MCODE-3: TIME-DEPENDENT DEPLETION ISOTOPICS WITH MCNP-5 AND SCALE-6.1

ARCHIVES
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
JUL 25 2012
LIBRARIES

by

THOMAS P. GERRITY III

Submitted to the Department of Nuclear Science and Engineering in
Partial Fulfillment of the Requirements for the Degree of

Bachelor of Science in Nuclear Science and Engineering
and
Master of Science in Nuclear Science and Engineering
at the
Massachusetts Institute of Technology

June 2012

© 2012 Massachusetts Institute of Technology
All rights reserved

Signature of Author: __
Department of Nuclear Science and Engineering
May 18, 2012

Certified by: _____
Benoit Forget, Ph.D.
Assistant Professor of Nuclear Science and Engineering
Thesis Supervisor

Thomas H. Newton, Jr., Ph.D.
Interim Director of Reactor Operations, Nuclear Reactor Laboratory
Thesis Reader

Accepted by: ________________________
Mujid S. Kazimi
TEPCO Professor of Nuclear Engineering
Chair, Department Committee on Graduate Students

MCODE-3: TIME-DEPENDENT DEPLETION ISOTOPICS WITH MCNP-5 AND SCALE-6.1

by

THOMAS P. GERRITY III

Submitted to the Department of Nuclear Science and Engineering on
June, 2012 in Partial Fulfillment of the Requirements for the Degree of
Bachelor of Science in Nuclear Science and Engineering and
Master of Science in Nuclear Science and Engineering

ABSTRACT

In order to operate a reactor safely and efficiently, computer simulations must be used to predict certain nuclear characteristics of the reactor. To determine how materials change in a fission power environment, a time-dependent depletion isotopic code must be used. Over the past several decades, the MIT Reactor (MITR) has taken many steps to prepare for its conversion from the use of highly enriched uranium (HEU) to low enriched uranium (LEU) in its fuel. Throughout this process, detailed neutronics simulations must be run to predict the characteristics of the reactor with its current HEU fuel, with potential forms of LEU fuel, and with combinations of the two.

MCODE Version 3 is a linkage code that performs time-dependent burnup calculations by combining the Monte Carlo N-Particle transport code, MCNP, with the Oak Ridge Isotope Generation point depletion code, ORIGEN-S. MCNP provides reaction rates and neutron flux in user-specified irradiation material regions. COUPLE, a data-editing code included in the SCALE-6.1 software package, uses these data from MCNP to update the cross section libraries, which ORIGEN then uses to perform nuclide depletion calculations in each irradiation zone. The MCNP model is then updated with the depleted material compositions, and the exchange is repeated. The MCNP/ORIGEN coupling utilizes an optional predictor-corrector capability.

As a newer version of MCODE Version 2.2, MCODE-3 offers three major changes from its predecessor. The first is the incorporation of ORIGEN-S. MCODE-2 used a previous version of ORIGEN, which is no longer supported by ORNL. ORIGEN-S provides newer nuclear data as well as additional functionality and usability. Secondly, MCODE-3 uses COUPLE to create an entirely unique cross section library from the regionally averaged 238-group flux, which means every cross section value that MCODE-3 uses in its depletion is specific to the input model. MCODE-2 only updates a fraction of nuclides' cross sections, the rest default to a pre-compiled library. Finally, while MCODE-2.2 was written in ANSI C, MCODE-3's main function has been rewritten in the Python scripting language. MCODE's `preproc`, `mcodeout`, and `mcnpxs` programs have not been edited, and are thus still written in ANSI C. Benchmarking has indicated that while the evolution of most nuclides is similar to an MCODE-2 calculation, over many depletion steps some nuclides can diverge due to COUPLE's use of the 238-group flux.

Thesis Supervisor: Benoit Forget, Ph.D.
Title: Assistant Professor of Nuclear Science and Engineering

Thesis Reader: Thomas H. Newton, Jr., Ph.D.
Title: Interim Director of Reactor Operations, Nuclear Reactor Laboratory

ACKNOWLEDGEMENTS

This research has been sponsored by the Global Threat Reduction Initiative (GTRI) within the National Nuclear Security Administration.

