HIGH THERMAL CONDUCTIVITY UO_2-BeO NUCLEAR FUEL: NEUTRONIC PERFORMANCE ASSESSMENTS AND OVERVIEW OF FABRICATION

A Thesis

by

MICHAEL JAMES NARAMORE

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

MASTER OF SCIENCE

August 2010

Major Subject: Nuclear Engineering

HIGH THERMAL CONDUCTIVITY UO_2-BeO NUCLEAR FUEL: NEUTRONIC PERFORMANCE ASSESSMENTS AND OVERVIEW OF FABRICATION

A Thesis

by

MICHAEL JAMES NARAMORE

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

MASTER OF SCIENCE

Approved by:

Co-Chairs of Committee,	Sean M. McDeavitt
	Jean C. Ragusa
Committee Members,	Karen Vierow
	Miladin Radovic
Head of Department,	Raymond J. Juzaitis

August 2010

Major Subject: Nuclear Engineering

ABSTRACT

High Thermal Conductivity UO_2-BeO Nuclear Fuel: Neutronic Performance Assessments and Overview of Fabrication. (August 2010)

Michael James Naramore, B.A., Texas A&M University

Co-Chairs of Advisory Committee, Dr. Sean M. McDeavitt
Dr. Jean C. Ragusa

The objective of this work was to evaluate a new high conductivity nuclear fuel form. Uranium dioxide (UO_2) is a very effective nuclear fuel, but it's performance is limited by its low thermal conductivity. The fuel concept considered here is a ceramic-ceramic composite structure containing UO2 with up to 10 volume percent beryllium oxide (BeO). Beryllium oxide has high thermal conductivity, good neutron moderation properties, neutron production from an (n,2n) reaction, and it is chemically stable with uranium at high temperatures. The UO2-BeO fuel concept employs a continuous lattice of BeO within the microstructure of the fuel in order to significantly increase the thermal conductivity of the fuel.

In order to better understand the effect of this fuel concept on reactor operations 2D infinite lattice neutronic simulations for a typical pressurized water reactor fuel assembly were performed using the code DRAGON. Parametric analysis of the beginning of cycle (BOC) effect of BeO and its corresponding temperature increase revealed that the introduction of 5% by volume BeO into UO2 fuel results in a ~400 pcm increase in BOC reactivity, while the 100 K temperature decrease with the introduction of 10% by volume BeO increased the BOC reactivity by ~350 pcm. Cycle length estimates for a PWR were performed with three and four-batch cycles while keeping the uranium-235 mass constant and the introduction of 10% by volume BeO was found to have a ~20 day increase in reactor operation, a 4000-5000 MWd/tHM increase in burnup, and a 2800-2900 pcm increase in BOC reactivity.

A portion of the work documented here includes the establishment of a UO_2-BeO fabrication method with the necessary equipment. The description of a processing vessel is provided and the step-by-step procedures for fabrication are described. The processing vessel has a linear variable differential transducer equipped in order to characterize the sintering behavior.

ACKNOWLEDGEMENTS

To my fiancée and my parents and sister for all of their support while pursuing my Master's Degree. I would also like to acknowledge my professor and committee members, Dr. Sean McDeavitt, Dr. Jean Ragusa, Dr. Karen Vierow, and Dr. Miladin Radovic. I would like to thank Michael Myers for helping me learn Perl and I would like to thank all of my colleagues and the undergraduate assistants who helped me with this project, Jeffrey Hausaman, Aaron Totemeier, Adam Parkinson, Robert Miller, and Spencer Mickum. I'd also like to thank IBC Advanced Alloys and Jim Malone in particular for their advice and help with this project.

TABLE OF CONTENTS

Page

Page

LIST OF FIGURES

Page

Page

LIST OF TABLES

Page

NOMENCLATURE

A	Area [cm^2]
h	Heat Transfer Coefficient [W/m^2-K]
H	Height [cm]
k	Thermal Conductivity [W/m-K]
m	Mass [kg]
M	Molar Mass [g/mol]
N	Atom Density [atoms/cm-barn]
N_A	Avogadro's Number [mol^{-1}]
q'''	Volumetric Heat Generation Rate [W/cm^3]
R	Radius [cm]
T	Temperature [K]
V	Volume [cm^3]
w	Weight Fraction
ρ	Density [g/cm^3]

LIST OF ABBREVIATIONS

BOC	Beginning of Cycle
BOL	Beginning of Life
BWR	Boiling Water Reactor
EOC	End of Cycle
EOL	End of Life
GG	Green Granules
LRM	Linear Reactivity Model
LVDT	Linear Variable Differential Transducer
LWR	Light Water Reactor
PWR	Pressurized Water Reactor
SB	Slug-Bisque
TD	Theoretical Density

1 INTRODUCTION

1.1 The Problem

For the last 40 to 50 years most commercial Light Water Reactors have used uranium dioxide as a fuel form despite its poor thermal properties. Specifically, the thermal conductivity of uranium dioxide is relatively low accounting for the large temperature gradient in the nuclear fuel pellet. This large temperature gradient is responsible for many performance and life limiting phenomena unique to nuclear fuel including fission gas release, void swelling, fuel pellet-cladding interactions, and fuel restructuring. Because of these limiting factors to a typical uranium dioxide fuel, research into ceramic additives aimed to enhance the thermal conductivity of nuclear fuels has been performed with beryllium oxide, silicon carbide, and other materials [1].

Lowering the temperatures inside the fuel pellet through the use of a higher thermal conductivity fuel additive would have the primary effect of lowering the fuel centerline to surface temperature gradient. The reduction of the large temperature gradient in the fuel pellet would also serve to decrease the effect of many of the performance reducing phenomena that occur in the fuel; for example, the amount of void swelling would be reduced and therefore the stress placed on the cladding from fuel deformation would be decreased [2].

This thesis follows the style of the Journal of Nuclear Materials.

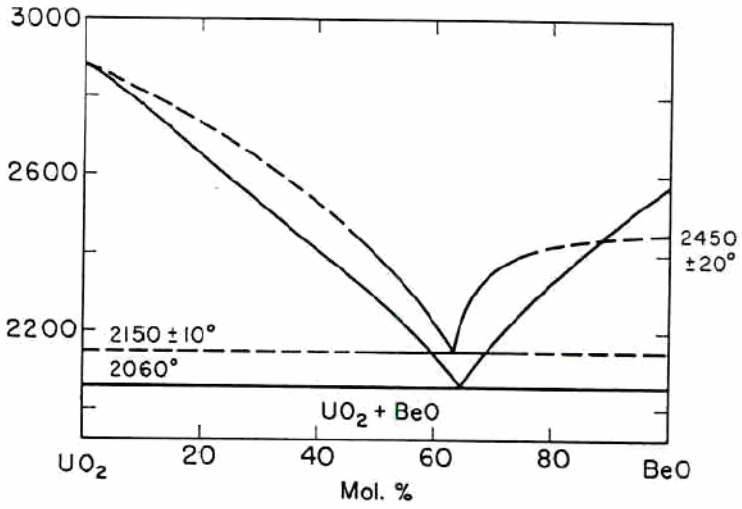


Figure 1.1 The Uranium Dioxide Beryllium Oxide Phase Diagram with the temperature in ^{o}C [3].

Beryllium oxide was chosen for use as a high thermal conductivity nuclear fuel additive because of its exceptionally high thermal conductivity for a ceramic (the highest known thermal conductivity for a non-metal except for diamond) and because of its superior chemical compatibility with uranium dioxide at high temperatures. As shown in Figure 1.1, a uranium dioxide and beryllium oxide mixture would remain unperturbed and in two separate phases up to a temperature of approximately 2060^{o}C, allowing for the formation of this continuous lattice of BeO at high temperatures.

1.2 Thermal Conductivity in a Typical Nuclear Fuel Pellet

1.2.1 Importance of Thermal Conductivity in Nuclear Fuels

The low thermal conductivity of uranium dioxide is the primary cause of the large temperature gradient created in traditional nuclear fuel. This property is the primary contributor to the interesting risk assessment and safety concerns in a nuclear reactor.

The temperature of the fuel has a huge impact on the safety and reactivity of the fuel in the reactor. In an accident scenario, such as a Loss of Coolant Accident (LOCA), the fuel could be left without being properly cooled for some time. A higher thermal

conductivity would decrease the amount of stored energy inside the fuel and therefore decrease the chance of fuel damage during an accident scenario.

An increased thermal conductivity would also serve to decrease the overall temperature in the fuel, which would have a resultant increase in reactivity. These are just a couple examples of how an enhanced thermal conductivity fuel would benefit a commercial power plant.

1.2.2 Thermal Conduction Mechanisms

Thermal conductivity is the measure of a material's ability to conduct heat, and there are three mechanisms through which the transfer of heat is done: phonon scattering, radiation transport, and electron transport (electrical conduction). Equation (1-1) is a general form of the thermal conductivity equation for a uranium dioxide fuel pellet where 'a', 'b', and 'c' are constants that are affected by the different modes of heat transfer.

$$k(T) = (a + bT)^{-1} + cT^3 \qquad (1\text{-}1)$$

Phonon scattering is the most common mode of heat conduction in uranium dioxide at temperatures below 1500^{o}C. The first term of equation (1-1) is the phonon term with the 'a' term as the phonon scattering term and the 'b' term accounting for the contribution of anharmonic interactions between phonons [4], [5], [6].

Heat conduction via radiation transport occurs much more prominently at temperatures above 1500^{o}C-1600^{o}C, as well as heat transfer through electron transport. These contributions explain the increase in thermal conductivity at very high temperatures as shown in Figure 1.2. The 'c' term in Equation (I-1) accounts for the radiative heat transfer, while the electronic contribution is usually thought to have an exponential form. The reason this exponential is not present in most uranium dioxide thermal conductivity equations is because it is difficult to pinpoint its contribution at the high temperatures that it takes effect at as described by Olander and Lambert [5], [6].

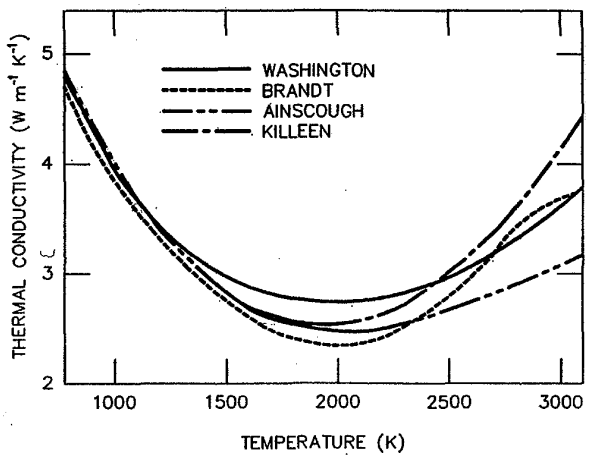


Figure 1.2 Thermal conductivity correlations for 100% TD UO_2 [6].

1.2.3 Determinants of Thermal Conductivity

Generally, there are five major factors that affect the thermal conductivity of a uranium dioxide fuel material: temperature, porosity, oxygen to metal fuel ratio, pellet cracking, and burnup.

1.2.3.1 Temperature

An increase in temperature has been shown to decrease the thermal conductivity of uranium dioxide until 1750^{o}C, after which it increases. A more detailed explanation of temperature's effect on thermal conductivity has already been given in Section 1.2.2.

1.2.3.2 Nonstoichiometric Effects on Thermal Conductivity

1.2.3.2.1 Porosity

Porosity, in general, decreases the thermal conductivity of a solid, making it desirable to minimize it, but in nuclear fuels void swelling is very prominent and porosity helps alleviate internal pressures to reduce fuel deformation. Therefore, a balance between thermal conductivity and fission gas accommodation is necessary to achieve a long lasting fuel [4].

1.2.3.2.2 Oxygen to Metal Fuel Ratio

The oxygen to metal fuel ratio describes the departure of the fuel from a stoichiometric solid. Whether this departure is hyper- or hypostoichiometric makes little difference, as either way the thermal conductivity will be reduced. This reduction in thermal conductivity is due primarily to an increase in phonon-defect interactions because of the deviation from the stoichiometric composition [4], [6].

1.2.3.2.3 Pellet Cracking

Pellet cracking leads to changes in the pellet-cladding contact and pellet-cladding gap conductance as well as altering the internal continuity within the fuel pellet. These changes not only reduce the thermal conductivity of the fuel (because of potential fuel material relocation), but also reduce the conductance to the inner surface of the cladding [4].

1.2.3.2.4 Irradiation and Burnup

Irradiation and burnup of the fuel has many varied effects on the thermal conductivity of the fuel, but overall the uranium dioxide fuel in a LWR becomes hyperstoichiometric as it is burned in the reactor and as a result, the thermal conductivity is reduced. Because of this induced hyperstoichiometric effect, the fuel thermal conductivity decreases because of an increase in phonon-defect interactions much as it does when the oxygen to metal fuel ratio is altered. The presence of fission products and the porosity due to fission gas bubbles also decrease the thermal conductivity [6].

1.2.4 Neutronics Effect of BeO

Beryllium oxide possesses many physical and nuclear characteristics that make it a very attractive material to use in a reactor system. These include its high thermal conductivity, the (n,2n) reaction with ^{9}Be, its chemical inertness with many materials at high temperature, its low capture cross section, its good physical strength, and its good neutron moderation. Because of these properties, the interest in using beryllium oxide in

nuclear reactors has always been high. In fact, beryllium is in some cases used as a neutron reflector in reactors [7].

It is the neutronic effects of beryllium oxide that are of unique interest to this project: the low neutron capture cross-section, the excellent moderating properties, and the (n,2n) reaction. The (n,2n) reaction and the low capture cross-section are both very useful properties for service in a nuclear reactor [7].

The effectiveness of BeO as a moderator arises from its low atomic mass. When neutrons scatter off of a Be atom, they lose on average more energy per collision than if they had scattered off a heavier atom like oxygen or uranium. Another property that makes BeO an effective moderator is the low neutron capture cross-section of Be. Because of this, a neutron is more likely to scatter off a Be atom rather than be absorbed [7].

Beryllium does not have a very high capture cross section at thermal energies, 7.6 mb which is comparable to the oxygen-16 and zirconium capture cross sections at thermal energies, 190 μb and 11.12 mb respectively. While at higher energies the (n,2n) reaction that beryllium possesses has the higher cross section of 269.1 mb at 14 MeV neutron energies. On the other hand, this fast flux contribution by the (n,2n) reaction is offset by the buildup of Li-6 and He-3 from the ensuing (n,α) reactions that take place afterwards [7].

1.3 High Conductivity Concept Description

Research into beryllium oxide fuel additives for nuclear fuels have been performed in the past, but it is the continuous UO_2-BeO ceramic network that has been shown to enhance the fuel's thermal conductivity most significantly. Other fuel concepts that using beryllium oxide (or other high thermal conductivity materials) have used a dispersal technique to uniformly disperse microspheres of beryllium oxide throughout a uranium dioxide fuel pellet. The continuous microstructure design primarily consists of UO_2 microspheres (nominal diameters between ~45 μm and ~500 μm) embedded in a

continuous matrix of UO_2 and BeO (nominal diameter ~1 μm). This continuous microstructure is illustrated in Figure 1.3 [1], [8], [9], [10].

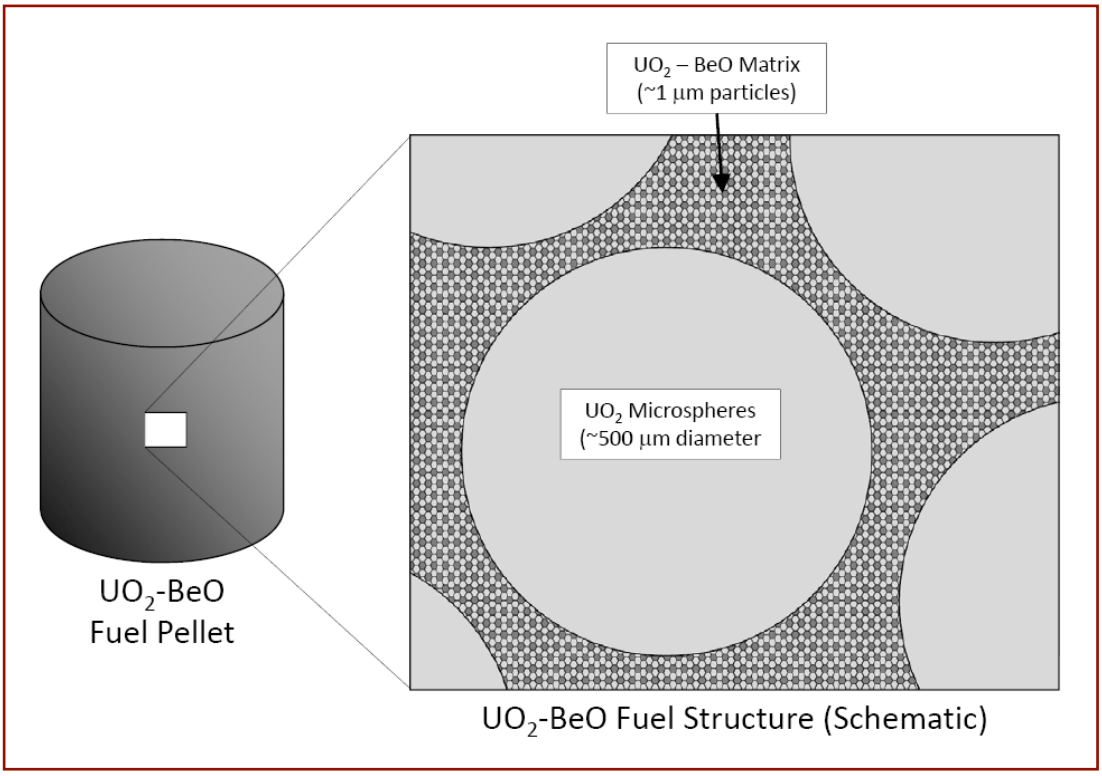


Figure 1.3. UO_2-BeO Fuel Concept [1].

1.4 Overview

The following Sections summarize the parallel activities[1] carried out to establish fabrication and neutronic simulation methods for the UO_2-BeO concept, Section 2 provides a summary of previous nuclear fuel concepts that have used BeO additives, other high conductivity fuel additives such as silicon carbide, and previous approaches to modeling thermal conductivity for fuel performance models. Appendix J describes the establishment of a UO_2-BeO fabrication method including a step-by-step procedure and a description of a processing vessel with a linear variable differential transducer to characterize the sintering behavior of the UO_2-BeO concept. The objectives of this were to quantify the neutronic behavior of the UO_2-BeO fuel in a typical pressurized water reactor using various batch cycle length estimates.

[1] The original objective of the research program sponsored by IBC Advanced Alloys Corp. was to establish and scale up the materials fabrication method described in Appendix J. However, program urgencies caused a change in emphasis. The fabrication work was halted and neutronic modeling became the primary focus of the study. The information in Appendix J is included to document the work accomplished, but the neutronic study in Sections 3 and 5 represent a complete body of research.

2 BACKGROUND

2.1 Beryllium Oxide Past Uses

Beryllium oxide has been used in concept reactors such as the Daneils Reactor and the Experimental Beryllium Oxide Reactor (EBOR). The EBOR was a gas-cooled reactor (Figure 2.1) that came about during research in the 1950s for aircraft nuclear propulsion [7], [11], [12].

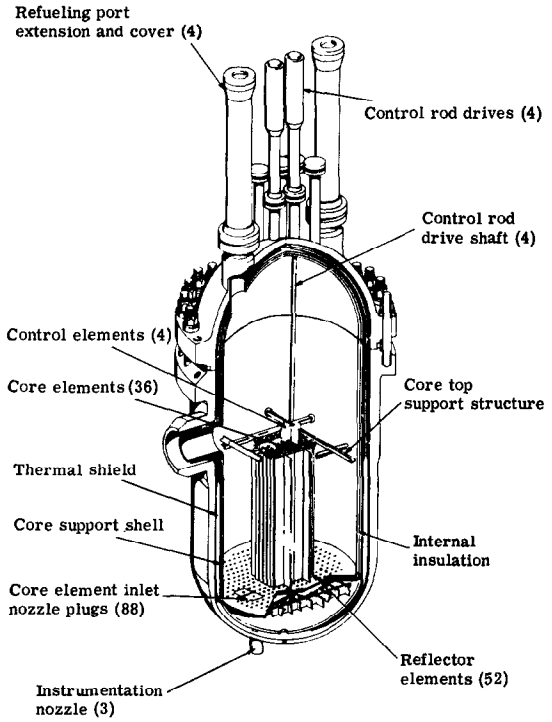


Figure 2.1 EBOR reactor core and pressure vessel schematic [11].

The first major experiment in this aircraft nuclear propulsion endeavor was the Aircraft Reactor Experiment (ARE). This reactor had BeO as a reflector and a moderator, and was liquid-cooled and liquid-fueled. In fact, BeO blocks from the Daniels Reactor were used to fabricate parts for the ARE core; BeO was selected for the ARE core because it

exhibited little to no reaction with the NaK coolant and also because of its superior thermal properties. Since using BeO in the ARE, data on its performance in reactor core situations has been developed in many situations [7], [11].

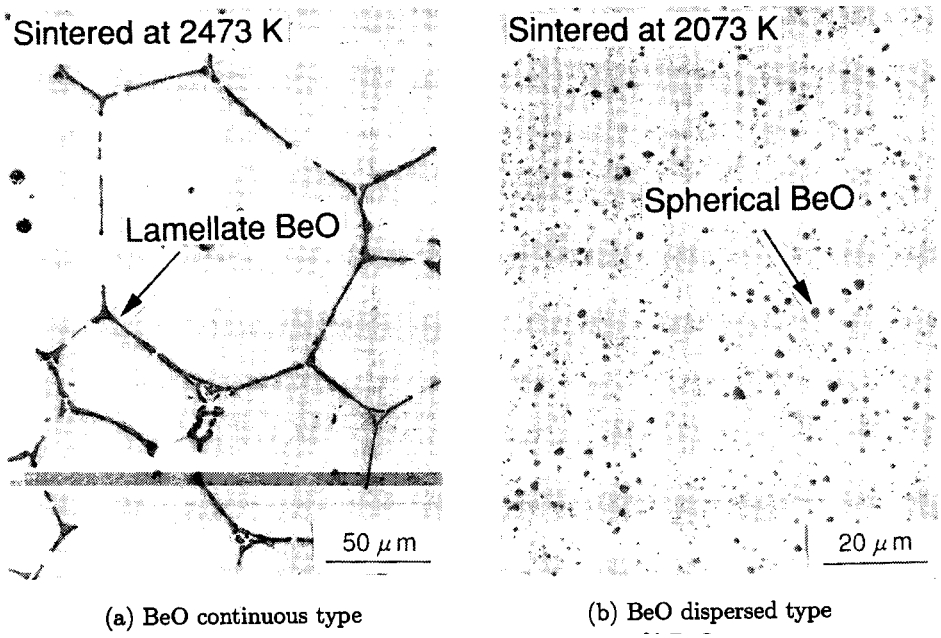
Around this same time, the Atomic Energy Commission was looking to develop a gas-cooled reactor for commercial use. Many materials were considered for the moderator in this Maritime Gas-Cooled Reactor (MGCR) including beryllia, but eventually it was decided that graphite would be best because of the expense of beryllia. Later in the project, after problems with graphite and more favorable data on beryllia had emerged, the decision was made to use BeO as the moderator and the reflector for the MGCR [7], [11].

The next step in the project was to create a prototype reactor: the EBOR. The fuels used in this reactor are pressed dispersions of (U, Th)O_2 with a BeO matrix; the nominal composition was 70% BeO to 30% fuel. Irradiation tests were completed on these types of fuels to test their viability [13], [14].

In one experiment, two types of dispersions were used: a coarse dispersion (100-200 micron fuel particles), and a fine dispersion (50 micron fuel particles). Both of these dispersions were put under thermal irradiation to about 55% burnup. Swelling in most of these tests was very small (around 1% or less in most cases), and there was little change in the microstructure, although some fission gas bubbles were observed [13], [14], [15].

2.2 New BeO Concept Fuels

More recently, much development into BeO as a fuel additive to a typical LWR uranium dioxide fuel pellet has been done. Two types of fuels were tested in one experiment, labeled a BeO continuous type, and a BeO dispersed type (Figure 2.2). In the continuous type, the BeO is precipitated almost continuously along the grain boundary while in the dispersed type the spherical BeO particles are dispersed into the matrix of UO_2. The continuous type was found to have higher thermal conductivity, although the difference between them lessened as the temperature increased [16].



(a) BeO continuous type

(b) BeO dispersed type

Figure 2.2 Ceramographs of UO_2-0.9wt% BeO pellets for the continuous (a) and dispersed (b) types [16].

2.3 Silicon Carbide as a High Conductivity Additive

Solomon et al. explored the feasibility of increasing the thermal conductivity of oxide fuels by the addition of a second, higher thermal conductivity solid phase. The two high conductivity additive considered are silicon carbide and beryllium oxide. For the SiC, the Polymer Impregnation and Pyrolysis (PIP) process is adapted for use with nuclear fuels. The PIP process is normally used to make ceramic matrix composites, and the purpose of the matrix phase for these is to improve its loading capacity whereas for nuclear fuels, it will be used to maximize density and purity which will increase thermal conductivity [8], [10], [17], [18].

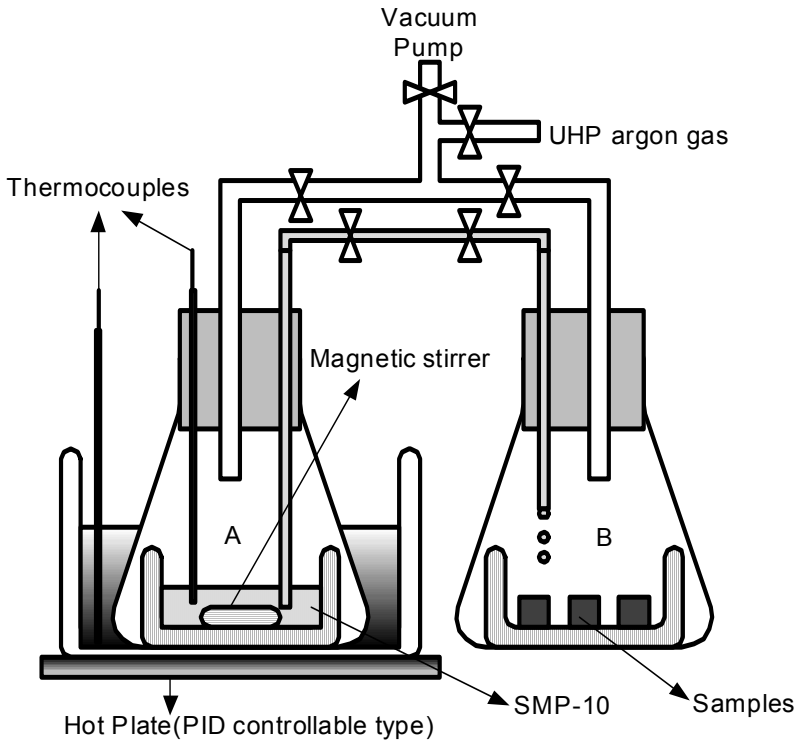


Figure 2.3 Vacuum impregnation method for the PIP process [10].

In order to use the PIP process properly, there was a need to first have a high density and easily impregnated UO_2 pellet. This was produced by pre-compacting the UO_2 at high pressure, granulating followed by sieving, and then pre-sintering the granules as shown in Figure 2.4. Next, the pre-sintered granules are pressed and sintered into their final form.

The vacuum impregnation, as illustrated in Figure 2.3, is followed by the very necessary curing and crystallization steps. During the curing stage, the volume of the impregnated SiC is decreased by approximately 70%, necessitating a cyclic process of impregnation, curing, and crystallization.

There is a reaction between UO_2 and SiC at 1370^{o}C and it was found that this reaction will happen independent of the processing methodology used to restrict the CO or SiO gases. All processing, therefore, must take place below this temperature. Because of

this low temperature restriction and the cyclic nature of the PIP process, only a 75% TD SiC phase after 6 to 9 cycles was achieved. Unfortunately, no increase in thermal conductivity was achieved [8], [10].

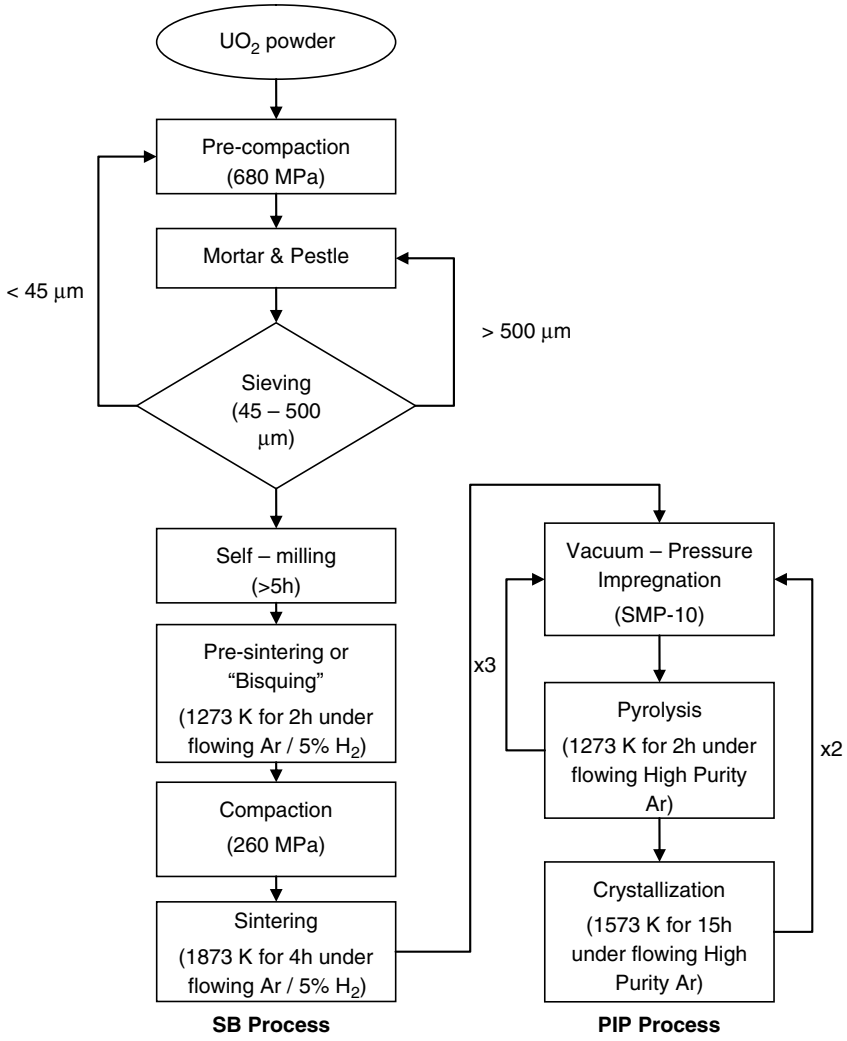


Figure 2.4 The fuel fabrication process for UO_2-SiC fuels [9].

This lack of thermal conductivity improvement stems from the low crystallization temperature used in the PIP process. As discovered in later experiments, the difference

in thermal conductivity from crystallization at 1300°C and 1500°C is very significant, because of phonon scattering in the very fine porosity [8], [10].

Following the SiC work, two methods of producing a continuous lattice of BeO additive in UO_2 pellets were explored: the Slug-Bisque method, and the Green Granules method. The slug-bisque method uses pre-sintered granules of uranium dioxide coated with BeO and pressed into pellets to create a separate stable continuous lattice of BeO, while the green granules method uses green granules coated in BeO and pressed. The green granules method yielded superior thermal conductivity enhancement [8], [10], [19].

