} \right)}{R_C^2 - R_B^2} - 1 \right) \quad (4.76)$$

where T_B is the bond surface temperature. With the use of Eq. (4.76), the average clad temperature $\overline{T_C}$ can be calculated and used for calculating cross sections.

4.6 Cross Section Treatment

The purpose of calculating average material temperatures is to calculate cross sections for use in the multigroup neutron diffusion equation. Temperature-dependent cross sections are calculated by interpolating between cross section libraries generated according to §1.3 in order to simulate the temperature feedback effects as they relate to neutron cross sections.

The main form of temperature feedback affecting microscopic neutron cross sections is the Doppler broadening of neutron absorption resonance peaks in heavy nuclei. Linear interpolation is valid for materials minimally effected by Doppler broadening. Such materials include cladding and sodium as these materials do not change temperature significantly over the operating range of fast reactors and have relatively few neutron absorption resonances. The fuel material is interpolated according to a square-root interpolation because the Doppler effect obeys square-root behavior. Square-root interpolation is necessary due to relatively high fuel temperatures and the many neutron absorption resonances in fuel material, specifically uranium isotopes such as ^{238}U .

Reactor material temperatures are updated periodically throughout the neutron diffusion calculation. The user may specify a fixed number of power iterations for which the diffusion calculation will be computed before updating the temperatures. Currently, ten power iterations are executed per temperature update and this provides efficient and convergent behavior. For a discussion of power iterations, see §2.4.

When the temperatures are updated, the cross sections are also updated. Updating the material cross sections requires recomputing the finite element matrix (see Eq. (2.34)) which proves to be the most computationally expensive part of the update. Note that the update of cross sections during a power iteration makes the iterative method non-linear and the convergence of the method is no longer guaranteed. However, this procedure is commonly used and, in practice, no convergence problems have been observed. The general procedure to update cross sections is given in Algorithm 4.1.

In Algorithm 4.1, the cross sections are updated using the average material temperatures. Average material temperatures are unique within each chunk. Therefore, instead of having cross sections constant throughout the reactor and only a function of material, cross sections are now unique within each chunk.

Algorithm 4.1 Temperature and Cross Section Update Procedure.

- 1: Complete N_{iter} power iterations to calculate unnormalized $\bar{\phi}_{e,g}$. Currently $N_{iter} = 10$.
 - 2: Normalize flux according to Eq. (4.5).
 - 3: Calculate power for each element as in Eq. (4.6) and Eq. (4.7).
 - 4: **for all** Chunks **do**
 - 5: Calculate power $q_{i,k}$ and $q'_{i,k}$ in Eq. (4.9) and Eq. (4.10).
 - 6: Calculate bulk coolant temperature using the procedure outlined in §4.4.
 - 7: Calculate surface temperatures and fuel centerline temperature using the procedure in §4.5.2.
 - 8: Calculate material average temperatures using the procedure outlined in §4.5.3.
 - 9: Update cross sections using average material temperatures.
 - 10: Reconstruct the finite element matrix \mathbf{A}_g .
-

4.6.1 Coolant Cross Sections

The main temperature effect on coolant cross sections is due to the change in fluid density. The mass density of liquid sodium is given as a function of temperature $\rho(T)$ by a state relationship [Fin95]. Natural sodium is entirely composed of ^{23}Na and has atomic mass $M_{Na} = 22.989769 \left[\frac{\text{gram}}{\text{mol}} \right]$ [Bau10]. Then, the number density of the coolant is given as

$$N_{Na}(T) = \frac{\rho_{Na}(T) N_A}{M_{Na}} \quad (4.77)$$

where N_A is Avogadro's number. When temperatures are calculated and cross sections are updated, the number density of sodium is updated and cross sections are calculated. $N_{Na}(T)$ is calculated using $T_{\infty,i,k}$ which is the bulk coolant temperature in each chunk. Therefore, each chunk will have a unique coolant number density.

Using the coolant temperatures for the cross section library structure described in §1.3, microscopic coolant cross sections are calculated using a linear interpolation. For coolant temperature $T_n < T_{\infty,i,k} < T_{n+1}$

$$\Sigma_{x,i,k,g} = N_{Na}(T_{\infty,i,k}) \left(\frac{T_{\infty,i,k} - T_n}{T_{n+1} - T_n} (\sigma_{x,Na,g,n+1} - \sigma_{x,Na,g,n}) + \sigma_{x,Na,g,n} \right) \quad (4.78)$$

where N_{Na} is calculated according to Eq. (4.77), $\sigma_{x,Na,g,n}$ and $\sigma_{x,Na,g,n+1}$ are the microscopic cross sections for reaction x in energy group g that correspond to library temperatures T_n and T_{n+1} respectively.

4.6.2 Clad Cross Sections

The temperature dependence of cross sections within cladding material is typically small as cladding material has fewer absorption resonances than fuel material. Compared to the fuel, cladding material does not change temperature as extremely as other reactor materials. For this reason, a simple linear interpolation is used to interpolate macroscopic cross sections within this material. For $T_n < \overline{T_{C,i,k}} < T_{n+1}$, the macroscopic cross section for reaction x and energy group g according to the linear interpolation method is given as

$$\Sigma_{x,i,k,g} = \frac{\overline{T_{C,i,k}} - T_n}{T_{n+1} - T_n} (\Sigma_{x,g,n+1} - \Sigma_{x,g,n}) + \Sigma_{x,g,n} \quad (4.79)$$

where $\Sigma_{x,g,n}$ and $\Sigma_{x,g,n+1}$ are the macroscopic cross sections that correspond to library temperatures T_n and T_{n+1} respectively.

4.6.3 Bond Cross Sections

While the average bond temperature is computed, the temperature and density dependence of bond cross sections is neglected. This is because the bond represents a small volume of the reactor and the sodium

cross sections do not change significantly with temperature. Therefore, it is assumed that, for purposes of calculating cross sections, the temperature in the coolant is the same as the temperature in the bond and cross sections in the bond are calculated as Eq. (4.78). This assumption is consistent with the smearing of hexagonal fuel assemblies.

4.6.4 Fuel Cross Sections

Temperature dependence of cross sections in the fuel region is dominated by the Doppler effect. This effect is large for high fuel temperatures; therefore, a square-root interpolation method is used. The square-root interpolation is commonly selected for fuel cross section interpolation because the Doppler effect is proportional to the square-root of absolute temperature. For $T_n < \overline{T_{F,i,k}} < T_{n+1}$, the macroscopic cross section for reaction x in energy group g within the fuel is calculated as

$$\Sigma_{x,i,k,g} = \frac{\sqrt{\overline{T_{F,i,k}}} - \sqrt{T_n}}{\sqrt{T_{n+1}} - \sqrt{T_n}} (\Sigma_{x,g,n+1} - \Sigma_{x,g,n}) + \Sigma_{x,g,n} \quad (4.80)$$

where $\Sigma_{x,g,n}$ and $\Sigma_{x,g,n+1}$ are the macroscopic cross sections of reaction x corresponding to library temperatures T_n and T_{n+1} respectively. Note that all temperatures used for this calculation are in units of absolute temperature.

4.7 Thermal Hydraulic Results

To demonstrate the expected behavior of these models, axial and radial temperature plots are generated for a typical case and compared to analytic results.

4.7.1 Total Reactor Power

To verify the calculation of enthalpy according to the axial heat convection model is implemented correctly, reactor power is calculated using enthalpy and compared to the user input Q_{Rx} . An energy balance using the reactor as a control volume dictates that

$$Q_{Rx} = \sum_{i=1}^{N_{chan}} \dot{m}_i (h_{i,N_z+1/2} - h_{inlet}) \quad (4.81)$$

where $h_{i,N_z+1/2}$ is the enthalpy at the upper elevation of the chunk located in channel i at the ultimate axial elevation N_z . The user inputs reactor power, Q_{Rx} , and the calculated value is compared to the input value. For a reactor model operating at full power, the calculated reactor power agreed to the user input reactor power to within machine precision.

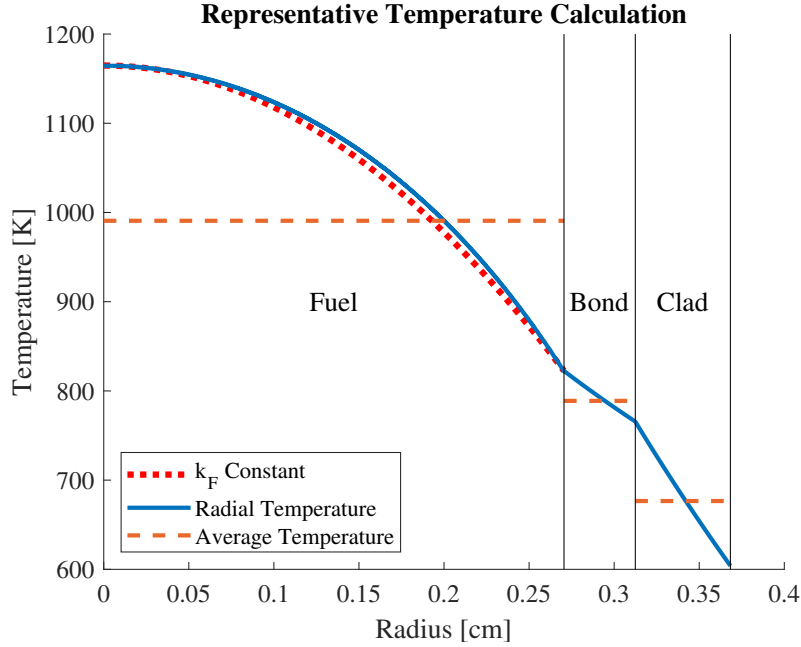


Figure 4.5 Radial Temperatures for Typical Fuel Rod.

4.7.2 Radial Results

Radial temperatures for a typical fuel rod model are shown in Fig. 4.5. The radial solution is only generated to demonstrate the expected behavior as the radial temperature functions themselves are not used in the reactor simulation. This model was generated for $q'''_{i,k} = 6.93 \times 10^9 \left[\frac{\text{W}}{\text{m}^3} \right]$ and $\dot{m}_i = 68.56 \left[\frac{\text{kg}}{\text{s}} \right]$, which are typical values generated during reactor simulation. It is seen that the temperature profile within the fuel region is approximately quadratic and temperature profiles in the bond and clad are approximately linear. These are the expected behavior for a constant thermal conductivity in each region. The fuel temperature is calculated assuming variable thermal conductivity $k_F(T_F(r))$ and the effective thermal conductivity $\overline{k_F}$ as in Eq. (4.64). The error due to assuming constant thermal conductivity in the fuel is small and less than 15 [K].

In Fig. 4.5, it is observed that the average temperatures as calculated above reasonably approximate the average temperature within each material. Average temperatures shown in Fig. 4.5 are from formulae in §4.5.3. A numerical integration was performed using the 1,000 data points plotted in Fig. 4.5 to estimate the average temperature in each region and these numerical results agreed with the modeled results to within 10 [K]. In reactor simulations, the average values from §4.5.3 are used.

Table 4.2 System Properties for Axial Model Verification.

Property	Value	
q_0'''	6.93×10^9	$\left[\frac{\text{W}}{\text{m}^3} \right]$
\dot{m}	68.56	$\left[\frac{\text{kg}}{\text{s}} \right]$
T_{inlet}	400.	$[\text{K}]$
h_{in}	246.62	$\left[\frac{\text{kJ}}{\text{kg}} \right]$
H	1.2	$[\text{m}]$
R_F	0.27051	$[\text{cm}]$
N_{rod}	271	

4.7.3 Axial Results

In this section, the results of the axial model are compared to reference results to verify the model is working as expected. An exact solution for axial temperatures exists for a given heat generation rate and for constant k_F . This exact model is compared to a numerical model as developed in §4.4. For this solution, assume

$$q'''(z) = q_0''' \sin\left(\frac{\pi z}{H}\right) \quad (4.82)$$

where q_0''' is dictated by the power of the reactor Q_{Rx} . Assume the properties in Table 4.2 where H is the channel height.

First, convert $q'''(z)$ to $q'(z)$ as

$$q'(z) = \pi R_F^2 N_{rod} q'''(z) \quad (4.83)$$

and as a result,

$$q'_0 = \pi R_F^2 N_{rod} q_0'''. \quad (4.84)$$

Begin with the calculation of $h(x)$ using Eq. (4.11). Then,

$$h(z) = h_{in} + \frac{1}{\dot{m}} \int_0^z q'_0 \sin\left(\frac{\pi z'}{H}\right) dz' \quad (4.85)$$

$$= h_{in} + \frac{q'_0 H}{\dot{m} \pi} \left(1 - \cos\left(\frac{\pi z}{H}\right)\right) \quad (4.86)$$

and $T_\infty(z) = T(h(z))$ given by a sodium state relationship.

Note that the discretization error is only due to the approximation of the integral in Eq. (4.11) with the summation in Eq. (4.12). Given the linear heat generation rate in Eq. (4.83), the total heat generated

in node i located $z \in [z_{i-1/2}, z_{i+1/2}]$ is

$$q_i = q'_0 \frac{H}{\pi} \left(\cos \left(\frac{\pi z_{i-1/2}}{H} \right) - \cos \left(\frac{\pi z_{i+1/2}}{H} \right) \right) \quad (4.87)$$

and then $h_{i+1/2}$ is computed using Eq. (4.12).

The result of the axial discretization is an error in the bulk coolant temperature axially, $T_\infty(z)$. All subsequent calculations of surface temperatures and average temperatures are analytical and exact to the approximations made (e.g. constant thermal conductivity). Therefore, the error is only computed as it relates to $T_\infty(z)$. Analytic and modeled results are shown in Fig. 4.6 as well as the difference between the two results. The numerical model used 36 axial levels to approximate the axial temperatures. The error for the discretization selected is less than 10 [K].

Average axial temperatures for the discrete model are shown in Fig. 4.7. Results are obtained for 36 axial levels and the radial heat conduction model from §4.5.3 is used at each axial level to calculate modeled temperatures.

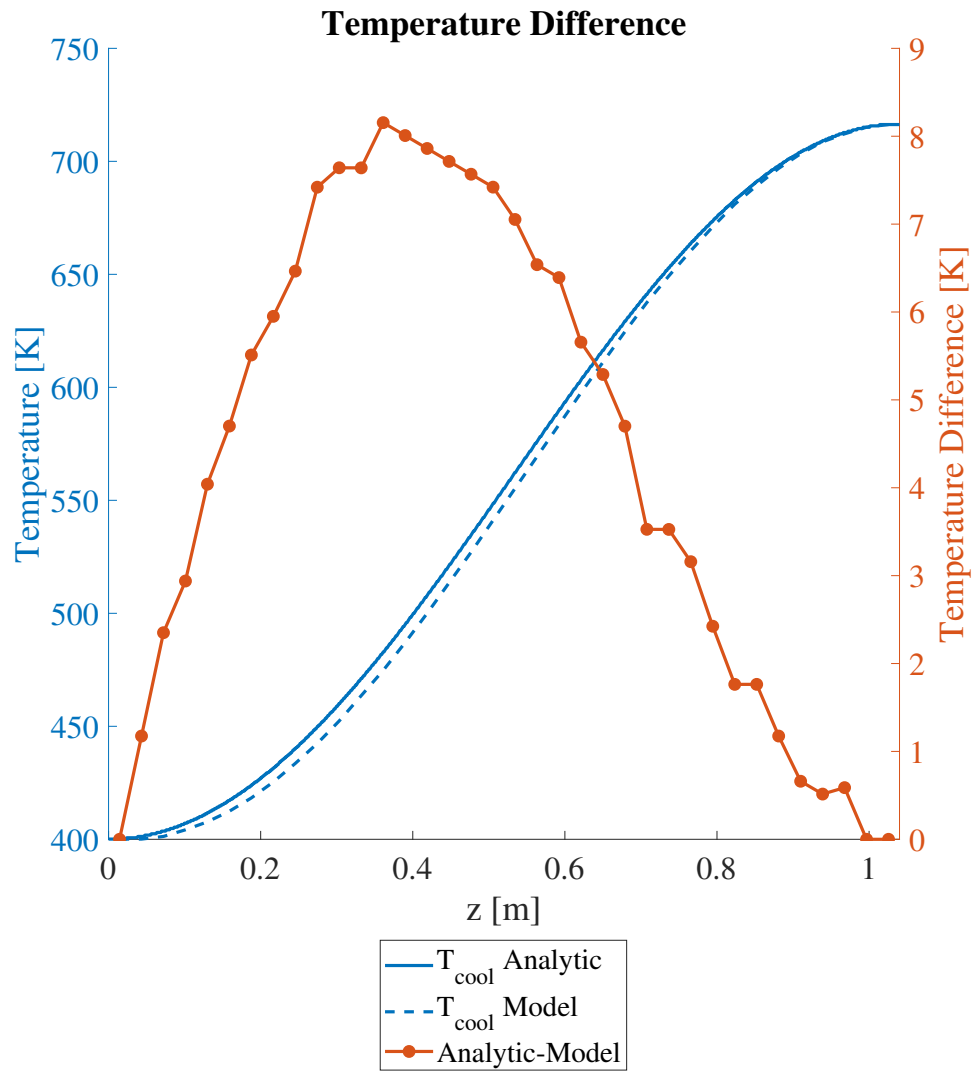


Figure 4.6 Difference Between Analytic and Modeled Axial Temperatures for 36 Axial Levels.

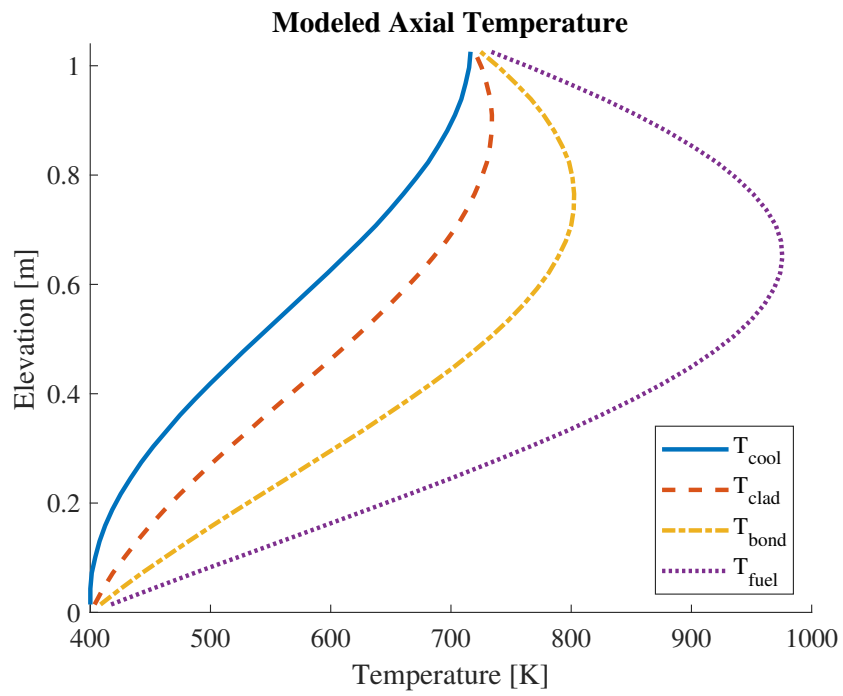


Figure 4.7 Average Axial Temperatures for Model Reactor Conditions.

CHAPTER

5

THERMAL EXPANSION

5.1 Necessity of Modeling

The fast reactor is entirely composed of metals and, as such, experiences significant thermal expansion. While other designs may employ non-metallic fuel material (e.g. oxides or carbides), these are not considered in this work. In the future, nonmetallic fuels could be considered by including new material properties in the model. Reactor designs with metal fuel include Experimental Breeder Reactor II (EBR-II), as designed and built by Argonne National Laboratory (ANL), and PRISM, as designed by GE-Hitachi Nuclear Energy Americas LLC (GEH).

In metal fueled reactors, thermal expansion represents a significant reactivity feedback effect. The preliminary safety information document for PRISM presents an estimate for thermal expansion feedback such that a 1% increase in radial dimension results in $-0.5 [\Delta k]$, indicating a significant effect [Gen87]. Additionally, thermal expansion has been demonstrated to serve as an inherent safety feature of fast reactors. In the remarkable EBR-II demonstrations in April 1986, two major accident demonstrations were performed on the reactor while operating at full power. Operators forced the reactor to undergo Unprotected Loss-Of-Flow (ULOF) and Unprotected Loss-Of-Heat-Sink (ULOHS) events. EBR-II was safely shutdown due to nothing other than inherent multiphysics feedback effects. These experiments demonstrated conclusively the passive safety of fast reactor designs, due in part to the thermal expansion of materials [Til11].

EBR-II system response to ULOF was demonstrated by bypassing the normal loss of flow scram (i.e. shutdown) function. The control rod drive motors were deenergized and the main coolant pumps were tripped off and allowed to coast down. Natural circulation flow followed and, after an initial temperature peak, material temperatures returned to normal operating temperatures [Pla87].

The ULOHS response was demonstrated by shutting down coolant pumps in the secondary coolant system, thereby disabling heat rejection from the primary coolant system. No subsequent automatic or manual action was taken. In the ULOHS demonstration, material temperatures did not peak and instead quickly reduce below operating temperatures [Pla87].

The inherent safety demonstration test at EBR-II demonstrated the passive safety of the liquid metal cooled and metal fueled reactor design. These tests are simply not possible with currently operating LWRs designs. If operating LWRs were to undergo these tests, the reactor cores would not survive. Using nothing but natural phenomena, the reactor was demonstrated to shutdown and subsequent heat removal was achieved [Pla87]. These tests are pivotal the strategy to demonstrate the performance of similar reactors in Anticipated Transient Without Scram (ATWS) events. The tests also demonstrate the importance of modeling multiphysics effects in fast reactor applications.