This thesis work would not have been possible without the guiding support of Ben Forget and Tom Newton. For the patience they showed in the face of my constant emails and endless questions, for the advice they offered when I got off track, and for the encouragement they gave me when I doubted myself, I am sincerely grateful. I would also like to acknowledge Nick Horelik and Paul Romano, who have always made the time to help me when I needed it.

I must also thank my mom and dad, without whose love and support I would not be who I am today. They have given me the freedom to discover my interests and the lessons and experiences to know how to pursue them, and have always been there for me.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 MCODE Background

People first thought of combining MCNP and ORIGEN in the 1980s. MCNP [3] is a commonly used Monte Carlo transport code that can perform generalized-geometry, continuous-energy calculations for neutrons, electrons, photons, or any neutron/electron/photon combination. It was developed by Los Alamos National Laboratory (LANL) and has since been updated to its current version, MCNP-6. However, MCNP only provides data for a single time-frame of a physical environment; it does not account for the time-dependent nature of nuclide concentrations in a fission environment, which change with radioactive decay or burnup. ORIGEN, a time-dependent depletion code, provides a solution to this issue.

ORIGEN [4, 5] is a point depletion code that provides time-dependent buildup, decay, and depletion calculations for radioactive materials. It solves large systems of coupled, first order ordinary differential equations with constant coefficients using the matrix exponential method. ORIGEN uses ENDF/B data to determine multi-group cross-section libraries, which can be updated manually, or, with SCALE-6.1 [4], can be manipulated through additional codes, such as ORNL's COUPLE code.

With MCNP, a continuous energy transport code, and ORIGEN, a point depletion code, time-dependent depletion isotopics can be calculated for any generalized, user-modeled geometry. However, the interdependency of the calculations of these two codes demands that a steady-state approximation be made. The validity of this approximation stems from the difference in the time scale of transport data and depletion data. While depletion and decay occur significantly on the order of days, the short mean free path of a neutron in a fissile environment such as a reactor is very short, with the average neutron lifetime on the order of 10^{-4} seconds. Thus, after MCNP makes a snap shot transport calculation, the material composition can be approximated as steady state for a number of days before another MCNP run need to be made. This allows for an iterative coupling of MCNP and ORIGEN, which can calculate time-dependent depletion isotopics over significant timescales.

Originally, the codes were coupled manually; a user would run MCNP, use the output file to type up the appropriate ORIGEN input, use the ORIGEN output to rewrite the MCNP nuclides, and so on. However, this approach was tedious, and became prohibitively time consuming for more than a small number of irradiation nuclides. Furthermore, the propensity for human error as well as the excessive time requirement makes this approach unacceptable for the routine burnup and safety analysis calculations necessary in licensed reactors. Therefore, the field saw a growing demand for an automated method of coupling continuous energy transport calculations with point depletion and decay calculations.

1.2 MCODE Alternatives

Once this demand was realized, several such coupling codes were developed. The first was the MCNP-ORIGEN Coupling Utility Program, MOCUP [6], which was released in 1997 by Idaho National Engineering Laboratory (now Idaho National Lab) as a program that automatically linked MCNP-4 and ORIGEN-2.1. MOCUP provided executables that would search the MCNP output tally files for the necessary cell, reaction rate, and flux data, use this data to write and run corresponding ORIGEN input files, and then scan the ORIGEN output punch files for nuclide concentrations to update the MCNP model. However, while MOCUP automated the neutron transport and depletion coupling to some degree, it required each step to be called manually, limiting its capability as a fully automatic depletion isotopic code.

In 1999, LANL released a similar MCNP/ORIGEN linkage code, Monteburns [7]. Unlike MOCUP, Monteburns only requires an input file with an MCNP input and some user-specified Monteburns options in order to automatically iterate between MCNP and ORIGEN. Thus, users only needed to be familiar with MCNP, which is more accessible and commonly used than ORIGEN. As will be discussed further in section 2.4, both MOCUP and Monteburns did not have predictor-corrector depletion capabilities.