2.4 Thermal Conductivity and Fuel Performance Modeling

Computational analysis of these high conductivity fuels is very important in order to predict the thermal properties of these concept nuclear fuels. Using ANSYS, the general methodology is to create a geometry for the system, define material properties in the geometries, choose elements and mesh, and then solve the model. For the BeO fuel type, a 2-D grid pattern was used to represent the microstructure of the UO2-BeO fuel. As shown in Figure 2.5, the UO2 (lighter phase) is surrounded by the BeO (darker phase) in varying amounts, depending on the volume percent of BeO in the fuel. Using the grain sizes reported from Ishimoto et al. these models resulted in thermal conductivity curves very similar to the experimentally derived curves measured by Ishimoto et al [8], [16].

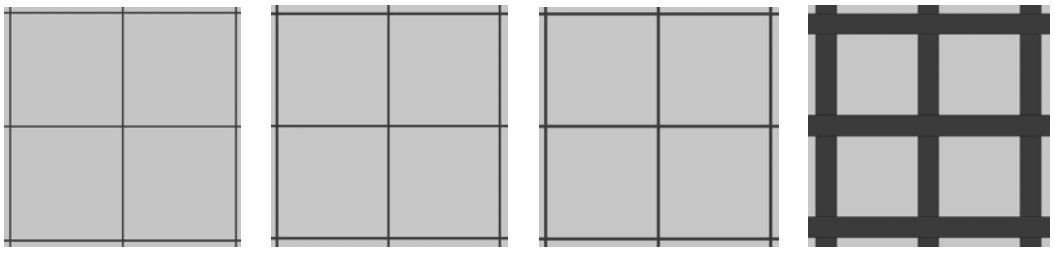


Figure 2.5 The ANSYS computational model geometry for a UO_2-BeO fuel [8].

Another experiment using statistical continuum mechanics focused instead on the difference between isotropic and anisotropic microstructures in nuclear fuels and how to model and predict their effect on the thermal properties of the fuel. It is thought that an enhanced thermal conductivity can be achieved in the radial direction through the use of an anisotropic microstructure. As shown in Figure 2.6, the lefthand plot is of material properties and the righthand plot is of microstructure development and these two sets of data are linked to one another. The microstructural designs achieved throughout the fuel processing and the corresponding properties of these microstructures need to be determined in order to fully utilize this materials design approach [20].

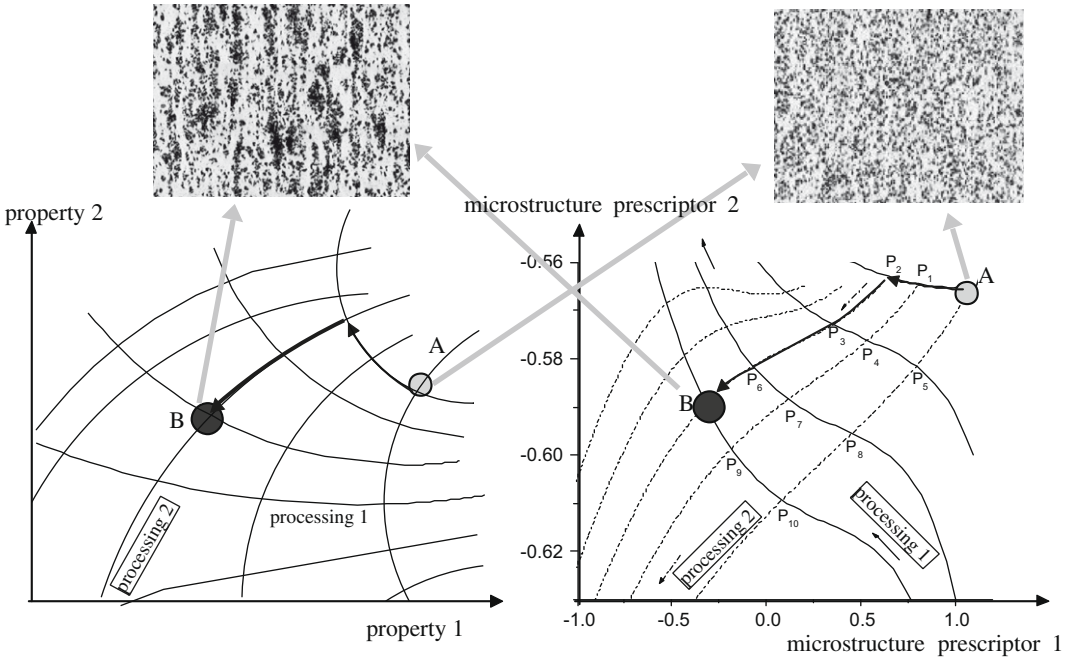


Figure 2.6 Illustration of the link between properties and microstructural development during furl processing [20].

Areva's proprietary COPERNIC code was used to calculate the fuel rod performance for these high conductivity fuels. COPERNIC is designed for use with commercial fuels in mind, and therefore does a full thermo-mechanical analysis of the fuel pin including rod

manufacturing characteristics, irradiation conditions, power histories, and thermal-hydraulics [19].

Because COPERNIC was not designed for use with experimental fuels, challenges with using this high conductivity fuel arose, including how to model the effect of the BeO in the fuel. Figure 2.7 shows the method used to properly simulate the BeO additive in COPERNIC; the amount of UO_2 in the two fuels are different depending on the amount of BeO in the concept fuel, but the power generated by the UO_2 is kept constant. So the UO_2-BeO fuel must be taken to higher burnup in order to extract the same power from it as the pure UO_2 fuel [8], [10], [19].

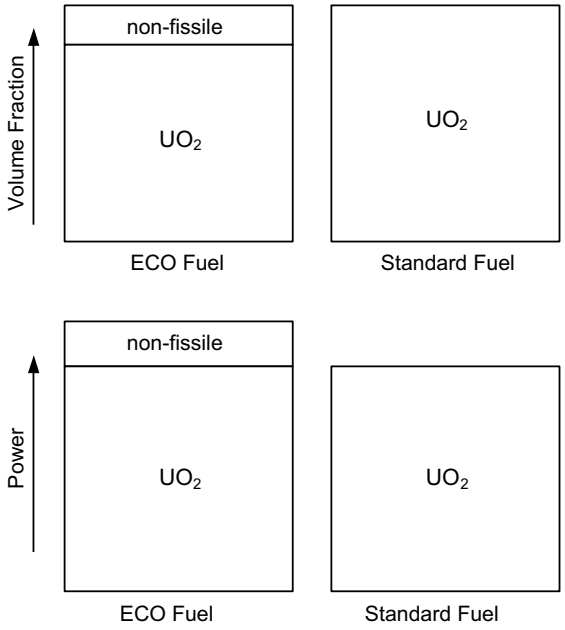


Figure 2.7 Schematic comparison of volume fraction to linear power generation between the UO_2-BeO fuel and the typical UO_2 fuel [19].

3 NEUTRONICS

3.1 Mechanics

3.1.1 DRAGON

3.1.1.1 The DRAGON Code

The DRAGON computer code was developed at *École Polytechnique de Montréal* as a comprehensive solver for the neutron transport equation. It is a single code with various numerical techniques and calculation methods to solve the neutron transport equation. The organizations responsible for the development and support of this include: at *École Polytechnique de Montréal*, Hydro-Quebec and the Hydro-Quebec chair in nuclear engineering, the Natural Science and Engineering Research Council of Canada (NSERC), the Atomic Energy of Canada limited (AECL), and the CANDU Owners Group (COG).

This code was also developed with the implementation of new models and algorithms in mind: the modular structure of DRAGON easily allows for the use of these new models and algorithms by separating the many calculations into modules that are linked by a GAN generalized driver. Some of the modules perform calculations including resonance self-shielding, the analysis of various geometries, generation of tracking files for collision probability evaluation, multigroup collision probability integration, solving the multigroup neutron transport equation using the collision probability method or the method of characteristics, and isotopic depletion [21].

3.1.1.2 The Input Deck

In DRAGON, many of the main calculations have been separated into individual execution modules that communicate with corresponding data structures. For the purposes of this, there are three important modules that control the majority of the variation in the code executions: the LIB module, the GEO module, and the EVO

module. Each of these modules are used to generate the MICROLIB, GEOMETRY, and BURNUP data structures respectively [21].

3.1.1.2.1 The LIB Module and MICROLIB Data Structure

The LIB module is used to generate the MICROLIB data structure for DRAGON that stores the microscopic and macroscopic cross sections for communication between the DRAGON modules. The LIB modules can read from many different microscopic cross-section libraries and this data can be formatted in several different manners. The microscopic cross-sections for each isotope are interpolated over temperature and dilution and then these cross sections are multiplied by their corresponding atomic concentrations to produce the internal MICROLIB that will be used in the neutronic calculations.

The LIB module separates its material compositions into 'mixtures' that have spatial locations specified by the GEO module. These mixtures are a list of isotopes (or natural elements as the case may be); each isotope has a specific atom density (atoms/cm-barn) and a set of values to specify how self-shielding will be handled. Also, a temperature is required for each mixture for accurate broadening of the neutron cross sections.

For the purpose of generating DRAGON decks, the fuel composition has been parameterized using the uranium-235 enrichment and the beryllium oxide content. When the BeO content changes it alters the value of the all the atom densities in the fuel as well as changes the effective temperature of the fuel because BeO improves thermal conductivity of the fuel. When the uranium-235 enrichment changes, only the uranium-235 and uranium-238 atom densities change. In this parametric study, there is no change in the coolant or cladding atom densities or temperatures.

3.1.1.2.2 The GEO Module and GEOMETRY Data Structure

Simply, the GEO module is used to generate or modify a geometry for use in DRAGON. This code uses the GEO module to create several simple fuel pin geometries and combines them into an eighth of an assembly, with reflective boundary counditions.

This reflected eighth of an assembly is shown in Figure 3.1 as a fuel nuclear fuel assembly.

Another very useful function of the GEO module is creating the discretization scheme for within the fuel pellet. The normal discretization chosen is 50%, 30%, 15%, and 5% of the pellet volume from the center of the fuel pellet. This is done in order to follow the isotopics in the subvolumes separately during depletion.

The main variable that would be changed within the GEO module would be the radius of the fuel pellet and from that, the exact radial distances of the discretization of said fuel pellet.

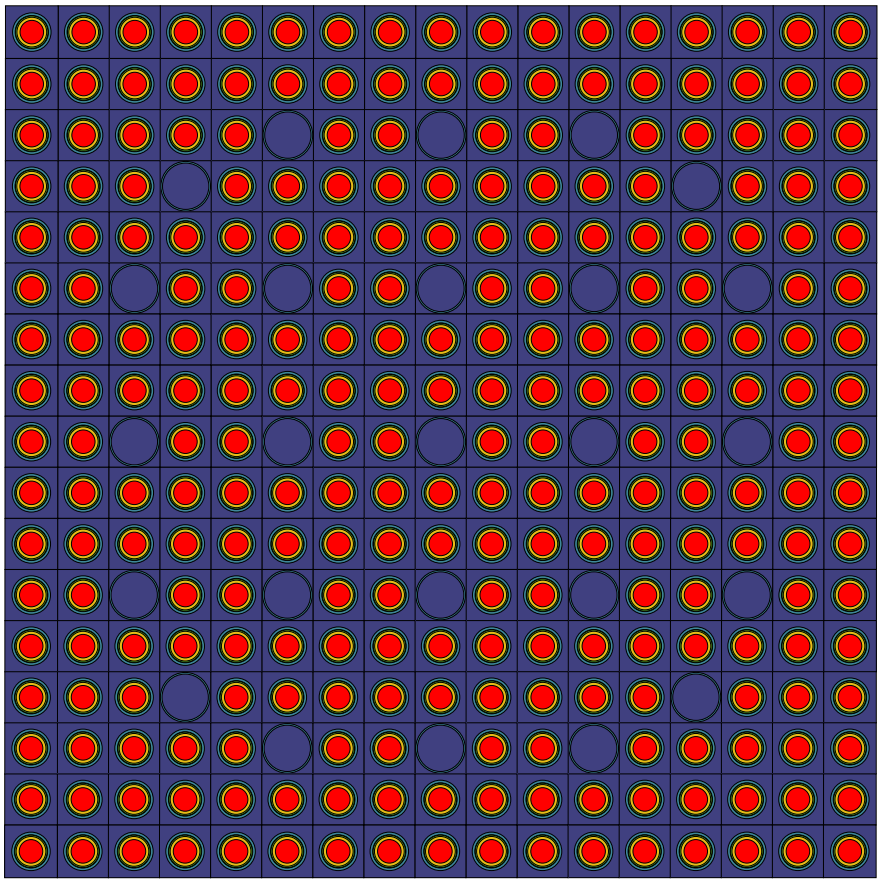


Figure 3.1 Fuel assembly geometry for a 2D infinite lattice simulation.

3.1.1.2.3 The EVO Module and BURNUP Data Structure

The EVO module defines the burnup steps that DRAGON will take. The step lengths may be constant or they may change as fuel is burned. For example, small burnup steps could be taken close to the beginning of life while the steps get progressively longer the further out the fuel is burned. This allows for a fine mesh to be used only in the area of interest, which shortens computational time.

3.1.2 Logic and Input

This DRAGON will be a simulation of a 2-dimensional infinite lattice of $1/8^{th}$ of a PWR fuel assembly. Because this is an infinite lattice, neutron leakage from the core is estimated using an approximate 3.5% leakage. These approximations are sufficient for bulk depletion calculations.

Using this eighth of a PWR fuel assembly test DRAGON deck as a baseline, certain values had to be altered depending on the nature of the simulation. The power of the core, material properties of the mixtures, the geometry, and the burnup scheme are typical values that had to be changed between various simulations. Neglecting the commented sections of each DRAGON deck that change with each run, the set of parameters that allowed the power of the core, material properties, geometry, and burnup to be changed are as follows:

- Specific Power
- Mixture Atom Densities
- Mixture Temperatures
- Fuel Pellet Radius and Discretization
- Cladding Outer Radius
- Burnup Steps

These values are updated and tracked for every DRAGON run. The specific power is a ratio of the total thermal power of the core to the total heavy metal mass in the core. The specific power mainly changes when BeO is introduced into the fuel. BeO displaces fuel (and heavy metal mass) and therefore the specific power needs to be increased in the fuel to yield the same linear heat rate. As shown in equation (3-1), the burnup of the fuel is directly related to the specific power of the reactor and the time spent in the reactor.

$$BU = P_{specific} \cdot t \tag{3-1}$$

The mixture atom densities are dependent upon many factors, but the two most common ones are uranium-235 enrichment and beryllium oxide content. These two factors affect the fuel mixture atom densities, while the only the geometry affects the cladding atom density and the coolant temperature affects the coolant atom densities.

The burnup steps used vary from simulation to simulation depending upon what is being studied. For example, if the beginning of life core reactivity across various initial criteria is what is being studied, then there will be very few (if any) burnup steps, in order to speed up computational time. If a measure of the longevity of the fuel is desired, then many burnup steps will be necessary.

3.1.2.1 What is held constant and Why?

In order to have a basis of comparison between simulations, some parameters need to stay constant. Typical values for a standard Pressurized Water Reactor are used in the majority of the simulations done, and if a simulation varies any of these values it will be explicitly noted.

- Fuel Theoretical Denisty = 95%
- BeO Density = 2.34 g/cm^3
- UO_2 Density = 10.97 g/cm^3
- Fuel Pellet Radius = 0.41cm

- Cladding Inner Radius = 0.418cm
- Cladding Outer Radius = 0.475cm
- Number of Assemblies in Core = 193
- Number of Fuel Pins per Fuel Assembly = 264
- Active Height of the Core = 3.66m
- Coolant Temperature = 575 K
- Coolant Pressure = 155 bars
- Amount of Boron in Coolant = 0ppm
- Linear Heat Rate = 17.8 kW/m

Many of these values are left constant because they represent a typical PWR, but some of them are left constant for other reasons. The amount of boron in the water is left at zero because the simulations wish to discover what sort of reactivity differences there are between normal fuel types and fuel types with BeO additions before such factors as burnable poisons, boron content in the water and fuel assembly shuffling are taken into account. Also, at the end of life in a reactor, the boron content is equal to zero. The linear heat rate is left constant because otherwise the total core power would change as the BeO content changed, resulting in comparisons between reactor of different power ratings.

3.1.2.2 Input

Many values may be changed or altered for these simulations, but the most common values to change are related to the fuel:

- Uranium-235 Content (wt%)
- Beryllium-Oxide Content (vol%)
- Effective Temperature of the Fuel (K)

$$T_{eff} = \frac{5}{9}T_{CL} + \frac{4}{9}T_{fs} \qquad (3\text{-}2)$$

The effective temperature of the fuel pellet is as shown in equation (3-2), with T_{eff} as the effective temperature, T_{CL} as the centerline or highest temperature in the fuel pellet, and T_{fs} as the temperature of the fuel pellet surface.

The effective temperature of the fuel is dependent upon the BeO content seeing as how the BeO increases the thermal conductivity of the fuel resulting in a decrease in temperature in the fuel pellet. Unfortunately, a detailed analysis of the temperature and BeO content dependent thermal conductivity for a UO_2-BeO fuel is not yet available. There are correlations for pure uranium dioxide, and experimentally derived correlations for specific percentages of BeO in a UO_2-BeO fuel, but for now estimates must be made using existing data.

3.1.2.3 Calculations for the Input

There are many parameters that need to be known in order to solve for the atom densities and specific power that DRAGON needs to run. Most of these equations are either ratios, or can be derived from a combination of equations (3-2)-(3-3).

Equation (3-3) describes how to calculate the molar mass, M, of a single atom of many isotopes. Each isotope has a molar mass of M_i and a corresponding weight percentage of w_i.

$$\frac{1}{M} = \sum_{i=1}^{I} \frac{w_i}{M_i} \qquad (3\text{-}3)$$

The molar mass for a molecule as shown in equation (3-4) (or composition of various atoms) is just the sum of the constituent molar masses M_j.

$$M = \sum_{j=1}^{J} M_j \qquad (3\text{-}4)$$

The atom density N in equation (3-5), is equal to the density of the material multiplied by Avogadro's number over the molar mass of the material.

$$N = \frac{\rho N_A}{M} \tag{3-5}$$

Equation (3-6) describes the dilution factor for the cladding with V_{clad} as the volume of the cladding and V_{gap} as the volume of the gap. Instead of modeling the fuel-cladding gap in the fuel pin, a dilution factor is added to the cladding. This factor reduces the density of the cladding and then it is just assumed that the cladding inner surface is equal to the surface of the fuel pellet.

$$D_{clad} = \frac{V_{clad}}{V_{clad} + V_{gap}} \tag{3-6}$$

Equations (3-7)-(3-13) are expansions of equation (3-5) for the isotopes present in the coolant, cladding, and fuel. All of these atom densities have the typical density, N_A, and molar mass similar to all atom densities, but each has ratios as well that are highly dependent upon what the user decides as the amount of BeO and UO_2.

$$N_{h1} = 2\frac{\rho_{H_2O} N_A}{M_{H_2O}} \tag{3-7}$$

$$N_{o16water} = \frac{\rho_{H_2O} N_A}{M_{H_2O}} \tag{3-8}$$

Equations (3-7) and (3-8) are the atom densities for the coolant water and they are really only dependent upon the density of the water which is temperature dependent.

$$N_{zr91} = D_{clad}\frac{\rho_{zr} N_A}{M_{Zr}} \tag{3-9}$$

Equation (3-9) is the atom density of the cladding with the dilution factor added.

$$N_{u235} = \left(1 - \frac{f_{BeO}}{100}\right)\left(\frac{\frac{w_{u235}}{100}}{\frac{M_{u235}}{M_U}}\right) TD_{UO_2} \frac{\rho_{UO_2} N_A}{M_{UO_2}} \tag{3-10}$$

$$N_{u238} = \left(1 - \frac{f_{BeO}}{100}\right)\left(\frac{1 - \frac{w_{u235}}{100}}{\frac{M_{u238}}{M_U}}\right) TD_{UO_2} \frac{\rho_{UO_2} N_A}{M_{UO_2}} \tag{3-11}$$

Equations (3-10)-(3-13) are the atom densities for the isotopes in the fuel: U-235, U238, Be-9, and O-16. Equations (3-10) and (3-11) are the atom densities for the uranium with f_{BeO} as the volumetric fraction of BeO in the fuel. Equation (3-12) is the atom density for beryllium.

$$N_{be9} = \left(\frac{f_{BeO}}{100}\right)\frac{\rho_{BeO} N_A}{M_{BeO}} \tag{3-12}$$

Equation (3-13) is the equation for the oxygen atom density in the fuel. It has to take into account the oxygen coming from both the UO_2 and the BeO, hence the two terms.

$$N_{o16\,fuel} = 2 \cdot \left(1 - \frac{f_{BeO}}{100}\right) TD_{UO_2} \frac{\rho_{UO_2} N_A}{M_{UO_2}} + \left(\frac{f_{BeO}}{100}\right)\frac{\rho_{BeO} N_A}{M_{BeO}} \tag{3-13}$$

Equation (3-14) is the equation for the heavy metal mass (uranium) of the entire reactor core with V_{core} as the volume of the reactor core.

$$m_{HM,core} = \frac{V_{core}\rho_{UO_2}}{1000}\frac{M_U}{M_{UO_2}}\left(1 - \frac{f_{BeO}}{100}\right) \tag{3-14}$$

$$P_{specific} = P_{total}\left(\frac{1000}{m_{HM,core}}\right) \tag{3-15}$$

And equation (3-15) is the equation for the specific power of the reactor with P_{total} as the total thermal power of the reactor. This specific power and the atom densities for

coolant, cladding, and fuel isotopes are some of the most important parameters for successful simulations in DRAGON.

3.1.3 Perl Automation

Because many of these DRAGON simulations needed to be run, a Perl script was created to calculate the necessary values for each DRAGON deck as well as create and run these decks. Perl was chosen because of its affinity for regular expressions, which allows for simple search and replace functions within text documents.

Since the DRAGON decks are ASCII files, a template DRAGON deck was created with search tags placed where all the requisite data needed to be. The Perl script then calculates the necessary values for the DRAGON deck and searches through the template deck for the corresponding search code and replaces the search code with the correct data that the deck requires to run successfully.

This is a simple pseudo-code describing the general function of the Perl script:

```
Program Initialization
        -Data Organization
        -Data Initialization
        -Variable Data Arrays
        Loop Initialization
                -Initialize Changes
                -Calculate Dependents
                -Create Replacement Hash
                -Copy and Rename Template File
                -Search Through Copy and Replace
                -Queue up Copy on Grove
        Loop End
Program End
```

The specific Perl and shell scripts used in the construction of the DRAGON decks and their subsequent analysis are located in Appendices A, B, C, and D, with the DRAGON template deck in Appendix E.

3.1.4 Cycle Length and the Linear Reactivity Model

In this paper, three and four-batch cycle strategies are explored using the linear reactivity model to simplify the reactivity calculations. Equation (3-16) describes the mixture reactivity of an n-batch cycle, with f_i as the fraction of core power delivered by assembly i, and ρ_i is the reactivity of assembly i. When f_i is equal for all batches, the equation simplifies into an average over the number of batches [22].

$$\rho_s = \sum_{i=1}^{n} f_i \rho_i \qquad (3\text{-}16)$$

The cycle length estimates simulated in this paper were made under the assumption of equal power in all batches, more intricate models include unequal power sharing.

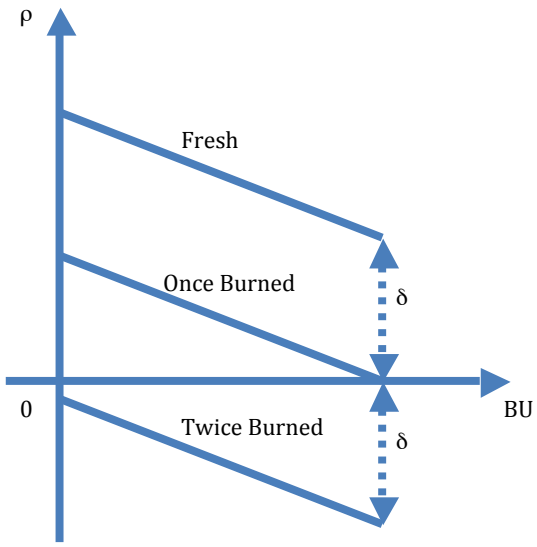


Figure 3.2 Simultaneous burnup schematic for a 3-batch cycle strategy.

The linear reactivity model describes a 3-batch cycle as shown in Figures 3.2 and 3.3. There is fresh fuel, once burned fuel, and twice burned fuel. Figure 3.2 illustrates the burnup that each type of assembly receives, fresh fuel starts and ends with excess

reactivity, twice burned fuel starts and ends with negative reactivity, while the once burned fuel is in between, usually reaching EOC with close to zero reactivity.

Approximating this type of batch strategy is illustrated in Figure 3.3. The equivalent core average burnup is calculated for a fuel assembly using equation (4-2). Now solve for the uranium-235 enrichment for which the intersection of this equivalent core average burnup and the EOC reactivity (k=1). This is the average fuel assembly that will reach EOC for a 3-batch cycle [22].

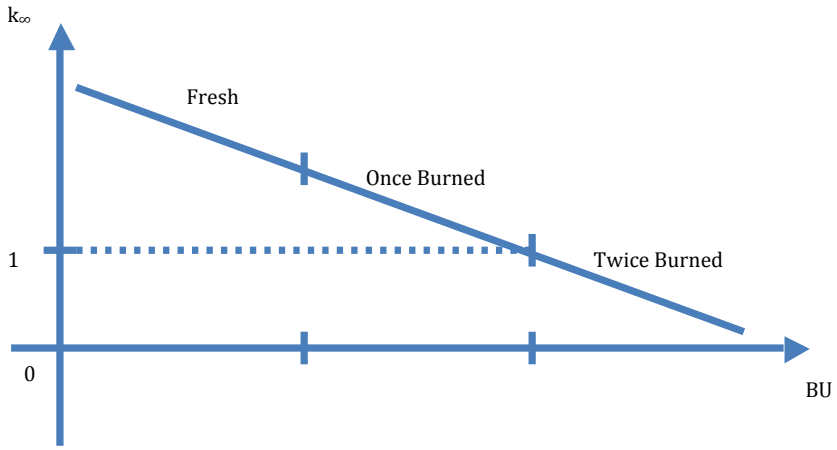


Figure 3.3 Burnup schematic for a 3-batch cycle strategy.

It is important to note that for the cycle length estimates in this paper, k_∞=1 is not EOC. These simulations are 2D infinite lattice simulation of $1/8^{th}$ of a fuel assembly and therefore, do not take core leakage into account. In order to account for core leakage, k_∞ is increased to 1.035 (3.5% neutron leakage).

3.1.5 Thermal Conductivity and T_{eff}

The thermal conductivity of the fuel pellet for these computational simulations presents unique challenges because of the limited knowledge (both empirically or theoretically) on the temperature dependent thermal conductivity of UO_2-BeO fuel pellets. For

example, the temperature dependent thermal conductivity of a typical UO2 fuel pellet may be described using equation (3-17), with temperature in K and the thermal conductivity in W/m-K [4].

$$k(T) = \frac{1}{11.8 + (0.0238)T} + (8.775e-13)T^3 \qquad (3\text{-}17)$$

This correlation is very accurate within the temperature range of interest for a fuel pellet in a typical PWR, but the challenge lies in adding beryllium to the mix. There is very limited data on the thermal conductivity of UO_2-BeO fuels, and that information that does exist is only for specific volume percentages of BeO.

Knowing the temperature dependent thermal conductivity is very important for accurate DRAGON calculations because it needs to know the effective temperature of its mixtures. The water and cladding thermal conductivities need not be known in so much detail because the temperature gradient across these are very small compared to the temperature gradient in the fuel pellet.

The effective temperature is used to represent the neutronic temperature inside the fuel. This temperature takes into account self-shielding, and is the temperature at which the neutron cross sections are broadened at. As shown in equation (3-1), the two parameters that T_{eff} depends on are the centerline temperature and the surface temperature of the fuel.

The surface temperature of the fuel pellet can be calculated from the average temperature of the coolant in the core, T_{H2O}, and the heat generation rate density, q”’. The heat generation rate density as shown in equation (3-18) is simply the total power of the reactor, P_{total}, over the total volume of the core, V_{core}.

$$q''' = \frac{P_{total}}{V_{core}} \qquad (3\text{-}18)$$

Next are two sets of simplified heat conduction equations. The first describes the temperature difference between the coolant and the outer surface of the cladding. The

second describes the temperature difference between the outer and inner surfaces of the cladding. If the gap were being taken into account in these calculations, then a third equation would be used for the temperature drop across the gap.

Equation (3-19) describes the temperature increase from the coolant to the outer surface of the cladding, with T_{H2O} as the average temperature of the coolant, R_{fp} as the fuel pellet radius, h_{conv} as the heat transfer coefficient of the coolant, and $R_{clad,o}$ as the outer radius of the cladding.

$$T_{clad,outer} = T_{H_2O} + \frac{q''' R_{fp}^2}{2 h_{conv} R_{clad,o}} \tag{3-19}$$

Equation (3-20) describes the temperature increase across the cladding, with $T_{clad,outer}$ from equation (3-19), $R_{clad,i}$ as the inner radius of the cladding, and k_{clad} as the average thermal conductivity of the cladding.

$$T_{clad,inner} = T_{clad,outer} + \frac{\ln\left(\frac{R_{clad,o}}{R_{clad,i}}\right) q''' R_{fp}^2}{2 k_{clad}} \tag{3-20}$$

$$T_{fs} = T_{clad,inner} + \frac{q''' R_{fp}}{2 h_{gap}} \tag{3-21}$$

The temperature increase across the gap is as shown in equation (3-21) with h_{gap} as the heat transfer coefficient of the gap and the temperature of the inner surface of the cladding coming from equation (3-20).

$$T_{CL} = T_{H_2O} + \Delta T_{film} + \Delta T_{clad} + \Delta T_{gap} + \Delta T_{fuel} \tag{3-22}$$

Equation (3-22) shows the relationship between equations (3-19) through (3-21). Starting with the known average temperature of the coolant, the centerline temperature is just equal to the sum of the temperature increases across the different sections of the fuel element. In order to get the temperature difference across the fuel pin, ΔT_{fuel}, the non-

linear heat conduction equation must be solved inside the fuel pin as shown in equation (3-23).

Newton's method
- -Initial Guess
- Loop Initialization
  - -Calculate function value at Guess
  - -Calculate function derivative's value at Guess
  - -Formulate New Guess
  - -Check for Convergence with Old Guess
  - -Loop Until Convergence
- Loop End

The process of Newton's method starts with an initial guess; the closer this guess is to the actual answer, the quicker Newton's method will converge. Also, if the initial guess is to far away from the actual answer, then Newton's method could never converge depending upon the specific non-linear equation in question.

The next step takes place inside a loop; this loop does not stop until convergence occurs. Equation (3-23) is the conductivity integral equation for a cylindrical fuel pellet rearranged to be set equal to zero. The *k(T)* function may be any valid correlation for UO_2 thermal conductivity with R_i being the radius at which the temperature is to be calculated at.

$$\int_{T_{fs}}^{T_i} k(T)dT - \left(\frac{q'''}{4}\left(R_{fp}^2 - R_i^2\right)\right) = 0 \qquad (3\text{-}23)$$

Equation (3-24) is the function that Newton's method will solve. Newton's method solves for the zeroes of a given function, so if an initial guess of the temperature of the coolant is used, then the value of this function may be calculated. The next step is to calculate the value of this function's derivative at that same initial guess.

$$f(T_i) = \int_{T_{fs}}^{T_i} k(T)dT - \left(\frac{q'''}{4}\left(R_{fp}^2 - R_i^2\right)\right) \qquad (3\text{-}24)$$

From these values, equation (3-25) may be used to calculate a new guess T_{i+1} with T_i being the last guess.

$$T_{i+1} = T_i - \frac{f(T_i)}{f'(T_i)} \tag{3-25}$$

The next step is to check for convergence by checking the absolute value of the difference between the current and previous guess. If this value is less then the chosen convergence criteria (e.g. 10^{-10}), then the method has converged, the loop ends and the method returns the temperature for the given radial distance from the center of the pellet.