5.2 Material Properties

In this thesis, all structural materials in the reactor are thermally expanded as HT9 stainless steel. Fuel material is thermally expanded as metallic uranium with 10% Zr by weight included (i.e. U10Zr). The equations for the Linear Expansion Factor (LEF) used in this work are given as functions of temperature

$$\left(\frac{\Delta L}{L}\right)_{\text{HT9}} = -2.191 \times 10^{-3} + 5.678 \times 10^{-6} T + 8.111 \times 10^{-9} T^2 - 2.576 \times 10^{-12} T^3, \quad (5.1)$$

$$\left(\frac{\Delta L}{L}\right)_{\text{U10Zr}} = \begin{cases} -7.3 \times 10^{-3} + 3.489 \times 10^{-5} T - 5.154 \times 10^{-8} T^2 + 4.39 \times 10^{-11} T^3 & T \leq 923 \text{ [K]}, \\ -0.25252 + 6.669 \times 10^{-4} T - 5.441 \times 10^{-7} T^2 + 1.518 \times 10^{-10} T^3 & \text{otherwise,} \end{cases} \quad (5.2)$$

for T in [K] [Lei88; Bas09]. Note that U10Zr undergoes a phase change at 923 [K] that increases the LEF at this point. Both Eq. (5.1) and Eq. (5.2) evaluate to zero near room temperature. The LEF of HT9 and U10Zr over the range of operating temperatures of fast reactors are plotted in Fig. 5.1. It is observed that the LEF of U10Zr is as much as twice that of HT9. This implies fuel material will expand significantly more than structural material.

All sodium in the reactor is assumed to be liquid. Therefore, effects of thermal expansion within sodium are described by the change in density as a function of temperature [Fin95], not by a LEF.

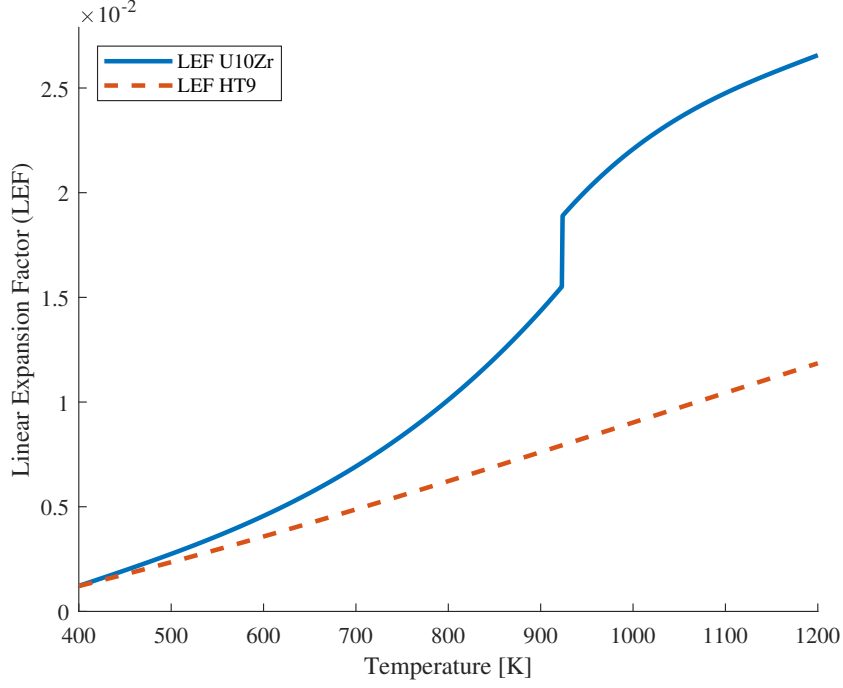


Figure 5.1 Linear Expansion Factor for HT9 Steel and U10Zr Fuel.

Consistent with this assumption, the mass of sodium within the reactor is not conserved with thermal expansion. Sodium coolant will flow into and out of the reactor vessel as structural components thermally expand. Sodium in the bond region will flow upward within the cladding into a gas plenum at the top of the fuel rod as the fuel thermally expands. This is not modeled as the sodium level in the bond is not tracked.

5.3 Model Details

Thermal expansion contributes two main categories of feedback effects: leakage effects and density effects. Increased reactor dimensions due to thermal expansion increases neutron leakage from the reactor. The neutron leakage fraction is the fraction of neutrons created in the fuel due to fission that exit the core. The leakage fraction can be expressed as

$$\mathcal{L} = \frac{k_{eff}}{k_{\infty}} \quad (5.3)$$

where k_{∞} is the effective neutron multiplication factor for an infinite medium. LWRs typically have low and ultra-low leakage designs with $\mathcal{L} \approx 2\%$ [Kni14]. However, fast reactors simulated in this work have $\mathcal{L} \approx 20\%$ and are therefore highly sensitive to changing reactor dimensions due to thermal expansion.

Density effects of thermal expansion are a consequence of the conservation of material within the reactor. As reactor dimensions expand, the reactor volume increases. However, the quantity of material in the reactor vessel remains unchanged. Therefore, material densities must be decreased proportionally to the increase of reactor dimensions. Decreasing material densities results in decreased macroscopic neutron cross sections due to the proportionality $\Sigma = N \sigma$ where Σ is a macroscopic cross section, N is a number density, and σ is a microscopic cross section.

Highly detailed thermal expansion modeling can be performed using the FEM to calculate local stresses and strains on all reactor structural components such as fuel pins, wire wraps, and assembly cans. However, this would require a significantly more advanced model and improved spatial resolution. The model developed here for the simulation of fast reactors does not estimate temperature and heat generation at all positions due to the smearing of hexagonal assemblies. Therefore, a simplified thermal expansion model is developed. This simplified model expands finite element coordinates uniformly throughout the reactor, expands area fractions within finite elements, and decreases number densities accordingly.

All dimensions in the reactor are expanded assuming the user input dimensions are at room-temperature conditions. Dimensions are expanded according to two user input temperatures, $T_{exp,fuel}$ and $T_{exp,struct}$. $T_{exp,fuel}$ corresponds to the average temperature of fuel material in the reactor and $T_{exp,struct}$ corresponds to the average temperature of structural material in the reactor. Typically, these values come from a previous coupled neutron diffusion and thermal hydraulics simulation. $T_{exp,fuel}$ and $T_{exp,struct}$ must be known before a thermal expansion simulation begins. This is acceptable because the LEFs from Eq. (5.1) and Eq. (5.2) are on the order 10^{-2} so small changes in user input temperatures will be insignificant.

5.3.1 Expansion of Finite Element Coordinates

In the simplified thermal expansion model developed for this work, the coordinates of finite elements are expanded uniformly in the axial and radial directions. In the radial (both x and y) directions, it is assumed that elements expand as structural material. Specifically, x and y coordinates are expanded using HT9 material properties from Eq. (5.1). The dominant effect on k_{eff} due to thermal expansion in the radial directions is the expansion of the hexagonal assemblies themselves and the expansion of the grid plate at the base of the reactor, all of which are composed of HT9 stainless steel. In the axial direction (the z direction), it is assumed that elements expand as fuel material, U10Zr, using the LEF from Eq. (5.2). The dominant effect on k_{eff} due to thermal expansion in the axial direction is the elongation of the metallic fuel.

Assuming the uniform radial expansion of elements specified by Eq. (5.1) and $T_{exp,struct}$, a radial LEF can be defined as

$$F_r(T_{exp,struct}) = 1 + \left(\frac{\Delta L}{L} \right)_{HT9} \quad (5.4)$$

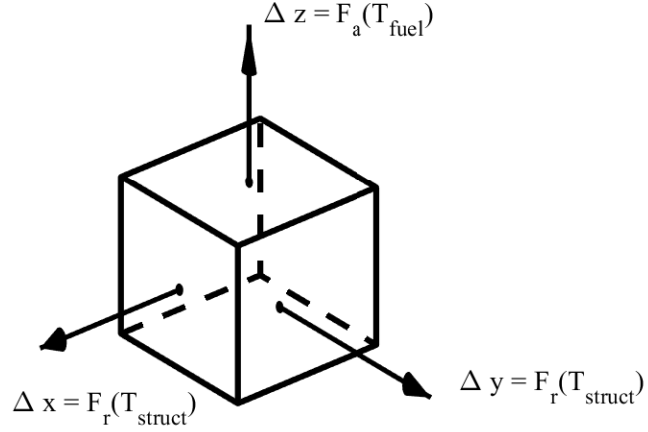


Figure 5.2 Thermal Expansion of General Volume.

which is a function of $T_{exp,struct}$. Similarly, an axial LEF can be defined using Eq. (5.2) and $T_{exp,fuel}$.

$$F_a(T_{exp,fuel}) = 1 + \left(\frac{\Delta L}{L} \right)_{U10Zr} \quad (5.5)$$

This assumption of uniform expansion of elements in the radial and axial directions is depicted in Fig. 5.2. Fig. 5.2 shows a general volume being thermally expanded in radial directions according to Eq. (5.4) and the axial direction according to Eq. (5.5).

Since all elements are being expanded at the same rate, they will not intersect or overlap. Due to the uniform expansion of elements, the “cold” coordinate (x^C, y^C, z^C) can be thermally expanded to the “hot” coordinate (x^H, y^H, z^H) using Eq. (5.4) and Eq. (5.5).

$$x^H = x^C F_r(T_{exp,struct}), \quad (5.6)$$

$$y^H = y^C F_r(T_{exp,struct}), \quad (5.7)$$

$$z^H = z^C F_a(T_{exp,fuel}), \quad (5.8)$$

Eq. (5.6), Eq. (5.7) and Eq. (5.8) uniformly expand the distance from each coordinate to the origin. Thus, the coordinate thermal expansion equations are applied to each coordinate in the finite element mesh to uniformly expand all elements.

5.3.2 Expansion of Area Fractions

The thermal expansion of the dimensions of hexagonal assemblies themselves has been modeled with the assumption of uniform expansion of finite elements in §5.3.1. Dimensions within a hexagonal assembly, such as fuel radius, are expanded by modifying area fractions. These thermally expanded area fractions are then used to homogenize cross sections within an assembly as in §1.3.

The radius of the fuel material, R_F , is expanded using the LEF for U10Zr from Eq. (5.2). All other dimensions within the hexagonal assembly (wire wrap diameter, can thickness, flat-to-flat measurement, etc.) are expanded using the LEF for HT9 from Eq. (5.1). Note that due to the magnitudes of these LEFs, it is possible for R_F to exceed the inner radius of the cladding. This is a non-physical result and will only occur for small sodium bond gaps and high thermal expansion temperatures. In these cases, the radius of the fuel is confined to the thermally expanded inner radius of the cladding. Though a more complex relationship describes the true value of these radii, the resolution of this model does not allow for such detail.

Thermally expanded area fractions are calculated by first thermally expanding each dimension within a hexagonal assembly. Then, cross-sectional areas and resulting area fractions are calculated using the formulae from §1.2. By using these formulae, the sodium bond and sodium coolant area fractions will be allowed to “float.” That is, the sodium area fraction will decrease appropriately to allow for the expansion of other materials within the assembly. Recall, the mass of sodium in the reactor is not conserved. Finally, the calculated areas are used to calculate area fractions and homogenize cross sections within a hexagonal assembly.

5.3.3 Conservation of Material and Cross Section Effects

After thermally expanding finite elements and area fractions within assemblies, material densities are decreased to conserve quantity of material in the reactor. In this derivation, the conservation of reactor material is expressed as a conservation of number of atoms. The conservation of the number of atoms for species i , can be expressed as

$$n_i^H = n_i^C \quad (5.9)$$

where n_i^H is the number of atoms of species i after thermal expansion, i.e. “hot,” and n_i^C is the number of atoms of species i before thermal expansion, i.e. “cold.” The number of atoms n_i can be written as

$$n_i = N_i V_i \quad (5.10)$$

where N_i is the number density of species i and V_i is the volume occupied by species i . Then, inserting Eq. (5.10) into Eq. (5.9) yields an expression for the thermally expanded number density of species i as

$$N_i^H V_i^H = N_i^C V_i^C, \quad (5.11)$$

$$N_i^H = N_i^C \frac{V_i^C}{V_i^H}, \quad (5.12)$$

where the term $\frac{V_i^C}{V_i^H} < 1$ and represents the expansion of the volume occupied by species i .

The volume V_i can be written in terms of element volume and area fraction. Let species i be contained in region j in finite element e . This work assumes area fractions are constant within an element and can be treated as volume fractions. Then, the volume V_i can be rewritten as $V_i = a_j V_e$ where a_j is the area fraction of region j and V_e is the volume of element e . Inserting this definition for V_i into Eq. (5.12).

$$N_i^H = N_i^C \frac{a_j^C V_e^C}{a_j^H V_e^H} \quad (5.13)$$

The ratio of area fractions, $\frac{a_j^C}{a_j^H}$, is calculate directly using cold and hot area fractions as described in §5.3.2. The ratio of element volumes $\frac{V_e^C}{V_e^H}$ can be rewritten using the relationships Eq. (5.4) and Eq. (5.5).

Begin by considering a volume such as Fig. 5.2. Define the volume $V^C = L_x^C L_y^C L_z^C$ with L_x^C , L_y^C , and L_z^C representing the “cold” lengths of the volume. The thermally expanded volume, V^H , can then be written

$$V^H = (L_x^C + \Delta L_x)(L_y^C + \Delta L_y)(L_z^C + \Delta L_z). \quad (5.14)$$

Recognizing the expansion of coordinates from Eq. (5.6), Eq. (5.7), and Eq. (5.8).

$$V^H = (L_x^C F_r(T_{exp,struct}))(L_y^C F_r(T_{exp,struct}))(L_z^C F_a(T_{exp,fuel})) \quad (5.15)$$

$$V^H = L_x^C L_y^C L_z^C (F_r(T_{exp,struct}))^2 F_a(T_{exp,fuel}) \quad (5.16)$$

$$V^H = V^C (F_r(T_{exp,struct}))^2 F_a(T_{exp,fuel}) \quad (5.17)$$

The element volume expansion ratio is then

$$\frac{V^C}{V^H} = \frac{1}{(F_r(T_{exp,struct}))^2 F_a(T_{exp,fuel})}. \quad (5.18)$$

Eq. (5.18) can now be substituted into Eq. (5.13).

$$N_i^H = N_i^C \frac{a_j^C}{a_j^H} \frac{1}{(F_r(T_{exp,struct}))^2 (F_a(T_{exp,fuel}))} \quad (5.19)$$

To preserve the number of atoms in the reactor, material number densities must be updated according to Eq. (5.19) in addition to expanding element coordinates and area fractions. Notice that Eq. (5.19) can also be used to directly update neutron cross sections directly as they are proportional to number density.

5.4 Results

The effect of thermal expansion on the effective neutron multiplication factor, k_{eff} , is shown in this section. As stated previously in §5.3, the user must input effective temperatures to which the reactor is thermally expanded. These user input values, $T_{exp,fuel}$ and $T_{exp,struct}$, are varied and all other thermal feedback effects are disabled in the simulation. For this simplified demonstration, $T_{exp,fuel} = T_{exp,struct}$. It is expected that thermal expansion will cause a significant decrease in k_{eff} and represent negative reactivity insertion. Effective neutron multiplication factor as a function of thermal expansion is plotted in Fig. 5.3 and the associated reactivity, calculated as

$$\Delta\rho \text{ [pcm]} = \frac{k_{eff} - k_{ref}}{k_{eff} k_{ref}} \times 10^5 \quad (5.20)$$

is plotted in Fig. 5.4. In Eq. (5.20), k_{eff} is the calculated effective neutron multiplication factor after thermal expansion and k_{ref} is the neutron multiplication factor without thermal expansion.

Given the assumptions in this model, Fig. 5.3 and Fig. 5.4 show that thermal expansion represents a significant reactivity effect and contributes as much as $-1,000$ [pcm] at extreme temperatures. Additionally, in this model the thermal expansion factor of the fuel given in Eq. (5.2) dominates and the phase change given by the formula can be seen at the expected 923 [K].

In the next chapter, the results of thermal expansion, along with thermal hydraulic feedback will be shown for a real reactor application.

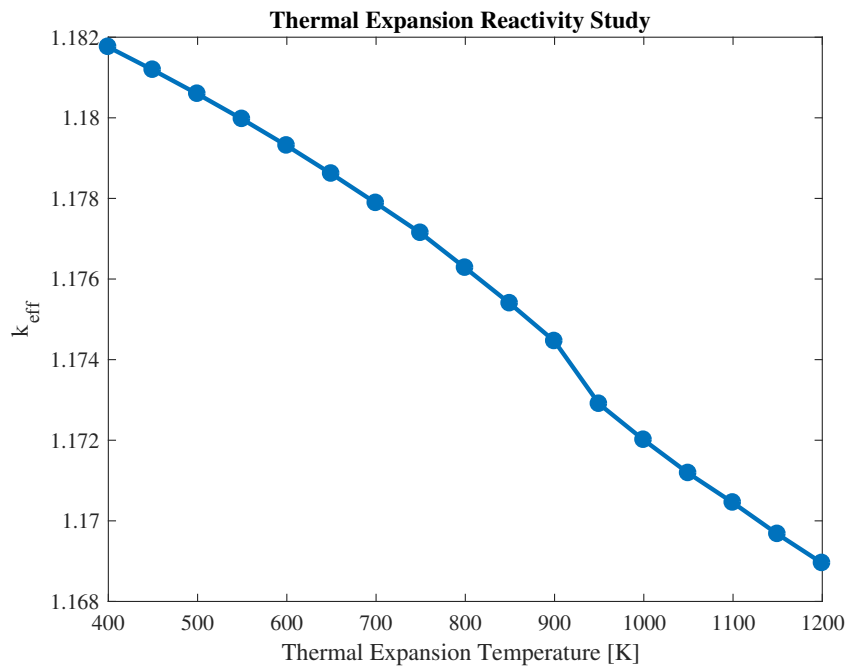


Figure 5.3 Effective Neutron Multiplication Factor as a Function of Thermal Expansion Temperature.

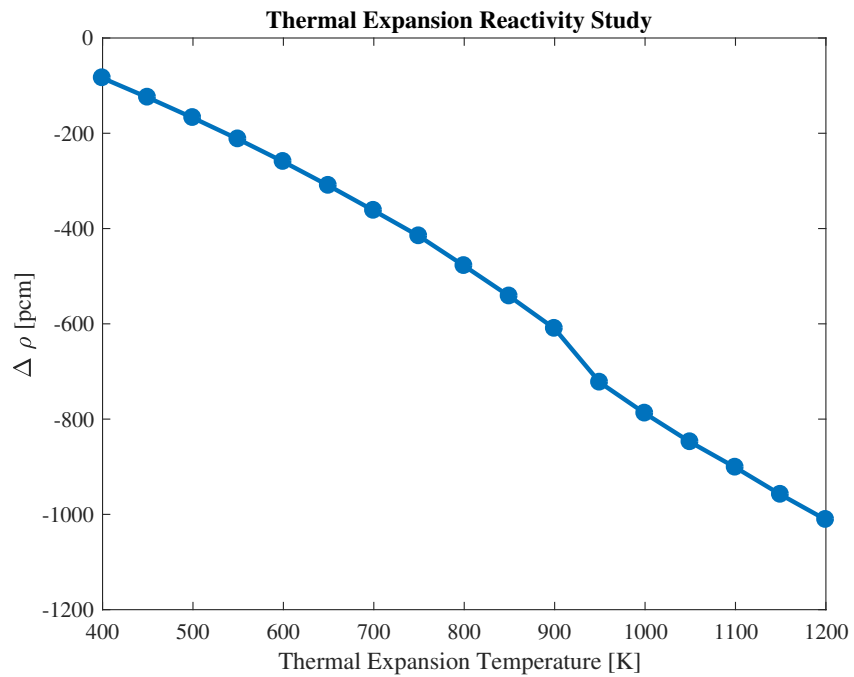


Figure 5.4 Reactivity as a Function of Thermal Expansion Temperature.

CHAPTER

6

COUPLED MULTIPHYSICS RESULTS

6.1 Power Reactor Modeling

The motivation for this work is to model nuclear power reactors with multiphysics feedback. This has been accomplished by modeling the reactor power distribution with the multigroup neutron diffusion equation solved via the FEM (Chapter 2). Axial heat convection and radial heat conduction models are used to estimate reactor material temperatures (Chapter 4). Simplified thermal expansion modeling is used to model reactor dimensions (Chapter 5). Combined, these multiphysics effects will provide feedback which can be estimated in a realistic core model.

To test the coupling of these models, a realistic reactor benchmark is provided and modeled in §6.2. Reactivity coefficients describing system feedback are defined in §6.3. Results are presented in §6.4.

6.2 Advanced Burner Reactor – MET-1000

The Advanced Burner Reactor (ABR) is a benchmark reactor design is proposed by Organisation for Economic Co-operation and Development (OECD) Nuclear Energy Agency (NEA) [OEC16]. The ABR is fueled with a ternary alloy metallic fuel and has a 1,000 [MWth] rating; hence, MET-1000. This is a medium-sized, metallic-fueled reactor with a total of 180 hexagonal assemblies and is 4.8 [m] tall. The benchmark is fully specified and 31 independent results have been submitted so far. Compared to other

Table 6.1 Advanced Burner Reactor Refinement Results.

Refinement	DIF3D k_{eff}	FEM k_{eff}	Difference [pcm]
0	1.017906	0.999339	1856.7
1	1.013614	1.006694	692.0

benchmark problems modeled in this thesis, this model is extremely large due to the large structural components included above and below the active fuel region.

