The Pennsylvania State University

The Graduate School

College of Engineering

COUPLED 3D NEUTRONICS AND THERMAL HYDRAULICS MODELING OF THE SAFARI-1 MATERIALS TESTING REACTOR

A Thesis in

Nuclear Engineering

by

Adam M. Rosenkrantz

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 2012

The thesis of Adam M. Rosenkrantz was reviewed and approved* by the following:

Kostadin N. Ivanov
Distinguished Professor of Nuclear Engineering
Thesis Advisor

Maria Avramova
Assistant Professor of Nuclear Engineering

Arthur T. Motta
Professor of Nuclear Engineering and Materials Science and Engineering
Chair of the Department of Nuclear Engineering

*Signatures are on file in the Graduate School.

ABSTRACT

The purpose of this study was to create a highly accurate model of the SAFARI-1 materials testing reactor, a facility mainly used for research and for the production of medical isotopes. The model was requested as part of the SAFARI-1 benchmarking project as a cooperative effort between the Pennsylvania State University (PSU) and the South African Nuclear Energy Corporation (NECSA). It was created using a coupling of state of the art nuclear reactor simulation tools, consisting of a neutronics code and a thermal hydraulics code. The reactor was subdivided into 2520 nodes for analysis. The neutronics tool used was the PSU code NEM, and the results of this component were validated using the NECSA neutronics code OSCAR, which was in turn validated experimentally. On the average, the multiplication factors of the neutronics models agreed to within 0.005%, and the radial cell powers agreed to within 0.07%.

The thermal hydraulics tool used was the PSU version of COBRA-TF (CTF), and the transient results of this component were validated against another thermal hydraulics code, RELAP, which was in turn validated experimentally. Although RELAP results were only released to a limited extent (averaged by assembly type), they fell within the range of values for the corresponding assemblies in the comprehensive CTF solution. The outcome of the study was the intended coupled neutronics/thermal hydraulics model of the SAFARI-1 reactor. The results found using the coupled NEM/CTF model were compared to accepted reactor theory and experimental results, and showed improvement over the stand-alone models. Though the present version of the coupling uses steady-state hot full power conditions, future work may be carried out for advanced analysis of the tank-in-pool type reactor, such as accident analysis or fuel management optimization.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
4D	Four Dimensional
COBRA-TF	Coolant Boiling in Rod Arrays-Two Fluid
CTF	The RDFMG Version of COBRA-TF
CyberSTAR	A Scalable Terascale Advanced Resource for Discovery through Computing
ECC	Engineered Cementitious Composite
HEU	Highly Enriched Uranium
INL	Idaho National Laboratories
I/PIE	Irradiation/Post Irradiation Examination
k_{eff}	Multiplication Factor
LEU	Low Enriched Uranium
LWR	Light Water Reactor
MCNP5	A General Monte Carlo N-Particle Transport Code – Version 5
MGRAC	Multi-Group Reactor Analysis Code
MTR	Materials Testing Reactor
NECSA	South African Nuclear Energy Corporation
NEM	Nodal Expansion Method
OSCAR	Overall System for Calculating Reactors
PARCS	Purdue Advanced Reactor Core Simulator
PCM	Per Cent Mille (or $\times 10^{-5}$)
PSU	The Pennsylvania State University
RDFMG	Reactor Dynamics and Fuel Management Group
RELAP	Reactor Excursion and Leak Analysis Probability
RM	Response Matrix
RRT	Radiation and Reactor Theory Group
SAFARI-1	South African Fundamental Atomic Research Installation

ACKNOWLEDGEMENTS

I would first like to acknowledge my advisor Dr. Kostadin Ivanov for his guidance and assistance, and Dr. Maria Avramova for her help and networking, which contributed to the thermal hydraulics of this thesis.

I would also like to thank David Colameco, Rian Prinsloo, and Damon Roberts for their help with the neutronics of model, as well as for their technical support.

This work was supported in part through instrumentation funded by the National Science Foundation through grant OCI–0821527.

Finally, I would like to thank my family for their continuous love and support as I work and travel in pursuit of my educational and professional goals.

CHAPTER 1. Introduction

The purpose of the performed nuclear reactor core neutronics, thermal hydraulics, and coupled modeling presented in this thesis is to contribute to the South African Fundamental Atomic Research Installation (SAFARI-1) benchmarking project. [1] The benchmark specification calls for groups of researchers around the world to contribute to the project by using computational analysis to simulate the reactor during various phases of its power cycle. The contribution described here focuses on the steady-state operating conditions of the reactor.

To perfectly model a nuclear reactor, an immensely time-intensive model (presumably of Monte Carlo type) would be needed, along with a tremendous amount of experimental data, implementation of physical processes which are not yet well explained, and an extensively multidisciplinary approach. Such a perfect model is not yet within the reach of modern scientific advancement. It is therefore necessary to utilize simplified modeling software, which makes use of engineering tools including experimentally derived correlations and iterative convergence methods, in order to achieve promising results through carefully selected assumptions and other simplifications. Furthermore, informed selection of the goals of the model increases overall efficiency when implemented correctly.

For the case of the SAFARI-1 benchmarking project, the emphasis is on the nuclear phenomena of the neutrons and atoms within the core, as well as on the fluid flow and heat transfer of the system. These were requested by the South African Nuclear Energy Corporation (NECSA), and led to the decision to a coupling of methods. The two methods coupled in this case were a neutronics simulator and a thermal hydraulics

simulator. To ensure that both coupling components were valid, they were created and tested separately, and compared to similar trials and other codes implemented by NECSA's Radiation and Reactor Theory Group (RRT). Since the thermal hydraulics code has already been verified by NECSA, there is a greater interest in the novel neutronics code used in the coupling.

The general approach of the project was to model the reactor using state of the art neutronics and thermal hydraulics computational software. However, before modeling could begin, the material composition and geometric layout of the SAFARI-1 materials testing reactor (MTR) had to be thoroughly defined and understood. In Chapter 2, these defining components of the reactor are comprehensively presented. First, an introductory explanation of the history and use of the reactor is described. Next, an overview of the core is given, including the core assembly layout, surrounding reflector description, and boundary conditions. Following the core overview, in depth analyses of the fuel assemblies, control and follower assemblies, and reflector assemblies are made. Finally, other important material and geometric details are given.

The reference solution against which this project's original neutronics model was compared is the Overall System for Calculation of Reactors (OSCAR). [2] Chapter 3 contains an overview of OSCAR for the purpose of justifying its validity as a reference, as well as an interpretation of the OSCAR reference solution for its neutronics model of the SAFARI-1 reactor. The Pennsylvania State University (PSU) code Nodal Expansion Method (NEM) was used for the original neutronics model. [3] This is a deterministic code that can be used for three dimensional (3D) nuclear reactor simulations. In Chapter 3, an overview of NEM is provided, followed by the details of the reactor neutronics

modeling processes used for this project, and then a presentation and interpretation of the solution obtained using NEM. Chapter 3 concludes with a comparison of the OSCAR and NEM results, including an assessment of the validity of the NEM results.

The reference solution against which this project's thermal hydraulics model was compared is the Reactor Excursion and Leak Analysis Program (RELAP). [4] Chapter 4 contains a brief overview of RELAP, including the SAFARI-1 model results. These results are shown in order to provide a reference which was developed solely by parties independent of this study. The release of the RELAP reference solution was limited, but various specific results were able to be compared to the original results from this study, and these are presented.

Chapter 4 goes on to describe basic details of the thermal hydraulics code used for the coupled code, which is Coolant Boiling in Rod Arrays-Two Fluid (COBRA-TF). The original source code has been modified by the Reactor Dynamics and Fuel Management Group (RDFMG) at PSU to create a new version called CTF, which has been previously used to contribute to several other benchmarking projects. [5] A CTF thermal hydraulics model was created in cooperation between PSU and NECSA staff in order to eventually run coupled neutronics/thermal hydraulics simulations modeling SAFARI-1, but these stand-alone CTF results were not released and the NECSA coupled code is presently incomplete. Thus, the CTF model was carefully scrutinized through stand-alone trials, the results of which are available here, before being implemented in the original coupled code created during the course of this study. Chapter 4 concludes with a comparison of the available RELAP results to the corresponding CTF results.

Chapter 5 presents the main goal of the study, which is the coupling of NEM and CTF. The coupled code that was created works exclusively with the SAFARI-1 model at present, but could be modified for other models or for general use in the future. The neutronics solution found by NEM provides power distributions for the fuel and control and follower assemblies to CTF. CTF provides fuel temperatures, moderator temperatures, and moderator densities to a polynomial interpreter. The interpreter uses a cross section library created with core-averaged state parameters to fabricate a new library based on the parameters from CTF. This library is used by NEM to create the next neutronics solution. This process is repeated until convergence is reached by the system, which is determined by changes in k_{eff} of less than 10^{-6}, and relative changes in the nodal powers and state parameters, namely fuel temperatures, moderator temperatures, and moderator densities, of less than 10^{-4}, for three consecutive iterations.

The coupled code is an offline coupling, meaning it requires some human interaction between iterations. Online couplings, those which require no human interaction between iterations, are usually preferred, but one was not feasible due to operating system requirements by the versions of NEM and CTF used for this study, which currently compile and run properly only on Linux and Windows operating systems respectively. Chapter 5 also discusses the workings of the 4 additional executable files created for this study, and the procedure involved for running the coupled code. The results of the coupled code are presented and compared to the stand-alone models and to well-established nuclear reactor theory and experiments.

Chapter 6 provides the conclusions of the project as a whole and suggestions for future work that could be done to improve, further verify, or expand upon the work completed in the course of this study.

Following the body of the report are six appendices. Appendix A is the 6-group input of the stand-alone NEM neutronics model of SAFARI-1 under the conditions examined in this study. Appendix B is the stand-alone CTF model used for the various 2-group and 6-group models. Appendix C is MakeCTF.f90, the source code of the executable which takes the power distributions from NEM outputs and inserts them into a CTF input deck. Appendix D is TakeCTF.f90, the source code of the executable which takes the state parameters from NEM outputs and inserts them into a separate text file in tabular format. Appendix E is generatelib.f90, the source code of the main file for the executable which takes the state parameters from the table created by TakeCTF.exe and uses them to create a cross section library using a polynomial interpreter. Appendix F is MakeNEM.f90, the source code of the executable which takes the polynomial cross section library and inserts them into an NEM input deck.

CHAPTER 2. SAFARI-1 Reactor Core Description

Brief History and Use

The SAFARI-1 nuclear reactor is a 20 MW tank-in-pool type research reactor located in Pelindaba, South Africa. Like all nuclear fission reactors, the primary purpose of the reactor is to free neutrons from atoms in order to utilize the neutrons' accompanying energy to induce fission in the reactor's fuel. This in turn frees more energy rich neutrons, causing the process to continue in a cyclic manner. In an MTR like SAFARI-1, the reactor core is not maintained for power production, but rather to further research and development efforts in the field of nuclear engineering. Advancements in the field of nuclear reactor science lead to better simulations and production of new reactors, as well as improved safety and efficiency of existing reactors. [6]

In addition to the research and development aspects of the SAFARI-1 reactor, it has historically been used for a plethora of other purposes, both academic and commercial in nature. These services include irradiation of materials to test their properties, activation of materials for radio-analysis, irradiation of irradiated materials which can be further processed into useful products (e.g. medical isotopes, silicon doping), radiographic and diffraction analysis, direct medical use (e.g. treatment of cancers via Boron Neutron Capture Therapy), and irradiation/post irradiation examination (I/PIE) both initial and production type of work. [6]

Recently, the reactor became instrumental in continuing the creation and distribution of radiopharmaceuticals, such as technetium-99m, to the international medical community. This followed a decrease in production from the National Research Universal Reactor of Canada, due to unexpected shutdowns for repairs. [7]

The reactor is owned and operated by NECSA. It has been in operation since being commissioned in 1965, and has been used for both local and international scientific endeavors, ranging from simulations to chiefly experimental techniques. Though the reactor has used highly enriched uranium (HEU) fuel for most of its time in operation, it was announced in 2005 that it would be switching to low enriched uranium (LEU) fuel in the near future. [8] However, it is still currently using HEU as of the start of the benchmarking project, which uses a fuel cycle that closely resembles the fuel cycle of SAFARI-1 in 2007. [9]

Core Overview

The core of the SAFARI-1 reactor consists of a 9 x 8 radial array comprised of a combination of various types of assemblies and material components. The types of assemblies present in the core are fuel assemblies, control and follower assemblies (assemblies which are comprised of a control segment and a follower segment), and reflector assemblies. The assemblies designated as reflector assemblies include aluminum water boxes, solid aluminum, hollow aluminum, solid beryllium, hollow beryllium, and solid lead assemblies. Each assembly has (x, y) Cartesian dimensions of 7.71 cm x 8.1 cm on a two dimensional (2D) plane. [1]

The 8 x 9 array is presented in Figure 1.

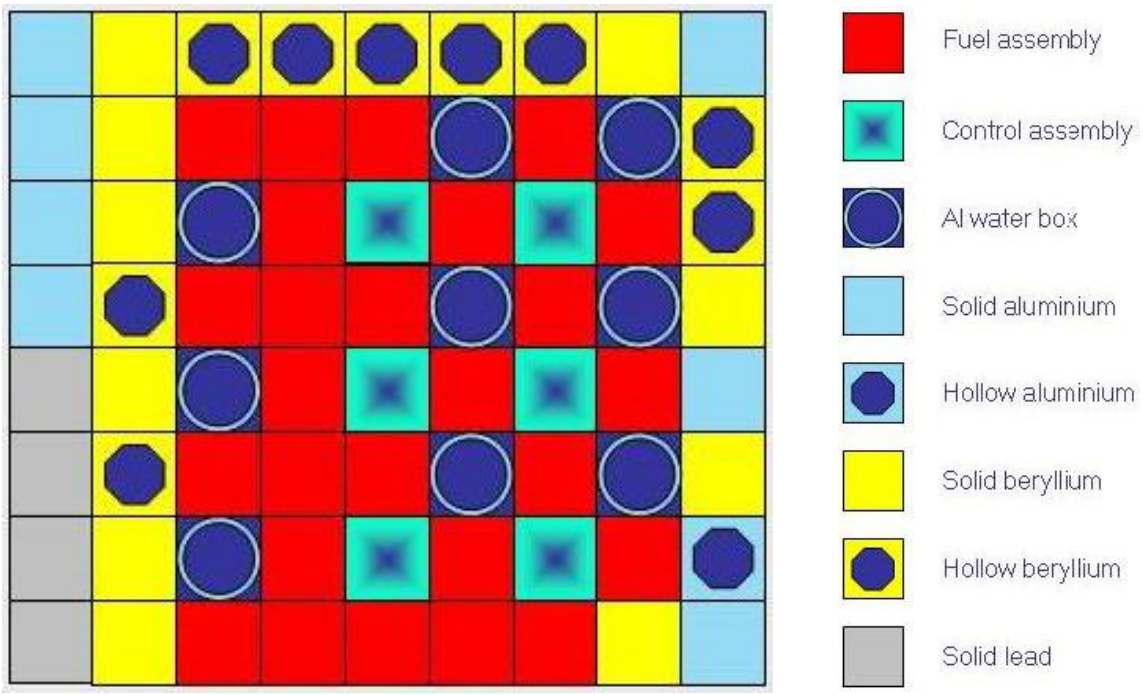

Figure 1: Assembly Layout of Reactor Core. This is an overhead view, and the top of the figure is facing south. [1]

According to the SAFARI-1 benchmark specification, the reactor core is surrounded by several layers of material. The 9 x 8 core array takes up a 69.39 cm x 64.8 cm space in the approximate center of the layout. Immediately surrounding the array is a layer of solid aluminum, known as the aluminum core box. This layer of aluminum has a thickness of 2.5 cm on the west, east, and south sides of the array, and 3.5 cm on the north side of the array. Surrounding the aluminum core box is a layer of water, which has a thickness of 20 cm on all sides of the aluminum. The boundary conditions for the benchmark dictate that a vacuum is present around the water layer. This is in place to simulate irreversible neutron leakage from the core through the edge of the water volume. The vacuum layer has a thickness of 20 cm on all sides of the water. [1]

The core and reflector layout is presented in Figure 2.

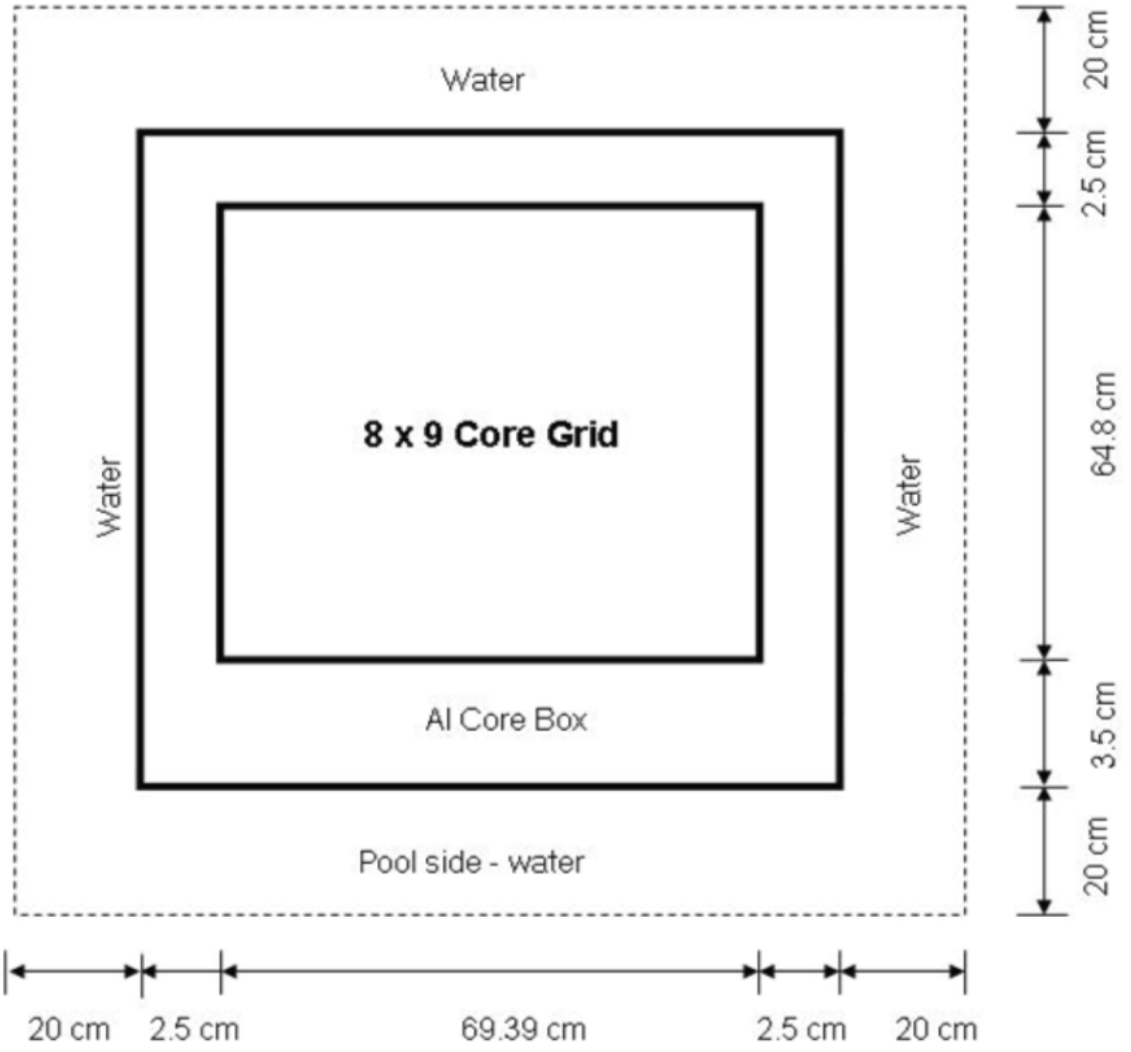

Figure 2: Radial View of SAFARI-1 Core and Reflector. This is an overhead view, and the top of the figure is facing south. [1]

Fuel Assemblies

The fuel of the SAFARI-1 reactor is uranium-aluminum alloy. The uranium is considered to be HEU, and is 90% enriched (uranium-235) by weight. The total uranium content in the alloy is 29% by weight. The density of the uranium in the alloy is 0.925 g/cm^3. The density of the aluminum in the alloy is 2.265 g/cm^3. In total, there are 26 fuel assemblies within the reactor core. Each assembly has 19 fuel plates, which are stacked horizontally next to one another. Each plate has a total of 15.79 ± 0.30 g of uranium-235, and each assembly has a total of 300 ± 4.0 g uranium-235. The fuel meat thickness is 0.0507 cm. The fuel cladding thickness is 0.0384 cm. [1]

To hold the fuel in place, end adaptors are utilized, though their exact geometry is not necessary for the level of computational analysis completed in this study.

The details of the geometry of the fuel assembly and affixing mechanisms can be seen in Figures 3 and 4.

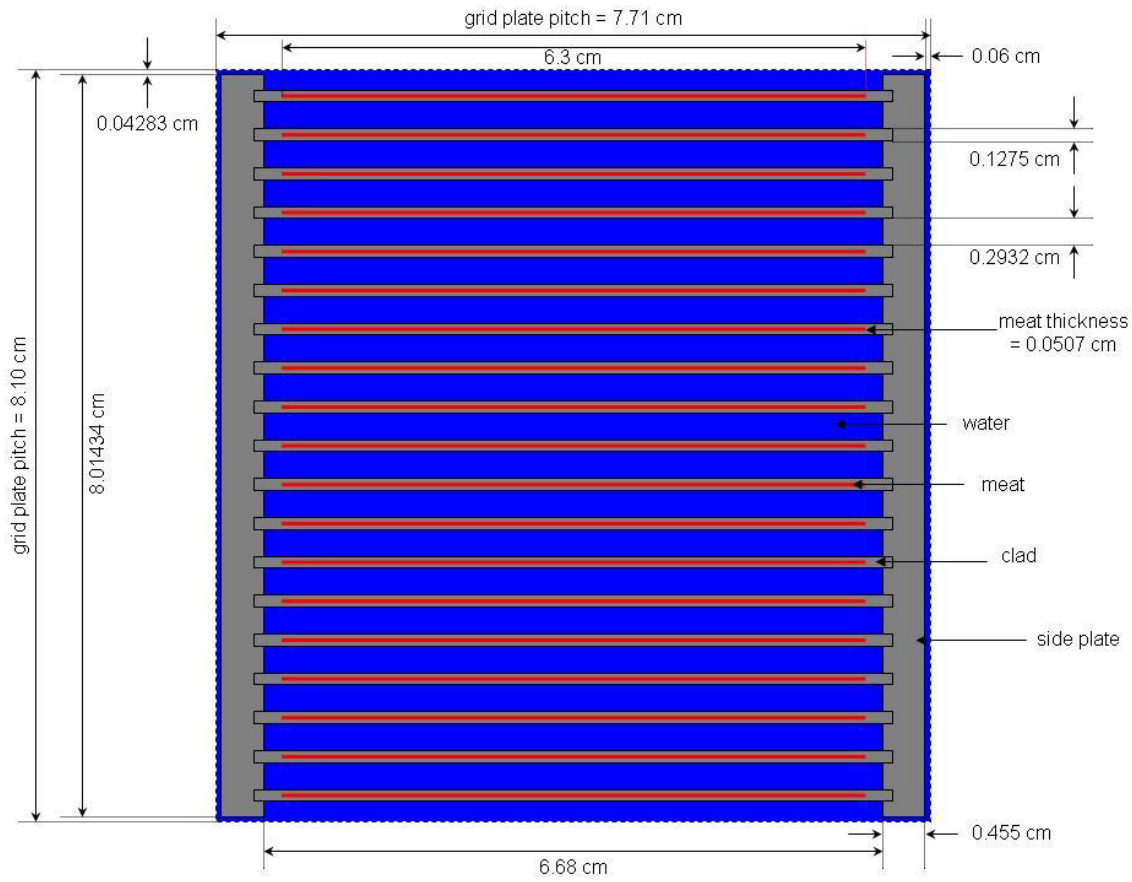

Figure 3: Radial View of a Fuel Assembly. [1]

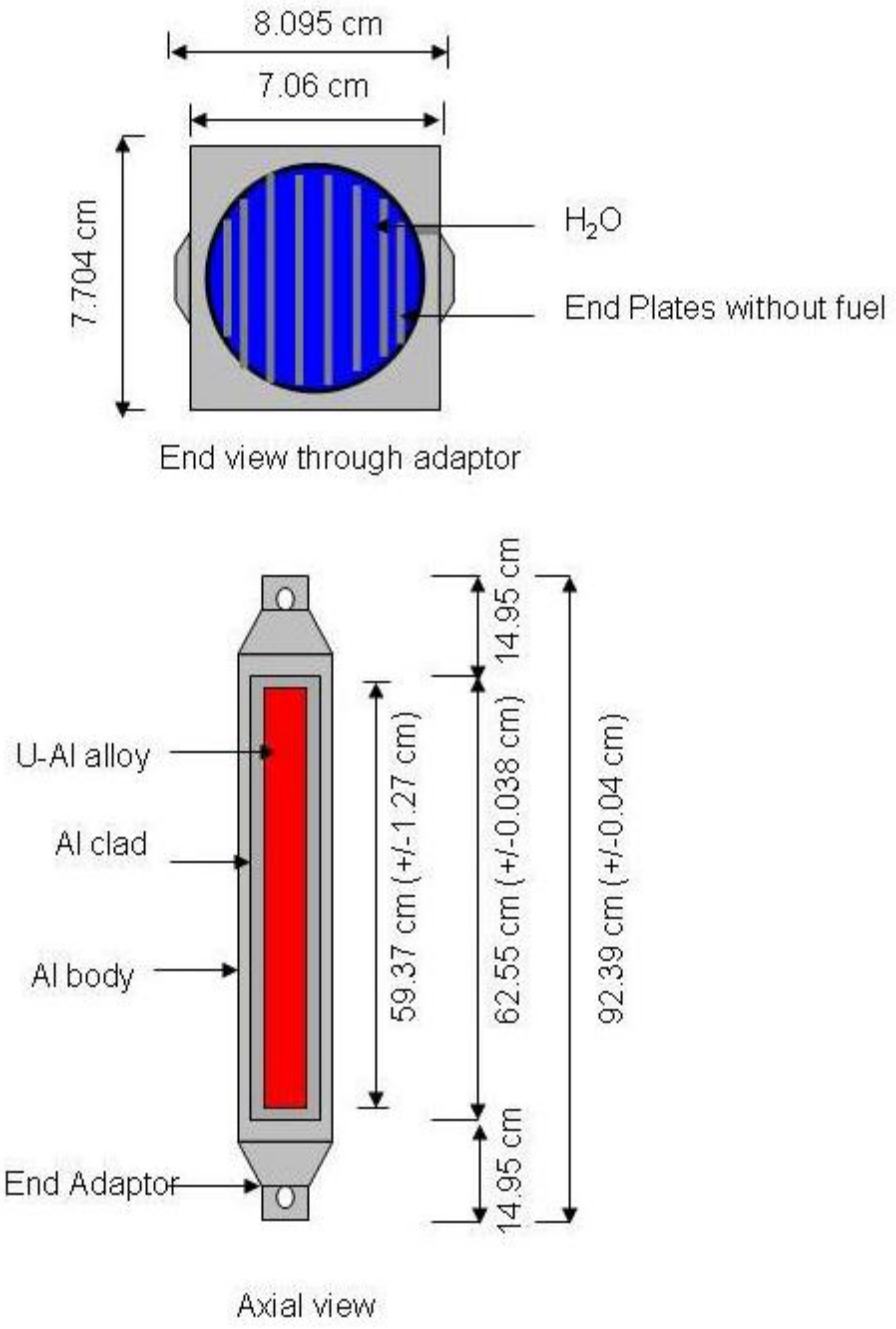

Figure 4: Radial View of Fuel Assembly Adaptor Mechanism (above) and Axial View of Entire Fuel Assembly (below). [1]

Control and Follower Assemblies

The primary control mechanism of the SAFARI-1 reactor consists of the control rod and fuel follower assemblies. The control rod is a cadmium absorber, and the fuel follower is made up of 15 fuel plates, which are stacked horizontally next to one another. Six such assemblies exist in the reactor core. The fuel of each fuel follower is uranium-aluminum alloy. The uranium is considered to be HEU, and is 90% enriched (uranium-235) by weight. Likewise, the total uranium content in the alloy is 29% by weight. The density of the uranium in the alloy is 0.826 g/cm^3. The density of the aluminum in the alloy is 2.0233 g/cm^3. Each plate has a total of 13.50 ± 0.25 g of uranium-235, and each assembly has a total of 202.5 ± 3.0 g uranium-235. The fuel meat thickness is 0.0507 cm. The fuel cladding thickness is 0.0384 cm.

During use of the reactor, the control rod and fuel follower assembly positions are described in distance from the fully inserted depth, 15.13 cm lower than the bottom of the core. The total range of axial movement of a control rod and fuel follower assembly is 74.5 cm. [1]

There is no physical rod guidance system, the adherence of the movement of the assemblies to a prescribed trajectory is instead maintained by the assembly geometry itself, as presented in Figures 5, 6, 7, and 8.

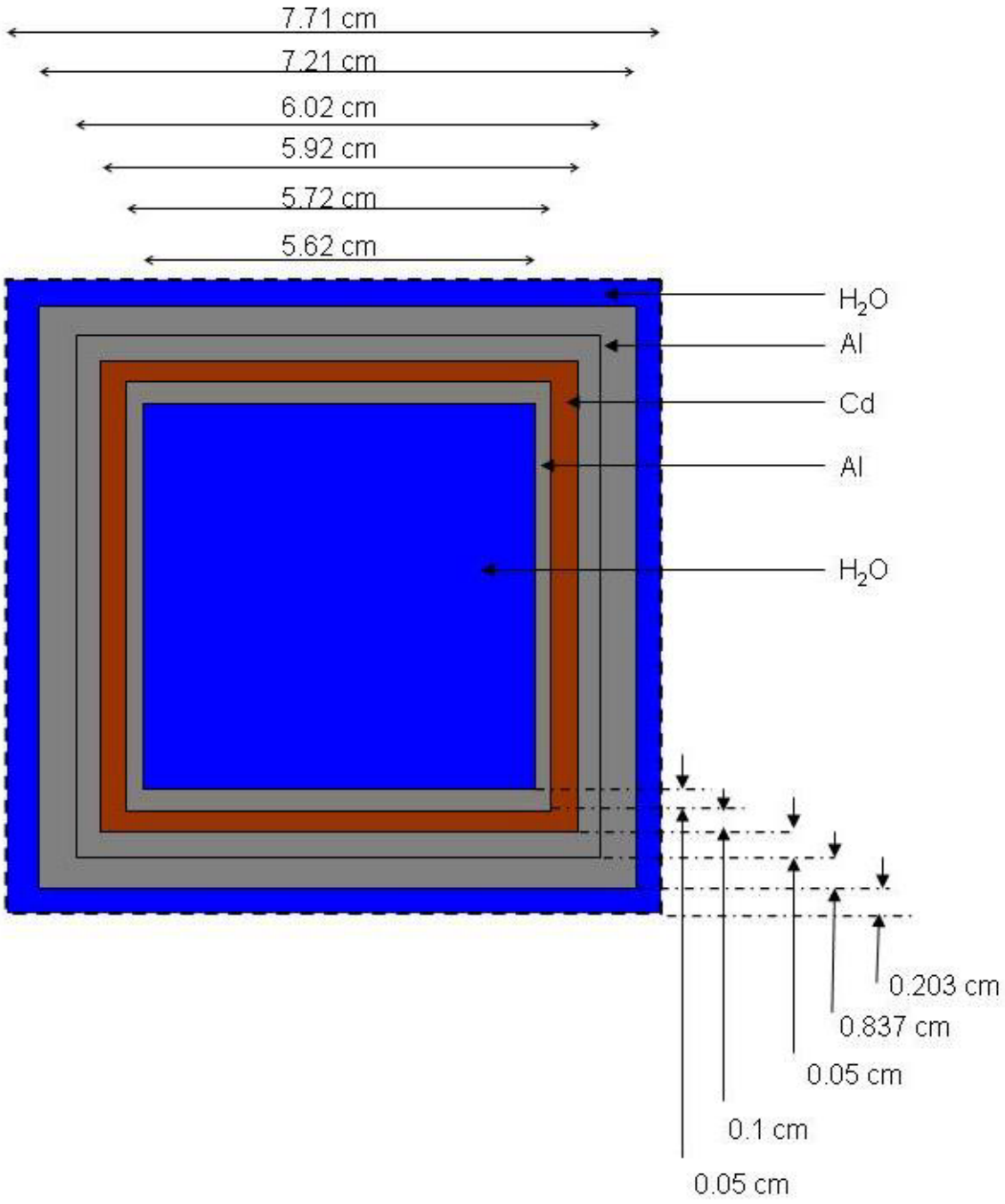

Figure 5: Radial View of the Cadmium Absorber Control Rod. [1]

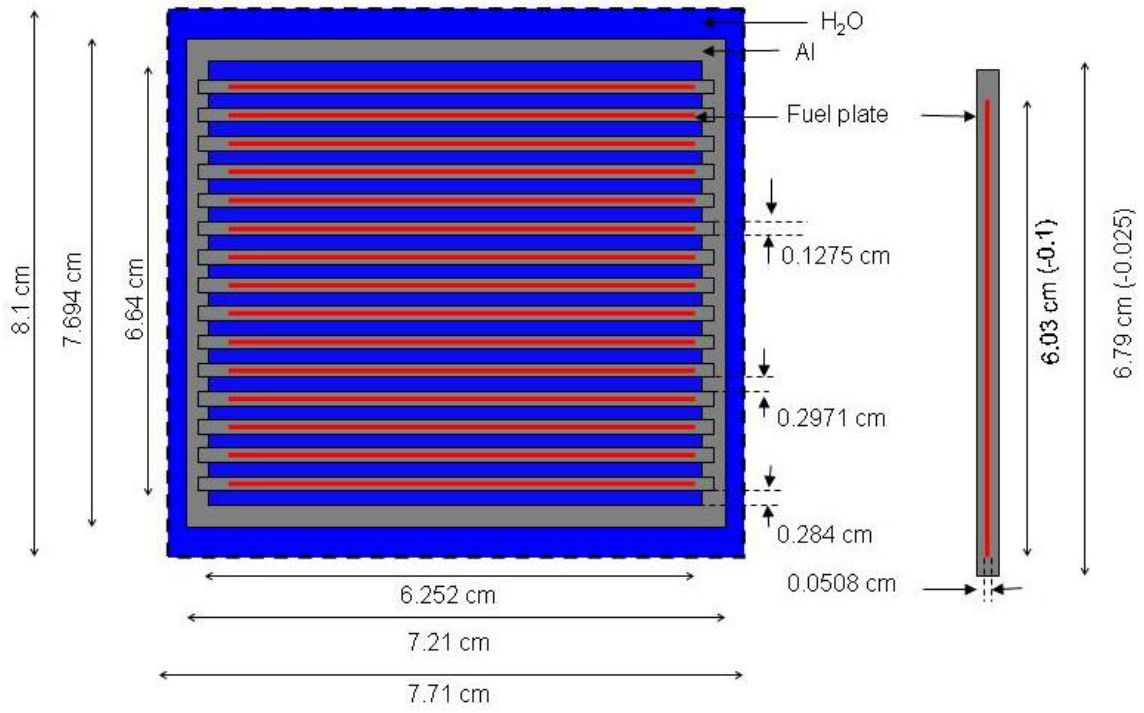

Figure 6: Radial View of the Fuel Follower. [1]

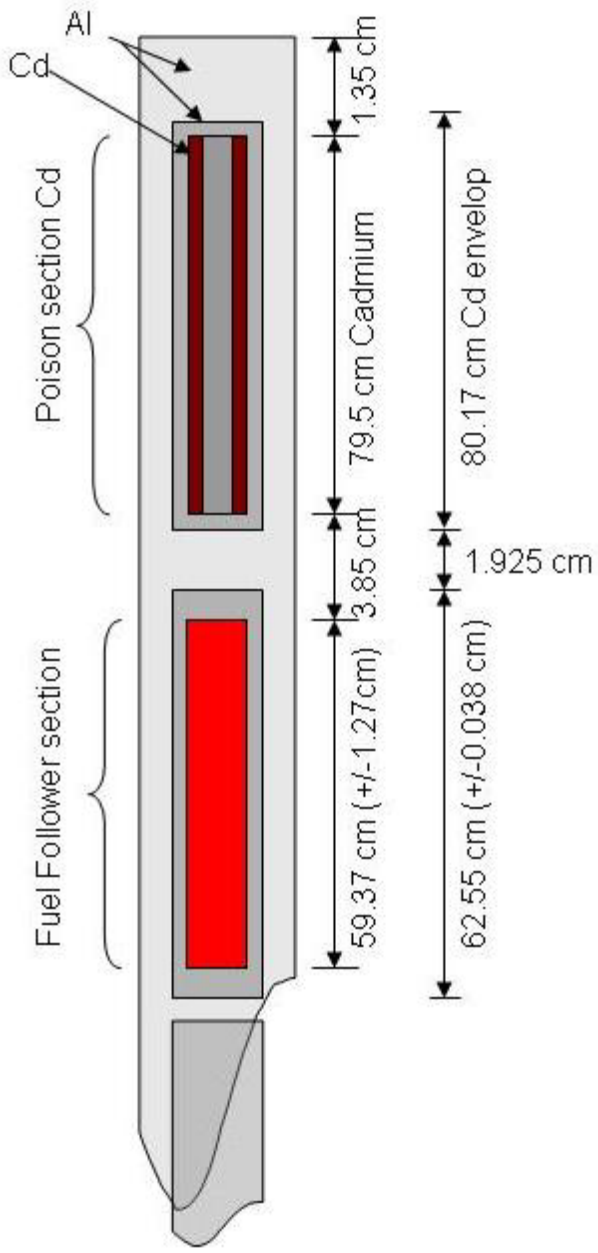

Figure 7: Axial View of the Control Rod (above) and Fuel Follower (below). [1]

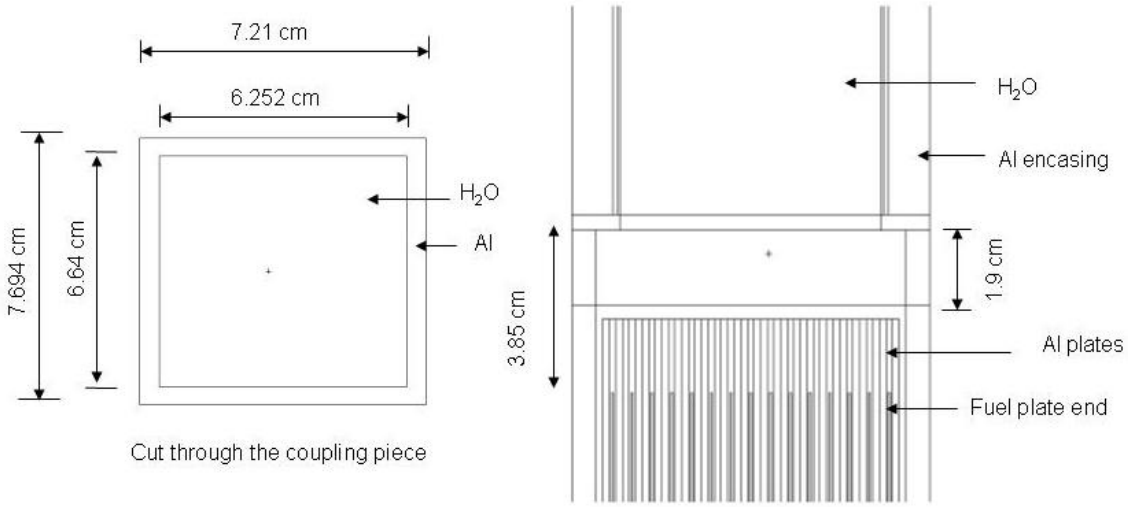

Figure 8: Radial (left) and Axial (right) View of Control Rod-Fuel Follower Coupling Unit. [1]

Reflector Assemblies

The SAFARI-1 reactor has six distinct types of reflector assemblies. These are the aluminum water box type assembly, the hollow aluminum type assembly, the solid aluminum type assembly, the hollow beryllium type assembly, the solid beryllium type assembly, and the solid lead assembly. The terms "hollow" and "solid" here refer to a larger or smaller cross sectional flow area of water through the assembly. The density of the aluminum in the reflector assemblies is uniform for aluminum material assemblies and the aluminum cladding used in the aluminum and other reflector assemblies, and is 2.7 g/cm^3. The density of the beryllium in the beryllium material assemblies is 1.85 g/cm^3. The density of the lead in the lead material assemblies is 11.4 g/cm^3.