If R_i is set to zero (i.e. the center of the fuel pellet), then the centerline temperature can easily be solved with this method and the effective temperature calculated with the use of equation (3-2). The temperature profile inside of a fuel element calculated using this method is shown below in Figure 3.4. The effective conductivity of the gap and the convective heat transfer coefficient for the coolant used in these calculations were 10^4 W/m^2-C and 3.6×10^4 W/m^2-C respectively. The dimensions of the fuel pin used are listed in section 3.1.2.1.

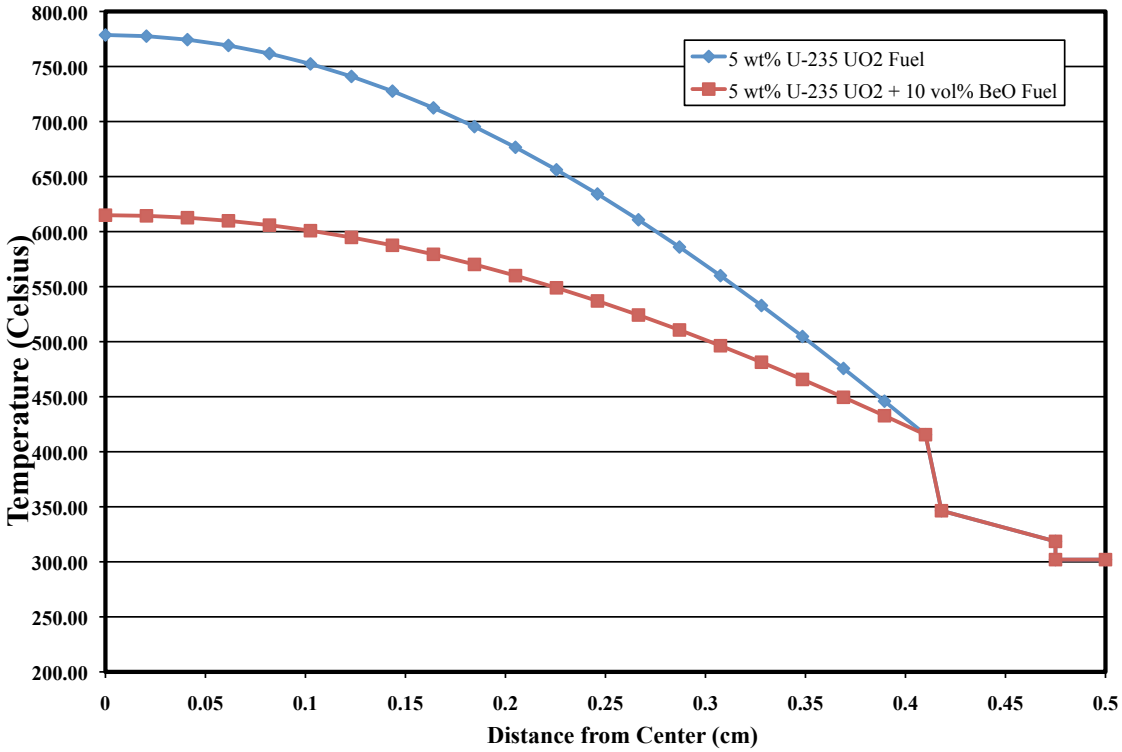


Figure 3.4 Temperature profile inside of a fuel element for UO_2 and UO_2-10 vol% BeO fuels.

4 RESULTS AND DISCUSSION

4.1 Parametric Variation

As shown in Figure 4.1 the thermal conductivity of a typical UO2 fuel pellet is highly dependent upon the effective temperature of the fuel pellet. Both correlations used in the plot are in very close agreement, especially over the temperature range of interest. The first correlation is equation (3-16) from Todreas and Kazimi, and the second correlation is a curve fit to the experimental data shown in Fink's Thermophysical Properties of Uranium Dioxide Review [4], [23], [24].

The final plot is a curve fit to the experimental measurements taken from the UO_2-10 vol% BeO pellet created by Solomon et al. The thermal conductivity for the UO_2-BeO fuel is significantly higher at the lower range of temperatures, but this difference decreases as the temperature increases. Even so, the thermal conductivity for the UO_2-10 vol% BeO pellet is still greater then twice the thermal conductivity of a typical UO_2 pellet. (Solomon et al. 2002)

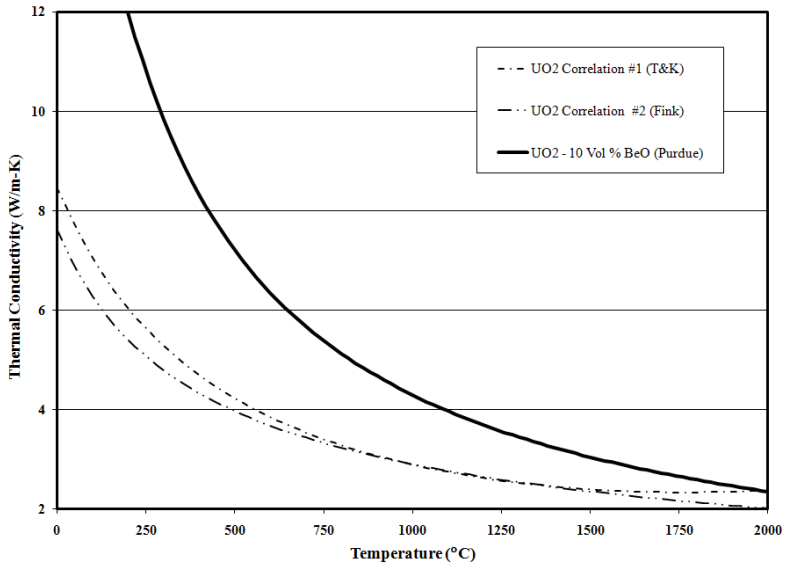


Figure 4.1 Comparison of thermal conductivity correlations for UO_2 from Todreas and Kazimi, and Fink with the Phase 1 correlation developed for UO_2-10 vol% BeO [1].

This difference in fuel thermal conductivity and temperature is very important to understanding phenomenologically what is happening inside the fuel pellet. What difference does the BeO have within the fuel and what difference does the fuel temperature decrease that the BeO creates have.

In these parametric variations the enrichment of the fuel is kept at a constant 5 wt% while the BeO content is varied, resulting in three curves per plot. In each plot, the temperature within the fuel is treated differently: in Figure 4.2 the temperature of the fuel is altered in a linear manner by the presence of the BeO, while in Figure 4.3 the temperature of the fuel is not altered by the presence of BeO.

It is important to note that because the BeO content changes and the enrichment is held constant, that the amount of U-235 in the fuel is not constant across varying amounts of BeO.

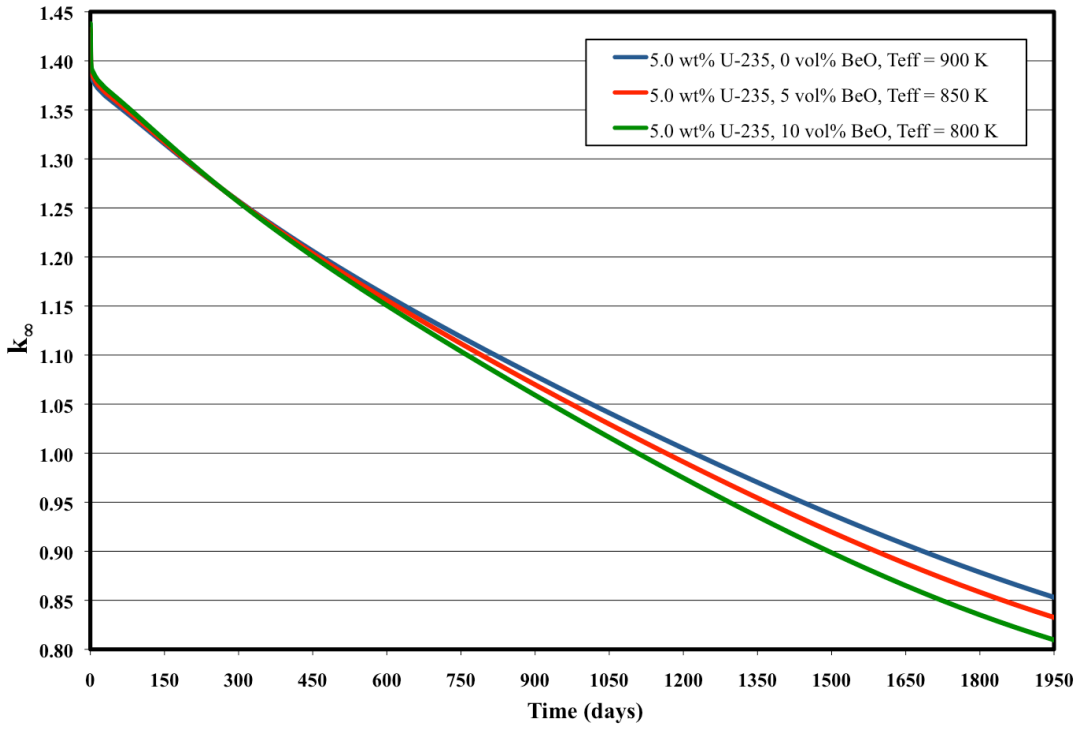


Figure 4.2 Plot of reactivity vs time with changing BeO content assuming a linear temperature relation.

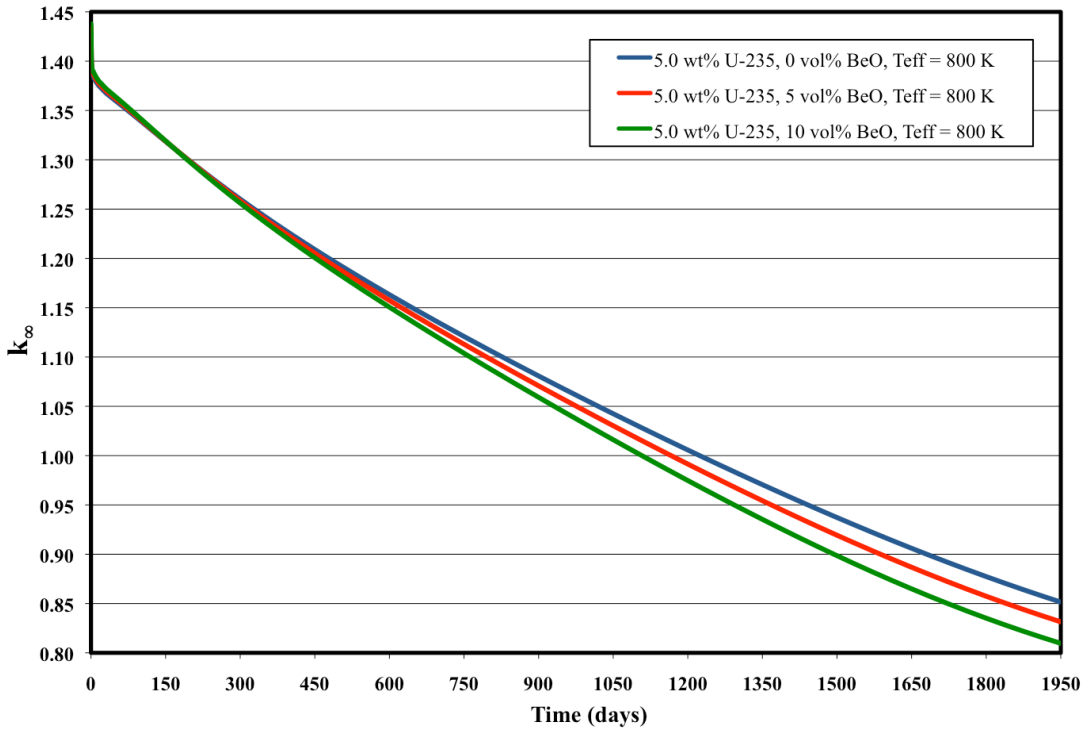


Figure 4.3 Plot of reactivity vs time with changing BeO content assuming a constant temperature relation.

These parametric variations study the same three scenarios with different temperature effects. The thermal output of the fuel is kept constant and so when the BeO is added in increasing amounts, the specific power of the fuel increases accordingly. The uranium-235 enrichment is kept constant between these cases (not the mass). The main change is the temperature effect of the BeO.

In Figure 4.2 the temperature effect of the BeO is taken into account and decreases the fuel effective temperature by 50 K for each 5 vol% BeO added to the fuel. In the second case, this temperature decrease is ignored. In this way, the neutronic effect of just the BeO may be better isolated from its temperature decrease in the fuel.

Figures 4.2 and 4.3 are very similar and are hard to differentiate visually, so Tables 4.1 and 4.2 have several important data points of the two plots in order to more easily see what is happening.

Table 4.1 Several data points from Figure 4.3.

	T_f = 800K	T_f = 800K	T_f = 800K
Time (days)	0vol% BeO	5vol% BeO	10vol% BeO
(BOC) 0	1.43007	1.43396	1.43788
1	1.39356	1.39630	1.39905
2	1.38699	1.38974	1.39248
200	1.29828	1.29790	1.29724
(EOC) 1950	0.85147	0.83144	0.80964

The introduction of BeO into the fuel increases the reactivity at BOC regardless of the temperature reduction that BeO imparts to the fuel. This is because of the increased moderation and perhaps in small part to the (n,2n) reaction of Be. Now, if the 0 vol% and 5 vol% BeO cases for BOC in Table 4.1 are compared, then it is obvious that an approximate 400 pcm increase in BOC reactivity is produced by the presence of the BeO. And since the temperatures in this chart are being artificially held constant, this increase in reactivity is just due to the presence of the BeO and not its temperature increase.

If the BOC 0 vol% BeO cases from Table 4.1 and Table 4.2 are compared then an approximate 350 pcm increase for a 100 K decrease in temperature, effectively the fuel temperature reactivity coefficient (which, as shown in the next section, matching very well with the calculated fuel temperature coefficients).

Taking into account the effect BeO has on the temperature of the fuel, and reactivity is increased even more. The effect of this BOC reactivity increase is reduced outside of the xenon transient (the first day or two of reactor operation) and as the fuel is depleted more, this increase in reactivity eventually disappears and there is a cross over in the plot, as shown in Figures 4.2 and 4.3. At this cross over, the reactivity of the BeO fuel type falls below the reactivity curve of the pure UO2 fuel. This happens as early as it does because the uranium-235 enrichment is being held constant so the mass of uranium-235 is not constant across increasing BeO cases. This cross over happens because of the

higher specific power in the UO_2-BeO fuel, so although this fuel starts out higher in reactivity then the normal UO_2 fuel, the higher rate of burnup catches up eventually. This is why there is less reactivity in the UO_2-BeO fuel at EOC regardless of the temperature effect in the fuel.

Table 4.2 Several data points from Figure 4.2.

	T_f = 900K	T_f = 850K	T_f = 800K
Time (days)	0vol% BeO	5vol% BeO	10vol% BeO
(BOC) 0	1.42662	1.43199	1.43788
1	1.39024	1.39441	1.39905
2	1.38369	1.38785	1.39248
200	1.29510	1.29609	1.29724
(EOC) 1950	0.85305	0.83253	0.80964

4.2 Temperature Reactivity Coefficients

The temperature reactivity coefficients in a nuclear reactor are very important to the overall stability of the reactor. These coefficients are a measure of the change in the reactivity of the reactor per degree temperature change in whatever material is being discussed. So the fuel temperature reactivity coefficient is the degree to which to reactivity of the reactor changes for every degree change in fuel temperature. These values are another good way to benchmark ones results, because even though a fuel additive is being simulated, the reactivity coefficients should still be in the same relative ballpark otherwise something is amiss.

These reactivity coefficients were calculated by running the DRAGON deck at low burnup and without the normal xenon transient at BOL. The simulation is run with the fuel/moderator at typical temperature and at 10 K higher temperature. The reactivity coefficient, as shown in equation (4-1), is equal to the change in k over the corresponding change in temperature.

$$\alpha \cong \frac{\partial k}{\partial T} \cong \frac{k(T_1) - k(T_2)}{T_1 - T_2} \qquad (4\text{-}1)$$

Table 4.3 Fuel temperature coefficients for various U-235 enrichments and BeO contents in [pcm/K].

		U235 wt%		
Fuel Temp (K)	BeO vol%	4	4.5	5
900	0	-3.42	-3.44	-3.47
850	5	-3.7	-3.72	-3.73
800	10	-3.68	-3.71	-3.73

Table 4.3 show the temperature reactivity coefficients for the fuel and Table 4.4 shows the temperature reactivity coefficients for the moderator. There are coefficients for three separate enrichments of uranium-235, and for three separate amounts of BeO.

Table 4.4 Moderator temperature coefficients for various U-235 enrichments and BeO contents in [pcm/K].

		U235 wt%		
Fuel Temp (K)	BeO vol%	4	4.5	5
900	0	-54.620	-56.570	-58.050
850	5	-52.660	-54.730	-56.320
800	10	-50.450	-52.660	-54.380

There is very little difference between these reactivity coefficients across three different enrichments and three different BeO contents. As uranium-235 enrichment increases the fuel temperature coefficients only increase very slightly, while they increase a little more when the BeO content is increased. The moderator coefficients on the other hand increase by approximately the same amount whether the BeO content or the uranium-235 enrichment increases.

These temperature coefficients also agree with typical PWR temperature reactivity coefficients as shown by Housiadas [25]. Typical reference values for fuel and moderator temperature coefficients (with zero boron in the water for the moderator coefficient) are -3.0 pcm/K and -30.0 pcm/K respectively. This shows that the fuel temperature reactivity coefficients are right on the reference value, while the moderator temperature coefficients are still very close, just not as close as the fuel coefficients. The stated range of variation for both of these reactivity coefficient references is about -1.0 to -6.0 pcm/K for the fuel, and -10.0 to -60.0 pcm/K for the moderator, placing all of the values in Tables 4.3 and 4.4 within the range of variation for a typical PWR.

4.3 Flux Spectra

In Figure 4.4, the neutron flux spectrum for uranium-235 mass equivalent cases is shown with the flux in units of lethargy. The UO_2 case is enriched to 4 wt% uranium-235, and the UO_2-BeO case has 4.44 wt% uranium-235 and 10 vol% BeO. In Figure 4.4, there are the thermal and fission Maxwellian spectra and the fast flux has inelastic scattering resonances clearly visible. The slowing down region follows the pattern of 1/E or is constant in lethargy with the effect of resonance absorptions apparent, especially for low lying resonances.

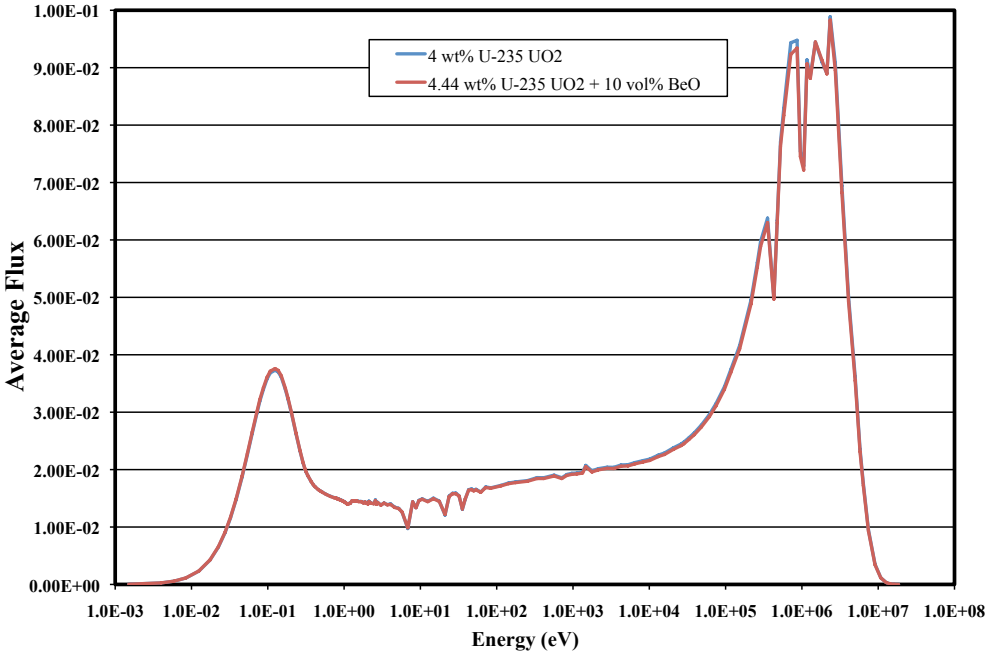


Figure 4.4 Neutron flux spectra comparison for uranium-235 mass equivalent cases.

As shown, there is very little visible difference in the two cases, so Figure 4.5 was created in order to better see the difference that BeO creates in the flux spectrum (this figure is the no BeO case subtracted from the 10 vol% BeO case). Beryllium oxide increases the thermal flux and decreases the resonance absorption in the slowing down region. It also significantly decreases the fast flux, except for one point around 1.84 MeV. This increase in fast flux corresponds to the (n,2n) reaction of Be-9.

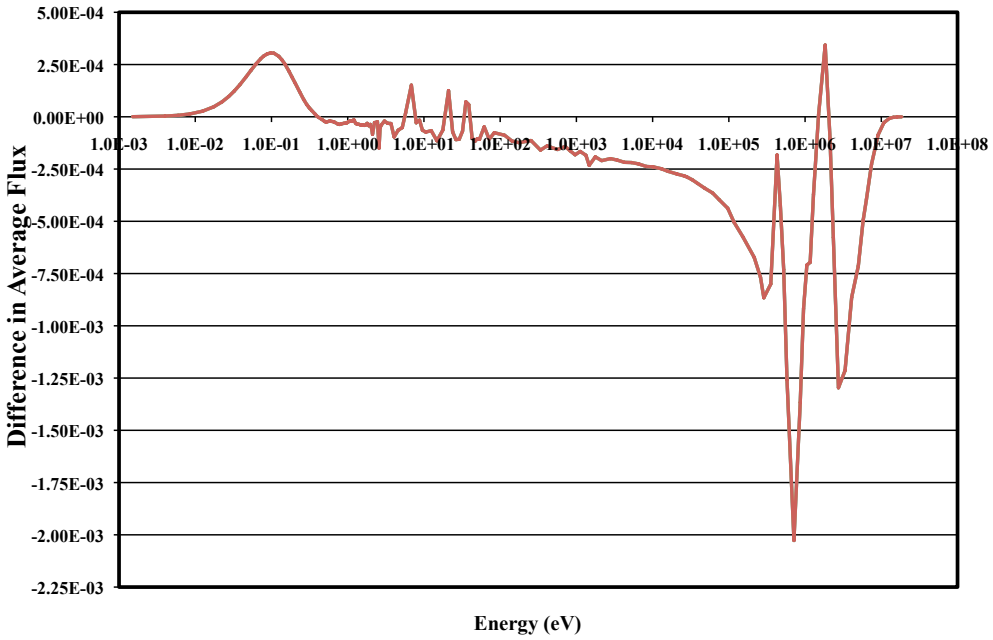


Figure 4.5 Neutron flux spectrum difference between the two cases in Figure 4.4 (ϕ_{BeO} - ϕ_{UOX}).

4.4 Uranium-235 Mass Equivalence Studies

In these studies, the mass of the uranium-235 is kept constant by keeping the atom density of uranium-235 constant between each scenario. So in each scenario, the thermal power output is kept constant and the addition of BeO in the fuel displaces uranium. This uranium displacement has the effect of increasing the specific power and increasing the burnup in the fuel as a result. Two burnup schemes were simulated: a three-batch cycle and a four-batch cycle. In each cycle, every batch has an equal length of 15 GWd/tHM, as shown in Figure 4.10 and 4.13. The core average burnup at EOC for the three-batch cycle is 30 GWd/tHM, and 37.5 GWd/tHM for the four-batch cycle.

$$B_{avg} = \left(\frac{n+1}{2n}\right)B_1 \qquad (4\text{-}2)$$

This average burnup is calculated from equation (4-2), with n being the number of batches, and B_1 being the discharge burnup of the assembly. For each cycle, the

uranium-235 enrichment to reach EOL with a k_∞=1.035 (which is unity plus the approximate 3.5% neutron leakage from the core) is calculated.

Because the uranium-235 mass is being held constant in these studies, the addition of BeO also increases the enrichment of the fuel (in order to keep the U-235 content the same).

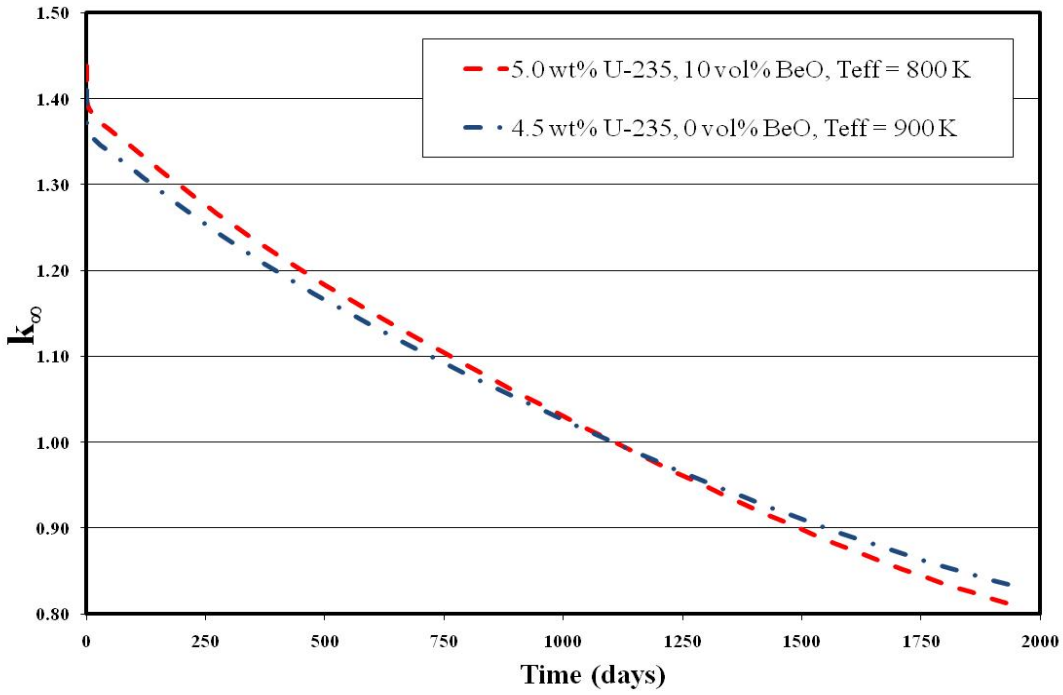


Figure 4.6 Uranium-235 mass equivalence example.

Essentially, these mass equivalence studies compare two scenarios: a fuel with the BeO additive, and a fuel without the BeO additive, but in both cases there is an equal amount of uranium-235 atoms. In this way, the effect of the BeO is more easily apparent.

In this type of comparison, the first step is to decide on an enrichment of uranium-235 for either the pure UO_2 case or the UO_2-BeO case. At this point, just equate their atom densities and solve for the unknown enrichment. Figure 4.6 shows a mass equivalence

study starting with 5 wt% U-235 in the UO_2-10 vol% BeO fuel and then solving for the UO2 fuel enrichment (~4.5 wt% U-235).

4.4.1 Three-Batch Strategy

This 3-batch cycle is the first of two cycle length estimates. The mass of uranium-235 will be kept constant between a UO_2 and a UO_2-10 vol% BeO fuel. This 3-batch cycle consists of 3 equal 15 GWd/tHM batches, with an average assembly burnup of 30 GWd/tHM as shown in Figure 4.7.

$$3\ Batch\ Cycle:$$
$$0-15 \rightarrow 15-30 \rightarrow 30-45 \rightarrow Avg = 30\ GWd/tHM$$

Figure 4.7 The three-batch strategy.

For the three-batch cycle, using the linear reactivity model, an enrichment of 3.86 wt% is required to reach the average burnup of 30 GWd/tHM. Since BeO replaces 10% of the fuel, approximately 10% more uranium-235 is required to achieve equivalent amounts of uranium-235 in each case. The increase in uranium-235 enrichment is slightly more then 10% because slight changes in the molar mass of uranium when the enrichment changes as seen in equation (3-10). The enrichments of the 10 vol% BeO case was calculated with the use of equation (3-10) to be 4.29 wt%.

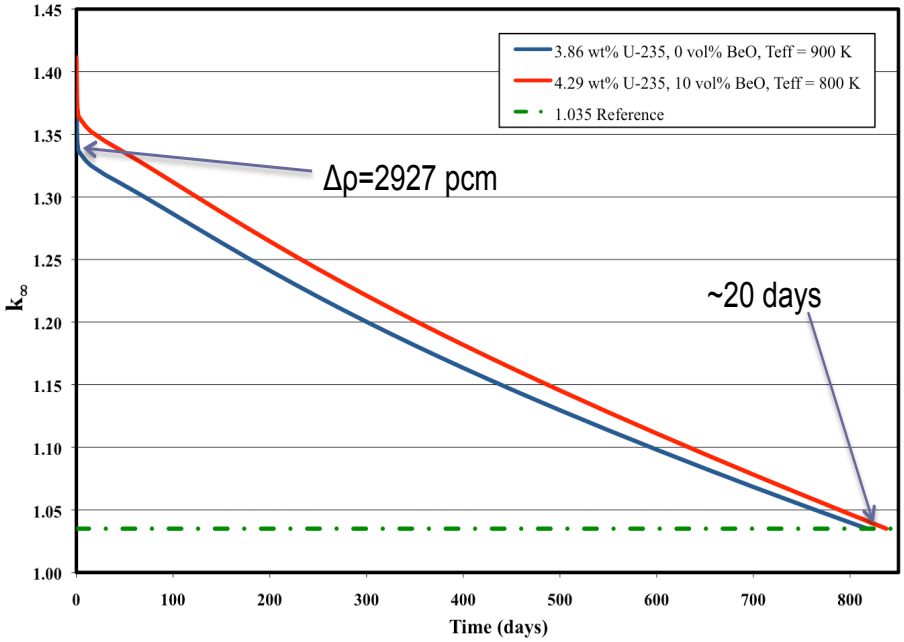


Figure 4.8 Reactivity vs time plot for U-235 mass equivalent cases with a 30 GWd/tHM average burnup.

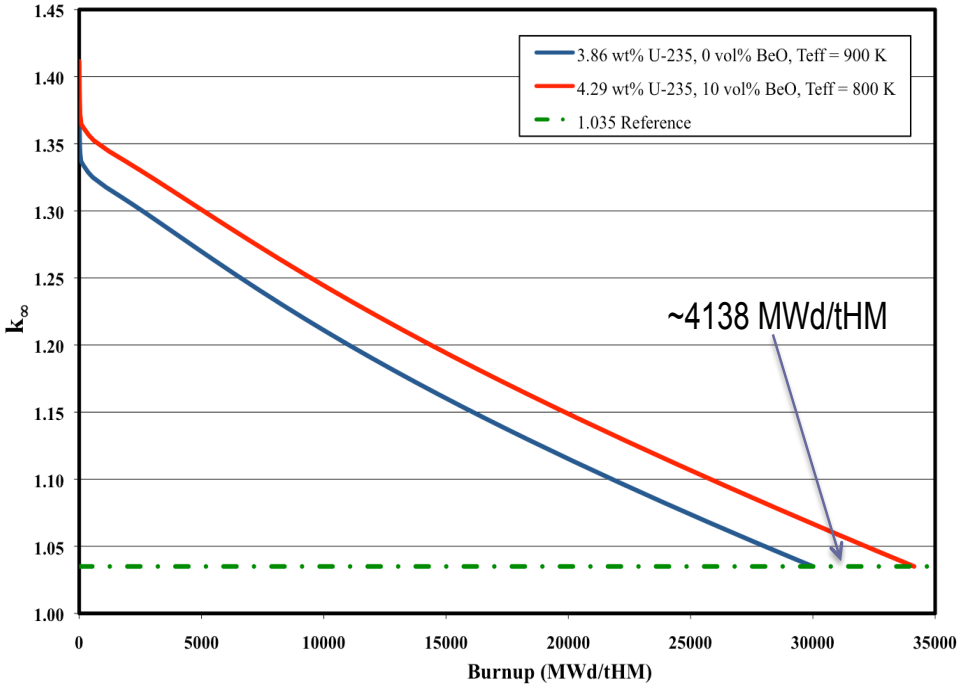


Figure 4.9 Reactivity vs burnup plot for U-235 mass equivalent cases with a 30 GWd/tHM average burnup.