Each submission to the benchmark uses independently generated cross sections from several different cross section libraries (e.g. ENDF/B-VII.0, JEFF3.1, etc.). Therefore, using this benchmark as a verification problem is not feasible. Cross sections were generated for this model using MC**2 and the procedure outlined in §1.3. This procedure resulted in temperature-dependent cross section libraries with 33 energy groups.

To verify the FEM model, the multigroup neutron diffusion equation is solved with the same cross sections using DIF3D and the FEM solution from Chapter 2. In this verification, cross sections are fixed at the value associated with nominal reactor temperatures and multiphysics feedback is disabled, as DIF3D has no multiphysics capabilities. DIF3D and the FEM, using the same cross sections, agree to 692 [pcm]. The DIF3D and FEM models were minimally refined. Initially, an unrefined geometry was modeled with six triangles per hexagon and 80 axial elevations, implying a wedge height of 6 [cm]. This unrefined model had a total of 30,320 elements. The model was then spatially refined with 24 triangles per hexagon and 160 axial elevations, implying a wedge height of 3 [cm]. The once refined model had a total of 242,560 elements. Results from this brief verification study are presented in Table 6.1. A more formal mesh refinement study would presumably show further error reduction.

Reactor materials in the benchmark are shown in Fig. 6.1. For the sole purpose of intuitive flux visualization, the 33-group energy structure is collapsed to a two-group energy structure. The collapsed fast (ϕ_1) and collapsed thermal (ϕ_2) are plotted in Fig. 6.2. Note: all subsequent results presented are generated with the full 33-group energy structure. Fast flux is shown to peak in the center of the core in the active fuel region. Thermal flux is shown to peak in core structural material, at the periphery of the active fuel region, as well as in control rod locations where the control rods have been withdrawn from the active fuel region.

Multiphysics results for this benchmark have not been published. Therefore, the coupled results are not compared to reference results. Instead, each of the multiphysics models has been investigated individually in the preceding chapters.

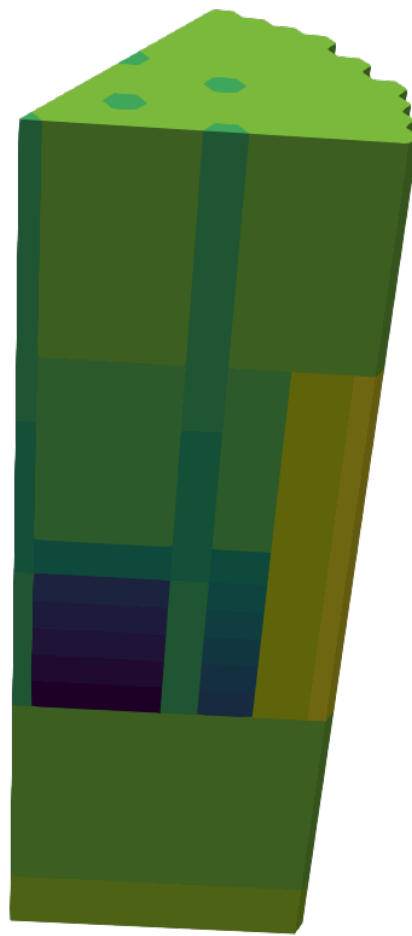


Figure 6.1 Materials in ABR.

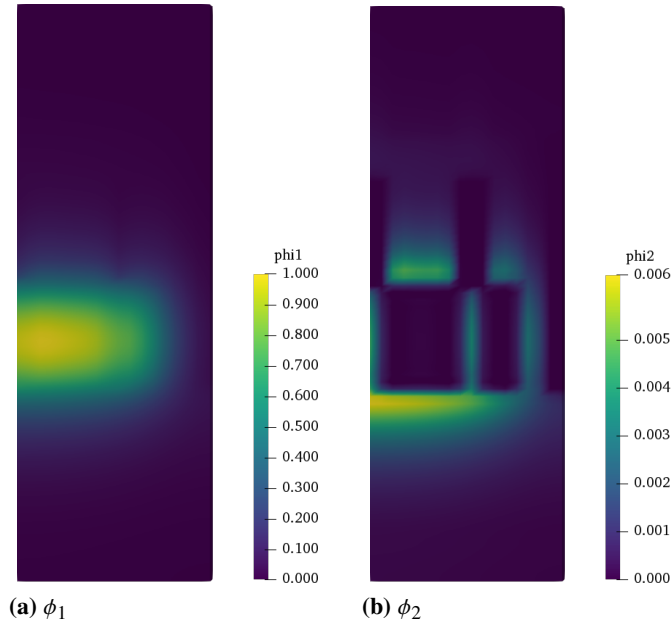


Figure 6.2 Fast and Thermal Neutron Flux in ABR.

6.3 Reactivity Coefficients

Reactivity of a reactor can be used to compare the state of the reactor to the critical state. Reactivity ρ is defined as

$$\rho \text{ [pcm]} = \frac{k_{eff} - 1}{k_{eff}} \times 10^5. \quad (6.1)$$

Recall $k_{eff} < 1$ for a subcritical reactor, $k_{eff} = 1$ for a critical reactor, and $k_{eff} > 1$ for a supercritical reactor. Therefore $\rho < 0$ for a subcritical reactor, $\rho = 0$ for a critical reactor, and $\rho > 0$ for a supercritical reactor.

A reactivity coefficient can be defined as a partial derivative with respect to a quantity of interest [Kni14]. Let α_x be the reactivity coefficient for quantity x , then

$$\alpha_x(x_i) = \left. \frac{\partial \rho}{\partial x} \right|_{x=x_i} \quad (6.2)$$

where ρ is the reactivity defined by Eq. (6.1). Eq. (6.2) is useful for estimating reactor dynamics. For some change in reactor state Δx , the reactivity response can be estimated as

$$\Delta \rho \approx \alpha_x(x_i) \Delta x. \quad (6.3)$$

It is expected that reactivity coefficients will vary as reactor conditions vary. Therefore, it will be necessary to calculate α_x as a function of reactor condition x_i . Specifically, reactor power, Q_{Rx} , will be varied in the

calculation of α . Therefore, a set of reactor powers varying from 0% to 100% as $Q_{Rx,i} = \{0\%, \dots, 100\%\}$ will be used.

Reactivity coefficients useful for fast reactor applications include the power coefficient, thermal expansion coefficient, fuel temperature (Doppler) coefficient, and Coolant Temperature Coefficient (CTC) [Kni14]. Reactivity coefficients will be estimated with a first-order, forward-Euler, finite-difference approximation such that

$$\alpha_x(x_i) \approx \frac{\rho(x_i) - \rho(x_i + \Delta x)}{\Delta x} \quad (6.4)$$

for a given Δx . The evaluation of Eq. (6.4) is discussed in the following sections for relevant reactivity coefficients.

6.3.1 Power Reactivity Coefficient

The power reactivity coefficient measures the reactivity response due to a power increase. In a stable reactor, $\alpha_{power} < 0$ to ensure an increase in reactor power requires a reactivity increase and to prevent a runaway power increase. To evaluate α_{power} , the reactor is simulated at a nominal reactor power, $Q_{Rx,i}$ resulting in $k_{eff}(Q_{Rx,i})$. Then, reactor power is increased by ΔQ_{Rx} resulting in $k_{eff}(Q_{Rx,i} + \Delta Q_{Rx})$. These k_{eff} values correspond to reactivities $\rho(Q_{Rx,i})$ and $\rho(Q_{Rx,i} + \Delta Q_{Rx})$ respectively as defined by Eq. (6.1). With these values, the power reactivity coefficient can be calculate as

$$\alpha_{power}(Q_{Rx,i}) = \frac{\rho(Q_{Rx,i}) - \rho(Q_{Rx,i} + \Delta Q_{Rx})}{\Delta Q_{Rx}}. \quad (6.5)$$

A typical value of ΔQ_{Rx} is 5% Q_{Rx} .

6.3.2 Thermal Expansion Reactivity Coefficient

The thermal expansion reactivity coefficient describes the reactivity response due solely to thermal expansion for a given increase in reactor power. It is expected that thermal expansion will be the dominant contribution to the power reactivity coefficient in fast reactors. This is due to two main reasons: the significant thermal expansion of metal fuels at high temperature and the large neutron leakage fraction ($\mathcal{L} \approx 20\%$) in fast reactors (see Chapter 5).

Thermal expansion temperatures, $T_{exp,fuel}$ and $T_{exp,struct}$, as implemented in Chapter 5 must be known before the simulation. Note from §6.3.1, a case must be simulated to calculate $k_{eff}(Q_{Rx,i} + \Delta Q_{Rx})$ in the calculation of the power reactivity coefficient in Eq. (6.5). Therefore, the temperature distribution resulting from this study can be used to calculate the thermal expansion temperatures for the case with increased power, $T_{exp}(Q_{Rx,i} + \Delta Q_{Rx})$. Then, the thermal expansion reactivity coefficient is

$$\alpha_{thexp}(Q_{Rx,i}) = \frac{\rho(T_{exp}(Q_{Rx,i})) - \rho(T_{exp}(Q_{Rx,i} + \Delta Q_{Rx}))}{\Delta Q_{Rx}} \quad (6.6)$$

where $T_{exp}(Q_{Rx,i})$ represents the thermal expansion temperatures for power $Q_{Rx,i}$ and $T_{exp}(Q_{Rx,i} + \Delta Q_{Rx})$ represents the thermal expansion temperatures for power $Q_{Rx,i} + \Delta Q_{Rx}$. Note that in Eq. (6.6), only thermal expansion temperatures are changed, not the true reactor power Q_{Rx} . A typical value of ΔQ_{Rx} is $5\%Q_{Rx}$.

6.3.3 Fuel Temperature Reactivity Coefficient

The fuel temperature reactivity coefficient measures the reactivity change due to an increase in fuel temperature. This coefficient is often termed the Doppler coefficient because the reactivity effect is due to the Doppler broadening of resonance absorption peaks in heavy nuclei such as ^{238}U [Kni14]. Briefly, at high fuel temperatures, neutrons are more likely to be parasitically absorbed by non-fissile nuclei than fissile nuclei in the fuel material.

To calculate $\alpha_{Doppler}$, fuel temperature is increased directly. A simulation is conducted with feedback for reactor power $Q_{Rx,i}$ and the temperature profile is stored. Then, the fuel temperature is uniformly increased in the reactor by ΔT_{fuel} and the simulation is conducted again. This procedure will result in $k_{eff}(Q_{Rx,i})$ and $k_{eff}(T_{fuel} + \Delta T_{fuel})$. Then, the Doppler reactivity coefficient follows.

$$\alpha_{Doppler}(Q_{Rx,i}) = \frac{\rho(Q_{Rx,i}) - \rho_i(T_{fuel} + \Delta T_{fuel})}{\Delta T_{fuel}} \quad (6.7)$$

Note that the Doppler reactivity coefficient is always negative. A typical value of ΔT_{fuel} is 5 [K]. The definition in Eq. (6.7) is a *uniform* Doppler coefficient as opposed to a *distributed* Doppler coefficient because temperatures are increased uniformly throughout the reactor.

6.3.4 Coolant Temperature Coefficient

The CTC describes the reactivity change due to an increase in coolant temperature. In LWRs, this may be called the Moderator Temperature Coefficient (MTC) but in fast reactors, the coolant is not designed to moderate neutrons. Feedback in the coolant is due to two main phenomena: the decrease in absorption cross sections in the coolant due to Doppler broadening and the decrease of density due to temperature increase. The dominant effect is the decrease of sodium density due to the temperature increase [Kni14].

Unlike all other reactivity coefficients presented here, the CTC of the ABR is positive as is common in fast reactors. This implies an increase in coolant temperature will lead to an increase in reactivity and cause a subsequent increase in reactor power. In fast reactors, the coolant acts as a parasitic neutron absorber. Therefore, a decrease in the sodium absorption cross section or a decrease in sodium density due to a temperature increase encourages neutron absorption in fissile material in the fuel and results in a reactivity increase. This does not pose a stability problem as long as the power reactivity coefficient remains negative.

To calculate α_{CTC} , coolant temperature is increased directly with a procedure similar to that for the Doppler reactivity coefficient in §6.3.3. A simulation is conducted with feedback for reactor power $Q_{Rx,i}$ and the temperature profile is stored. Then, the coolant temperature is uniformly increased by ΔT_{cool} and the simulation is conducted again. This procedure will result in $k_{eff}(Q_{Rx,i})$ and $k_{eff}(T_{cool} + \Delta T_{cool})$. Then, the CTC follows.

$$\alpha_{CTC}(Q_{Rx,i}) = \frac{\rho(Q_{Rx,i}) - \rho(T_{cool} + \Delta T_{cool})}{\Delta T_{cool}} \quad (6.8)$$

A typical value for ΔT_{cool} is 5 [K].

6.4 Results

Returning to the ABR MET-1000 benchmark, reactivity coefficients are modeled for this reactor. The methodology and formulae from §6.3 are used. The reactor k_{eff} as a function of reactor power is plotted in Fig. 6.3. Note k_{eff} decreases as power increases implying a negative power coefficient.

The reactivity coefficients themselves are plotted in Fig. 6.4. The combined power reactivity coefficient is plotted in Fig. 6.4a. Note, $\alpha_{power} < 0$ for all powers. The CTC is positive as shown in Fig. 6.4d and as expected. The Doppler coefficient is negative as shown in Fig. 6.4c and as expected. The CTC becomes more positive at high powers but the magnitude does not change significantly. The Doppler coefficient becomes less negative at high powers. However, reactor temperatures also increase at high power so the net effect is a net reactivity decrease due to temperature increase at high reactor power. Additionally, the magnitudes of the CTC and Doppler coefficients are comparable and have opposite sign so the effects largely cancel.

At high reactor powers, α_{power} becomes more negative due to the dominance of the thermal expansion reactivity coefficient plotted in Fig. 6.4b. Should reactor power continue to increase beyond the 100% nominal value, the power reactivity coefficient would continue to become more negative as thermal expansion will continue to dominate.

Multiphysics effects are summarized in Table 6.2. To separate the contributions of thermal hydraulics and thermal expansion, the reactor was simulated using only the thermal expansion model and using both thermal expansion and thermal hydraulics models. The data in Table 6.2 indicates a total power defect of -589.49 [pcm]. This power defect is slightly lower than other SFRs, but this reactor was designed for steady-state neutronics calculations, not multiphysics simulations. The majority of the power defect, -559.64 [pcm], is attributed to thermal expansion effects. The remaining -29.85 [pcm] is attributed to thermal hydraulics and cross section feedback effects. This measurement of thermal hydraulic reactivity feedback is a bit deceiving as there is a cancellation of errors occurring. The Doppler reactivity coefficient and CTC, as plotted in Fig. 6.4c and Fig. 6.4d respectively, are shown to have similar magnitude and opposite signs. For the ABR design, these coefficients are similar in magnitude but other reactors may

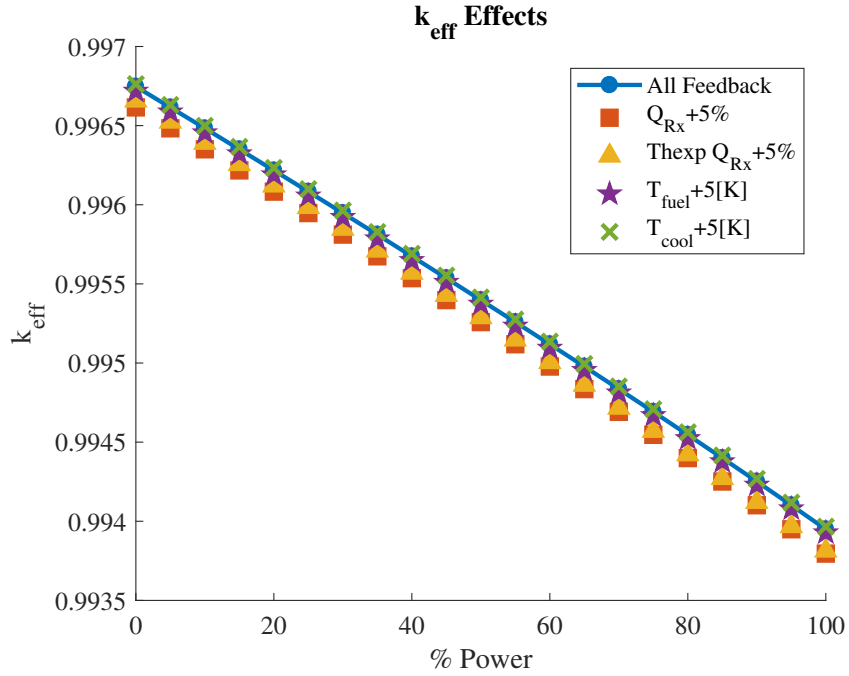
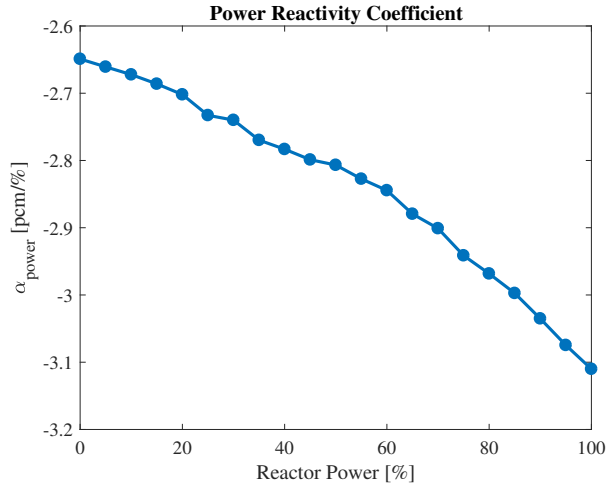


Figure 6.3 Feedback Effects on k_{eff} .

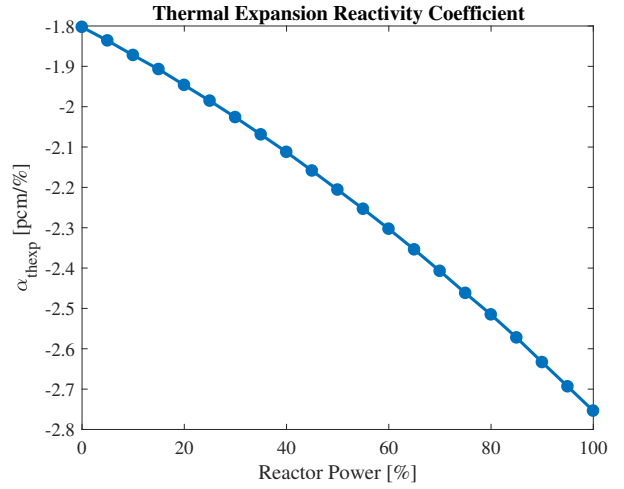
Table 6.2 Multiphysics Contributions to Total Power Defect.

Case	Thermal Expansion Power	Thermal Hydraulic Power	k_{eff}	Reactivity [pcm]
1	0%	0%	0.999808	
2	100%	0%	0.994246	-559.64
3	100%	100%	0.993950	-589.49

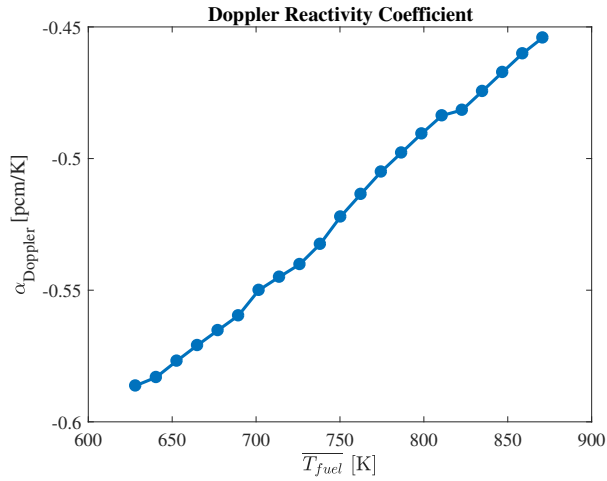
not have such a cancellation. Therefore, the thermal hydraulics effects remain important for accurate simulation of a general fast reactor.



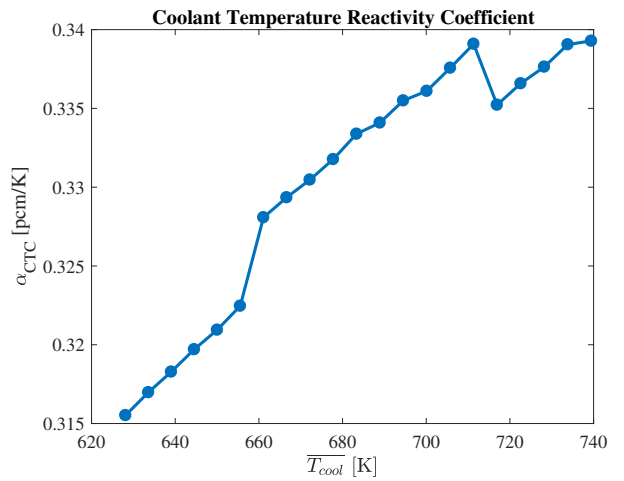
(a) Power Reactivity Coefficient.



(b) Thermal Expansion Reactivity Coefficient.



(c) Doppler Reactivity Coefficient.



(d) Coolant Temperature Reactivity Coefficient.

Figure 6.4 ABR Reactivity Coefficients.

CHAPTER

7

SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

7.1 Summary of Simulation Results

The purpose of this thesis is to simulate a fast nuclear power reactor at operating conditions with coupled multiphysics feedback. The method developed allows for solution to the multigroup neutron diffusion equation for general unstructured mesh and an arbitrary energy group structure. The coupled multiphysics models allows for inherent modeling thermal feedback effects instead of using extremely simplified models or calculations by hand. By including all multiphysics models in a single simulation suite, a user can easily observe the interaction of physical phenomena and feedback.