More contemporary codes include VESTA [8], ALEPH [9], and BGCORE [10]. VESTA 2.0.3 is being developed in France and boasts extensive flexibility in its capacity to tailor to the user's needs. Similar to MOCUP and Monteburns, it couples a Monte Carlo transport module and a depletion module, currently MCNP or MCNPX and ORIGEN-2.2. However, it is designed to interface with different transport and depletion codes, with the ultimate goal being for it to be compatible with any transport or depletion code. In keeping with its objective of flexibility, it offers the common predictor-corrector depletion approach as well as a predictor only and a midpoint approach, which can be customized by the user. VESTA uses standard ENDF formatted data by default, but these libraries can be updated when new data is released, and users can customize specific nuclide data as desired. Like MCODE, it offers constant flux and constant power irradiation options, using the Monte Carlo generated flux spectrum to calculate absolute flux values for the former option and uses the total recoverable energy from fission to determine the power distribution in the latter case.

ALEPH is a similar wrapper code that couples a Monte Carlo transport module and a point depletion module. These codes are typically limited by the accuracy of using the matrix exponential method to solve first-order linear differential equations with the limited data provided, as ORIGEN-2.2 does. However, the new version of ALEPH, Version 2, incorporates its own depletion module, which employs the backwards differential formulas and implicit Runge-Kutta methods. With customizable tolerances, these methods can provide considerably more accurate depletion results than ORIGEN calculations. However, this accuracy comes at a serious computing cost for large systems of ODEs. While ORIGEN-2.2 has displayed a weak dependence on the system size, the computation time of both the backwards differential formulas and the implicit Runge-Kutta methods grows quickly with the number of differential equations. While computational demand is comparable for under 500 differential equations, by 1000 equations these alternative methods take ~10 times the run time as ORIGEN-2.2. These approaches also offer the inclusion of photon-induced reactions, so may be more fitting in such an environment. Furthermore the ability to adjust materials, temperature, density, and fuel assembly permutation make this a strong fuel management code.

BGCORE is another Monte Carlo transport and point depletion linkage code, which is being developed at Israel's Ben-Gurion University of the Negev in response to its growing interest in fast reactors and the greatly increased computing power seen over recent years. BGCORE primarily differs from the standard MCNP – ORIGEN coupling in two ways. First, it uses a Monte Carlo transport calculation to provide multi-group flux data, but calculates the reaction rates outside of MCNP, which can reduce computing time. Secondly, it incorporates its own depletion module, SARAF, which compares well with ORIGEN-2.2. Currently, BGCORE does not use a predictor-corrector algorithm, but is planning to adopt one in future versions.

1.3 Motivation for Version 3

MCODE 2.2 was written in 2006, coupling MCNP with ORIGEN-2.2. Since ORIGEN-2.2 is no longer supported by ORNL [4], ORIGEN-S has taken its place, boasting increased functionality and usability, as well as the latest nuclear data. In response to this, MCODE must be updated to incorporate ORIGEN-S. ORIGEN-S is part of the SCALE-6.1 software package and comes with options for automatically updating the cross-section data libraries before each ORIGEN run using the COUPLE module. Thus, updating to ORIGEN-S also provides an easy opportunity to incorporate an automated library updating code into MCODE-3, which would be simpler than MCODE-2's approach, which is to update the libraries manually from MCODE's main function.

This transition provides additional opportunities for improving MCODE. MCODE-2 was written in standard ANSI C, which, while computationally efficient as a language, is not easily accessible when minor changes or additions need be made to the code. In addition, MCODE-2's source code is extremely compact and sparsely commented, compounding its complexity and making it time-consuming to parse. This need is addressed in MCODE-3 by converting MCODE to the Python scripting language with liberal, but comprehensive and consistent commenting. Python is intuitive and easy to learn, and by converting MCODE to Python, future updates to the code will be made much easier.

Generally, a program written in Python will be significantly slower than an equivalent program written in C; however, there are several reasons why that does not have a large impact on the overall computing time of an MCODE run. The most immediate reason is that the vast majority of MCODE's runtime is due to MCNP, with the rest being split between the MCODE wrapper code and ORIGEN. Even if the MCODE wrapper code takes more time, it will only be a fraction of the amount of time that the MCNP runs take. Thus the total computing time should not see a significant change. In addition, Python has an extensive library of built-in functions and is very effective with file I/O, which is most of what MCODE does. Utilizing these Python functions will reduce computation time. Furthermore, several functions in MCODE-2 have been identified as being easy to improve, and can be rewritten in MCODE-3 with much better computational efficiency.