Most of the reflector assemblies also facilitate a constant flow of water, which may be present in either liquid or vaporous molecular form, or as a combination of gaseous hydrogen and oxygen. The density of liquid water in the reflector assemblies is 0.9923 g/cm^3 at 1 bar and 45° C. The density of gaseous hydrogen in the reflector assemblies is 0.066294 g/cm^3. The density of gaseous oxygen in the reflector assemblies is 0.033197 g/cm^3. [1]

The specific geometries of each of the six types of reflector assembly, as well as the reflector assembly casing, are presented below in Figures 9, 10, 11, 12, 13, 14, and 15.

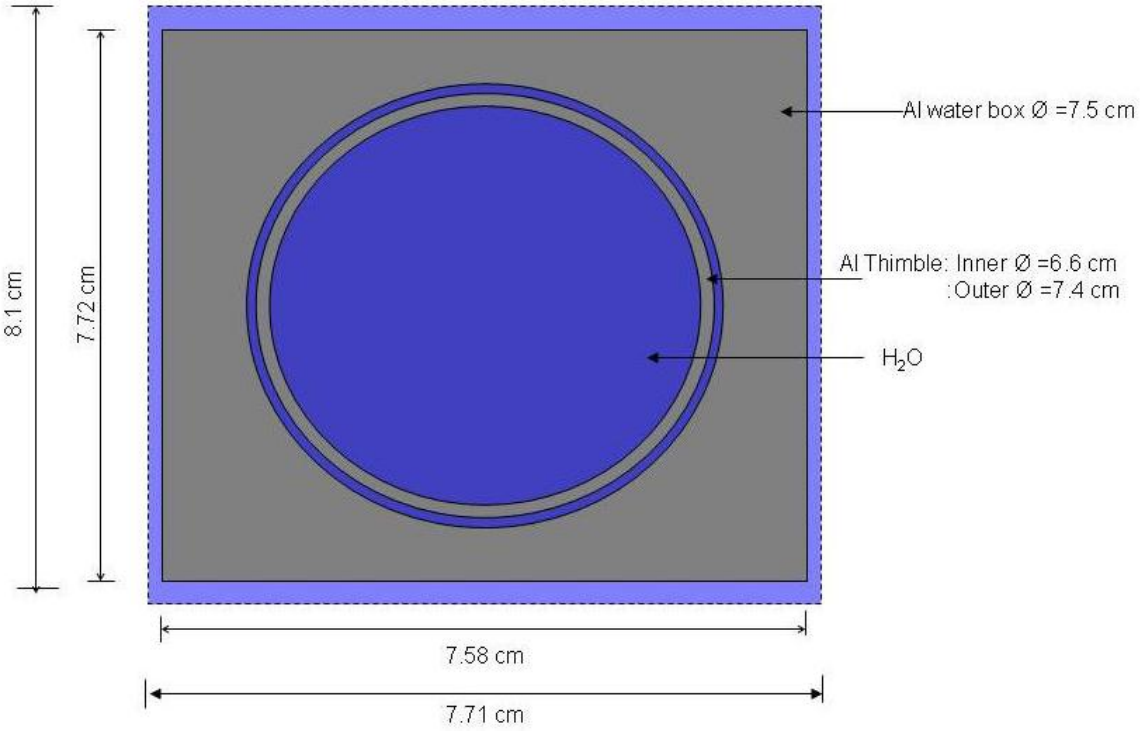

Figure 9: Radial View of the Aluminum Water Box Type Reflector Assembly. [1]

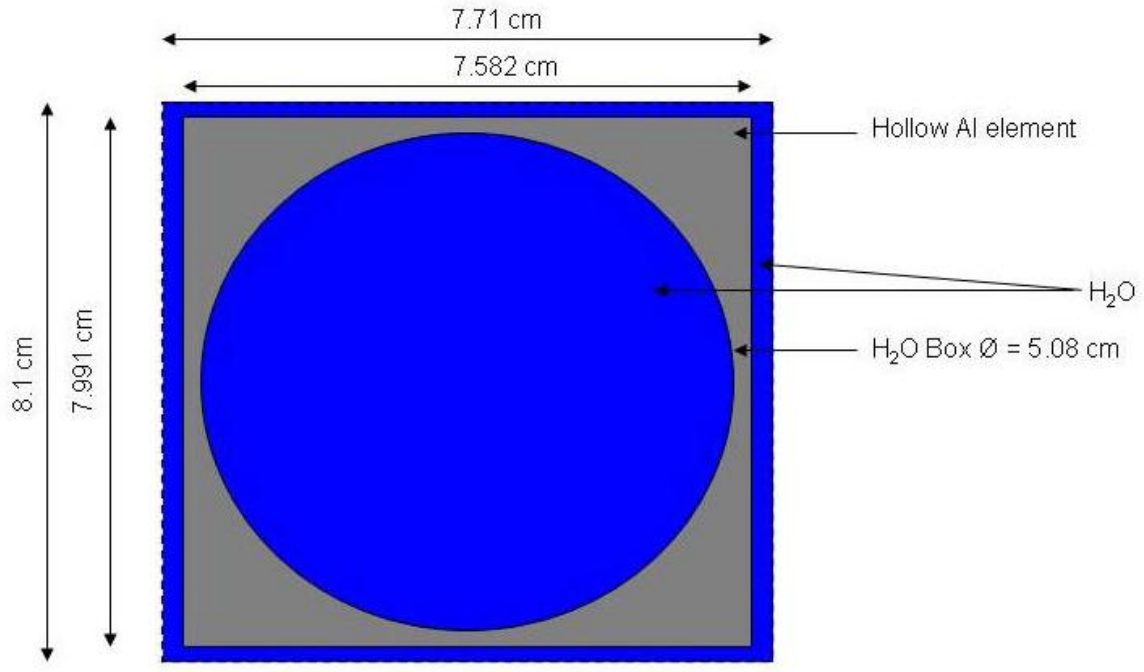

Figure 10: Radial View of the Hollow Aluminum Type Reflector Assembly. [1]

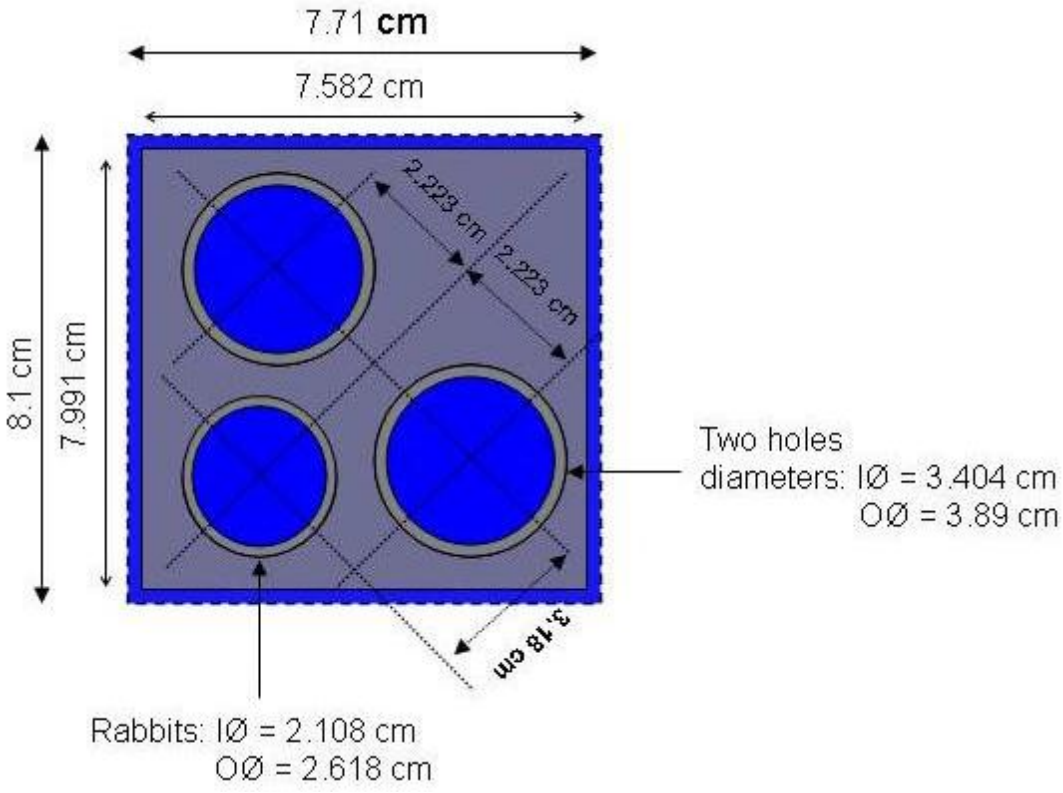

Figure 11: Radial View of the Solid Aluminum Type Reflector Assembly. [1]

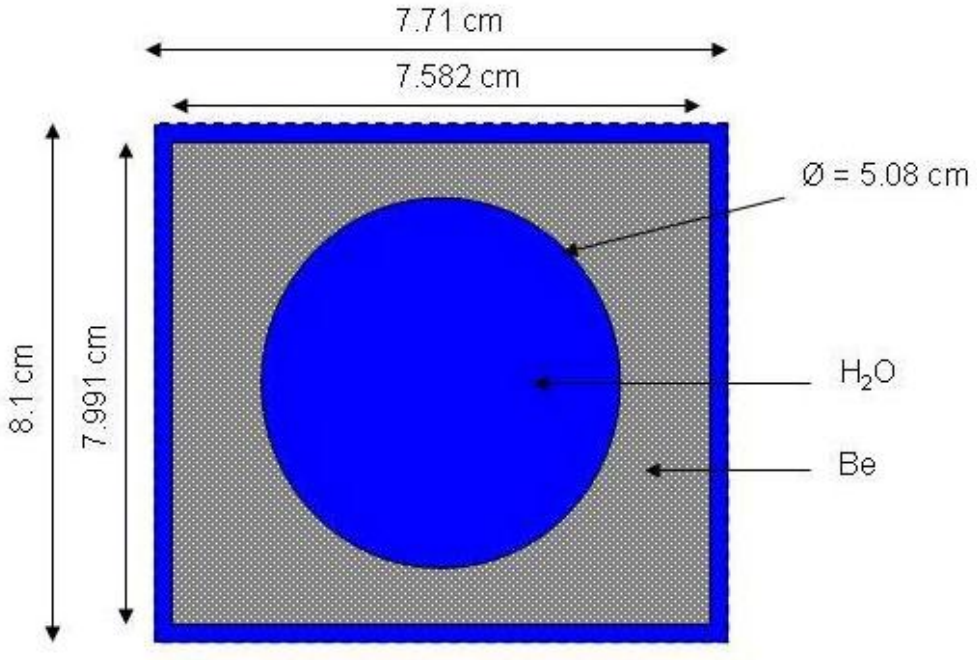

Figure 12: Radial View of the Hollow Beryllium Type Reflector Assembly. [1]

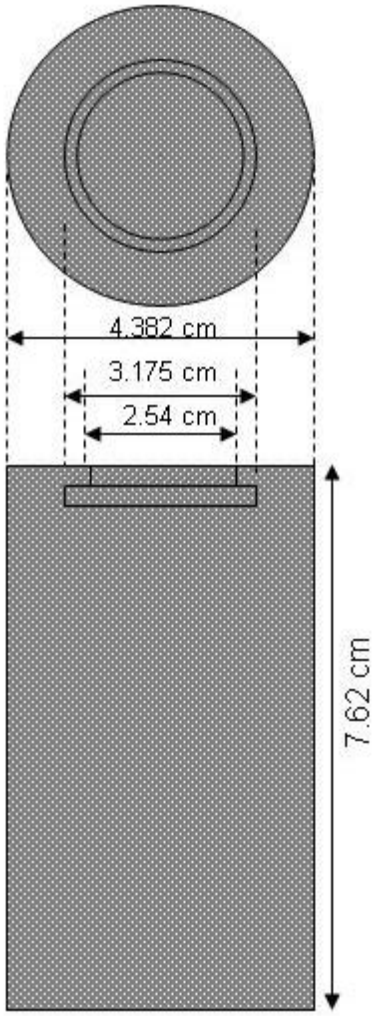

Figure 13: Radial (above) and Axial (below) View of the Beryllium Plug. [1] The beryllium plug is inserted into a hollow beryllium type reflector assembly to form the solid beryllium type reflector assembly.

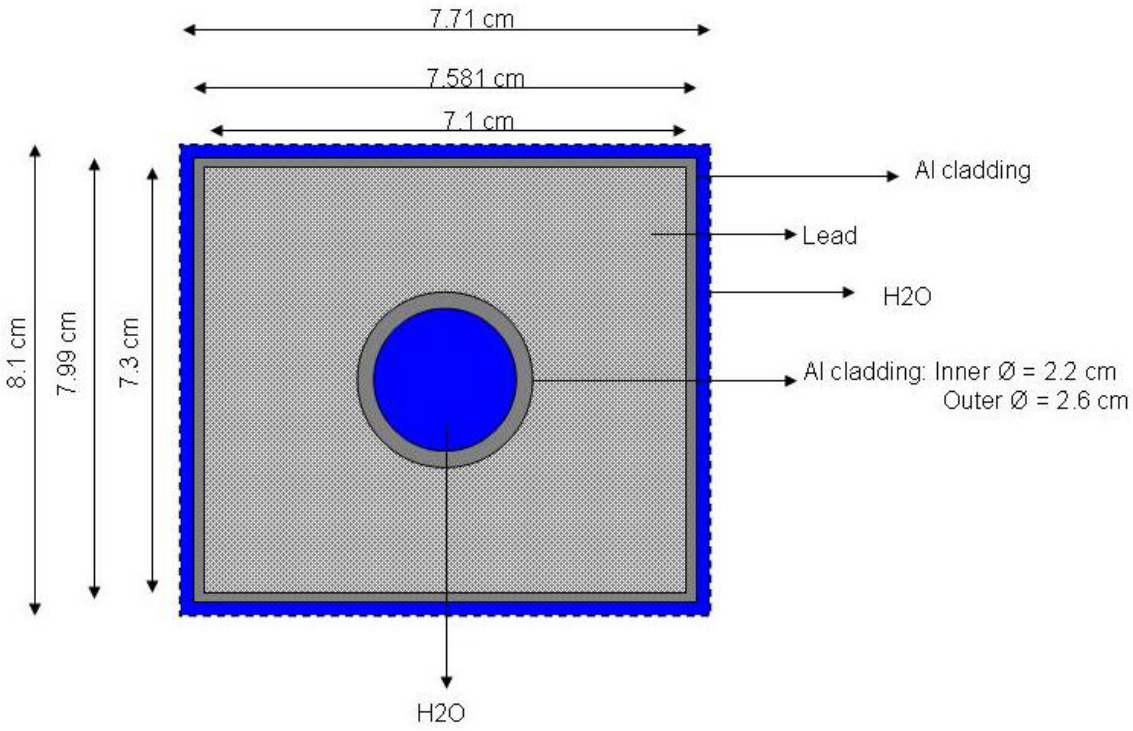

Figure 14: Radial View of the Solid Lead Type Reflector Assembly. [1]

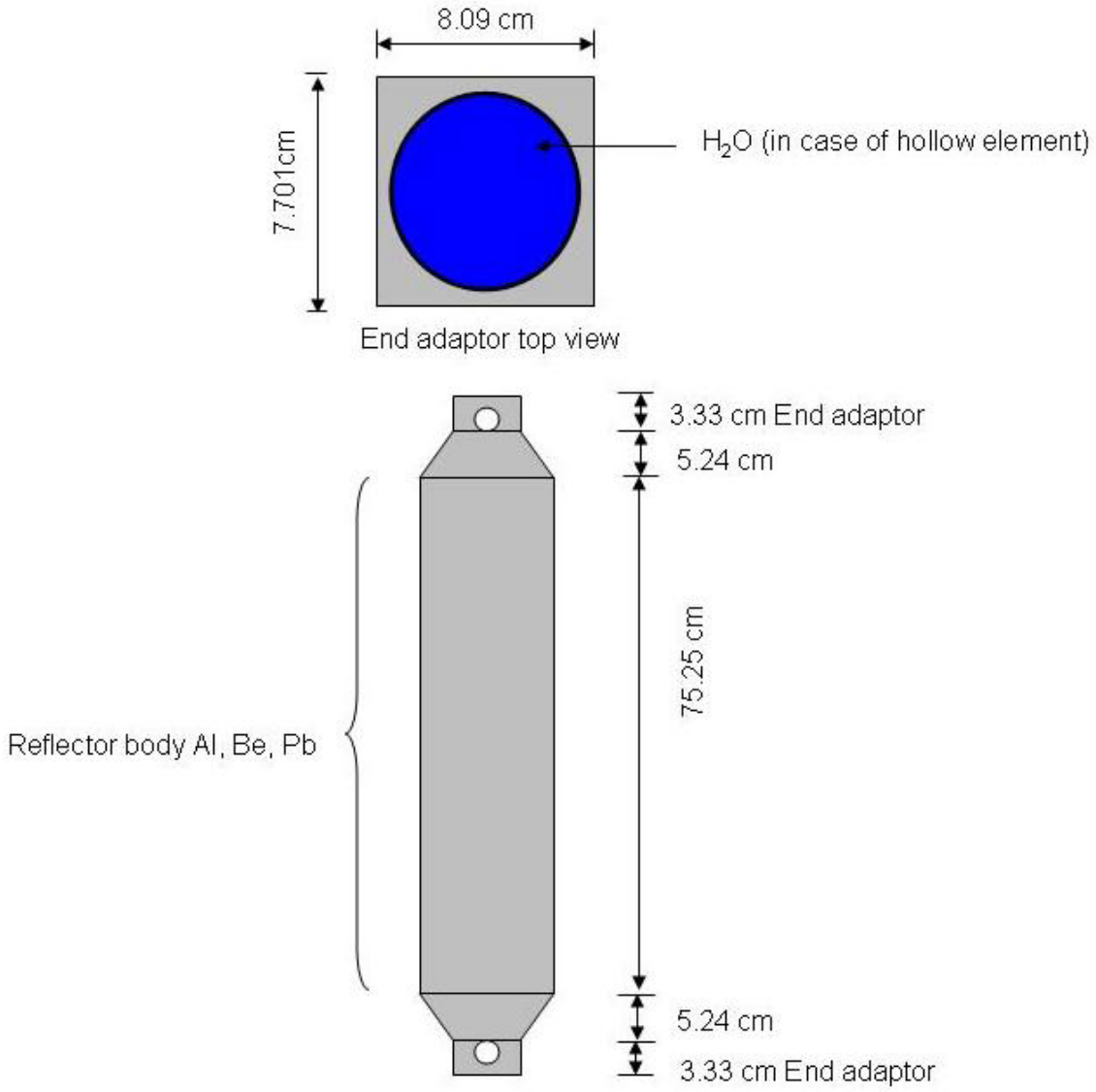

Figure 15: Radial (above) and Axial (below) View of the Reflector Assembly Casing. [1]

All of the reflectors have the same aluminum casing, including adaptors at the top and bottom to hold the assembly in place in the reactor core.

Other Material and Geometric Details

Though the fuel in the SAFARI-1 reactor core would ideally be a pure uranium-aluminum alloy, there are some notable impurities present in the uranium used. However, since determination of the exact mass percent of each impurity is very difficult, only data pertaining to the known limit of each impurity per unit volume of uranium is known. The chemical composition of the uranium has the following properties: at least 99.9% uranium by mass, no more than 0.04% carbon by mass, and the density of the uranium-235 used is 0.84245 g/cm^3. The absorption of the uranium-aluminum alloy does not exceed the equivalent absorption of 4.45 mg boron-10 / kg assembly.

The absorption levels of the significantly absorbing, non-uranium materials found in assemblies within the reactor core are specified. The boron in the core has the equivalent absorption to 0.1832 mg boron-10 / mg assembly. The cadmium in the core has the equivalent absorption to 0.0569 mg boron-10 / mg assembly. The lithium in the core has the equivalent absorption to 0.0266 mg boron-10 / mg assembly.

The surface of all uranium used in the core has been freed of slag and nearly all other surface contamination. Only visible uranium oxide film has been left on the uranium metal. [1]

For clarity, a direct, side by side, axial comparison of the position of a reflector assembly and an active fuel assembly is shown in Figure 16.

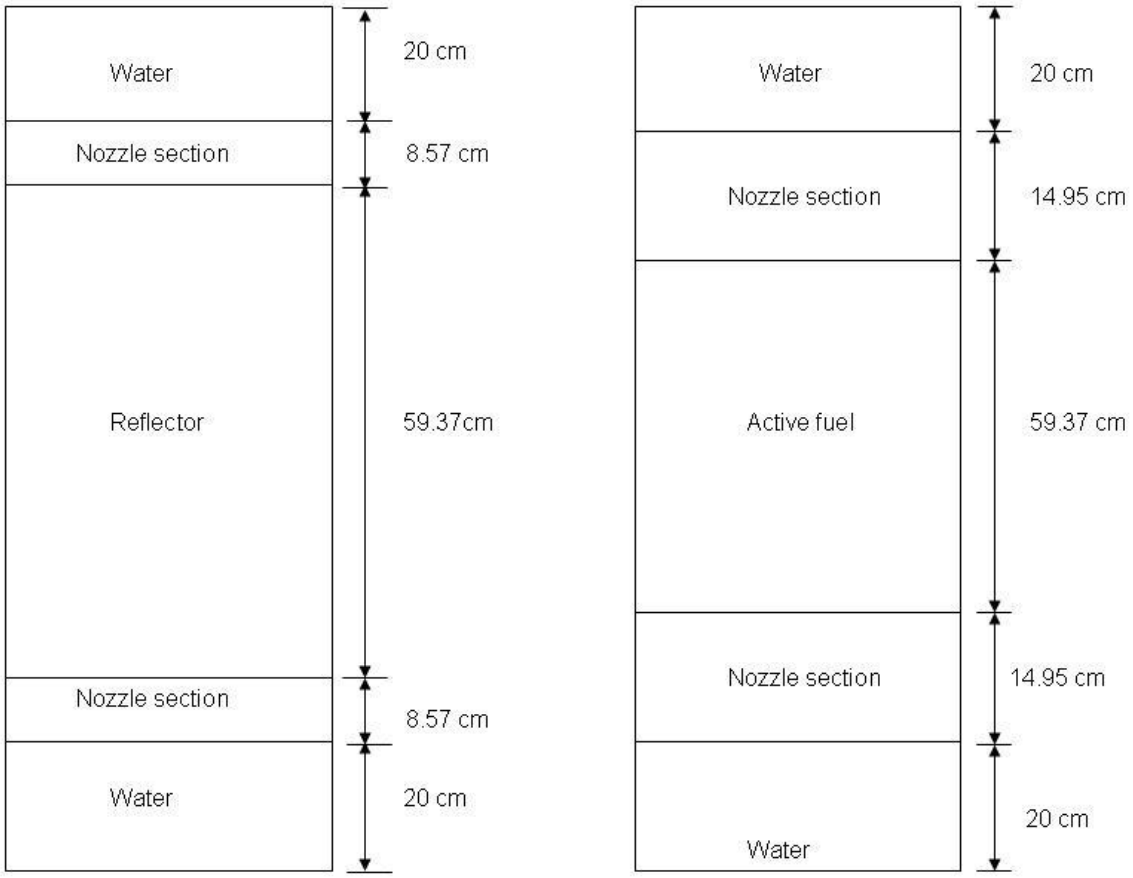

Figure 16: Axial Comparison of a Reflector Assembly (left) and an Active Fuel Assembly (right). [1]

CHAPTER 3. Neutronics Model Using NEM

Introduction to Reactor Neutronics

In the course of simulating a nuclear reactor, it is useful to consider the phenomena generally most directly associated with nuclear reactors, the study of which is collectively referred to as reactor physics, or neutronics. Neutronics is a blanket term used to describe the states and reactions of neutrons within a nuclear reactor as they interact with one another, interact with atoms, or escape from the volume of interest. To model the neutronics of a nuclear reactor, the free neutrons in the reactor must be simulated, whether by calculation, through the use of correlative methods, or some combination of both. The simulation must include the activity, location, energy, and quantity of neutrons from the time they are freed from atoms until the time they are absorbed, or leak out of the core, including any scattering during that period.

From a neutronics simulation, many things can be determined about the reactor. Of utmost importance is the calculation of the whole reactor multiplication factor. The whole reactor multiplication factor is the ratio of the number of neutrons created in a new generation to the number of neutrons in the last generation. This value is not necessarily time dependent, but rather assumes that the all neutrons that exist in the reactor at a given time will eventually either induce fission or lose the ability to do so due to taking part in another phenomenon.

The dependency of the multiplication factor on time is introduced in order to account for delayed neutrons. In contrast to prompt neutrons, which are freed as initial products of the previous neutron generation, delayed neutrons are freed due to atomic decay resulting from the prompt neutron interactions in the current generation. A

relatively significant amount of time may pass between the release of prompt neutrons and some atomic decays that free delayed neutrons. The time dependency may then be reduced out of the principal equations using advanced differential calculus and several experimentally or computationally derived coefficients.

The amount of power generated by a fission reactor can be directly attributed to the characteristics of the fission reactions occurring in the reactor. The aforementioned value known as the multiplication factor is a common indicator of the current state of increase, decrease, or stability of the core. The multiplication factor is solved for analytically or computationally as the result of an eigenvalue problem. This eigenvalue is the desired generational ratio of neutron yields. To easily refer to this ratio when citing its effect on other reactor parameters, the terms, critical, supercritical, or subcritical are often used. A critical reactor has a ratio equal to 1, a supercritical reactor has a ratio greater than 1, and a subcritical reactor has a ratio less than 1. For the majority of power reactors, the multiplication factor should indicate that the reactor is critical while in steady-state operation. Though there are some notable exceptions to this rule, the SAFARI-1 MTR is consistent with this rule.

A very important property in evaluating neutronics phenomena is the set of material cross sections. A cross section is a way of determining the distance which a neutron will most likely pass through the given material before an associated phenomenon takes place. The cross section is the mathematical inverse of the mean free path, a value which directly gives the aforementioned average distance. There is a cross section for each type of neutronics phenomenon, including scattering, absorption, and fission.

The cross section for a given material and phenomenon is dependent on many qualities, but the temperature of the surroundings and the neutron energy are two of the most important values to know well and to have available for use when determining said cross section. Neutron energy is calculated by neutronics simulators, but temperature is usually assumed to be constant throughout the reactor in stand-alone analyses. Chapter 4 of this paper will address a means of finding the temperature through the use of reactor thermal hydraulics simulation, and Chapter 5 will address the coupling of the codes for a more accurate result.

Overview of OSCAR

OSCAR is a deterministic neutronics system utilized by NECSA for simulation and analysis of SAFARI-1. OSCAR has been in use since 1994, and uses a combination of utility codes, 2D lattice analysis, and 3D nodalization methods to accurately model the nuclear phenomena in the uniquely designed MTR. OSCAR itself is comprised of a large number of simpler codes, including utility codes created by NECSA, cross sectional derivation and homogenization tools, and a built-in spatial package called Multi-Group Reactor Analysis Code (MGRAC). OSCAR is maintained and improved upon by NECSA's RRT group. [2]

The primary operations of OSCAR comprise a system which generates parameters using homogenized nodal equivalencies, which factor in discontinuity factors at adjacent cell borders. The nodes are traditionally placed radially as entire assemblies, and axially at various points along those assemblies. A color-set configuration is available for modeling the various types of control rod mechanisms, but the homogenization limits the accuracy of such models. The control rods themselves, the irradiation rigs, and the reflector segments are also available to model with the color-set, but again the cross section homogenization is a limiting aspect.

Since its development, the staff of the SAFARI-1 reactor has relied upon OSCAR to provide accurate predictions about upcoming fuel cycles. Additionally, OSCAR has been used to simulate previous fuel cycles before their implementation, and the code has been verified by the success of those designs when used with the actual reactor. OSCAR is regularly used for safety analysis of the reactor, one of the most technically and politically scrutinized areas of nuclear technology.

The code has proven to be a valuable tool for NECSA. As such, it is used to calculate reload parameters for the reactor, which have been successful for the duration of the code's use. Due to the delicate nature of nuclear reactor calculations, specifically control and prediction analyses that are used in practical applications, the continuing utilization of OSCAR as such a tool by NECSA speaks to its validation as an accurate neutronics simulator for use with the SAFARI-1 reactor. Thus, the steady-state analyses of the SAFARI-1 fuel cycle details solved by OSCAR for the SAFARI-1 benchmarking project serve as an appropriate, if not the best possible, reference solutions to which NEM results could have been compared.

OSCAR Reference Solutions

The full solution of the SAFARI-1 steady-state neutronics as determined by OSCAR is expansive, and presenting it in its entirety would be both excessive and detailed perhaps beyond usefulness. It was therefore prudent to extract from the full solution only those quantities which are directly relevant to the study presented here, and/or specifically used by the coupling process. These desired goals effectively filtered down the important results to a few values and sets of values.

First, the whole reactor multiplication factor (k_{eff}) was found. Experimentally, k_{eff} for SAFARI-1 is exactly 1 at steady-state, but the models predict it to be slightly higher. A likely contribution to the discrepancy is the energy group discretization of the cross sections. This is somewhat corroborated by the tendency of k_{eff} to approach 1 as the number of energy groups in the model increases. Using a continuous energy spectrum, as is the case in reality, would theoretically predict a k_{eff} closest to reality, though some error for other sources might remain. Use of a continuous energy distribution is not currently possible using OSCAR, though some Monte Carlo codes, such as A General Monte Carlo N-Particle Transport Code – Version 5 (MCNP5), have this capability. The 2-group and 6-group multiplication factors as calculated by OSCAR are presented below in Table 1.

Table 1: Multiplication Factors for OSCAR 2-Group and 6-Group Models.

Model	k_{eff}
2-Group	1.07355
6-Group	1.02288

Second, the radial power distribution was found, as it is important and represents the x and y directions. This is of macroscopic use while preparing fuel placement for refueling the reactor, as well as being one of the components used by the thermal hydraulics component of the coupled codes. The power distribution of all 210 radial cells was found, and is presented in two ways each for the 2-group and 6-group models. Figures 17 and 19 present the radial power distributions in numerical arrays which are useful for technical applications, and Figures 18 and 20 present the same distributions in an intuitively easy to grasp 3D format.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1.245	1.232	1.322	0	1.244	0	0	0	0	0
0	0	0	0	0	0	1.183	0.816	1.168	0.918	1.116	0	0	0	0
0	0	0	0	0	1.243	1.1	0.953	0	1.175	0	0	0	0	0
0	0	0	0	0	0	0.98	0.631	0.947	0.748	1.008	0	0	0	0
0	0	0	0	0	1.03	0.947	0.9	0	1.003	0	0	0	0	0
0	0	0	0	0	0	1.07	0.561	0.893	0.841	0.915	0	0	0	0
0	0	0	0	0	0.98	0.964	0.983	0.983	0.902	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 17: Numerical Array of the OSCAR Radial 2-Group Power Distribution. Values are for steady-state operating conditions, and have been normalized over the non-zero assemblies. Cells colored in blue are producing the least power, and cells colored in red are producing the most power. View is overhead, and the top of the figure is facing south.