In Chapter 2, a rigorous and general framework is developed for solving the multigroup neutron diffusion equation via the FEM for unstructured mesh. Insights are provided into the use of both two-dimensional triangular elements and three-dimensional wedge (pentahedral) elements. Both of these geometries are natural choices for fast reactors which typically employ hexagonal geometries. Using the methods developed, Chapter 3 then demonstrates code verification and solution solution verification and for both analytic and reactor benchmark problems. The multigroup neutron diffusion solver as implemented is shown to converge to the correct k_{eff} value and flux distribution at the correct convergence rate.

Chapter 4 presents the details of the thermal hydraulic models employed. These models include an axial heat convection model and a radial heat conduction model. Temperatures resulting from these thermal hydraulic models are used to interpolate cross section tables and update coolant density to generate temperature-dependent cross sections. The formulae for the interpolation of temperature-dependent cross sections are also provided.

In Chapter 5, a simplified thermal expansion model is presented. The model expands materials linearly assuming materials thermally expand as either HT9 Stainless-Steel structural material or U10Zr fuel material. The model requires an *a priori* assumption of material temperatures but results are not highly sensitive to these temperatures due to the magnitudes of thermal expansion coefficients.

The culmination of these multiphysics models is presented in a coupled simulation in Chapter 6. A typical fast reactor, presented in a benchmark problem, is simulated [OEC16]. Using the models developed, reactivity coefficients can be estimated for an operating reactor. These coefficients provide insights into dynamic reactor behavior and agree with expected values. These reactivity coefficients also describe the mechanisms for inherent safety in a fast reactor.

7.2 Conclusions

It has been demonstrated that the FEM can be used to efficiently simulate the power distribution in a nuclear power reactor. Use of the FEM has allowed for the local simulation of multiphysics effects within elements. The FEM framework allows for intuitive coupling of multiphysics effects because physics simulations can all share the same mesh.

By simulating thermal hydraulic feedback and thermal expansion effects, reactivity effects are estimated. Prior to this work, the simulation of feedback effects required either manual iteration between thermal hydraulic calculations and a neutron diffusion solver or the use of other simplified estimates. This work has demonstrated a practical implementation of fast reactor multiphysics simulations by utilizing the FEM and a finite element framework.

7.3 Recommendations for Future Research

In addition to general efficiency improvements (e.g. increased use of sparse matrices and vectorization) future work includes code enhancements, added features, and simulation of new reactors. Ultimately, the goal is to develop a reactor simulation suite that can be used by industry professionals to perform core design optimization calculations and analyze dynamic reactor behavior.

7.3.1 Depletion Capabilities

To serve as a useful reactor simulator, this suite must be expanded to include depletion capabilities. The code as presented in this work can be used to design and optimize a clean (i.e. cycle zero) fast reactor. However, any practical reactor design must be capable of extended and continuous operation during an operating cycle. Depletion calculations inform the reactor design by establishing the required excess reactivity at the beginning of cycle. Control strategies must then be designed to control this excess reactivity.

Depletion calculations require solving the Bateman equations and the associated matrix exponential system of equations. Preliminary investigations have been conducted for the implementation of the Chebyshev Rational Approximation Method (CRAM) as the depletion solver in this suite [Pus13]. The CRAM will be implemented using the FEM mesh. A subsequent mesh study will be required to determine if it is possible to use a coarse mesh for depletion or if the same fine mesh used for the multigroup neutron diffusion solution must also be used in depletion calculations.

7.3.2 Higher Order Finite Elements

Recall the FEM as presented in Chapter 2 employs only linear triangular and linear wedge elements. However, the FEM as derived can be implemented for polynomials of arbitrary order. Work by Hosseini & Vosoughi [Hos13] suggests that for finite element solutions to the multigroup neutron diffusion equation, quadratic elements provide a significant reduction of error without the need to reduce the mesh spacing. With linear finite elements, the FEM is second-order spatially convergent. Quadratic finite elements convey third-order spatial convergence and cubic finite elements convey fourth-order spatial convergence. Incorporating higher order finite elements will allow for simple solution refinement without mesh regeneration.

7.3.3 Simplified P_N Solution

The Simplified P_N (SP_N) equations can provide considerably improved accuracy compared to the diffusion equation by incorporating angular dependence of flux rather than assuming isotropic scalar flux. The SP_N equations reduce to a system of diffusion-like equations. Work by Ryu & Joo [Ryu13] has demonstrated that the FEM can be used to solve the SP_N equations in general geometry. Implementation of the SP_N equations would require the development of a new solver but much of the FEM framework can be reused. Implementation of the SP_N method could provide enhanced accuracy for full-scale reactor problems.

7.3.4 Encouraging Code Usage

The simulation suite is designed to be used for practical fast nuclear power reactor simulations. An important step in this code becoming practical is its adoption by users. Encouraging the usage of the

simulation package will allow for user feedback. Additionally, users employing the code in a variety of simulations will encourage the development of new features that may be necessary for unique reactor designs. Future possible reactors to be simulated include the SuperPhenix benchmark and simulation of the EBR-II inherent safety demonstrations [Pla87].

Currently, the code is maintained in a private Github repository so it may easily be shared in the future. The author may be contacted*for access to the repository.

*William Christopher Dawn: wcdawn@ncsu.edu

BIBLIOGRAPHY

- [Ame77] American Nuclear Society & Argonne National Laboratory. *Argonne Code Center: Benchmark Problem Book, ANL-7416(Suppl.2)*. Tech. rep. 1977.
- [Aya15] Ayachit, U. *The ParaView Guide: A Parallel Visualization Application*. Kitware, 2015.
- [Bas09] Basak, C. et al. “An evaluation of the Properties of As-Cast U-Rich U–Zr Alloys.” *Journal of Alloys and Compounds* **480.2** (2009), pp. 857–862.
- [Bau10] Baum, E. et al. *Nuclides and Isotopes*. New York: Knolls Atomic Power Laboratory, 2010.
- [Cha95] Chao, Y. A. & Shatilla, Y. A. “Conformal Mapping and Hexagonal Nodal Methods: Implementation in the ANC-H Code.” *Nuclear Science and Engineering* **121.2** (1995), pp. 210–225.
- [Cut69] Cuthill, E. & McKee, J. “Reducing the Bandwidth of Sparse Symmetric Matrices.” *Proceedings of the 1969 24th National Conference*. New York, NY, USA: Association for Computing Machinery, 1969, pp. 157–172.
- [Duf02] Duff, I. S. et al. “An Overview of the Sparse Basic Linear Algebra Subprograms.” *Association for Computing Machinery Transactions on Mathematical Software* **28.2** (2002), pp. 239–267.
- [Dun85] Dunavant, D. A. “High Degree Efficient Symmetrical Gaussian Quadrature Rules for the Triangle.” *International Journal for Numerical Methods in Engineering* **21** (1985), pp. 1129–1148.
- [Fel04] Felippa, C. A. *Advanced Finite Element Methods*. Boulder: University of Colorado at Boulder, 2004.
- [Fin95] Fink, J. K. & Leibowitz, L. *Thermodynamic and Transport Properties of Sodium Liquid and Vapor*. Tech. rep. ANL/RE-95/2. Argonne National Laboratory, 1995.
- [Geh92] Gehin, J. C. “A Quasi-Static Polynomial Nodal Method for Nuclear Reactor Analysis.” PhD thesis. Massachusetts Institute of Technology, 1992.
- [Gen87] General Electric Advance Nuclear Technology. *PRISM Preliminary Safety Information Document*. Tech. rep. GEFR-00793. 1987.
- [Hen75] Henry, A. F. *Nuclear-Reactor Analysis*. Cambridge: MIT Press, 1975.
- [Hos17] Hosseini, S. A. “Sensitivity Analysis of the Galerkin Finite Element Method Neutron Diffusion Solver to the Shape of the Elements.” *Nuclear Engineering and Technology* **49.1** (2017), pp. 29–42.

- [Hos15] Hosseini, S. A. & Saadatian-Derakhshandeh, F. “Galerkin and Generalized Least Squares Finite Element: A Comparative Study for Multi-group Diffusion Solvers.” *Progress in Nuclear Energy* **85** (2015), pp. 473–490.
- [Hos13] Hosseini, S. A. & Vosoughi, N. “Development of Two-Dimensional, Multigroup Neutron Diffusion Computer Code Based on GFEM with Unstructured Triangle Elements.” *Annals of Nuclear Energy* **51** (2013), pp. 213–226.
- [Hug87] Hughes, T. J. R. *The Finite Element Method*. Englewood Cliffs, NJ: Prentice-Hall, 1987.
- [Ips09] Ipsen, I. C. F. *Numerical Matrix Analysis: Linear Systems and Least Squares*. Philadelphia: Society for Industrial and Applied Mathematics, 2009.
- [Joh09] Johnson, C. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Mineola, NY: Dover, 2009.
- [Kel95] Kelley, C. T. *Iterative Methods for Linear and Nonlinear Equations*. Society for Industrial and Applied Mathematics, 1995, p. 166.
- [Kim14] Kim, Y. S. et al. “Thermal Conductivities of Actinides (U, Pu, Np, Cm, Am) and Uranium-Alloys (U-Zr, U-Pu-Zr and U-Pu-TRU-Zr).” *Journal of Nuclear Materials* **445**.1-3 (2014), pp. 272–280.
- [Kni14] Knief, R. A. *Nuclear Engineering: Theory and Technology of Commerical Nuclear Power*. Lagrange Park, IL: American Nuclear Society, Inc., 2014.
- [Kom78] Komano, Y. et al. *Improved Few-Group Coarse-Mesh Method for Calculating Three-Dimensional Power Distribution in Fast Breeder Reactor*. Tech. rep. NEACRP-L-204. Nuclear Energy Agency, 1978.
- [Lam66] Lamarsh, J. R. *Introduction to Nuclear Reactor Theory*. Reading, MA: Addison-Wesley, 1966. Chap. 9.
- [Law05] Lawrence Livermore National Laboratory. *VisIt User’s Manual*. UCRL-SM-220449. 2005.
- [Lee12] Lee, C. H. & Yang, W. S. *MC2-3 : Multigroup Cross Section Generation Code for Fast Reactor Analysis*. Tech. rep. ANL/NE-11-41. Argonne National Laboratory, 2012.
- [Lei88] Leibowitz, L. & Blomquist, R. A. “Thermal Conductivity and Thermal Expansion of Stainless Steels D9 and HT9.” *International Journal of Thermophysics* **9**.5 (1988), pp. 873–883.
- [Lew08] Lewis, E. E. *Fundamentals of Nuclear Reactor Physics*. Boston: Elsevier Inc., 2008.
- [Li18] Li, Z. et al. *Numerical Solution of Differential Equations*. Cambridge (England): Cambridge University Press, 2018.
- [Nak77] Nakamura, S. *Computational Methods in Engineering Science*. New York: Wiley, 1977.

- [Obe07] Oberkampf, W. L. & Trucano, T. G. *Verification and Validation*. Tech. rep. Albuquerque: Sandia National Laboratories, 2007.
- [OEC16] OECD Nuclear Energy Agency. *Benchmark for Neutronic Analysis of Sodium-cooled Fast Reactor Cores with Various Fuel Types and Core Sized*. Tech. rep. NEA/NSC/R(2015)9. 2016.
- [Pfr07] Pfrang, W. & Struwe, D. *Assessment of Correlations for Heat Transfer to the Coolant for Heavy Liquid Metal Cooled Core Designs*. Tech. rep. FZKA 7352. Karlsruhe: Forschungszentrum Karlsruhe GmbH, 2007.
- [Pla87] Planchon, H. et al. “Implications of the EBR-II Inherent Safety Demonstration Test.” *Nuclear Engineering and Design* **101.1** (1987), pp. 75–90.
- [Pus13] Pusa, M. “Numerical Methods for Nuclear Fuel Burnup Calculations.” PhD thesis. Espoo, Finland: Aalto University, 2013.
- [Ryu13] Ryu, E. H. & Joo, H. G. “Finite Element Method Solution of the Simplified P3 Equations for General Geometry Applications.” *Annals of Nuclear Energy* **56** (2013), pp. 194–207.
- [Sch06] Schroeder, W. et al. *The Visualization Toolkit*. 4th ed. Kitware, 2006.
- [Tak91] Takeda, T. & Ikeda, H. “3-D Neutron Transport Benchmarks.” *Journal of Nuclear Science and Technology* **28.7** (1991), pp. 656–669.
- [Til11] Till, C. & Chang, Y. *Plentiful Energy*. CreateSpace Independent Publishing Platform, 2011.
- [Wal11] Waltar, A. et al. *Fast Spectrum Reactors*. New York: Springer, 2011.
- [Whi85] White, R. E. *An Introduction to the Finite Element Method with Applications to Nonlinear Problems*. New York: Wiley, 1985.

APPENDICES

APPENDIX

A

ANALYTIC SOLUTIONS TO THE NEUTRON DIFFUSION EQUATION

A.1 Introduction

This appendix contains analytic solutions to the neutron diffusion equations that are used for code verification of one-dimensional, two-dimensional, and three-dimensional numerical solvers. One-group and two-group problems are addressed. Analytic solutions are important because they provide an exact solution that allows for analysis of convergence properties for a numerical method. A case matrix of the analytic solutions provided in this appendix is provided in Table A.1. A variety of cases are provided including fixed source and criticality problems. The number of spatial dimensions, number of energy groups, and number of materials are varied.

Table A.1 Case Matrix for Analytic Solutions.

Case	Dimensions	Groups	Criticality	Materials
1	1	1		1
2	1	1	✓	1
3	2	1	✓	1
4	1	2	✓	1
5	1	1	✓	2
6	3	1	✓	1

For the one-group reference problems, the neutron diffusion problem is written

$$-D\nabla^2\phi + \Sigma_r\phi = \frac{1}{k_{eff}}\nu\Sigma_f\phi + q_{fixed} \quad (\text{A.1})$$

where

- D = diffusion coefficient [cm],
- ϕ = scalar neutron flux $\left[\frac{1}{\text{cm}^2 \text{ s}}\right]$,
- Σ_r = macroscopic removal cross section $\left[\frac{1}{\text{cm}}\right]$,
- k_{eff} = effective neutron multiplication factor,
- $\nu\Sigma_f$ = number of fission neutrons per unit neutron flux $\left[\frac{1}{\text{cm}}\right]$,
- q_{fixed} = fixed volumetric neutron source $\left[\frac{1}{\text{cm}^3 \text{ s}}\right]$.

Eq. (A.1) is valid for problems with constant coefficients (homogeneous materials). For problems considered here, material properties such as D are constant in some region, either the entire problem or a finite subdomain.

For two-group neutron diffusion problems, the two-group neutron diffusion equation is written

$$-D_1\nabla^2\phi_1 + \Sigma_{r1}\phi_1 = \frac{1}{k_{eff}}(\nu\Sigma_{f1}\phi_1 + \nu\Sigma_{f2}\phi_2), \quad (\text{A.2})$$

$$-D_2\nabla^2\phi_2 + \Sigma_{r2}\phi_2 = \Sigma_{s1\rightarrow2}\phi_1, \quad (\text{A.3})$$

where notation is the same as Eq. (A.1) with the addition of subscripts indicating the energy groups and ϕ_1 is the higher energy group and ϕ_2 is the lower energy group such that $E_1 > E_2$. This formulation assumes all fission neutrons are created in the high energy group ($\chi_1 = 1$ and $\chi_2 = 0$) and there is no

scattering that results in an increase in neutron energy; i.e., no up-scattering ($\Sigma_{s2 \rightarrow 1} = 0$). These are realistic assumptions for typical diffusive neutron systems.

Analytic solutions are provided herein. One-dimensional problems can be replicated in a two-dimensional solver using a square geometry and select boundary conditions. To reduce the dimensionality of a quadrilateral, two of the boundary conditions are set to mirror flux conditions and two are set to zero-flux ($\phi = 0$) conditions. For true two-dimensional and three-dimensional problems, all of the boundary conditions are set to zero-flux conditions.

These formulae are common to second order partial differential equations, but the formulation here is based in part from Lewis [Lew08].

A.2 One-Dimension, One-Group, Fixed Source

This one-dimensional problem is in the domain $x \in [0, L_x]$. The material is homogeneous within the problem and has fixed coefficient properties. The diffusion equation for this problem is

$$-D \frac{d^2}{dx^2} \phi(x) + \Sigma_r \phi(x) = q_{fixed} \quad (\text{A.4})$$

with boundary conditions specified as

$$\phi(0) = 0, \quad (\text{A.5})$$

$$\phi(L_x) = 0. \quad (\text{A.6})$$

Eq. (A.4) can be rewritten as

$$\frac{d^2}{dx^2} \phi(x) - \kappa^2 \phi(x) = -\frac{q_{fixed}}{D} \quad (\text{A.7})$$

Where κ is a shape term and is given for this problem as

$$\kappa^2 = \frac{\Sigma_r}{D}. \quad (\text{A.8})$$

Begin by allowing the solution to be composed of homogeneous and particular solutions.

$$\phi(x) = \phi_H(x) + \phi_P(x) \quad (\text{A.9})$$

Using the form of Eq. (A.7), the homogeneous solution satisfies

$$\frac{d^2}{dx^2} \phi_H(x) - \kappa^2 \phi_H(x) = 0. \quad (\text{A.10})$$

The homogeneous solution $\phi_H(x)$ has form

$$\phi_H(x) = c_1 \cosh(\kappa x) + c_2 \sinh(\kappa x) \quad (\text{A.11})$$

where c_1 and c_2 are unknown constants. The particular solution is given for a constant value q_{fixed} .

$$\phi_P(x) = \frac{q_{fixed}}{D \kappa^2}. \quad (\text{A.12})$$

Combining the homogeneous and particular solutions.

$$\phi(x) = c_1 \cosh(\kappa x) + c_2 \sinh(\kappa x) + \frac{q_{fixed}}{D \kappa^2} \quad (\text{A.13})$$

Next, boundary conditions are considered to solve for the unknown coefficients in Eq. (A.13). Beginning with the boundary at $x = 0$ as specified in Eq. (A.5).

$$\phi(0) = 0 \quad (\text{A.14})$$

$$= c_1 + \frac{q_{fixed}}{D \kappa^2} \quad (\text{A.15})$$

$$\therefore c_1 = -\frac{q_{fixed}}{D \kappa^2} \quad (\text{A.16})$$

Eq. (A.16) is then inserted into Eq. (A.13).

$$\phi(x) = -\frac{q_{fixed}}{D \kappa^2} \cosh(\kappa x) + c_2 \sinh(\kappa x) + \frac{q_{fixed}}{D \kappa^2} \quad (\text{A.17})$$

$$\phi(x) = \frac{q_{fixed}}{D \kappa^2} (1 - \cosh(\kappa x)) + c_2 \sinh(\kappa x) \quad (\text{A.18})$$

The next boundary condition is evaluated at $x = L_x$ as specified in Eq. (A.6).

$$\phi(L_x) = 0 \quad (\text{A.19})$$

$$= \frac{q_{fixed}}{D \kappa^2} (1 - \cosh(\kappa L_x)) + c_2 \sinh(\kappa L_x) \quad (\text{A.20})$$

$$\therefore c_2 = \frac{\frac{q_{fixed}}{D \kappa^2} (\cosh(\kappa L_x) - 1)}{\sinh(\kappa L_x)} \quad (\text{A.21})$$

All constants of the problem are now specified in Eq. (A.16) and Eq. (A.21) and can be substituted into Eq. (A.13).

$$\phi(x) = \left(\frac{q_{fixed}}{D} \right) \left(1 - \cosh(\kappa x) + \frac{\cosh(\kappa L_x) - 1}{\sinh(\kappa L_x)} \sinh(\kappa x) \right) \quad (\text{A.22})$$

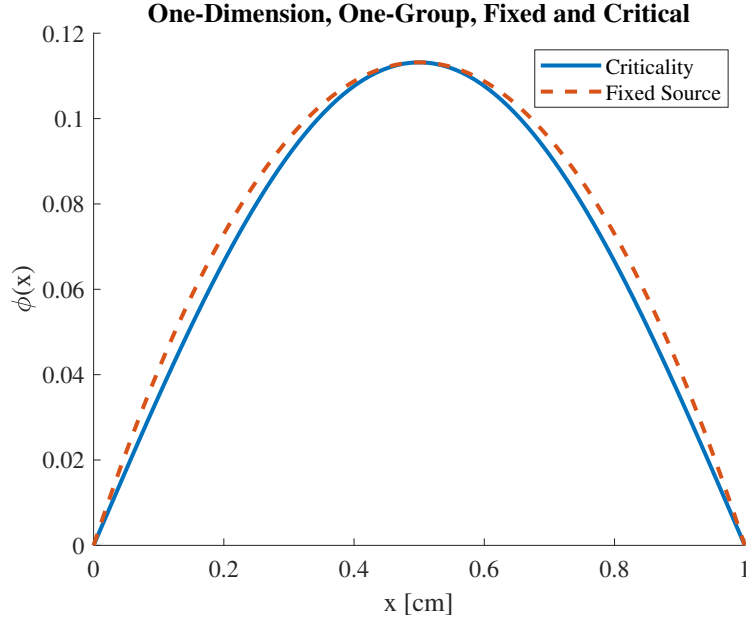


Figure A.1 Fixed Source and Criticality Flux Shapes for One-Dimension, One-Group Problems.

Recall $\kappa^2 = \frac{\Sigma_r}{D}$ as in Eq. (A.8). A typical result of Eq. (A.22) is presented in Fig. A.1. Note that the magnitude of the function is not arbitrary and is specified by the magnitude of the external source q_{fixed} .