2 REVIEW OF MCODE

2.1 Overview

MCODE-3 is largely similar to MCODE-2.2, and this chapter will serve to recapitulate the basic functionality of MCODE. The following chapter will review in detail the differences between MCODE-2.2 and MCODE-3.

The MCODE package consists of four total programs, which are run through the command line. MCODE-3 only made changes to the mcode program; the remaining three are unchanged. While there is a brief description below of all four programs, more information can be found on the unchanged three in Zhiwen Xu 2006 Thesis, "MCODE, Version 2.2 – An MCNP-ORIGEN DEpletion Program". The four programs are:

- preproc – preprocess the MCODE input file, creates the temporary directory in the current directory, and builds the MCNP continuous-energy neutron library with the appropriate reaction types for the corresponding nuclide IDs;
- mcode – the main MCODE function that iterates between MCNP and ORIGEN;
- mcodeout – parses the mcode output files in the temporary directory and collects them into a central, detailed output;
- mcnpxs – prepares a summary table of the continuous-energy MCNP cross section libraries.

Running MCODE only requires the use of preproc, mcode, and, depending on the data needed, mcodeout in sequence. mcnpxs is a purely auxiliary program. Again, more information on the preproc, mcodeout, and mcnpxs programs can be found in Zhiwen's thesis. For the remainder of this paper, MCODE will refer to the main mcode program unless otherwise specified.

The functional flow of MCODE-3 is the same as MCODE-2.2, and can be seen graphically in figure 2.1. Before the depletion loop, MCODE parses through the MCODE input file to initialize all MCODE variables, including but not limited to the materials, their compositions, and the MCODE options. If this MCODE run is starting from the first depletion point, an initial MCNP run is made with the initial materials to determine the beginning-of-timestep reaction rate and flux data before entering the depletion loop. If this MCODE run is starting after the first depletion point, then the initial MCNP run is unnecessary and MCODE skips straight to the depletion loop.

The depletion loop will iterate the following processes over all the depletion points, as specified in the MCODE input file. First, the beginning-of-timestep reaction rate and flux data (equivalent to the end-of-timestep data from the previous step) is extracted and used to update the depletion cross-section libraries and perform the predictor depletion run for every user-specified irradiation material using COUPLE and ORIGEN-S. From the ORIGEN-S output tables, MCODE selects the important nuclides based upon their contributions to total neutron absorption. MCODE then extracts the nuclide concentration data from the ORIGEN-S output for these important nuclides and uses it to write a new MCNP input with updated material compositions, which it runs, giving the predictor end-of-timestep reaction rate and flux data.