Now, comparing the UO_2 case to the UO_2-10 vol% BeO case, as shown in Figures 4.8 and 4.9, has some interesting results: an increase in BOC reactivity, more operational time before EOC, and extra burnup to the fuel. The 10 vol% BeO case has an increase in BOL reactivity of 2927 pcm. This reactivity increase occurs because of the good neutronic properties of BeO including its high thermal conductivity that decreases the temperature in the fuel, its good neutron moderation, and its (n,2n) reaction. This increase in reactivity is about a 2.12% increase from the typical UO_2 case.

This higher reactivity also results in 20 extra days of reactor operation prior to reaching EOL (at k_∞=1.035) for this three-batch cycle, which directly translates to greater fuel economy. Now, because of the greater time spent in the reactor and the higher specific power for this UO_2-BeO fuel, the fuel also receives an extra 4138.68 MWd/tHM equivalent core average burnup for the 10 vol% BeO case. This means that this average fuel assembly is at an equivalent core average burnup of 34.1 GWd/tHM instead of 30 GWd/tHM with the 10 vol% BeO additive.This increase in burnup is great for new, modern fuels that wish to achieve higher burnup of the fuel and therefore higher uranium utilization. The percentage increase to the burnup and operational time are 13.8% and 2.4% respectively.

The discharge burnup of a fuel assembly from this 3-batch strategy is 45 GWd/tHM for the typical UO_2, while it is approximately 51 GWd/tHM for the 10 vol% BeO case. This is a significant increase in EOL burnup, about 6 GWd/tHM.

4.4.2 Four-Batch Strategy

This 4-batch cycle is the second of two cycle length estimates. The mass of uranium-235 will be kept constant between a UO_2 and a UO_2-10 vol% BeO fuel. This 4-batch cycle consists of 4 equal 15 GWd/tHM batches, with an average assembly burnup of 37.5 GWd/tHM as shown in Figure 4.10.

$$4\ Batch\ Cycle:$$
$$0-15 \rightarrow 15-30 \rightarrow 30-45 \rightarrow 45-60 \rightarrow Avg = 37.5\ GWd/tHM$$

Figure 4.10 The four-batch strategy.

For the four-batch cycle, using the linear reactivity model, an enrichment of 4.76 wt% is required to reach the average burnup of 37.5 GWd/tHM. Since BeO replaces 10% of the fuel, approximately 10% more uranium-235 is required to achieve equivalent amounts of uranium-235 in this case. The enrichments of the 10 vol% BeO case was calculated with the use of equation (3-10) to be 5.29 wt%.

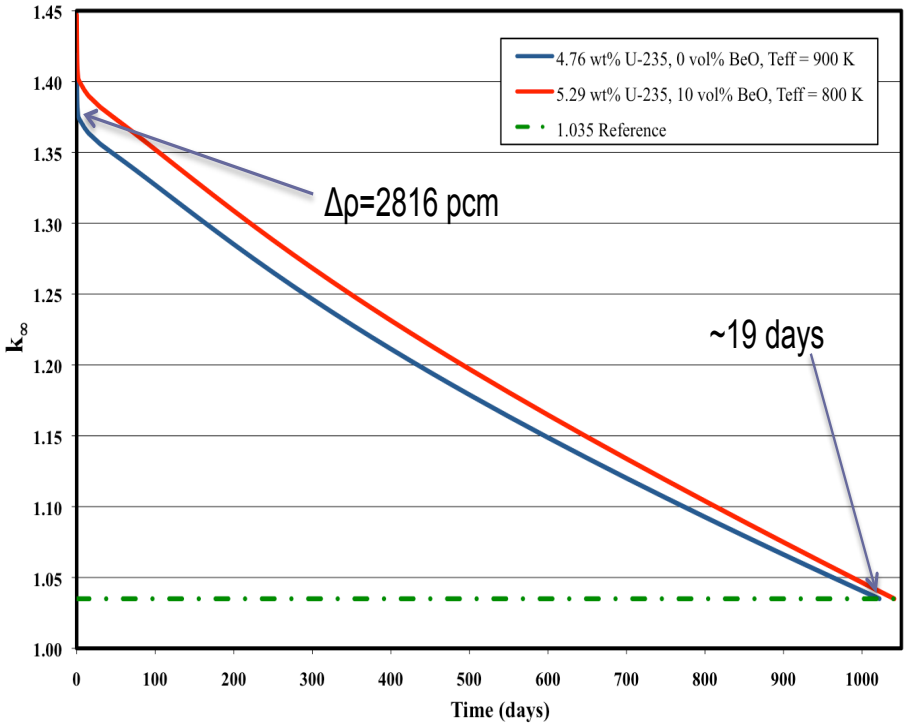


Figure 4.11 Reactivity vs time plot for U-235 mass equivalent cases with a 37.5 GWd/tHM average burnup.

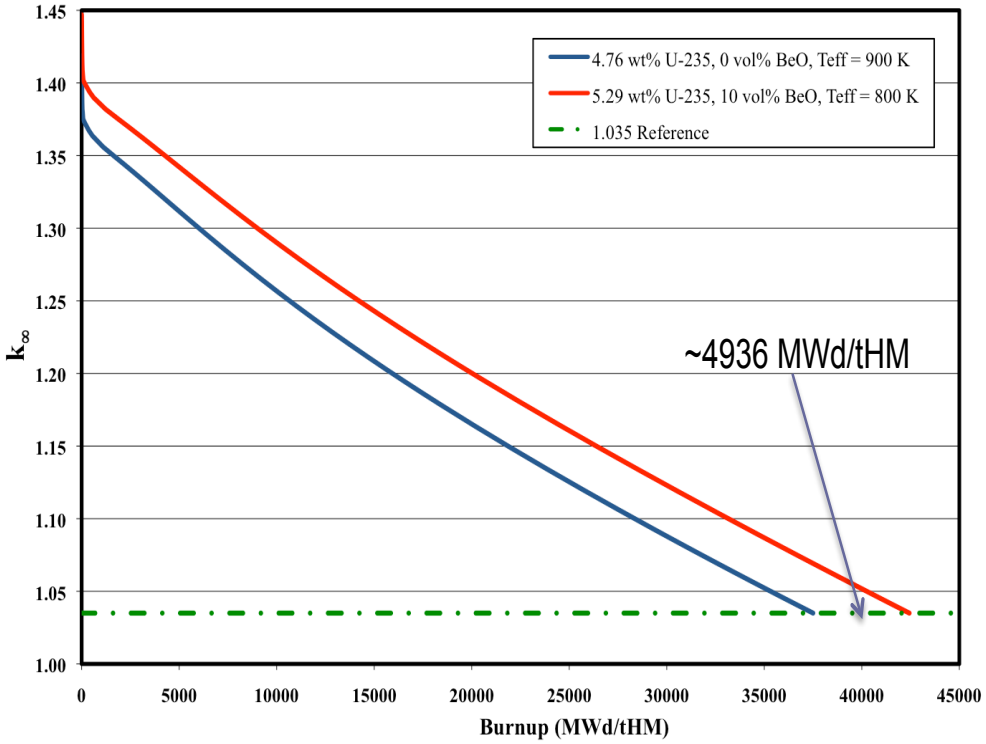


Figure 4.12 Reactivity vs burnup plot for U-235 mass equivalent cases with a 37.5 GWd/tHM average burnup.

When comparing the UO_2 case to the UO_2-10 vol% BeO , as shown in Figures 4.11 and 4.12, there is an increase in BOC reactivity of 2816 pcm because of the introduction of BeO into the fuel. With this increase in reactivity there is also an increase in the time the reactor can operate before EOC of approximately 19 days. There is also an increase in the equivalent average burnup of the fuel of 4936 MWd/tHM.

The percentage increases to the initial reactivity, operational time, and burnup are very similar to the percentage increases found in the 3-batch case: 1.98% increase in BOC reactivity, 1.8% increase in operational time, and13.2% increase in equivalent core average burnup. The discharge burnup of a fuel assembly from this 4-batch strategy is 60 GWd/tHM for the typical UO_2 fuel, and 68 GWd/tHM for the 10 vol% BeO additive fuel. This shows that although the use of BeO as a fuel additive in nuclear reactors has a dependence on the specific batch strategy used, it is a weak dependence.

The main reason that 10 vol% BeO is used in these case studies is because it gives a good illustration of the effects that BeO can have when introduced as a fuel additive as well as the fact that past research into UO_2-BeO fuels has focused on 10 vol% BeO and therefore more data as to the thermal-physical properties of UO_2-BeO are available. Because of the greater amount of data on the thermal properties of UO_2-BeO fuels at 10 vol% BeO, greater accuracy in the effect of BeO on the temperature in the fuel pellet was possible. Although, because 10 vol% BeO was chosen, the enrichment necessary to carry out the four-batch cycle analysis was 5.29 wt%, which is above current NRC regulations on the maximum enrichment of uranium in commercial reactors. Even so, the subtle decrease in both the extra time in the core before EOL and the extra percentage burnup, illustrate the burnup cycle dependent of the BeO effect very well.

Table 4.5 summarizes the results of the two batch strategies.

Table 4.5 The amount of excess time and the amount of extra burnup granted by using the BeO fuel additive.

	Average Burnup (GWd/tHM)	BOC $\Delta\rho$ (pcm)	Excess Time (days)	Extra Burnup (MWd/tHM)
3-Batch Cycle	30	2927	19.75	4138.68
4-Batch Cycle	37.5	2816	18.88	4936.62

5 SUMMARY AND RECOMMENDATIONS

Beryllium oxide has an extensive history of being used in the nuclear industry including being used in the Experimental Beryllium Oxide Reactor and the Daneils Reactor, as well as being used in many reactors as a neutron reflector. Many avenues have also been explored with using BeO as a fuel additive in UO_2 in order to increase its thermal conductivity and thereby improve its performance during irradiation by reducing high-temperature performance-limiting phenomena such as void swelling, pellet cracking, and fission gas release, as well as decreasing the stored heat in the fuel.

Two-dimensional, infinite lattice neutronic simulations using the DRAGON code have been performed for a typical PWR using variations of this UO_2-BeO fuel and typical UO_2 fuel in order to assess the effect that BeO and its corresponding fuel temperature reduction has on the reactivity over the lifetime of the reactor. It has been shown that with an equal amount of uranium-235, the temperature effect of the BeO increases the BOC reactivity of the reactor by approximately 2800-2900 pcm that results in approximately 20 more days of reactor operation before reaching EOC. Along with this, the UO_2-BeO fuel experiences about a 13% higher burnup then the reference UO_2 fuel. With the increasingly higher burnups demanded of modern day nuclear fuels, this BeO additive definitely improves the lifetime of the fuel.

Also, material fabrication work has been started, including the gathering of all the necessary equipment to start lab-scale fabrication of UO_2-BeO fuel pellets using the methodology outlined in Figure J.1 in Appendix J. The pellet pressing system has been finalized and a ceramic processing vessel has been designed and built for use with the custom open-air Deltech furnace situated in the Fuel Cycle and Materials Lab at Texas A&M University. The processing vessel also is equipped with a LVDT to measure the change in axial height of the fuel pellet while at temperature, allowing the density to be extrapolated from this change in axial height. This will allow further characterization of the fuel sintering process.

5.1 Conclusion

Beryllium oxide as an additive to nuclear fuel has many beneficial properties including chemical stability with uranium dioxide up to 2150^{o}C, good neutron moderation, potential neutron production with its (n,2n) reaction, and most importantly it high thermal conductivity. Its use as a high thermal conductivity fuel additive has been proposed investigated in the past, but this continuous BeO lattice has been shown to have much greater improvements to the thermal conductivity.

Using BeO to enhance the thermal conductivity of nuclear fuel comes with the draw back of displacing fissile and fertile fuel with a non-fissionable material. This means that the specific power of the remaining fuel must be increased to compensate for the lost fuel. Also, the enrichment of the UO_2-BeO fuel is has to be increased to compensate for the lost uranium-235 that the BeO is displacing. It should be noted that if the thermal conductivity enhancement of the BeO additive could be realized at lower volume percentages of BeO, then the uranium displacement would be less and the overall effect would be improved.

These disadvantages are balanced by the potential advantages that increasing the thermal conductivity of the fuel for a LWR brings: decreased fission gas release, reduced void swelling and reduced pellet cracking. Reducing the effect of these phenomena would have a huge impact on the overall physical state of the fuel after long-term irradiation in a reactor.

5.2 Future Work

The material fabrication work needs to be continued and expanded into a full parametric study and characterization of the property changes in the fuel pellets with respect to changes in the methodology of the process. More specifically, the sintering process of these fuel pellets needs to be characterized carefully, with the density change during the sintering being an important study to look into.

Next, the continuous BeO lattice microstructure at different volume percents of BeO needs to be characterized. Specifically, the BeO volume percent limit for the continuous microstructure needs to be identified (i.e. the volume percent of BeO necessary to form the continuous lattice), if one exists.

The next step for fabrication and fuel testing is irradiation testing. Medium and long-term irradiation tests for the fuel need to be run in order to see the effect this continuous lattice of BeO has on the restructuring of the fuel and the release of fission gas, as well as void swelling and pellet cracking. These phenomena need to be characterized for various amounts of burnup and various amounts of BeO in order to assess the long-term benefit of this continuous lattice.

The neutronic simulations also need to continue in order to gain a more complete understanding of the effects of this continuous BeO lattice in reactors. These simulations should eventually be extended to full 3D modeling of the cores with fuel cycles for both PWRs and BWRs. Also, accurate data for current generation BWRs and PWRs needs to be found in order to provide the simulations with the highest fidelity possible.

REFERENCES

1. McDeavitt S.M. “High Conductivity in Oxide Nuclear Fuel: Enhancement by Beryllium Oxide Addition.” Concept paper: IBC Advanced Alloys, LLC, Purdue University, and Texas A&M University (2010).
2. Bagger C., Mogensen M., Walker C.T. “Temperature measurements in high burnup UO_2 nuclear fuel: Implication for thermal conductivity, grain growth and gas release.” Journal of Nuclear Materials 211 (1994): 11-29.
3. Levin E.M., Robbins C.R., McMurdie H.F., Phase Diagrams for Ceramists-II.
4. Todreas N.E. and Kazimi M.S. “Nuclear Systems I Thermal Hydraulic Fundamentals.” New York: Taylor and Francis Group, LLC (1990).
5. Olander D.R. Fundamental Aspects of Nuclear Reactor Fuel Elements. Berkeley California: Technical Information Center, Office of Public Affairs, Energy Research and Development Administration (1976).
6. Lambert D.B. and Strain R. “Oxide Fuels.” Argonne National Laboratory, Argonne, IL.
7. Manly W.D. Part II. Utilization of beryllia in nuclear reactors 3-68. “Utilization of BeO in Reactors.” Journal of Nuclear Materials 14 (1964): 3-18.
8. Latta R.N. “Finite Element Analysis and Measurement of Enhanced Thermal Conductivity Oxide Nuclear Fuels” Purdue University (2006).
9. Sarma K.H., Fourcade J., Lee S-G., Solomon A.A. “New Processing Methods to Produce Silicon Carbide and Beryllium Oxide Inert Matrix and Enhanced Thermal Conductivity Oxide Fuels.” Journal of Nuclear Materials 352 (2006): 324-333.
10. Solomon A.A., Revankar S., McCoy K. “Enhanced Thermal Conductivity Oxide Fuels, Final Report.” Project No. 02-180 (2002).
11. Goodjohn A.J. and Stewart H.B. “The Use of BeO in Advanced Reactor Concepts.” Journal of Nuclear Materials 14 (1964): 19-28.

12. Mills R.G., Barner J.O., Johnson D.E., and Simnad M.T. “Irradiation Effects on Dispersion Type BeO-UO_2 Fuels for EBOR.” Journal of Nuclear Materials 14 (1964): 482-486.
13. Hanna G.L., Hilditch R.J., and Hickman B.S. “The Effects of Fission Product Damage In BeO Dispersion Fuels.” Journal of Nuclear Materials 14 (1964):473-481.
14. Johnson D.E. and Mills R.G. “Irradiation Behavior of BeO-UO_2 Fuel as a Function of Fuel-Particle Size.” San Diego, California: General Atomic Division, General Dynamics Corporation (1963).
15. Kurina I.S., Popov V.V., and Rumyantsev V.N. “Investigation of the Properties of Modified Uranium Dioxide.” Atomic Energy 101.5 (2006).
16. Ishimoto S., Hirai M., Ito K., and Korei Y. “Thermal Conductivity of UO_2-BeO Pellet.” Journal of Nuclear Science and Technology 33.2 (1996):134-140.
17. Wang J. “Developing a High Thermal Conductivity Nuclear Fuel with Silicon Carbide Additives.” University of Florida (2008).
18. Verrall R.A., Vlajic M.D., and Krstic V.D. “Silicon carbide as an inert-matrix for a thermal reactor fuel.” Journal of Nuclear Materials 274 (1999): 54-60.
19. McCoy K. and Mays C. “Enhanced thermal conductivity oxide nuclear fuels by co-sintering with BeO: II. Fuel performance and neutronics.” Journal of Nuclear Materials 375 (2008): 157-167.
20. Li D.S., Garmestani H., and Schwartz J. “Modeling thermal conductivity in UO_2 with BeO additions as a function of microstructure.” Journal of Nuclear Materials 392 (2009): 22-27.
21. Marleau G., Hébert A., and Roy R. “A User Guide for DRAGON Version4.” Institut de génie nucléaire, Departement de génie mécanique, École Polytechnique de Montréal (2009) Technical Report IGE-294.
22. Driscoll M.J., Downar T.J., and Pilat E.E., The Linear Reactivity Model for Nuclear Fuel Management. Illinois: American Nuclear Society, 1990. Print.

23. Fink J.K. Review. “Thermophysical properties of uranium dioxide.” Journal of Nuclear Materials 279 (2000): 1-18.
24. Solomon A.A. et al. “Appendix B FEM Thermal Models of UO2/BeO Nuclear Fuels and Temperature Difference Profiles.” (2002).
25. Housiadas C. and Antonopoulos-Domis M. “The Effect of Fuel Temperature on the Esitimation of the Moderator Temperature Coefficient in PWRs.” Annals of Nuclear Energy 26 (1999):1395-1405.
26. Bailly H., Ménessier D., and Prunier C., The Nuclear Fuel of Pressurized Water Reactors and Fast Reactors: Design and Behavior, 1999. Print.
27. Cochran R.G. and Tsoulfanidis N., The Nuclear Fuel Cycle: Analysis and Management. 2nd Edition. 1999. Print.
28. Kuczynski G.C. “Physics and Chemistry of Sintering.” Advances in Colloid and Interface Science, Elsevier Publishing Company, Amsterdam – Printed in The Netherlands.

APPENDIX A

The automation script: creates a series of DRAGON decks based on given information.


```
#!/usr/bin/perl
use Math::Trig;
use File::Copy;             # a small Perl package for copy files

#use warnings

################################################
# Modify Existing "Template" Dragon Deck and Run
################################################

# Data Organization
my $eps = 1e-10;                        # Convergence Tolerance for Newton's Method

my $N_a = 0.60221415;                   # Avogadro's Number

my %fuel = (
    "specify"           => 1,           # wt % or at %
    "u235contentw"      => 1,           # wt %
    "u235contenta"      => 1,           # at %
    "beoContent"        => 1,           # volume %
    "TD"                => 1,           # %
    "temp"              => 1,           # Kelvin
    "centerlineTemp"    => 1,           # Celcius
    "surfaceTemp"       => 1,           # Celcius
    "radius"            => 1,           # cm
    "qTriplePrime"      => 1,           # W/cm^3
    "beoFactor"         => 1,           # dimensionless factor
    "radialPeak"        => ( pi / 2 ),  # dimensionless factor
    "axialPeak"         => 2.32,        # dimensionless factor
    "k"                 => 1,           # dimensionless
);
my %clad = (
    "innerRadius"       => 1,           # cm
    "outerRadius"       => 1,           # cm
    "dilution"          => 1,           # dimensionless factor
    "temp"              => 1,           # Kelvin
    "innerTemp"         => 1,           # Celcius
    "outerTemp"         => 1,           # Celcius
    "tConductivity"     => 13.0,        # W/m^2-C
    "convectiveCoef"    => 3.6e4,       # W/m^2-C
    "gapCoef"           => 10**4,       # W/m^2-C
);
my %assembly = (
    "numAssem"          => 193,         # dimensionless
    "numPins"           => 264,         # dimensionless
    "coreHeight"        => 425,         # cm
    "R1"                => 1,           # cm
    "R2"                => 1,           # cm
    "R3"                => 1,           # cm
);

my %u = (
    "density"           => 19.1,        # g/cc
    "atomicWeight235"   => 235.0439231, # g/mol
    "atomicWeight238"   => 238.0507826, # g/mol
    "molarMass"         => 1,           # g/mol
);
```

```
my %be = (
    "density"           => 1.85,        # g/cc
    "atomicWeight"      => 9.012182,    # g/mol
);
my %h = (
    "density"           => 0.08988,     # g/L
    "atomicWeight"      => 1.007825,    # g/mol
);
my %o = (
    "density"           => 1.429,       # g/L
    "atomicWeight"      => 15.9949146,  # g/mol
);
my %b = (
    "density"           => 2.34,        # g/cc
    "atomicWeight10"    => 10.012937,   # g/mol
    "atomicWeight11"    => 11.0093055,  # g/mol
    "b10contenta"       => 19.9,        # at %
    "b10contentw"       => 1,           # wt %
    "molarMass"         => 1,           # g/mol
    "atomDensity"       => 1,           # at/cm-b
);
my %zr = (
    "density"           => 6.52,        # g/L
    "atomicWeight"      => 90.905645,   # g/mol
);
my %uo2 = (
    "tdensity"          => 10.97,       # g/cc
    "density"           => 1,           # g/cc
    "molarMass"         => 1,           # g/mol
    "atomDensity"       => 1,           # at/cm-b
);
my %beo = (
    "density"           => 3.02,        # g/cc
    "molarMass"         => 25.0116,     # g/mol
    "atomDensity"       => 1,           # at/cm-b
);
my %h2o = (
    "temp"              => 577,         # Kelvin
    "pressure"          => 155,         # bars
    "density"           => 1,           # g/cc
    "molarMass"         => 1,           # g/mol
    "atomDensity"       => 1,           # at/cm-b
);
my %uo2_beo = (
    "tdensity"          => 1,           # g/cc
    "density"           => 1,           # g/cc
    "molarMass"         => 1,           # g/mol
);
my %h2o_b = (
    "ppmBoron"          => 1,           # ppm
    "percentBoron"      => 1,           # %
    "molarMass"         => 1,           # g/mol
    "density"           => 1,           # g/cc
);

my %atomDensity = (
    "u235"              => 1,           # at/cm-b
```

```
    "u238"              => 1,           # at/cm-b
    "be9"               => 1,           # at/cm-b
    "o16_fuel"          => 1,           # at/cm-b
    "zr91"              => 1,           # at/cm-b
    "h1"                => 1,           # at/cm-b
    "o16_water"         => 1,           # at/cm-b
    "b10"               => 1,           # at/cm-b
    "b11"               => 1,           # at/cm-b
);

my %fuelVolume = (
    "core"              => 1,           # cm^3
    "assembly"          => 1,           # cm^3
    "pin"               => 1,           # cm^3
);
my %hmMass = (
    "core"              => 1,           # kg
    "assembly"          => 1,           # kg
    "pin"               => 1,           # kg
);
my %power = (
    "specify"           => 1,           # total, density or linear
    "total"             => 1,           # MW(th)
    "density"           => 1,           # kW/kg
    "linear"            => 1,           # kW/m
);
my %burnup = (
    "specify"           => 1,           # total or time
    "total"             => 1,           # GWd/tHM
    "time"              => 1,           # days
    "step"              => 1,           # days
    "numSteps"          => 1,           # days
    "postDay"           => 1,           # days
);


# Template Management
    my $file_in= $ARGV[0];
    my $file_out_start = $ARGV[1];
    print "input file is $file_in\n";


# Altered Values per Run
$fuel{"specify"} = "wt %";
$fuel{"u235contentw"} = 4.0;
$fuel{"beoContent"} = 0;
$fuel{"TD"} = 95;
$fuel{"radius"} = 0.41;
$fuel{"beoFactor"} = 1.0;
$clad{"innerRadius"} = 0.418;
$clad{"outerRadius"} = 0.475;
$assembly{"numAssem"} = 193;
$assembly{"numPins"} = 264;
$assembly{"coreHeight"} = 366;
$h2o{"temp"} = 575;
$h2o{"pressure"} = 155;
$h2o{"density"} = 0.72234636;
```

```
$h2o_b{"ppmBoron"} = 0;
$power{"specify"} = "linear";
$power{$power{"specify"}} = 17.8;
$burnup{"specify"} = "time";
$burnup{$burnup{"specify"}} = 1950;
$burnup{"postDay"} = 1000;
$burnup{"numSteps"} = 50;

my @peakFactors_changes = (     # axialPeak, radialPeak
    [1.0,1.0],
#   [1.0,(pi/2)],
#   [2.32,1.0],
#   [2.32,(pi/2)],
);
my @u235_changes = (            # enrichment wt %
#   3.8603858025,
#   4.7596896427,
#   4.700,
#   4.705,
#   4.710,
#   4.715,
#   4.720,
#   4.725,
#   4.730,
#   4.735,
#   4.740,
#   4.745,
#   4.750,
#   4.755,
#   4.7596896427,       # best guess run#13
#   4.760,
#   4.765,
#   4.770,
#   4.775,
#   4.780,
#   4.785,
#   4.790,
#   4.795,
#   4.800,
    0.0,
    5.0,
    10.0,
);
my @beo_changes = (             # format: beo%
    0.0,
#   1.0,
#   2.0,
#   3.0,
#   4.0,
#   5.0,
#   6.0,
#   7.0,
#   8.0,
#   9.0,
    10.0,
#   11.0,
#   12.0,
```

```
#   13.0,
#   14.0,
#   15.0,
#   16.0,
#   17.0,
#   18.0,
#   19.0,
#   20.0,
);


# loop on multiple changes
for my $h (0..$#peakFactors_changes) {
for my $i (0..$#beo_changes) {
    for my $j (0..$#u235_changes) {
#for my $i (0..0) {
#   for my $j (0..0) {

        # Loop Quantities
            $fuel{"beoContent"} = $beo_changes[$i];
#           my $old_u235content = 3.8603858025;
#           my $old_u235content = 4.7596896427;
            my $old_u235content = $u235_changes[$j];
            my $old_beocontent = 0;
            my $temp_constant = ((1-($old_beocontent/100))/(1-($fuel{"beoContent"}/100))) * $old
            $fuel{"u235contentw"} = $temp_constant * (1+((2*$o{"atomicWeight"})/$u{"atomicWeight
#           $fuel{"u235contentw"} = $u235_changes[$j];
#           $fuel{"axialPeak"} = $peakFactors_changes[$h][0];
#           $fuel{"radialPeak"} = $peakFactors_changes[$h][1];

        # Update Variable Quantities
            # Atomic or Weight % for Uranium
                if ($fuel{"specify"} eq "wt %") {
                    $fuel{"u235contenta"} = 100 / ( 1 + ( ($u{"atomicWeight235"}/$u{"atomicWeigh
                } else {    # at %
                    $fuel{"u235contentw"} = ( ($u{"atomicWeight235"}/$u{"atomicWeight238"}) * $f
                }
            # Dilution Factor
                $V_gap = pi * ( $clad{"innerRadius"}**2 - $fuel{"radius"}**2 );
                $V_clad = pi * ( $clad{"outerRadius"}**2 - $clad{"innerRadius"}**2 );
                $clad{"dilution"} = $V_clad / ( $V_clad + $V_gap );
            # Finite Differences Radii
                @areaRatios = (0.5,0.3,0.15,0.05);
                $pelletArea = ( pi * $fuel{"radius"}**2 );
                $assembly{"R1"} = sqrt( $areaRatios[0] * $pelletArea / pi );
                $assembly{"R2"} = sqrt( ( $areaRatios[1] * $pelletArea / pi ) + $assembly{"R1"}*
                $assembly{"R3"} = sqrt( ( $areaRatios[2] * $pelletArea / pi ) + $assembly{"R2"}*
            # Uranium Molar Mass
                $u{"molarMass"} = 100 / ( ( $fuel{"u235contentw"} / $u{"atomicWeight235"} ) + (
            # Boron Weight % B10, Molar Mass, and Atom Density
                $b{"b10contentw"} = ( ($b{"atomicWeight10"}/$b{"atomicWeight11"}) * $b{"b10conte
                $b{"molarMass"} = 100 / ( ( $b{"b10contentw"} / $b{"atomicWeight10"} ) + ( ( 100
                $b{"atomDensity"} = ( $b{"density"} * $N_a / $b{"molarMass"} );
            # Uranium Dioxide Density, Molar Mass, and Atom Density
                $uo2{"density"} = $uo2{"tdensity"} * $fuel{"TD"} / 100;
                $uo2{"molarMass"} = ( 2 * $o{"atomicWeight"} ) + $u{"molarMass"};
                $uo2{"atomDensity"} = ( $uo2{"density"} * $N_a / $uo2{"molarMass"} );
```

```
            # Beryllium Oxide Atom Density
                $beo{"atomDensity"} = ( $beo{"density"} * $N_a / $beo{"molarMass"} );
            # Water Density
#               $h2o{"density"} = 0.718092971;      # @ T = 577 K and P = 155 bars
            # Water Molar Mass and Atom Density
                $h2o{"molarMass"} = ( 2 * $h{"atomicWeight"} ) + $o{"atomicWeight"};
                $h2o{"atomDensity"} = ( $h2o{"density"} * $N_a / $h2o{"molarMass"} );
            # UO2 + BeO Theoretical Density, Density, and Molar Mass
                $uo2_beo{"tdensity"} = ( $uo2{"tdensity"} * (100-$fuel{"beoContent"})/100 ) + (
                $uo2_beo{"density"} = $uo2_beo{"tdensity"} * $fuel{"TD"} / 100;
                $uo2_beo{"molarMass"} = ( $beo{"molarMass"} * $fuel{"beoContent"}/100 ) + ( $uo2
            # H2O + Boron Percent Boron, Molar Mass, and Density
                $h2o_b{"percentBoron"} = $h2o_b{"ppmBoron"}/1000000;
                $h2o_b{"molarMass"} = ( $b{"molarMass"} * $h2o_b{"percentBoron"}/100 ) + ( $h2o{
                $h2o_b{"density"} = ( $b{"density"} * $h2o_b{"percentBoron"}/100 ) + ( $h2o{"den
            # Final Atom Densities (for input into Dragon)
                $atomDensity{"h1"} = (1-($h2o_b{"percentBoron"}/100)) * $h2o{"atomDensity"} * 2;
                $atomDensity{"o16_water"} = (1-($h2o_b{"percentBoron"}/100)) * $h2o{"atomDensity
                $atomDensity{"b10"} = ( ($b{"b10contentw"}/100) / ( $b{"atomicWeight10"} / $b{"m
                $atomDensity{"b11"} = ( (1-($b{"b10contentw"}/100)) / ( $b{"atomicWeight11"} / $
                $atomDensity{"zr91"} = $clad{"dilution"} * ( $zr{"density"} * $N_a / $zr{"atomic
                $atomDensity{"u235"} = ( ($fuel{"u235contentw"}/100) / ( $u{"atomicWeight235"} /
                $atomDensity{"u238"} = ( (1-($fuel{"u235contentw"}/100)) / ( $u{"atomicWeight238
                $atomDensity{"be9"} = ($fuel{"beoContent"}/100) * $beo{"atomDensity"};
                $atomDensity{"o16_fuel"} = ( (1-($fuel{"beoContent"}/100)) * $uo2{"atomDensity"}
            # Volumes
                $fuelVolume{"pin"} = pi * $assembly{"coreHeight"} * ( $fuel{"radius"}**2 );
                $fuelVolume{"assembly"} = $fuelVolume{"pin"} * $assembly{"numPins"};
                $fuelVolume{"core"} = $fuelVolume{"assembly"} * $assembly{"numAssem"};
            # Heavy Metal Masses
                $hmMass{"pin"} = $fuelVolume{"pin"} * $uo2{"density"} * ($u{"molarMass"}/$uo2{"m
                $hmMass{"assembly"} = $fuelVolume{"assembly"} * $uo2{"density"} * ($u{"molarMass
                $hmMass{"core"} = $fuelVolume{"core"} * $uo2{"density"} * ($u{"molarMass"}/$uo2{
            # Total Power, Power Density, and/or Linear Heat Rate
                if ( $power{"specify"} eq "total" ) {
                    $power{"density"} = $power{"total"} * 1000 / $hmMass{"core"};
                    $power{"linear"} = $power{"total"} * 10**5 / ( $assembly{"numAssem"} * $asse
                } else {
                    if ( $power{"specify"} eq "density" ) {
                        $power{"total"} = $power{"density"} * $hmMass{"core"} / 1000;
                        $power{"linear"} = $power{"density"} * $hmMass{"pin"} * 100 / $assembly{
                    } else {    # linear
                        $power{"density"} = $power{"linear"} * $assembly{"coreHeight"} / (100*$h
                        $power{"total"} = $power{"linear"} * $assembly{"numAssem"} * $assembly{"
                    }
                }
            # Total Burnup or Burnup Time and Burnup Step
                if ( $burnup{"specify"} eq "total" ) {
                    $burnup{"time"} = 1000 * $burnup{"total"} / $power{"density"};
                } else {    # time
                    $burnup{"total"} = $burnup{"time"} * $power{"density"} / 1000;
                }
                $burnup{"step"} = ($burnup{"time"} - $burnup{"postDay"})/$burnup{"numSteps"};
            # Temperature Calculations
                $fuel{"qTriplePrime"} = $power{"total"}*10**6 / ( $fuelVolume{"core"} / 10**6 );
#               print ("\n\nqTriplePrime = ".$fuel{"qTriplePrime"}." W/m^3\n\n");
                $clad{"outerTemp"} = ($h2o{"temp"} - 273) + ( $fuel{"qTriplePrime"} * ($fuel{"ra
```

```
                $clad{"innerTemp"} = $clad{"outerTemp"} + ( log($clad{"outerRadius"}/$clad{"inne
                $fuel{"surfaceTemp"} = $clad{"innerTemp"} + ( $fuel{"qTriplePrime"} * ($fuel{"ra
#               print ("Tmod = ".($h2o{"temp"} - 273)." Tclad = ".$clad{"outerTemp"}." Tgap = ".
#               for my $fuel_ring (0..10) {
#                   my $ring_radius = ($fuel_ring/10) * $fuel{"radius"}/100;
#                   my $newtons_variable = $h2o{"temp"} - 273;
#                   print "Newton's Variable = ".$newtons_variable."\n";
#                   for my $newton_i (1..100000) {
#                       if ( $fuel{"k"} == 1 ) {
#                           # Ragusa's Formula: (1)
#                               $evaluated_function = $fuel{"beoFactor"}*(1.05*($newtons_variabl
#                               $evaluated_derivative = $fuel{"beoFactor"}*(1.05 + 2150.0/($newt
#                       } elsif ( $fuel{"k"} == 2 ) {
#                           # Westinghouse PWR UO2 Formula: (2)
#                               $evaluated_derivative = $fuel{"beoFactor"}*100*(( 1 / ( 11.8 + (
#                               $evaluated_function = $fuel{"beoFactor"}*100*( (((8.775e-13)/4)
#                       } elsif ( $fuel{"k"} == 3 ) {
#                           # Fink Fit (3)
#                               $a = 257.99;
#                               $b = -0.6275;
#                               $evaluated_function = $fuel{"beoFactor"} * ($a/($b+1))*((($newto
#                               $evaluated_derivative = $fuel{"beoFactor"} * $a * (($newtons_var
#                       } elsif ( $fuel{"k"} == 4 ) {
#                           # GG BeO Fit (4)
#                               $a = 7117.0;
#                               $b = -1.0368;
#                               $evaluated_function = ($a/($b+1))*((($newtons_variable+273)**(1+
#                               $evaluated_derivative = $a * (($newtons_variable+273)**(1+$b));
#                       } elsif ( $fuel{"k"} == 5 ) {
#                           # SB BeO Fit (5)
#                               $a = 326.03;
#                               $b = -0.6222;
#                               $evaluated_function = ($a/($b+1))*((($newtons_variable+273)**(1+
#                               $evaluated_derivative = $a * (($newtons_variable+273)**(1+$b));
#                       } else {
#                           print ("\n\nError\n\n");
#                       }
#                       $newtons_variable = $newtons_variable - ($evaluated_function/$evaluated_
#                       print "$newtons_variable\n";
#                       if (abs($evaluated_function/$evaluated_derivative) < $eps) {
#                           if ($fuel_ring == 10) {
#                               print ("radius = " . ($ring_radius*100) . "0cm, Temp = " . $newt
#                           } else {
#                               print ("radius = " . ($ring_radius*100) . "cm, Temp = " . $newto
#                           }
#                       $fuelTemp[$fuel_ring] = $newtons_variable;
#                           last;
#                       }
#                   }
#               }
#               print ("Tsurface = ".$fuel{"surfaceTemp"}." Tsurface(newtons) = ".$fuelTemp[10].
#               $fuel{"centerlineTemp"} = $fuelTemp[0];
                $clad{"temp"} = ( ( $clad{"outerTemp"} + $clad{"innerTemp"} ) / 2 ) + 273;
#               $fuel{"temp"} = ( (5.0/9.0) * $fuel{"centerlineTemp"} ) + ( (4.0/9.0) * $fuel{"s
#               print ("T_effective = ".$fuel{"temp"}." K\n\n");
            # Fuel Average Temperature Distribution
                $fuel{"temp"} = 900 - ( 10 * $fuel{"beoContent"} );
```

```
#               $fuel{"temp"} = 800;
            # Replacement Hash
                my %replacement = (
                    "AAC01"     => "* Uranium Dioxide - Beryllium Oxide Fuel Criticality Simulat
                    "AAC02"     => "* UOX " . $fuel{"u235contentw"} . "wt% U235 + " . $fuel{"bec
                    "AAC03"     => "* Theoretical Density = " . $fuel{"TD"} . "%",
                    "AAC04"     => "* # of Fuel Assemblies = " . $assembly{"numAssem"},
                    "AAC05"     => "* # of Fuel Pins per Fuel Assembly = " . $assembly{"numPins"
                    "AAC06"     => "* Core Height = " . $assembly{"coreHeight"} . " cm",
                    "AAC07"     => "* Fuel Pellet Radius = " . $fuel{"radius"} . " cm",
                    "AAC08"     => "* Cladding Inner Radius = " . $clad{"innerRadius"} . " cm",
                    "AAC09"     => "* Cladding Outer Radius = " . $clad{"outerRadius"} . " cm",
                    "AAC10"     => "* Boron Content in Water = " . $h2o_b{"ppmBoron"} . " ppm",
                    "AAC11"     => "* Water Pressure = " . $h2o{"pressure"} . " bars",
                    "AAC12"     => "* Fuel Temperature = " . $fuel{"temp"} . " K",
                    "AAC13"     => "* Cladding Temperature = " . $clad{"temp"} . " K",
                    "AAC14"     => "* Water Temperature = " . $h2o{"temp"} . " K",
                    "AAC15"     => 1,
                    "AAC16"     => 1,
                    "AAD01"     => $power{"density"},
                    "AAD02"     => $h2o{"temp"},
                    "AAD03"     => $atomDensity{"h1"},
                    "AAD04"     => $atomDensity{"o16_water"},
                    "AAD05"     => $clad{"temp"},
                    "AAD06"     => $atomDensity{"zr91"},
                    "AAD07"     => $fuel{"temp"},
                    "AAD08"     => $atomDensity{"u235"},
                    "AAD09"     => $atomDensity{"u238"},
                    "AAD10"     => $atomDensity{"be9"},
                    "AAD11"     => $atomDensity{"o16_fuel"},
                    "AAE01"     => $assembly{"R1"},
                    "AAE02"     => $assembly{"R2"},
                    "AAE03"     => $assembly{"R3"},
                    "AAE04"     => $fuel{"radius"},
                    "AAE05"     => $clad{"outerRadius"},
                    "AAF01"     => $burnup{"step"},
                    "AAF02"     => $burnup{"time"},
                );
            #   Comment Lines for the Replacement Hash
                if ($replacement{"AAD01"} !~ /\d+(\.\d)/) {
                    $replacement{"AAD01"} = $replacement{"AAD01"} . ".0";
                }
                if ($replacement{"AAD02"} !~ /\d+(\.\d)/) {
                    $replacement{"AAD02"} = $replacement{"AAD02"} . ".0";
                }
                if ($replacement{"AAD05"} !~ /\d+(\.\d)/) {
                    $replacement{"AAD05"} = $replacement{"AAD05"} . ".0";
                }
                if ($replacement{"AAD07"} !~ /\d+(\.\d)/) {
                    $replacement{"AAD07"} = $replacement{"AAD07"} . ".0";
                }
                if ( $power{"specify"} eq "total" ) {
                    $replacement{"AAC15"} = "* Total Power = " . $power{"total"} . " MW(th)";
                } else {
                    if ( $power{"specify"} eq "density" ) {
                        $replacement{"AAC15"} = "* Power Density = " . $power{"density"} . " kW/
                    } else {    # linear
```

```
                        $replacement{"AAC15"} = "* Linear Heat Rate = " . $power{"linear"} . " k
                    }
                }

                if ( $burnup{"specify"} eq "total" ) {
                    $replacement{"AAC16"} = "* Total Burnup = " . $burnup{"total"} . " GWd/tHM";
                } else {    # time
                    $replacement{"AAC16"} = "* Burnup Time = " . $burnup{"time"} . " days";
                }
                if ($replacement{"AAF01"} !~ /\d+(\.\d)/) {
                    $replacement{"AAF01"} = $replacement{"AAF01"} . ".0";
                }
                if ($replacement{"AAF02"} !~ /\d+(\.\d)/) {
                    $replacement{"AAF02"} = $replacement{"AAF02"} . ".0";
                }


        # create new deck filename (not the deck, just its NAME)
            $new_deck= $file_out_start . "_";
            if ( ($h+1) < 10 ) {
                $new_deck = $new_deck . "0" . ($h+1);
            } else {
                $new_deck = $new_deck . ($h+1);
            }
            if ( ($i+1) < 10 ) {
                $new_deck = $new_deck . "-0" . ($i+1);
            } else {
                $new_deck = $new_deck . "-" . ($i+1);
            }
            if ( ($j+1) < 10 ) {
                $new_deck = $new_deck . "-0" . ($j+1);
            } else {
                $new_deck = $new_deck . "-" . ($j+1);
            }
            $new_deck= $new_deck.".x2m";
            print "new deck is: $new_deck\n";

        # Debug
#           my $temp_print = $power{"total"};
#           print "Total Power = $temp_print MW(th)";
#           print "\n\n";
            foreach $key (sort (keys(%replacement))) {
                $value = $replacement{$key};
                print "$key => $value\n";
            }
            print "\n\n";

        # go into the data directory to modify the decks
            $newdir="./data";
            chdir("$newdir");
            system("pwd");

        # copy generic deck into new one
            copy($file_in,$new_deck);

        # open generic and new deck
            open my $in,  '<', $file_in or die "Can't input $file_in $!";
```

```
            open my $out, '>', $new_deck or die "Can't input $new_deck $!";

        # change the dummy characters for the computed densities
            @lines = <$in>;
            foreach $lines (@lines) {
                foreach $key (sort (keys(%replacement))) {
                    $replace = $replacement{$key};
                    $lines =~ s/$key/$replace/g;
                }
                if ($lines =~ /^             Be9      0/) {
                    $lines = "" if ($lines !~ /^             Be9      0.0/);
                }
                print $out $lines;
            }
            close $in; close $out;

        # move up one directory to get ready to run dragon
            $newdir="./..";
            print "$newdir\n";
            chdir("$newdir");
            system("pwd");

        # run dragon
            $cmd= "bsub rdragon data/$new_deck";
            system($cmd);
    }
}
}


exit 66;
```