A.3 One-Dimension, One-Group, Criticality

This one-dimensional problem is in the domain $x \in [0, L_x]$. The material is homogeneous within the problem and has fixed coefficient properties. The diffusion equation for this problem is

$$-D \frac{d^2}{dx^2} \phi(x) + \Sigma_r \phi(x) = \frac{1}{k_{eff}} \nu \Sigma_f \phi(x) \quad (\text{A.23})$$

with boundary conditions

$$\phi(0) = 0, \quad (\text{A.24})$$

$$\phi(L_x) = 0. \quad (\text{A.25})$$

Eq. (A.23) can be rewritten as

$$\frac{d^2}{dx^2} \phi(x) + B^2 \phi(x) = 0 \quad (\text{A.26})$$

where B is the buckling term defined as

$$B^2 = \frac{\frac{1}{k_{eff}} \nu \Sigma_f - \Sigma_r}{D}. \quad (\text{A.27})$$

Eq. (A.26) has general solution

$$\phi(x) = c_1 \cos(Bx) + c_s \sin(Bx) \quad (\text{A.28})$$

where c_1 and c_2 are unknown constants. Using Eq. (A.28), the first boundary condition is applied at $x = 0$ as specified in Eq. (A.24).

$$\phi(0) = 0 \quad (\text{A.29})$$

$$= c_1 \quad (\text{A.30})$$

$$\therefore c_1 = 0 \quad (\text{A.31})$$

Eq. (A.31) is then inserted into Eq. (A.28), giving

$$\phi(x) = c_2 \sin(Bx). \quad (\text{A.32})$$

Next, the second boundary condition is evaluated at $x = L_x$ as specified in Eq. (A.25).

$$\phi(L_x) = 0, \quad (\text{A.33})$$

$$= c_2 \sin(B L_x). \quad (\text{A.34})$$

For a non-trivial solution, this problem is an eigenvalue problem with B determined by the boundary condition such that

$$0 = \sin(B_g L_x) \quad (\text{A.35})$$

where B_g is the geometric buckling which is specified by the geometry of the problem. This term is only zero for specific values of B_g . This implies c_2 is arbitrary and the magnitude of the flux can be normalized to a constant. For simplicity, allow $c_2 = \phi_0$. Then, noting the zeros of the sine function.

$$B_g L_x = n \pi, \quad (\text{A.36})$$

$$B_g = \frac{n \pi}{L_x}, \quad (\text{A.37})$$

Table A.2 One-Group Sample Cross Sections.

Cross Section	Value
D [cm]	1
Σ_r $\left[\frac{1}{\text{cm}}\right]$	1
$\nu\Sigma_f$ $\left[\frac{1}{\text{cm}}\right]$	2

where n is an integer. For the fundamental mode, $n = 1$,

$$B_g = \frac{\pi}{L_x}. \quad (\text{A.38})$$

Substitute Eq. (A.38) into Eq. (A.32) for the solution to the fundamental eigenmode.

$$\phi(x) = \phi_0 \sin\left(\frac{\pi}{L_x}x\right) \quad (\text{A.39})$$

A typical flux shape is plotted in Fig. A.1 along with the solution for a fixed-source problem. Recall that the magnitude is arbitrary so the flux is normalized to have maximum equal to the fixed solution for visualization purposes. Recalling the definition of B from Eq. (A.27), the fundamental eigenvalue of the problem, k_{eff} , can be solved for by setting buckling equal to geometric buckling.

$$B_g = B \quad (\text{A.40})$$

$$\left(\frac{\pi}{L_x}\right)^2 = \frac{\frac{1}{k_{eff}}\nu\Sigma_f - \Sigma_r}{D} \quad (\text{A.41})$$

$$k_{eff} = \frac{\nu\Sigma_f}{D\left(\frac{\pi}{L_x}\right)^2 + \Sigma_r} \quad (\text{A.42})$$

For a problem with material cross sections given in Table A.2 and $L_x = 100$ [cm], the resulting effective multiplication factor is

$$k_{eff} = 1.998028. \quad (\text{A.43})$$

A.4 Two-Dimension, One-Group, Criticality

This two-dimensional problem is in the rectangular domain $[0, L_x] \times [0, L_y]$. That is, $x \in [0, L_x]$ and $y \in [0, L_y]$. The material is homogeneous within the problem and has fixed coefficient properties. The diffusion equation for this problem is

$$-D\nabla^2\phi(x, y) + \Sigma_r\phi(x, y) = \frac{1}{k_{eff}}\nu\Sigma_f\phi(x, y) \quad (\text{A.44})$$

with boundary conditions

$$\phi(0, y) = 0, \quad (\text{A.45})$$

$$\phi(L_x, y) = 0, \quad (\text{A.46})$$

$$\phi(x, 0) = 0, \quad (\text{A.47})$$

$$\phi(x, L_y) = 0. \quad (\text{A.48})$$

Eq. (A.44) can be rewritten as

$$\nabla^2\phi(x, y) + B^2\phi(x, y) = 0 \quad (\text{A.49})$$

where B is the buckling term and is given for this problem is given as

$$B^2 = \frac{\frac{1}{k_{eff}}\nu\Sigma_f - \Sigma_r}{D}. \quad (\text{A.50})$$

The Laplacian, ∇^2 , in Eq. (A.49) for two-dimensional Cartesian geometry is then expanded into partial derivatives as

$$\frac{\partial^2}{\partial x^2}\phi(x, y) + \frac{\partial^2}{\partial y^2}\phi(x, y) + B^2\phi(x, y) = 0. \quad (\text{A.51})$$

The method of separation of variables is used to solve Eq. (A.51). Begin by assuming that $\phi(x, y)$ is separable as functions of x and y .

$$\phi(x, y) = X(x)Y(y) \quad (\text{A.52})$$

Then, inserting Eq. (A.52) into Eq. (A.51).

$$\frac{\partial^2}{\partial x^2}(X(x)Y(y)) + \frac{\partial^2}{\partial y^2}(X(x)Y(y)) + B^2X(x)Y(y) = 0 \quad (\text{A.53})$$

$$Y(y)\frac{d^2}{dx^2}X(x) + X(x)\frac{d^2}{dy^2}Y(y) + B^2X(x)Y(y) = 0 \quad (\text{A.54})$$

Each of the derivatives operates on a function of only one variable, so the partial derivatives have become standard derivatives. Dividing by the quantity $X(x) Y(y)$ yields

$$\frac{1}{X(x)} \frac{d^2}{dx^2} X(x) + \frac{1}{Y(y)} \frac{d^2}{dy^2} Y(y) + B^2 = 0. \quad (\text{A.55})$$

The first two terms of Eq. (A.55) are functions of only x and y respectively, and their sum is equal to a constant. Therefore, both of the terms must also be equal to a constant [Lam66]. Splitting Eq. (A.55) into two equations yields

$$\frac{1}{X(x)} \frac{d^2}{dx^2} X(x) + \alpha^2 = 0, \quad (\text{A.56})$$

$$\frac{1}{Y(y)} \frac{d^2}{dy^2} Y(y) + \beta^2 = 0, \quad (\text{A.57})$$

where α and β are constants such that

$$\alpha^2 + \beta^2 = B^2. \quad (\text{A.58})$$

Consider Eq. (A.56) and boundary conditions specified in Eq. (A.45) and Eq. (A.46). Then, Eq. (A.56) can be rewritten.

$$\frac{d^2}{dx^2} X(x) + \alpha^2 X(x) = 0 \quad (\text{A.59})$$

The form of Eq. (A.59) is similar to the previous problem in Eq. (A.26) which has general solution of the form

$$X(x) = c_1 \cos(\alpha x) + c_2 \sin(\alpha x). \quad (\text{A.60})$$

$X(x)$ has solution similar to the form of the one-dimension, one-group, criticality problem from §A.3 as the boundary conditions are similar. Using the method from §A.3, the fundamental mode is given as

$$\alpha = \frac{\pi}{L_x}, \quad (\text{A.61})$$

$$X(x) = \phi_{0,x} \sin\left(\frac{\pi}{L_x} x\right), \quad (\text{A.62})$$

where $\phi_{0,x}$ is an arbitrary constant. Similar procedure is performed to solve for $Y(y)$ yielding

$$Y(y) = \phi_{0,y} \sin\left(\frac{\pi}{L_y} y\right) \quad (\text{A.63})$$

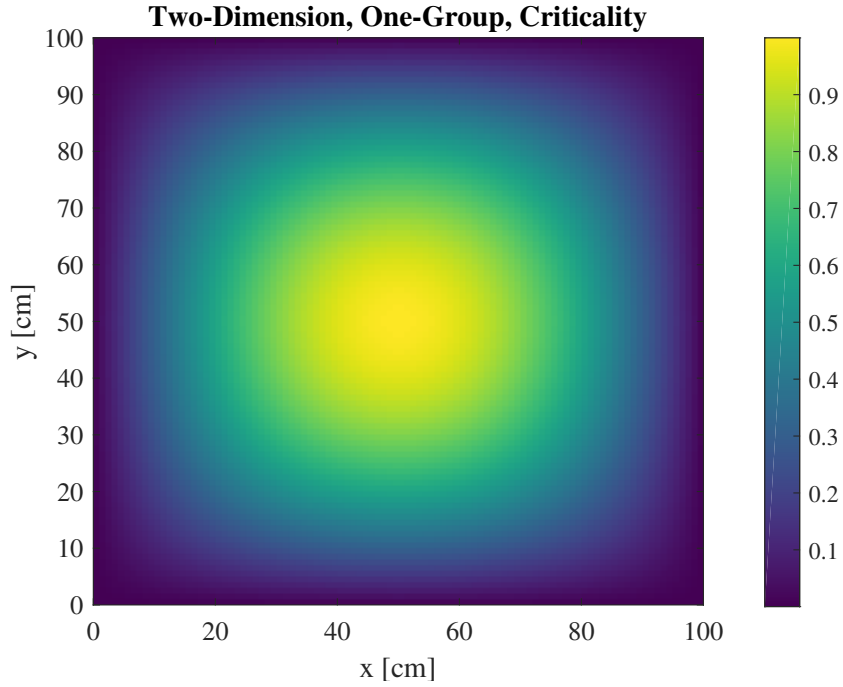


Figure A.2 Two-Dimension Criticality Flux Shape.

and

$$\beta = \frac{\pi}{L_y}. \quad (\text{A.64})$$

Eq. (A.62) and Eq. (A.63) are substituted into Eq. (A.52) giving

$$\phi(x, y) = \phi_0 \sin\left(\frac{\pi}{L_x}x\right) \sin\left(\frac{\pi}{L_y}y\right) \quad (\text{A.65})$$

where $\phi_0 = \phi_{0,x}\phi_{0,y}$ and is also arbitrary. A typical flux shape given by Eq. (A.65) is plotted in Fig. A.2.

Using Eq. (A.58) along with Eq. (A.61) and a Eq. (A.64), the geometric buckling can be written.

$$B_g^2 = \left(\frac{\pi}{L_x}\right)^2 + \left(\frac{\pi}{L_y}\right)^2 \quad (\text{A.66})$$

Setting Eq. (A.66) equal to Eq. (A.50) yields an expression for k_{eff} .

$$\left(\frac{\pi}{L_x}\right)^2 + \left(\frac{\pi}{L_y}\right)^2 = \frac{\frac{1}{k_{eff}} \nu \Sigma_f - \Sigma_r}{D} \quad (\text{A.67})$$

$$k_{eff} = \frac{\nu \Sigma_f}{D \left(\left(\frac{\pi}{L_x}\right)^2 + \left(\frac{\pi}{L_y}\right)^2 \right) + \Sigma_r} \quad (\text{A.68})$$

Using the material coefficients specified in Table A.2, and $L_x = L_y = 100$ [cm] gives

$$k_{eff} = 1.996060. \quad (\text{A.69})$$

A.5 One-Dimension, Two-Group, Criticality

This one-dimensional problem is meant to test the multigroup solution method of the solver. The problem is in the domain $x \in [0, L_x]$ and has two energy groups. The material is homogeneous within the problem and has fixed coefficient properties. With this geometry, Eq. (A.2) and Eq. (A.3) become

$$-D_1 \frac{d^2}{dx^2} \phi_1(x) + \Sigma_{r1} \phi_1(x) = \frac{1}{k_{eff}} (\nu \Sigma_{f1} \phi_1(x) + \nu \Sigma_{f2} \phi_2(x)) \quad (\text{A.70})$$

$$-D_2 \frac{d^2}{dx^2} \phi_2(x) + \Sigma_{r2} \phi_2(x) = \Sigma_{s1 \rightarrow 2} \phi_1(x) \quad (\text{A.71})$$

with boundary conditions

$$\phi_g(0) = 0, \quad (\text{A.72})$$

$$\phi_g(L_x) = 0. \quad (\text{A.73})$$

This is a bare core problem ($\phi = 0$ on all boundaries); therefore, the general solution to Eq. (A.70) and Eq. (A.71) has the form common to one-group bare core problems,

$$\frac{d^2}{dx^2} \phi_g(x) + B^2 \phi_g(x) = 0 \quad \text{for } g = 1, 2 \quad (\text{A.74})$$

where B is a buckling term common to both energy groups (not group specific) [Hen75]. The general solution to this equation is

$$\phi_g(x) = c_{1g} \cos(Bx) + c_{2g} \sin(Bx). \quad (\text{A.75})$$

Boundary conditions are applied next. Beginning with the boundary condition at $x = 0$ as specified in Eq. (A.72) gives

$$\phi_g(0) = 0, \quad (\text{A.76})$$

$$= c_{1g}, \quad (\text{A.77})$$

$$\therefore c_{1g} = 0. \quad (\text{A.78})$$

Then, using Eq. (A.78), Eq. (A.75) can be rewritten as

$$\phi_g(x) = c_{2g} \sin(Bx). \quad (\text{A.79})$$

For notational clarity, let $k_g = c_{2g}$. Evaluating the boundary condition at $x = L_x$ as specified in Eq. (A.73).

$$\phi_g(L_x) = 0, \quad (\text{A.80})$$

$$= k_g \sin(BL_x). \quad (\text{A.81})$$

The buckling B is then specified by the zeros of the sine function. The fundamental mode is given.

$$B = \frac{\pi}{L_x} \quad (\text{A.82})$$

To find the unknown coefficients k_g , differentiate Eq. (A.79) twice with respect to x .

$$\frac{d^2}{dx^2} \phi_g(x) = -B^2 k_g \sin(Bx) \quad (\text{A.83})$$

Insert Eq. (A.79) and Eq. (A.83) into Eq. (A.70) and Eq. (A.71).

$$D_1 B^2 k_1 \sin(Bx) + \Sigma_{r1} k_1 \sin(Bx) = \frac{1}{k_{eff}} (\nu \Sigma_{f1} k_1 \sin(Bx) + \nu \Sigma_{f2} k_2 \sin(Bx)) \quad (\text{A.84})$$

$$D_2 B^2 k_2 \sin(Bx) + \Sigma_{r2} k_2 \sin(Bx) = \Sigma_{s1 \rightarrow 2} k_1 \sin(Bx) \quad (\text{A.85})$$

Dividing through the equations by $\sin(Bx)$.

$$D_1 B^2 k_1 + \Sigma_{r1} k_1 = \frac{1}{k_{eff}} (\nu \Sigma_{f1} k_1 + \nu \Sigma_{f2} k_2) \quad (\text{A.86})$$

$$D_2 B^2 k_2 + \Sigma_{r2} k_2 = \Sigma_{s1 \rightarrow 2} k_1 \quad (\text{A.87})$$

Beginning with Eq. (A.87) and solving for the quantity $\frac{k_2}{k_1}$ and recalling Eq. (A.82).

$$\frac{k_2}{k_1} = \frac{\Sigma_{s1 \rightarrow 2}}{D_2 B^2 + \Sigma_{r2}} \quad (\text{A.88})$$

$$\frac{k_2}{k_1} = \frac{\Sigma_{s1 \rightarrow 2}}{D_2 \left(\frac{\pi}{L_x} \right)^2 + \Sigma_{r2}} \quad (\text{A.89})$$

The quantity $\frac{k_2}{k_1}$ relates the magnitudes of $\phi_1(x)$ and $\phi_2(x)$. The relative magnitude is not arbitrary as expressed in Eq. (A.89) but k_1 is arbitrary. Allow $k_1 = \phi_0$ where ϕ_0 is an arbitrary constant.

Returning to Eq. (A.86) and solving for the eigenvalue k_{eff} .

$$k_{eff} = \frac{\nu \Sigma_{f1} k_1 + \nu \Sigma_{f2} k_2}{D_1 B^2 k_1 + \Sigma_{r1} k_1} \quad (\text{A.90})$$

$$k_{eff} = \frac{\nu \Sigma_{f1} k_1 + \nu \Sigma_{f2} k_2}{D_1 \left(\frac{\pi}{L_x} \right)^2 k_1 + \Sigma_{r1} k_1} \quad (\text{A.91})$$

Divide the top and bottom of the ratio by k_1 .

$$k_{eff} = \frac{\nu \Sigma_{f1} + \nu \Sigma_{f2} \frac{k_2}{k_1}}{D_1 \left(\frac{\pi}{L_x} \right)^2 + \Sigma_{r1}} \quad (\text{A.92})$$

Recalling the quantity $\frac{k_2}{k_1}$ from Eq. (A.89).

The flux shape for $\phi_1(x)$ and $\phi_2(x)$ are given by combining Eq. (A.79), Eq. (A.82), and Eq. (A.89). Combining these equations and rewriting yields

$$\phi_1(x) = \phi_0 \sin \left(\frac{\pi}{L_x} x \right), \quad (\text{A.93})$$

$$\phi_2(x) = \frac{k_2}{k_1} \phi_0 \sin \left(\frac{\pi}{L_x} x \right), \quad (\text{A.94})$$

recalling ϕ_0 is an arbitrary constant. An example solution is plotted in Fig. A.3. The ratio $\frac{k_2}{k_1}$ is given in Eq. (A.89). Finally, k_{eff} is given in Eq. (A.92). The cross sections from VVER440 benchmark, material MAT1, described in §B.2.1 are used for this problem. For ease of reference, the cross sections are also

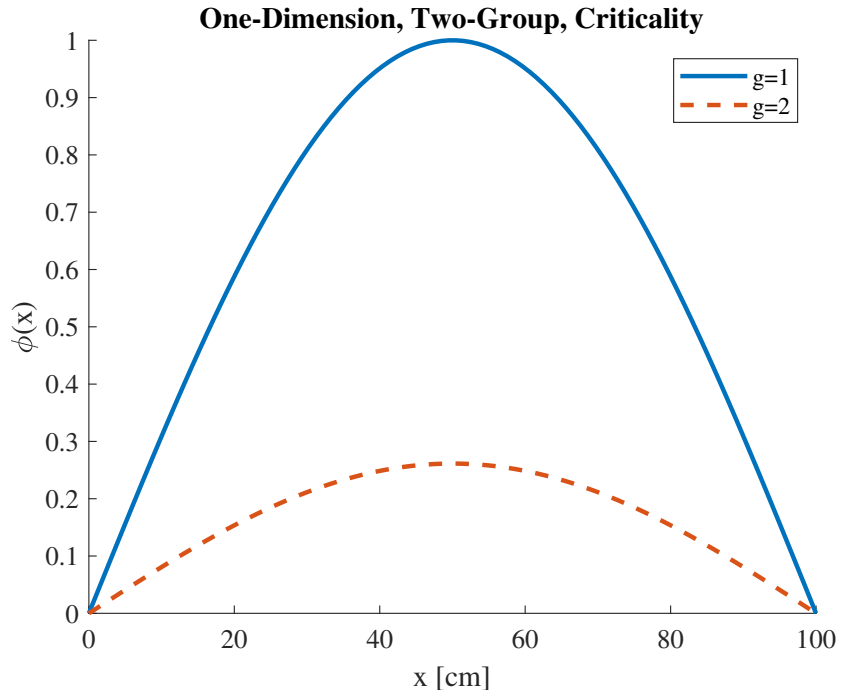


Figure A.3 Example Two-Group Flux Plot.

presented in Table A.3. For the specified cross sections and $L_x = 100$ [cm], the solutions are

$$\frac{k_2}{k_1} = 0.0261324, \quad (\text{A.95})$$

$$k_{eff} = 0.892349. \quad (\text{A.96})$$

Table A.3 Two-Group VVER440 Material Constants.

MAT1	
D_1 [cm]	1.3466E+00
D_2 [cm]	3.7169E-01
Σ_{r1} $\left[\frac{1}{\text{cm}}\right]$	2.5255E-02
Σ_{r2} $\left[\frac{1}{\text{cm}}\right]$	6.4277E-02
$\Sigma_{s1 \rightarrow 2}$ $\left[\frac{1}{\text{cm}}\right]$	1.6893E-02
$\nu \Sigma_{f1}$ $\left[\frac{1}{\text{cm}}\right]$	4.4488E-03
$\nu \Sigma_{f2}$ $\left[\frac{1}{\text{cm}}\right]$	7.3753E-02

A.6 One-Dimension, One-Group, Two-Region, Criticality

This one-dimensional problem is meant to simulate a reactor with non-fissile, reflective material at the edge of the reactor. The purpose of this problem is to test the materials mapping of a multigroup neutron diffusion solver. The problem is in the domain $x \in [0, L_R]$. Geometry is shown in Fig. A.4. Fuel material (subscript F) is located in $x \in [0, L_F]$ and reflector material (subscript R) is located in $x \in [L_F, L_R]$. The material is homogeneous within each respective section, and has fixed coefficient properties in each region. The diffusion equation for this problem is

$$-D_F \frac{d^2}{dx^2} \phi_F(x) + \Sigma_{rF} \phi_F(x) = \frac{1}{k_{eff}} \nu \Sigma_{fF} \phi_F(x) \quad \forall x \in [0, L_F] \quad (\text{A.97})$$

$$-D_R \frac{d^2}{dx^2} \phi_R(x) + \Sigma_{rR} \phi_R(x) = 0 \quad \forall x \in [L_F, L_R] \quad (\text{A.98})$$

with boundary and interfacial conditions

$$\left. \frac{d}{dx} \phi_F(x) \right|_{x=0} = 0, \quad (\text{A.99})$$

$$\phi_F(L_F) = \phi_R(L_F), \quad (\text{A.100})$$

$$D_F \left. \frac{d}{dx} \phi_F(x) \right|_{x=L_F} = D_R \left. \frac{d}{dx} \phi_R(x) \right|_{x=L_F}, \quad (\text{A.101})$$

$$\phi_R(L_R) = 0. \quad (\text{A.102})$$

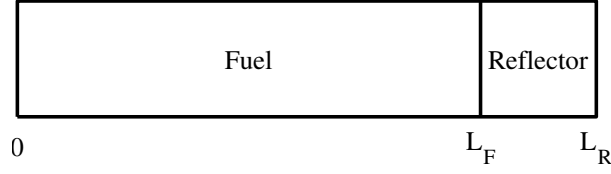


Figure A.4 Geometry for Two-Region Problem.