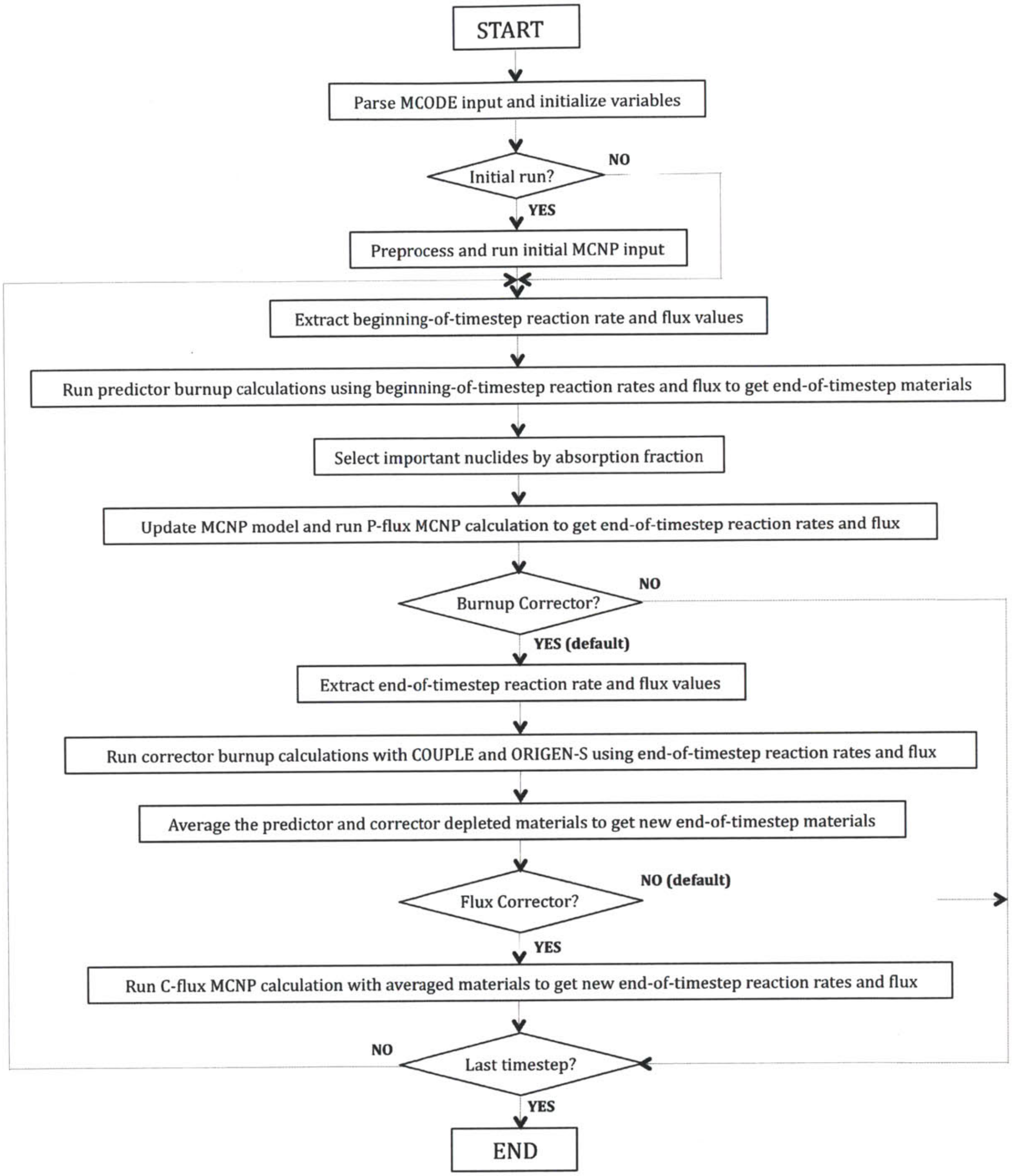

START
Parse MCODE input and initialize variables
Initial run?
NO
YES
Preprocess and run initial MCNP input
Extract beginning-of-timestep reaction rate and flux values
Run predictor burnup calculations using beginning-of-timestep reaction rates and flux to get end-of-timestep materials
Select important nuclides by absorption fraction
Update MCNP model and run P-flux MCNP calculation to get end-of-timestep reaction rates and flux
Burnup Corrector?
NO
YES (default)
Extract end-of-timestep reaction rate and flux values
Run corrector burnup calculations with COUPLE and ORIGEN-S using end-of-timestep reaction rates and flux
Average the predictor and corrector depleted materials to get new end-of-timestep materials
Flux Corrector?
NO (default)
YES
Run C-flux MCNP calculation with averaged materials to get new end-of-timestep reaction rates and flux
NO
Last timestep?
YES
END

Figure 2.1. A flow diagram of the MCODE main function.

At this point, MCODE will return to the beginning of the depletion loop unless this was the final depletion point, or unless one of two corrector options have been requested in the MCODE input file. If the first option, the depletion corrector option, has been requested, another depletion run is performed using beginning-of-timestep material compositions and predictor end-of-timestep reaction rate and flux data. The corrector depletion output material compositions are then averaged with the predictor material compositions. This predictor-corrector burnup model has been found to give extremely accurate results with only a marginal increase in computation time, thus the use of the corrector burnup option is highly recommended. The second option, the corrector flux, or C-Flux, option, calls for an additional MCNP run to be performed using the averaged materials from the predictor and corrector depletion outputs. This gives new end-of-timestep reaction rate and flux data to be used for the next time step. While this approach is more accurate, the flux difference in most cases is insignificant, and it requires a second MCNP run for each timestep, which nearly doubles the total computation time of the MCODE run, but can allow the user to take longer timesteps for a desired level of accuracy. Thus, this option is not requested by default, and, as explained in Appendix A, the user must specify the desire for it in the MCODE input file.