APPENDIX B

Back-end data sorter, pulls out the k_{inf} vs burnup data and puts it into a text file.


```
#!/usr/bin/perl

# B Bradley, Texas A&M University

#use warnings

##############
#Obtain the kinf as a function of burnup
##############

$file     = $ARGV[0];
print "The input is  : $file\n";
$file =~ s/result/kinf/;
$file_out = $file.'.txt' ;
$file =~ s/kinf/result/;

print "The output is : $file_out\n";


open ( INPUT , "<$file" );
open ( OUT   , ">$file_out" );


# array definition

$debug=0;

#read the output file portion with the 2D power
$i=0;
$bu=0;
$start=0;
$end=0;

$read_bu=0;
@bu_array = ();
$bu_array[0]= "0.000000E+00 ";
@kinf_array = ();
@time_array = ();
$time_array[0]= "0.000000E+00 ";
$title_thing = "";
$temp_thing = "";

while (<INPUT>) {
    $line = $_ ;
    chomp $line;
        # debug print
    if($debug>10){
        print "line  read"."\n".$line."\n" ;
    }

    # find title
    if($line=~ /\* UOX /) {
          print "$line\n" ;
        $line=~ s/\s{3,}[0-9]{4}/ / ;
        $title_thing = $line;
        next;
    }
```

```
    # find temperature
    if($line=~ /\* Fuel Temperature/) {
           print "$line\n" ;
        $line=~ s/\s{3,}[0-9]{4}/ / ;
        $temp_thing = $line;
        next;
    }

    # look for time steps
    if($line=~ /\ -> FINAL BURNUP AT TIME/) {
           print "$line\n" ;
        $line=~ s/\ -> FINAL BURNUP AT TIME  \=   / / ;
        $line=~ s/ DAYS/ / ;
           $line=~ s/\s+// ;
        $time_array[$dim+1] = $line;
        next;
    }

    # look for bu steps
    if($line=~ /\    FUEL BURNUP/) {
           print "$line\n" ;
        $line=~ s/\    FUEL BURNUP           \=   / / ;
        $line=~ s/ MW\*D\/TONNE/ / ;
           $line=~ s/\s+// ;
        $bu_array[$dim+1] = $line;
        next;
    }

     #build kinf array
     if($line=~ /\+\+ TRACKING CALLED\=/){
              print "$line\n" ;
           $line=~ s/ \+\+ TRACKING CALLED\=[\s\d][\s\d]+ TIMES FINAL/ / ;
        $line=~ s/KINF\=/ / ;
        $line=~ s/FINAL[\s\w\W]+/ /;
              $dim = $#kinf_array;
        $dim = $dim+1 ;
           $kinf_array[$dim] = $line;
            #  print "$dim\n";
              next;
    }
}
print OUT "$title_thing\n";
print OUT "$temp_thing\n";
print OUT "Time days         Burnup MWd/t      Kinf\n" ;
$dim_bu = $#bu_array;
for ($i = 0 ; $i <= $dim_bu ; $i++ ) {
    print OUT "$time_array[$i]."     "."$bu_array[$i]."  "."$kinf_array[$i]\n";
     #  print "$i\n";
}
print OUT "\n";


exit 0;
```

APPENDIX C

Organizes all of the k_{inf} vs burnup text files into a single file.


```
#!/usr/bin/perl

#use warnings


$file_start     = $ARGV[0];     # UO2-BeO_BOL-b
$firstNum_extent = $ARGV[3];
$secondNum_extent = $ARGV[2];
$thirdNum_extent = $ARGV[1];

@info_extent = ();
$max_infoExtent = 0;
$max_counter = 0;
@info_array = ();
@formated_array = ();

$file_out = $file_start . "_organized.txt";

# Grab the Data
for my $k (1..$thirdNum_extent) {
    for my $j (1..$secondNum_extent) {
        for my $i (1..$firstNum_extent) {

            $file_in = $file_start . "_";
            if ( $k < 10 ) {
                $file_in = $file_in . "0" . $k;
            } else {
                $file_in = $file_in . $k;
            }
            if ( $j < 10 ) {
                $file_in = $file_in . "-0" . $j;
            } else {
                $file_in = $file_in . "-" . $j;
            }
            if ( $i < 10 ) {
                $file_in = $file_in . "-0" . $i;
            } else {
                $file_in = $file_in . "-" . $i;
            }
            $file_in = $file_in . ".kinf.txt";
            print "new deck is: $file_in\n";

            open my $in,  '<', $file_in or die "Can't input $file_in $!";
            $debug = 0;
            $dim = 0;

            $info_array[$dim][$i-1][$j-1][$k-1] = "$file_in";
            while (<$in>) {
                $line = $_ ;
                chomp $line;

                # debug print
                if($debug>10){
                    print "line  read"."\n".$line."\n" ;
                }

                # Add this File to the Array
```

```
                $dim = $dim + 1;
                $info_array[$dim][$i-1][$j-1][$k-1] = "$line";
            }

            $info_extent[$i-1][$j-1][$k-1] = $dim;
            $max_infoExtent = $dim if ( $dim > $max_infoExtent );
#           print ("number of lines = $max_infoExtent\n");

            close $in;

        }
    }
}

# Format the Data
for $k (1..$thirdNum_extent) {
    for $j (1..$secondNum_extent) {
        for $i (1..$firstNum_extent) {
            for my $h (0..$max_infoExtent) {

                my $counter = $h + ( ($max_infoExtent + 1) * ($j-1) ) + ( ( ($max_infoExtent + 1
                my $white_space = "                                                                ";
                my $string_length = length($info_array[$h][$i-1][$j-1][$k-1]);

                if ( $h == 0 ) {
                    $formated_array[$counter] = $formated_array[$counter] . $info_array[$h][$i-1
                } elsif ( $h == 1 ) {
                    $formated_array[$counter] = $formated_array[$counter] . $info_array[$h][$i-1
                } elsif ( $h == 2 ) {
                    $formated_array[$counter] = $formated_array[$counter] . $info_array[$h][$i-1
                } elsif ( $h == 3 ) {
                    $formated_array[$counter] = $formated_array[$counter] . $info_array[$h][$i-1
                } else {
                    $formated_array[$counter] = $formated_array[$counter] . $info_array[$h][$i-1
                }

                $max_counter = $counter if ( $counter > $max_counter );

            }
        }
    }
}
for $k (1..$thirdNum_extent) {
    my $temp_counter = ( ( ($max_infoExtent + 1) * $secondNum_extent ) * $k ) - 1;
    $formated_array[$temp_counter] = $formated_array[$temp_counter] . "\n\n";
}

# Output the Data
open my $out, '>', $file_out or die "Can't input $file_out $!";
#print "\n\n";
#for $k (1..$thirdNum_extent) {
#   for $j (1..$secondNum_extent) {
#       for $i (1..$firstNum_extent) {
#           for $h (0..$info_extent[$i-1][$j-1][$k-1]) {
#               print "$info_array[$h][$i-1][$j-1][$k-1]\n";
#           }
#       }
```

```
#   }
#}
#print "\n\n";
for $h (0..$max_counter) {
    print $out "$formated_array[$h]\n";
}

close $out;


exit 0;
```

APPENDIX D

The shell script runs the kinf_v1.pl for all the files in the folder it resides in.

```
#!/bin/bash

for file in $(find -type f -name "*.result")
do
        perl kinf_v1.pl $file
done
```

APPENDIX E

This is the DRAGON template deck used by the automation script.


```
* ----
AAC01
* 17 x 17 Fuel Assembly
AAC02
* Dragon DLIB Library
AAC03
AAC04
AAC05
AAC06
AAC07
AAC08
AAC09
AAC10
AAC11
AAC12
AAC13
AAC14
AAC15
AAC16
* ----
* Define STRUCTURES and MODULES used
* ----
LINKED_LIST
LIBRARY LIBRARY2 ASSMB VOLMATF PIJ FLUX BURNUP COOL1 COOL2
 DATABASE ISOT PMAP ;

SEQ_ASCII
database ;

MODULE
GEO: SYBILT: USS: ASM: FLU: EVO: EDI: COMPO: DELETE: END: LIB: ;
*
* ----
* Define variables and initialize
* ----
REAL
Power Delt Timec Timei Timef Multi :=
AAD01 0.2 1.0 0.0 0.0 3.3 ;
* ----
* Depletion data from file DLIB_J2
* Microscopic cross sections from file DLIB_J2
* ----

*----
* CONCENTRATIONS ARE CALCULATED FROM "UO2-BeO Criticality Data.xls" (Created by Michael Naramore
* ----

LIBRARY := LIB: :: EDIT 1
   NMIX 6 CTRA NONE
   SUBG      (*HELIOS TYPE PROBABILITY TABLES*)
   DEPL LIB: DRAGON FIL: DLIB_J2
   MIXS LIB: DRAGON FIL: DLIB_J2
   MIX 1 AAD02
                 H1_H2O   AAD03
                 O16      AAD04
   MIX 2 AAD05
      Zr91 = Zr0  AAD06      2 IRSET 0.0 81
```

```
   MIX 3 AAD07
             U235     AAD08   1 IRSET 0.0 81
             U238     AAD09   1 IRSET 0.0 81
             Be9      AAD10
             O16      AAD11
   MIX 4 COMB 3 1.0
   MIX 5 COMB 3 1.0
   MIX 6 COMB 3 1.0
;
*----
*----
* Geometry ASSMB : a 17 X 17 normal PWR assembly
* contains C1 : cell without fuel, WH
* C2 : normal fuel cell
* C3 : peripheral cell
* C4 : corner cell
* C5 : IT cell
*----
ASSMB := GEO: :: CAR2D 9 9 EDIT 5
  X- DIAG X+ REFL Y- SYME Y+ DIAG
  CELL  C1 C2 C2 C1 C2 C2 C1 C2 C3
           C2 C2 C2 C2 C2 C2 C2 C3
              C2 C2 C2 C2 C2 C2 C3
                 C1 C2 C2 C1 C2 C3
                    C2 C2 C2 C2 C3
                       C1 C2 C2 C3
                          C2 C2 C3
                             C2 C3
                                C4
 ::: C1 := GEO: CARCEL 2
                MESHX 0.0 1.26 MESHY 0.0 1.26
                RADIUS 0.0 0.5715 0.6121 MIX 1 2 1 ;
 ::: C2 := GEO: CARCEL 5
                MESHX 0.0 1.26 MESHY 0.0 1.26
                   RADIUS 0.0
                          AAE01
                          AAE02
                          AAE03
                          AAE04
                          AAE05
                          MIX 6 5 4 3 2 1 ;
*                          MIX 3 3 3 3 2 1 ;
 ::: C3 := GEO: CARCEL 5
                MESHX 0.0 1.30    MESHY 0.0 1.26
                   RADIUS 0.0
                          AAE01
                          AAE02
                          AAE03
                          AAE04
                          AAE05
*                         OFFCENTER -0.02 0 0
                          MIX 6 5 4 3 2 1 ;
*                          MIX 3 3 3 3 2 1 ;
 ::: C4 := GEO: CARCEL 5
                MESHX 0.0 1.30    MESHY 0.0 1.30
                   RADIUS 0.0
                          AAE01
```

```
                          AAE02
                          AAE03
                          AAE04
                          AAE05
*                         OFFCENTER -0.02 -0.02 0
                          MIX 6 5 4 3 2 1 ;
*                          MIX 3 3 3 3 2 1 ;
;


*----
* Self-Shielding calculation SHI
* Transport calculation SYBILT
* Flux calculation for keff
*----
VOLMATF := SYBILT: ASSMB ::
 EDIT 3
 MAXR 5000 MAXZ 150000 QUA2 20 8 ;


LIBRARY2 := USS: LIBRARY VOLMATF :: EDIT 0 PASS 3
  CALC
         REGI W1 U235  ALL
         REGI W1 U238  ALL
         REGI W1 ZrO   ALL
    ENDC ;

PIJ := ASM: LIBRARY2 VOLMATF ::
  EDIT 0 ARM ;
FLUX := FLU: PIJ LIBRARY2 VOLMATF ::
  TYPE B ;

PMAP := EDI: FLUX LIBRARY2 VOLMATF ::
  EDIT 3
  MERG
  REGI 1 1 1
       2 2 2 2 2 2
       3 3 3 3 3 3
       4 4 4
       5 5 5 5 5 5
       6 6 6 6 6 6
       7 7 7
       8 8 8 8 8 8
       9 9 9 9 9 8
       10 10 10 10 10 10
       11 11 11 11 11 11
       12 12 12 12 12 12
       13 13 13 13 13 13
       14 14 14 14 14 14
       15 15 15 15 15 15
       16 16 16 16 16 16
       17 17 17 17 17 17
       18 18 18 18 18 18
       19 19 19 19 19 19
       20 20 20 20 20 20
       21 21 21 21 21 21
       22 22 22 22 22 22
```

```
       23 23 23 23 23 23
       24 24 24 24 24 24
       25 25 25
       26 26 26 26 26 26
       27 27 27 27 27 27
       28 28 28
       29 29 29 29 29 29
       30 30 30 30 30 30
       31 31 31 31 31 31
       32 32 32 32 32 32
       33 33 33 33 33 33
       34 34 34 34 34 34
       35 35 35 35 35 35
       36 36 36
       37 37 37 37 37 37
       38 38 38 38 38 38
       39 39 39 39 39 39
       40 40 40 40 40 40
       41 41 41 41 41 41
       42 42 42 42 42 42
       43 43 43 43 43 43
       44 44 44 44 44 44
       45 45 45 45 45 45
  COND
  SAVE ;


*----
* Burnup loop: for first step BURNUP is created
* while for other steps it is modified
*----
EVALUATE Timei := 0.0 ;
WHILE Timei Timec < DO
EVALUATE Timef := Timei Delt + ;
IF Timei 0.0 = THEN
  BURNUP LIBRARY2 := EVO: LIBRARY2 FLUX VOLMATF ::
    EDIT 3 DEPL <<Timei>> <<Timef>> DAY POWR <<Power>> ;
ELSE
  BURNUP LIBRARY2 := EVO: BURNUP LIBRARY2 FLUX VOLMATF ::
    EDIT 3 EXTR DEPL <<Timei>> <<Timef>> DAY POWR <<Power>> ;
ENDIF ;
LIBRARY2 := USS: LIBRARY LIBRARY2 VOLMATF :: EDIT 0 PASS 3
  CALC
         REGI W1 U235  ALL
         REGI W1 U238  ALL
         REGI W1 ZrO   ALL
    ENDC ;

PIJ := DELETE: PIJ ;
PIJ := ASM: LIBRARY2 VOLMATF ::
  EDIT 0 ARM ;
FLUX := FLU: FLUX PIJ LIBRARY2 VOLMATF ::
  TYPE B ;

PMAP := EDI: PMAP FLUX LIBRARY2 VOLMATF ::
  EDIT 3
  MERG
```

```
  REGI 1 1 1
       2 2 2 2 2 2
       3 3 3 3 3 3
       4 4 4
       5 5 5 5 5 5
       6 6 6 6 6 6
       7 7 7
       8 8 8 8 8 8
       9 9 9 9 9 8
       10 10 10 10 10 10
       11 11 11 11 11 11
       12 12 12 12 12 12
       13 13 13 13 13 13
       14 14 14 14 14 14
       15 15 15 15 15 15
       16 16 16 16 16 16
       17 17 17 17 17 17
       18 18 18 18 18 18
       19 19 19 19 19 19
       20 20 20 20 20 20
       21 21 21 21 21 21
       22 22 22 22 22 22
       23 23 23 23 23 23
       24 24 24 24 24 24
       25 25 25
       26 26 26 26 26 26
       27 27 27 27 27 27
       28 28 28
       29 29 29 29 29 29
       30 30 30 30 30 30
       31 31 31 31 31 31
       32 32 32 32 32 32
       33 33 33 33 33 33
       34 34 34 34 34 34
       35 35 35 35 35 35
       36 36 36
       37 37 37 37 37 37
       38 38 38 38 38 38
       39 39 39 39 39 39
       40 40 40 40 40 40
       41 41 41 41 41 41
       42 42 42 42 42 42
       43 43 43 43 43 43
       44 44 44 44 44 44
       45 45 45 45 45 45
  COND
  SAVE ;

BURNUP LIBRARY2 := EVO: BURNUP LIBRARY2 FLUX VOLMATF ::
   EDIT 3 SAVE <<Timef>> DAY POWR <<Power>> ;

*----
* change delta t for burnup and final time if required
*----
IF Timef Timec = THEN
  IF Timec 1000.0 = THEN
   EVALUATE Delt Timec := AAF01 AAF02 ;
```

```
  ENDIF ;
  IF Timec 500.0 = THEN
   EVALUATE Delt Timec := 50.0 1000.0 ;
  ENDIF ;
  IF Timec 150.0 = THEN
   EVALUATE Delt Timec := 25.0 500.0 ;
  ENDIF ;
  IF Timec 15.0 = THEN
   EVALUATE Delt Timec := 15.0 150.0 ;
  ENDIF ;
  IF Timec 5.0 = THEN
   EVALUATE Delt Timec := 5.0 15.0 ;
  ENDIF ;
  IF Timec 1.0 = THEN
   EVALUATE Delt Timec := 1.0 5.0 ;
  ENDIF ;
ENDIF ;
 EVALUATE Timei := Timef ;
ENDWHILE ;
END: ;
QUIT "LIST" .
```