Beginning in the fueled region, the diffusion equation in Eq. (A.97) can be rewritten with a fuel buckling term B_F as

$$\frac{d^2}{dx^2} \phi_F(x) + B_F^2 \phi_F(x) = 0 \quad (\text{A.103})$$

where B_F^2 is the buckling term in the fuel and is given by

$$B_F^2 = \frac{\frac{1}{k_{eff}} \nu \Sigma_{fF} - \Sigma_{rF}}{D_F}. \quad (\text{A.104})$$

Eq. (A.103) has general solution of the form

$$\phi_F(x) = c_{1F} \cos(B_F x) + c_{2F} \sin(B_F x) \quad (\text{A.105})$$

where c_{1F} and c_{2F} are unknown constant coefficients. The derivative of Eq. (A.105) is also provided as it will be useful for evaluating boundary and interface conditions.

$$\frac{d}{dx} \phi_F(x) = -c_{1F} B_F \sin(B_F x) + c_{2F} B_F \cos(B_F x) \quad (\text{A.106})$$

Using Eq. (A.106), the boundary condition is evaluated at $x = 0$ as specified in Eq. (A.99).

$$\left. \frac{d}{dx} \phi_F(x) \right|_{x=0} = 0, \quad (\text{A.107})$$

$$= c_{2F} B_F, \quad (\text{A.108})$$

$$\therefore c_{2F} = 0. \quad (\text{A.109})$$

Eq. (A.109) is inserted into Eq. (A.105) yielding

$$\phi_F(x) = c_{1F} \cos(B_F x) \quad (\text{A.110})$$

and its derivative is also provided

$$\frac{d}{dx}\phi_F(x) = -c_{1F} B_F \sin(B_F x). \quad (\text{A.111})$$

Next, the reflector region is considered as in Eq. (A.98). This equation can be rewritten as

$$\frac{d^2}{dx^2}\phi_R(x) - \kappa_R^2\phi_R(x) = 0 \quad (\text{A.112})$$

where κ_R^2 is the shape term in the reflector and is given as

$$\kappa_R^2 = \frac{\Sigma_{rR}}{D_R}. \quad (\text{A.113})$$

Eq. (A.112) has general solution of the form

$$\phi_R(x) = c_{1R} \cosh(\kappa_R (x - L_F)) + c_{2R} \sinh(\kappa_R (x - L_F)) \quad (\text{A.114})$$

where c_{1R} and c_{2R} are unknown constant coefficients. Note the use of $(x - L_F)$ instead of x in Eq. (A.114). This choice is valid because hyperbolic cosine and hyperbolic sine can be rewritten as exponential functions and the subtraction in the argument is equivalent to multiplication by a constant. The derivative of Eq. (A.114) is also provided as it will be useful for evaluating boundary conditions.

$$\frac{d}{dx}\phi_R(x) = c_{1R} \kappa_R \sinh(\kappa_R (x - L_F)) + c_{2R} \kappa_R \cosh(\kappa_R (x - L_F)) \quad (\text{A.115})$$

The zero-flux boundary condition at $x = L_R$ is treated as specified in Eq. (A.102).

$$\phi_R(L_R) = 0 \quad (\text{A.116})$$

$$= c_{1R} \cosh(\kappa_R (L_R - L_F)) + c_{2R} \sinh(\kappa_R (L_R - L_F)) \quad (\text{A.117})$$

Then, solving for c_{2R} .

$$c_{2R} = \frac{-c_{1R} \cosh(\kappa_R (L_R - L_F))}{\sinh(\kappa_R (L_R - L_F))} \quad (\text{A.118})$$

$$c_{2R} = -c_{1R} \frac{1}{\tanh(\kappa_R (L_R - L_F))} \quad (\text{A.119})$$

Next, the current continuity condition at $x = L_F$ is considered as specified in Eq. (A.101). Eq. (A.111) and Eq. (A.115) are used to evaluate the functions. Evaluating the boundary condition yields

$$D_F \frac{d}{dx} \phi_F(x) \Big|_{x=L_F} = D_R \frac{d}{dx} \phi_R(x) \Big|_{x=L_F}, \quad (\text{A.120})$$

$$-D_F c_{1F} B_F \sin(B_F L_F) = D_R \kappa_R c_{2R}. \quad (\text{A.121})$$

Then, solving for c_{2R} gives

$$c_{2R} = \frac{-D_F c_{1F} B_F \sin(B_F L_F)}{D_R \kappa_R}. \quad (\text{A.122})$$

Next, the flux continuity condition at $x = L_F$ is considered as specified in Eq. (A.100). Eq. (A.110) and Eq. (A.114) are used to evaluate the functions as

$$\phi_F(L_F) = \phi_R(L_F), \quad (\text{A.123})$$

$$c_{1F} \cos(B_F L_F) = c_{1R}, \quad (\text{A.124})$$

$$c_{1R} = c_{1F} \cos(B_F L_F). \quad (\text{A.125})$$

Now the constant coefficients c are solved. Setting Eq. (A.119) equal to Eq. (A.122) gives

$$-c_{1R} \frac{1}{\tanh(\kappa_R (L_R - L_F))} = \frac{-D_F c_{1F} B_F \sin(B_F L_F)}{D_R \kappa_R}, \quad (\text{A.126})$$

$$c_{1R} \frac{1}{\tanh(\kappa_R (L_R - L_F))} = \frac{D_F c_{1F} B_F \sin(B_F L_F)}{D_R \kappa_R}. \quad (\text{A.127})$$

Inserting the expression for c_{1R} from Eq. (A.125).

$$\frac{c_{1F} \cos(B_F L_F)}{\tanh(\kappa_R (L_R - L_F))} = \frac{D_F c_{1F} B_F \sin(B_F L_F)}{D_R \kappa_R} \quad (\text{A.128})$$

In Eq. (A.128), c_{1F} cancels and is arbitrary. For simplicity, allow $c_{1F} = \phi_0$. This is an expected result of an eigenvalue problem. Continuing with the derivation.

$$\frac{\cos(B_F L_F)}{\tanh(\kappa_R (L_R - L_F))} = \frac{D_F B_F \sin(B_F L_F)}{D_R \kappa_R} \quad (\text{A.129})$$

$$\frac{D_R \kappa_R}{\tanh(\kappa_R (L_R - L_F))} = \frac{D_F B_F \sin(B_F L_F)}{\cos(B_F L_F)} \quad (\text{A.130})$$

Leading to the transcendental relation

$$D_F B_F \tan(B_F L_F) = \frac{D_R \kappa_R}{\tanh(\kappa_R (L_R - L_F))} \quad (\text{A.131})$$

where κ_R is defined according to Eq. (A.113). Eq. (A.131) is the most simplified solution for B_F . Unfortunately, there is no analytic expression for B_F . Therefore, a numeric solver such as MATLAB's `vpasolve()` or a generic bisection method must be used. The tangent function causes the solution B_F to be especially sensitive to the starting guess or bounds of the search. Once, B_F is known, the eigenvalue, k_{eff} can be solved using Eq. (A.104).

$$k_{eff} = \frac{\nu \Sigma_{fF}}{D_F B_F^2 + \Sigma_{rF}} \quad (\text{A.132})$$

The fundamental eigenmode can be expressed compactly as well recalling the general forms Eq. (A.110) and Eq. (A.114) as well as coefficients Eq. (A.125) and Eq. (A.122).

$$\phi_F(x) = \phi_0 \cos(B_F x) \quad (\text{A.133})$$

$$\phi_R(x) = \phi_0 \cos(B_F L_F) \left(\cosh(\kappa_R (x - L_F)) - \frac{\sinh(\kappa_R (x - L_F))}{\tanh(\kappa_R (L_R - L_F))} \right) \quad (\text{A.134})$$

$$\phi(x) = \begin{cases} \phi_F(x) & 0 \leq x \leq L_F \\ \phi_R(x) & L_F \leq x \leq L_R \end{cases} \quad (\text{A.135})$$

Where ϕ_0 is the arbitrary normalization constant. For $L_F = 80$ [cm] and $L_R = 100$ [cm] and material cross sections given in Table A.4, the following are solutions. Note B_F is given to approximately machine precision (16 significant digits) as the solution is particularly sensitive to this value. Additionally, a flux shape is shown in Fig. A.5.

$$B_F = 0.0171685973659098101 \left[\frac{1}{\text{cm}} \right] \quad (\text{A.136})$$

$$k_{eff} = 0.982622 \quad (\text{A.137})$$

Table A.4 Two-Region Material Constants.

	Fuel	Reflector
D [cm]	1.2	0.7
Σ_r [$\frac{1}{\text{cm}}$]	0.02	0.015
$\nu\Sigma_f$ [cm]	0.02	

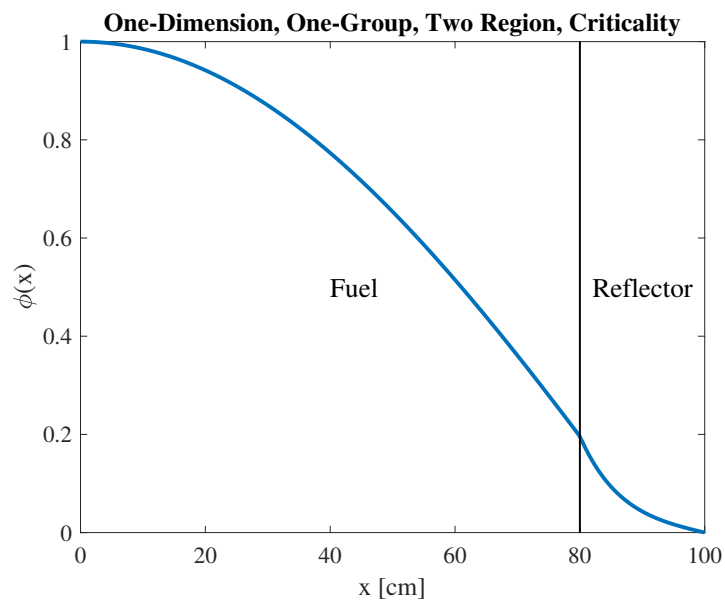


Figure A.5 Two-Region Flux Shape.

A.7 Finite-Cylinder, One-Group, Criticality

This three-dimensional problem can be simulated in three Cartesian dimensions and, using symmetry, reduced to two cylindrical dimensions for simplified analytic solution. This problem is located in $r \in [0, T]$ and $z \in [0, H]$ where r and z are the radial and axial coordinates respectively. The material is homogeneous within the problem and has fixed coefficient properties. The diffusion equation for this problem is

$$-D\nabla^2\phi(r, z) + \Sigma_r\phi(r, z) = \frac{1}{k_{eff}}\nu\Sigma_f\phi(r, z) \quad (\text{A.138})$$

with boundary conditions

$$\phi(r, 0) = 0, \quad (\text{A.139})$$

$$\phi(r, H) = 0, \quad (\text{A.140})$$

$$\phi(T, z) = 0, \quad (\text{A.141})$$

$$\phi(0, z) \text{ is Finite.} \quad (\text{A.142})$$

Eq. (A.138) can be rewritten

$$\nabla^2\phi(r, z) + B^2\phi(r, z) = 0 \quad (\text{A.143})$$

where B is the buckling term and is given for this problem as

$$B^2 = \frac{\frac{1}{k_{eff}}\nu\Sigma_f - \Sigma_r}{D}. \quad (\text{A.144})$$

The Laplacian in Eq. (A.143) is expanded in two dimension cylindrical coordinates

$$\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial}{\partial r} \phi(r, z) \right) + \frac{\partial^2}{\partial z^2} \phi(r, z) + B^2\phi(r, z) = 0. \quad (\text{A.145})$$

Using the method of separation of variables, assume $\phi(r, z)$ is the product of two functions as

$$\phi(r, x) = R(r) Z(z). \quad (\text{A.146})$$

Inserting Eq. (A.146) into Eq. (A.145).

$$\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial}{\partial r} (R(r)Z(z)) \right) + \frac{\partial^2}{\partial z^2} (R(r)Z(z)) + B^2 R(r) Z(z) = 0 \quad (\text{A.147})$$

$$Z(z) \frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial}{\partial r} R(r) \right) + R(r) \frac{\partial^2}{\partial z^2} Z(z) + B^2 R(r) Z(z) = 0 \quad (\text{A.148})$$

Divide by the quantity $R(r) Z(z)$.

$$\frac{1}{R(r)} \frac{1}{r} \frac{d}{dr} \left(r \frac{d}{dr} R(r) \right) + \frac{1}{Z(z)} \frac{d^2}{dz^2} Z(z) + B^2 = 0 \quad (\text{A.149})$$

Each of the derivatives operates on a function of only one variable, so the partial derivatives have become standard derivatives.

The first two terms of Eq. (A.149) are function of only r and z respectively, and their sum is equal to a constant. Therefore, both of the terms must also be equal to a constant [Lam66]. Splitting Eq. (A.149) into two equations yields

$$\frac{1}{R(r)} \frac{1}{r} \frac{d}{dr} \left(r \frac{d}{dr} R(r) \right) + \beta^2 = 0, \quad (\text{A.150})$$

$$\frac{1}{Z(z)} \frac{d^2}{dz^2} Z(z) + \gamma^2 = 0, \quad (\text{A.151})$$

where

$$B^2 = \beta^2 + \gamma^2. \quad (\text{A.152})$$

Beginning with the radial direction from Eq. (A.150).

$$\frac{1}{r} \frac{d}{dr} \left(r \frac{d}{dr} R(r) \right) + \beta^2 R(r) = 0 \quad (\text{A.153})$$

Multiplying by the radial coordinate r .

$$\frac{d}{dr} \left(r \frac{d}{dr} R(r) \right) + \beta^2 r = 0 \quad (\text{A.154})$$

Noting the product rule of differentiation.

$$r \frac{d}{dr} R(r) + R(r) + \beta^2 R(r) = 0 \quad (\text{A.155})$$

Dividing by the radial coordinate r .

$$\frac{d}{dr}R(r) + \frac{1}{r}R(r) + \beta^2 R(r) = 0 \quad (\text{A.156})$$

Eq. (A.156) has general solution of the form

$$R(r) = c_1 J_0(\beta r) + c_2 Y_0(\beta r) \quad (\text{A.157})$$

where J_0 is the Bessel function of the first kind, zeroth order and Y_0 is the Bessel function of the second kind, zeroth order. Requiring the flux to be finite at $r = 0$ as per Eq. (A.142).

$$\lim_{r \rightarrow 0} Y_0(r) \rightarrow -\infty, \quad (\text{A.158})$$

$$\therefore c_2 = 0. \quad (\text{A.159})$$

Eq. (A.159) is inserted into Eq. (A.157).

$$R(r) = c_1 J_0(\beta r) \quad (\text{A.160})$$

Evaluating the boundary condition at $r = T$ using Eq. (A.160) as specified in Eq. (A.141).

$$R(T) = 0, \quad (\text{A.161})$$

$$= c_1 J_0(\beta T). \quad (\text{A.162})$$

For non-trivial $R(r)$, this is an eigenvalue problem. Then, c_1 is an arbitrary constant and the eigenmode is specified as

$$\beta T = \alpha_n \quad (\text{A.163})$$

$$\beta = \frac{\alpha_n}{T} \quad (\text{A.164})$$

where α_n are the zeros of the J_0 function such that $J_0(\alpha_n) = 0$. The fundamental mode is given for $n = 0$ as

$$\beta = \frac{\alpha_0}{T} \quad (\text{A.165})$$

where α_0 is the first zero ($\alpha_0 \approx 2.4048$). Then the solution for the function $R(r)$ is

$$R(r) = c_1 J_0\left(\frac{\alpha_0}{T}r\right). \quad (\text{A.166})$$

A similar process is repeated for $Z(z)$. The solution to $Z(z)$ is similar to the one-dimension, one-group, criticality problem presented in §A.3. This derivation is briefly repeated here. Recall Eq. (A.151) which may be rewritten.

$$\frac{d^2}{dz^2} Z(z) + \gamma^2 Z(z) = 0 \quad (\text{A.167})$$

which has general solution of the form

$$Z(z) = c_3 \cos(\gamma z) + c_4 \sin(\gamma z). \quad (\text{A.168})$$

Evaluating Eq. (A.168) at $z = 0$ considering the boundary condition specified in Eq. (A.139).

$$Z(0) = 0, \quad (\text{A.169})$$

$$= c_3, \quad (\text{A.170})$$

$$\therefore c_3 = 0. \quad (\text{A.171})$$

Substituting Eq. (A.171) into Eq. (A.168).

$$Z(z) = c_4 \sin(\gamma z) \quad (\text{A.172})$$

Evaluating Eq. (A.172) at $z = H$ and considering boundary condition from Eq. (A.140).

$$Z(H) = 0, \quad (\text{A.173})$$

$$= c_4 \sin(\gamma H). \quad (\text{A.174})$$

For non-trivial $Z(z)$, the term $\sin(\gamma H) = 0$ is required and c_4 is trivial and γ specified by geometry as

$$\gamma H = n\pi, \quad (\text{A.175})$$

$$\gamma = \frac{n\pi}{H}, \quad (\text{A.176})$$

$$(\text{A.177})$$

where n is an integer. For the fundamental mode, $n = 1$,

$$\gamma = \frac{\pi}{H}. \quad (\text{A.178})$$

Coefficient c_4 is arbitrary. Then

$$Z(z) = c_4 \sin\left(\frac{\pi}{H} z\right) \quad (\text{A.179})$$

Table A.5 Finite Cylinder Cross Sections.

Cross Section	Value
D [cm]	1
Σ_r $\left[\frac{1}{\text{cm}}\right]$	1
$\nu\Sigma_f$ $\left[\frac{1}{\text{cm}}\right]$	1

Eq. (A.166) and Eq. (A.179) are combined according to the separation of variables from Eq. (A.146)

$$\phi(r, z) = \phi_0 J_0 \left(\frac{\alpha_0}{T} r \right) \sin \left(\frac{\pi}{H} z \right) \quad (\text{A.180})$$

where ϕ_0 is arbitrary and $\phi_0 = c_1 c_4$. An example flux plot for a slice of the cylinder is shown in Fig. A.6. Recall Eq. (A.144) and the form Eq. (A.152). Coefficients β and γ are given in Eq. (A.165) and Eq. (A.178). Then, an expression for k_{eff} can be constructed.

$$k_{eff} = \frac{\nu\Sigma_f}{D \left(\left(\frac{\alpha_0}{T} \right)^2 + \left(\frac{\pi}{H} \right)^2 \right) + \Sigma_r} \quad (\text{A.181})$$

For $T = 50$ [cm], $H = 100$ [cm], and material cross sections specified in Table A.5

$$k_{eff} = 0.996711 \quad (\text{A.182})$$

The shape of the boundary as described by the discrete mesh is extremely important for a circular (or curved) boundary. A simple example of this is shown in Fig. A.7. In a mesh refinement study, the mesh must be regenerated with a halved mesh parameter, h , for each refinement, rather than simply splitting nodes as splitting nodes would not improve the description of the boundary.

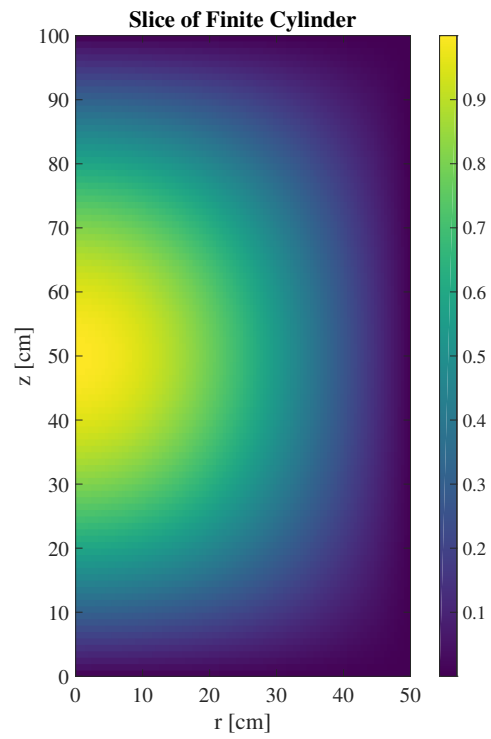


Figure A.6 Example Finite Cylinder Flux Shape.