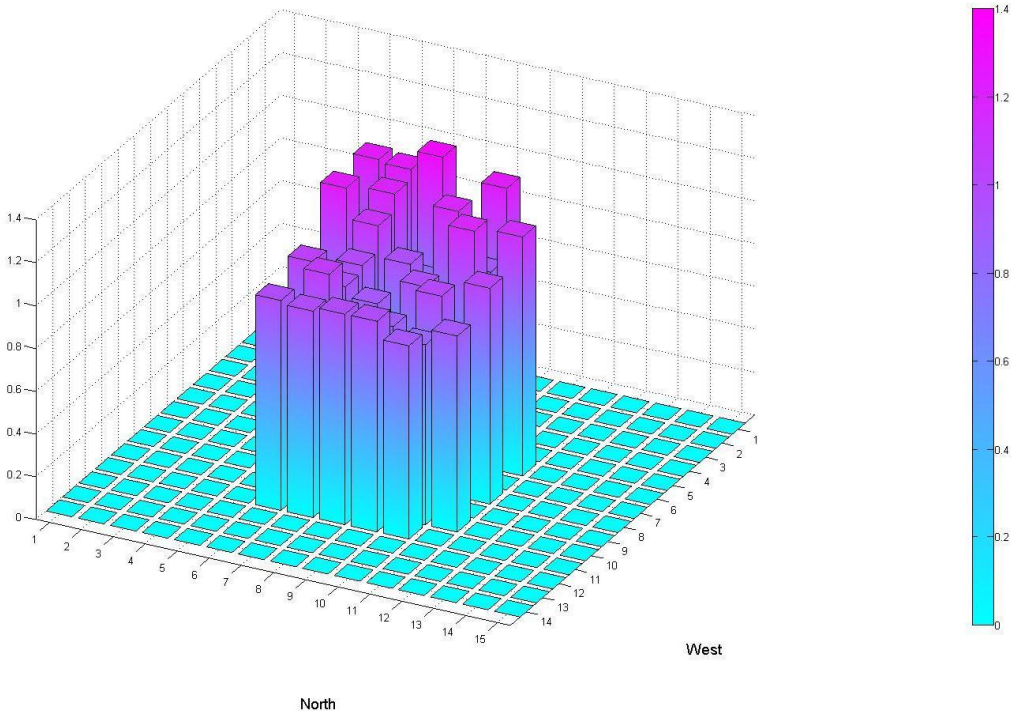

Figure 18: Three Dimensional Visualization of the OSCAR Radial 2-Group Power Distribution. Values are for steady-state operating conditions, and have been normalized over the non-zero assemblies. This is an overhead view.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1.199	1.168	1.275	0	1.214	0	0	0	0	0
0	0	0	0	0	0	1.148	0.81	1.168	0.901	1.117	0	0	0	0
0	0	0	0	0	1.245	1.097	0.979	0	1.196	0	0	0	0	0
0	0	0	0	0	0	1.002	0.662	0.995	0.773	1.048	0	0	0	0
0	0	0	0	0	1.059	0.961	0.937	0	1.036	0	0	0	0	0
0	0	0	0	0	0	1.067	0.574	0.911	0.839	0.933	0	0	0	0
0	0	0	0	0	0.961	0.928	0.956	0.955	0.886	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 19: Numerical Array of the OSCAR Radial 6-Group Power Distribution. Values are for steady-state operating conditions, and have been normalized over the non-zero assemblies. Cells colored in blue are producing the least power, and cells colored in red are producing the most power. View is overhead, and the top of the figure is facing south.

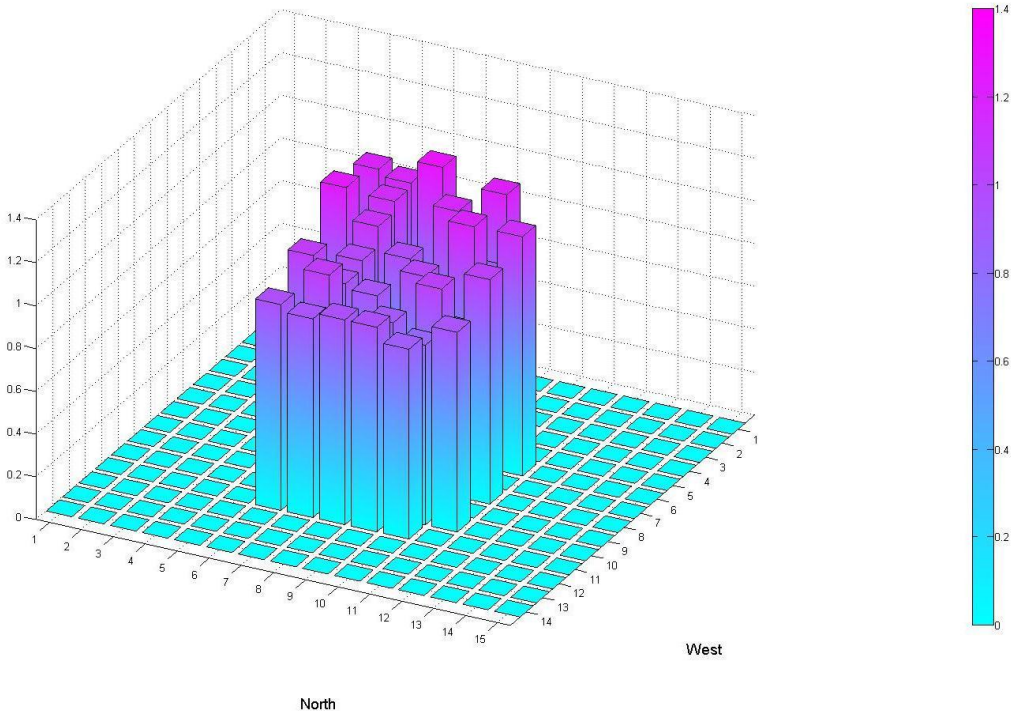

Figure 20: Three Dimensional Visualization of the OSCAR Radial 6-Group Power Distribution. Values are for steady-state operating conditions, and have been normalized over the non-zero assemblies. This is an overhead view.

Overview of NEM

NEM is a deterministic, 3D neutronics solver. NEM was developed and is maintained by the RDFMG at PSU. NEM was designed by PSU for use with discrete energy groups, rather than continuous groups as in some Monte Carlo tools, and currently supports up to 7 such groups. The models created using NEM may exist in Cartesian, hexagonal, or cylindrical geometries. The primary solving method used by the NEM code is the transverse integration method, which makes use of a semi-analytical transverse flux representation and a transverse leakage approximation. Neutron transport in the code is approximated through standard neutron diffusion using the diffusion equation for however many groups the simulation has been configured to analyze. [3]

NEM makes substantial use of nodalization techniques, though the nodes can be refined to a great degree so long as an adequate cross section library is available. By itself, NEM is unable to generate cross sections, and instead relies on manual input of all applicable cross section and material data. Neutron flux which is dependent on time is approximated through the use of a first order fully implicit finite difference scheme, which includes an exponential transformation technique. Neutron precursor distributions are modeled with a simpler linear time-integrated approximation.

NEM allows for flexibility in assigning a set number of inner and outer iterations, which are necessary for proper convergence behavior in a deterministic code. The number of upscatter iterations per outer iteration is also variable, and is assigned separately. Point and average normal convergence criteria for nodal fission sources and effective multiplication factor are, like the aforementioned variables, controlled via the

input. [10] NEM uses the Response Matrix (RM) technique for inner iterations with respect to each evaluated energy group. [3]

NEM is written in Fortran, and defines a specific format governing the creation of user created input decks. The input deck is divided into subsections, which are the general problem definition and control input, energy group and delayed neutron input, cross section input, nodalization and boundary conditions, material specification input, assembly input, and course-mesh nodalization. [10]

Modeling SAFARI-1 with NEM

Creation of the NEM neutronics model was done primarily using the NEM user manual [10], the source code of NEM, the SAFARI-1 benchmark specification [1], and externally homogenized cross section descriptions. The stand-alone neutronics model was created with NEM in two versions, a 2-group and a 6-group version. When comparing the OSCAR and NEM results, it is beneficial to have multiple distributions by energy group quantity, as to eliminate variation of energy groups as a possible source of significant error when troubleshooting. The models made use of both delayed and prompt neutrons.

The NEM models were created using Cartesian geometry, due to the unique assembly geometry types. Whereas many power reactors make use of cylindrical or hexagonal symmetry, the SAFARI-1 has a largely rectangular build, and no convenient axes of symmetry. Both models assume steady-state hot full power operating conditions. The number of iterations varied between trials, since coarser results were acceptable for preliminary results. In the course of testing, it was concluded that optimal results with respect to accuracy and computational resources were reached with 100 outer iterations, and 4 inner iterations per outer iteration.

Upscattering was considered for transfer from all energy groups to all higher energy groups. It was determined that two upscatter iterations per outer iteration produced the most accurate solutions. Fission cross sections were chosen rather than input power cross sections. The fission and other cross sections were homogenized by node, and were prepared by the NECSA staff based on their experimentally verified models, designed specifically for the SAFARI-1 MTR. [9] After receiving the NECSA cross sections, they were reformatted with a short Fortran script for use with NEM.

Probably the most particular details of a reactor model are the way in which it is nodalized for use of the homogenized cross sections. The nodalization was done for the entire 3D space of the reactor, and is presented below in Figure 21 and Table 2.

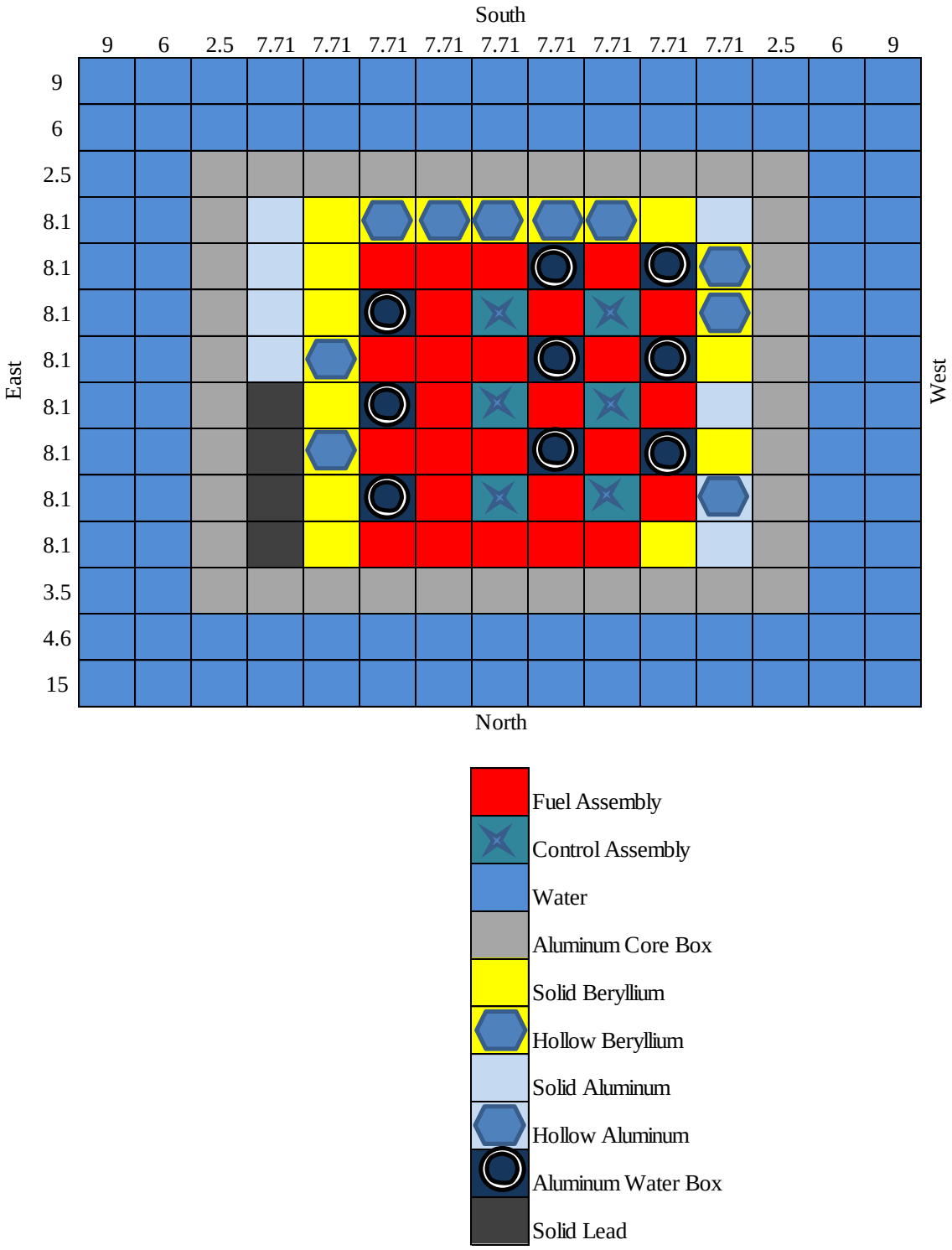

Figure 21: Radial Nodalization Scheme of the SAFARI-1 Utilized within NEM Models. Numerical labels along the left and top edge of Figure 17 are nodal dimensions along the corresponding edge of the adjacent node, and are given in cm. Dimensions labels are accurate, but figure itself is not to scale. This is an overhead view. [11]

Table 2: Axial Nodalization Scheme of the SAFARI-1 Utilized within NEM Models.

Axial Layer (1 is lowest)	Height [cm]
12	7.5
11	7.5
10	7.421
9	7.421
8	7.421
7	7.421
6	7.421
5	7.421
4	7.421
3	7.421
2	7.5
1	7.5

NEM Solutions

The full solution of the SAFARI-1 steady-state neutronics as determined by NEM is expansive, like that of OSCAR, and likewise presenting it in its entirety would be both excessive and detailed perhaps beyond usefulness. Therefore the same set of desired goals was used to select the results for close examination. Again, these results were the multiplication factors for the whole reactor and the radial power distributions.

First, the whole reactor multiplication factor k_{eff} was found. Experimentally, k_{eff} is exactly 1 at steady-state, but as with the case of the OSCAR solutions, the NEM models predict it to be slightly higher. A likely contribution to the discrepancy is the energy group discretization of the cross sections, which is again somewhat corroborated by the tendency of k_{eff} to approach 1 as the number of energy groups in the model increases. As with OSCAR, this particular source of error could be amended if a method of implementation of a continuous energy spectrum was devised for this code, though some error for other sources might remain. The 2-group and 6-group multiplication factors are presented below in Table 3.

Table 3: Multiplication Factors for NEM 2-Group and 6-Group Models.

Model	k_{eff}
2-Group	1.07360
6-Group	1.02294

Second, the radial power distribution was found, as it is important and represents the x and y directions. As in the case of the OSCAR predictions, this is of macroscopic use while preparing fuel placement for refueling the reactor, as well as being one of the components used by the thermal hydraulics component of the coupled codes. The power distribution of all 210 radial cells was found using NEM, and is presented in two ways each for the 2-group and 6-group models. Figures 22 and 24 present the radial power distributions in numerical arrays which are useful for technical applications, and Figures 23 and 25 present the same distributions in an intuitively easy to grasp 3D format.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1.244	1.23	1.322	0	1.243	0	0	0	0	0
0	0	0	0	0	0	1.182	0.816	1.168	0.917	1.115	0	0	0	0
0	0	0	0	0	1.242	1.1	0.954	0	1.176	0	0	0	0	0
0	0	0	0	0	0	0.98	0.631	0.948	0.749	1.007	0	0	0	0
0	0	0	0	0	1.03	0.947	0.901	0	1.003	0	0	0	0	0
0	0	0	0	0	0	1.07	0.561	0.894	0.841	0.915	0	0	0	0
0	0	0	0	0	0.98	0.964	0.983	0.983	0.902	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 22: Numerical Array of the NEM Radial 2-Group Power Distribution. Values are for steady-state operating conditions, and have been normalized over the non-zero assemblies. Cells colored in blue are producing the least power, and cells colored in red are producing the most power. View is overhead, and the top of the figure is facing south.

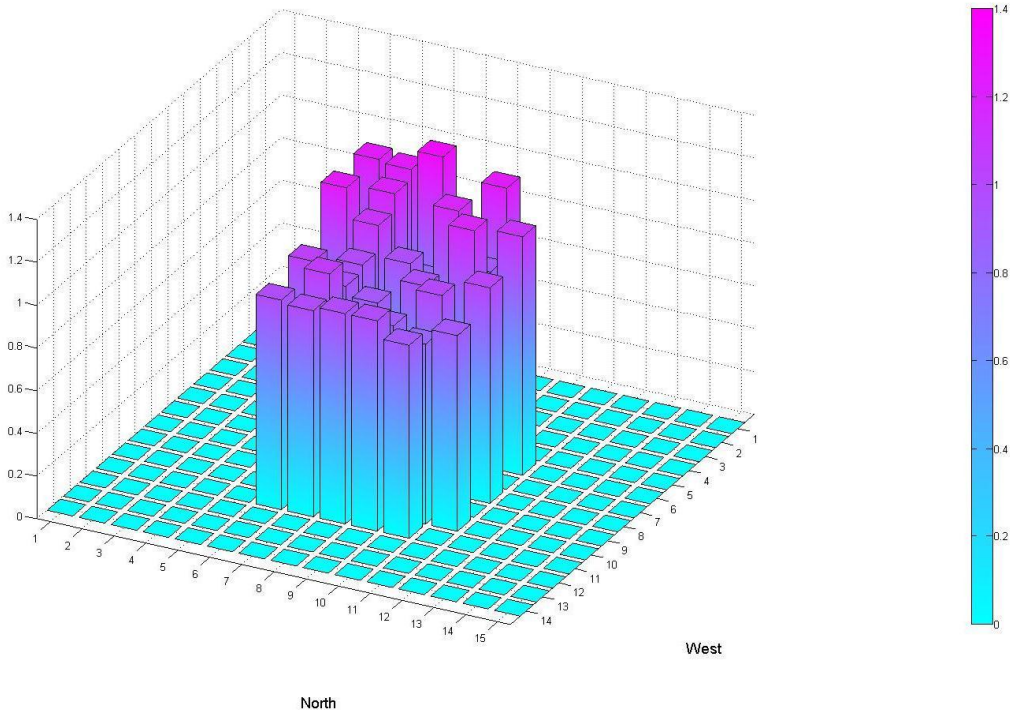

Figure 23: Three Dimensional Visualization of the NEM Radial 2-Group Power Distribution. Values are for steady-state operating conditions, and have been normalized over the non-zero assemblies. This is an overhead view.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1.198	1.166	1.274	0	1.213	0	0	0	0	0
0	0	0	0	0	0	1.147	0.81	1.169	0.9	1.116	0	0	0	0
0	0	0	0	0	1.244	1.097	0.981	0	1.196	0	0	0	0	0
0	0	0	0	0	0	1.004	0.664	0.997	0.773	1.047	0	0	0	0
0	0	0	0	0	1.059	0.961	0.939	0	1.037	0	0	0	0	0
0	0	0	0	0	0	1.066	0.575	0.912	0.839	0.932	0	0	0	0
0	0	0	0	0	0.96	0.927	0.956	0.955	0.886	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 24: Numerical Array of the NEM Radial 6-Group Power Distribution. Values are for steady-state operating conditions, and have been normalized over the non-zero assemblies. Cells colored in blue are producing the least power, and cells colored in red are producing the most power. View is overhead, and the top of the figure is facing south.

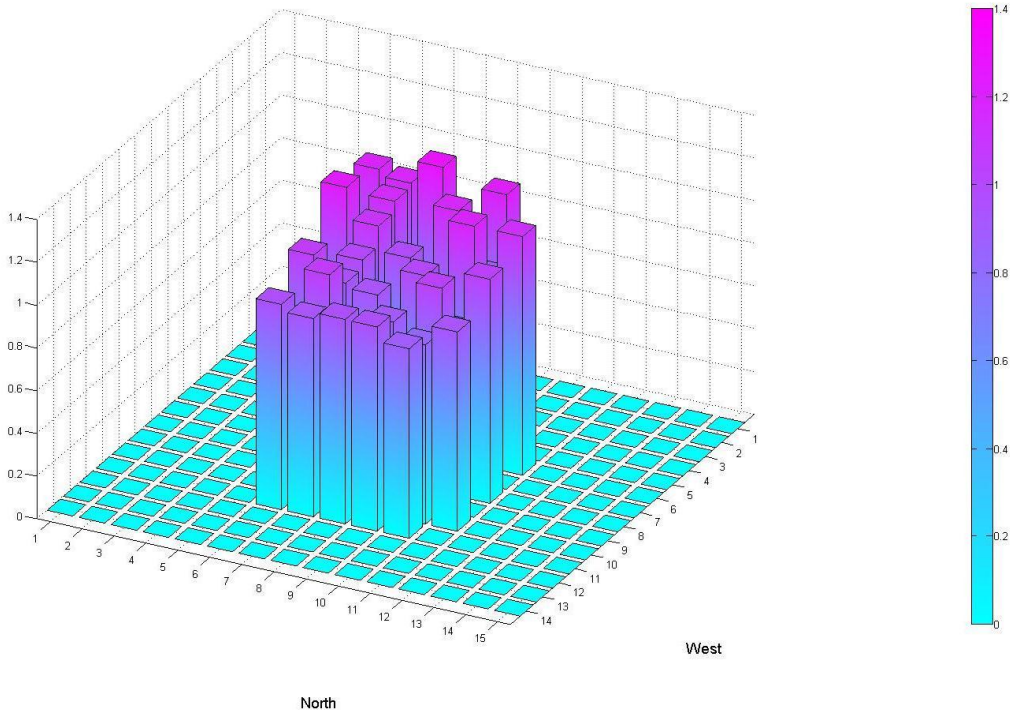

Figure 25: Three Dimensional Visualization of the NEM Radial 6-Group Power Distribution. Values are for steady-state operating conditions, and have been normalized over the non-zero assemblies. This is an overhead view.

Comparison of OSCAR and NEM Solutions

Having established and presented the OSCAR and NEM solutions to the SAFARI-1 benchmark problem for steady-state conditions, it is useful to directly compare the two sets of results. The goal of the stand-alone NEM solution is to be similar to that of the OSCAR solution, since the OSCAR solution is regularly and successfully used in practical applications with the SAFARI-1. However, due to the nature of using multiple means of finding the same solution for a benchmark, the end results are unlikely to be identical. In fact, having identical results for something as complicated as the application studied here would probably indicate that the same means were used, thereby eliminating the significance of achieving similar results utilizing somewhat variant methods.

The first part of the solutions to be compared is the whole reactor multiplication factor. Unlike typical benchmarking project contributions and the comparisons made therein, and indeed unlike most practical engineering applications, the multiplication factor for a nuclear reactor must be calculated with a nearly impeccable level of accuracy. Therefore absolute percent difference, a standard level of precision for considering multiple like trends, is usually not used for the multiplication factor. Instead, per cent mille (pcm) is used. One percent difference is equivalent to one thousand pcm. Whereas a few pcm is a fairly negligible error in a multiplication factor, one percent absolute (1000 pcm) is usually considered to be an unacceptable level or error. Ultimately, the NEM derived multiplication factors were shared with the NECSA staff, who indeed confirmed the accuracy of the results of this study due to the level of agreement agreement with the

OSCAR results. The comparison of the OSCAR and NEM multiplication factors is presented below in Tables 4 and 5 for the 2-group and 6-group models respectively.

Table 4: Multiplication Factors Comparison of OSCAR and NEM 2-Group Models.

Code or Statistic	k_{eff}
OSCAR	1.07355
NEM	1.07360
Difference [pcm]	5

Table 5: Multiplication Factors Comparison of OSCAR and NEM 6-Group Models.

Code or Statistic	k_{eff}
OSCAR	1.02288
NEM	1.02294
Difference [pcm]	6

These discrepancies in the multiplication factor are reasonable. A major difference between NEM and OSCAR that may have led to the differences is that NEM uses polynomial nodal expansion, while OSCAR uses analytical nodal expansion. Additionally, differences in the computing environment may have altered the multiplication factor. These may include the compiler used, the computational hardware, and the operating system.

The second part of the solution to be compared was the radial power distribution. According to an industry professional, nodal power distribution results in a reactor benchmarking project can be considered acceptable if the trends stay within a few absolute percent, very close if they are within 0.5 absolute percent, and ideal if they are within 0.1 absolute percent. [12] These goals were used when solving using NEM. As with the multiplication factors, the decision of acceptability of the NEM derived radial power distributions was left with the NECSA staff. As with the multiplication factors, NECSA approved of the accuracy of the NEM nodal power distributions due to their level of agreement with the OSCAR results. The reasons for the differences in the power distributions are the same as those given for differences in the multiplication factors between codes.

Figures 26 and 27 provide the detailed numerical array formatted comparisons for each radial cell of the two codes 2-group and 6-group models respectively, and Tables 6 and 7 present the overall statistical analyses of the comparison of the two codes' overall distributions.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	-0.1	-0.16	-0.04	0	-0.1	0	0	0	0	0
0	0	0	0	0	0	-0.05	0.001	0.03	-0.06	-0.07	0	0	0	0
0	0	0	0	0	-0.06	-0.04	0.094	0	0.046	0	0	0	0	0
0	0	0	0	0	0	0.05	0.08	0.108	0.06	-0.04	0	0	0	0
0	0	0	0	0	0.009	0.03	0.114	0	0.097	0	0	0	0	0
0	0	0	0	0	0	-0.01	0.069	0.076	0.001	-0.03	0	0	0	0
0	0	0	0	0	-0.03	-0.05	0.009	-0.03	-0.02	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 26: Numerical Array Comparison of the OSCAR and NEM 2-Group Power Distributions. Values are percent difference by cell between the normalized, steady-state values found in Figures 17 and 22. Cells colored in green are overestimations by NEM, cells colored in red are underestimations by NEM, and cells colored in yellow are in agreement between the codes. Unpowered cells are not colored. View is overhead, and the top of the figure is facing south.

Table 6: Radial Power Distribution Comparison of OSCAR and NEM 2-Group Models. Averages were taken over non-zero cells only, since agreement in non-fuel cells producing no power is trivial.

Statistic	Absolute Percent Difference
Average Per Cell	0.05
Maximum in a Cell	0.16

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	-0.17	-0.2	-0.1	0	-0.13	0	0	0	0	0
0	0	0	0	0	0	-0.06	0.02	0.063	-0.12	-0.12	0	0	0	0
0	0	0	0	0	-0.05	0.01	0.2	0	0.07	0	0	0	0	0
0	0	0	0	0	0	0.122	0.139	0.186	0.075	-0.07	0	0	0	0
0	0	0	0	0	0.009	0.075	0.203	0	0.13	0	0	0	0	0
0	0	0	0	0	0	-0.04	0.11	0.112	-0.06	-0.07	0	0	0	0
0	0	0	0	0	-0.1	-0.11	-0.01	-0.08	-0.04	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 27: Numerical Array Comparison of the OSCAR and NEM 6-Group Power Distributions. Values are percent difference by cell between the normalized, steady-state values found in Figures 19 and 24. Cells colored in green are overestimations by NEM, cells colored in red are underestimations by NEM, and cells colored in yellow are in agreement between the codes. Unpowered cells are not colored. View is overhead, and the top of the figure is facing south.

Table 7: Radial Power Distribution Comparison of OSCAR and NEM 6-Group Models. Averages were taken over non-zero cells only, since agreement in non-fuel cells producing no power is trivial.

Statistic	Absolute Percent Difference
Average Per Cell	0.10
Maximum in a Cell	0.20

CHAPTER 4. Thermal Hydraulics Model Using CTF

Introduction to Reactor Thermal Hydraulics

In many types of engineering, such as mechanical, electrical, civil, and nuclear engineering, the temperature of a system is an important consideration when designing solutions to various problems. In nuclear engineering, the temperature of a given volume of material affects the nuclear cross sections of the volume, which in turn affects the neutronics phenomena. Therefore the temperatures of the coolant and fuel are of particular interest. Since the nodal volumes are constant by definition within the code, but temperatures will vary in a transient analysis of this type, the moderator density is also of particular interest.

Heat transfer in a system may occur through various methods, including conduction and convection. Conduction refers to the depositing of energy by one material into another through direct contact, while convection refers to energy transfer which is specifically reliant on the motion of adjacent particles with respect to one another. Other methods of heat transfer include energy exchange between materials due to emission of radiation and transfer of mass to or from a volume of interest. Radiation may occur in liquid and/or moving bodies as well as in solid and/or stationary bodies, but mass transfer, like convection, is caused by the movement of particles. Because of the dependencies of some heat transfer modes on motion, it is necessary to analyze that motion in order to analyze the subsequent exchange of energy. Thus, thermal hydraulics focuses on both heat transfer and fluid flow in order to solve complex problems.

In a nuclear reactor, where energy is transferred as heat throughout the static and fluid materials, a more detailed analysis combining the local temperatures with fluid flow

is required for accurate determination of associated system properties. Most of the major aspects of thermal hydraulic simulation are well-defined, though some, like turbulent crossflow between channels, are still actively being refined.

Overview of RELAP

Although the CTF code used for this study was created and verified primarily in direct coordination with experiments run by NECSA with the SAFARI-1 MTR, it is generally desirable to compare any code used in an original study such as this one with another code. This is the nature of benchmarking, and thus a previous thermal hydraulics code used to run simulations with SAFARI-1 was used to verify the CTF results. The RELAP solution uses various sizes of mesh for spatial distribution and calculates all state parameters for assemblies as equivalent to those of other assemblies of the same type. Though this is different than the method used by CTF, the temperatures and other measured parameters should generally be within an order of magnitude from what was predicted by the CTF model. Unlike the OSCAR solutions used as guidelines for the NEM model when capturing the neutronics of SAFARI-1 in hot full power steady-state operating conditions, the CTF model was written entirely independently of its predecessor RELAP model.

RELAP is a program developed to model and analyze many aspects of transient periods in nuclear reactors, specifically those of a light water reactor (LWR) type build. Other types of reactors catered to by RELAP are fusion reactors, space reactors, gas cooled reactors, fast breeder reactors, and cardiovascular blood flow simulations. It was produced by the Idaho National Laboratory (INL), and had its most recent update RELAP5 in 2005. Features of the RELAP code include nonhomogeneous modeling of two-phase systems without equilibrium, which is designed for efficiency in balancing speed and computational accuracy. To do so, mostly first order effects are considered during calculations. [4]

RELAP can model a number of components through its default functions, including pumps, valves, pipes, heat releasing or absorbing structures, electric heaters, jet pumps, turbines, separators, annuli, pressurizers, feed water heaters, engineered cementitious composite (ECC) mixers, accumulators, and control system components. Some special features of RELAP include monitoring of form loss, flow in an area of abrupt change, branching, choked flow, boron tracking, and incondensable gas transport. The code also has a user friendly interface which alerts users to errors and inconsistencies within a working input. Some of the latest abilities of RELAP relate to its coupling with several other subroutines, resulting in RELAP5-3D, which models the aforementioned situations with 3D considerations. [4]

RELAP Reference Solutions

The RELAP solution in its entirety would not serve much use for verification purposes due to the solutions being found over different regions and to a generally less precise degree than the CTF equivalents. However, the orders of magnitude should agree, and this is what is examined here. It is thus prudent to tabulate the data which can be compared directly to the CTF solution. Although many parameters may be examined, it is sufficient for the purpose of a rough comparison to examine only a few important parameters thoroughly. Therefore, the liquid and vaporized coolant temperatures of the fuel assemblies and the control and follower assemblies will be presented. This data is shown below in Table 8.

Table 8: Stand-Alone RELAP Results. All temperatures are degrees Fahrenheit and have been homogenized by assembly type. [13]

Assembly Type	Material	Temperature	Mesh Size
Fuel Assembly	Liquid Coolant	111.335	100
Fuel Assembly	Vaporized Coolant	246.472	106.5
Control and Follower Assembly	Liquid Coolant	105.850	106.5
Control and Follower Assembly	Vaporized Coolant	247.388	106.5

Overview of CTF

COBRA-TF is a subchannel code used for thermal hydraulics analysis of nuclear reactors. In this study, a version called CTF was used, which is the result of improvements made by the RDFMG. The primary use of CTF is for estimation of safety margins in LWR layouts. CTF makes use of a three field representation of two-phase flow using semi-implicit, time-averaged conservation equations and donor cell differencing for convective phenomena. The code was designed to support the use of both Cartesian and subchannel geometric systems, the latter configuration being the more frequently used for standard power reactors. [5]

CTF is a deterministic code governed by convergence criteria and iterative limits, similarly to NEM. The mathematical calculations that go into the results output by the system are controlled by standard heat transfer and fluid conservation equations. The conservation equations used by CTF are mass, momentum, and energy. The phases and material conditions simulated by CTF include continuous vapor, continuous liquid, entrained liquid drops, non-condensable gas mixture, and additional fields to represent the small drops field as a result of large drops breaking up upon impact with spacer grids or other blockages. [5]

Specific phenomena modeled by CTF include: gravitation; transverse flow between subchannels caused by lateral pressure gradients, otherwise known as diversion crossflow; pressure losses due to friction, head losses, and interfacial drag forces; lateral exchange between subchannels caused by molecular and turbulent diffusion in axial flow, otherwise known as turbulent mixing; void drift; entrainment of droplets in annular flow; and deposition of droplets in annular flow. [5]

CTF processes the data through input decks created by a user. The code is written in Fortran, but has its own specialized format for input decks. Input includes the discrete definitions for main problem control data, including: selection of physical models, global boundary conditions, and initial conditions; channel descriptions; transverse channel connection data, otherwise known as gaps; vertical channel connection data; geometry variation data; channels and gaps affected by variation tables; local pressure loss coefficient and grid spacer data; rod and unheated conductor data; conductor geometry description; material properties tables; axial power distribution tables, radial power distribution, and transient forcing functions; turbulent mixing and void drift data; boundary condition data; output options; and time domain data.

Modeling SAFARI-1 with CTF

The CTF model of the SAFARI-1 MTR was created by in cooperation between PSU and NECSA staff for the purpose of eventually creating coupled neutronics/thermal hydraulics models. The version of the COBRA-TF thermal hydraulics simulator known as CTF, which was created by the RDFMG, is used for this study. The important aspects of the problem geometry were verified with the intention of error checking and for documentation purposes. Several assumptions were made in the CTF model, most of which were fairly trivial. Some such assumptions were made for practical purposes, such as ignoring insignificant, thin films of water. Other assumptions were made to simplify the model due to resource constraints, such as modeling hollow aluminum and beryllium assemblies as their solid counterparts.

The model used assumes steady-state hot full power operating conditions for the SAFARI-1 MTR. The nodalization scheme is the same as for NEM and OSCAR, so it will not be repeated here. There are 72 subchannels modeled, consisting of all reactor assemblies, and neglecting the aluminum core box and water. The duration simulated is 3.0 seconds, which technically makes the trials transient, though the constant power effectively models steady-state conditions. While CTF has the capacity to use time dependencies in step format for its various input parameters, none were used in this case. This is because the conditions of a nuclear reactor operating at steady-state remain the same when considering a relatively short duration with significant physical changes. In a more advanced simulation, for example a rod ejection problem, time dependent input parameters would be utilized.

The first set of important values used in the stand-alone CTF model of the SAFARI-1 reactor are the initial conditions. These are presented below in Table 9.

Table 9: Stand-Alone CTF Initial Conditions. The conditions represent realistic, steady-state operating conditions. Unlike most reactors, coolant in SAFARI-1 flows from the top of the reactor to the bottom.

Parameter	Value
Total Inlet Mass Flow Rate	788.756 [kg/s]
Average Linear Heat Rate Per Rod	58.6833 [kW/m]
Initial Pressure in the Fluid Domain	2.5 [bar]
Initial Enthalpy in the Fluid domain	167.62 [kJ/kg]

The second set of important values used in the CTF model of the SAFARI-1 reactor relate to the subchannels. These are presented below in Table 10.

Table 10: Stand-Alone CTF Subchannel Definitions. As stated previously, the hollow aluminum and beryllium assemblies have been modeled as their solid equivalents.