APPENDIX F

DRAGON test deck for a fuel pin.


```
* ----
* TEST CASE U4
* 1 square cell
* UOX 4% U235
* Dragon DLIB Library
* ----
* BURN POWER (KW/KG) = 34.00
* NUMBER OF DAYS = 1950
*
* ----
* Define variables and initialize
* Burnup paremeters
* a) Power density = 34 kw/kg
* b) Burnup time interval Delt
* = 1 day for 0 to 1 day
* = 4 days for 1 to 5 days
* = 5 days for 5 to 10 days
* = 10 days for 10 to 50 days
* = 20 days for 50 to 150 days
* = 75 days for 150 to 1950 days
* c) Days with burnup interval changes
* = 1.0, 5.0, 10.0, 50.0, 150.0 and 1950.0 days
* d) Burnup control time variables Timei, Timef
* Timei = initial time
* Timef = final time
* ----
*
REAL
Power Delt Timec Timei Timef :=
34.00  1.   1.0   0.0   0.0 ;
* ----
*
* ----
* Define STRUCTURES and MODULES used
* ----
LINKED_LIST
LIBRARY LIBRARY2 ASSMB VOLMATF PIJ FLUX BURNUP EDITION DATABASE ISOT ;

SEQ_ASCII
database ;

MODULE
GEO: SYBILT: USS: ASM: FLU: EVO: EDI: COMPO: DELETE: END: LIB: ;

* ----
* Depletion data from file DLIB_J2
* Microscopic cross sections from file DLIB_J2
* ----

LIBRARY := LIB: :: EDIT 0
   NMIX 6 CTRA NONE
   SUBG      (*HELIOS TYPE PROBABILITY TABLES*)
   DEPL LIB: DRAGON FIL: DLIB_J2
   MIXS LIB: DRAGON FIL: DLIB_J2
   MIX 1 577.
                H1_H2O   4.748013802E-02
                O16      2.374006901E-02
```

```
                 B10      4.6795E-06
                 B11      1.8718E-05
   MIX 2 605.
      Zr91 = Zr0  3.767706E-02      2 IRSET 0.0 81
   MIX 3 811.
            O16      6.448028885E-02
            U235     7.704124924E-04   1 IRSET 0.0 81
            U238     4.361306728E-02   1 IRSET 0.0 81
            Be9      2.504499917E-02   1 IRSET 0.0 81
            Gd155    2.763526276E-03

   MIX 4 COMB 3 1.0
   MIX 5 COMB 3 1.0
   MIX 6 COMB 3 1.0
;
*----
* Geometry ASSMB : a regular PWR assembly cell
* Use the Santamarina 50% 30% 15% 5% recipe for distributed self shielding
ASSMB := GEO: :: CARCEL 5
          X- REFL X+ REFL MESHX 0.0 1.26
          Y- REFL Y+ REFL MESHY 0.0 1.26
          RADIUS 0.0 0.2917947 0.3690943 0.4022112
                 0.4095 0.475
*          MIX 3 3 3 3 2 1
          MIX 6 5 4 3 2 1
;

*----
* Self-Shielding calculation USS
* Transport calculation ASM+FLU
* Flux calculation for keff
*----
VOLMATF := SYBILT: ASSMB ::
 EDIT 0
 MAXR 5000 MAXZ 150000 QUA2 20 8 ;

* perform self-shielding calculation
* U-238 is treated differently than the rest of the self-shielded isotopes
LIBRARY2 := USS: LIBRARY VOLMATF :: EDIT 0 PASS 3
  CALC
         REGI W1 U235  ALL
         REGI W1 U238  ALL
      *  REGI W1 Be9   ALL
      *  REGI W1 Gd155 ALL

         ENDC
;
PIJ := ASM: LIBRARY2 VOLMATF ;

FLUX := FLU: PIJ LIBRARY2 VOLMATF ::
  TYPE K ;

*----
* Burnup loop: for the first step, structure BURNUP is created
* while for other steps it is modified
* this leads to two different calls to the LIB: module
*----
```

```
*----
* while Timei < Timec, do
*----
WHILE Timei Timec < DO
*----
*   final time = initial time + time increment
*----
    EVALUATE Timef := Timei Delt + ;
*----
*   separate call for the initial time step
*----
    IF Timei 0.0 = THEN
        BURNUP LIBRARY2 := EVO: LIBRARY2 FLUX VOLMATF ::
        EDIT 3 DEPL <<Timei>> <<Timef>> DAY POWR <<Power>> ;
    ELSE
        BURNUP LIBRARY2 := EVO: BURNUP LIBRARY2 FLUX VOLMATF ::
        EDIT 3 EXTR DEPL <<Timei>> <<Timef>> DAY POWR <<Power>> ;
    ENDIF ;
*----
*   perform self-shielding calculation
*----
    LIBRARY2 := USS: LIBRARY LIBRARY2 VOLMATF :: EDIT 0 PASS 3
    CALC    REGI W1 U235  ALL
            REGI W1 U238  ALL
       *    REGI W1 Be9   ALL
       *    REGI W1 Gd155 ALL

        ENDC
    ;
*----
*   compute new flux
*----
    PIJ := DELETE: PIJ ;
    PIJ := ASM: LIBRARY2 VOLMATF :: EDIT 0 ARM ;
    FLUX := FLU: FLUX PIJ LIBRARY2 VOLMATF ::  TYPE K ;
*----
*   finish up time step
*----
    BURNUP LIBRARY2 := EVO: BURNUP LIBRARY2 FLUX VOLMATF ::
        EDIT 3 SAVE <<Timef>> DAY POWR <<Power>> ;

*----
* change delta t for burnup and final time if required
* Timec is the time variable for which a time increment is changed
*----
    IF Timef Timec = THEN
        IF Timec 150.0 = THEN
            EVALUATE Delt Timec := 90.0 1950.0 ;
        ENDIF ;
        IF Timec 50.0 = THEN
            EVALUATE Delt Timec := 20.0 150.0 ;
        ENDIF ;
        IF Timec 10.0 = THEN
            EVALUATE Delt Timec := 10.0 50.0 ;
        ENDIF ;
        IF Timec 5.0 = THEN
```

```
            EVALUATE Delt Timec := 5.0 10.0 ;
        ENDIF ;
        IF Timec 1.0 = THEN
            EVALUATE Delt Timec := 4.0 5.0 ;
        ENDIF ;
    ENDIF ;

    EVALUATE Timei := Timef ;

*----
ENDWHILE ;
*----

END: ;
QUIT "LIST" .
```

APPENDIX G

DRAGON test deck for an 1/8 of a fuel assembly.


```
* ----
* PU GENERATING CASE U3.7
* 17 x 17 Fuel Assembly
* UOX 3.86w% U235
* Dragon DLIB Library
* BURN POWER (KW/KG) = 36.05
* Total Burnup (MWD/KG) = 45.0
* Porosity = (1 - 95%)
* ----
* Define STRUCTURES and MODULES used
* ----
LINKED_LIST
LIBRARY LIBRARY2 ASSMB VOLMATF PIJ FLUX BURNUP COOL1 COOL2
 DATABASE ISOT PMAP ;

SEQ_ASCII
database ;

MODULE
GEO: SYBILT: USS: ASM: FLU: EVO: EDI: COMPO: DELETE: END: LIB: ;
*
* ----
* Define variables and initialize
* Burnup paremeters
* a) Irradiation
* = 36.05 kw/kg for 0.0 to 1248.0  days
* b) Decay
* = Cooled for 20 years after burnup
* ----
REAL
Power Delt Timec Timei Timef Multi :=
36.05 0.2 1.0 0.0 0.0 3.3 ;
* ----
* Depletion data from file DLIB_J2
* Microscopic cross sections from file DLIB_J2
* ----

*----
* CONCENTRATIONS ARE TAKEN FROM APOLLO-2, USING IMPS FOR THE RELEVANT MEDIA
*
* only the following modifications were performed:
* 1) Apollo-2 gives the concentration of H2O. We used that number for O16 and double that number
* 2) idem for UO2
* 3) ZRNAT does not exist in the Dragon lib, it has been replaced by Zr91
* ----

LIBRARY := LIB: :: EDIT 1
   NMIX 6 CTRA NONE
   SUBG      (*HELIOS TYPE PROBABILITY TABLES*)
   DEPL LIB: DRAGON FIL: DLIB_J2
   MIXS LIB: DRAGON FIL: DLIB_J2
   MIX 1 581.00
                 H1_H2O   4.8208E-02
                 O16      2.4140E-02
                 B10      4.80406520189331E-6
                 B11      1.93369659633997E-5
```

```
   MIX 2 581.00
                 Fe54 7.7079E-06
                 Fe56 1.2189E-04
                 Fe57 2.9237E-06
                 Fe58 3.7210E-07
                 Cr50 2.9568E-06
                 Cr52 5.6953E-05
                 Cr53 6.4572E-06
                 Cr54 1.6041E-06
                 O16  2.7620E-04
                 Zr0  3.8012E-02 2 IRSET 0.0 81
   MIX 3 900.00
                  O16      4.645959E-02
                  U234     1.236619E-06   1 IRSET 0.0 81
                  U235     9.076920E-04   1 IRSET 0.0 81
*                  U236     2.2358E-02  1 IRSET 0.0 81
                  U238     2.232087E-02   1 IRSET 0.0 81
*                   Pu238    2.2031E-17   1 IRSET 0.0 81
*                   Pu239    2.1938E-17   1 IRSET 0.0 81
*                   Pu240    2.1847E-17   1 IRSET 0.0 81
*                   Pu241    2.1756E-17   1 IRSET 0.0 81
*                   Pu242    2.1666E-17   1 IRSET 0.0 81
*                   Am241    2.1756E-17   1 IRSET 0.0 81
   MIX 4 COMB 3 1.0
   MIX 5 COMB 3 1.0
   MIX 6 COMB 3 1.0
;
*----
*----
* Geometry ASSMB : a 17 X 17 normal PWR assembly
* contains C1 : cell without fuel, WH
* C2 : normal fuel cell
* C3 : peripheral cell
* C4 : corner cell
* C5 : IT cell
*----
ASSMB := GEO: :: CAR2D 9 9 EDIT 5
  X- DIAG X+ REFL Y- SYME Y+ DIAG
  CELL  C1 C2 C2 C1 C2 C2 C1 C2 C3
           C2 C2 C2 C2 C2 C2 C2 C3
              C2 C2 C2 C2 C2 C2 C3
                 C1 C2 C2 C1 C2 C3
                    C2 C2 C2 C2 C3
                       C1 C2 C2 C3
                          C2 C2 C3
                             C2 C3
                                C4
 ::: C1 := GEO: CARCEL 2
                MESHX 0.0 1.26 MESHY 0.0 1.26
                RADIUS 0.0 0.5715 0.6121 MIX 1 2 1 ;
 ::: C2 := GEO: CARCEL 5
                MESHX 0.0 1.26 MESHY 0.0 1.26
                   RADIUS 0.0
                          0.2917947
                          0.3690943
                          0.4022112
                          0.4096
```

```
                          0.4750
                          MIX 6 5 4 3 2 1 ;
*                          MIX 3 3 3 3 2 1 ;
 ::: C3 := GEO: CARCEL 5
                MESHX 0.0 1.30    MESHY 0.0 1.26
                   RADIUS 0.0
                          0.2917947
                          0.3690943
                          0.4022112
                          0.4096
                          0.4750
*                         OFFCENTER -0.02 0 0
                          MIX 6 5 4 3 2 1 ;
*                          MIX 3 3 3 3 2 1 ;
 ::: C4 := GEO: CARCEL 5
                MESHX 0.0 1.30    MESHY 0.0 1.30
                   RADIUS 0.0
                          0.2917947
                          0.3690943
                          0.4022112
                          0.4096
                          0.4750
*                         OFFCENTER -0.02 -0.02 0
                          MIX 6 5 4 3 2 1 ;
*                          MIX 3 3 3 3 2 1 ;
;


*----
* Self-Shielding calculation SHI
* Transport calculation SYBILT
* Flux calculation for keff
*----
VOLMATF := SYBILT: ASSMB ::
 EDIT 3
 MAXR 5000 MAXZ 150000 QUA2 20 8 ;


LIBRARY2 := USS: LIBRARY VOLMATF :: EDIT 0 PASS 3
  CALC   REGI W1 U234  ALL
         REGI W1 U235  ALL
         REGI W1 U236  ALL
         REGI W1 Pu238 ALL
         REGI W1 Pu239 ALL
         REGI W1 Pu240 ALL
         REGI W1 Pu241 ALL
         REGI W1 Pu242 ALL
         REGI W1 Zr0   ALL

*        REGI W1 U238 ALL
         REGI W1 U238 3
         REGI W2 U238 4
         REGI W3 U238 5
         REGI W4 U238 6
    ENDC ;

PIJ := ASM: LIBRARY2 VOLMATF ::
```

```
  EDIT 0 ARM ;
FLUX := FLU: PIJ LIBRARY2 VOLMATF ::
  TYPE B ;

PMAP := EDI: FLUX LIBRARY2 VOLMATF ::
  EDIT 3
  MERG
  REGI 1 1 1
       2 2 2 2 2 2
       3 3 3 3 3 3
       4 4 4
       5 5 5 5 5 5
       6 6 6 6 6 6
       7 7 7
       8 8 8 8 8 8
       9 9 9 9 9 8
       10 10 10 10 10 10
       11 11 11 11 11 11
       12 12 12 12 12 12
       13 13 13 13 13 13
       14 14 14 14 14 14
       15 15 15 15 15 15
       16 16 16 16 16 16
       17 17 17 17 17 17
       18 18 18 18 18 18
       19 19 19 19 19 19
       20 20 20 20 20 20
       21 21 21 21 21 21
       22 22 22 22 22 22
       23 23 23 23 23 23
       24 24 24 24 24 24
       25 25 25
       26 26 26 26 26 26
       27 27 27 27 27 27
       28 28 28
       29 29 29 29 29 29
       30 30 30 30 30 30
       31 31 31 31 31 31
       32 32 32 32 32 32
       33 33 33 33 33 33
       34 34 34 34 34 34
       35 35 35 35 35 35
       36 36 36
       37 37 37 37 37 37
       38 38 38 38 38 38
       39 39 39 39 39 39
       40 40 40 40 40 40
       41 41 41 41 41 41
       42 42 42 42 42 42
       43 43 43 43 43 43
       44 44 44 44 44 44
       45 45 45 45 45 45
  COND
  SAVE ;

*----
```

```
* Burnup loop: for first step BURNUP is created
* while for other steps it is modified
*----
EVALUATE Timei := 0.0 ;
WHILE Timei Timec < DO
EVALUATE Timef := Timei Delt + ;
IF Timei 0.0 = THEN
  BURNUP LIBRARY2 := EVO: LIBRARY2 FLUX VOLMATF ::
    EDIT 3 DEPL <<Timei>> <<Timef>> DAY POWR <<Power>> ;
ELSE
  BURNUP LIBRARY2 := EVO: BURNUP LIBRARY2 FLUX VOLMATF ::
    EDIT 3 EXTR DEPL <<Timei>> <<Timef>> DAY POWR <<Power>> ;
ENDIF ;
LIBRARY2 := USS: LIBRARY LIBRARY2 VOLMATF :: EDIT 0 PASS 3
  CALC   REGI W1 U234  ALL
         REGI W1 U235  ALL
         REGI W1 U236  ALL
         REGI W1 Pu238 ALL
         REGI W1 Pu239 ALL
         REGI W1 Pu240 ALL
         REGI W1 Pu241 ALL
         REGI W1 Pu242 ALL
         REGI W1 Zr0   ALL

*        REGI W1 U238 ALL
         REGI W1 U238 3
         REGI W2 U238 4
         REGI W3 U238 5
         REGI W4 U238 6
    ENDC ;

PIJ := DELETE: PIJ ;
PIJ := ASM: LIBRARY2 VOLMATF ::
  EDIT 0 ARM ;
FLUX := FLU: FLUX PIJ LIBRARY2 VOLMATF ::
  TYPE B ;

PMAP := EDI: PMAP FLUX LIBRARY2 VOLMATF ::
  EDIT 3
  MERG
  REGI 1 1 1
       2 2 2 2 2 2
       3 3 3 3 3 3
       4 4 4
       5 5 5 5 5 5
       6 6 6 6 6 6
       7 7 7
       8 8 8 8 8 8
       9 9 9 9 9 8
       10 10 10 10 10 10
       11 11 11 11 11 11
       12 12 12 12 12 12
       13 13 13 13 13 13
       14 14 14 14 14 14
       15 15 15 15 15 15
       16 16 16 16 16 16
       17 17 17 17 17 17
```

```
       18 18 18 18 18 18
       19 19 19 19 19 19
       20 20 20 20 20 20
       21 21 21 21 21 21
       22 22 22 22 22 22
       23 23 23 23 23 23
       24 24 24 24 24 24
       25 25 25
       26 26 26 26 26 26
       27 27 27 27 27 27
       28 28 28
       29 29 29 29 29 29
       30 30 30 30 30 30
       31 31 31 31 31 31
       32 32 32 32 32 32
       33 33 33 33 33 33
       34 34 34 34 34 34
       35 35 35 35 35 35
       36 36 36
       37 37 37 37 37 37
       38 38 38 38 38 38
       39 39 39 39 39 39
       40 40 40 40 40 40
       41 41 41 41 41 41
       42 42 42 42 42 42
       43 43 43 43 43 43
       44 44 44 44 44 44
       45 45 45 45 45 45
  COND
  SAVE ;

BURNUP LIBRARY2 := EVO: BURNUP LIBRARY2 FLUX VOLMATF ::
   EDIT 3 SAVE <<Timef>> DAY POWR <<Power>> ;

*----
* change delta t for burnup and final time if required
*----
IF Timef Timec = THEN
  IF Timec 830.0 = THEN
   EVALUATE Delt Timec := 209.0 1248.0 ;
  ENDIF ;
  IF Timec 150.0 = THEN
   EVALUATE Delt Timec := 136.0 830.0 ;
  ENDIF ;
  IF Timec 50.0 = THEN
   EVALUATE Delt Timec := 20.0 150.0 ;
  ENDIF ;
  IF Timec 10.0 = THEN
   EVALUATE Delt Timec := 10.0 50.0 ;
  ENDIF ;
  IF Timec 5.0 = THEN
   EVALUATE Delt Timec := 5.0 10.0 ;
  ENDIF ;
  IF Timec 1.0 = THEN
   EVALUATE Delt Timec := 4.0 5.0 ;
  ENDIF ;
ENDIF ;
```

```
 EVALUATE Timei := Timef ;
ENDWHILE ;

*----
* COOL2 loop: cool for 20 years
*----
EVALUATE Timei := 0.0 ;
EVALUATE Timef Multi := 0.001 3.3 ;
WHILE Timei Timef < DO
IF Timei 0.0 = THEN
  COOL2 LIBRARY2 := EVO: LIBRARY2 VOLMATF ::
     EDIT 3 EXTR DEPL <<Timei>> <<Timef>> YEAR COOL ;
ELSE
  COOL2 LIBRARY2 := EVO: COOL2 LIBRARY2 VOLMATF ::
     EDIT 3 EXTR DEPL <<Timei>> <<Timef>> YEAR COOL ;
ENDIF ;
EVALUATE Timei := Timef ;
IF Timef 13.0 < THEN
  EVALUATE Timef := Timei Multi * ;
ELSE
  EVALUATE Timef := 20.0 ;
ENDIF ;
ENDWHILE ;

QUIT "LIST" .
```

APPENDIX H

Perl Script description will be moved here.

Program Initialization
- -Data Organization
- -Data Initialization
- -Variable Data Arrays
- Loop Initialization
  - -Initialize Changes
  - -Calculate Dependents
  - -Create Replacement Hash
  - -Copy and Rename Template File
  - -Search Through Copy and Replace
  - -Queue up Copy on Grove
- Loop End

Program End

After the initialization of the program, the first step is to organize how all the data used in the calculations will be stored. Hashes are used to store this data. For example, there is a uranium hash called ‘u’. This hash contains keys like ‘molarMass’, ‘density’, and ‘atomDensity’ with corresponding values to each of these keys that are the molar mass, density, and atomic density for uranium respectively. By organizing the data in this manner, whenever equations are written in the code they will be that much easier to decipher because of how the data is referenced.

The next step is to initialize the data that will remain constant during the program by simply listing the parameters and setting them equal to their respective value.

The next step is to first decide what variables will change within the loop (usually uranium-235 and beryllium oxide content) and secondly to create the array of values that these changing variables will take on for each loop.

Next, there is a loop over all of the changes in which the rest of the program operates. The first step within the loop is to choose and initialize the changes from the variable arrays that will be used.

Next is the process of calculating all of the intermediate and final parameters used in the DRAGON deck that depend upon the values being looped over. Some of these parameters include the specific power of the reactor, molar masses, and mixture atom densities. After all the necessary values are known, the replacement hash is created and formatted. As described above, this hash is a list of 'keys' that are embedded in the template DRAGON deck where it's corresponding value is supposed to exist.

Next is file management; the template file is copied and renamed. Then the file is opened and every line is examined to find the specific search codes listed in the replacement hash and replace those search codes with the corresponding value in the hash. What remains is a DRAGON deck with the necessary changes made to it.

The last step within the loop is to queue the deck in the computer cluster at the Department of Nuclear Engineering at Texas A&M University called Grove. The loop ends and is repeated for each change listed, running a DRAGON simulation for each change.

The result is a program that creates a working DRAGON deck for each of the changes specified and queues the deck in the computational cluster that the nuclear engineering department has available, exponentially decreasing the amount of time spent in setting up the DRAGON decks for each set of simulations required.

Perl and shell scripts are also used on the back-end of data analysis considering the large amount of information provided by the DRAGON code. For example, there is a Perl script that searches through the DRAGON output and creates a text file with a list of k_{inf} values for each burnup step taken. There is a shell script that runs that Perl script for each output file in whatever folder is specified. And finally, there is a Perl script that takes all these k_{inf} versus burnup files and organizes them into a single file so as to be easily pasted and analyzed in a spreadsheet.

APPENDIX I

First plot replication:

The first step in the modeling of the UO_2-BeO concept fuel was to benchmark this model against data already developed on the subject. Figure I.1 shows kinf vs time for three different fuel cases.

These fuel cases represent two standard UO_2 fuels at 5% and 4.64% enrichment and concept fuel case with 10% by volume BeO and the rest UO_2 at 5% enrichment. As shown from the data listed in the figure, the effective fuel temperature used was about 100 K less for the 10% BeO case as opposed to the pure UO_2 cases, but also, because there is 10% less fuel in that case, the specific power was increased to compensate.

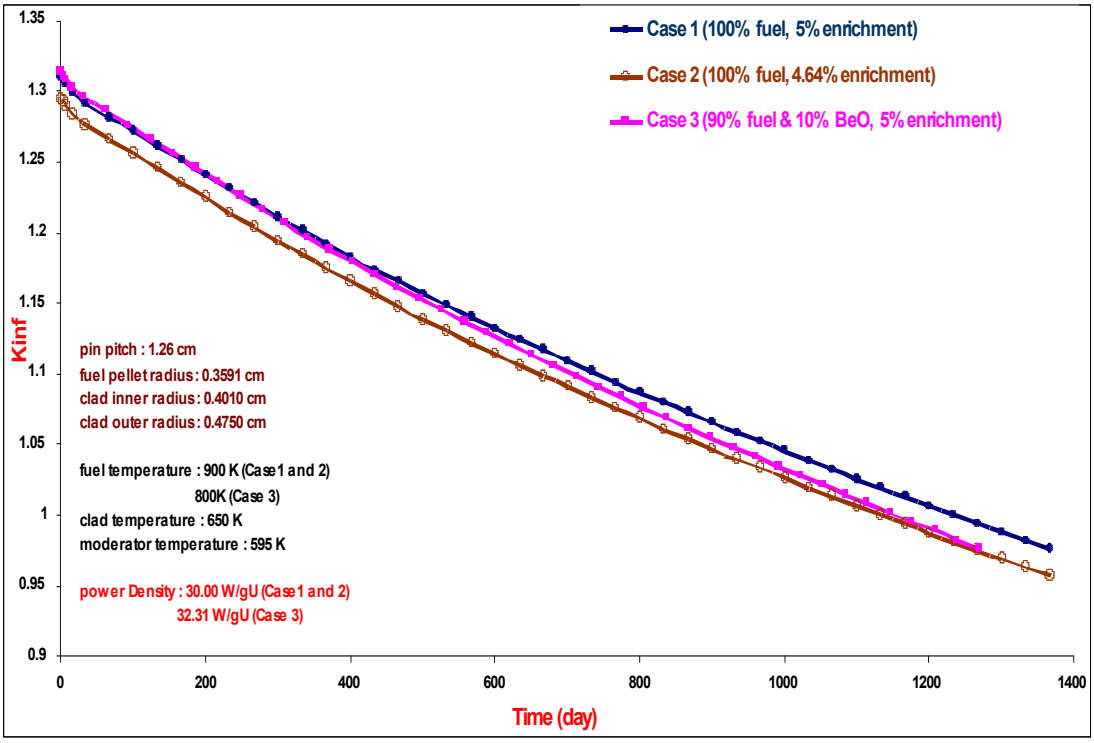


Figure I.1 Benchmark plot by Purdue University.

Multiple simulations were done in order to make sure that the data in Figure I.1 was correct. For example, the fuel pellet radius listed in Figure I.1 seemed to be exceptionally low for inner cladding radius given and it was speculated that the five and the nine in the radius might have been switched accidently. So both 0.3591 cm and 0.3951 cm were simulated and the results compared.

In order to replicate Figure I.1 multiple simulation runs were necessary, because some of the data listed on the side was suspect. For example, the radius of the fuel pellet as compared to the radius of the cladding seemed much to small (0.3591cm to 0.4010cm) for a typical nuclear reactor. It was suggested that perhaps the '5' and the '9' were switched accidently.

In addition, the figure does not specify atom % or weight % for the U-235 enrichments listed. So these were tested just to be sure as well. Table I.1 lists the first set of simulations done to replicate Figure I.1.

Table I.1 Initial set of simulation constraints.

Run #	U-235 wt%	BeO vol%	Rfuel (cm)	T_fuel (K)	T_clad (K)	T_cool (K)	Specific power (W/gU)
1	4.64	0	0.3951	900	650	595	30
2	5	0	0.3951	900	650	595	30
3	4.58	0	0.3951	900	650	595	30
4	4.94	0	0.3951	900	650	595	30
5	5	10	0.3951	800	650	595	32.31
6	4.94	10	0.3951	800	650	595	32.31
7	5	9.09	0.3951	800	650	595	32.31
8	4.94	9.09	0.3951	800	650	595	32.31
9	4.64	0	0.3591	900	650	595	30
10	5	0	0.3591	900	650	595	30
11	4.58	0	0.3591	900	650	595	30
12	4.94	0	0.3591	900	650	595	30
13	5	10	0.3591	800	650	595	32.31
14	4.94	10	0.3591	800	650	595	32.31
15	5	9.09	0.3591	800	650	595	32.31
16	4.94	9.09	0.3591	800	650	595	32.31

Out of these tests, runs 9, 10, and 13 seemed to best match Figure I.1. Unfortunately, the data for Figure I.1 is not available and all the comparisons had to be made visually. For this reason, the figures generated from these simulations have the same limits as Figure I.1 to make visual comparisons easier.

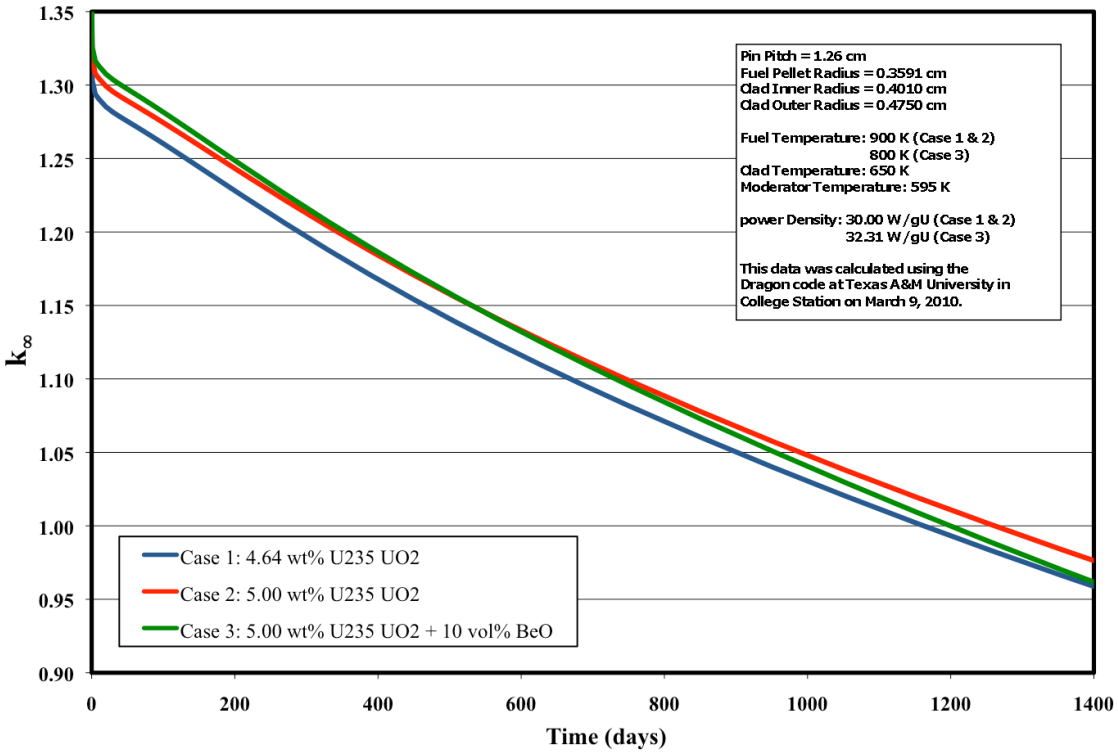


Figure I.2 Simulation runs 9, 10, and 13 plot of reactivity vs time.

Figure I.2 are the results from simulation runs 9, 10, and 13, which were done at the radius listed in Figure I.1 (0.3591cm).

Figure I.3 are the results from simulation runs 1, 2, and 5, which were done at the proposed radius of 0.3951cm and although they follow the same general pattern as the results in Figure I.1, the starting and ending points are noticeably lower on the graph when compared to Figure I.3.

Unfortunately, although these results seem to very closely replicate the results of Figure I.1, they are not valid because the methods used to calculate the atom densities in these simulations are incorrect. After comparing methods with Dr. Jean Ragusa, and verifying atom density formulae, the final replication plots were created as shown in the next section.

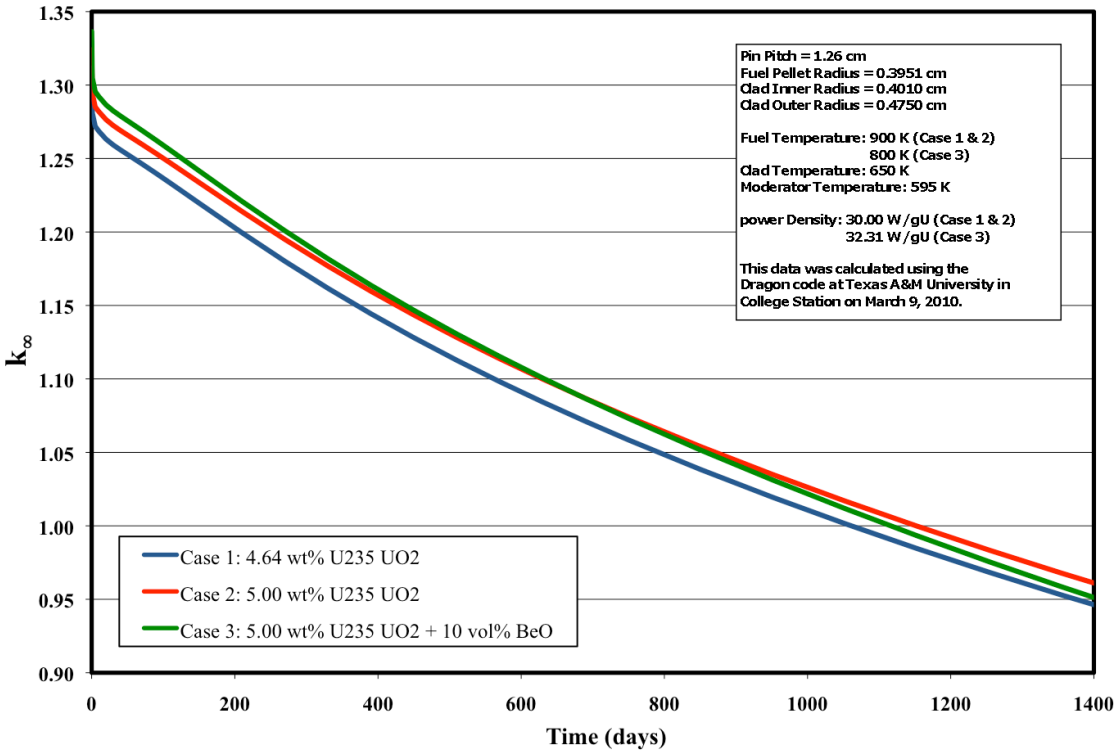


Figure I.3 Simulation runs 1, 2, and 5 plot of reactivity vs time.

In Figures I.4 and I.5, comparisons between the correctly (b) and incorrectly calculated replications of Figure I.1 have been placed side by side in order to emphasize the difference between what it was thought to be, and what it actually is. The mistake made in the calculation of the atom densities of the coolant, cladding, and fuel lowered each respective atom density by up to a factor of two. This is why the reactivity of the Figure I.4 (b) is significantly higher than the reactivity of Figure I.4 (a).

The original point of this was to show the difference between a fuel with the BeO additive and without. It was created by first plotting cases 2 and 3 and then finding the uranium-235 enrichment for a non-BeO fuel that would end at the same k_∞ as the 5 wt% U-235 10% BeO case; this enrichment turned out to be approximately 4.64 wt%.