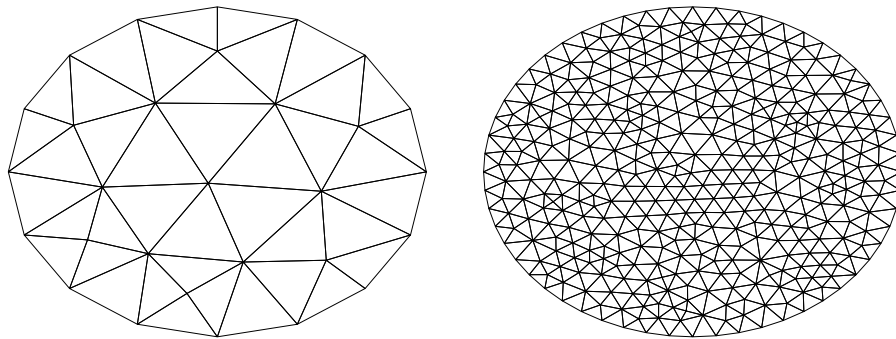


Figure A.7 Mesh Refinement of Curved Mesh.

APPENDIX

B

BRIEF COMPENDIUM OF NEUTRON DIFFUSION BENCHMARKS

B.1 Introduction

This appendix includes a description of several benchmark reactor problems. These include two-dimensional and three-dimensional reactor benchmark problems. These benchmark descriptions are not original works but replications of data obtained from other sources. The intent is to make the data more easily accessible for others solving similar problems. The benchmarks presented in this appendix are summarized in the case matrix in Table B.1. The benchmarks presented in this appendix include two-dimensional and three-dimensional geometries as well as a variety of energy group structures.

All reactors in this section are based on a hexagonal design common to fast reactors. However, not all benchmarks presented are fast reactors. These benchmarks were used to perform solution verification for the neutron diffusion equation solution method in Chapter 2. As such, the cross sections presented are those necessary to solve the multigroup neutron diffusion equation.

Table B.1 Case Matrix for Benchmark Solutions.

Benchmark	Dimensions	Groups	Reactor Type	Neutron Spectrum
VVER440	2	2	LWR	Thermal
SNR	2	4	SFR	Fast
HWR	2	2	HWR	Thermal
IAEA ($\times 4$)	2	2	PWR	Thermal
MONJU	3	3	SFR	Fast
KNK	3	4	SFR	Fast

For each benchmark problem, geometry and cross sections are presented. The notation for cross sections are common to the field and noted below.

D_g = diffusion coefficient for energy group g [cm],

$\Sigma_{r,g}$ = macroscopic removal cross section for energy group g [$\frac{1}{\text{cm}}$],

$\nu\Sigma_{f,g}$ = number of fission neutrons times macroscopic fission cross section in energy group g [$\frac{1}{\text{cm}}$],

$\Sigma_{s,g'\rightarrow g}$ = macroscopic scatter cross section from energy group g' to energy group g [$\frac{1}{\text{cm}}$],

χ_g = effective fission spectrum for energy group g ,

k_{ref} = effective neutron multiplication factor.

A reference k_{ref} is presented for each benchmark to the precision of the solution. When available, reference power distributions are presented. Note that the reference solutions presented are not analytic or exact solutions. The solutions presented in this appendix are provided from other sources. The precision of the reference depends on the source.

B.2 Two-Dimensional Benchmark Problems

The two-dimensional benchmarks presented represent various energy group structures, geometries, assembly sizes, and boundary conditions. By varying these parameters, the combination of these benchmarks constitutes a rigorous testing suite that can be used for solution verification of general multigroup neutron diffusion solvers.

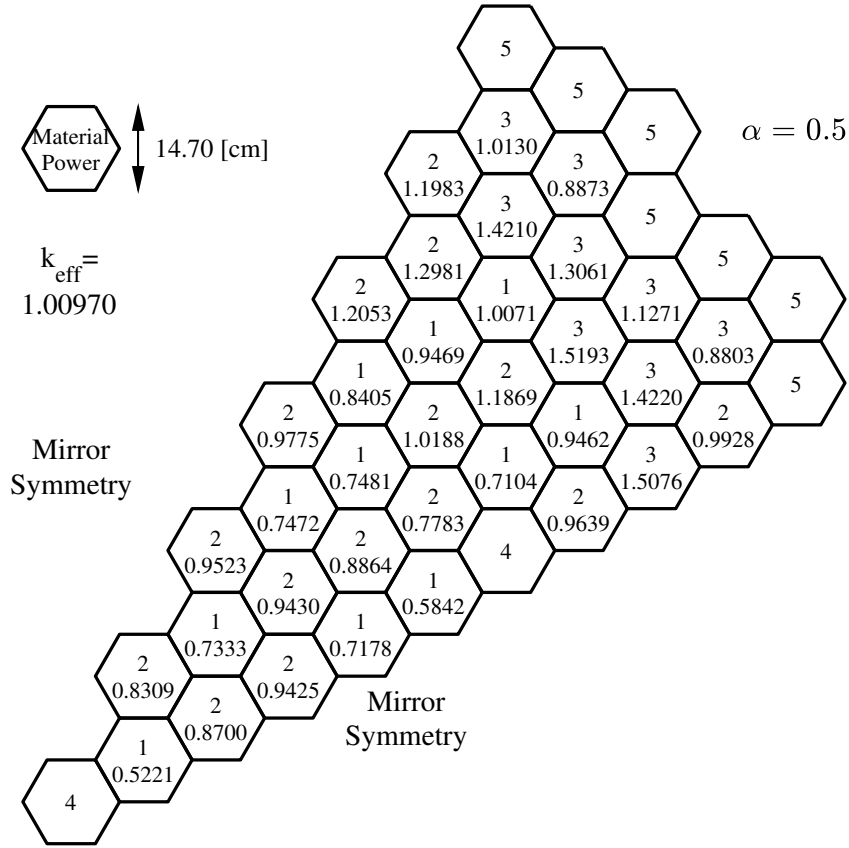


Figure B.1 VVER440 Geometry.

B.2.1 VVER440

This benchmark is presented with solution by Chao & Shatilla [Cha95]. The problem is one-twelfth of a VVER-440 reactor with seven control rods inserted. There are 25 assemblies across the core diameter. The last ring of assemblies at the core periphery are non-fissile reflector assemblies as shown in Fig. B.1. Each assembly has a flat-to-flat measurement of 14.70 [cm]. Vacuum boundary condition ($\alpha = 0.5$) is applied on the core periphery and all other boundaries are mirror boundary conditions ($\nabla \phi_g \cdot \hat{\mathbf{n}} = 0$). To the precision of the benchmark, the effective neutron multiplication factor is $k_{ref} = 1.00970$. Two-group cross sections for the problem are given in Table B.2. The fission spectrum is given in Table B.3. Assembly powers are given in Fig. B.1.

Table B.2 VVER440 Cross Sections.

	MAT1	MAT2	MAT3	MAT4	MAT5
D_1	1.346600E+00	1.337700E+00	1.332200E+00	1.195300E+00	1.448500E+00
D_2	3.716900E-01	3.691800E-01	3.650200E-01	1.931300E-01	2.517600E-01
Σ_{r1}	2.525500E-02	2.470900E-02	2.435000E-02	3.563600E-02	3.318400E-02
Σ_{r2}	6.427700E-02	7.936100E-02	1.001000E-01	1.349800E-01	3.283900E-02
$\Sigma_{s1 \rightarrow 2}$	1.689300E-02	1.591200E-02	1.488800E-02	2.226400E-02	3.226200E-02
$\nu \Sigma_{f1}$	4.448800E-03	5.533700E-03	7.039100E-03		
$\nu \Sigma_{f2}$	7.375300E-02	1.058100E-01	1.496400E-01		

Table B.3 VVER440 Fission Spectrum.

Fission Spectrum	
χ_1	1.0
χ_2	0.0

B.2.2 SNR

This benchmark is presented with solution in the Argonne Code Center Benchmark Problem Book [Ame77]. It is based on the MARK-I core design of the SNR-300. The problem is one-sixth of a reactor with no control rods inserted. There are 17 assemblies across the core diameter with the outer rings being blanket assemblies as shown in Fig. B.2. Each assembly has a flat-to-flat measurement of 11.20 [cm]. Vacuum boundary condition ($\alpha = 0.5$) is applied on the core periphery and all other boundaries are mirror boundary conditions ($\nabla \phi_g \cdot \hat{\mathbf{n}} = 0$). To the precision of the benchmark, the effective neutron multiplication factor is $k_{ref} = 1.124$. Four-group cross sections for the problem are given in Table B.4. The fission spectrum is given in Table B.5. Assembly powers are not provided in the benchmark but can be calculated from existing codes (e.g. DIF3D).

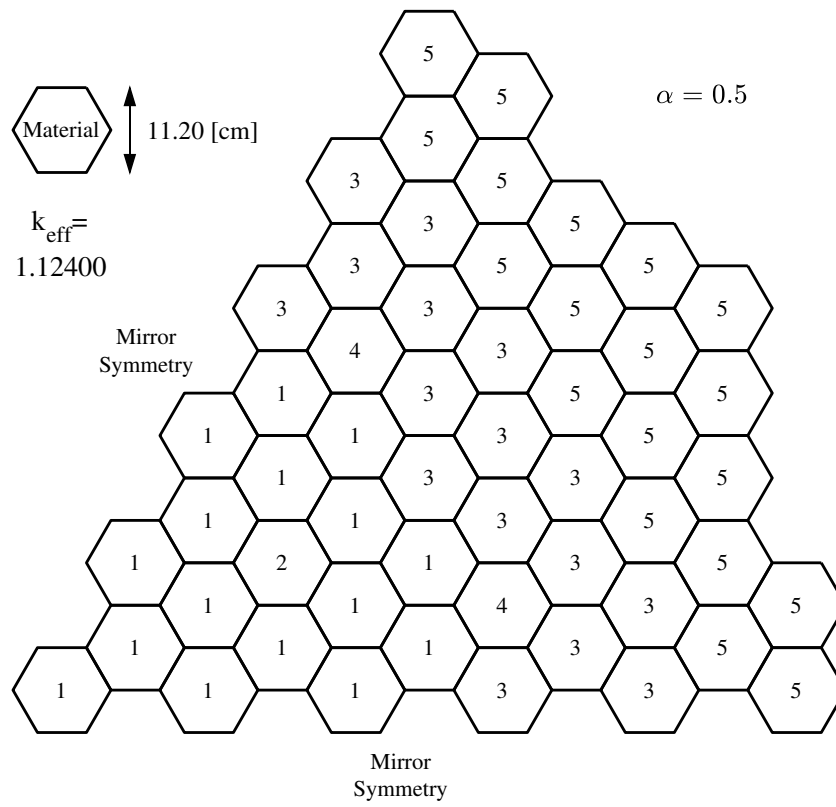


Figure B.2 SNR Geometry.

Table B.4 SNR Cross Sections.

	MAT1	MAT2	MAT3	MAT4	MAT5	MAT6
D_1	2.876787E+00	2.876539E+00	2.285610E+00	2.716653E+00	2.503066E+00	4.616422E+00
D_2	1.570845E+00	1.571363E+00	1.171935E+00	1.440943E+00	1.314665E+00	2.901831E+00
D_3	7.224859E-01	7.127076E-01	6.324751E-01	7.203469E-01	5.742770E-01	1.021179E+00
D_4	9.641993E-01	9.429781E-01	8.183574E-01	9.876836E-01	6.153695E-01	1.729625E+00
Σ_{r1}	2.820400E-02	2.878200E-02	3.595900E-02	2.909300E-02	2.481400E-02	1.315900E-02
Σ_{r2}	5.274700E-03	6.049100E-03	5.885500E-03	4.490900E-03	1.641200E-02	1.455900E-03
Σ_{r3}	1.761200E-02	1.951000E-02	1.604100E-02	1.308200E-02	7.212200E-02	4.600100E-03
Σ_{r4}	2.654600E-02	3.371400E-02	1.334900E-02	9.956200E-03	1.686800E-01	7.866000E-04
$\Sigma_{s1 \rightarrow 2}$	2.359700E-02	2.326200E-02	3.207100E-02	2.632200E-02	2.294600E-02	1.294200E-02
$\Sigma_{s1 \rightarrow 3}$	4.079100E-06	4.645100E-06	3.888000E-06	2.890700E-06	1.032000E-06	6.878000E-07
$\Sigma_{s2 \rightarrow 3}$	1.615300E-03	1.571800E-03	2.777600E-03	2.288900E-03	3.768700E-03	1.287100E-03
$\Sigma_{s1 \rightarrow 4}$	4.449300E-08	4.996800E-08	4.503900E-08	3.324800E-08	1.048900E-08	6.990300E-09
$\Sigma_{s2 \rightarrow 4}$	4.230900E-08	4.072400E-08	9.001800E-08	6.213300E-08	7.036100E-12	4.363300E-12
$\Sigma_{s3 \rightarrow 4}$	4.683800E-03	4.341400E-03	5.897100E-03	5.353600E-03	8.681500E-03	3.453300E-03
$\nu\Sigma_{f1}$	1.187800E-02	1.494300E-02	7.742700E-03	5.427900E-03		
$\nu\Sigma_{f2}$	5.325200E-03	7.688700E-03	1.082500E-04	7.585700E-05		
$\nu\Sigma_{f3}$	1.047100E-02	1.480900E-02	2.974200E-04	2.121799E-04		
$\nu\Sigma_{f4}$	2.661100E-02	3.815900E-02	8.468699E-04	5.759200E-04		

Table B.5 SNR Fission Spectrum.

Fission Spectrum	
χ_1	0.768
χ_2	0.232
χ_3	0.000
χ_4	0.000

B.2.3 HWR

This benchmark is presented with solution by Chao & Shatilla [Cha95]. It is based on a very large Heavy Water Reactor (HWR). The problem is one-sixth rotationally symmetric (*not* reflective). For code packages without rotational boundary conditions, it will be necessary to simulate a full core. The problem is large with 35 assemblies across the core diameter as shown in Fig. B.3. Each assembly has a flat-to-flat measurement of 17.78 [cm]. The fueled assemblies are surrounded by a tritium generating zone which is then surrounded by a reflector zone. Zero-flux boundary condition ($\phi_g = 0$) is applied on the core periphery. This is the only benchmark with such a boundary condition. Two-group cross sections are provided in Table B.6. The fission spectrum is specified in Table B.7. To the precision of the benchmark, the effective neutron multiplication factor is $k_{ref} = 0.991965$. Assembly powers are given in Fig. B.3.

B.2.4 IAEA Hex

The IAEA Hex Pressurized Water Reactor (PWR) is presented with solution by Chao & Shatilla [Cha95]. This benchmark is based on the two-dimensional IAEA PWR benchmark for Cartesian geometry. The problem is one-twelfth of a reactor core. This benchmark has a total of four cases, with and without a reflector ring added to the external core and with the albedo condition set to $\alpha = 0.5$ and $\alpha = 0.125$. Each assembly has a flat-to-flat measurement of 20.00 [cm]. In the unreflected case, there are seven 15 assemblies across the core diameter. In the reflected case, there are 16 assemblies across the core diameter. To the precision of the benchmark, the effective neutron multiplication factors for each of the four cases are summarized in Table B.8. Assembly geometry and powers are provided geometries with and without reflector in Fig. B.4. Two-group cross sections are provided in Table B.9. Fission spectrum is provided in Table B.10.

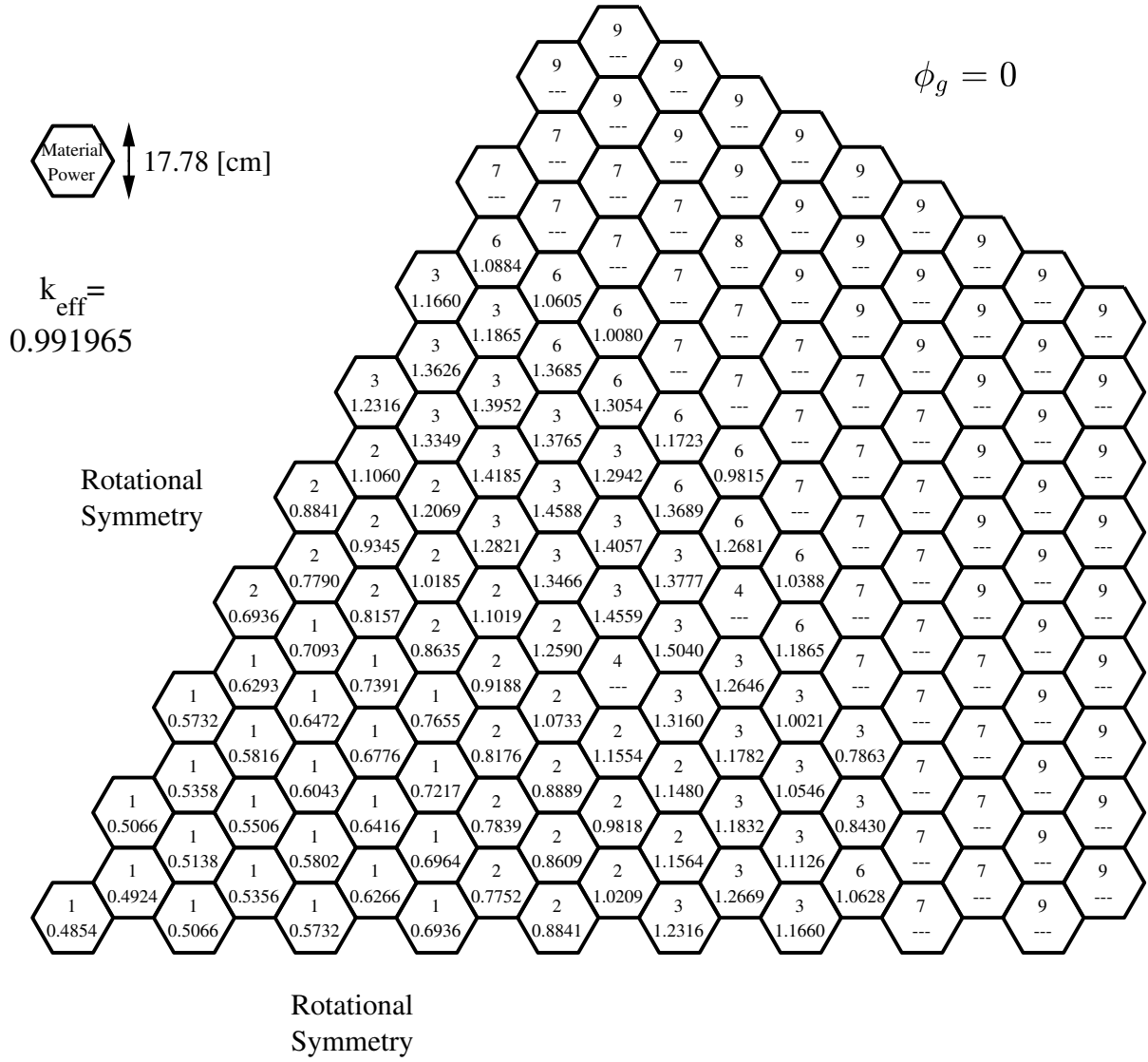


Figure B.3 HWR Geometry.

Table B.6 HWR Cross Sections.

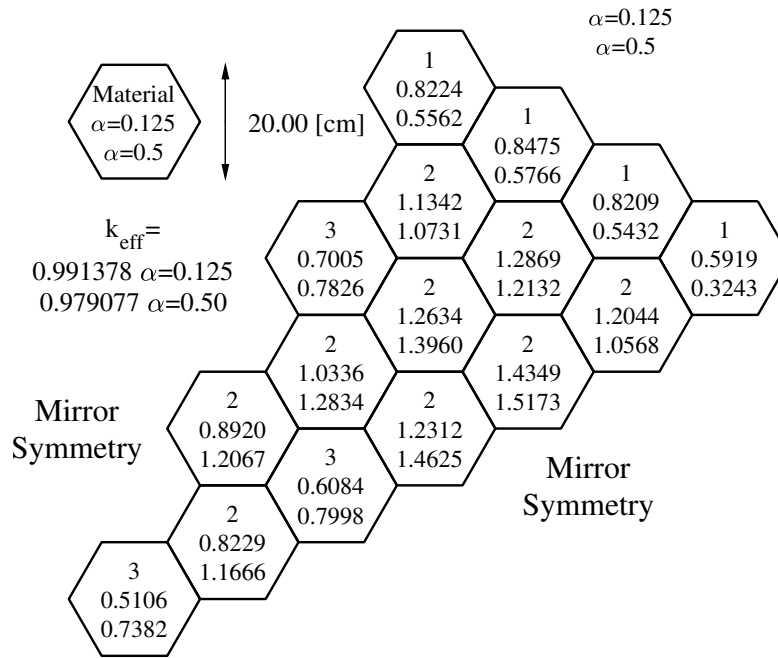
	MAT1	MAT2	MAT3	MAT4	MAT5
D_1	1.382500E+00	1.382550E+00	1.374420E+00	1.311980E+00	1.200000E+00
D_2	8.975220E-01	8.974900E-01	8.883680E-01	8.799140E-01	9.000010E-01
Σ_{r1}	1.110580E-02	1.117460E-02	1.062040E-02	1.268800E-02	1.268800E-02
Σ_{r2}	2.230650E-02	2.238760E-02	1.694650E-02	5.290090E-04	5.300000E-04
$\Sigma_{s1 \rightarrow 2}$	8.164570E-03	8.223780E-03	8.088160E-03	1.231150E-02	1.231150E-02
$\nu\Sigma_{f1}$	2.262160E-03	2.227500E-03	2.142810E-03		
$\nu\Sigma_{f2}$	2.306230E-02	2.268490E-02	2.048870E-02		
	MAT6	MAT7	MAT8	MAT9	
D_1	1.381390E+00	1.305990E+00	1.291930E+00	1.065100E+00	
D_2	9.036710E-01	8.372560E-01	8.193410E-01	3.228290E-01	
Σ_{r1}	1.056310E-02	1.173130E-02	1.191530E-02	2.834620E-02	
Σ_{r2}	2.190300E-02	4.333040E-03	3.005650E-04	3.334890E-02	
$\Sigma_{s1 \rightarrow 2}$	7.765680E-03	1.109750E-02	1.155820E-02	2.619800E-02	
$\nu\Sigma_{f1}$	2.394690E-03				
$\nu\Sigma_{f2}$	2.662100E-02				

Table B.7 HWR Fission Spectrum.