Subchannel Type	Number of Channels of Given Type Modeled	Nominal Channel Area [m^2]	Channel Wetted Perimeter [m]
Solid Aluminum Assembly	8	0.00018632	0.31146000
Solid Beryllium Assembly	19	0.00018632	0.311146000
Aluminum Water Box Assembly	9	0.00051031	1.65620000
Solid Lead Assembly	4	0.00056706	0.31144000
Fuel Assembly	26	0.00389757	2.82872360
Control and Follower	6	0.00365336	2.32561200

The third set of important values used in the CTF model of the SAFARI-1 reactor is the radial power distribution. While the radial distribution is very important in its role as an output of the neutronics, which is input to the thermal hydraulics in the coupled model, the initial distribution read into CTF only acts as a first guess. Because of this, the radial power distribution used in the stand-alone CTF only needs to be accurate enough to vaguely resemble realistic conditions. The stand-alone input contains a distribution

similar to the OSCAR and NEM results. Though not identical to those neutronics solutions, they were close enough for stand-alone comparisons. The radial power distribution used by the stand-alone CTF model of SAFARI-1 is presented below in Figure 28.

1.208	1.159	1.3		1.223	
	1.264	0.979	1.267	0.95	1.103
1.292	1.142	1.021		1.163	
	1.061	0.785	1.037	0.877	0.957
1.092	0.937	0.943		1.004	
	0.969	0.665	0.895	0.804	0.813
0.816	0.766	0.818	0.794	0.763	

Figure 28: Numerical Array of the Stand-Alone CTF Radial Power Distribution. While CTF defines the power distribution over all nodes, only the nodes containing fuel have non-zero power, so the values for the fuel and control and follower assemblies are the only ones shown here. Values are for steady-state operating conditions, and have been normalized over all defined assemblies. Cells colored in yellow are producing the least power, and cells colored in red are producing the most power. View is overhead, and the top of the figure is facing south.

The final set of important values used in the CTF model of the SAFARI-1 reactor is the set of axial power distributions. A different distribution is given for each fuel and control and follower assembly. Like the radial power distributions, the axial power distributions used in the stand-alone CTF test were close to those produced by the neutronics codes, and will not be of great significance in the final coupled model, since they will be produced anew by NEM for each iteration.

CTF Solutions

The primary role of thermal hydraulics analysis in this study is to serve as a coupled code component along with the neutronics. The neutronics code NEM outputs the power and power distributions that are used in the input of the thermal hydraulics code CTF. CTF returns temperatures for each node in the coupled model, which are input into NEM in order to select cross sections from a four dimensional (4D) library. Thus, even though CTF outputs a large quantity of information which may be very useful in some applications, the values of interest in this study are the nodal fuel temperatures, moderator temperatures, and moderator densities.

The CTF output gives the temperature for the fuel assemblies and control and follower assemblies individually, and additionally averaged over the entire core. The former of these resultant arrays are further divided into axial nodes for each radial node. For the purpose of verifying the stand-alone CTF model, it will suffice to provide radial temperature arrays integrated over the axial nodes. Additionally, it is necessary to investigate the temperature change with elevation in various types of assemblies.

The first radial array illustrates the temperature on the inside surface of the assemblies modeled by CTF. This is presented below in Figure 29.

104.204	104.446	105.192	105.383	105.417	105.252	105.32	104.863	104.201
104.203	104.685	113.476	113.014	115.67	104.238	114.416	104.216	104.36
104.201	104.619	104.183	115.216	113.47	112.92	113.47	113.687	104.817
104.766	105.01	111.999	112.508	113.216	104.221	113.567	105.349	104.788
105.428	105.335	105.081	111.485	113.47	112.542	113.47	111.047	104.975
105.294	105.545	109.666	110.535	109.837	105.376	110.933	105.291	104.851
105.334	105.445	105.314	111.456	113.47	110.085	113.47	109.423	104.97
105.494	105.64	109.56	109.159	107.874	109.024	108.398	105.251	104.932

Figure 29: Inside Surface Temperature Array from Stand-Alone CTF. Temperatures are in degrees Fahrenheit. The reactor is in steady-state operating conditions. Cells colored in yellow are the coolest, and cells colored in red are the hottest. View is overhead, and the top of the figure is facing south.

The second radial array illustrates the temperature of the vaporized coolant inside the assemblies modeled by CTF. This is presented below in Figure 30.

249.883	249.434	246.763	246.231	246.754	247.647	247.441	248.706	249.553
249.807	249.156	245.204	244.426	245.852	244.102	246.068	233.512	249.43
249.628	249.115	245.516	246.582	245.204	246.943	245.204	246.427	249.031
249.061	248.51	245.312	243.708	246.398	238.069	246.515	247.52	249.017
246.723	247.502	248.485	246.659	245.204	246.848	245.204	246.321	248.881
247.125	246.604	245.313	243.802	246.412	246.82	246.518	247.092	248.909
247.198	246.83	247.291	246.61	245.204	247.024	245.204	246.266	248.867
246.535	246.281	245.093	244.214	245.598	244.27	245.673	246.946	248.986

Figure 30: Vaporized Coolant Temperature Array from Stand-Alone CTF. Temperatures are in degrees Fahrenheit. The reactor is in steady-state operating conditions. Cells colored in yellow are the coolest, and cells colored in red are the hottest. View is overhead, and the top of the figure is facing south.

The third radial array illustrates the temperature of the liquid coolant inside the assemblies modeled by CTF. This is presented below in Figure 31.

104.135	104.546	105.421	105.714	105.879	106.04	105.676	105.348	104.131
104.134	104.839	113.498	113.021	115.679	104.22	114.422	104.195	104.433
104.131	104.894	104.164	115.228	113.498	112.928	113.498	113.697	105.049
104.9	105.274	112.004	112.511	113.223	104.202	113.574	105.931	105.283
105.816	105.749	105.366	111.494	113.498	112.55	113.498	111.053	105.254
106.071	105.888	109.67	110.539	109.841	105.694	110.938	105.976	105.438
105.714	106.104	105.67	111.464	113.498	110.089	113.498	109.428	105.252
105.957	106.02	109.566	109.161	107.875	109.028	108.4	105.574	105.315

Figure 31: Liquid Coolant Temperature Array from Stand-Alone CTF. Temperatures are in degrees Fahrenheit. The reactor is in steady-state operating conditions. Cells colored in yellow are the coolest, and cells colored in red are the hottest. View is overhead, and the top of the figure is facing south.

The fourth radial array illustrates the temperature on the outside surface of the assemblies modeled by CTF. This is presented below in Figure 32.

104.366	104.368	104.37	104.37	104.371	104.37	104.37	104.369	104.366
104.366	104.369	113.47	113.012	115.669	104.369	114.415	104.369	104.368
104.366	104.369	104.369	115.214	113.463	112.915	113.463	113.685	104.369
104.372	104.37	111.996	112.504	113.216	104.369	113.566	104.596	104.369
104.629	104.371	104.54	111.484	113.463	112.54	113.463	111.047	104.374
104.582	104.372	109.666	110.534	109.836	104.618	110.931	104.584	104.369
104.603	104.371	104.598	111.455	113.463	110.084	113.463	109.422	104.374
104.641	104.372	109.557	109.159	107.874	109.024	108.398	104.37	104.373

Figure 32: Outside Surface Temperature Array from Stand-Alone CTF. Temperatures are in degrees Fahrenheit. The reactor is in steady-state operating conditions. Cells colored in yellow are the coolest, and cells colored in red are the hottest. View is overhead, and the top of the figure is facing south.

Having examined the radial temperature arrays averaged over the axial nodes, the remaining step is to investigate the axial temperature change in various types of assemblies. Since liquid coolant is the working fluid, this will be the medium for the axial temperature analysis. The assemblies investigated will be the hottest fuel assembly and the coolest reflector assembly, based on the numerical array given in Figure 31.

The elevation dependent temperature for the hottest fuel cell is presented below in Figure 33, and the elevation dependent temperature of the coolest reflector assembly is presented below in Figure 34.

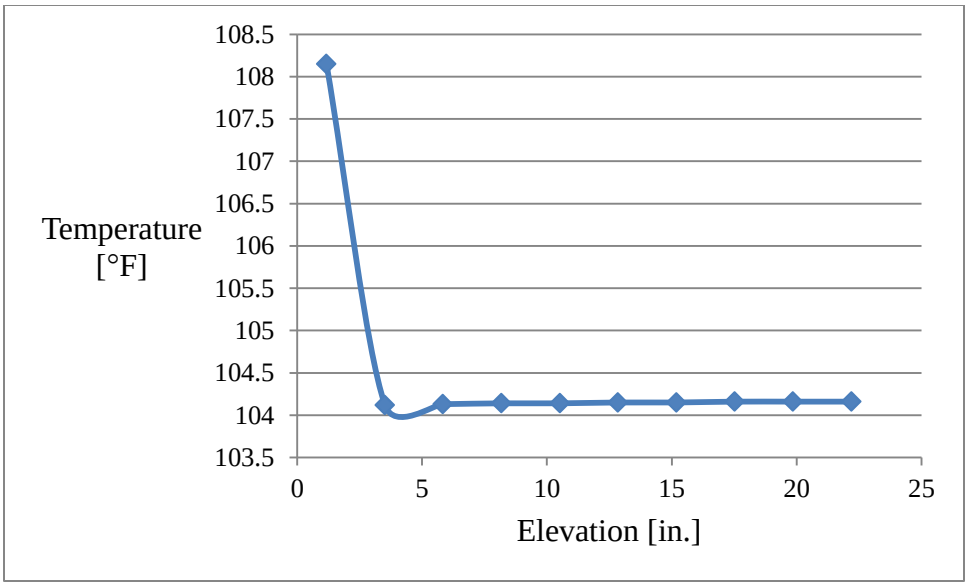

Figure 33: Coolest Reflector Assembly Liquid Coolant Temperature by Elevation. Liquid temperature decreases with elevation due to downward coolant flow.

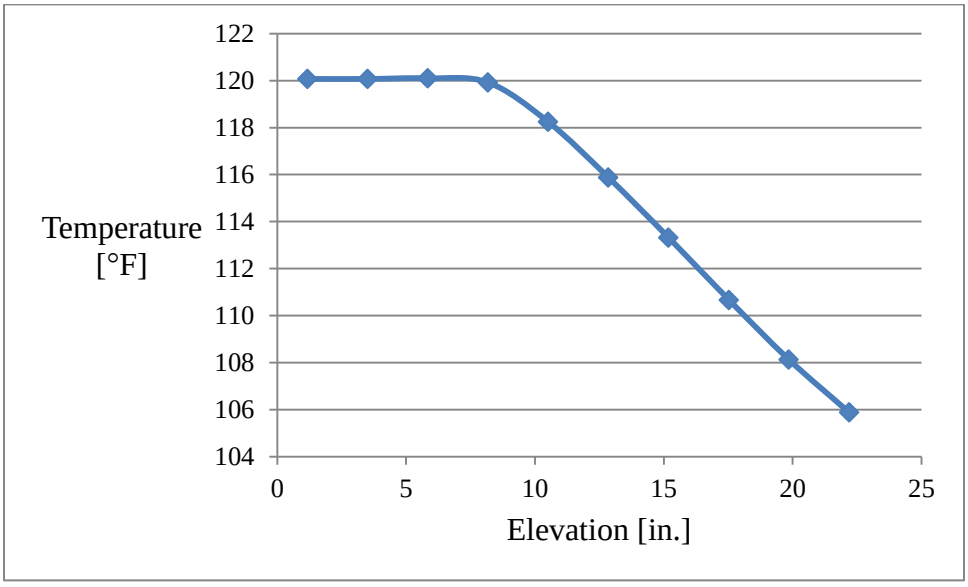

Figure 34: Hottest Fuel Assembly Liquid Coolant Temperature by Elevation. Liquid temperature decreases with elevation due to downward coolant flow.

Comparison of RELAP and CTF Solutions

Having presented both a representative sample of the thorough CTF results calculated for this study and the summary of the limited release RELAP results provided by NECSA, a comparison of the two thermal hydraulics solutions will now be shown. The restricting factor is the amount of data provided for RELAP, so the comparison can only be made in the form of a direct analysis of the RELAP solutions temperatures homogenized by assembly type with respect to the range of CTF solution temperatures by node. This comparison is shown below in Table 11.

Table 11: Comparison and Analysis of RELAP and CTF Solutions. All temperatures are in degrees Fahrenheit. RELAP results have been homogenized by assembly type, while CTF results are presented as a range over axially homogenized radial nodes.

Assembly Type	Material	RELAP Temperature	CTF Temperature Range	RELAP Result Within CTF Range?
Fuel Assembly	Liquid Coolant	111.335	107.875 to 115.679	YES
Fuel Assembly	Vaporized Coolant	246.472	244.27 to 246.943	YES
Control and Follower Assembly	Liquid Coolant	105.850	104.195 to 105.976	YES
Control and Follower Assembly	Vaporized Coolant	247.388	233.512 to 247.520	YES

CHAPTER 5. Coupled Model Using NEM and CTF

Introduction to Coupled Codes

In the field of nuclear engineering, there are two standout techniques when considering computational analysis of reactors. The first technique is referred to as Monte Carlo analysis, and the second is referred to as deterministic analysis. Monte Carlo analysis, such as that performed by the popular code MCNP5, tracks nearly every detail of the scenario modeled, such that the results are extremely accurate. However, the time and hardware limitations of modern computing systems prevent these methods from being useful in some applications. The deterministic approach utilizes certain assumptions that allow the time needed to complete each calculation to be greatly reduced. The drawback of these assumptions is that each one somewhat reduces the accuracy of the resulting model.

To create an accurate model of a nuclear reactor, the most difficult aspect at this point in time remains to be obtaining the set of values comprising the nuclear cross sections. Cross sections are found separately for each material, isotope, and nuclear phenomenon, and vary according to state parameters in a way which is often established experimentally and tabulated for use. Neutronics models are used primarily to predict multiplication factors and power distributions, so a set of cross sections, sometimes referred to as a cross section library, must be available for use. Since the state parameters at a given point in time cannot be predicted without the power, they must be assumed to have some reasonable set of values in order to intelligently select cross sections which are as appropriate as possible. Though verified for some reactor types, these generalized cross sections are not as accurate as ones calculated for specific applications. In the case

of this study, this means finding the nodal state parameters for materials in the core of the SAFARI-1 reactor.

Thermal hydraulics codes may be used to find, among a great many other things, state parameters including fuel burnup rates, material temperatures, and material densities at various points in a system. For nuclear reactors, these values can be used to find the nuclear cross sections at the points examined. However, thermal hydraulics codes require intelligent selection of power per fuel cell, which means that the power cannot also be calculated.

The process of coupling a neutronics code and a thermal hydraulics code is essentially akin to taking the strengths of each tool to correct a fundamental weakness in the other. Whereas neutronics codes require a guess at the state parameters to determine cross sections, thermal hydraulics codes can actively find them. Likewise, whereas thermal hydraulics codes require a guess at the power per fuel cell, the neutronics code can actively find the power distribution. Though an initial guess at the unknown values must still be provided, an iterative process allows each code to find its results as usual while effectively using the other code as a highly sophisticated reference function. The initial guesses eventually become negligible, and are replaced with much better values, which in turn allows for more neutronics and thermal hydraulics solutions which are all more accurate than either code would have arrived at alone.

Goals of the NEM/CTF Coupled Code

NEM/CTF is the general name for a coupled code which combines the neutronics of NEM with the thermal hydraulics of CTF. This type of coupling has been created by multiple groups for various purposes, usually requiring alteration or fabrication of completely new components for the given application or for adaptation to specific versions of NEM and/or CTF. The codes can be combined in one of two ways, usually referred to as an online or an offline coupling. An online coupling is the preferred method in terms of ease of use, as it is defined by requiring no human interaction between iterations. An offline coupling requires some human interaction between iterations, but may be more practical to create due to situational factors. Because the versions of NEM and CTF used in this study compile and run exclusively on Linux and Windows operating systems respectively, an offline coupling was required.

The reactor nodalization scheme for the coupled code was not modified from its previous description as presented in Chapters 3 and 4. Thus, the coupling created here will be a fixed coupling, one which maintains a one to one relationship between simulated nodal volumes. Each assembly gets one radial node in both NEM and CTF, and axial nodes are equivalent between codes and based on materials. Despite the fixed coupling for nodes modeled in both NEM and CTF, there are still some nodes modeled by NEM and not CTF, namely the aluminum core box and water encompassing the core. These nodes do not have any components which contribute significantly to power production, so it is acceptable to neglect them in the CTF model used by the coupled code.

In the steady-state coupling procedure, which is the mode used for this study, various feedback mechanisms serve as the primary communication relays between NEM and CTF. At the beginning of one iteration, the neutronics solution found by NEM provides power distributions for the fuel and control and follower assemblies to CTF. CTF provides fuel temperatures, moderator temperatures, and moderator densities to a polynomial interpreter. The interpreter uses a cross section library created with core-averaged state parameters to fabricate a new library based on the parameters from CTF. This library is used by NEM to create the next neutronics solution. This process is repeated until convergence is reached by the system, which will be determined by changes in k_{eff} of less than 10^{-6}, and relative changes in the nodal state parameters and powers of less than 10^{-4}, for three consecutive iterations. This process essentially reproduces transient steady-state time steps by running CTF through a transient time period, running NEM at a single moment, as NEM is independent of time, and then continuing through the desired time period with CTF and NEM iterations.

An independent version of NEM/CTF has been previously verified for use with nuclear reactor core analyses against the Purdue Advanced Reactor Core Simulator (PARCS), a kinetic core simulator module produced by Purdue University. [14] While the general idea of coupling NEM and CTF is shared between that code and the one created for this study, there are no components in common. The only parts of the two coupled systems which have some likenesses, the base codes NEM and CTF, are different versions.

Creation of the NEM/CTF Coupled Code

The general idea of the coupled code was to allow for feedback between the thermal hydraulics code and the neutronics code. From there, it was desirable to maximize the number of feedback values passed between the codes while minimizing the complexity of the system. Available resources included: Windows operating systems, Linux operating systems, remote access to the supercomputing cluster A Scalable Terascale Advanced Resource for Discovery through Computing (CyberSTAR) which runs on Linux, and Fortran compilers for both Windows and Linux.

Since the version of NEM used only compiles on Linux, it was decided that a Linux system would be required during coupled code runs. The outputs that CTF produced were not read properly by the older text editor used on available Linux systems, so it was additionally decided that a Windows system would be required. Thus, the code would consist of a multi-platform system run on Windows with live remote access to CyberSTAR. Since NEM would be run on Linux and CTF would be run on Windows, the code would necessarily be an offline coupling. Possible methods of transition to an online coupling will be addressed in Chapter 6. To make the coupled system as simple as possible, all data analysis and formatting needed between runs of NEM and CTF were to be set up as stand-alone executable files.

NECSA provided an executable which reads from three input files and creates a cross section library. This executable was unique in that it could only be created by NECSA, since it had to be based on experimental SAFARI-1 data, much of which has not been released. This meant that the rest of the coupled code system had to be built in a way that would allow correct interaction with NECSA's executable. The first of the three

required inputs is a full 6-group cross section library for SAFARI-1 designed with core-averaged thermal hydraulic properties of burnup, fuel temperature, moderator temperature, and moderator density, and this input was provided by NECSA based on experimentally verified programs used regularly with the reactor. The second input is a table of the state parameters fuel temperature, moderator temperature, and moderator density, all specified for individual fuel nodes in the core. This input would be created for this study, and will be discussed in the next paragraph of this section. The third input is a polynomial interpreter that reads values from the second input, and was provided by NECSA. The output is a 6-group cross section library which checks each node to detect if there is data for that node in the second input, creates new data using the second and third inputs if there is, and copies the data from the first input if there is not. This executable file is composed of a main program and several other modules, but only the source code for the main component is available for reference here as Appendix E: generatelib.f90.

In order to create the input composed of a state parameter table to be used by generatelib.f90, a code was written in Fortran 90 to read through a CTF output file, extract the desired data, and create a text file consisting of such a table. The source code for this component is available for reference here as Appendix D: TakeCTF.f90.

In order to prepare the cross section library created by generatelib.f90 for use within NEM, a separate executable file was fabricated for reformatting purposes. Additionally, this file rewrites the reformatted library onto an existing NEM input file, which is then ready for immediate use by NEM. The source code for this component is available for reference here as Appendix F: MakeNEM.f90. An example of a resulting NEM input file is available for reference here as Appendix A: NEM 6-Group Input Deck.

Feedback from NEM consists of the radial power distribution for the fuel and control and follower assemblies, as well as for the axial power distributions for the individual assemblies by node. An executable was created to read an NEM output file, which contains the desired power distributions, and reformat it for use with CTF. The same executable goes on to rewrite the reformatted power distributions onto an existing CTF input file, which is then ready for immediate use by CTF. The source code for this component is available for reference here as Appendix C: MakeCTF.f90. An example of a resulting CTF input file is available for reference here as Appendix B: CTF Input Deck.

NEM/CTF Coupled System Procedure

This section will consist of practical instructions for running the coupled code system such that the results from the study documented in this paper may be reproduced by future, interested parties. The current system at the time of this study requires a Windows folder and a Linux folder in which to store input and output files read by the executable files composing the coupled code, as well as a live remote access program so that the files can be transferred between operating systems quickly.

First, MakeCTF.exe is run in Windows. This file uses nemout (the nem output file) and deck.inp (the CTF input file), and alters deck.inp. Next, CTF_Safari.exe (the CTF executable) is run in Windows. This file uses deck.inp and creates 11 output files, including results_channels.out (this file contains the state parameters for each node). Then, TakeCTF.exe is run in Windows. This file uses results_channels.out, and creates adam.INP (the state parameter table for the fuel assemblies).

At this point, adam.INP must be transferred to the Linux folder. After it has been transferred, generatelib is run in Linux. This file uses SAFARI.lib (the library with core-averaged thermal hydraulic properties), adam.INP, and SSFE_300g_element.SPX (the polynomial interpreter), and creates SAFARI_out.lib (the cross section library made with CTF feedback considered). SAFARI_out.lib must then be transferred to the Windows folder.

With the transfer completed, MakeNEM is run in Windows. This file uses SAFARI_out.lib and nemin (the NEM input file), and alters nemin. After that, nemin is transferred to the Linux folder. Having done that, NEM_21.81 (the NEM executable) is run in Linux. This file uses nemin and creates nemout. The final step is to transfer

nemout to the Windows folder. Following the file transfer, an optional archiving process is carried out in which deck.inp, deck.out (a very detailed CTF output file which includes the data comprising results_channels.out), nemin, and nemout are saved to a separate folder such that the intermediate steps before convergence can be examined.

The above steps are repeated until the convergence criteria are met. Presently, each iteration takes approximately 10 minutes, and convergence is reached in 5 iterations. A maximum of 10 iterations for one trial has been run to verify that the convergence reached after 5 iterations was stable, which it was. Thus, the desired convergence criteria was achieved, characterized in this study as a change in k_{eff} of less than 10^{-6}, and relative changes in the nodal state parameters and powers of less than 10^{-4}, for three consecutive iterations. This established completion of the coupled simulation, and the solutions are described in the most recently created deck.out and nemout files.

NEM/CTF Coupled Code Solutions

There are three general aspects of the coupled solution of interest for this study. First, the iterative convergence will be examined through graphical representation of the multiplication factor evolution over the course of the transient simulation runs. Though the nodal state parameters and power distributions were used as convergence criteria as well, it would require an additional 2480 plots to present their change over time in addition to that of the multiplication factor. Thus, the multiplication factor plot alone will suffice out of both convenience and economy of space. Though the plots are not provided here, the calculations required to verify convergence were carried out.

Second, the converged neutronics results will be analyzed in a similar fashion to the neutronics models results from NEM and OSCAR, as exhibited in Chapter 3. Though there is no coupled reference solution, the NEM results, which were verified by the OSCAR results, can be used as a baseline for general comparison. Since the RELAP results were not comprehensive enough for a thorough verification of the CTF results, the thermal hydraulics results from the coupled code will not be presented here. It is sufficient to note that convergence was reached and that the coupled CTF results were not considerably unlike those of the stand-alone model.

Third, we will parallel the development of the neutronics models to error reduction in the multiplication factor with respect to the theoretical case of criticality under steady-state hot full power operating conditions. This will provide quantifiable evidence in support of the endeavor to improve stand-alone models via a coupling process in order to better predict reality. Specific improvements examined will be increasing the number of energy groups in models without a continuous energy spectrum,

using a polynomial interpreter with at least core-averaged state parameters, and incorporating thermal hydraulics simulations for use in coupled calculations.

As discussed earlier, the first aspect of the coupled results considered will be the iterative convergence, which will be examined through graphical representation of the multiplication factor evolution over the course of the transient simulations run. This is shown below in Figure 35.

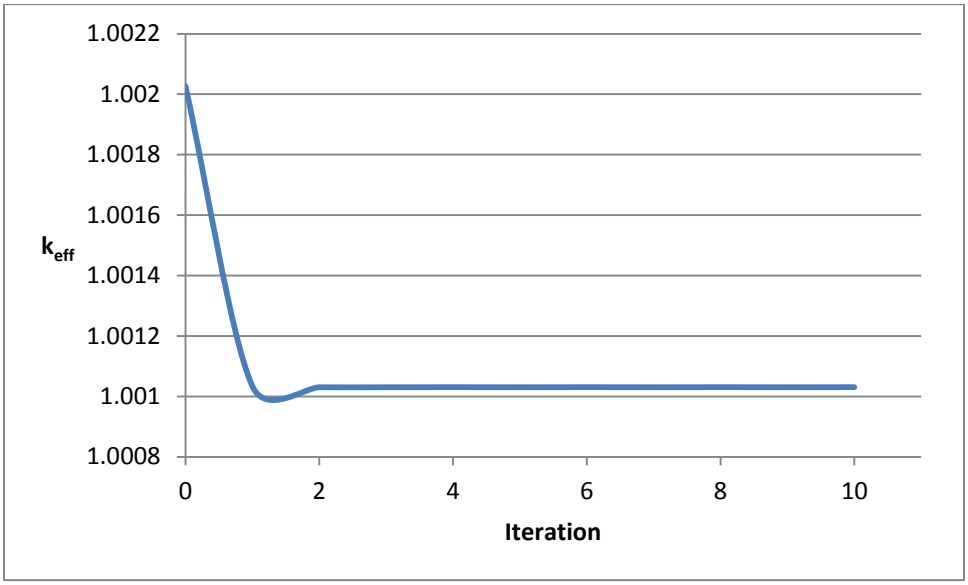

Figure 35: Iterative Convergence of the Multiplication Factor. The multiplication factor labeled as iteration zero is the polynomial interpreter model with core-averaged cross sections. It was used as the baseline neutronics model before the inclusion of thermal hydraulics feedback.

The second aspect of the coupled code results which is of interest is the neutronics solution. As was the case with the NEM and OSCAR results for the various stand-alone models, the full solution of the SAFARI-1 steady-state neutronics as determined by the coupled code is expansive, and likewise presenting it in its entirety would both excessive and detailed perhaps beyond usefulness. Instead, the neutronics values which have been of interest throughout the previous chapters will be analyzed, namely the multiplication factor for the whole reactor and the radial power distribution. These values will be shown for iteration 10. Though convergence was reached at iteration 5, the only negative facet of carrying out further iterations is the time required to carry out the necessary computational calculations, while a positive outcome is that the solution can only either remain the same or become more accurate.

Since the creation of a table is not practical for displaying a lone value, the multiplication factor will simply be stated here. For iteration 10, a multiplication factor of 1.00103062688202260 was observed. Next, the radial power distribution will be examined. Figure 36 presents the radial power distributions in a numerical array which is useful for technical applications, and Figures 37 presents the same distribution in an intuitively easy to grasp 3D format.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1.163	1.147	1.262	0	1.172	0	0	0	0	0
0	0	0	0	0	0	1.147	0.855	1.185	0.942	1.104	0	0	0	0
0	0	0	0	0	1.234	1.086	1.003	0	1.213	0	0	0	0	0
0	0	0	0	0	0	0.991	0.707	1.019	0.823	1.057	0	0	0	0
0	0	0	0	0	1.025	0.938	0.94	0	1.065	0	0	0	0	0
0	0	0	0	0	0	1.051	0.605	0.923	0.886	0.928	0	0	0	0
0	0	0	0	0	0.897	0.891	0.934	0.935	0.873	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 36: Numerical Array of the Coupled Code Radial Power Distribution. Values are for steady-state operating conditions, and have been normalized over the non-zero assemblies. Cells colored in blue are producing the least power, and cells colored in red are producing the most power. View is overhead, and the top of the figure is facing south.

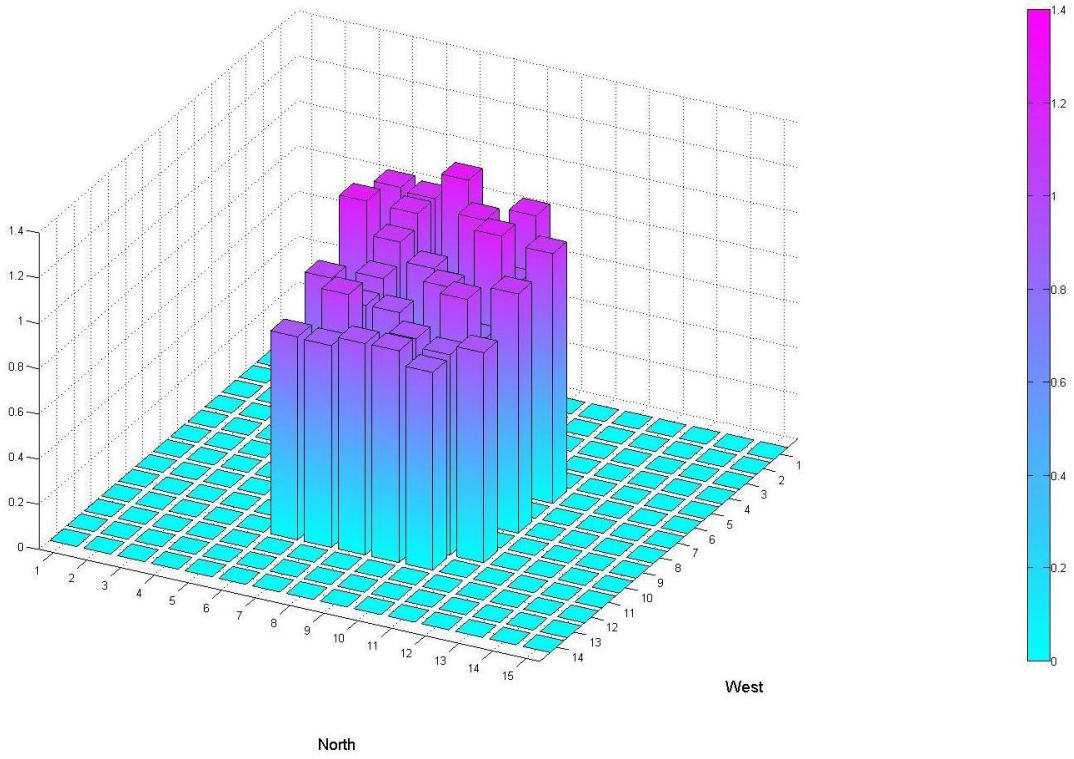

Figure 37: Three Dimensional Visualization of the Coupled Code Radial Power Distribution. Values are for steady-state operating conditions, and have been normalized over the non-zero assemblies. This is an overhead view.

The final aspect of the coupled code results assessed in this study is the development of the neutronics models and subsequent error reduction in the whole reactor multiplication factor with respect to the theoretical case of criticality under steady-state hot full power operating conditions. We will consider the relative error under the following two assumptions. First, we will assume a multiplication factor of exactly 1 for SAFARI-1 at criticality under steady-state operating conditions. This is supported by well-established reactor theory and numerous experiments. Second, we will assume that the 2-group stand-alone model is the least accurate neutronics model. This is supported by the multiplication factor for this model deviating by the largest amount of all the models from the aforementioned theoretical and experimental multiplication factor associated with reactor criticality. The multiplications factors and relative error for all neutronics models created in this study are presented below in Table 12.

Table 12: Multiplication Factors and Relative Error for Neutronics Models.

Model or Other Source	k_{eff}	Relative Error
NEM 2-Group Stand-Alone	1.07360	100.0%
NEM 6-Group Stand-Alone	1.02294	31.17%
NEM 6-Group Polynomial	1.00203	2.758%
Coupled Code Iteration 10	1.00103	1.399%
Theoretical and Experimental	1.00000	0.000%

CHAPTER 6. Conclusions and Future Work

Conclusions

The purpose of this study was to create a highly accurate model of the SAFARI-1 materials testing reactor, which is a facility mainly used for research and for the production of medical isotopes. The model was requested as part of the ongoing SAFARI-1 benchmarking project. Since the task was complex, it was split into several separate tasks. These tasks were the creation of a stand-alone neutronics model, a stand-alone thermal hydraulics model, and a coupled model which combined the two stand-alone models using feedback and convergence criteria.

First, a stand-alone computational neutronics model was created. The code utilized was NEM, which is a deterministic code used by research universities as well as companies in industry. The model of SAFARI-1 was created using cross sections provided by NECSA, and the results were inspected for obvious errors. After troubleshooting, a suitable solution was sent to NECSA for analysis. The results of this component were validated using NECSA's neutronics code OSCAR, which was validated experimentally using the SAFARI-1 MTR itself. On the average, the multiplication factors of the neutronics models agreed to 0.00511%, and the power distributions per radial cell agreed to within 0.0749%. NECSA approved the accuracy of the NEM stand-alone solution based on its agreement with the OSCAR stand-alone solution, thereby completing the first phase of the study.

Second, a stand-alone computational thermal hydraulics model was created. The thermal hydraulics code utilized was CTF, which is another deterministic code, used nearly exclusively by the students and faculty at PSU. CTF is a version of COBRA-TF, a

more widely used thermal hydraulics code. The CTF model of the SAFARI-1 MTR was originally created in collaboration between the RDFMG at PSU and the RRT team employed by NECSA, and the whole of the model was verified for physical accuracy against the SAFARI-1 Benchmark Specification as a part of this study. The model was run and its solutions were analyzed and validated against another thermal hydraulics code RELAP which was validated experimentally. RELAP results were only released to a limited extent and were averaged by assembly type, but fell within the range of values for said assemblies in the more comprehensive CTF solution.

A coupled system to combine NEM and CTF was successfully created using offline methods and a combination of Windows and Linux operating systems. The results found using the coupled NEM/CTF model could not be compared to any other coupled models of SAFARI-1, since it is the first such model. However, k_{eff} was closer to exactly 1 than in any of the stand-alone NEM models, suggesting an overall improvement in the accuracy of the model. The outcome of the study was the intended coupled neutronics/thermal hydraulics model of the SAFARI-1 reactor, with all components having been verified through previously accepted codes and experiments.

Several important inferences can be made from the success and accuracy of the model created in this study. In terms of the SAFARI-1 reactor, NEM/CTF can now be used to simulate transient periods not only in steady-state hot full power mode, but in other modes as well. Without discussing future work extensively, as it will be in the next section of this paper, it can be concluded that NECSA has gained a powerful new tool for safety analysis and fuel management optimization.

This study also serves as a validation for NEM, as it has correctly modeled significant upscattering contributions to flux and power, something not fully accounted for in previous studies with the utilized version of NEM. Though upscattering plays a relatively small role compared to downscattering (downscattering being the more frequently occurring of the two scattering directions), upscattering still needs to be considered and is a valuable phenomenon to be able to include in neutronics simulations that include fission.

Future Work

Future work using the original model and/or coupling system is virtually unlimited due to the state of continuous advancement in the field of nuclear engineering. One weakness of the coupled models created over the course of this study is the uncertainty in the calculated cross sections. The cross sections are based on a finite number of data points determined by a computational system, which are then used by the coupling through polynomial extrapolation. With more data points, the cross sections will become more realistic. This improvement would be directly applied to both the physical structures and nuclear phenomena modeled.

One area of future work relates to applications that would be specifically catered to aiding NECSA. Namely, were NECSA to decide to rearrange the fuel, control and follower, or reflector assemblies, or make any number of other physical changes to the core of the SAFARI-1 MTR, the coupled code could be fairly easily modified to reflect those changes. As far as safety is concerned, the code would probably be most useful as a mechanism for predicting the consequences of making those alterations, allowing reactor staff to avoid potential dangers.