2.2 MCNP Reaction Rates and Flux Tallies

In order to use COUPLE to update the ORIGEN-S cross section libraries, MCNP must be used to produce nuclide specific reaction rates and region-averaged one-group flux. With the update to MCODE-3, 238-group neutron flux is also tallied; this is discussed further in chapter 3. MCNP obtains these data through track length estimates, or F4 tallies, which, using the FMn tally multiplier, MCNP automatically converts to reaction rates, one-group flux, and 238-group flux through the following equations:

$$\begin{aligned} \Phi_i &= \int \phi_i(E)dE \\ R_{ijk} &= \int \sigma_k^j(E)\phi_i(E)dE \\ \Phi_{i,g} &= \int_E^{E'} \phi_i(E) \end{aligned} \tag{2.1}$$

where $\boldsymbol{\Phi}_i$ = the region-averaged one-group flux for material i;
$\phi_i(E)$= the region-averaged energy-dependent flux for material i;
R_{ijk} = the nuclide density specific reaction rate of type k for nuclide j of material i;
$\sigma_j^k(E)$ = the microscopic cross section of reaction type k for nuclide j;
$\Phi_{i,g}$ = the region averaged flux of energy group, g, for material i.

However, COUPLE only accepts cross section values in its input file, so once read from the MCNP output files, the nuclide density specific reaction rates and flux must be converted back to microscopic cross section values by:

$$\sigma_{ijk} = \frac{R_{ijk}}{\Phi_i} \tag{2.2}$$

Because the ratio of tracks to collisions within a given cell is high, track length estimates are generally very accurate. In order to measure the accuracy of MCNP tallies, a factor called relative error is employed, defined as the ratio between the standard deviation and the average value. Thus, a small relative error would indicate that the actual value is statistically close to the calculated average value, and therefore the calculation is accurate. The MCNP manual supplies recommendations on acceptable relative error in table 2.1.

In general, relative error decreases with the number of neutron histories, N, by approximately $\sim 1/\sqrt{N}$. MCODE will not terminate in the case of high relative error; COUPLE and ORIGEN-S will use all cross section data regardless of statistical quality. It is up to the user to decide whether or not the level of error is acceptable or not. If it is not acceptable, the user has the option to include more neutron histories in the kcode card of the MCODE input file's MCNP model.

Table 2.1. Guidelines for interpreting relative error, R.

Range of R	Quality of Tally
0.5 to 1.0	Not meaningful
0.2 to 0.5	Factor of a few
0.1 to 0.2	Questionable
< 0.10	Generally reliable
< 0.05	Generally reliable for point detectors

2.3 Flux Normalization

When performing constant flux depletions, which are required for next-step calculations, MCODE must supply the flux calculated by MCNP to ORIGEN-S. Because MCNP outputs one-group flux normalized to the total power of its environment, MCODE must use a multiplicative factor, the flux multiplicative factor (FMF), to convert the flux units to those that ORIGEN-S will accept. When MCNP performs a standard calculation, it automatically outputs its one-group flux data with units [neutrons per cm^2 per source neutron]. A source neutron is a neutron produced by a fission reaction, or transitively by a fission neutron that could potentially be the source of another fission reaction. However, ORIGEN-S only accepts one-group flux of standard units [neutrons per cm^2 per second]. The FMF, of units [source neutrons per second], is defined as:

$$FMF = \frac{P \bullet \bar{v}}{Q_{ave} \bullet k_{eff}} \tag{2.3}$$

where P = the total system power;
$\bar{\upsilon}$ = the average number of fission neutrons produced per fission;
Q_{ave} = the average recoverable fission energy released per fission;
k_{eff} = the system's criticality eigenvalue representing source neutrons per fission neutron.