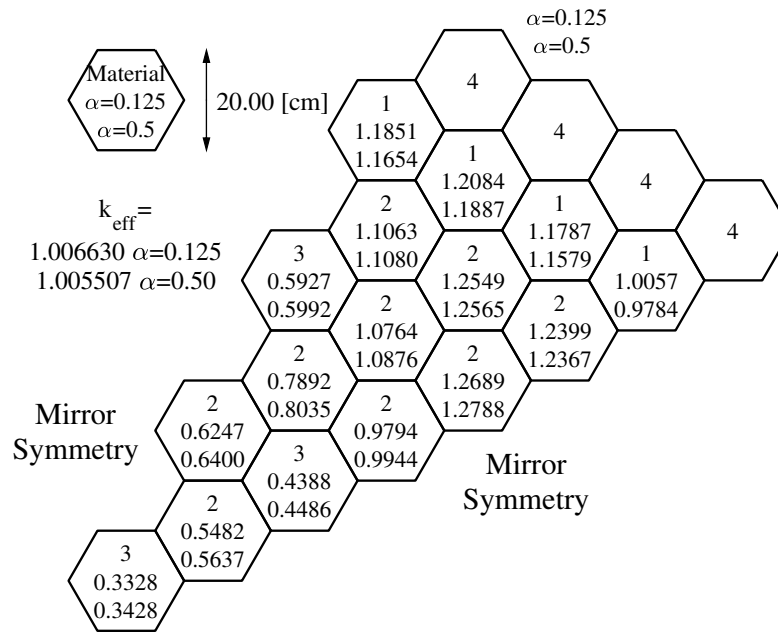
Fission Spectrum	
χ_1	1.0
χ_2	0.0

Table B.8 IAEA Hex Effective Neutron Multiplication Factors.

Reflector	albedo (α)	k_{ref}
Without	0.125	0.991378
Without	0.5	0.978077
With	0.125	1.006630
With	0.5	1.005507



(a) Unreflected IAEA Hex.



(b) Reflected IAEA Hex.

Figure B.4 IAEA Hex Geometry.

Table B.9 IAEA Hex Cross Sections.

	MAT1	MAT2	MAT3	MAT4
D_1	1.500001E+00	1.500000E+00	1.500001E+00	1.500000E+00
D_2	4.000001E-01	4.000000E-01	4.000001E-01	4.000000E-01
Σ_{r1}	3.000000E-02	3.000000E-02	3.000000E-02	4.000000E-02
Σ_{r2}	8.000000E-02	8.500000E-02	1.300000E-01	1.000000E-02
$\Sigma_{s1 \rightarrow 2}$	2.000000E-02	2.000000E-02	2.000000E-02	4.000000E-02
$\nu \Sigma_{f1}$	0.000000E+00	0.000000E+00	0.000000E+00	
$\nu \Sigma_{f2}$	1.350000E-01	1.350000E-01	1.350000E-01	

Table B.10 IAEA Hex Fission Spectrum.

Fission Spectrum	
χ_1	1.0
χ_2	0.0

B.3 Three-Dimensional Benchmark Problems

For three-dimensional benchmarks, control rod worth measurement is used to compare diffusion solution to the benchmark solution. To measure control rod worth, three cases are modeled, $\{A, B, C\}$, with control rods fully removed in case A , partially inserted in case B , and fully inserted in case C . Rod worth is presented in units $[\Delta k]$ and calculated as

$$\text{Rod Worth}_x [\Delta k] = \frac{k_{eff,A} - k_{eff,x}}{k_{eff,A} k_{eff,x}} \quad (\text{B.1})$$

for $x = \{B, C\}$. That is, rod worth is always compared to the case with control rods fully removed, case A . Additionally, Rod Difference is presented in units $[\% \Delta k]$ as

$$\text{Rod Difference}_x [\% \Delta k] = (k_{eff,A} - k_{eff,x}) \times 100\% \quad (\text{B.2})$$

for $x = \{B, C\}$.

Table B.11 MONJU Effective Neutron Multiplication Factors and Rod Worths.

Pattern	k_{ref}	Rod Worth [Δk]	Rod Difference [$\% \Delta k$]
A	1.0723		
B	1.0464	0.023	2.59
C	1.0224	0.046	4.99

B.3.1 MONJU

This benchmark is presented with solution by Komano et al. [Kom78]. It is based on a prototype Fast Breeder Reactor (FBR) with hexagonal pitch. The reactor has one-third rotational (*not* reflective) symmetry. For code packages without rotational boundary conditions, it will be necessary to simulate a full core. There are 21 assemblies across the core diameter with outer rings being blanket assemblies as shown in Fig. B.5. Fuel assemblies are shown axially in Fig. B.6a. Each assembly has a flat-to-flat measurement of 11.56 [cm]. Vacuum boundary condition ($\alpha = 0.5$) is applied on the core periphery and above and below the reactor. Three-group cross sections are provided in Table B.12. No fission spectrum is specified in the benchmark. The fission spectrum was estimated and is shown in Table B.13. In determining this fission spectrum, it was observed that the benchmark results are not highly sensitive to the fission spectrum.

The reactor is simulated with three different control rod configurations; patterns A, B, and C. In pattern A, all control rods are withdrawn and control rod channels are simulated with sodium in the channel for the extents of the problem. In pattern B, control rods in rings six and seven (shaded in Fig. B.5) are half inserted and all others remain with drawn. In pattern C, the control assemblies in rings six and seven that were previously half inserted are fully inserted. Control rod withdrawal patterns are shown in Fig. B.6b.

Reference effective neutron multiplication factors and control rod worths are provided in Table B.11.

B.3.2 KNK

This benchmark is presented with solution by Takeda & Ikeda [Tak91]. It is based on a small FBR with hexagonal-z geometry and is a model of the KNK-II core. The original benchmark specification is for a transport solution, not a diffusion solution. The reactor has one-third rotational (*not* reflective) symmetry. For code packages without rotational boundary conditions, it will be necessary to simulate a full core. There are 15 assemblies across the core diameter with outer rings of reflectors and steel as shown in

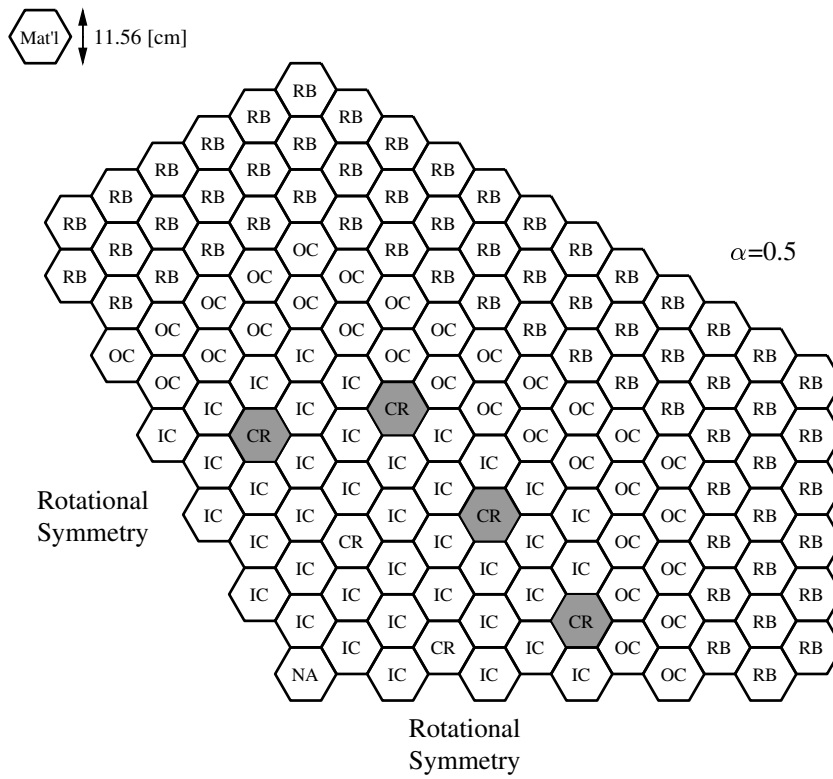


Figure B.5 MONJU Geometry.

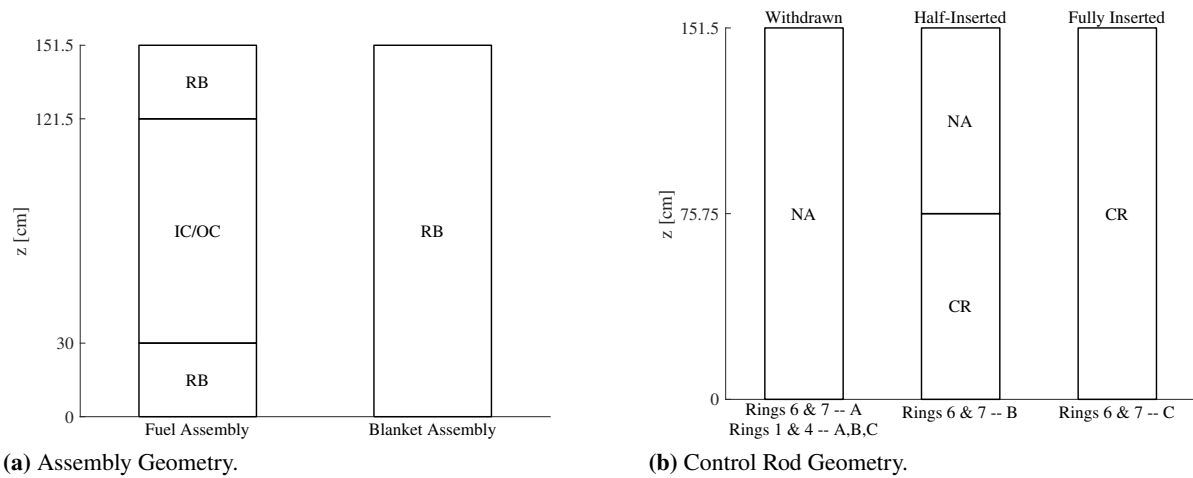


Figure B.6 MONJU Assembly Geometries.

Table B.12 MONJU Cross Sections.

	IC	OC	RB	CR	NA
D_1	2.540000E+00	2.547990E+00	2.173000E+00	2.500010E+00	4.805000E+00
D_2	1.724000E+00	1.725000E+00	1.439000E+00	1.681000E+00	3.262000E+00
D_3	1.264000E+00	1.269000E+00	1.026000E+00	1.269000E+00	2.431000E+00
Σ_{r1}	3.098650E-02	3.121380E-02	3.793080E-02	2.328030E-02	1.152508E-02
Σ_{r2}	9.490000E-03	9.875000E-03	1.184300E-02	1.272700E-02	3.648740E-03
Σ_{r3}	7.333000E-03	8.099000E-03	7.611000E-03	1.497000E-02	3.072000E-04
$\Sigma_{s1 \rightarrow 2}$	2.544000E-02	2.497000E-02	3.288000E-02	2.185000E-02	1.130000E-02
$\Sigma_{s1 \rightarrow 3}$	5.625000E-04	5.548000E-04	7.468000E-04	2.163000E-04	6.718000E-05
$\Sigma_{s2 \rightarrow 3}$	6.551000E-03	6.341000E-03	1.000000E-02	9.379000E-03	3.571000E-03
$\nu \Sigma_{f1}$	1.235000E-02	1.467000E-02	8.631000E-03		
$\nu \Sigma_{f2}$	5.225000E-03	6.955000E-03	5.995000E-04		
$\nu \Sigma_{f3}$	7.684000E-03	9.986000E-03	1.381000E-03		

Table B.13 MONJU Fission Spectrum.

Fission Spectrum [†]	
χ_1	0.78120
χ_2	0.20994
χ_3	0.00886

[†] Estimated values. Not part of original specification.

Table B.14 KNK Effective Neutron Multiplication Factors and Rod Worths.

Pattern	k_{ref}	Rod Worth [Δk]	Rod Difference [$\% \Delta k$]
A	1.0951		
B	1.9833	0.104	11.18
C	0.8799	0.223	21.52

Fig. B.7. Fuel assemblies are shown axially in Fig. B.8. Each assembly has a flat-to-flat measurement of 12.99 [cm]. Vacuum boundary condition ($\alpha = 0.5$) is applied on the core periphery and above and below the reactor. Four-group cross sections are provided in Table B.15 and Table B.16. The fission spectrum is provided in Table B.17.

The reactor is simulated with three different control rod configurations; patterns A, B, and C. In pattern A, all control rods are withdrawn. In pattern B, control rods are half inserted into the active fuel region. In pattern C, control rods are fully inserted into the active fuel region. All axial geometries are shown in Fig. B.9. The axial geometry for control rod withdrawal is shown in Fig. B.9.

Reference effective neutron multiplication factors and control rod worths are provided in Table B.14.

The KNK benchmark problem results are presented for the solution to the neutron transport equation. The reference presents Monte Carlo solutions and other solutions to the neutron transport equation and these converge to the same solution. For diffusion solutions to these simulations, agreement is expected to within some tolerance. However, comparison can also be difficult as the equations solved are entirely different.

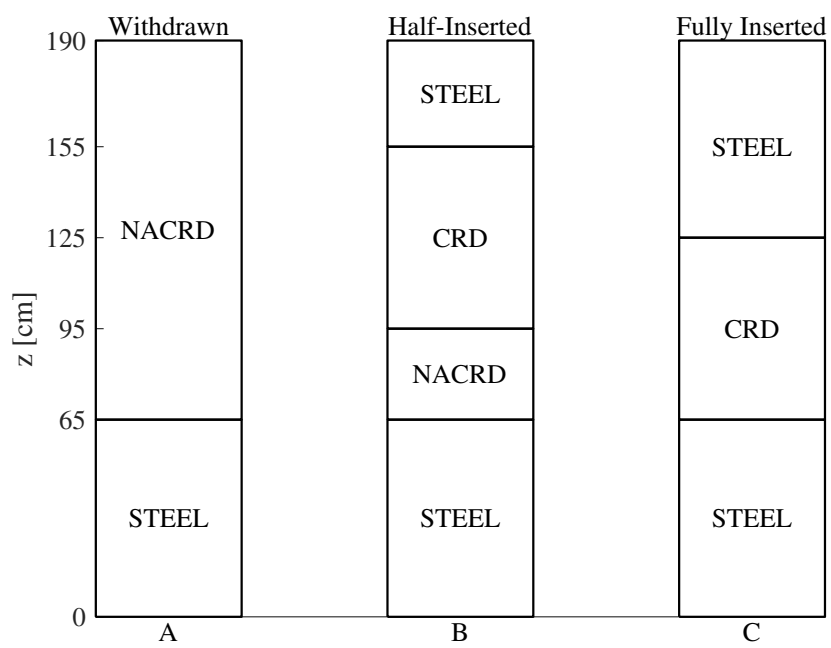


Figure B.9 KNK Control Rod Geometry.

Table B.15 KNK Cross Sections (Part A).

	STEEL	AXBL	AXRF	TEST	DRIV	DRMOD	REFL
D_1	3.388781E+00	2.373121E+00	2.507529E+00	2.676817E+00	2.377115E+00	2.356912E+00	2.091884E+00
D_2	2.466578E+00	1.477974E+00	1.867089E+00	1.658169E+00	1.460419E+00	1.358360E+00	1.540678E+00
D_3	1.483136E+00	1.019165E+00	1.177228E+00	1.163065E+00	1.023104E+00	8.369847E-01	9.559535E-01
D_4	1.177370E+00	9.768754E-01	7.212400E-01	9.039009E-01	7.968248E-01	7.645435E-01	5.339750E-01
Σ_{r1}	7.758804E-03	1.665700E-02	9.938010E-03	1.856200E-02	2.033901E-02	2.709099E-02	1.137701E-02
Σ_{r2}	4.558990E-03	8.274010E-03	5.436010E-03	1.165498E-02	1.303201E-02	3.338801E-02	5.945000E-03
Σ_{r3}	5.202000E-03	9.116980E-03	5.957010E-03	1.639202E-02	1.892099E-02	4.616198E-02	6.607000E-03
Σ_{r4}	2.410000E-03	9.943000E-03	3.569000E-03	4.981199E-02	5.742100E-02	6.511802E-02	4.942950E-03
$\Sigma_{s1 \rightarrow 1}$	9.060500E-02	1.238050E-01	1.229950E-01	1.059640E-01	1.198870E-01	1.143370E-01	1.479690E-01
$\Sigma_{s1 \rightarrow 2}$	7.423770E-03	1.454830E-02	9.412310E-03	1.127380E-02	1.307900E-02	2.096640E-02	1.066070E-02
$\Sigma_{s2 \rightarrow 2}$	1.305810E-01	2.172600E-01	1.730950E-01	1.893700E-01	2.152130E-01	2.120060E-01	2.104100E-01
$\Sigma_{s1 \rightarrow 3}$	1.181630E-04	1.702760E-04	1.937910E-04	1.461920E-04	1.599380E-04	1.391320E-03	2.499560E-04
$\Sigma_{s2 \rightarrow 3}$	4.352500E-03	6.788850E-03	5.098810E-03	3.648470E-03	4.001170E-03	2.672690E-02	5.467110E-03
$\Sigma_{s3 \rightarrow 3}$	2.195470E-01	3.179480E-01	2.771940E-01	2.702070E-01	3.068850E-01	3.520930E-01	3.420850E-01
$\Sigma_{s1 \rightarrow 4}$	8.258900E-07	9.370830E-07	1.393070E-06	9.621780E-07	1.071660E-06	6.102810E-05	1.825650E-06
$\Sigma_{s2 \rightarrow 4}$	3.416750E-07	6.047930E-06	7.050750E-07	1.068880E-06	1.827160E-06	1.081860E-03	1.001570E-06
$\Sigma_{s3 \rightarrow 4}$	4.645940E-03	4.387820E-03	5.096010E-03	1.804790E-03	1.673410E-03	3.290300E-02	5.368790E-03
$\Sigma_{s4 \rightarrow 4}$	2.807070E-01	3.312810E-01	4.585980E-01	3.189600E-01	3.609060E-01	3.708720E-01	6.193060E-01
$\nu \Sigma_{f1}$		2.961000E-03		1.790430E-02	1.598780E-02	1.016630E-02	
$\nu \Sigma_{f2}$		6.561700E-05		1.599610E-02	1.644460E-02	9.463600E-03	
$\nu \Sigma_{f3}$		1.146300E-04		2.408560E-02	2.714500E-02	1.873250E-02	
$\nu \Sigma_{f4}$		4.934825E-04		7.331050E-02	8.458075E-02	8.253350E-02	

Table B.16 KNK Cross Sections (Part B).

	RFMOD	KNKREF	NASTL	CRD	NACRD
D_1	2.395256E+00	2.198131E+00	3.453884E+00	2.396616E+00	4.581354E+00
D_2	1.349566E+00	2.341120E+00	3.376912E+00	1.461014E+00	3.326083E+00
D_3	7.367704E-01	2.018587E+00	2.483855E+00	1.045568E+00	2.074220E+00
D_4	6.215936E-01	4.141579E-01	8.077479E-01	5.313220E-01	2.199117E+00
Σ_{r1}	3.325300E-02	1.321701E-02	8.154706E-03	2.136299E-02	6.395295E-03
Σ_{r2}	6.217301E-02	4.880000E-03	3.460198E-03	3.345301E-02	4.094401E-03
Σ_{r3}	7.935300E-02	4.410000E-03	3.444000E-03	7.445399E-02	4.686990E-03
Σ_{r4}	2.415300E-02	5.912960E-03	3.037990E-03	3.125500E-01	1.207990E-03
$\Sigma_{s1 \rightarrow 1}$	1.059110E-01	1.384270E-01	8.835500E-02	1.177220E-01	6.636340E-02
$\Sigma_{s1 \rightarrow 2}$	2.964850E-02	1.239010E-02	7.734090E-03	1.260660E-02	6.233930E-03
$\Sigma_{s2 \rightarrow 2}$	1.848200E-01	1.375020E-01	9.524930E-02	1.946990E-01	9.612360E-02
$\Sigma_{s1 \rightarrow 3}$	3.065020E-03	3.669300E-04	1.947190E-04	1.333140E-04	7.021210E-05
$\Sigma_{s2 \rightarrow 3}$	5.917800E-02	4.419270E-03	3.225680E-03	4.322190E-03	4.013750E-03
$\Sigma_{s3 \rightarrow 3}$	3.730720E-01	1.607220E-01	1.307560E-01	2.443520E-01	1.560160E-01
$\Sigma_{s1 \rightarrow 4}$	1.416970E-04	1.690360E-06	8.896150E-07	1.088390E-06	4.163880E-07
$\Sigma_{s2 \rightarrow 4}$	2.692290E-03	1.632800E-06	7.984940E-07	1.854910E-07	1.269390E-07
$\Sigma_{s3 \rightarrow 4}$	7.813260E-02	3.330750E-03	2.904810E-03	3.687810E-04	4.491110E-03
$\Sigma_{s4 \rightarrow 4}$	5.121030E-01	7.989320E-01	4.096320E-01	3.148160E-01	1.503680E-01
$\nu \Sigma_{f1}$					
$\nu \Sigma_{f2}$					
$\nu \Sigma_{f3}$					
$\nu \Sigma_{f4}$					

Table B.17 KNK Fission Spectrum.

Fission Spectrum	
χ_1	0.908564
χ_2	0.087307
χ_3	0.004129
χ_4	0.000000