Another area of future work linked to safety would be alteration of the transient models to simulate reactor accidents. By simulating unregulated changes in standard operation, even those already addressed using OSCAR or other codes, the additional information could be used to implement more realistic emergency protocols and precautions.

A broader subject for future work based on the effort and results documented in this study would be to contribute to the overall scientific knowledge of the tank-in-pool

type materials testing reactor. At this time, a large emphasis is being placed on the pursuit of increased efficiency in nuclear power reactors, which of course produce much more consistent monetary profit than academic and other research reactors. The revenue generated by power reactors allows for use of private funding to study those reactors and improve on their designs. Therefore the study of the SAFARI-1 reactor by the RDFMG serves as part of a relatively smaller collection of knowledge, and the results herein provide an excellent opportunity for dissemination of knowledge to interested parties seeking information pertaining to research reactors.

Additionally, an improvement could be made to the coupling system used in this study to increase the accuracy of the final results. The core-averaged library referred to in Chapter 5 as SAFARI.lib defines the core-averaged burnup for each fuel assembly, but the polynomial interpreter SSFE_300g_element.SPX provided by NECSA does not have the required data about SAFARI-1 to incorporate this parameter during the creation of SAFARI_out.lib. If the polynomial interpreter was updated to include burnup, TakeCTF.f90 could be modified to extract the corresponding burnup data and include it in adam.INP, and the executable file source generatelib.f90 (along with the related modules) could be slightly altered for the new functionality.

Yet another possibility for future work is that the current offline coupling system could be changed to an online coupling system. This could be done in one of two ways. First, this could be achieved by converting either the CTF or NEM source file to be compatible with Linux or Windows respectively. Alternately, an executable created using a program with automated file transfer and execution capabilities could be utilized. Such an executable would run through the NEM/CTF coupled system procedure, as outlined in

Chapter 5, incorporating the desired convergence criteria. Either of these two methods could reduce the run time of the NEM/CTF coupled code by eliminating the human interaction time necessary for the present offline coupling. The former of the two options could further reduce run time by eliminating the reading and writing of intermittent input and output files, and replacing that feedback communication type with agreement of variables within the codes.

Finally, it is desired by RDFMG staff that this model be catalogued as a sample input set that may be utilized in conjunction with forthcoming improvements to NEM. Since the NEM results of stand-alone neutronics simulations of the SAFARI-1 MTR in steady-state hot full power mode have been documented in the course of this study, and since they have additionally been confirmed to agree with results from the OSCAR solutions, the inputs and outputs can reasonably be used as references by newer versions of NEM as they are developed. This combination of a reactor simulation with known geometry, known cross sections, and successful computational modeling will likely prove to be a valuable resource for future studies at PSU.

REFERENCES

1. Prinsloo, Rian, Lesego Moloko, Pavel Bokov, Gerhardt Stander, Khalid Elsakhawy, and Suzanne Theron. *SAFARI-1 Benchmark Specification*. South Africa: South African Nuclear Energy Corporation, 2008. Print.
2. Stander, Gerhardt, Rian Prinsloo, Erwin Müller, and Djordje Tomašević. "OSCAR-4 Code System Application to the SAFARI-1 Reactor." *International Conference on Reactor Physics, Nuclear Power: A Sustainable Resource, Casino-Kursaal Conference Center, Interlaken, Switzerland, 14-19 Sep.* 2008. Print.
3. Reactor Dynamics and Fuel Management Group. *NEM Theory Manual*. University Park, PA: Department of Mechanical and Nuclear Engineering, The Pennsylvania State University, 2009. Print.
4. The RELAP5-3D Code Development Team. *RELAP-3D Code Manual Volume I: Code Structure, System Models and Solution Methods*. Idaho Falls, Idaho: Idaho National Laboratory, 2005. Print.
5. Reactor Dynamics and Fuel Management Group. *CTF – A Thermal-Hydraulic Code for LWRs Transient Analysis, Users' Manual*. University Park, PA: Department of Mechanical and Nuclear Engineering, The Pennsylvania State University, 2003. Print.
6. Ditsi, Veronica. "Irradiation Services." *Nuclear Services Online*. South African Nuclear Energy Corporation. Web. 11 Jan. 2012. <http://nuclearservices.radwaste.co.za/safari.htm>.

7. Zakzouk, Mohamed. “The 2009-2010 Medical Isotope Shortage: Cause, Effects and Future Consideration.” *Parliament of Canada*. Publication No. 2009-04E. Revised 17 November 2010. Parliamentary Information and Research Service, Library of Parliament. Web. 11 Jan. 2012. <http://www.parl.gc.ca/Content/LOP/ResearchPublications/2009-04-e.htm>.
8. Maqubela, Tseliso. “Minister of Minerals and Energy announces the phasing out of the use of Highly Enriched Uranium for the Pelindaba Research Reactor Nuclear Fuel.” *South African Government Information*. 18 July 2005. Department of Minerals and Energy. Web. 11 Jan. 2012. <http://www.info.gov.za/speeches/2005/05072012151001.htm>.
9. Prinsloo, Rian. Personal Correspondence. 3 Nov. 2011.
10. Reactor Dynamics and Fuel Management Group. *NEM Standalone Input Manual*. University Park, PA: Department of Mechanical and Nuclear Engineering, The Pennsylvania State University, 2010. Print.
11. Kravec, Matt. *Summary of Work Performed for SAFARI-1 obtained using Nodal Expansion Method (NEM)*. University Park, PA: Department of Mechanical and Nuclear Engineering, The Pennsylvania State University, 2 May 2011. Print.
12. Colameco, David. Personal Correspondence. 28 Oct. 2011.
13. Elsakhawy, Khalid. *SAFARI-1 RELAP Results*. Pelindaba, South Africa: South African Nuclear Energy Corporation, 2012. Print.

14. Gouja, Ilyes, Maria Avramova, and Adam Rubin. "Development and Optimization of Coupling Interfaces Between Reactor Core Neutronics and Thermal Hydraulics Codes." *PHYSOR 2010 – Advances in Reactor Physics to Power the Nuclear Renaissance, Pittsburgh, Pennsylvania, USA, 9-14 May 2010.* Print.

APPENDIX A: NEM 6-Group Input Deck

```
*Card 1
      NDGR     IGEOM        NG    ITRANS      IADF      ITRS    IKAPPA
         6         0         6         0         0         0         0
*Card 4
*    NXNDS     NYNDS     NZNDS   NXSETS      NASMX     NASMY     NASMZ
        15        14        12      2520        15        14        12
*Card 5
*    NXCDS     NYCDS     NZCDS    NXSCDS    NINNER      ILNG       IIN
        15        14        12       210         4         0         0
*Card 6
*   MOUTER      IREB      IAEX       IQL    APTCVG    AAVCVG     AKCVG
       100         0         1         1   1.0E-07   1.0E-07   1.0E-07
*Card 7
*   IUSCAT    MXCMIT     CMEMD     IWEIL   OMEGOR  NRXSETS
         1       100     0.000         0    1.200  0
*Card 8
*   ITUGRP   ITERSUM
         1         2
*Card 9
  0.00000005  0.00000005  0.00000005  0.00000005  0.00000005
0.00000005  0.00000005
*Card 10
*     BETA    LAMBDA
 2.470E-04 1.270E-02
 1.384E-03 3.170E-02
 1.222E-03 1.150E-01
 2.646E-03 3.110E-01
 8.320E-04 1.400E+00
 1.690E-04 3.870E+00

**** Delayed neutrons yields, prompt neutron yields, and cross sections
are included in the code, but are not present here due to length. ****

*Card 22, DELXT
 1r 9.000     1r 6.000     1r 2.500     9r 7.710     1r 2.500     1r
6.000     1r 9.000
*Card 23, DELYT
 1r 9.000     1r 6.000     1r 2.500     8r 8.100     1r 3.500     1r
4.600     1r15.000
*Card 24, DELZT
 2r 7.500     8r 7.421     2r 7.500
*Card 25, NLNDS
         1         1         1         1         1         1         1
1
         1         1         1         1         1         1
*Card 26, NXYNDS
        15        15        15        15        15        15        15
15
        15        15        15        15        15        15
*Card 27, NLCDS
         1         1         1         1         1         1         1
1
         1         1         1         1         1         1
*Card 28, NXYCDS
```

```
        15         15         15         15         15         15         15
15
        15         15         15         15         15         15
*Card 29, NLADS
         1          1          1          1          1          1          1
1
         1          1          1          1          1          1
*Card 30, NXYADS
        15         15         15         15         15         15         15
15
        15         15         15         15         15         15
*Card 31
*     IOBR       IOBL       IOBT       IOBB       IOBU       IOBD
         1          1          1          1          1          1
*Card 32
     1     2     3     4     5     6     7     8     9    10    11
12    13    14    15
    16    17    18    19    20    21    22    23    24    25    26
27    28    29    30
    31    32    33    34    35    36    37    38    39    40    41
42    43    44    45
    46    47    48    49    50    51    52    53    54    55    56
57    58    59    60
    61    62    63    64    65    66    67    68    69    70    71
72    73    74    75
    76    77    78    79    80    81    82    83    84    85    86
87    88    89    90
    91    92    93    94    95    96    97    98    99   100   101
102   103   104   105
   106   107   108   109   110   111   112   113   114   115   116
117   118   119   120
   121   122   123   124   125   126   127   128   129   130   131
132   133   134   135
   136   137   138   139   140   141   142   143   144   145   146
147   148   149   150
   151   152   153   154   155   156   157   158   159   160   161
162   163   164   165
   166   167   168   169   170   171   172   173   174   175   176
177   178   179   180
   181   182   183   184   185   186   187   188   189   190   191
192   193   194   195
   196   197   198   199   200   201   202   203   204   205   206
207   208   209   210
*Card 33
     1    13    25    37    49    61    73    85    97   109   121
133   145   157   169   181   193   205   217   229   241   253   265
277   289   301   313   325   337   349   361   373   385   397   409
421   433   445   457   469   481   493   505   517   529   541   553
565   577   589   601   613   625   637   649   661   673   685   697
709   721   733   745   757   769   781   793   805   817   829   841
853   865   877   889   901   913   925   937   949   961   973   985
997  1009  1021  1033  1045  1057  1069  1081  1093  1105  1117  1129
1141  1153  1165  1177  1189  1201  1213  1225  1237  1249  1261  1273
1285  1297  1309  1321  1333  1345  1357  1369  1381  1393  1405  1417
1429  1441  1453  1465  1477  1489  1501  1513  1525  1537  1549  1561
1573  1585  1597  1609  1621  1633  1645  1657  1669  1681  1693  1705
1717  1729  1741  1753  1765  1777  1789  1801  1813  1825  1837  1849
```

1861 1873 1885 1897 1909 1921 1933 1945 1957 1969 1981 1993
2005 2017 2029 2041 2053 2065 2077 2089 2101 2113 2125 2137
2149 2161 2173 2185 2197 2209 2221 2233 2245 2257 2269 2281
2293 2305 2317 2329 2341 2353 2365 2377 2389 2401 2413 2425
2437 2449 2461 2473 2485 2497 2509

2 14 26 38 50 62 74 86 98 110 122

134 146 158 170 182 194 206 218 230 242 254 266
278 290 302 314 326 338 350 362 374 386 398 410
422 434 446 458 470 482 494 506 518 530 542 554
566 578 590 602 614 626 638 650 662 674 686 698
710 722 734 746 758 770 782 794 806 818 830 842
854 866 878 890 902 914 926 938 950 962 974 986
998 1010 1022 1034 1046 1058 1070 1082 1094 1106 1118 1130
1142 1154 1166 1178 1190 1202 1214 1226 1238 1250 1262 1274
1286 1298 1310 1322 1334 1346 1358 1370 1382 1394 1406 1418
1430 1442 1454 1466 1478 1490 1502 1514 1526 1538 1550 1562
1574 1586 1598 1610 1622 1634 1646 1658 1670 1682 1694 1706
1718 1730 1742 1754 1766 1778 1790 1802 1814 1826 1838 1850
1862 1874 1886 1898 1910 1922 1934 1946 1958 1970 1982 1994
2006 2018 2030 2042 2054 2066 2078 2090 2102 2114 2126 2138
2150 2162 2174 2186 2198 2210 2222 2234 2246 2258 2270 2282
2294 2306 2318 2330 2342 2354 2366 2378 2390 2402 2414 2426
2438 2450 2462 2474 2486 2498 2510

3 15 27 39 51 63 75 87 99 111 123

135 147 159 171 183 195 207 219 231 243 255 267
279 291 303 315 327 339 351 363 375 387 399 411
423 435 447 459 471 483 495 507 519 531 543 555
567 579 591 603 615 627 639 651 663 675 687 699
711 723 735 747 759 771 783 795 807 819 831 843
855 867 879 891 903 915 927 939 951 963 975 987
999 1011 1023 1035 1047 1059 1071 1083 1095 1107 1119 1131
1143 1155 1167 1179 1191 1203 1215 1227 1239 1251 1263 1275
1287 1299 1311 1323 1335 1347 1359 1371 1383 1395 1407 1419
1431 1443 1455 1467 1479 1491 1503 1515 1527 1539 1551 1563
1575 1587 1599 1611 1623 1635 1647 1659 1671 1683 1695 1707
1719 1731 1743 1755 1767 1779 1791 1803 1815 1827 1839 1851
1863 1875 1887 1899 1911 1923 1935 1947 1959 1971 1983 1995
2007 2019 2031 2043 2055 2067 2079 2091 2103 2115 2127 2139
2151 2163 2175 2187 2199 2211 2223 2235 2247 2259 2271 2283
2295 2307 2319 2331 2343 2355 2367 2379 2391 2403 2415 2427
2439 2451 2463 2475 2487 2499 2511

4 16 28 40 52 64 76 88 100 112 124

136 148 160 172 184 196 208 220 232 244 256 268
280 292 304 316 328 340 352 364 376 388 400 412
424 436 448 460 472 484 496 508 520 532 544 556
568 580 592 604 616 628 640 652 664 676 688 700
712 724 736 748 760 772 784 796 808 820 832 844
856 868 880 892 904 916 928 940 952 964 976 988
1000 1012 1024 1036 1048 1060 1072 1084 1096 1108 1120 1132
1144 1156 1168 1180 1192 1204 1216 1228 1240 1252 1264 1276
1288 1300 1312 1324 1336 1348 1360 1372 1384 1396 1408 1420
1432 1444 1456 1468 1480 1492 1504 1516 1528 1540 1552 1564
1576 1588 1600 1612 1624 1636 1648 1660 1672 1684 1696 1708
1720 1732 1744 1756 1768 1780 1792 1804 1816 1828 1840 1852
1864 1876 1888 1900 1912 1924 1936 1948 1960 1972 1984 1996
2008 2020 2032 2044 2056 2068 2080 2092 2104 2116 2128 2140
2152 2164 2176 2188 2200 2212 2224 2236 2248 2260 2272 2284

2296 2308 2320 2332 2344 2356 2368 2380 2392 2404 2416 2428
2440 2452 2464 2476 2488 2500 2512

5 17 29 41 53 65 77 89 101 113 125

137 149 161 173 185 197 209 221 233 245 257 269
281 293 305 317 329 341 353 365 377 389 401 413
425 437 449 461 473 485 497 509 521 533 545 557
569 581 593 605 617 629 641 653 665 677 689 701
713 725 737 749 761 773 785 797 809 821 833 845
857 869 881 893 905 917 929 941 953 965 977 989
1001 1013 1025 1037 1049 1061 1073 1085 1097 1109 1121 1133
1145 1157 1169 1181 1193 1205 1217 1229 1241 1253 1265 1277
1289 1301 1313 1325 1337 1349 1361 1373 1385 1397 1409 1421
1433 1445 1457 1469 1481 1493 1505 1517 1529 1541 1553 1565
1577 1589 1601 1613 1625 1637 1649 1661 1673 1685 1697 1709
1721 1733 1745 1757 1769 1781 1793 1805 1817 1829 1841 1853
1865 1877 1889 1901 1913 1925 1937 1949 1961 1973 1985 1997
2009 2021 2033 2045 2057 2069 2081 2093 2105 2117 2129 2141
2153 2165 2177 2189 2201 2213 2225 2237 2249 2261 2273 2285
2297 2309 2321 2333 2345 2357 2369 2381 2393 2405 2417 2429
2441 2453 2465 2477 2489 2501 2513

6 18 30 42 54 66 78 90 102 114 126

138 150 162 174 186 198 210 222 234 246 258 270
282 294 306 318 330 342 354 366 378 390 402 414
426 438 450 462 474 486 498 510 522 534 546 558
570 582 594 606 618 630 642 654 666 678 690 702
714 726 738 750 762 774 786 798 810 822 834 846
858 870 882 894 906 918 930 942 954 966 978 990
1002 1014 1026 1038 1050 1062 1074 1086 1098 1110 1122 1134
1146 1158 1170 1182 1194 1206 1218 1230 1242 1254 1266 1278
1290 1302 1314 1326 1338 1350 1362 1374 1386 1398 1410 1422
1434 1446 1458 1470 1482 1494 1506 1518 1530 1542 1554 1566
1578 1590 1602 1614 1626 1638 1650 1662 1674 1686 1698 1710
1722 1734 1746 1758 1770 1782 1794 1806 1818 1830 1842 1854
1866 1878 1890 1902 1914 1926 1938 1950 1962 1974 1986 1998
2010 2022 2034 2046 2058 2070 2082 2094 2106 2118 2130 2142
2154 2166 2178 2190 2202 2214 2226 2238 2250 2262 2274 2286
2298 2310 2322 2334 2346 2358 2370 2382 2394 2406 2418 2430
2442 2454 2466 2478 2490 2502 2514

7 19 31 43 55 67 79 91 103 115 127

139 151 163 175 187 199 211 223 235 247 259 271
283 295 307 319 331 343 355 367 379 391 403 415
427 439 451 463 475 487 499 511 523 535 547 559
571 583 595 607 619 631 643 655 667 679 691 703
715 727 739 751 763 775 787 799 811 823 835 847
859 871 883 895 907 919 931 943 955 967 979 991
1003 1015 1027 1039 1051 1063 1075 1087 1099 1111 1123 1135
1147 1159 1171 1183 1195 1207 1219 1231 1243 1255 1267 1279
1291 1303 1315 1327 1339 1351 1363 1375 1387 1399 1411 1423
1435 1447 1459 1471 1483 1495 1507 1519 1531 1543 1555 1567
1579 1591 1603 1615 1627 1639 1651 1663 1675 1687 1699 1711
1723 1735 1747 1759 1771 1783 1795 1807 1819 1831 1843 1855
1867 1879 1891 1903 1915 1927 1939 1951 1963 1975 1987 1999
2011 2023 2035 2047 2059 2071 2083 2095 2107 2119 2131 2143
2155 2167 2179 2191 2203 2215 2227 2239 2251 2263 2275 2287
2299 2311 2323 2335 2347 2359 2371 2383 2395 2407 2419 2431
2443 2455 2467 2479 2491 2503 2515

8	20	32	44	56	68	80	92	104	116	128	
140	152	164	176	188	200	212	224	236	248	260	272
284	296	308	320	332	344	356	368	380	392	404	416
428	440	452	464	476	488	500	512	524	536	548	560
572	584	596	608	620	632	644	656	668	680	692	704
716	728	740	752	764	776	788	800	812	824	836	848
860	872	884	896	908	920	932	944	956	968	980	992
1004	1016	1028	1040	1052	1064	1076	1088	1100	1112	1124	1136
1148	1160	1172	1184	1196	1208	1220	1232	1244	1256	1268	1280
1292	1304	1316	1328	1340	1352	1364	1376	1388	1400	1412	1424
1436	1448	1460	1472	1484	1496	1508	1520	1532	1544	1556	1568
1580	1592	1604	1616	1628	1640	1652	1664	1676	1688	1700	1712
1724	1736	1748	1760	1772	1784	1796	1808	1820	1832	1844	1856
1868	1880	1892	1904	1916	1928	1940	1952	1964	1976	1988	2000
2012	2024	2036	2048	2060	2072	2084	2096	2108	2120	2132	2144
2156	2168	2180	2192	2204	2216	2228	2240	2252	2264	2276	2288
2300	2312	2324	2336	2348	2360	2372	2384	2396	2408	2420	2432
2444	2456	2468	2480	2492	2504	2516					
9	21	33	45	57	69	81	93	105	117	129	
141	153	165	177	189	201	213	225	237	249	261	273
285	297	309	321	333	345	357	369	381	393	405	417
429	441	453	465	477	489	501	513	525	537	549	561
573	585	597	609	621	633	645	657	669	681	693	705
717	729	741	753	765	777	789	801	813	825	837	849
861	873	885	897	909	921	933	945	957	969	981	993
1005	1017	1029	1041	1053	1065	1077	1089	1101	1113	1125	1137
1149	1161	1173	1185	1197	1209	1221	1233	1245	1257	1269	1281
1293	1305	1317	1329	1341	1353	1365	1377	1389	1401	1413	1425
1437	1449	1461	1473	1485	1497	1509	1521	1533	1545	1557	1569
1581	1593	1605	1617	1629	1641	1653	1665	1677	1689	1701	1713
1725	1737	1749	1761	1773	1785	1797	1809	1821	1833	1845	1857
1869	1881	1893	1905	1917	1929	1941	1953	1965	1977	1989	2001
2013	2025	2037	2049	2061	2073	2085	2097	2109	2121	2133	2145
2157	2169	2181	2193	2205	2217	2229	2241	2253	2265	2277	2289
2301	2313	2325	2337	2349	2361	2373	2385	2397	2409	2421	2433
2445	2457	2469	2481	2493	2505	2517					
10	22	34	46	58	70	82	94	106	118	130	
142	154	166	178	190	202	214	226	238	250	262	274
286	298	310	322	334	346	358	370	382	394	406	418
430	442	454	466	478	490	502	514	526	538	550	562
574	586	598	610	622	634	646	658	670	682	694	706
718	730	742	754	766	778	790	802	814	826	838	850
862	874	886	898	910	922	934	946	958	970	982	994
1006	1018	1030	1042	1054	1066	1078	1090	1102	1114	1126	1138
1150	1162	1174	1186	1198	1210	1222	1234	1246	1258	1270	1282
1294	1306	1318	1330	1342	1354	1366	1378	1390	1402	1414	1426
1438	1450	1462	1474	1486	1498	1510	1522	1534	1546	1558	1570
1582	1594	1606	1618	1630	1642	1654	1666	1678	1690	1702	1714
1726	1738	1750	1762	1774	1786	1798	1810	1822	1834	1846	1858
1870	1882	1894	1906	1918	1930	1942	1954	1966	1978	1990	2002
2014	2026	2038	2050	2062	2074	2086	2098	2110	2122	2134	2146
2158	2170	2182	2194	2206	2218	2230	2242	2254	2266	2278	2290
2302	2314	2326	2338	2350	2362	2374	2386	2398	2410	2422	2434
2446	2458	2470	2482	2494	2506	2518					
11	23	35	47	59	71	83	95	107	119	131	
143	155	167	179	191	203	215	227	239	251	263	275
287	299	311	323	335	347	359	371	383	395	407	419

```
431   443   455   467   479   491   503   515   527   539   551   563
575   587   599   611   623   635   647   659   671   683   695   707
719   731   743   755   767   779   791   803   815   827   839   851
863   875   887   899   911   923   935   947   959   971   983   995
1007  1019  1031  1043  1055  1067  1079  1091  1103  1115  1127  1139
1151  1163  1175  1187  1199  1211  1223  1235  1247  1259  1271  1283
1295  1307  1319  1331  1343  1355  1367  1379  1391  1403  1415  1427
1439  1451  1463  1475  1487  1499  1511  1523  1535  1547  1559  1571
1583  1595  1607  1619  1631  1643  1655  1667  1679  1691  1703  1715
1727  1739  1751  1763  1775  1787  1799  1811  1823  1835  1847  1859
1871  1883  1895  1907  1919  1931  1943  1955  1967  1979  1991  2003
2015  2027  2039  2051  2063  2075  2087  2099  2111  2123  2135  2147
2159  2171  2183  2195  2207  2219  2231  2243  2255  2267  2279  2291
2303  2315  2327  2339  2351  2363  2375  2387  2399  2411  2423  2435
2447  2459  2471  2483  2495  2507  2519
    12    24    36    48    60    72    84    96   108   120   132
144   156   168   180   192   204   216   228   240   252   264   276
288   300   312   324   336   348   360   372   384   396   408   420
432   444   456   468   480   492   504   516   528   540   552   564
576   588   600   612   624   636   648   660   672   684   696   708
720   732   744   756   768   780   792   804   816   828   840   852
864   876   888   900   912   924   936   948   960   972   984   996
1008  1020  1032  1044  1056  1068  1080  1092  1104  1116  1128  1140
1152  1164  1176  1188  1200  1212  1224  1236  1248  1260  1272  1284
1296  1308  1320  1332  1344  1356  1368  1380  1392  1404  1416  1428
1440  1452  1464  1476  1488  1500  1512  1524  1536  1548  1560  1572
1584  1596  1608  1620  1632  1644  1656  1668  1680  1692  1704  1716
1728  1740  1752  1764  1776  1788  1800  1812  1824  1836  1848  1860
1872  1884  1896  1908  1920  1932  1944  1956  1968  1980  1992  2004
2016  2028  2040  2052  2064  2076  2088  2100  2112  2124  2136  2148
2160  2172  2184  2196  2208  2220  2232  2244  2256  2268  2280  2292
2304  2316  2328  2340  2352  2364  2376  2388  2400  2412  2424  2436
2448  2460  2472  2484  2496  2508  2520
*Card 34
     1         2         3         4         5         6         7
     8         9        10        11        12        13        14
    15
*Card 35
     1         2         3         4         5         6         7
     8         9        10        11        12        13        14
*Card 36
     1         2         3         4         5         6         7
     8         9        10        11        12
*Card 37
     1         2         3         4         5         6         7
     8         9        10        11        12        13        14
    15
*Card 38
     1         2         3         4         5         6         7
     8         9        10        11        12        13        14
*Card 39
     1         2         3         4         5         6         7
     8         9        10        11        12
```

APPENDIX B: CTF Input Deck

```
*************************************************************************
********
* INPUT DECK
*
* SAFARI-1 Full Core                                                    *
*************************************************************************
********
* CARD INPUT 1
*       ICOBRA
              1
* CARD INPUT 2
*      INITIAL         DUMPF
              1             0
* CARD INPUT34
*         EPSO        OITMAX        IITMAX
          0.001             5            40
* CARD COBRA 1
*<--------------TEXT----------------->
           **SAFARI-1_FA**
*
*************************************************************************
**********
* GROUP 1 - Calculation Variables and Initial Conditions
*
*************************************************************************
**********
* NGR
    1
* Card 1.1
* NGAS IRFC EDMD IMIX ISOL NDM6 NDM7 NDM8 NDM9 NM10 NM11 NM12 NM13 NM14
     1    1    0    0    0    0    0    0    0    0    0    0    0    0
* Card 1.2
*     GTOT     AFLUX    DHFRAC
     788.756    58.6833      .0
*     18.0       58.68     0.0
* Card 1.3
*     PREF       HIN      HGIN    VFRAC1    VFRAC2
       2.5     167.62     124.0       1.0     .9999
* Card 1.3
* GTYPE         VFRAC
  air            .0001
*************************************************************************
**********
* GROUP 2 - Channel Description
*
*************************************************************************
**********
* NGR
    2
* Card 2.1
* NCHA NDM2 NDM3 NDM4 NDM5 NDM6 NDM7 NDM8 NDM9 NM10 NM11 NM12 NM13 NM14
    72   0    0    0    0    0    0    0    0    0    0    0    0    0
* Axial section 1, subchannel 1
* Card 2.2
```

```
* Solid Al channels
* I          AN          PW   ABOT   ATOP  NMGP
  1  0.00018632 0.31146000    0.0    0.0     0
  9  0.00018632 0.31146000    0.0    0.0     0
 10  0.00018632 0.31146000    0.0    0.0     0
 19  0.00018632 0.31146000    0.0    0.0     0
 28  0.00018632 0.31146000    0.0    0.0     0
 45  0.00018632 0.31146000    0.0    0.0     0
 63  0.00018632 0.31146000    0.0    0.0     0
 72  0.00018632 0.31146000    0.0    0.0     0
* Solid Be channels
* I          AN          PW   ABOT   ATOP  NMGP
  2  0.00018632 0.31146000    0.0    0.0     0
  3  0.00018632 0.31146000    0.0    0.0     0
  4  0.00018632 0.31146000    0.0    0.0     0
  5  0.00018632 0.31146000    0.0    0.0     0
  6  0.00018632 0.31146000    0.0    0.0     0
  7  0.00018632 0.31146000    0.0    0.0     0
  8  0.00018632 0.31146000    0.0    0.0     0
 11  0.00018632 0.31146000    0.0    0.0     0
 18  0.00018632 0.31146000    0.0    0.0     0
 20  0.00018632 0.31146000    0.0    0.0     0
 27  0.00018632 0.31146000    0.0    0.0     0
 29  0.00018632 0.31146000    0.0    0.0     0
 36  0.00018632 0.31146000    0.0    0.0     0
 38  0.00018632 0.31146000    0.0    0.0     0
 47  0.00018632 0.31146000    0.0    0.0     0
 54  0.00018632 0.31146000    0.0    0.0     0
 56  0.00018632 0.31146000    0.0    0.0     0
 65  0.00018632 0.31146000    0.0    0.0     0
 71  0.00018632 0.31146000    0.0    0.0     0
* Al water boxes
* I          AN          PW   ABOT   ATOP  NMGP
 15  0.00051031 1.65620000    0.0    0.0     0
 17  0.00051031 1.65620000    0.0    0.0     0
 21  0.00051031 1.65620000    0.0    0.0     0
 33  0.00051031 1.65620000    0.0    0.0     0
 35  0.00051031 1.65620000    0.0    0.0     0
 39  0.00051031 1.65620000    0.0    0.0     0
 51  0.00051031 1.65620000    0.0    0.0     0
 53  0.00051031 1.65620000    0.0    0.0     0
 57  0.00051031 1.65620000    0.0    0.0     0
* Lead Channel
* I          AN          PW   ABOT   ATOP  NMGP
 37  0.00056706 0.31144000    0.0    0.0     0
 46  0.00056706 0.31144000    0.0    0.0     0
 55  0.00056706 0.31144000    0.0    0.0     0
 64  0.00056706 0.31144000    0.0    0.0     0
* Fuel Assemblies
* I          AN          PW   ABOT   ATOP  NMGP
 12 0.00389757  2.82872360    0.0    0.0     0
 13 0.00389757  2.82872360    0.0    0.0     0
 14 0.00389757  2.82872360    0.0    0.0     0
 16 0.00389757  2.82872360    0.0    0.0     0
 22 0.00389757  2.82872360    0.0    0.0     0
 24 0.00389757  2.82872360    0.0    0.0     0
 26 0.00389757  2.82872360    0.0    0.0     0
```

```
 30 0.00389757  2.82872360    0.0    0.0     0
 31 0.00389757  2.82872360    0.0    0.0     0
 32 0.00389757  2.82872360    0.0    0.0     0
 34 0.00389757  2.82872360    0.0    0.0     0
 40 0.00389757  2.82872360    0.0    0.0     0
 42 0.00389757  2.82872360    0.0    0.0     0
 44 0.00389757  2.82872360    0.0    0.0     0
 48 0.00389757  2.82872360    0.0    0.0     0
 49 0.00389757  2.82872360    0.0    0.0     0
 50 0.00389757  2.82872360    0.0    0.0     0
 52 0.00389757  2.82872360    0.0    0.0     0
 58 0.00389757  2.82872360    0.0    0.0     0
 60 0.00389757  2.82872360    0.0    0.0     0
 62 0.00389757  2.82872360    0.0    0.0     0
 66 0.00389757  2.82872360    0.0    0.0     0
 67 0.00389757  2.82872360    0.0    0.0     0
 68 0.00389757  2.82872360    0.0    0.0     0
 69 0.00389757  2.82872360    0.0    0.0     0
 70 0.00389757  2.82872360    0.0    0.0     0
* Control Fuel Assemblies
* I          AN          PW    ABOT    ATOP  NMGP
 23 0.00365336  2.32561200    0.0    0.0     0
 25 0.00365336  2.32561200    0.0    0.0     0
 41 0.00365336  2.32561200    0.0    0.0     0
 43 0.00365336  2.32561200    0.0    0.0     0
 59 0.00365336  2.32561200    0.0    0.0     0
 61 0.00365336  2.32561200    0.0    0.0     0
************************************************************************
********
* Group 3 - Transverse Channel Connection (Gap) Data
************************************************************************
********

**** Transverse Channel Connection (Gap) Data is included in the code,
but is not present here due to length. ****

************************************************************************
**********
* GROUP 4 - Vertical Channel Connection Data
*
************************************************************************
**********
* NGR
    4
* Card 4.1
* NSEC NSIM IREB NDM4 NDM5 NDM6 NDM7 NDM8 NDM9 NM10 NM11 NM12 NM13 NM14
     1    1    0    0    0    0    0    0    0    0    0    0    0    0
* Axial section 1
* Card 4.2
* ISEC NCHN NONODE       DXS    IVAR
     1   72   10        0.05936   0
* Card 4.4
*   I  KCHA  KCHA  KCHA  KCHA  KCHA  KCHA  KCHB  KCHB  KCHB  KCHB  KCHB
KCHB
    1     1     0     0     0     0     0     1     0     0     0     0
0
```

2	2	0	0	0	0	0	2	0	0	0	0
3	3	0	0	0	0	0	3	0	0	0	0
4	4	0	0	0	0	0	4	0	0	0	0
5	5	0	0	0	0	0	5	0	0	0	0
6	6	0	0	0	0	0	6	0	0	0	0
7	7	0	0	0	0	0	7	0	0	0	0
8	8	0	0	0	0	0	8	0	0	0	0
9	9	0	0	0	0	0	9	0	0	0	0
10	10	0	0	0	0	0	10	0	0	0	0
11	11	0	0	0	0	0	11	0	0	0	0
12	12	0	0	0	0	0	12	0	0	0	0
13	13	0	0	0	0	0	13	0	0	0	0
14	14	0	0	0	0	0	14	0	0	0	0
15	15	0	0	0	0	0	15	0	0	0	0
16	16	0	0	0	0	0	16	0	0	0	0
17	17	0	0	0	0	0	17	0	0	0	0
18	18	0	0	0	0	0	18	0	0	0	0
19	19	0	0	0	0	0	19	0	0	0	0
20	20	0	0	0	0	0	20	0	0	0	0
21	21	0	0	0	0	0	21	0	0	0	0
22	22	0	0	0	0	0	22	0	0	0	0
23	23	0	0	0	0	0	23	0	0	0	0
24	24	0	0	0	0	0	24	0	0	0	0
25	25	0	0	0	0	0	25	0	0	0	0
26	26	0	0	0	0	0	26	0	0	0	0
27	27	0	0	0	0	0	27	0	0	0	0
28	28	0	0	0	0	0	28	0	0	0	0
29	29	0	0	0	0	0	29	0	0	0	0