While total power, P, is specified in the MCODE input file, and $\bar{\upsilon}$ and k_{eff} are calculated by MCNP, MCODE calculates Q_{ave} itself using the function

```
averageQ(mt, n)
```

This function calculates the average recoverable fission energy per fission by summing the recoverable fission energy release of each nuclide, weighted by the number of atoms of that nuclide, for all materials, according to:

$$Q_{ave} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m_i} Q_j \int \phi_i(E)\sigma_f^{ij}(E)dE \bullet N_{ij} \bullet V_i}{\sum_{i=1}^{n}\sum_{j=1}^{m_i} \int \phi_i(E)\sigma_f^{ij}(E)dE \bullet N_{ij} \bullet V_i} \tag{2.4}$$

where Q_j = the recoverable fission energy release of nuclide j;
$\int \phi_i(E)\sigma_f^j(E)dE$ = the fission reaction rate of nuclide j of material i;
N_{ij} = the number density of nuclide j of material i;
V_i = the volume of material i.

MCODE calculates the individual recoverable fission energy release for each nuclide by the approximation:

$$Q(Z,A) = (1.29927 \bullet 10^{-3}) \bullet Z^2 \bullet \sqrt{A} = 33.12 \qquad (2.5)$$

where $Q(Z,A)$ = the recoverable fission energy release for a nuclide of A and Z;
Z = the atomic number;
A = the atomic mass.

When performing constant power depletions, which are used by default for the predictor and corrector depletion calculations, ORIGEN-S computes the Q value internally to convert the power, specified in the input file, to flux. Instead of using the approximation above, ORIGEN-S uses archived data for the fission and capture energy from a fission reaction for each nuclide to determine the average Q value. The archived values used by ORIGEN-S are compared to the MCODE calculated values for several nuclides in table 2.2.

Table 2.2. The total recoverable energy from fission, Q, as calculated by ORIGEN-S and MCODE-3 for several actinides.

	ORIGEN-S			MCODE-3
	Fission [MeV]	Capture [MeV]	Total, Q [MeV]	Total, Q [MeV]
U-233	191.29	6.841	198.13	200.98
U-235	194.02	6.545	200.57	201.70
U-238	198.12	4.804	202.92	202.77
Pu-239	200.05	6.533	206.58	210.60
Pu-241	202.22	6.301	208.521	211.34

2.4 Coupling Transport and Depletion

Classifying MCODE as a linkage or wrapper code refers to its coupling of MCNP and ORIGEN in order to iteratively perform transport and depletion calculations. As introduced in section 1.2, other methods are and have been used to provide this coupling. The small differences between these methods can have a large impact on burnup results over many time steps.

In order to perform burnup calculations, it is important to understand the time-dependent factors that affect how a given material changes. Through the MCODE input file, a user specifies certain irradiation materials, which correspond to materials from the included MCNP model. Each of these materials may consist of multiple nuclides, which are assumed to be spatially uniform throughout the calculation. MCODE's objective is to provide an array of material vectors, which hold the number densities of every nuclide of each material at predetermined time-steps. For nuclide j of material i, that number density is calculated as:

$$\frac{dX_{ij}}{dt} = \sum_{k=1}^{m_i} l_{k\rightarrow j}\lambda_k X_{ik} + \sum_{k=1}^{m_i} f_{k\rightarrow j} X_{ik} \int \sigma_k(E)\phi_i(E)dE - \lambda_j X_{ij} - X_{ij}\int \sigma_j(E)\phi_i(E)dE \qquad (2.6)$$

where X_{ij} = atom number density of nuclide *j* in material *i*;

m_i = number of nuclides in material *i*;

$l_{k\rightarrow j}$ = fraction of radioactive disintegration by *k* that lead to formation of *j*;

λ_k = radioactive decay constant of nuclide *k*;

$f_{k\rightarrow j}$ = fraction of neutron absorptions by *k* that lead to formation of *j*;

$\phi_i(E)$ = spatial-average neutron energy spectrum in cell *i*;

$\sigma_k(E)$ = neutron absorption cross section of nuclide *k*.