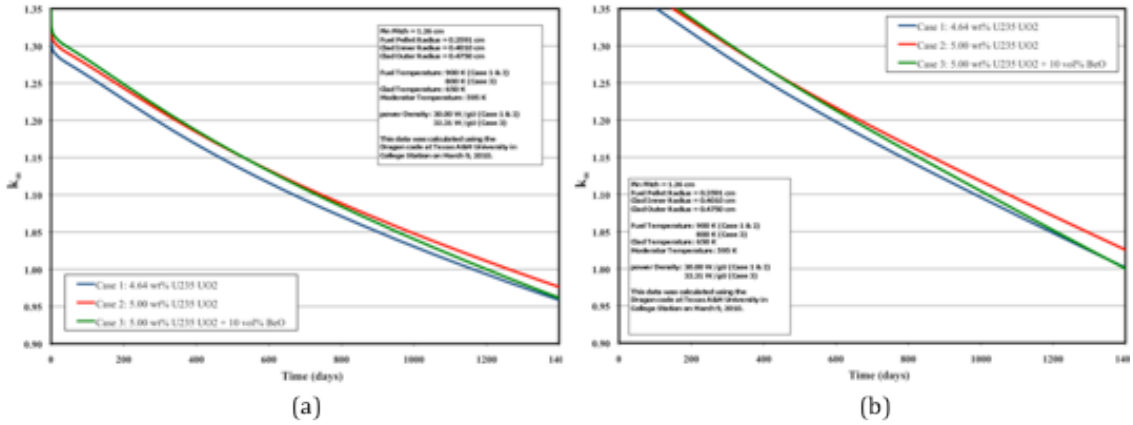


Figure I.4 Plot comparison between Figure I.1 (a) and the same calculations redone with correct correlations Figure I.4 (b).

The comparisons made for Figure I.5 are essentially identical to the observations of Figure I.4 because the only difference in conditions between these figures is the radius of the fuel pellet (radius is 0.3591 cm for Figure I.4 and 0.3951 cm for Figure I.5). The main difference is that both Figure I.5 (a) and (b) are shifted down in reactivity slightly from Figures I.4 (a) and (b).

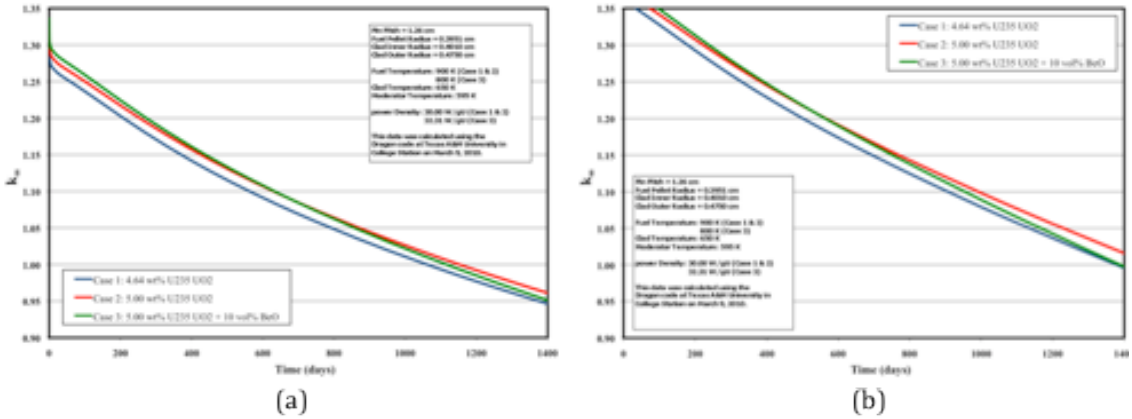


Figure I.5 Plot comparison between Figure I.1 (a) and the same calculations redone with correct correlations Figure I.6 (b).

Through the process of verifying these plots, some of the values used in their construction came into question. Specifically, the effective temperatures used for the fuel in these reactors: 900 K for normal UO_2, and 800 K for UO_2-10 vol% BeO. These values were verified to be accurate to within a couple percent of error for the conditions in which they were used.

Figures I.6 and I.7 show the DRAGON simulations using the pellet radius of 0.3591 cm with the limits that Figure I.1 have (Figure I.6) and with limits showing the entirety of its data (Figure I.7). The same is done for Figures I.8 and I.9 the pellet radius 0.3951 cm.

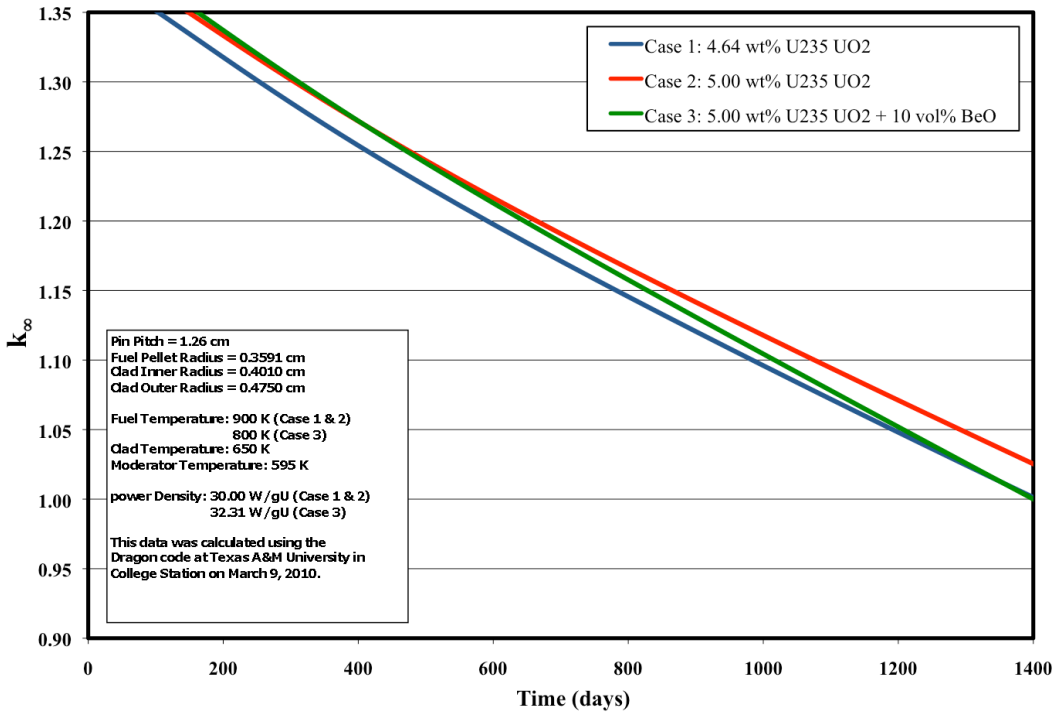


Figure I.6 Replication of Figure I.1 using DRAGON with the same limits (with pellet radius = 0.3591 cm).

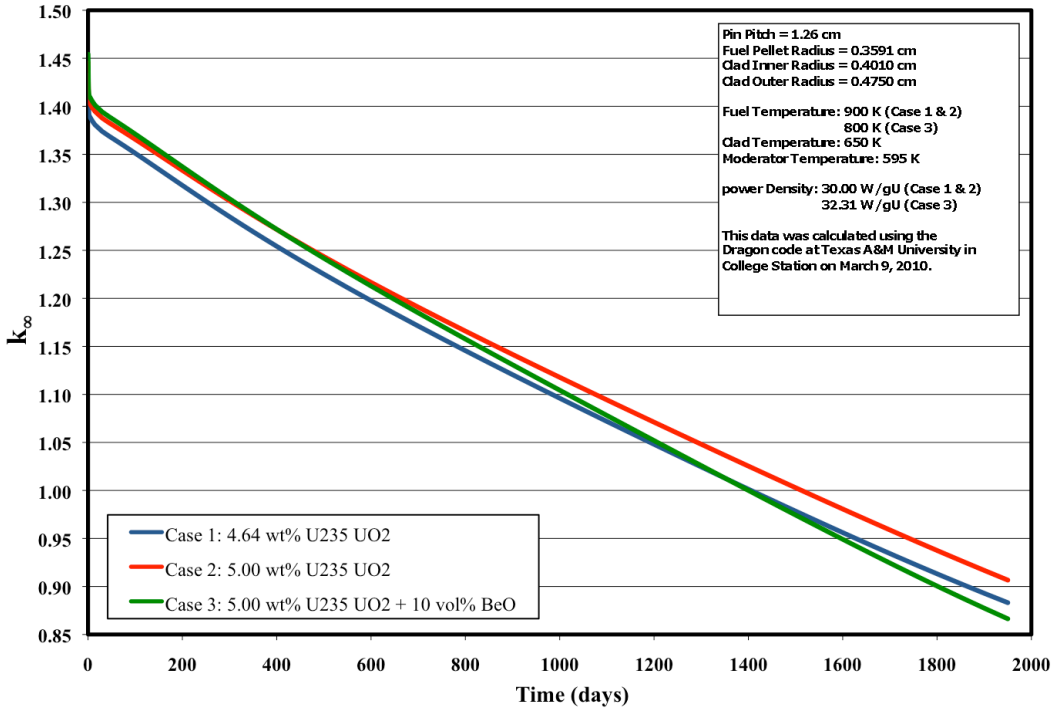


Figure I.7 Replication of Figure I.1 using DRAGON (with pellet radius = 0.3591 cm).

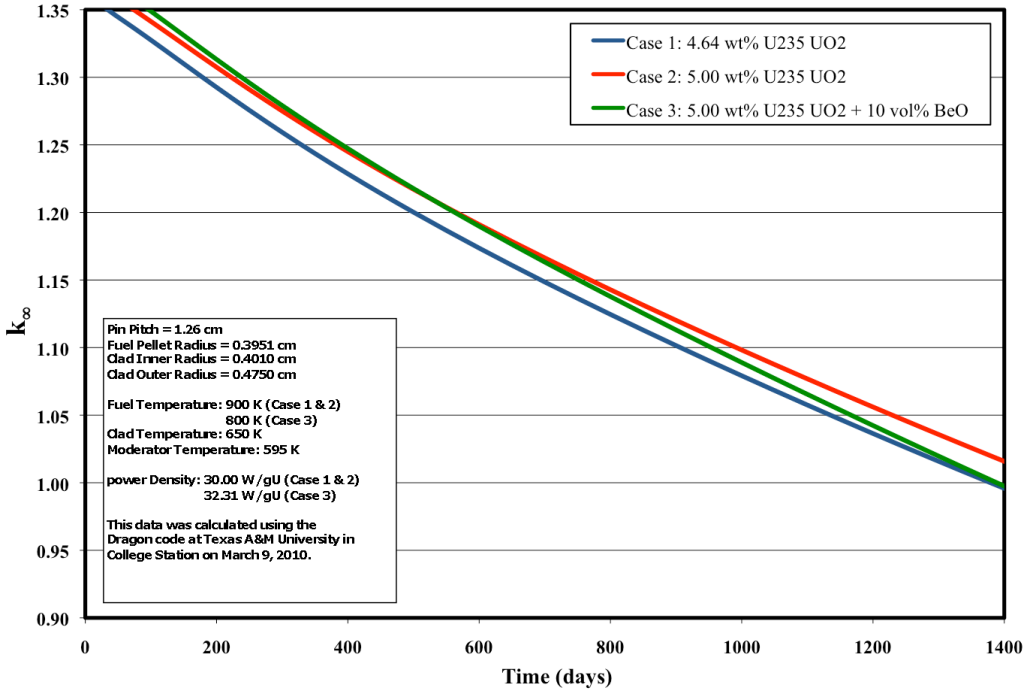


Figure I.8 Replication of Figure I.1 using DRAGON with the same limits (with pellet radius = 0.3951 cm).

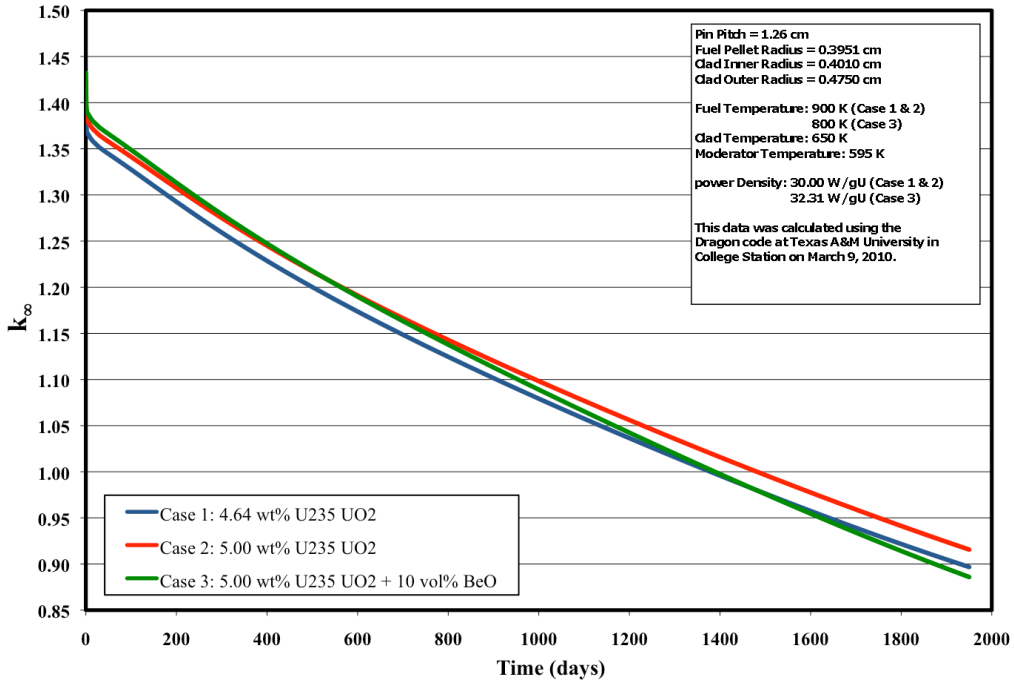


Figure I.9 Replication of Figure I.1 using DRAGON (with pellet radius = 0.3951 cm).

It is obvious from these plots that the results calculated from DRAGON do not replicate the results presented in Figure I.1. After reevaluating and re-deriving all the calculations done for the DRAGON simulations, comparing the results DRAGON gave for a typical PWR case to other neutronic codes, and taking into account that the data for Figure I.1 is unavailable, the decision was made that these simulations done in DRAGON are correct.

APPENDIX J

J. MATERIAL FABRICATION

J.1 The Process and System

The UO_2-BeO Fabrication Process, as shown in Figure J.1 below, has many steps and required processes including ceramic compacting, grinding, sieving, milling, and sintering. These processes require both general and specialized equipment including a mortar and pestle, a ball mill, sieves and a shaker table, a hydraulic press, a punch and die system, and a furnace system.

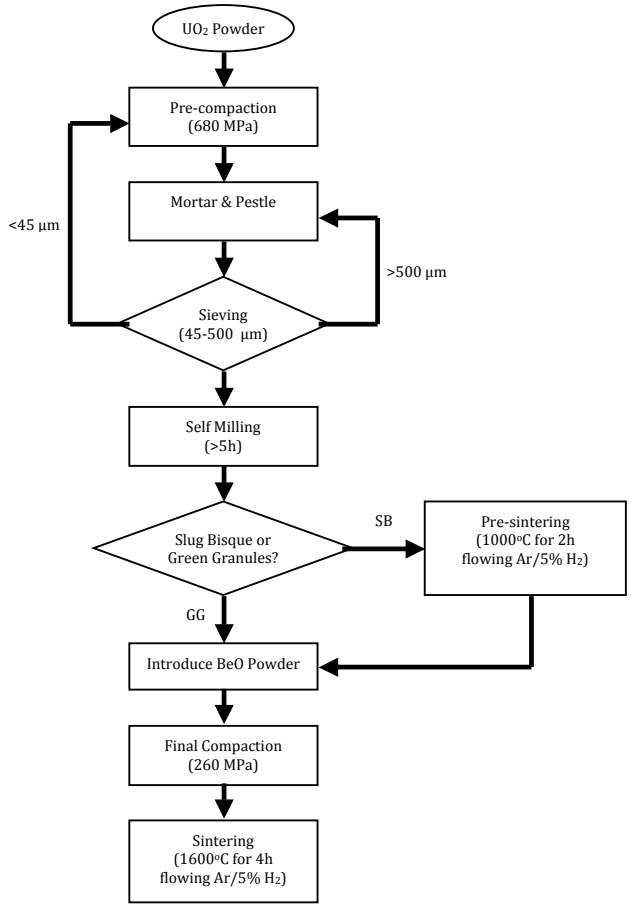


Figure J.1 Flowchart for the Material Fabrication Method of UO_2-BeO fuel concept (SB = Slug-Bisque and GG = Green Granules) [9].

The first step is pre-compaction, where the uranium dioxide powder is pressed using the Carver C Hydraulic Press and the punch and die system to create a pellet. This pellet is then ground into small granules using a mortar and pestle. Sieves are then used to select granules that are within the optimal size range (45-500 µm).

The whole purpose of the first several steps is to create larger granules of UO_2 powder so that the BeO powder will be able to coat the larger UO_2 granules creating a layer of BeO around the UO_2 granules. In this way, when the BeO "coated" UO_2 granules are pressed together in the final compaction a continuous, three-dimensional lattice of BeO is created in the microstructure as shown in Figure J.2.

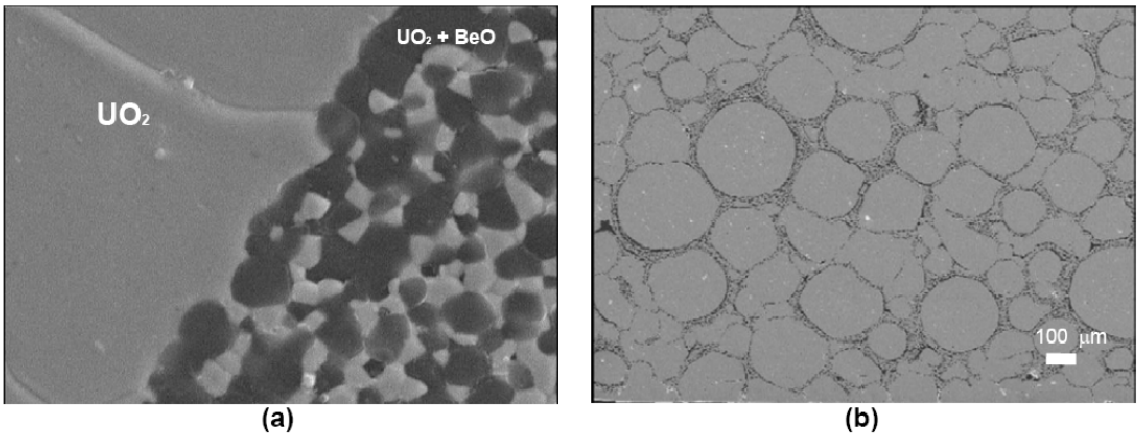


Figure J.2 Electron microscope images of a cross section of a UO2-BeO sintered pellet with a high magnification of mixed oxide matrix (a), and a lower magnification view of the general microstructure (b) [1] [10].

The next step is the self-milling process. The UO_2 granules are placed in the ball mill with no grinding media and milled for approximately five hours. This makes the granules more spherical and uniform; it also creates an excess amount of UO_2 powder that will be necessary in the final compaction, so it isn't sieved out.

The green granules method uses a mixed oxide continuous lattice of beryllium oxide and uranium dioxide as shown in Figure J.2 while the slug-bisque method uses pre-sintered pellets and has only beryllium oxide in its continuous lattice. Of the two beryllium-oxide methods designed, the green granules method resulted in much more favorable thermal conductivity properties as compared to the slug-bisque method. Because of this, the slug-bisque method will not be pursued in this research.

After the self-milling is completed, the BeO powder is introduced to the mixture. Continue to self-mill the mixture for another 30 minutes to ensure that the BeO powder has mixed with the UO_2 fines and coated the UO_2 granules sufficiently and uniformly. There will be loose UO_2 powder and BeO powder; these loose powders will be used in the final compaction to create the mixed oxide matrix.

The final compaction is the next step, and uses a significantly lower pressure then the pre-compaction step (260 MPa). After the final pellet is created, it should be placed in the ceramic processing vessel and sintered for four hours at 1600^{o}C in flowing Ar/5% H_2. The result is a UO_2 fuel pellet with a continuous lattice of BeO [8], [9], [10].

All the powder operations described in this section were demonstrated using Al_2O_3 powder, not UO_2 and BeO powder.

J.1.1 Pellet Pressing

The process of pressing ceramic powders involves the powder with a compatible binding agent and high pressure applied at room temperature. This is achieved through the use of a hydraulic press (Figure J.3) and a punch and die (Figure J.4). The powder used in the test cases was alumina powder, and the binding agent used is zinc stearate (approximately 0.3wt% zinc stearate is used) [26], [27].

Figure J.3 Carver C Hydraulic Press in the Fuel Cycle and Materials Lab.

Figure J.4 Example punch and die system: (a) heat treated die, (b) the short and long heat treated punches, and (c) a heat treated extraction cylinder.

There are two pressures used in the process: the pre-compaction pressure of 680 MPa, and the final compaction pressure of 260 MPa. The punch and die system was designed

and tested using the higher pressure. A total of three punch and die systems were created; the first two had design flaws and had to be replaced. All of the punch and die systems follow the same basic design scheme: a cylindrical die with a hole down the center, and two punches, one of which has to be taller than the die is so that the pellet inside can be easily removed (see Figure J.5). The final pellet pressing system uses a Carver C Hydraulic Press, and punch and die system was made from heat-treated A-2 Tool Steel. The heat-treatment schematic used for the A-2 tool steel is shown in Appendix I.

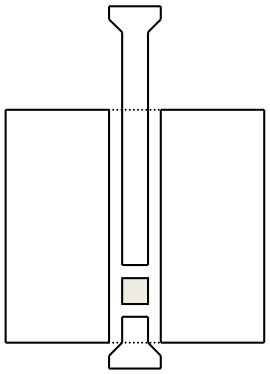


Figure J.5 Schematic of a typical punch and die system.

J.1.1.1 Final Punch and Die System

All of the pieces in the final punch and die system were made using A-2 tool steel and are heat-treated (except for the angle iron stand). The smaller punch length is elongated to facilitate less potential angular movement.

This successful punch and die design has included several innovations: an angle iron stand for the hydraulic press and an extraction piece. The extraction piece is meant to aid in the extraction of the compacted pellet from the punch and die. It is essentially an annular cylinder with an outer radius equal to that of the die, and an inner radius large enough to accommodate a punch. The height of the extraction piece is taller than the

total height of the smaller punch. Therefore, the hydraulic press may be used to push both the pellet and smaller punch out of the die using a gentle, constant pressure in order to minimize damage to the pellet.

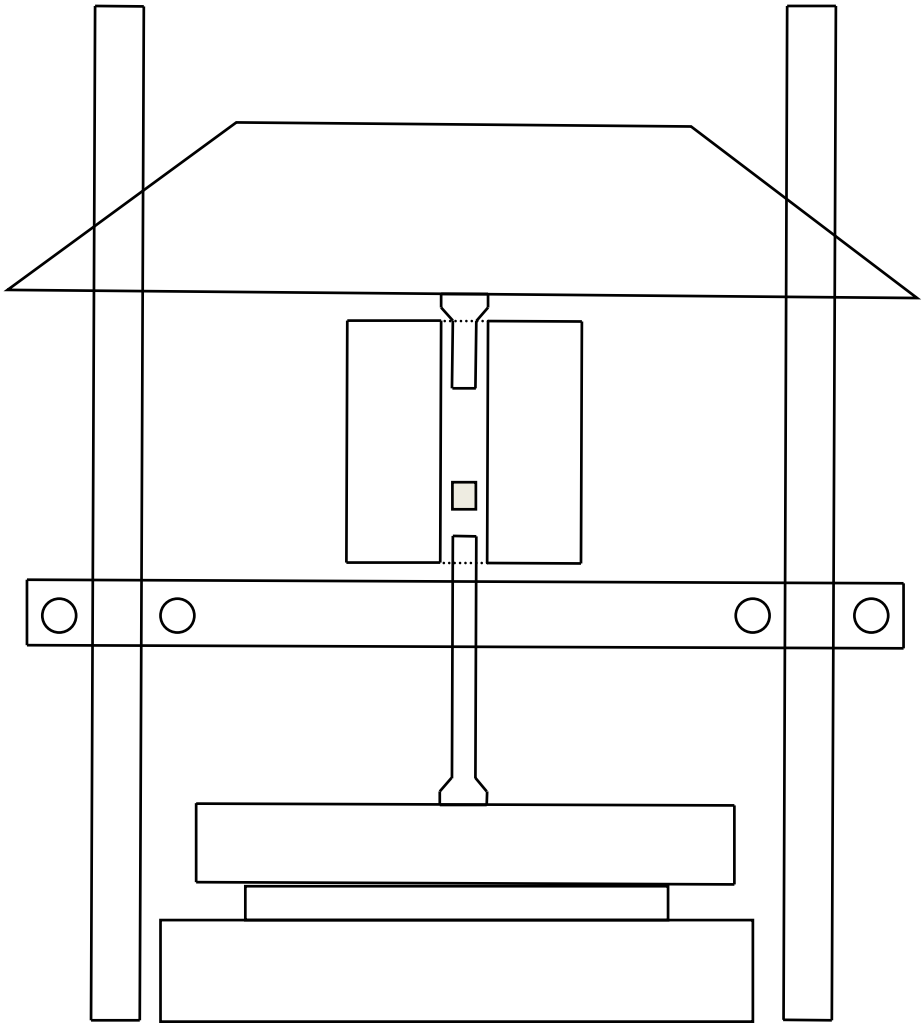


Figure J.6 Basic schematic of the final punch and die system set up on a hydraulic press.

The angle iron stand (Figure J.7) is designed to facilitate the set-up of the entire system and to enable the double-action punch movement. The two pieces of angle iron are fixed

in position with respect to the moving hydraulic press plates as shown in Figure J.6. The die sits on top of the angle iron stand with the longer punch going into the bottom of the die through the gap in the angle iron stand.

This set-up alleviates the stress between the punches and the die by allowing the stand to hold the die in place and the punches to freely press the powder from above and below.

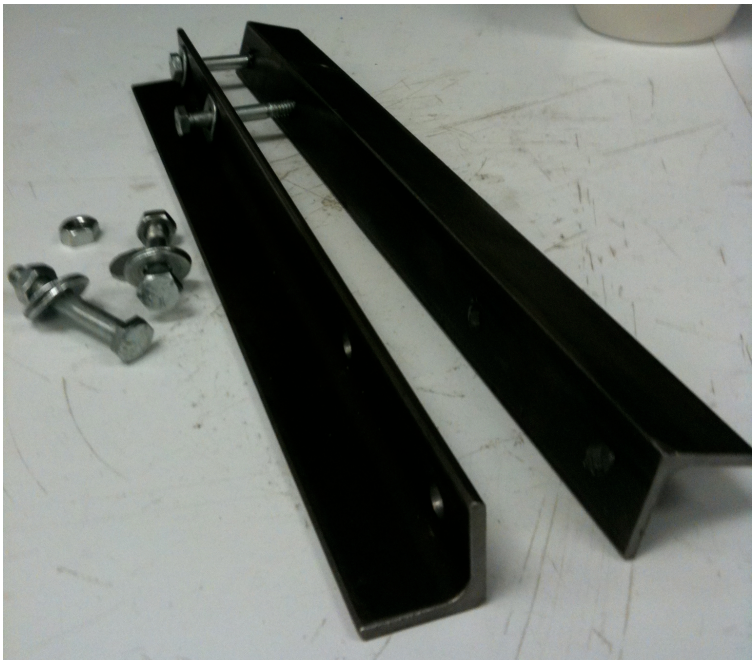


Figure J.7 Angle iron stand.

The setup procedure involves attaching the stand to the hydraulic press in a level position; it is very important that the stand be level to minimize damage to the punches and die. At this point, the longer punch and the die can be set up on the stand as shown in Figure J.8 and J.9. Now, the powder needs to be inserted into the die using a tube and a funnel. After the powder is in the die, the smaller punch is slowly inserted into the die.

Figure J.8 Close-up of the punch and die system on the press just prior to compaction.

Figure J.9 Pellet pressing apparatus while filling the die with powder at two angles.

At this point pressure is applied with a slow ramp to the desired pressure; check that the punch and die stays in proper alignment. If the setup is out of alignment, stop and release pressure immediately. Also, the punch and die assembly should be centered on the press as much as possible because this will reduce torque on the system. The required force using this punch and die system is approximately 48,500 Newtons (~11,000 lb). The ceramic pellet should be ready after approximately 30-60 seconds of applied pressure.

The process of pellet extraction does not need the angle iron stand, so that is removed from the hydraulic press and the punch and die is set-up as shown in Figure J.10. Also, if the smaller punch slides out (gently pull with fingers) then do that prior to extraction.

Figure J.10 Extraction setup.

After the extraction is setup properly assembled, apply constant gentle pressure with the hydraulic press until the longer punch is all the way through the die as shown in Figure J.11. Carefully remove the extraction cylinder and the smaller punch without damaging the pellet. The pellet should intact on top of the punch; gently remove and store. The pellet should look something similar to Figure J.12 (for alumina pellets).

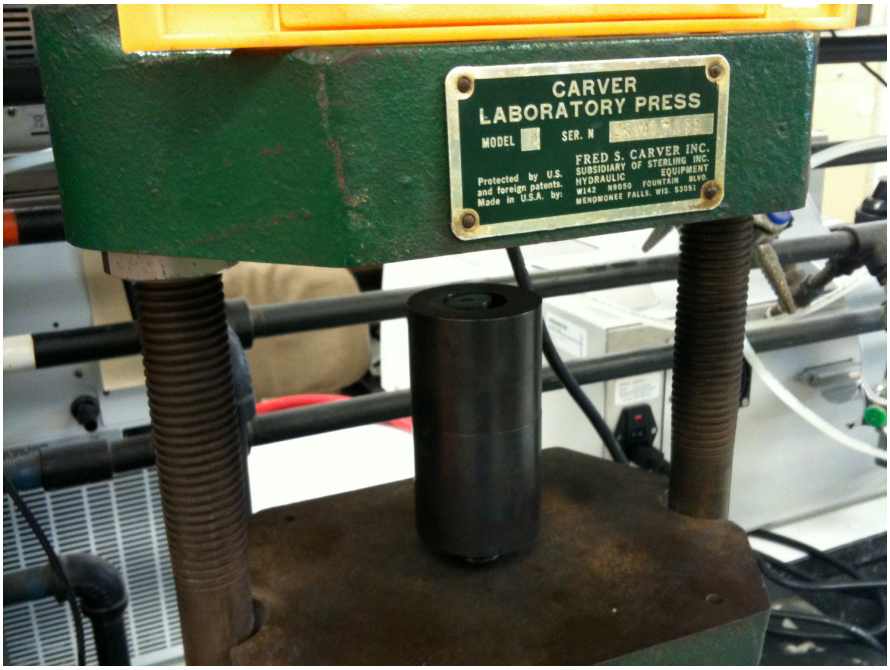


Figure J.11 Extraction process.

Figure J.12 Pre-compacted alumina pellets.

J.1.2 Grinding and Sieving

Grinding the pre-compacted pellets into granules is accomplished using a mortar and pestle and a series of sieves (Figure J.13). Granules that are too large are sent back to the mortar and pestle to be ground again and granules that are too small can be pre-compacted again.

Figure J.13 The mortar and pestle (a) and the sieves (b) used.

The sieve mesh sizes used are mesh size 35 and mesh size 325. Mesh size 325 corresponds to a hole size of approximately 44 microns, and a mesh size of 35 corresponds to a hole size of approximately 500 microns. The shaker table, as shown in Figure J.14, is used to speed up the process of sieving, and it was found that a sieving time of about 10 minutes sorts out the granules very well when using aluminum oxide powder to demonstrate the process.

There are three results from sieving: 1) granules are too small, 2) granules are too large, or 3) granules are within the correct size range. Of these results, the latter two are the most common. Very little (if any) of the sieved granules fall through the 325 mesh size sieve, so most of the time, the larger granules may be ground further and sieved again.

Figure J.14 The shaker table with example sieves attached.

J.1.3 Milling

The self-milling is a simple process that requires a ball mill (Figure J.15) and time. The material to be milled is placed in a glass jar (preferably not plastic or any other material that might leave particulate inside after long milling periods), sealed, and placed on the mill rollers. No milling balls or other milling media are used. The granules self mill themselves to round their surfaces and produce a fine residue of powder.

The required amount of time will vary depending upon the speed of rotation, the size of the jar used on the ball mill, and the amount of granulated material in the jar.

Figure J.15 Ball Mill.