30	30	0	0	0	0	0	30	0	0	0	0
31	31	0	0	0	0	0	31	0	0	0	0
32	32	0	0	0	0	0	32	0	0	0	0
33	33	0	0	0	0	0	33	0	0	0	0
34	34	0	0	0	0	0	34	0	0	0	0
35	35	0	0	0	0	0	35	0	0	0	0
36	36	0	0	0	0	0	36	0	0	0	0
37	37	0	0	0	0	0	37	0	0	0	0
38	38	0	0	0	0	0	38	0	0	0	0
39	39	0	0	0	0	0	39	0	0	0	0
40	40	0	0	0	0	0	40	0	0	0	0
41	41	0	0	0	0	0	41	0	0	0	0
42	42	0	0	0	0	0	42	0	0	0	0
43	43	0	0	0	0	0	43	0	0	0	0
44	44	0	0	0	0	0	44	0	0	0	0
45	45	0	0	0	0	0	45	0	0	0	0
46	46	0	0	0	0	0	46	0	0	0	0
47	47	0	0	0	0	0	47	0	0	0	0
48	48	0	0	0	0	0	48	0	0	0	0
49	49	0	0	0	0	0	49	0	0	0	0
50	50	0	0	0	0	0	50	0	0	0	0
51	51	0	0	0	0	0	51	0	0	0	0
52	52	0	0	0	0	0	52	0	0	0	0
53	53	0	0	0	0	0	53	0	0	0	0
54	54	0	0	0	0	0	54	0	0	0	0
55	55	0	0	0	0	0	55	0	0	0	0
56	56	0	0	0	0	0	56	0	0	0	0
57	57	0	0	0	0	0	57	0	0	0	0

```
   58    58     0     0     0     0     0    58     0     0     0     0
0
   59    59     0     0     0     0     0    59     0     0     0     0
0
   60    60     0     0     0     0     0    60     0     0     0     0
0
   61    61     0     0     0     0     0    61     0     0     0     0
0
   62    62     0     0     0     0     0    62     0     0     0     0
0
   63    63     0     0     0     0     0    63     0     0     0     0
0
   64    64     0     0     0     0     0    64     0     0     0     0
0
   65    65     0     0     0     0     0    65     0     0     0     0
0
   66    66     0     0     0     0     0    66     0     0     0     0
0
   67    67     0     0     0     0     0    67     0     0     0     0
0
   68    68     0     0     0     0     0    68     0     0     0     0
0
   69    69     0     0     0     0     0    69     0     0     0     0
0
   70    70     0     0     0     0     0    70     0     0     0     0
0
   71    71     0     0     0     0     0    71     0     0     0     0
0
   72    72     0     0     0     0     0    72     0     0     0     0
0
* Card 4.5
* IWDE
    72
* Card 4.6
* MSIM
   720
*
**********************************************************************
**********
* GROUP 8 - Rod and Unheated Conductor Data
*
**********************************************************************
**********
* NGR
    8
* Card 8.1
* NRRD NSRD   NC NRTB NRAD NLTY NSTA  NXF NCAN RADF   W3 NM12 NM13 NM14
    32   72    1    1    0    0    0    0    0    0    0    0    0    0
* Group 8.2
*    N   IFTY IAXP NRND  DAXMIN   RMULT     HGAP ISECR   HTAMB    TAMB
     1     1    1    0  0.00000    19.0       0.0    1     0.0     0.0
*Group 8.3
*NSCH  PIE NSCH  PIE NSCH  PIE NSCH  PIE  NSCH   PIE  NSCH   PIE  NSCH
PIE  NSCH   PIE
   12  1.0    0   0.0   0  0.0    0  0.0     0   0.0     0   0.0     0
0.0     0   0.0
    2    1    2    0 0.00000     19.0       0.0    1     0.0     0.0
```

```
   13  1.0    0   0.0   0  0.0    0  0.0     0   0.0     0   0.0      0
0.0     0   0.0
    3    1    3    0 0.00000     19.0       0.0    1     0.0      0.0
   14  1.0    0   0.0   0  0.0    0  0.0     0   0.0     0   0.0      0
0.0     0   0.0
    4    1    4    0 0.00000     19.0       0.0    1     0.0      0.0
   16  1.0    0   0.0   0  0.0    0  0.0     0   0.0     0   0.0      0
0.0     0   0.0
    5    1    5    0 0.00000     19.0       0.0    1     0.0      0.0
   22  1.0    0   0.0   0  0.0    0  0.0     0   0.0     0   0.0      0
0.0     0   0.0
    6    1    7    0 0.00000     19.0       0.0    1     0.0      0.0
   24  1.0    0   0.0   0  0.0    0  0.0     0   0.0     0   0.0      0
0.0     0   0.0
    7    1    9    0 0.00000     19.0       0.0    1     0.0      0.0
   26  1.0    0   0.0   0  0.0    0  0.0     0   0.0     0   0.0      0
0.0     0   0.0
    8    1   10   0 0.00000     19.0       0.0    1     0.0      0.0
   30  1.0    0   0.0   0  0.0    0  0.0     0   0.0     0   0.0      0
0.0     0   0.0
    9    1   11   0 0.00000     19.0       0.0    1     0.0      0.0
   31  1.0    0   0.0   0  0.0    0  0.0     0   0.0     0   0.0      0
0.0     0   0.0
   10    1   12   0 0.00000     19.0       0.0    1     0.0      0.0
   32  1.0    0   0.0   0  0.0    0  0.0     0   0.0     0   0.0      0
0.0     0   0.0
   11    1   13   0 0.00000     19.0       0.0    1     0.0      0.0
   34  1.0    0   0.0   0  0.0    0  0.0     0   0.0     0   0.0      0
0.0     0   0.0
   12    1   14   0 0.00000     19.0       0.0    1     0.0      0.0
   40  1.0    0   0.0   0  0.0    0  0.0     0   0.0     0   0.0      0
0.0     0   0.0
   13    1   16   0 0.00000     19.0       0.0    1     0.0      0.0
   42  1.0    0   0.0   0  0.0    0  0.0     0   0.0     0   0.0      0
0.0     0   0.0
   14    1   18   0 0.00000     19.0       0.0    1     0.0      0.0
   44  1.0    0   0.0   0  0.0    0  0.0     0   0.0     0   0.0      0
0.0     0   0.0
   15    1   19   0 0.00000     19.0       0.0    1     0.0      0.0
   48  1.0    0   0.0   0  0.0    0  0.0     0   0.0     0   0.0      0
0.0     0   0.0
   16    1   20   0 0.00000     19.0       0.0    1     0.0      0.0
   49  1.0    0   0.0   0  0.0    0  0.0     0   0.0     0   0.0      0
0.0     0   0.0
   17    1   21   0 0.00000     19.0       0.0    1     0.0      0.0
   50  1.0    0   0.0   0  0.0    0  0.0     0   0.0     0   0.0      0
0.0     0   0.0
   18    1   22   0 0.00000     19.0       0.0    1     0.0      0.0
   52  1.0    0   0.0   0  0.0    0  0.0     0   0.0     0   0.0      0
0.0     0   0.0
   19    1   23   0 0.00000     19.0       0.0    1     0.0      0.0
   58  1.0    0   0.0   0  0.0    0  0.0     0   0.0     0   0.0      0
0.0     0   0.0
   20    1   25   0 0.00000     19.0       0.0    1     0.0      0.0
   60  1.0    0   0.0   0  0.0    0  0.0     0   0.0     0   0.0      0
0.0     0   0.0
   21    1   27   0 0.00000     19.0       0.0    1     0.0      0.0
```

```
   62  1.0    0   0.0   0  0.0    0  0.0     0   0.0     0   0.0      0
0.0     0   0.0
   22    1    28   0 0.00000     19.0       0.0    1      0.0      0.0
   66  1.0    0   0.0   0  0.0    0  0.0     0   0.0     0   0.0      0
0.0     0   0.0
   23    1    29   0 0.00000     19.0       0.0    1      0.0      0.0
   67  1.0    0   0.0   0  0.0    0  0.0     0   0.0     0   0.0      0
0.0     0   0.0
   24    1    30   0 0.00000     19.0       0.0    1      0.0      0.0
   68  1.0    0   0.0   0  0.0    0  0.0     0   0.0     0   0.0      0
0.0     0   0.0
   25    1    31   0 0.00000     19.0       0.0    1      0.0      0.0
   69  1.0    0   0.0   0  0.0    0  0.0     0   0.0     0   0.0      0
0.0     0   0.0
   26    1    32   0 0.00000     19.0       0.0    1      0.0      0.0
   70  1.0    0   0.0   0  0.0    0  0.0     0   0.0     0   0.0      0
0.0     0   0.0
* Control rod assemblies
   27    2    6    0 0.00000     15.0       0.0    1      0.0      0.0
   23  1.0    0   0.0   0  0.0    0  0.0     0   0.0     0   0.0      0
0.0     0   0.0
   28    2    8    0 0.00000     15.0       0.0    1      0.0      0.0
   25  1.0    0   0.0   0  0.0    0  0.0     0   0.0     0   0.0      0
0.0     0   0.0
   29    2    15   0 0.00000     15.0       0.0    1      0.0      0.0
   41  1.0    0   0.0   0  0.0    0  0.0     0   0.0     0   0.0      0
0.0     0   0.0
   30    2    17   0 0.00000     15.0       0.0    1      0.0      0.0
   43  1.0    0   0.0   0  0.0    0  0.0     0   0.0     0   0.0      0
0.0     0   0.0
   31    2    24   0 0.00000     15.0       0.0    1      0.0      0.0
   59  1.0    0   0.0   0  0.0    0  0.0     0   0.0     0   0.0      0
0.0     0   0.0
   32    2    26   0 0.00000     15.0       0.0    1      0.0      0.0
   61  1.0    0   0.0   0  0.0    0  0.0     0   0.0     0   0.0      0
0.0     0   0.0
* Card 8.5
*    N ISTY      HPERIM   HPERIMI     RMULT NOCH NSCH      HTMB TAMB
* Al solid assembly
     1    3     0.31146        0.        1.    1    0        0.   0.
     2    3     0.31146        0.        1.    9    0        0.   0.
     3    3     0.31146        0.        1.   10    0        0.   0.
     4    3     0.31146        0.        1.   19    0        0.   0.
     5    3     0.31146        0.        1.   28    0        0.   0.
     6    3     0.31146        0.        1.   45    0        0.   0.
     7    3     0.31146        0.        1.   63    0        0.   0.
     8    3     0.31146        0.        1.   72    0        0.   0.
* Be solid assemblies
     9    4     0.31146        0.        1.    2    0        0.   0.
    10    4     0.31146        0.        1.    3    0        0.   0.
    11    4     0.31146        0.        1.    4    0        0.   0.
    12    4     0.31146        0.        1.    5    0        0.   0.
    13    4     0.31146        0.        1.    6    0        0.   0.
    14    4     0.31146        0.        1.    7    0        0.   0.
    15    4     0.31146        0.        1.    8    0        0.   0.
    16    4     0.31146        0.        1.   11    0        0.   0.
    17    4     0.31146        0.        1.   18    0        0.   0.
```

```
    18    4     0.31146        0.        1.   20    0        0.   0.
    19    4     0.31146        0.        1.   27    0        0.   0.
    20    4     0.31146        0.        1.   29    0        0.   0.
    21    4     0.31146        0.        1.   36    0        0.   0.
    22    4     0.31146        0.        1.   38    0        0.   0.
    23    4     0.31146        0.        1.   47    0        0.   0.
    24    4     0.31146        0.        1.   54    0        0.   0.
    25    4     0.31146        0.        1.   56    0        0.   0.
    26    4     0.31146        0.        1.   65    0        0.   0.
    27    4     0.31146        0.        1.   71    0        0.   0.
* Lead solid assemblies
    28    5     0.31144        0.        1.   37    0        0.   0.
    29    5     0.31144        0.        1.   46    0        0.   0.
    30    5     0.31144        0.        1.   55    0        0.   0.
    31    5     0.31144        0.        1.   64    0        0.   0.
* Al water boxes
    32    6     1.65620        0.        1.   15    0        0.   0.
    33    6     1.65620        0.        1.   17    0        0.   0.
    34    6     1.65620        0.        1.   21    0        0.   0.
    35    6     1.65620        0.        1.   33    0        0.   0.
    36    6     1.65620        0.        1.   35    0        0.   0.
    37    6     1.65620        0.        1.   39    0        0.   0.
    38    6     1.65620        0.        1.   51    0        0.   0.
    39    6     1.65620        0.        1.   53    0        0.   0.
    40    6     1.65620        0.        1.   57    0        0.   0.
* Al side plates in FA
    41    7  0.57912360        0.        1.   12    0        0.   0.
    42    7  0.57912360        0.        1.   13    0        0.   0.
    43    7  0.57912360        0.        1.   14    0        0.   0.
    44    7  0.57912360        0.        1.   16    0        0.   0.
    45    7  0.57912360        0.        1.   22    0        0.   0.
    46    7  0.57912360        0.        1.   24    0        0.   0.
    47    7  0.57912360        0.        1.   26    0        0.   0.
    48    7  0.57912360        0.        1.   30    0        0.   0.
    49    7  0.57912360        0.        1.   31    0        0.   0.
    50    7  0.57912360        0.        1.   32    0        0.   0.
    51    7  0.57912360        0.        1.   34    0        0.   0.
    52    7  0.57912360        0.        1.   40    0        0.   0.
    53    7  0.57912360        0.        1.   42    0        0.   0.
    54    7  0.57912360        0.        1.   44    0        0.   0.
    55    7  0.57912360        0.        1.   48    0        0.   0.
    56    7  0.57912360        0.        1.   49    0        0.   0.
    57    7  0.57912360        0.        1.   50    0        0.   0.
    58    7  0.57912360        0.        1.   52    0        0.   0.
    59    7  0.57912360        0.        1.   58    0        0.   0.
    60    7  0.57912360        0.        1.   60    0        0.   0.
    61    7  0.57912360        0.        1.   62    0        0.   0.
    62    7  0.57912360        0.        1.   66    0        0.   0.
    63    7  0.57912360        0.        1.   67    0        0.   0.
    64    7  0.57912360        0.        1.   68    0        0.   0.
    65    7  0.57912360        0.        1.   69    0        0.   0.
    66    7  0.57912360        0.        1.   70    0        0.   0.
* Al side plates in control FA
    67    8  0.68912000        0.        1.   12    0        0.   0.
    68    8  0.68912000        0.        1.   12    0        0.   0.
    69    8  0.68912000        0.        1.   12    0        0.   0.
    70    8  0.68912000        0.        1.   12    0        0.   0.
```

```
    71    8  0.68912000        0.        1.   12    0        0.   0.
    72    8  0.68912000        0.        1.   12    0        0.   0.
* Card 8.6
*    I NRT1 NST1 NRX1
     1   32   72    12
* Card 8.7
* IRTB IRTB IRTB IRTB IRTB IRTB IRTB IRTB IRTB IRTB IRTB IRTB
     1    2    3    4    5    6    7    8    9   10   11   12
    13   14   15   16   17   18   19   20   21   22   23   24
    25   26   27   28   29   30   31   32
* Card 8.8
* ISTB ISTB ISTB ISTB ISTB ISTB ISTB ISTB ISTB ISTB ISTB ISTB
     1    2    3    4    5    6    7    8    9   10   11   12
    13   14   15   16   17   18   19   20   21   22   23   24
    25   26   27   28   29   30   31   32   33   34   35   36
    37   38   39   40   41   42   43   44   45   46   47   48
    49   50   51   52   53   54   55   56   57   58   59   60
    61   62   63   64   65   66   67   68   69   70   71   72
* Card 8.9: Initial temperature profile for rods and unheated
conductors
*   AXIALT    TRINIT
       0.0      40.0
       0.05937  40.0
       0.11874  40.0
       0.17811  40.0
       0.23748  40.0
       0.29685  40.0
       0.35622  40.0
       0.41559  40.0
       0.47496  40.0
       0.53433  40.0
       0.5433  40.0
       0.5936   45.0
*********************************************************************
**********
* GROUP 9 - Conductor Geometry Description
*
*********************************************************************
**********
* NGR
    9
* Card 9.1
* NFLT IRLF ICNF IMWR NDM5 NDM6 NDM7 NDM8 NDM9 NM10 NM11 NM12 NM13 NM14
     8    0    0    0    0    0    0    0    0    0    0    0    0    0
* Card 9.6 and 9.7 continue
*
* Fuel plates in FA
* I FTYP      DROD    DFUL     NFUL  IMTF IMTC IMOX DCORE  TCLD
FTDS  IGPC IGFC IRDP
*  1 nucl     0.126  0.0001295    5     1    4    4     0   0.000768
0.95   0     0   0
*   1 nucl     0.126  0.0001295    2     0    0    0     0   0.000768
0.95   0     0   0
*
* I FTYP      DROD    DIN      NFUL IMTF IMTC NDM8 NDM9 NM10 NM11  NM12
NM13 NM14
```

```
  1 wall     0.126  0.0001295   2    1    1    0    0    0    0     0
0    0
* NODR MATR      TREG      QREG
   2    1    0.00006475       .50
   2    1    0.00006475       .50
*
* Card 9.6 and 9.7 continue
*
* Fuel plates in control FA
* I FTYP      DROD    DFUL   NFUL IMTF IMTC IMOX DCORE   TCLD     FTDS
IGPC IGFC IRDP
*  2 nucl    0.126  0.000508   5     1    4    4    0    0.000768
0.95  0     0    0
*
*    I FTYP      DROD       DIN NFUL IMTF IMTC NDM8 NDM9 NM10 NM11 NM12
NM13 NM14
     2 wall    0.126  0.000508   3    1    1    0    0    0    0    0
0    0
* NODR MATR      TREG      QREG
    2    1  0.000169333       1.0
    2    1  0.000169333       1.0
    2    1  0.000169333       1.0
*
* Unheated structures in Al solid assembly
*    I FTYP      DROD       DIN NFUL IMTF IMTC NDM8 NDM9 NM10 NM11 NM12
NM13 NM14
     3 wall 0.3114600   0.07582    1    4    4    0    0    0    0    0
0    0
*      3 wall 0.3114600   0.07582    1    0    0    0    0    0    0
0    0    0
* NODR MATR      TREG      QREG
     4    4  0.07582       0.0
*
* Unheated structures in Be assemblies
*    I FTYP      DROD       DIN NFUL IMTF IMTC NDM8 NDM9 NM10 NM11 NM12
NM13 NM14
     4 wall 0.3114600   0.07582    1    2    2    0    0    0    0    0
0    0
* NODR MATR      TREG      QREG
     4    2   0.07582       0.0
*
* Unheated structures in Pb assemblies
*    I FTYP      DROD       DIN NFUL IMTF IMTC NDM8 NDM9 NM10 NM11 NM12
NM13 NM14
     5 wall 0.3114400   0.07581    1    3    3    0    0    0    0    0
0    0
* NODR MATR      TREG      QREG
     4    3   0.07581       0.0
*
* Unheated structures in Al water boxes
*    I FTYP      DROD       DIN NFUL IMTF IMTC NDM8 NDM9 NM10 NM11 NM12
NM13 NM14
     6 wall 1.6562000   0.03790    1    4    4    0    0    0    0    0
0    0
* NODR MATR      TREG      QREG
     2    4   0.03790       0.0
* Unheated structures in FA
```

```
*    I FTYP      DROD       DIN NFUL IMTF IMTC NDM8 NDM9 NM10 NM11 NM12
NM13 NM14
     7 wall 0.5791236   0.00455  1      4    4    0    0    0    0    0
0    0
*     7 wall 0.5791236   0.00455  1      0    0    0    0    0    0
0    0    0
* NODR MATR      TREG      QREG
     2    4   0.00455       0.0
*
* Unheated structures in control FA
*     I FTYP      DROD       DIN NFUL IMTF IMTC NDM8 NDM9 NM10 NM11
NM12 NM13 NM14
      8 wall 0.6891200   0.00479    1    4    4    0    0    0    0
0    0    0
*     8 wall 0.6891200   0.00479    1    0    0    0    0    0    0
0    0    0
* NODR MATR      TREG      QREG
     2    4   0.00479       0.0
*
**********************************************************************
**********
* GROUP 10 - Material Properties Tables
*
**********************************************************************
**********
* NGR
   10
* Card 10.1
* NMAT NDM2 NDM3 NDM4 NDM5 NDM6 NDM7 NDM8 NDM9 NM10 NM11 NM12 NM13 NM14
     4    0    0    0    0    0    0    0    0    0    0    0    0    0
* Card 10.2
*    N NTDP     RCOLD                         IMATAN
     1    2     12200.0                         U3si2-al
* Card 10.3
*     TPROP      CPF1      THCF     TPROP      CPF1      THCF
       60.0     0.599      59.0     500      0.731       129.0
*    N NTDP     RCOLD                         IMATAN
     2    1    1850.0                         Be
* Card 10.3
*     TPROP      CPF1      THCF     TPROP      CPF1      THCF
       40.0      1.83     218.0
*
*    N NTDP     RCOLD                         IMATAN
     3    1   11400.0                         Lead
* Card 10.3
*     TPROP      CPF1      THCF     TPROP      CPF1      THCF
       40.0      0.13      35.0
* card 10.2
*     N  NTDP     RCOLD                          IMATAN
     4  2       2699.0                           AL
* card 10.3
*      TPROP      CPF1      THCF     TPROP      CPF1      THCF
      27.0       0.873     273.0    127.0     0.941      240.0
*
**********************************************************************
**********
```

```
* GROUP 11 - Axial Power Tables and Forcing Functions
*
**********************************************************************
**********
* NGR
   11
* Card 11.1
* NQA NAXP MNXN   NQ NGPF  NQR NDM7 NDM8 NDM9 NM10 NM11 NM12 NM13 NM14
    1  32   13    0    0    1    0    0    0    0    0    0    0    0
*
*--- Axial Power Forcing Functions ---
* Card 11.2
*      YQA
        0.0
```

**** Axial Power Forcing Functions are included in the code, but are not present here due to length. ****

```
*
*--- Radial Power Forcing Functions ---
* Card 11.6
*      YQR
       0.0

* Card 11.7
*    FQR(1)    FQR(2)   FQR(3)    FQR(4)    FQR(5)    FQR(6)     FQR(7)
FQR(8)
     1.208     1.159   1.300      1.223     1.264     0.979      1.267
0.950
     1.103     1.292   1.142      1.021     1.163     1.061      0.785
1.037
     0.877     0.957   1.092      0.937     0.943     1.004      0.969
0.665
     0.895     0.804   0.813      0.816     0.766     0.818      0.794
0.763
*      1.0        1.0     1.0        1.0       1.0        1.0        1.0
1.0
*      1.0        1.0     1.0        1.0       1.0        1.0        1.0
1.0
*      1.0        1.0     1.0        1.0       1.0        1.0        1.0
1.0
*      1.0        1.0     1.0        1.0       1.0        1.0        1.0
1.0
**********************************************************************
**********
* GROUP 13 - Boundary Condition Data
*
**********************************************************************
**********
* NGR
   13
* Card 13.1
* NBND NKBD NFUN NGBD NDM5 NDM6 NDM7 NDM8 NDM9 NM10 NM11 NM12 NM13 NM14
   144    0    0    0    0    0    0    0    0    0    0    0    0    0
* Card 13.4
*----------- Inlet b.c. ------------------------------------------------
* IBD1 IBD2 ISPC N1FN N2FN N3FN  BCVALUE1  BCVALUE2  BCVALUE3   INITGAS
```

1	12	2	0	0	0	0.0	167.61	0.0	1
2	12	2	0	0	0	0.0	167.61	0.0	1
3	12	2	0	0	0	0.0	167.61	0.0	1
4	12	2	0	0	0	0.0	167.61	0.0	1
5	12	2	0	0	0	0.0	167.61	0.0	1
6	12	2	0	0	0	0.0	167.61	0.0	1
7	12	2	0	0	0	0.0	167.61	0.0	1
8	12	2	0	0	0	0.0	167.61	0.0	1
9	12	2	0	0	0	0.0	167.61	0.0	1
10	12	2	0	0	0	0.0	167.61	0.0	1
11	12	2	0	0	0	0.0	167.61	0.0	1
12	12	2	0	0	0	0.0	167.61	0.0	1
13	12	2	0	0	0	0.0	167.61	0.0	1
14	12	2	0	0	0	0.0	167.61	0.0	1
15	12	2	0	0	0	0.0	167.61	0.0	1
16	12	2	0	0	0	0.0	167.61	0.0	1
17	12	2	0	0	0	0.0	167.61	0.0	1
18	12	2	0	0	0	0.0	167.61	0.0	1
19	12	2	0	0	0	0.0	167.61	0.0	1
20	12	2	0	0	0	0.0	167.61	0.0	1
21	12	2	0	0	0	0.0	167.61	0.0	1
22	12	2	0	0	0	0.0	167.61	0.0	1
23	12	2	0	0	0	0.0	167.61	0.0	1
24	12	2	0	0	0	0.0	167.61	0.0	1
25	12	2	0	0	0	0.0	167.61	0.0	1
26	12	2	0	0	0	0.0	167.61	0.0	1
27	12	2	0	0	0	0.0	167.61	0.0	1
28	12	2	0	0	0	0.0	167.61	0.0	1
29	12	2	0	0	0	0.0	167.61	0.0	1
30	12	2	0	0	0	0.0	167.61	0.0	1
31	12	2	0	0	0	0.0	167.61	0.0	1
32	12	2	0	0	0	0.0	167.61	0.0	1
33	12	2	0	0	0	0.0	167.61	0.0	1
34	12	2	0	0	0	0.0	167.61	0.0	1
35	12	2	0	0	0	0.0	167.61	0.0	1
36	12	2	0	0	0	0.0	167.61	0.0	1
37	12	2	0	0	0	0.0	167.61	0.0	1
38	12	2	0	0	0	0.0	167.61	0.0	1
39	12	2	0	0	0	0.0	167.61	0.0	1
40	12	2	0	0	0	0.0	167.61	0.0	1
41	12	2	0	0	0	0.0	167.61	0.0	1
42	12	2	0	0	0	0.0	167.61	0.0	1
43	12	2	0	0	0	0.0	167.61	0.0	1
44	12	2	0	0	0	0.0	167.61	0.0	1
45	12	2	0	0	0	0.0	167.61	0.0	1
46	12	2	0	0	0	0.0	167.61	0.0	1
47	12	2	0	0	0	0.0	167.61	0.0	1
48	12	2	0	0	0	0.0	167.61	0.0	1
49	12	2	0	0	0	0.0	167.61	0.0	1
50	12	2	0	0	0	0.0	167.61	0.0	1
51	12	2	0	0	0	0.0	167.61	0.0	1
52	12	2	0	0	0	0.0	167.61	0.0	1
53	12	2	0	0	0	0.0	167.61	0.0	1
54	12	2	0	0	0	0.0	167.61	0.0	1
55	12	2	0	0	0	0.0	167.61	0.0	1
56	12	2	0	0	0	0.0	167.61	0.0	1
57	12	2	0	0	0	0.0	167.61	0.0	1

58	12	2	0	0	0	0.0	167.61	0.0	1
59	12	2	0	0	0	0.0	167.61	0.0	1
60	12	2	0	0	0	0.0	167.61	0.0	1
61	12	2	0	0	0	0.0	167.61	0.0	1
62	12	2	0	0	0	0.0	167.61	0.0	1
63	12	2	0	0	0	0.0	167.61	0.0	1
64	12	2	0	0	0	0.0	167.61	0.0	1
65	12	2	0	0	0	0.0	167.61	0.0	1
66	12	2	0	0	0	0.0	167.61	0.0	1
67	12	2	0	0	0	0.0	167.61	0.0	1
68	12	2	0	0	0	0.0	167.61	0.0	1
69	12	2	0	0	0	0.0	167.61	0.0	1
70	12	2	0	0	0	0.0	167.61	0.0	1
71	12	2	0	0	0	0.0	167.61	0.0	1
72	12	2	0	0	0	0.0	167.61	0.0	1

*---------- Outlet b.c. ----------

* IBD1	IBD2	ISPC	N1FN	N2FN	N3FN	BCVALUE1	BCVALUE2	BCVALUE3	INITGAS
1	1	1	0	0	0	0.0	198.96	2.0	1
2	1	1	0	0	0	0.0	198.96	2.0	1
3	1	1	0	0	0	0.0	198.96	2.0	1
4	1	1	0	0	0	0.0	198.96	2.0	1
5	1	1	0	0	0	0.0	198.96	2.0	1
6	1	1	0	0	0	0.0	198.96	2.0	1
7	1	1	0	0	0	0.0	198.96	2.0	1
8	1	1	0	0	0	0.0	198.96	2.0	1
9	1	1	0	0	0	0.0	198.96	2.0	1
10	1	1	0	0	0	0.0	198.96	2.0	1
11	1	1	0	0	0	0.0	198.96	2.0	1
12	1	1	0	0	0	0.0	198.96	2.0	1
13	1	1	0	0	0	0.0	198.96	2.0	1
14	1	1	0	0	0	0.0	198.96	2.0	1
15	1	1	0	0	0	0.0	198.96	2.0	1
16	1	1	0	0	0	0.0	198.96	2.0	1
17	1	1	0	0	0	0.0	198.96	2.0	1
18	1	1	0	0	0	0.0	198.96	2.0	1
19	1	1	0	0	0	0.0	198.96	2.0	1
20	1	1	0	0	0	0.0	198.96	2.0	1
21	1	1	0	0	0	0.0	198.96	2.0	1
22	1	1	0	0	0	0.0	198.96	2.0	1
23	1	1	0	0	0	0.0	198.96	2.0	1
24	1	1	0	0	0	0.0	198.96	2.0	1
25	1	1	0	0	0	0.0	198.96	2.0	1
26	1	1	0	0	0	0.0	198.96	2.0	1
27	1	1	0	0	0	0.0	198.96	2.0	1
28	1	1	0	0	0	0.0	198.96	2.0	1
29	1	1	0	0	0	0.0	198.96	2.0	1
30	1	1	0	0	0	0.0	198.96	2.0	1
31	1	1	0	0	0	0.0	198.96	2.0	1
32	1	1	0	0	0	0.0	198.96	2.0	1
33	1	1	0	0	0	0.0	198.96	2.0	1
34	1	1	0	0	0	0.0	198.96	2.0	1
35	1	1	0	0	0	0.0	198.96	2.0	1
36	1	1	0	0	0	0.0	198.96	2.0	1
37	1	1	0	0	0	0.0	198.96	2.0	1
38	1	1	0	0	0	0.0	198.96	2.0	1
39	1	1	0	0	0	0.0	198.96	2.0	1
40	1	1	0	0	0	0.0	198.96	2.0	1

```
    41    1    1    0    0    0        0.0    198.96        2.0          1
    42    1    1    0    0    0        0.0    198.96        2.0          1
    43    1    1    0    0    0        0.0    198.96        2.0          1
    44    1    1    0    0    0        0.0    198.96        2.0          1
    45    1    1    0    0    0        0.0    198.96        2.0          1
    46    1    1    0    0    0        0.0    198.96        2.0          1
    47    1    1    0    0    0        0.0    198.96        2.0          1
    48    1    1    0    0    0        0.0    198.96        2.0          1
    49    1    1    0    0    0        0.0    198.96        2.0          1
    50    1    1    0    0    0        0.0    198.96        2.0          1
    51    1    1    0    0    0        0.0    198.96        2.0          1
    52    1    1    0    0    0        0.0    198.96        2.0          1
    53    1    1    0    0    0        0.0    198.96        2.0          1
    54    1    1    0    0    0        0.0    198.96        2.0          1
    55    1    1    0    0    0        0.0    198.96        2.0          1
    56    1    1    0    0    0        0.0    198.96        2.0          1
    57    1    1    0    0    0        0.0    198.96        2.0          1
    58    1    1    0    0    0        0.0    198.96        2.0          1
    59    1    1    0    0    0        0.0    198.96        2.0          1
    60    1    1    0    0    0        0.0    198.96        2.0          1
    61    1    1    0    0    0        0.0    198.96        2.0          1
    62    1    1    0    0    0        0.0    198.96        2.0          1
    63    1    1    0    0    0        0.0    198.96        2.0          1
    64    1    1    0    0    0        0.0    198.96        2.0          1
    65    1    1    0    0    0        0.0    198.96        2.0          1
    66    1    1    0    0    0        0.0    198.96        2.0          1
    67    1    1    0    0    0        0.0    198.96        2.0          1
    68    1    1    0    0    0        0.0    198.96        2.0          1
    69    1    1    0    0    0        0.0    198.96        2.0          1
    70    1    1    0    0    0        0.0    198.96        2.0          1
    71    1    1    0    0    0        0.0    198.96        2.0          1
    72    1    1    0    0    0        0.0    198.96        2.0          1
********************************************** *****
**********************
* Group 14 - Output Options
*
************************************************************************
**********
* NGR
   14
* Card 14.1
*  N1 NOU1 NOU2 NOU3 NOU4 IPRP IOPT NDM8 NDM9 NM10 NM11 NM12 NM13 NM14
    5    0    0    0    0    0    0    0    0    0    0    0    0    0
************************************************************************
**********
* Group 15 - TIME DOMAIN DATA
*
************************************************************************
**********
* CARD GROUP 15
* NGR
   15
* Card 15.1
*        DTMIN          DTMAX           TEND    EDINT   DMPINT
RTWFP
   0.0000000001         0.001           3.0     30.0    10.0
1.0
```

```
*   0.00000001           0.01           100.0   100.0   20.0
0.0
* Card 15.2
*        DTMIN (if negative stop)
         -1.0            0.0           0.0      0.0      0.0
0.0
*         -0.0001          0.0           0.0      0.0      0.0
0.0
*
**********************************************************************
*********
* END GROUP TIME DOMAIN DATA
**********************************************************************
*********
```

APPENDIX C: MakeCTF.f90

```
!-----------------------------------------------------------------------
      PROGRAM MakeCTF
!-----------------------------------------------------------------------
! This program reads in the data from the nemout and generates
! output that can be copied and pasted into a CTF input file
!
! This program was written by Adam Rosenkrantz
!
! February 11, 2012
!



!Read the power distribution data into memory

!-----------------------------------------------------------------------

!c=power channel #, a=layer number, b=radial layer

double precision
dummy,c1a1,c2a1,c3a1,c4a1,c5a1,c6a1,c7a1,c8a1,c9a1,c10a1,c11a1,c12a1,c1
3a1,c14a1,c15a1,c16a1,c17a1,c18a1,c19a1,c20a1,c21a1,c22a1,c23a1,c24a1,c
25a1,c26a1,c27a1,c28a1,c29a1,c30a1,c31a1,c32a1
double precision
c1a2,c2a2,c3a2,c4a2,c5a2,c6a2,c7a2,c8a2,c9a2,c10a2,c11a2,c12a2,c13a2,c1
4a2,c15a2,c16a2,c17a2,c18a2,c19a2,c20a2,c21a2,c22a2,c23a2,c24a2,c25a2,c
26a2,c27a2,c28a2,c29a2,c30a2,c31a2,c32a2
double precision
c1a3,c2a3,c3a3,c4a3,c5a3,c6a3,c7a3,c8a3,c9a3,c10a3,c11a3,c12a3,c13a3,c1
4a3,c15a3,c16a3,c17a3,c18a3,c19a3,c20a3,c21a3,c22a3,c23a3,c24a3,c25a3,c
26a3,c27a3,c28a3,c29a3,c30a3,c31a3,c32a3
double precision
c1a4,c2a4,c3a4,c4a4,c5a4,c6a4,c7a4,c8a4,c9a4,c10a4,c11a4,c12a4,c13a4,c1
4a4,c15a4,c16a4,c17a4,c18a4,c19a4,c20a4,c21a4,c22a4,c23a4,c24a4,c25a4,c
26a4,c27a4,c28a4,c29a4,c30a4,c31a4,c32a4
double precision
c1a5,c2a5,c3a5,c4a5,c5a5,c6a5,c7a5,c8a5,c9a5,c10a5,c11a5,c12a5,c13a5,c1
4a5,c15a5,c16a5,c17a5,c18a5,c19a5,c20a5,c21a5,c22a5,c23a5,c24a5,c25a5,c
26a5,c27a5,c28a5,c29a5,c30a5,c31a5,c32a5
double precision
c1a6,c2a6,c3a6,c4a6,c5a6,c6a6,c7a6,c8a6,c9a6,c10a6,c11a6,c12a6,c13a6,c1
4a6,c15a6,c16a6,c17a6,c18a6,c19a6,c20a6,c21a6,c22a6,c23a6,c24a6,c25a6,c
26a6,c27a6,c28a6,c29a6,c30a6,c31a6,c32a6
double precision
c1a7,c2a7,c3a7,c4a7,c5a7,c6a7,c7a7,c8a7,c9a7,c10a7,c11a7,c12a7,c13a7,c1
4a7,c15a7,c16a7,c17a7,c18a7,c19a7,c20a7,c21a7,c22a7,c23a7,c24a7,c25a7,c
26a7,c27a7,c28a7,c29a7,c30a7,c31a7,c32a7
double precision
c1a8,c2a8,c3a8,c4a8,c5a8,c6a8,c7a8,c8a8,c9a8,c10a8,c11a8,c12a8,c13a8,c1
4a8,c15a8,c16a8,c17a8,c18a8,c19a8,c20a8,c21a8,c22a8,c23a8,c24a8,c25a8,c
26a8,c27a8,c28a8,c29a8,c30a8,c31a8,c32a8
double precision
c1a9,c2a9,c3a9,c4a9,c5a9,c6a9,c7a9,c8a9,c9a9,c10a9,c11a9,c12a9,c13a9,c1
```

```
4a9,c15a9,c16a9,c17a9,c18a9,c19a9,c20a9,c21a9,c22a9,c23a9,c24a9,c25a9,c
26a9,c27a9,c28a9,c29a9,c30a9,c31a9,c32a9
double precision
c1a10,c2a10,c3a10,c4a10,c5a10,c6a10,c7a10,c8a10,c9a10,c10a10,c11a10,c12
a10,c13a10,c14a10,c15a10,c16a10,c17a10,c18a10,c19a10,c20a10,c21a10,c22a
10,c23a10,c24a10,c25a10,c26a10,c27a10,c28a10,c29a10,c30a10,c31a10,c32a1
0

double precision
c1b,c2b,c3b,c4b,c5b,c6b,c7b,c8b,c9b,c10b,c11b,c12b,c13b,c14b,c15b,c16b,
c17b,c18b,c19b,c20b,c21b,c22b,c23b,c24b,c25b,c26b,c27b,c28b,c29b,c30b,c
31b,c32b

integer i

open(20,file="nemout",status="UNKNOWN")

! skip to the beginning of the nodal power distributions

do i=1,769248 ! skip to the beginning of the nodal power distributions
      read(20,*)
end do

i=1

! read in the nodal distributions

!Layer c1

read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c28a1,c29a1,c30a1,c31a1,c32a1
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,c23a1,c24a1,c25a1,c26a1,c27a1
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c19a1,c20a1,c21a1,dummy,c22a1
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,c14a1,c15a1,c16a1,c17a1,c18a1
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c10a1,c11a1,c12a1,dummy,c13a1
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,c5a1,c6a1,c7a1,c8a1,c9a1
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c1a1,c2a1,c3a1,dummy,c4a1

do i=1,22
      read(20,*)
end do

i=1

!Layer c2

read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c28a2,c29a2,c30a2,c31a2,c32a2
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,c23a2,c24a2,c25a2,c26a2,c27a2
```

```
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c19a2,c20a2,c21a2,dummy,c22a2
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,c14a2,c15a2,c16a2,c17a2,c18a2
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c10a2,c11a2,c12a2,dummy,c13a2
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,c5a2,c6a2,c7a2,c8a2,c9a2
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c1a2,c2a2,c3a2,dummy,c4a2

do i=1,22
      read(20,*)
end do

i=1

!Layer c3

read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c28a3,c29a3,c30a3,c31a3,c32a3
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,c23a3,c24a3,c25a3,c26a3,c27a3
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c19a3,c20a3,c21a3,dummy,c22a3
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,c14a3,c15a3,c16a3,c17a3,c18a3
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c10a3,c11a3,c12a3,dummy,c13a3
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,c5a3,c6a3,c7a3,c8a3,c9a3
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c1a3,c2a3,c3a3,dummy,c4a3

do i=1,22
      read(20,*)
end do

i=1

!Layer c4

read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c28a4,c29a4,c30a4,c31a4,c32a4
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,c23a4,c24a4,c25a4,c26a4,c27a4
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c19a4,c20a4,c21a4,dummy,c22a4
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,c14a4,c15a4,c16a4,c17a4,c18a4
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c10a4,c11a4,c12a4,dummy,c13a4
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,c5a4,c6a4,c7a4,c8a4,c9a4
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c1a4,c2a4,c3a4,dummy,c4a4
```

```
do i=1,22
      read(20,*)
end do

i=1

!Layer c5

read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c28a5,c29a5,c30a5,c31a5,c32a5
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,c23a5,c24a5,c25a5,c26a5,c27a5
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c19a5,c20a5,c21a5,dummy,c22a5
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,c14a5,c15a5,c16a5,c17a5,c18a5
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c10a5,c11a5,c12a5,dummy,c13a5
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,c5a5,c6a5,c7a5,c8a5,c9a5
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c1a5,c2a5,c3a5,dummy,c4a5

do i=1,22
      read(20,*)
end do

i=1

!Layer c6

read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c28a6,c29a6,c30a6,c31a6,c32a6
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,c23a6,c24a6,c25a6,c26a6,c27a6
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c19a6,c20a6,c21a6,dummy,c22a6
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,c14a6,c15a6,c16a6,c17a6,c18a6
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c10a6,c11a6,c12a6,dummy,c13a6
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,c5a6,c6a6,c7a6,c8a6,c9a6
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c1a6,c2a6,c3a6,dummy,c4a6

do i=1,22
      read(20,*)
end do

i=1

!Layer c7

read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c28a7,c29a7,c30a7,c31a7,c32a7
```

```
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,c23a7,c24a7,c25a7,c26a7,c27a7
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c19a7,c20a7,c21a7,dummy,c22a7
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,c14a7,c15a7,c16a7,c17a7,c18a7
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c10a7,c11a7,c12a7,dummy,c13a7
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,c5a7,c6a7,c7a7,c8a7,c9a7
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c1a7,c2a7,c3a7,dummy,c4a7

do i=1,22
      read(20,*)
end do

i=1

!Layer c8

read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c28a8,c29a8,c30a8,c31a8,c32a8
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,c23a8,c24a8,c25a8,c26a8,c27a8
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c19a8,c20a8,c21a8,dummy,c22a8
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,c14a8,c15a8,c16a8,c17a8,c18a8
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c10a8,c11a8,c12a8,dummy,c13a8
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,c5a8,c6a8,c7a8,c8a8,c9a8
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c1a8,c2a8,c3a8,dummy,c4a8

do i=1,22
      read(20,*)
end do

i=1

!Layer c9

read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c28a9,c29a9,c30a9,c31a9,c32a9
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,c23a9,c24a9,c25a9,c26a9,c27a9
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c19a9,c20a9,c21a9,dummy,c22a9
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,c14a9,c15a9,c16a9,c17a9,c18a9
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c10a9,c11a9,c12a9,dummy,c13a9
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,c5a9,c6a9,c7a9,c8a9,c9a9
```

```
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c1a9,c2a9,c3a9,dummy,c4a9

do i=1,22
      read(20,*)
end do

i=1

!Layer c10

read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c28a10,c29a10,c30a10,c31a10,c32a10
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,c23a10,c24a10,c25a10,c26a10,c
27a10
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c19a10,c20a10,c21a10,dummy,c22a10
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,c14a10,c15a10,c16a10,c17a10,c
18a10
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c10a10,c11a10,c12a10,dummy,c13a10
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,c5a10,c6a10,c7a10,c8a10,c9a10
read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,c1a10,c2a10,c3a10,dummy,c4a10

do i=1,86
      read(20,*)
end do

i=1

!Radial Layer > b

read(20,*) dummy,dummy,dummy,dummy,dummy,c28b,c29b,c30b,c31b,c32b
read(20,*) dummy,dummy,dummy,dummy,dummy,dummy,c23b,c24b,c25b,c26b,c27b
read(20,*) dummy,dummy,dummy,dummy,dummy,c19b,c20b,c21b,dummy,c22b
read(20,*) dummy,dummy,dummy,dummy,dummy,dummy,c14b,c15b,c16b,c17b,c18b
read(20,*) dummy,dummy,dummy,dummy,dummy,c10b,c11b,c12b,dummy,c13b
read(20,*) dummy,dummy,dummy,dummy,dummy,dummy,c5b,c6b,c7b,c8b,c9b
read(20,*) dummy,dummy,dummy,dummy,dummy,c1b,c2b,c3b,dummy,c4b

close(20)

!Write the power distribution into a static text file

!------------------------------------------------------------------------

open(40,file="deck.inp",status="UNKNOWN")

do i=1,797
    read(40,*)
end do

i=1
```

```
write(40,*) " 1    11   "
write(40,*) " 0.0000        0.0000"
write(40,100) "  0.0900",c1a1
write(40,100) "  0.1500",c1a2
write(40,100) "  0.2242",c1a3
write(40,100) "  0.2984",c1a4
write(40,100) "  0.3726",c1a5
write(40,100) "  0.4468",c1a6
write(40,100) "  0.5210",c1a7
write(40,100) "  0.5952",c1a8
write(40,100) "  0.6694",c1a9
write(40,100) "  0.7436",c1a10

write(40,*) " 2    11   "
write(40,*) " 0.0000        0.0000"
write(40,100) "  0.0900",c2a1
write(40,100) "  0.1500",c2a2
write(40,100) "  0.2242",c2a3
write(40,100) "  0.2984",c2a4
write(40,100) "  0.3726",c2a5
write(40,100) "  0.4468",c2a6
write(40,100) "  0.5210",c2a7
write(40,100) "  0.5952",c2a8
write(40,100) "  0.6694",c2a9
write(40,100) "  0.7436",c2a10

write(40,*) " 3    11   "
write(40,*) " 0.0000        0.0000"
write(40,100) "  0.0900",c3a1
write(40,100) "  0.1500",c3a2
write(40,100) "  0.2242",c3a3
write(40,100) "  0.2984",c3a4
write(40,100) "  0.3726",c3a5
write(40,100) "  0.4468",c3a6
write(40,100) "  0.5210",c3a7
write(40,100) "  0.5952",c3a8
write(40,100) "  0.6694",c3a9
write(40,100) "  0.7436",c3a10

write(40,*) " 4    11   "
write(40,*) " 0.0000        0.0000"
write(40,100) "  0.0900",c4a1
write(40,100) "  0.1500",c4a2
write(40,100) "  0.2242",c4a3
write(40,100) "  0.2984",c4a4
write(40,100) "  0.3726",c4a5
write(40,100) "  0.4468",c4a6
write(40,100) "  0.5210",c4a7
write(40,100) "  0.5952",c4a8
write(40,100) "  0.6694",c4a9
write(40,100) "  0.7436",c4a10

write(40,*) " 5    11   "
write(40,*) " 0.0000        0.0000"
write(40,100) "  0.0900",c5a1
write(40,100) "  0.1500",c5a2
```

```
write(40,100) "  0.2242",c5a3
write(40,100) "  0.2984",c5a4
write(40,100) "  0.3726",c5a5
write(40,100) "  0.4468",c5a6
write(40,100) "  0.5210",c5a7
write(40,100) "  0.5952",c5a8
write(40,100) "  0.6694",c5a9
write(40,100) "  0.7436",c5a10

write(40,*) " 6    11   "
write(40,*) " 0.0000        0.0000"
write(40,100) "  0.0900",c6a1
write(40,100) "  0.1500",c6a2
write(40,100) "  0.2242",c6a3
write(40,100) "  0.2984",c6a4
write(40,100) "  0.3726",c6a5
write(40,100) "  0.4468",c6a6
write(40,100) "  0.5210",c6a7
write(40,100) "  0.5952",c6a8
write(40,100) "  0.6694",c6a9
write(40,100) "  0.7436",c6a10

write(40,*) " 7    11   "
write(40,*) " 0.0000        0.0000"
write(40,100) "  0.0900",c7a1
write(40,100) "  0.1500",c7a2
write(40,100) "  0.2242",c7a3
write(40,100) "  0.2984",c7a4
write(40,100) "  0.3726",c7a5
write(40,100) "  0.4468",c7a6
write(40,100) "  0.5210",c7a7
write(40,100) "  0.5952",c7a8
write(40,100) "  0.6694",c7a9
write(40,100) "  0.7436",c7a10

write(40,*) " 8    11   "
write(40,*) " 0.0000        0.0000"
write(40,100) "  0.0900",c8a1
write(40,100) "  0.1500",c8a2
write(40,100) "  0.2242",c8a3
write(40,100) "  0.2984",c8a4
write(40,100) "  0.3726",c8a5
write(40,100) "  0.4468",c8a6
write(40,100) "  0.5210",c8a7
write(40,100) "  0.5952",c8a8
write(40,100) "  0.6694",c8a9
write(40,100) "  0.7436",c8a10

write(40,*) " 9    11   "
write(40,*) " 0.0000        0.0000"
write(40,100) "  0.0900",c9a1
write(40,100) "  0.1500",c9a2
write(40,100) "  0.2242",c9a3
write(40,100) "  0.2984",c9a4
write(40,100) "  0.3726",c9a5
write(40,100) "  0.4468",c9a6
write(40,100) "  0.5210",c9a7
```

```
write(40,100) "  0.5952",c9a8
write(40,100) "  0.6694",c9a9
write(40,100) "  0.7436",c9a10

write(40,*) " 10    11  "
write(40,*) " 0.0000        0.0000"
write(40,100) "  0.0900",c10a1
write(40,100) "  0.1500",c10a2
write(40,100) "  0.2242",c10a3
write(40,100) "  0.2984",c10a4
write(40,100) "  0.3726",c10a5
write(40,100) "  0.4468",c10a6
write(40,100) "  0.5210",c10a7
write(40,100) "  0.5952",c10a8
write(40,100) "  0.6694",c10a9
write(40,100) "  0.7436",c10a10

write(40,*) " 11    11  "
write(40,*) " 0.0000        0.0000"
write(40,100) "  0.0900",c11a1
write(40,100) "  0.1500",c11a2
write(40,100) "  0.2242",c11a3
write(40,100) "  0.2984",c11a4
write(40,100) "  0.3726",c11a5
write(40,100) "  0.4468",c11a6
write(40,100) "  0.5210",c11a7
write(40,100) "  0.5952",c11a8
write(40,100) "  0.6694",c11a9
write(40,100) "  0.7436",c11a10

write(40,*) " 12    11  "
write(40,*) " 0.0000        0.0000"
write(40,100) "  0.0900",c12a1
write(40,100) "  0.1500",c12a2
write(40,100) "  0.2242",c12a3
write(40,100) "  0.2984",c12a4
write(40,100) "  0.3726",c12a5
write(40,100) "  0.4468",c12a6
write(40,100) "  0.5210",c12a7
write(40,100) "  0.5952",c12a8
write(40,100) "  0.6694",c12a9
write(40,100) "  0.7436",c12a10

write(40,*) " 13    11  "
write(40,*) " 0.0000        0.0000"
write(40,100) "  0.0900",c13a1
write(40,100) "  0.1500",c13a2
write(40,100) "  0.2242",c13a3
write(40,100) "  0.2984",c13a4
write(40,100) "  0.3726",c13a5
write(40,100) "  0.4468",c13a6
write(40,100) "  0.5210",c13a7
write(40,100) "  0.5952",c13a8
write(40,100) "  0.6694",c13a9
write(40,100) "  0.7436",c13a10

write(40,*) " 14    11  "
```

```
write(40,*) " 0.0000        0.0000"
write(40,100) "  0.0900",c14a1
write(40,100) "  0.1500",c14a2
write(40,100) "  0.2242",c14a3
write(40,100) "  0.2984",c14a4
write(40,100) "  0.3726",c14a5
write(40,100) "  0.4468",c14a6
write(40,100) "  0.5210",c14a7
write(40,100) "  0.5952",c14a8
write(40,100) "  0.6694",c14a9
write(40,100) "  0.7436",c14a10

write(40,*) " 15    11  "
write(40,*) " 0.0000        0.0000"
write(40,100) "  0.0900",c15a1
write(40,100) "  0.1500",c15a2
write(40,100) "  0.2242",c15a3
write(40,100) "  0.2984",c15a4
write(40,100) "  0.3726",c15a5
write(40,100) "  0.4468",c15a6
write(40,100) "  0.5210",c15a7
write(40,100) "  0.5952",c15a8
write(40,100) "  0.6694",c15a9
write(40,100) "  0.7436",c15a10

write(40,*) " 16    11  "
write(40,*) " 0.0000        0.0000"
write(40,100) "  0.0900",c16a1
write(40,100) "  0.1500",c16a2
write(40,100) "  0.2242",c16a3
write(40,100) "  0.2984",c16a4
write(40,100) "  0.3726",c16a5
write(40,100) "  0.4468",c16a6
write(40,100) "  0.5210",c16a7
write(40,100) "  0.5952",c16a8
write(40,100) "  0.6694",c16a9
write(40,100) "  0.7436",c16a10

write(40,*) " 17    11  "
write(40,*) " 0.0000        0.0000"
write(40,100) "  0.0900",c17a1
write(40,100) "  0.1500",c17a2
write(40,100) "  0.2242",c17a3
write(40,100) "  0.2984",c17a4
write(40,100) "  0.3726",c17a5
write(40,100) "  0.4468",c17a6
write(40,100) "  0.5210",c17a7
write(40,100) "  0.5952",c17a8
write(40,100) "  0.6694",c17a9
write(40,100) "  0.7436",c17a10

write(40,*) " 18    11  "
write(40,*) " 0.0000        0.0000"
write(40,100) "  0.0900",c18a1
write(40,100) "  0.1500",c18a2
write(40,100) "  0.2242",c18a3
write(40,100) "  0.2984",c18a4
```

```
write(40,100) "  0.3726",c18a5
write(40,100) "  0.4468",c18a6
write(40,100) "  0.5210",c18a7
write(40,100) "  0.5952",c18a8
write(40,100) "  0.6694",c18a9
write(40,100) "  0.7436",c18a10

write(40,*) " 19    11  "
write(40,*) " 0.0000        0.0000"
write(40,100) "  0.0900",c19a1
write(40,100) "  0.1500",c19a2
write(40,100) "  0.2242",c19a3
write(40,100) "  0.2984",c19a4
write(40,100) "  0.3726",c19a5
write(40,100) "  0.4468",c19a6
write(40,100) "  0.5210",c19a7
write(40,100) "  0.5952",c19a8
write(40,100) "  0.6694",c19a9
write(40,100) "  0.7436",c19a10

write(40,*) " 20    11  "
write(40,*) " 0.0000        0.0000"
write(40,100) "  0.0900",c20a1
write(40,100) "  0.1500",c20a2
write(40,100) "  0.2242",c20a3
write(40,100) "  0.2984",c20a4
write(40,100) "  0.3726",c20a5
write(40,100) "  0.4468",c20a6
write(40,100) "  0.5210",c20a7
write(40,100) "  0.5952",c20a8
write(40,100) "  0.6694",c20a9
write(40,100) "  0.7436",c20a10

write(40,*) " 21    11  "
write(40,*) " 0.0000        0.0000"
write(40,100) "  0.0900",c21a1
write(40,100) "  0.1500",c21a2
write(40,100) "  0.2242",c21a3
write(40,100) "  0.2984",c21a4
write(40,100) "  0.3726",c21a5
write(40,100) "  0.4468",c21a6
write(40,100) "  0.5210",c21a7
write(40,100) "  0.5952",c21a8
write(40,100) "  0.6694",c21a9
write(40,100) "  0.7436",c21a10

write(40,*) " 22    11  "
write(40,*) " 0.0000        0.0000"
write(40,100) "  0.0900",c22a1
write(40,100) "  0.1500",c22a2
write(40,100) "  0.2242",c22a3
write(40,100) "  0.2984",c22a4
write(40,100) "  0.3726",c22a5
write(40,100) "  0.4468",c22a6
write(40,100) "  0.5210",c22a7
write(40,100) "  0.5952",c22a8
write(40,100) "  0.6694",c22a9
```

```
write(40,100) "  0.7436",c22a10

write(40,*) " 23    11  "
write(40,*) " 0.0000        0.0000"
write(40,100) "  0.0900",c23a1
write(40,100) "  0.1500",c23a2
write(40,100) "  0.2242",c23a3
write(40,100) "  0.2984",c23a4
write(40,100) "  0.3726",c23a5
write(40,100) "  0.4468",c23a6
write(40,100) "  0.5210",c23a7
write(40,100) "  0.5952",c23a8
write(40,100) "  0.6694",c23a9
write(40,100) "  0.7436",c23a10

write(40,*) " 24    11  "
write(40,*) " 0.0000        0.0000"
write(40,100) "  0.0900",c24a1
write(40,100) "  0.1500",c24a2
write(40,100) "  0.2242",c24a3
write(40,100) "  0.2984",c24a4
write(40,100) "  0.3726",c24a5
write(40,100) "  0.4468",c24a6
write(40,100) "  0.5210",c24a7
write(40,100) "  0.5952",c24a8
write(40,100) "  0.6694",c24a9
write(40,100) "  0.7436",c24a10

write(40,*) " 25    11  "
write(40,*) " 0.0000        0.0000"
write(40,100) "  0.0900",c25a1
write(40,100) "  0.1500",c25a2
write(40,100) "  0.2242",c25a3
write(40,100) "  0.2984",c25a4
write(40,100) "  0.3726",c25a5
write(40,100) "  0.4468",c25a6
write(40,100) "  0.5210",c25a7
write(40,100) "  0.5952",c25a8
write(40,100) "  0.6694",c25a9
write(40,100) "  0.7436",c25a10

write(40,*) " 26    11  "
write(40,*) " 0.0000        0.0000"
write(40,100) "  0.0900",c26a1
write(40,100) "  0.1500",c26a2
write(40,100) "  0.2242",c26a3
write(40,100) "  0.2984",c26a4
write(40,100) "  0.3726",c26a5
write(40,100) "  0.4468",c26a6
write(40,100) "  0.5210",c26a7
write(40,100) "  0.5952",c26a8
write(40,100) "  0.6694",c26a9
write(40,100) "  0.7436",c26a10

write(40,*) " 27    11  "
write(40,*) " 0.0000        0.0000"
write(40,100) "  0.0900",c27a1
```

```
write(40,100) "  0.1500",c27a2
write(40,100) "  0.2242",c27a3
write(40,100) "  0.2984",c27a4
write(40,100) "  0.3726",c27a5
write(40,100) "  0.4468",c27a6
write(40,100) "  0.5210",c27a7
write(40,100) "  0.5952",c27a8
write(40,100) "  0.6694",c27a9
write(40,100) "  0.7436",c27a10

write(40,*) " 28    11  "
write(40,*) " 0.0000        0.0000"
write(40,100) "  0.0900",c28a1
write(40,100) "  0.1500",c28a2
write(40,100) "  0.2242",c28a3
write(40,100) "  0.2984",c28a4
write(40,100) "  0.3726",c28a5
write(40,100) "  0.4468",c28a6
write(40,100) "  0.5210",c28a7
write(40,100) "  0.5952",c28a8
write(40,100) "  0.6694",c28a9
write(40,100) "  0.7436",c28a10

write(40,*) " 29    11  "
write(40,*) " 0.0000        0.0000"
write(40,100) "  0.0900",c29a1
write(40,100) "  0.1500",c29a2
write(40,100) "  0.2242",c29a3
write(40,100) "  0.2984",c29a4
write(40,100) "  0.3726",c29a5
write(40,100) "  0.4468",c29a6
write(40,100) "  0.5210",c29a7
write(40,100) "  0.5952",c29a8
write(40,100) "  0.6694",c29a9
write(40,100) "  0.7436",c29a10

write(40,*) " 30    11  "
write(40,*) " 0.0000        0.0000"
write(40,100) "  0.0900",c30a1
write(40,100) "  0.1500",c30a2
write(40,100) "  0.2242",c30a3
write(40,100) "  0.2984",c30a4
write(40,100) "  0.3726",c30a5
write(40,100) "  0.4468",c30a6
write(40,100) "  0.5210",c30a7
write(40,100) "  0.5952",c30a8
write(40,100) "  0.6694",c30a9
write(40,100) "  0.7436",c30a10

write(40,*) " 31    11  "
write(40,*) " 0.0000        0.0000"
write(40,100) "  0.0900",c31a1
write(40,100) "  0.1500",c31a2
write(40,100) "  0.2242",c31a3
write(40,100) "  0.2984",c31a4
write(40,100) "  0.3726",c31a5
write(40,100) "  0.4468",c31a6
```

```
write(40,100) "  0.5210",c31a7
write(40,100) "  0.5952",c31a8
write(40,100) "  0.6694",c31a9
write(40,100) "  0.7436",c31a10

write(40,*) " 32    11  "
write(40,*) " 0.0000        0.0000"
write(40,100) "  0.0900",c32a1
write(40,100) "  0.1500",c32a2
write(40,100) "  0.2242",c32a3
write(40,100) "  0.2984",c32a4
write(40,100) "  0.3726",c32a5
write(40,100) "  0.4468",c32a6
write(40,100) "  0.5210",c32a7
write(40,100) "  0.5952",c32a8
write(40,100) "  0.6694",c32a9
write(40,100) "  0.7436",c32a10

write(40,110) "0.0"

write(40,120) c1b,c2b,c3b,c4b,c5b,c6b,c7b,c8b
write(40,120) c9b,c10b,c11b,c12b,c13b,c14b,c15b,c16b
write(40,120) c17b,c18b,c19b,c20b,c21b,c22b,c23b,c24b
write(40,120) c25b,c26b,c27b,c28b,c29b,c30b,c31b,c32b

write(40,130) "13"

write(40,130) "144    0    0    0    0    0    0    0    0    0    0
0    0    0      "

do i=1,9
    write(40,140) i,"12    2    0    0    0       0.0    167.61
0.0         1"
end do

do i=10,72
    write(40,150) i,"12    2    0    0    0       0.0    167.61
0.0         1"
end do

i=1

do i=1,9
    write(40,140) i," 1    1    0    0    0       0.0    198.96
2.0         1"
end do

do i=10,72
    write(40,150) i," 1    1    0    0    0       0.0    198.96
2.0         1"
end do

i=1

write(40,130) "14"
```

```
write(40,130) " 5    0    0    0    0    0    0    0    0    0    0
0    0    0"

write(40,130) "15"

write(40,130) "0.0000000001          0.001          3.0     30.0    10.0
1.0"

write(40,160) "-1.0            0.0           0.0      0.0      0.0
0.0 "

close(40)

100 format(a,t17,f6.4)
110 format(7x,a)
120
format(5x,f5.3,5x,f5.3,5x,f5.3,5x,f5.3,5x,f5.3,5x,f5.3,5x,f5.3,5x,f5.3,5x,f5.3)
130 format(3x,a)
140 format(5x,i1,3x,a)
150 format(4x,i2,3x,a)
160 format(9x,a)

End PROGRAM MakeCTF
```

APPENDIX D: TakeCTF.f90

```
!----------------------------------------------------------------------
      PROGRAM TakeCTF
!----------------------------------------------------------------------
! This program reads in the data from the results_channels.out file
! and generates output that is placed into a text file adam.INP
!
! This program was created by Adam Rosenkrantz
!
! February 14, 2012
!



! Read the CTF state parameter data into memory

!----------------------------------------------------------------------

double precision matrix(208,4)

character dummy

integer num(208)

integer i,j

open(20,file="results_channels.out",status="UNKNOWN")

! Assign Index numbers
do i=1,8
      num(i)=i+782
end do

do i=9,16
      num(i)=i+786
end do

do i=17,24
      num(i)=i+790
end do

do i=25,32
      num(i)=i+806
end do

do i=33,40
      num(i)=i+942
end do

do i=41,48
      num(i)=i+958
end do

do i=49,56
      num(i)=i+974
```

```
end do

do i=57,64
      num(i)=i+1086
end do

do i=65,72
      num(i)=i+1090
end do

do i=73,80
      num(i)=i+1094
end do

do i=81,88
      num(i)=i+1110
end do

do i=89,96
      num(i)=i+1246
end do

do i=97,104
      num(i)=i+1262
end do

do i=105,112
      num(i)=i+1278
end do

do i=113,120
      num(i)=i+1390
end do

do i=121,128
      num(i)=i+1394
end do

do i=129,136
      num(i)=i+1398
end do

do i=137,144
      num(i)=i+1414
end do

do i=145,152
      num(i)=i+1550
end do

do i=153,160
      num(i)=i+1566
end do

do i=161,168
      num(i)=i+1582
end do
```

```
do i=169,176
      num(i)=i+1694
end do

do i=177,184
      num(i)=i+1698
end do

do i=185,192
      num(i)=i+1702
end do

do i=193,200
      num(i)=i+1706
end do

do i=201,208
      num(i)=i+1710
end do

!Read channel 12
do i=1,1272
      read(20,*)
end do

do i=1,8
      j=9-i
    read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy
,matrix(j,4)

end do

do i=1,22
      read(20,*)
end do

do i=1,8
      j=9-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,matrix(j,2),dummy,dummy,dummy
,dummy,matrix(j,3)
end do

!Read channel 13
do i=1,68
      read(20,*)
end do

do i=9,16
      j=25-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy
,matrix(j,4)

end do
```

```
do i=1,22
      read(20,*)
end do

do i=9,16
      j=25-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,matrix(j,2),dummy,dummy,dummy
,dummy,matrix(j,3)
end do

!Read channel 14
do i=1,68
      read(20,*)
end do

do i=17,24
      j=41-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy
,matrix(j,4)

end do

do i=1,22
      read(20,*)
end do

do i=17,24
      j=41-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,matrix(j,2),dummy,dummy,dummy
,dummy,matrix(j,3)
end do

!Read channel 16
do i=1,174
      read(20,*)
end do

do i=25,32
      j=57-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy
,matrix(j,4)

end do

do i=1,22
      read(20,*)
end do

do i=25,32
      j=57-i
```

```
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,matrix(j,2),dummy,dummy,dummy
,dummy,matrix(j,3)
end do

!Read channel 22
do i=1,598
      read(20,*)
end do

do i=33,40
      j=73-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy
,matrix(j,4)

end do

do i=1,22
      read(20,*)
end do

do i=33,40
      j=73-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,matrix(j,2),dummy,dummy,dummy
,dummy,matrix(j,3)
end do

!Read channel 24
do i=1,174
      read(20,*)
end do

do i=41,48
      j=89-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy
,matrix(j,4)

end do

do i=1,22
      read(20,*)
end do

do i=41,48
      j=89-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,matrix(j,2),dummy,dummy,dummy
,dummy,matrix(j,3)
end do

!Read channel 26
do i=1,174
      read(20,*)
end do
```

```
do i=49,56
      j=105-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy
,matrix(j,4)

end do

do i=1,22
      read(20,*)
end do

do i=49,56
      j=105-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,matrix(j,2),dummy,dummy,dummy
,dummy,matrix(j,3)
end do

!Read channel 30
do i=1,386
      read(20,*)
end do

do i=57,64
      j=121-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy
,matrix(j,4)

end do

do i=1,22
      read(20,*)
end do

do i=57,64
      j=121-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,matrix(j,2),dummy,dummy,dummy
,dummy,matrix(j,3)
end do

!Read channel 31
do i=1,68
      read(20,*)
end do

do i=65,72
      j=137-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy
,matrix(j,4)

end do
```

```
do i=1,22
      read(20,*)
end do

do i=65,72
      j=137-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,matrix(j,2),dummy,dummy,dummy
,dummy,matrix(j,3)
end do

!Read channel 32
do i=1,68
      read(20,*)
end do

do i=73,80
      j=153-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy
,matrix(j,4)

end do

do i=1,22
      read(20,*)
end do

do i=73,80
      j=153-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,matrix(j,2),dummy,dummy,dummy
,dummy,matrix(j,3)
end do

!Read channel 34
do i=1,174
      read(20,*)
end do

do i=81,88
      j=169-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy
,matrix(j,4)

end do

do i=1,22
      read(20,*)
end do

do i=81,88
      j=169-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,matrix(j,2),dummy,dummy,dummy
,dummy,matrix(j,3)
```

```
end do

!Read channel 40
do i=1,598
      read(20,*)
end do

do i=89,96
      j=185-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy
,matrix(j,4)

end do

do i=1,22
      read(20,*)
end do

do i=89,96
      j=185-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,matrix(j,2),dummy,dummy,dummy
,dummy,matrix(j,3)
end do

!Read channel 42
do i=1,174
      read(20,*)
end do

do i=97,104
      j=201-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy
,matrix(j,4)

end do

do i=1,22
      read(20,*)
end do

do i=97,104
      j=201-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,matrix(j,2),dummy,dummy,dummy
,dummy,matrix(j,3)
end do

!Read channel 44
do i=1,174
      read(20,*)
end do

do i=105,112
      j=217-i
```

```
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy
,matrix(j,4)

end do

do i=1,22
      read(20,*)
end do

do i=105,112
      j=217-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,matrix(j,2),dummy,dummy,dummy
,dummy,matrix(j,3)
end do

!Read channel 48
do i=1,386
      read(20,*)
end do

do i=113,120
      j=233-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy
,matrix(j,4)

end do

do i=1,22
      read(20,*)
end do

do i=113,120
      j=233-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,matrix(j,2),dummy,dummy,dummy
,dummy,matrix(j,3)
end do

!Read channel 49
do i=1,68
      read(20,*)
end do

do i=121,128
      j=249-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy
,matrix(j,4)

end do

do i=1,22
      read(20,*)
end do
```

```
do i=121,128
      j=249-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,matrix(j,2),dummy,dummy,dummy
,dummy,matrix(j,3)
end do

!Read channel 50
do i=1,68
      read(20,*)
end do

do i=129,136
      j=265-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy
,matrix(j,4)

end do

do i=1,22
      read(20,*)
end do

do i=129,136
      j=265-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,matrix(j,2),dummy,dummy,dummy
,dummy,matrix(j,3)
end do

!Read channel 52
do i=1,174
      read(20,*)
end do

do i=137,144
      j=281-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy
,matrix(j,4)

end do

do i=1,22
      read(20,*)
end do

do i=137,144
      j=281-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,matrix(j,2),dummy,dummy,dummy
,dummy,matrix(j,3)
end do

!Read channel 58
```

```
do i=1,598
      read(20,*)
end do

do i=145,152
      j=297-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy
,matrix(j,4)

end do

do i=1,22
      read(20,*)
end do

do i=145,152
      j=297-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,matrix(j,2),dummy,dummy,dummy
,dummy,matrix(j,3)
end do

!Read channel 60
do i=1,174
      read(20,*)
end do

do i=153,160
      j=313-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy
,matrix(j,4)

end do

do i=1,22
      read(20,*)
end do

do i=153,160
      j=313-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,matrix(j,2),dummy,dummy,dummy
,dummy,matrix(j,3)
end do

!Read channel 62
do i=1,174
      read(20,*)
end do

do i=161,168
      j=329-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy
,matrix(j,4)
```

```
end do

do i=1,22
      read(20,*)
end do

do i=161,168
      j=329-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,matrix(j,2),dummy,dummy,dummy
,dummy,matrix(j,3)
end do

!Read channel 66
do i=1,386
      read(20,*)
end do

do i=169,176
      j=345-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy
,matrix(j,4)

end do

do i=1,22
      read(20,*)
end do

do i=169,176
      j=345-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,matrix(j,2),dummy,dummy,dummy
,dummy,matrix(j,3)
end do

!Read channel 67
do i=1,68
      read(20,*)
end do

do i=177,184
      j=361-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy
,matrix(j,4)

end do

do i=1,22
      read(20,*)
end do

do i=177,184
      j=361-i
```

```
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,matrix(j,2),dummy,dummy,dummy
,dummy,matrix(j,3)
end do

!Read channel 68
do i=1,68
      read(20,*)
end do

do i=185,192
      j=377-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy
,matrix(j,4)

end do

do i=1,22
      read(20,*)
end do

do i=185,192
      j=377-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,matrix(j,2),dummy,dummy,dummy
,dummy,matrix(j,3)
end do

!Read channel 69
do i=1,68
      read(20,*)
end do

do i=193,200
      j=393-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy
,matrix(j,4)

end do

do i=1,22
      read(20,*)
end do

do i=193,200
      j=393-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,matrix(j,2),dummy,dummy,dummy
,dummy,matrix(j,3)
end do

!Read channel 70
do i=1,68
      read(20,*)
end do
```

```
do i=201,208
      j=409-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy,dummy
,matrix(j,4)
end do

do i=1,22
      read(20,*)
end do

do i=201,208
      j=409-i
      read(20,*)
dummy,dummy,dummy,dummy,dummy,dummy,dummy,matrix(j,2),dummy,dummy,dummy
,dummy,matrix(j,3)
end do

! Convert units for the table
do i=1,208
      matrix(i,4)=matrix(i,4)/1000
      matrix(i,2)=matrix(i,2)+273.15
      matrix(i,3)=matrix(i,3)+273.15
end do

close(20)

!Write the state parameters into a static text file

!-----------------------------------------------------------------------

open(30,file="adam.INP",status="UNKNOWN")

!Header
write(30,100) "SAFARI.lib","SAFARI_out.lib   "

write(30,110) "SSFE_300g_element.SPX   "

write(30,100) "208","3"

!Copy down node ID's and state parameters
do i=1,208
      write(30,120) num(i),matrix(i,2),matrix(i,3),matrix(i,4)
end do

close(30)

100 format(a,1x,a)
110 format(a)
120 format(i4,1x,f6.2,1x,f6.2,1x,f7.5)

End PROGRAM TakeCTF
```