The self-milling process is used for two reasons: to make the granules uniform and spherical and to create excess loose powder. Both of these characteristics have a positive effect on the final microstructure of the UO_2-BeO with respect to thermal conductivity. Making the granules more uniform and spherical results in a more uniform coverage of BeO powder and with that a more uniform continuous lattice of BeO in the microstructure.

After the self-milling process is completed, the BeO is to be introduced to the UO_2 granules by placing the BeO powder in the jar and putting it back on the ball mill for another 20-30 minutes to ensure even coverage of the UO_2 granules. At this point, a mixture of UO_2 granules coated in BeO with loose BeO powder and UO_2 will be created in the jar and ready for the final compaction stage.

J.1.4 Sintering

The sintering step was never demonstrated for this thesis, but a controlled-atmosphere sintering assembly was designed and built. The process of sintering is a physical and chemical process where individual powder particles coalesce into a compact whole under increased pressures and temperatures. As shown in Figure J.16, surface tension drives the diffusion of atoms in the individual particles to the contact points between particles. As time progresses, these powder particles are drawn together, eventually eliminating the pores in between them over time [28].

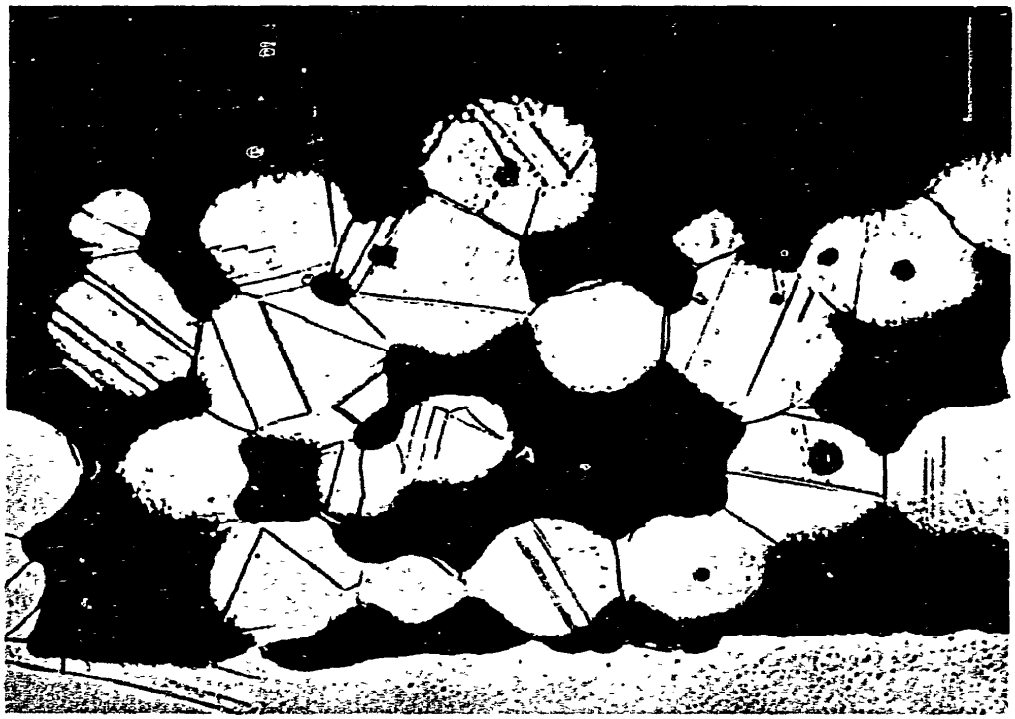


Figure J.16 Spherical copper powder sintered at 800^{o}C for 6h at 500x magnification [28].

Porosity is reduced through volumetric diffusion for ceramics. As shown in Figure J.17, the neck created by particle contact from sintering forms very quickly (this figure was 10 minutes of sintering), and as the process continues, more porosity is displaced, creating a

material that is homogenized and dense. Both of these properties serve to increase the thermal conductivity of the material (and allow more fuel per unit volume from a nuclear reactor perspective) [28].

Figure J.17 The neck created between two wires prepared from homogenized Cu + 4.5 at. % Ag alloy after sintering for 10 min at 920^{o}C [28].

J.1.4.1 The Furnace

The furnace availabe for the sintering system described here is an open-air furnace (Deltech) with twelve molydisilicide heating elements with a maximum temperature of ~1650^{o}C (Figure J.18). Because of the high temperatures used, and because it is an open-air furnace, a controlled atmosphere ceramic processing vessel was designed.

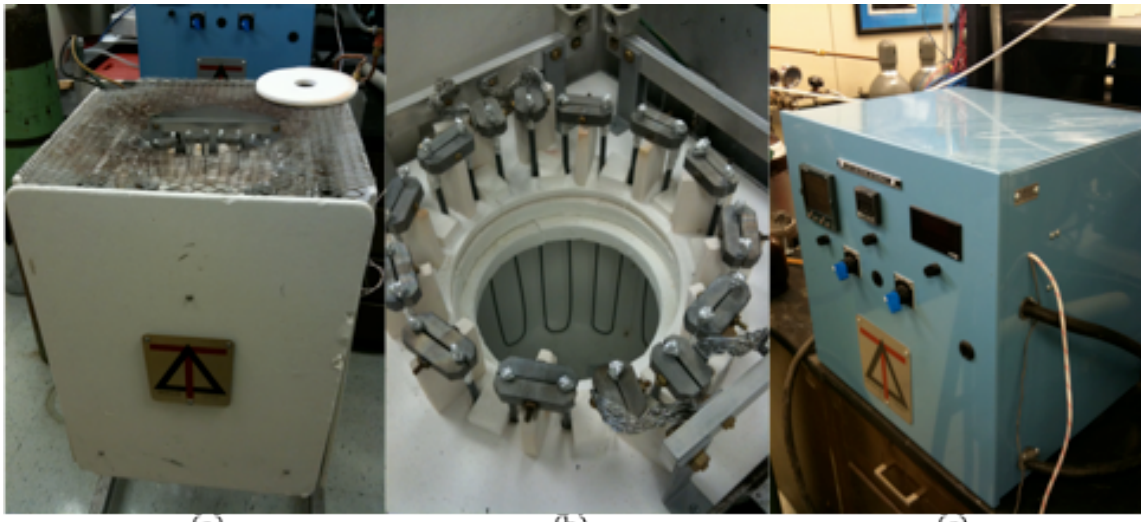


Figure J.18 The custom Deltech furnace (a), with a close-up of the fuel elements and hot zone (b), and the power supply and control panel (c).

J.1.4.2 The Processing Vessel

The processing vessel was designed to control the atmosphere around a sintering specimen at elevated temperatures. The components inside the processing vessel must withstand temperatures above 1600^{o}C. In order to hold an inert atmosphere in the processing vessel, it needs to have a gas inlet and outlet for the flowing Ar gas during the sintering process. The temperature near the sample must be used to control the furnace, so a long type C thermocouple was installed.

The vessel is a closed end alumina ceramic tube sealed at the top with a stainless steel flange (Figure J.19). The flange provides feedthrough access into the vessel for the central LVDT tube, the type C thermocouple, the inlet/outlet support rod, and the other two support rods. This entire structure is placed inside the large ceramic vessel while at temperature.

An LVDT is a linear variable differential transducer, and is used to measure linear displacement. There are three solenoid coils in the annular core of the LVDT. The

center coil is the primary through which an alternating current is passed, and the two outer coils are the secondaries, with a cylindrical magnetic core in the tube. This current in the primary causes a voltage to be induced in the secondaries, and as the core moves, the voltages induced in the secondaries change with respect to its position in the tube.

Connected under the flange are the support rods, heat shields, and the sample holder as shown in Figure J.20. The support rods extend the length of the ceramic vessel and are there to hold the heat shields and sample holder. The heat shields exist to ensure that the steel flange and other metal parts on the top of the processing vessel do not exceed their melting points. There are three 1" thick alumina SALI-2 heat shields in place above the heating zone of the processing vessel. The sample holder is another 1" thick alumina circle with a hole drilled for an yttrium oxide crucible large enough for the pellets.

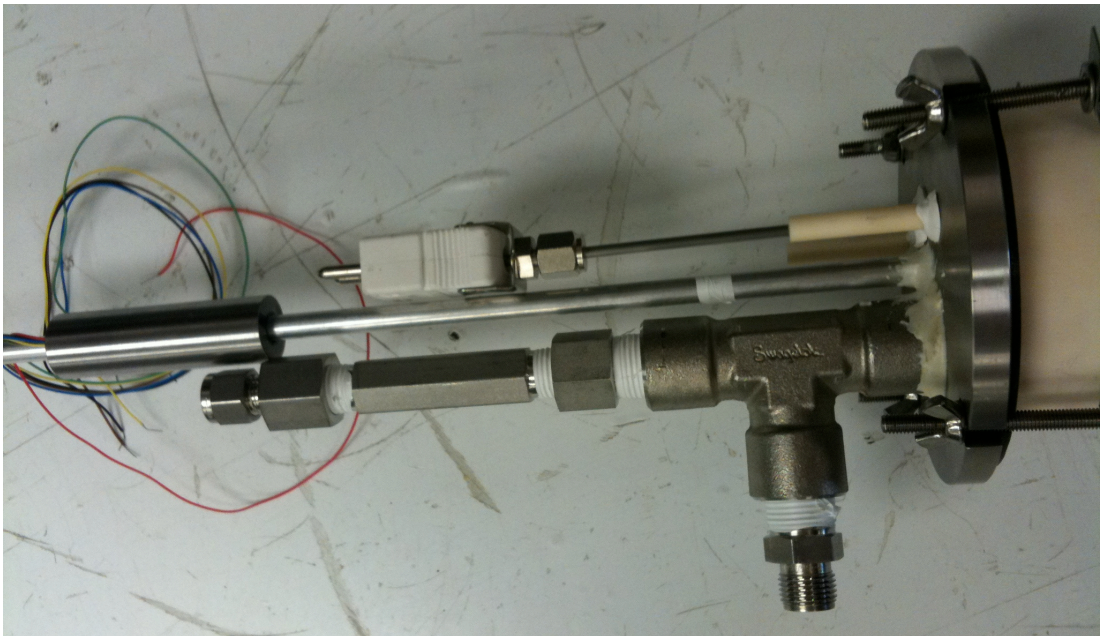


Figure J.19 Close-up of the flange components of the furnace processing vessel.

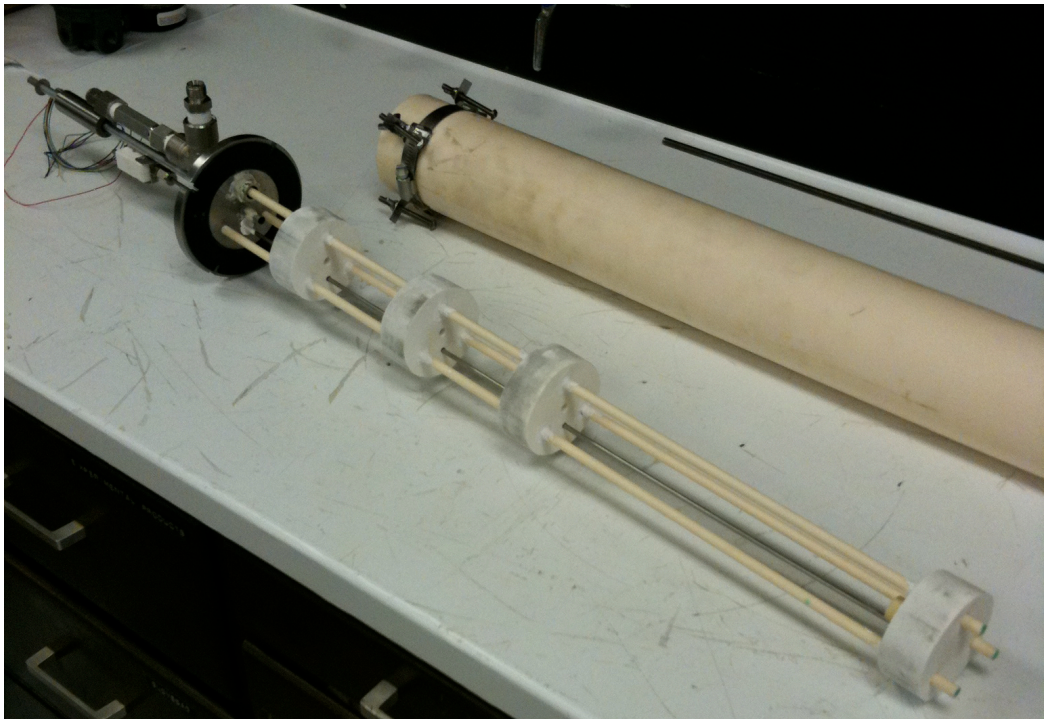


Figure J.20 Overview of the furnace processing vessel and ceramic containment.

The processing vessel incorporates the LVDT to measure the change in axial height of the sample pellet. The change in axial height may be used to extrapolate the density during sintering as well as provide the experimenter with real time feedback on the sintering process. The LVDT and the LVDT containment column are shown in Figure J.21 next to the gas inlet and thermocouple. The transducer is outside of the containment vessel. The magnetic LVDT core is inside the aluminum tube resting on top of a long ceramic tube that sits on the sample.

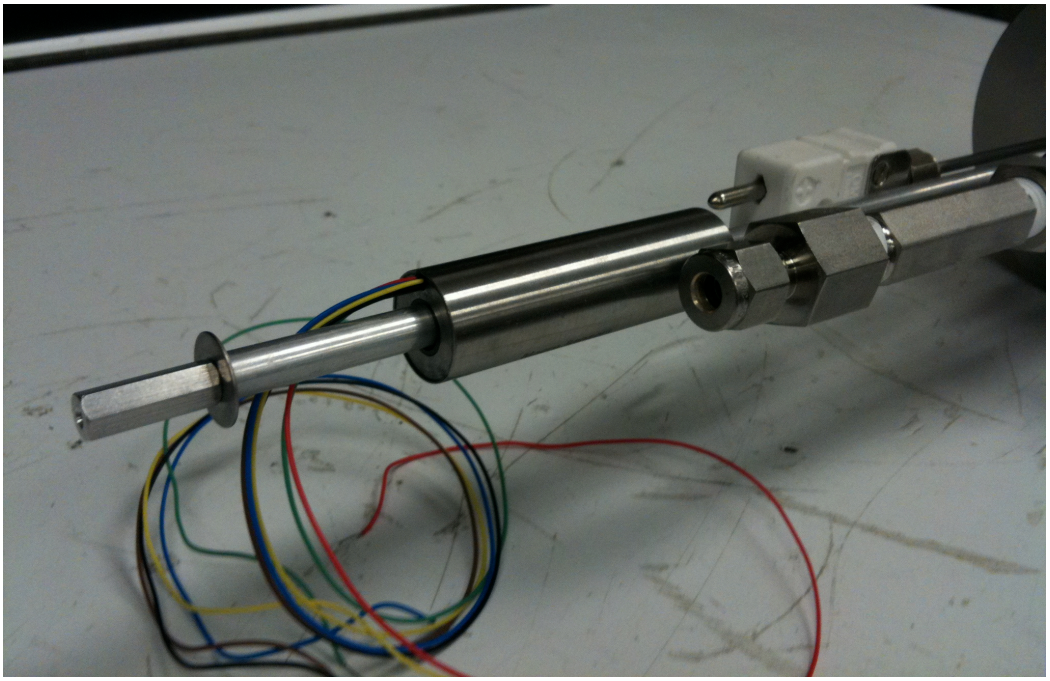


Figure J.21 Close-up of the LVDT and its containment column.

The thermocouple used is a type C molybdenum sheathed custom thermocouple that is rated to 2300^{o}C (Figure J.22). It is connected directly to the Eurotherm control unit.

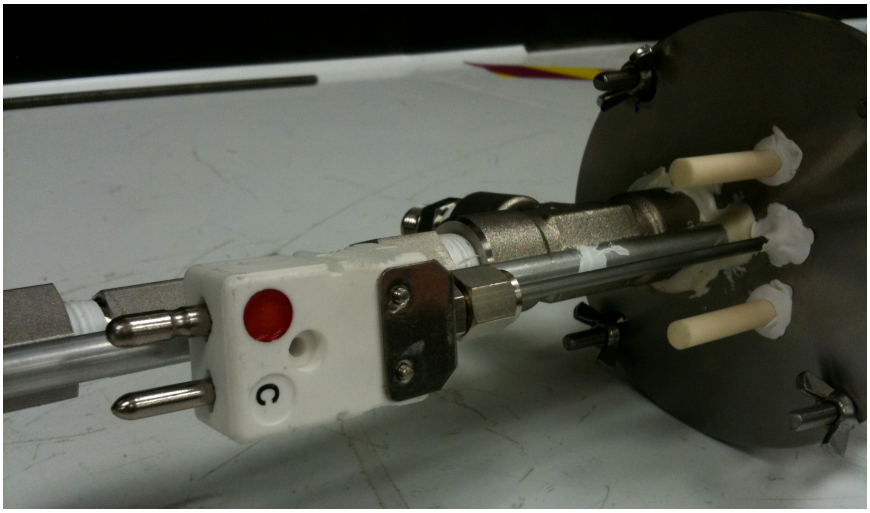


Figure J.22 Type C molybdenum thermocouple as attached to the processing vessel.

The gas inlet is at the top of the metal flange, while the outlet comes out the side from the T-joint (Figure J.23a). The gas goes from the inlet to the bottom of the processing vessel through the hollow ceramic support rod that is glued into the pipe fittings using a ceramic alumina-based glue. The T-joint pipe fitting is large enough for the alumina support rod to fit through with excess room around it (Figure J.23b), this allows for gas to exit the processing vessel through the T-joint, but not up through the top part of the T-joint.

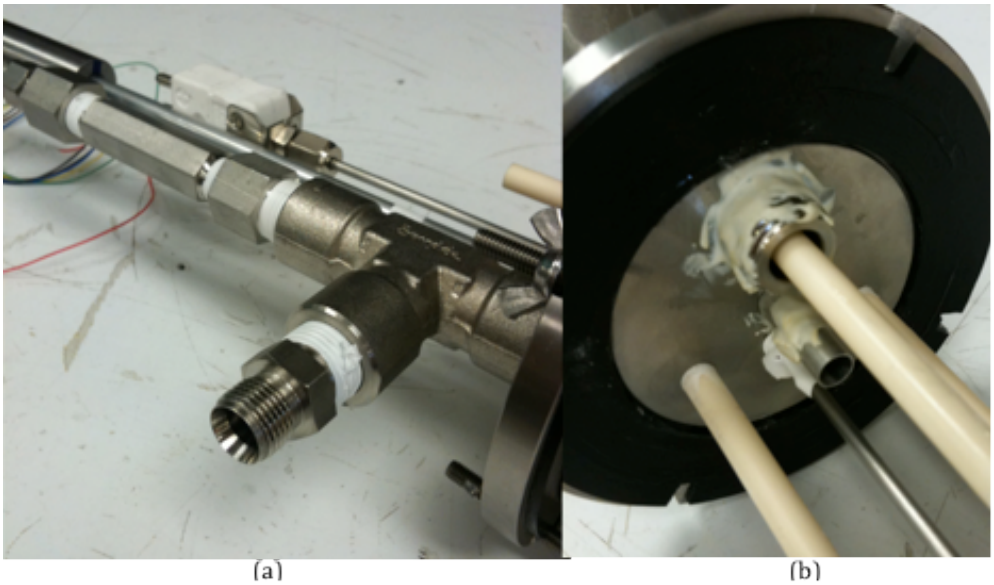


Figure J.23 Gas inlet/outlet close-up (a) and gas outlet from inside the containment (b).

The sample crucible (Figure J.24) was made of yttrium oxide to deter ceramic-ceramic interactions at high temperature. The crucible is not glued into place in the sample holder just in case it needs to be cleaned and to provide easy access to the sample pellet both before and after the sintering process.

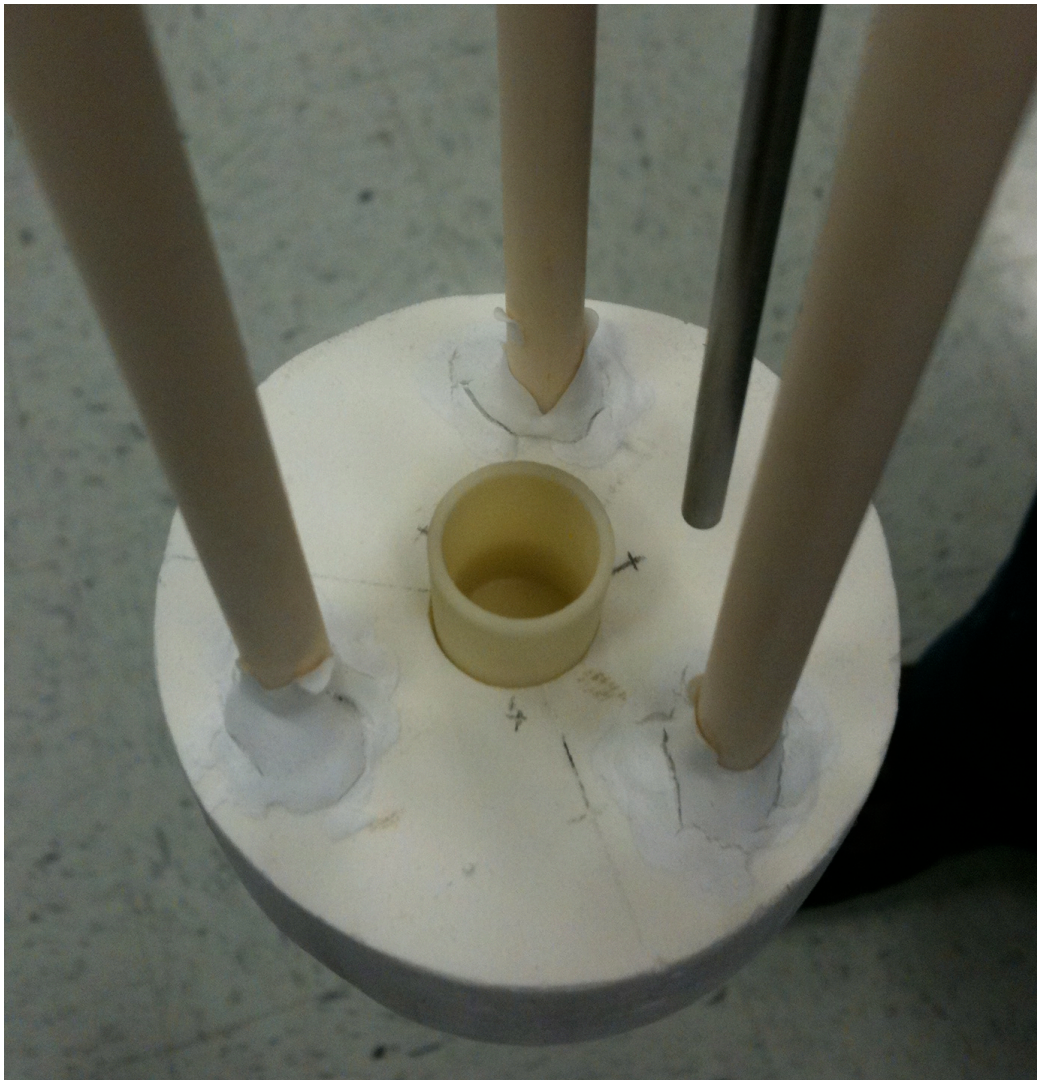


Figure J.24 Yttrium oxide crucible and end of the thermocouple at the bottom of the processing vessel.

The materials for the ceramic processing vessel used are as follows: steel for the pipe fittings and the flange, full-fired alumina for the vessel and rods, a alumina-silica composite for the heat shields (called SALI-2 from Zircar Ceramics), and yttrium oxide for the crucible.

J.2 Results

J.2.1 Alumina Pellet Pressing Tests

The first pressing experiment used only alumina powder (no binding agent), but the punch and die system failed as the longer punch bent after reaching approximately 22,250 Newtons (5000 lbs).

The next five pressing experiments used the second punch and die system, and the sixth experiment used zinc stearate as a binding agent.

The second through sixth experiments used the second punch and die system, and the sixth experiment was the only one that used zinc stearate as a binding agent. The second experiment reached the full 44,500 Newtons (10,000 lb), but it was only left at that pressure for a few seconds. The low time at full pressure and the presence of no binding agent resulted in the pellet falling apart as it was removed from the die.

In the third experiment, water was used as a surrogate binding agent (a few drops), and the system was kept at full pressure for about 30 seconds. This resulted in a successful pellet that although fragile, did stay together after evacuating it from the die.

In the fourth experiment, the water component was removed to determine how important the presence of a binding agent is. The pellet fell apart again as in experiment #2.

The fifth experiment was identical to experiment #3 both in result and initial conditions.

The sixth experiment was the first experiment to use zinc stearate as a binding agent (instead of water) at 0.3wt%. Unfortunately, the punch and die system failed with the shorter punch digging into the un-heat-treated die at a slight angle. This failure occurred at approximately 31,100 Newtons (7,000 lb), and still resulted in a fairly intact pellet all things considered.

The seventh experiment was the first to use the final and successful punch and die system and also used zinc stearate as a binding agent at 0.3wt%. The seventh and eighth

experiment both were kept at 48,500 Newtons (11,000 lb) of force for approximately 60 seconds, and both successfully made identical pellets.

APPENDIX K

First two pellet pressing systems:

The first punch and die system had several design issues: it was made of an unknown metal, was not heat treated, the larger punch was too long, and the heads of both the punches were too small. All of these issues contributed to an unstable system when put under high pressures, which resulted in the longer punch bending while under load as shown in Figure K.1.

The reason the metal was unknown was because the punch and die were made from left over metal (unlabeled) in the lab. This 'convenience' turned out to be a major oversight in hindsight. Secondly, the metal was not heat treated, and this problem is highly related to the first problem. Because the metal was unknown, the heat treatment scheme was unknown. The third issue was that the larger punch was too long (about 0.5" longer then the die), which put more torque on the system while under pressure. The fourth issue was that the heads of the punches were too small. This is because left over material from another project was used so there was no choice in the size.

Figure K.1 Preliminary punch and die system with the longer punch bent.

This system failed in the first experiment after reaching approximately 5000 lbs of force on the hydraulic press. Out of the four issues mentioned, the heat treatment and the size of the punch heads are the major contributors to failure.

The second punch and die system used A-2 tool steel and heat-treated the punches but not the die. The intent behind not heat-treating the die was that if a catastrophic accident were to occur, then the die would 'give' because it was softer, and the system would fail safely. Although the punch failing safely is a good design philosophy, it presents the greater challenge of maintaining the die's physical integrity on a repeatable basis. There were two problems with this second system: the die was not heat-treated, and the smaller punch was too short. Both of these issues put a lot of stress on the die because of the potential for angular movement at the bottom punch as shown in Figure K.2.

This system failed on its fifth experiment. It got up to pressure (approximately 11,000 lb) and it began to tilt to one side. The experiment was ended immediately and it was shown that the bottom punch had dug into to die at an angle as shown in Figure K.3. Figure K.2 is a schematic emphasizing the potential for angular movement in a punch

and die system with a short punch. This issue is remedied in the final design by elongating the smaller punch, reducing the tolerance between the punch and die, and creating chamfers at the lip of the die; all of which serve to reduce the stress on the die.

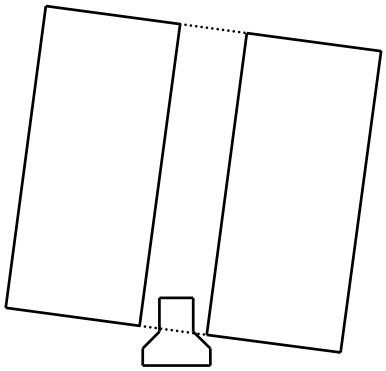


Figure K.2 Example schematic of how a short punch with tolerance can angularly move in a die.

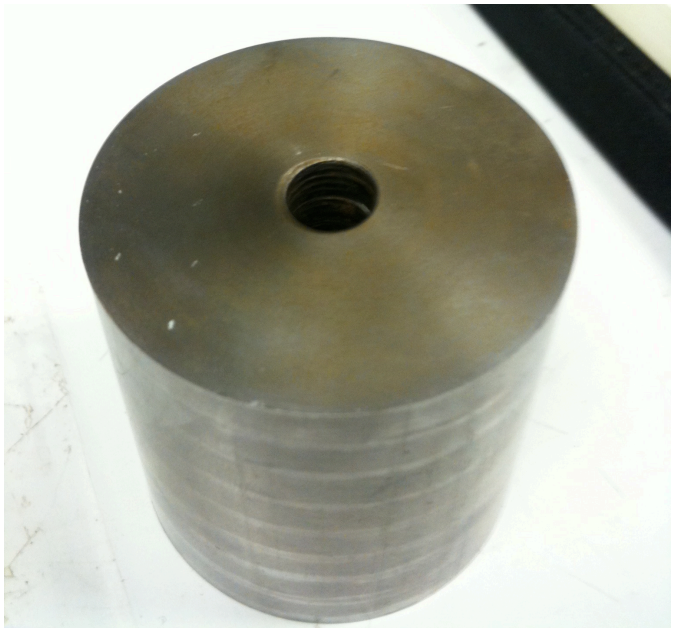
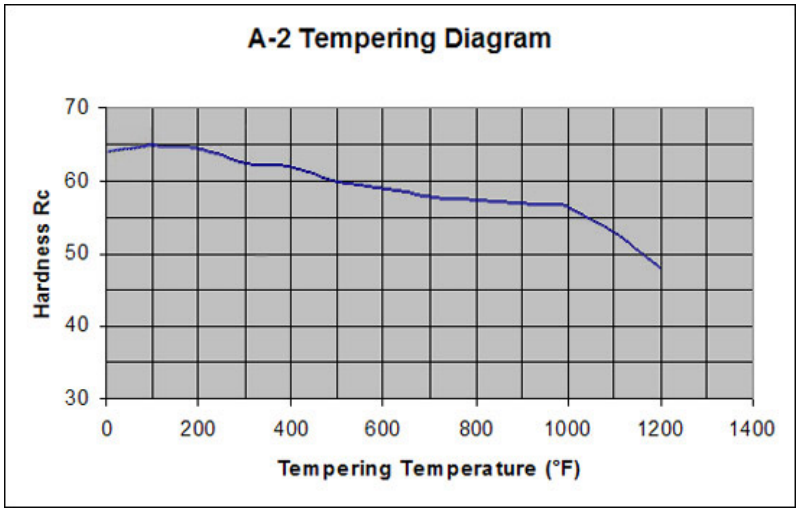


Figure K.3 Picture of the damage done to the lip of the die from the angular movement of the punch.

APPENDIX L

Heat treatment scheme for A-2 tool steel.

TREATMENT	TEMPERATURE RANGE	COOLING/QUENCHING	NOTES
FORGING	2000-2100° F	Cool Slowly	Heat slowly and uniformly. Cool in vermiculite or in other insulating media. Anneal after forging.
ANNEALING	1545-1580° F	Cool slowly at a rate of 25° F per hour to 1110° F.	Protect against surface decarburization using controlled atmosphere or by pack-annealing.
STRESS RELIEVING	1020-1200° F	Cool slowly in furnace to 950° F. Then in air.	Stress relieve after rough machining
PREHEATING	1450-1500° F		Preheat time in furnace is ¾ Hr. per inch of thickness. Heating up to temperature as slow as possible.
HARDENING	1700-1800° F	In air.	Temper immediately after hardening. Allow to cool to 125-150° F before tempering. Soak time is 15 minutes per inch of thickness. Minimum of 30 minutes.
TEMPERING	(See Chart)		Complete equalization of temperature throughout the tools is essential for good results. Double temper. Recommended temperature is 400-1000° F.



VITA

Name:	Michael James Naramore
Address:	Department of Nuclear Engineering TAMU NUEN 3133 TAMU College Station, TX 77843
Email Address:	naramore@gmail.com
Education:	B.A., Nuclear Engineering, Texas A&M University at College Station, 2008 M.S., Nuclear Engineering, Texas A&M University at College Station, 2010