APPENDIX E: generatelib.f90

```
program GenerateLib
! Program reads input file (example below) and produces MGRAC style
cross-section format file at given state conditions
! Only specified material numbers are updated
! Library assumes materials are numbered as z=1,MaxZ, then x=1,MaxX and
then y=1,MaxY, with y=1,x=1,z=1 indicated the axial bottom of the top
left corner assembly (when looking at SAFARI core map)
! Also reads base .lib file containing fixed cross-sections for all
materials
! Reads approximation library containing polynomial format (.spx file)
! Note that burnup should be not specified, since only TH feedback is
supported in this version of the code (easy to add if needed). The
burnup value in the base .lib file remains untouched
! Not all materials (or even fuel have to be reconstructured) - only
the materials specified in input is updated, rest is assumed fixed from
SAFARI.lib library.



! Example input file
!SAFARI.lib SAFARI_out.lib  ! fixed cross-section mgrac input file, and
fixed cross-section file for output
!Fuel.SPX                   ! SPX file (polynomial library)
!208 3                      ! number of materials with feedback
representation and number of state parameters
!783 313.15 303.15 0.99284    !material id, state parameters values
(fuel temp, mod temp, mod density)
!784 313.15 303.15 0.99284    !material id, state parameters values
(fuel temp, mod temp, mod density)
!...




USE Kinds, only         :     doubel
use read_approx_lib
use data_types
use reconstruct
use FileHandle

    implicit none

    character (len = 255)           :: lib_file         ! MGRAC fixed
cross-sections file format (state parameter values additionally
written)
    character (len =255)            :: input_file       ! name of the
small input file to this reconstruction program
    character (len =255)            :: spx_file         ! global
polynomial xs library
    character (len =255)            :: lib_file_out     ! global
polynomial xs library

    integer                         :: unit_input, ierror, aerror
```

```
    integer                         :: i, j, k          ! loop
counter(s)
    integer                         :: groups,numMaterials_mgrac,
num_to_fit, num_state

    integer, allocatable            :: ids(:)
    real(kind = doubel), allocatable:: values(:,:)

    TYPE Statepoint
       character(len=255)                                 :: StateParam
! Name of this state parameter
       real(kind = doubel)                                :: value
! value of this state parameter
       integer                                            :: StateIndex
! Index regarding the order of this state parameter
    END TYPE Statepoint

!Details/Cross Sections for a given material
    TYPE MaterialData
       integer                                            :: matID
! ID of material
       character(len=255)                                 :: matName
! Name of material
       real(kind=doubel),dimension(:,:),allocatable       :: Dif
! Diffusion coefficient per group per direction
       real(kind=doubel),dimension(:),allocatable         :: Chi
! Chi cross-section
       real(kind=doubel),dimension(:),allocatable         :: Pow
! Power cross-section
       real(kind=doubel),dimension(:),allocatable         :: nuFis
! nuFission cross-section
       real(kind=doubel),dimension(:),allocatable         :: Fis
! Fission cross-section
       real(kind=doubel),dimension(:),allocatable         :: Abs
! Absorption cross-section
       real(kind=doubel),dimension(:),allocatable         :: Rem
! Removal cross-section
       real(kind=doubel),dimension(:,:),allocatable       :: Scat
! Scttering matrix
       integer                                            :: StateDim
! Dimensionality of cross-section model
       TYPE(Statepoint),dimension(:),allocatable          :: State
! Current state parameter values for this material
       character(len=255)                                 :: matType
! material type name
    END TYPE MaterialData



    TYPE (MaterialData), dimension(:),allocatable         :: Material
! Materials for the given core



    call GETARG(1, input_file)
    unit_input = openFile(trim(adjustl(input_file)),'OLD','READ')
```

```
    !read input file
    read(unit_input,*) lib_file,lib_file_out
    read(unit_input,*) spx_file
    read(unit_input,*) num_to_fit, num_state
    allocate(ids(num_to_fit), values(num_to_fit,num_state), STAT=
aerror)

    do i=1, num_to_fit
       read(unit_input,*) ids(i),values(i,:)
    end do

    call readXSMGRAC(ierror)
    write(*,*) 'Writing Fixed cross-section output
file...',trim(adjustl(lib_file_out))
    call writeXSMGRAC(ierror)




    !read fixed format mgrac xs file

    call destruct_structures



contains



!---------------------------------------------------------------------
Subroutine setXSfromLib(id,name,g,ndim,statepoint,value)
!Subroutine to overwrite selected cross-sections from reconstruction
from library (SPX format)
!---------------------------------------------------------------------
USE data_types, only   : materialname, map_size, materialmap, findXS
USE reconstruct, only  : normalise, getCS

integer                                    :: xs
integer, INTENT(IN)                        :: id,ndim,g
character(len=255),INTENT(IN)              :: name
real(kind = doubel), INTENT(OUT)           :: value
real(kind = doubel), INTENT(IN)            :: statepoint(1:ndim)
real(kind = doubel)                        ::
normalised_statepoint(1:ndim)

value=0.0_doubel

! Reconstruct capture cross-section: '
   call findXS(xs,name,g,0)  ! find only macroscopic cross sections at
this stage
   if (xs /= -1) then
      call normalise(statepoint, normalised_statepoint)
      call getCS(normalised_statepoint, value, xs)
```

```
!      write(800,*) 'Reconstruct Mat',id,' for
',trim(adjustl(name)),g,0,' for statepoint
',normalised_statepoint,value,' Compare : ',Material(id)%Rem(g),' %
Diff ',100*(abs(value-Material(id)%Rem(g)))/Material(id)%Rem(g)

   else
      write(*,*) 'Cannot find required cross-section:
',trim(adjustl(name)),g,0,' in library - fatal error'
      stop
   end if




End Subroutine setXSfromLib
!----------------------------------------------------------------------


Subroutine readXSMGRAC(ierror   &
)

USE read_approx_lib, only : read_lib
USE data_types, only   : materialname, map_size,
materialmap,ndim_param,lower_bound, upper_bound, param_id_num,
param_id_name

!Subroutine to read cross-section data into memory
!----------------------------------------------------------------------
integer                                    ::
status,i,j,k,g,unit_xs,aerror,found,dummyi,id
integer                                    ::
ucount,vcount,wcount,thermal     ! Counters to check that material
numbers are assigned
integer, INTENT(OUT)                       :: ierror
integer,dimension(:),allocatable           :: dzones
character(len = 255)                       :: dummy,xsname
real(kind = doubel)                        :: temp,value
real(kind = doubel),allocatable            :: statepoint(:)

ierror = 0


write(*,*) 'Reading SPX cross-section library:
',trim(adjustl(spx_file))
call read_lib(trim(adjustl(spx_file)))
allocate(statepoint(ndim_param),STAT=ierror)


write(*,*) 'Processing XS file in MGRAC format'
unit_xs = openFile(trim(adjustl(lib_file)),"OLD","READ")

if (unit_xs /= -1) then
   read (unit_xs,*) dummy !read SFOI
   read (unit_xs,*) dummy !read XSEC V1.0.0

   read (unit_xs,*) numMaterials_mgrac, groups,thermal
```

```
   write(*,*) 'XS structure in ',groups,' groups.'
   allocate(Material(numMaterials_mgrac),STAT=aerror)
   found = 0

   do i=1,numMaterials_mgrac

      Material(i)%matID = i
      allocate(Material(i)%Dif(3,groups),STAT=aerror)
      allocate(Material(i)%nuFis(groups),STAT=aerror)
      allocate(Material(i)%Fis(groups),STAT=aerror)
      allocate(Material(i)%Pow(groups),STAT=aerror)
      allocate(Material(i)%Abs(groups),STAT=aerror)
      allocate(Material(i)%Chi(groups),STAT=aerror)
      allocate(Material(i)%Rem(groups),STAT=aerror)
      allocate(Material(i)%Scat(groups,groups),STAT=aerror)

      Material(i)%stateDim = ndim_param
      read(unit_xs,*)
Material(i)%matName,dummy,dummy,dummy,dummy,statepoint(:)
!        scale to mwd/Tn
         statepoint(1) = statepoint(1) * 1000
         allocate(Material(i)%State(ndim_param),STAT=aerror)
         do j=1, ndim_param
            Material(i)%State(j)%StateIndex=param_id_num(j)
            Material(i)%State(j)%StateParam=param_id_name(j)
            Material(i)%State(j)%value=statepoint(j)

         end do
! read state parameters from input file to overwrite that which is
written on library
         if (any(materialmap(:) == i)) then
! find set of state values in input for this material
            do j=1,num_to_fit
               if (i == ids(j)) then
!                 write(*,*) 'Found adjustment for material: ',i,'
from ',Material(i)%State(2:ndim_param)%value,' to ',
values(j,1:num_state)
                  Material(i)%State(2:ndim_param)%value =
values(j,1:num_state)
                  statepoint(:) = Material(i)%State(1:ndim_param)%value
                  found = found + 1
                  exit
               end if
            end do

         end if


      read (unit_xs,*) (Material(i)%Chi(j),j=1,groups)


      do g=1,groups
         read(unit_xs,*) Material(i)%Dif(1,g)
      end do
      Material(i)%Dif(2,:)=Material(i)%Dif(1,:)
      Material(i)%Dif(3,:)=Material(i)%Dif(1,:)
```

```
      do g = 1, groups
         read(unit_xs,*)
Material(i)%Rem(g),Material(i)%nuFis(g),(Material(i)%Scat(j,g),j=1,grou
ps),Material(i)%Pow(g)
         if (any(materialmap(:) == i)) then
            xsname='XSC'
            call setXSfromLib(i,xsname,g,ndim_param,statepoint,value)
            Material(i)%Rem(g)=value
            Material(i)%Abs(g)=value
            xsname='XSF'
            call setXSfromLib(i,xsname,g,ndim_param,statepoint,value)
            Material(i)%Pow(g)=value
            xsname='XSTR'
            call setXSfromLib(i,xsname,g,ndim_param,statepoint,value)
            Material(i)%Dif(:,g)=value
            xsname='XSNU'
            call setXSfromLib(i,xsname,g,ndim_param,statepoint,value)
            Material(i)%nuFis(g)=value



         end if


      end do
       !adjust absorption to removal cross-section
      do g = 1, groups
        Material(i)%Rem(g) = Material(i)%Rem(g) +
sum(Material(i)%Scat(g,:))
     end do


  end do
  write(*,*) ' Updates made to cross-sections of ',found,' materials.'

  call closeFile(unit_xs)
else
   ierror=1
   write(*,*) 'Error reading cross-section input file:
',trim(adjustl(lib_file))

end if
write(*,*) 'Done processing XS file'
End Subroutine readXSMGRAC



!---------------------------------------------------------------------

Subroutine writeXSMGRAC(ierror)

!Subroutine to read cross-section data into memory
!---------------------------------------------------------------------
```

```
integer                                         ::
status,i,j,k,g,unit_xs,aerror,found,dummyi
integer                                         ::
ucount,vcount,wcount,thermal,numgroups     ! Counters to check that
material numbers are assigned
integer, INTENT(OUT)                            :: ierror
integer,dimension(:),allocatable                :: dzones
character(len = 255)                            :: dummy, templine



ierror = 0

unit_xs = openFile(trim(adjustl(lib_file_out)),'REPLACE','WRITE')



if (unit_xs /= -1) then
   write (unit_xs,*) 'SFOI'
   write (unit_xs,*) 'XSEC V1.0.0'

   write (unit_xs,500) numMaterials_mgrac, groups,2

   500 format(2x,i4,1x,i1,1x,i1)
!
   do i=1,numMaterials_mgrac

      Material(i)%State(1)%value = Material(i)%State(1)%value /
1000_doubel
      write(unit_xs,510)
trim(adjustl(Material(i)%matName)),0,0,0,0,(Material(i)%State(j)%value,
j=1,Material(i)%stateDim)

      510
format(1x,a,1x,i1,1x,i1,1x,i1,1x,i1,1x,i1,3x,f9.5,3x,f9.5,3x,f9.5,3x,f9.5)

      write (unit_xs,'(10ES10.2)') (Material(i)%CHI(j) ,j=1,groups)



      do g=1,groups
         write(unit_xs,'(F9.5)') Material(i)%Dif(1,g)
      end do


      do g = 1, groups
         Material(i)%Abs(g) = Material(i)%Rem(g) -
sum(Material(i)%scat(g,:))
      end do



      do g = 1, groups
```

```
         write(unit_xs,'(20F9.5)')
Material(i)%abs(g),Material(i)%nufis(g),(Material(i)%scat(j,g),j=1,grou
ps),Material(i)%pow(g)

      end do

   end do
   call closeFile(unit_xs)
else
   ierror=1
   write(*,*) 'Error writing cross-section input file:
',trim(adjustl(lib_file_out))

end if

End Subroutine writeXSMGRAC

!---------------------------------------------------------------------




end program Generatelib
```

APPENDIX F: MakeNEM.f90

```
!-----------------------------------------------------------------------
      PROGRAM MakeNEM
!-----------------------------------------------------------------------
! This program reads in the data from the SAFARI.XS file and generates
! output that is directly pasted into a nemin file
!
! This program is an altered subroutine produced by Rian.
!
! April 19, 2011
!
! Updated by Adam Rosenkrantz from separate file to direct paste,
! and various other minor changes.
!
! February 13, 2012
!



!Read the cross-section data into memory

!-----------------------------------------------------------------------

integer                                         ::
status,i,j,k,g,unit_xs,aerror,found,dummyi
integer                                         :: numMaterials
character(len = 6)                              :: Material_Name

integer                                         ::
ucount,vcount,wcount,groups,thermal,Material     ! Counters to check
that material numbers are assigned

! integer, INTENT(OUT)                          :: ierror

! integer,dimension(:),allocatable              :: dzones

integer                                        :: dummy
character(len = 1)                             :: dummy_p
character(len =20)                             :: dummy_c

real,   allocatable, dimension(:,:)            :: Dif, nuFis, Fis, Pow, Abs,
Chi, Rem

real,   allocatable, dimension(:,:,:)          :: Scat



ierror = 0

open(20,file="SAFARI_out.lib",status="UNKNOWN")

if (20 /= -1) then
   read (20,*) dummy_c !read SFOI
```

```
   read (20,*) dummy_c !read XSEC V1.0.0

   read (20,*) numMaterials, groups,thermal
   allocate (Dif(numMaterials,groups))
   allocate (nuFis(numMaterials,groups))
   allocate (Fis(numMaterials,groups))
   allocate (Pow(numMaterials,groups))
   allocate (Abs(numMaterials,groups))
   allocate (Chi(numMaterials,groups))
   allocate (Rem(numMaterials,groups))
   allocate (Scat(numMaterials,groups,groups))


   do i=1,numMaterials  ! loop over all materials in the file
      read(20,*) Material,dummy,dummy,dummy,dummy   ! read options

      read(20,*) (Chi(Material,j),j=1,groups)           ! read fission
spectrum
      do g=1,groups
         read(20,*) Dif(Material,g)    !read diffusion coefficients Ã?-
1 per line
      end do

      do g = 1, groups
         read(20,*)
Rem(Material,g),nuFis(Material,g),(Scat(Material,j,g),j=1,groups),Pow(M
aterial,g)
      end do
      !adjust absorption to removal cross-section
!      do g = 1, groups
!         do k = 1, groups
!            Rem(Material,g) = Rem(Material,g) + Scat(Material,g,k)
!         end do
!      end do
    end do
else
   ierror=1
   write(unit_out,*) 'Error reading cross-section input file: '
   write(*,*) 'Error reading cross-section input file: '
end if

close(20)

! Open a file to print pastable fission spectra and cross sections

open(30,file="nemin",status="UNKNOWN")

!
!
!================================
!    Paste into nemin
!================================
!
i=1
do i=1,14
    read(30,*)
end do
```

```
i=1

do i=1,numMaterials
   ! Print the Fission Spectrum
   write(30,100) (Chi(i,j),j=1,groups)
end do

! Print the cross sections

do i=1,numMaterials
   write(30,110) 1          ! The leading 1
   write(30,110) 0          ! The zero
   do g=1,groups
      write(30,120) 1,1,0,1    ! The 1 1 0 1 indicating that we have 3
cross sections minus fission
      write(30,130) Dif(i,g)
      write(30,130) Rem(i,g)
      write(30,130) nuFis(i,g)
   end do
   write(30,140) 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1   ! 36 1's indicating
we have a full scattering matrix (groups x groups)
   write(30,140) 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
   write(30,150) 1,1,1,1
   do g=1,groups
      do j=1,groups
         write(30,130) Scat(i,j,g)
      end do
   end do
end do

write(30,510) "1r 9.000     1r 6.000     1r 2.500     9r 7.710     1r
2.500     1r 6.000     1r 9.000   "
write(30,510) "1r 9.000     1r 6.000     1r 2.500     8r 8.100     1r
3.500     1r 4.600     1r15.000   "
write(30,510) "2r 7.500     8r 7.421     2r 7.500"

i=1
do i=1,3
    write(30,510) "        1         1         1         1         1
1         1         1"
    write(30,510) "        1         1         1         1         1
1"
    write(30,510) "       15        15        15        15        15
15        15        15"
    write(30,510) "       15        15        15        15        15
15"
end do
i=1

write(30,510) "        1         1         1         1         1
1"

write(30,510) "    1     2     3     4     5     6     7     8     9
10    11    12    13    14    15"
write(30,510) "   16    17    18    19    20    21    22    23    24
25    26    27    28    29    30"
```

```
write(30,510) "   31    32    33    34    35    36    37    38    39
40    41    42    43    44    45"
write(30,510) "   46    47    48    49    50    51    52    53    54
55    56    57    58    59    60"
write(30,510) "   61    62    63    64    65    66    67    68    69
70    71    72    73    74    75"
write(30,510) "   76    77    78    79    80    81    82    83    84
85    86    87    88    89    90"
write(30,510) "   91    92    93    94    95    96    97    98    99
100   101   102   103   104   105"
write(30,510) "  106   107   108   109   110   111   112   113   114
115   116   117   118   119   120"
write(30,510) "  121   122   123   124   125   126   127   128   129
130   131   132   133   134   135"
write(30,510) "  136   137   138   139   140   141   142   143   144
145   146   147   148   149   150"
write(30,510) "  151   152   153   154   155   156   157   158   159
160   161   162   163   164   165"
write(30,510) "  166   167   168   169   170   171   172   173   174
175   176   177   178   179   180"
write(30,510) "  181   182   183   184   185   186   187   188   189
190   191   192   193   194   195"
write(30,510) "  196   197   198   199   200   201   202   203   204
205   206   207   208   209   210"

write(30,510) "    1    13    25    37    49    61    73    85    97
109   121   133   145   157   169   181   193   205   217   229   241
253   265   277   289   301   313   325   337   349   361   373   385
397   409   421   433   445   457   469   481   493   505   517   529
541   553   565   577   589   601   613   625   637   649   661   673
685   697   709   721   733   745   757   769   781   793   805   817
829   841   853   865   877   889   901   913   925   937   949   961
973   985   997  1009  1021  1033  1045  1057  1069  1081  1093  1105
1117  1129  1141  1153  1165  1177  1189  1201  1213  1225  1237  1249
1261  1273  1285  1297  1309  1321  1333  1345  1357  1369  1381  1393
1405  1417  1429  1441  1453  1465  1477  1489  1501  1513  1525  1537
1549  1561  1573  1585  1597  1609  1621  1633  1645  1657  1669  1681
1693  1705  1717  1729  1741  1753  1765  1777  1789  1801  1813  1825
1837  1849  1861  1873  1885  1897  1909  1921  1933  1945  1957  1969
1981  1993  2005  2017  2029  2041  2053  2065  2077  2089  2101  2113
2125  2137  2149  2161  2173  2185  2197  2209  2221  2233  2245  2257
2269  2281  2293  2305  2317  2329  2341  2353  2365  2377  2389  2401
2413  2425  2437  2449  2461  2473  2485  2497  2509"
write(30,510) "    2    14    26    38    50    62    74    86    98
110   122   134   146   158   170   182   194   206   218   230   242
254   266   278   290   302   314   326   338   350   362   374   386
398   410   422   434   446   458   470   482   494   506   518   530
542   554   566   578   590   602   614   626   638   650   662   674
686   698   710   722   734   746   758   770   782   794   806   818
830   842   854   866   878   890   902   914   926   938   950   962
974   986   998  1010  1022  1034  1046  1058  1070  1082  1094  1106
1118  1130  1142  1154  1166  1178  1190  1202  1214  1226  1238  1250
1262  1274  1286  1298  1310  1322  1334  1346  1358  1370  1382  1394
1406  1418  1430  1442  1454  1466  1478  1490  1502  1514  1526  1538
1550  1562  1574  1586  1598  1610  1622  1634  1646  1658  1670  1682
1694  1706  1718  1730  1742  1754  1766  1778  1790  1802  1814  1826
1838  1850  1862  1874  1886  1898  1910  1922  1934  1946  1958  1970
```

```
1982  1994  2006  2018  2030  2042  2054  2066  2078  2090  2102  2114
2126  2138  2150  2162  2174  2186  2198  2210  2222  2234  2246  2258
2270  2282  2294  2306  2318  2330  2342  2354  2366  2378  2390  2402
2414  2426  2438  2450  2462  2474  2486  2498  2510"
write(30,510) "    3    15    27    39    51    63    75    87    99
111   123   135   147   159   171   183   195   207   219   231   243
255   267   279   291   303   315   327   339   351   363   375   387
399   411   423   435   447   459   471   483   495   507   519   531
543   555   567   579   591   603   615   627   639   651   663   675
687   699   711   723   735   747   759   771   783   795   807   819
831   843   855   867   879   891   903   915   927   939   951   963
975   987   999  1011  1023  1035  1047  1059  1071  1083  1095  1107
1119  1131  1143  1155  1167  1179  1191  1203  1215  1227  1239  1251
1263  1275  1287  1299  1311  1323  1335  1347  1359  1371  1383  1395
1407  1419  1431  1443  1455  1467  1479  1491  1503  1515  1527  1539
1551  1563  1575  1587  1599  1611  1623  1635  1647  1659  1671  1683
1695  1707  1719  1731  1743  1755  1767  1779  1791  1803  1815  1827
1839  1851  1863  1875  1887  1899  1911  1923  1935  1947  1959  1971
1983  1995  2007  2019  2031  2043  2055  2067  2079  2091  2103  2115
2127  2139  2151  2163  2175  2187  2199  2211  2223  2235  2247  2259
2271  2283  2295  2307  2319  2331  2343  2355  2367  2379  2391  2403
2415  2427  2439  2451  2463  2475  2487  2499  2511"
write(30,510) "    4    16    28    40    52    64    76    88   100
112   124   136   148   160   172   184   196   208   220   232   244
256   268   280   292   304   316   328   340   352   364   376   388
400   412   424   436   448   460   472   484   496   508   520   532
544   556   568   580   592   604   616   628   640   652   664   676
688   700   712   724   736   748   760   772   784   796   808   820
832   844   856   868   880   892   904   916   928   940   952   964
976   988  1000  1012  1024  1036  1048  1060  1072  1084  1096  1108
1120  1132  1144  1156  1168  1180  1192  1204  1216  1228  1240  1252
1264  1276  1288  1300  1312  1324  1336  1348  1360  1372  1384  1396
1408  1420  1432  1444  1456  1468  1480  1492  1504  1516  1528  1540
1552  1564  1576  1588  1600  1612  1624  1636  1648  1660  1672  1684
1696  1708  1720  1732  1744  1756  1768  1780  1792  1804  1816  1828
1840  1852  1864  1876  1888  1900  1912  1924  1936  1948  1960  1972
1984  1996  2008  2020  2032  2044  2056  2068  2080  2092  2104  2116
2128  2140  2152  2164  2176  2188  2200  2212  2224  2236  2248  2260
2272  2284  2296  2308  2320  2332  2344  2356  2368  2380  2392  2404
2416  2428  2440  2452  2464  2476  2488  2500  2512"
write(30,510) "    5    17    29    41    53    65    77    89   101
113   125   137   149   161   173   185   197   209   221   233   245
257   269   281   293   305   317   329   341   353   365   377   389
401   413   425   437   449   461   473   485   497   509   521   533
545   557   569   581   593   605   617   629   641   653   665   677
689   701   713   725   737   749   761   773   785   797   809   821
833   845   857   869   881   893   905   917   929   941   953   965
977   989  1001  1013  1025  1037  1049  1061  1073  1085  1097  1109
1121  1133  1145  1157  1169  1181  1193  1205  1217  1229  1241  1253
1265  1277  1289  1301  1313  1325  1337  1349  1361  1373  1385  1397
1409  1421  1433  1445  1457  1469  1481  1493  1505  1517  1529  1541
1553  1565  1577  1589  1601  1613  1625  1637  1649  1661  1673  1685
1697  1709  1721  1733  1745  1757  1769  1781  1793  1805  1817  1829
1841  1853  1865  1877  1889  1901  1913  1925  1937  1949  1961  1973
1985  1997  2009  2021  2033  2045  2057  2069  2081  2093  2105  2117
2129  2141  2153  2165  2177  2189  2201  2213  2225  2237  2249  2261
```

```
2273  2285  2297  2309  2321  2333  2345  2357  2369  2381  2393  2405
2417  2429  2441  2453  2465  2477  2489  2501  2513"
write(30,510) "    6    18    30    42    54    66    78    90   102
114   126   138   150   162   174   186   198   210   222   234   246
258   270   282   294   306   318   330   342   354   366   378   390
402   414   426   438   450   462   474   486   498   510   522   534
546   558   570   582   594   606   618   630   642   654   666   678
690   702   714   726   738   750   762   774   786   798   810   822
834   846   858   870   882   894   906   918   930   942   954   966
978   990  1002  1014  1026  1038  1050  1062  1074  1086  1098  1110
1122  1134  1146  1158  1170  1182  1194  1206  1218  1230  1242  1254
1266  1278  1290  1302  1314  1326  1338  1350  1362  1374  1386  1398
1410  1422  1434  1446  1458  1470  1482  1494  1506  1518  1530  1542
1554  1566  1578  1590  1602  1614  1626  1638  1650  1662  1674  1686
1698  1710  1722  1734  1746  1758  1770  1782  1794  1806  1818  1830
1842  1854  1866  1878  1890  1902  1914  1926  1938  1950  1962  1974
1986  1998  2010  2022  2034  2046  2058  2070  2082  2094  2106  2118
2130  2142  2154  2166  2178  2190  2202  2214  2226  2238  2250  2262
2274  2286  2298  2310  2322  2334  2346  2358  2370  2382  2394  2406
2418  2430  2442  2454  2466  2478  2490  2502  2514"
write(30,510) "    7    19    31    43    55    67    79    91   103
115   127   139   151   163   175   187   199   211   223   235   247
259   271   283   295   307   319   331   343   355   367   379   391
403   415   427   439   451   463   475   487   499   511   523   535
547   559   571   583   595   607   619   631   643   655   667   679
691   703   715   727   739   751   763   775   787   799   811   823
835   847   859   871   883   895   907   919   931   943   955   967
979   991  1003  1015  1027  1039  1051  1063  1075  1087  1099  1111
1123  1135  1147  1159  1171  1183  1195  1207  1219  1231  1243  1255
1267  1279  1291  1303  1315  1327  1339  1351  1363  1375  1387  1399
1411  1423  1435  1447  1459  1471  1483  1495  1507  1519  1531  1543
1555  1567  1579  1591  1603  1615  1627  1639  1651  1663  1675  1687
1699  1711  1723  1735  1747  1759  1771  1783  1795  1807  1819  1831
1843  1855  1867  1879  1891  1903  1915  1927  1939  1951  1963  1975
1987  1999  2011  2023  2035  2047  2059  2071  2083  2095  2107  2119
2131  2143  2155  2167  2179  2191  2203  2215  2227  2239  2251  2263
2275  2287  2299  2311  2323  2335  2347  2359  2371  2383  2395  2407
2419  2431  2443  2455  2467  2479  2491  2503  2515"
write(30,510) "    8    20    32    44    56    68    80    92   104
116   128   140   152   164   176   188   200   212   224   236   248
260   272   284   296   308   320   332   344   356   368   380   392
404   416   428   440   452   464   476   488   500   512   524   536
548   560   572   584   596   608   620   632   644   656   668   680
692   704   716   728   740   752   764   776   788   800   812   824
836   848   860   872   884   896   908   920   932   944   956   968
980   992  1004  1016  1028  1040  1052  1064  1076  1088  1100  1112
1124  1136  1148  1160  1172  1184  1196  1208  1220  1232  1244  1256
1268  1280  1292  1304  1316  1328  1340  1352  1364  1376  1388  1400
1412  1424  1436  1448  1460  1472  1484  1496  1508  1520  1532  1544
1556  1568  1580  1592  1604  1616  1628  1640  1652  1664  1676  1688
1700  1712  1724  1736  1748  1760  1772  1784  1796  1808  1820  1832
1844  1856  1868  1880  1892  1904  1916  1928  1940  1952  1964  1976
1988  2000  2012  2024  2036  2048  2060  2072  2084  2096  2108  2120
2132  2144  2156  2168  2180  2192  2204  2216  2228  2240  2252  2264
2276  2288  2300  2312  2324  2336  2348  2360  2372  2384  2396  2408
2420  2432  2444  2456  2468  2480  2492  2504  2516"
```

```
write(30,510) "    9    21    33    45    57    69    81    93   105
117   129   141   153   165   177   189   201   213   225   237   249
261   273   285   297   309   321   333   345   357   369   381   393
405   417   429   441   453   465   477   489   501   513   525   537
549   561   573   585   597   609   621   633   645   657   669   681
693   705   717   729   741   753   765   777   789   801   813   825
837   849   861   873   885   897   909   921   933   945   957   969
981   993  1005  1017  1029  1041  1053  1065  1077  1089  1101  1113
1125  1137  1149  1161  1173  1185  1197  1209  1221  1233  1245  1257
1269  1281  1293  1305  1317  1329  1341  1353  1365  1377  1389  1401
1413  1425  1437  1449  1461  1473  1485  1497  1509  1521  1533  1545
1557  1569  1581  1593  1605  1617  1629  1641  1653  1665  1677  1689
1701  1713  1725  1737  1749  1761  1773  1785  1797  1809  1821  1833
1845  1857  1869  1881  1893  1905  1917  1929  1941  1953  1965  1977
1989  2001  2013  2025  2037  2049  2061  2073  2085  2097  2109  2121
2133  2145  2157  2169  2181  2193  2205  2217  2229  2241  2253  2265
2277  2289  2301  2313  2325  2337  2349  2361  2373  2385  2397  2409
2421  2433  2445  2457  2469  2481  2493  2505  2517"
write(30,510) "   10    22    34    46    58    70    82    94   106
118   130   142   154   166   178   190   202   214   226   238   250
262   274   286   298   310   322   334   346   358   370   382   394
406   418   430   442   454   466   478   490   502   514   526   538
550   562   574   586   598   610   622   634   646   658   670   682
694   706   718   730   742   754   766   778   790   802   814   826
838   850   862   874   886   898   910   922   934   946   958   970
982   994  1006  1018  1030  1042  1054  1066  1078  1090  1102  1114
1126  1138  1150  1162  1174  1186  1198  1210  1222  1234  1246  1258
1270  1282  1294  1306  1318  1330  1342  1354  1366  1378  1390  1402
1414  1426  1438  1450  1462  1474  1486  1498  1510  1522  1534  1546
1558  1570  1582  1594  1606  1618  1630  1642  1654  1666  1678  1690
1702  1714  1726  1738  1750  1762  1774  1786  1798  1810  1822  1834
1846  1858  1870  1882  1894  1906  1918  1930  1942  1954  1966  1978
1990  2002  2014  2026  2038  2050  2062  2074  2086  2098  2110  2122
2134  2146  2158  2170  2182  2194  2206  2218  2230  2242  2254  2266
2278  2290  2302  2314  2326  2338  2350  2362  2374  2386  2398  2410
2422  2434  2446  2458  2470  2482  2494  2506  2518"
write(30,510) "   11    23    35    47    59    71    83    95   107
119   131   143   155   167   179   191   203   215   227   239   251
263   275   287   299   311   323   335   347   359   371   383   395
407   419   431   443   455   467   479   491   503   515   527   539
551   563   575   587   599   611   623   635   647   659   671   683
695   707   719   731   743   755   767   779   791   803   815   827
839   851   863   875   887   899   911   923   935   947   959   971
983   995  1007  1019  1031  1043  1055  1067  1079  1091  1103  1115
1127  1139  1151  1163  1175  1187  1199  1211  1223  1235  1247  1259
1271  1283  1295  1307  1319  1331  1343  1355  1367  1379  1391  1403
1415  1427  1439  1451  1463  1475  1487  1499  1511  1523  1535  1547
1559  1571  1583  1595  1607  1619  1631  1643  1655  1667  1679  1691
1703  1715  1727  1739  1751  1763  1775  1787  1799  1811  1823  1835
1847  1859  1871  1883  1895  1907  1919  1931  1943  1955  1967  1979
1991  2003  2015  2027  2039  2051  2063  2075  2087  2099  2111  2123
2135  2147  2159  2171  2183  2195  2207  2219  2231  2243  2255  2267
2279  2291  2303  2315  2327  2339  2351  2363  2375  2387  2399  2411
2423  2435  2447  2459  2471  2483  2495  2507  2519"
write(30,510) "   12    24    36    48    60    72    84    96   108
120   132   144   156   168   180   192   204   216   228   240   252
264   276   288   300   312   324   336   348   360   372   384   396
```

```
408   420   432   444   456   468   480   492   504   516   528   540
552   564   576   588   600   612   624   636   648   660   672   684
696   708   720   732   744   756   768   780   792   804   816   828
840   852   864   876   888   900   912   924   936   948   960   972
984   996  1008  1020  1032  1044  1056  1068  1080  1092  1104  1116
1128  1140  1152  1164  1176  1188  1200  1212  1224  1236  1248  1260
1272  1284  1296  1308  1320  1332  1344  1356  1368  1380  1392  1404
1416  1428  1440  1452  1464  1476  1488  1500  1512  1524  1536  1548
1560  1572  1584  1596  1608  1620  1632  1644  1656  1668  1680  1692
1704  1716  1728  1740  1752  1764  1776  1788  1800  1812  1824  1836
1848  1860  1872  1884  1896  1908  1920  1932  1944  1956  1968  1980
1992  2004  2016  2028  2040  2052  2064  2076  2088  2100  2112  2124
2136  2148  2160  2172  2184  2196  2208  2220  2232  2244  2256  2268
2280  2292  2304  2316  2328  2340  2352  2364  2376  2388  2400  2412
2424  2436  2448  2460  2472  2484  2496  2508  2520"

i=1
do i=1,2
    write(30,510) "    1         2         3         4         5
6         7"
    write(30,510) "    8         9        10        11        12
13        14"
    write(30,510) "   15"
    write(30,510) "    1         2         3         4         5
6         7"
    write(30,510) "    8         9        10        11        12
13        14"
    write(30,510) "    1         2         3         4         5
6         7"
    write(30,510) "    8         9        10        11        12"
end do
i=1

write(30,*)
write(30,*)

close (30)

100 format(6(2x,es10.4))
110 format(9x,i1)
120 format(4(4x,i1))
130 format(6x,es12.6)
140 format(16(4x,i1))
150 format(4(4x,i1))
500 format(i4,4(1x,i1))
510 format(1x,a)

End PROGRAM MakeNEM
```