MCODE makes steady state flux approximations during each timestep, which are dictated by the user-specified depletion points as time or burnup markers. Let us denote each timestep as $[t_{l-1}, t_l]$ for each depletion point, t_l. The nuclide concentration vectors for all *n* materials, $\mathbf{X}_i$, are known at the beginning-of-timestep. In order to calculate the end-of-timestep, MCODE must solve equation the nuclide differential equations, which can be written in compact vector form:

$$\frac{dX_i(t)}{dt} = A_i(t)X_i(t), (i = 1,2, \dots, n) \qquad (2.7)$$

where the matrix, **A**, is the transition matrix. As section 2.2 describes, MCNP produces each material's transition matrix from flux spectrum and reaction rate data.

There are several methods available to solve this ODE for $\mathbf{X}_i$ (t_l), the end-of-timestep concentration vectors. MOCUP utilizes perhaps the simplest method: the beginning-of-timestep flux and reaction rate data are used to create the transition matrices for that entire timestep. Thus, the material vectors at the end-of-timestep look like:

$$X_i(t_l) = \left[\exp(A_i(t_{l-1}) \bullet (t_l - t_{l-1}))\right] \bullet X_i(t_{l-1}) \qquad (2.8)$$

This is clearly an inaccurate method, as the flux and reaction rate data change throughout the timestep, so any steps of significant length quickly diverge from the true solution.

Monteburns tries to mitigate this by incorporating a middle-of-timestep approach. Here, the code estimates the middle-of-timestep flux and reaction rate data and uses them to create the transition matrices. This solution looks like:

$$X_i(t_l) = \left[\exp(A_i(\frac{t_{l-1} + t_l}{2}) \bullet (t_l - t_{l-1}))\right] \bullet X_i(t_{l-1}) \qquad (2.9)$$

While flux and reaction rates are not linearly dependent on time, this is a more accurate model than only using the beginning-of-timestep values.

MCODE uses a predictor-corrector method. With this approach, two depletion calculations are performed. The first, the predictor depletion, is calculated with beginning-of-timestep flux and reaction rate data. Then, another MCNP is performed with the predicted end-of-timestep material composition to estimate the end-of-timestep flux and reaction rate data. These are then used to perform the second depletion, the corrector depletion. The final nuclide densities are obtained by averaging the results of the predictor and corrector runs:

$$X_i^P(t_l) = \left[\exp(A_i(t_{l-1}) \bullet (t_l - t_{l-1}))\right] \bullet X_i(t_{l-1}) \tag{2.10}$$

$$X_i^C(t_l) = \left[\exp(A_i^P(t_l) \bullet (t_l - t_{l-1}))\right] \bullet X_i(t_{l-1}) \tag{2.11}$$

$$X_i(t_l) = \frac{X_i^P(t_l) + X_i^C(t_l)}{2} \tag{2.12}$$

The predictor-corrector method has been shown to yield more accurate depletion results [12].

Consider a simple, numerical example of the ordinary differential equation in which the vector variables have been replaced by scalar equivalents and the equation is evaluated over a single timestep:

$$\frac{dx(t)}{dt} = a(x)x(t), \quad t \in [0, 0.1] \tag{2.13}$$

Here, $x(t)$ is the scalar concentration value of a nuclide in units density, and $a(x)$ is the scalar equivalent of the transition matrix $\mathbf{A}(t)$ for the single-nuclide system. The initial density, $x(t = 0)$ is set to 1. This simple ODE can be solved for various example functions, $a(x)$, and the results are shown for the beginning-of-timestep method, the middle-of-timestep method, and the predictor-corrector method in table 2.3.

Table 2.3. Numerical examples of various coupling schemes.

$a(x)$	Exact solution		Beginning-of-time-step method		Middle-of-time-step method		Predictor-corrector method	
	$x(t)$	$x(t{=}0.1)$	$x(t{=}0.1)$	Error	$x(t{=}0.1)$	Error	$x(t{=}0.1)$	Error
C (const)	e^{Ct}	$e^{0.1C}$	$e^{0.1C}$	0	$e^{0.1C}$	0	$e^{0.1C}$	0
$1/x$	$1+t$	1.1	1.1052	0.47%	1.0998	−0.02%	1.0999	−0.01%
$\sqrt{x}$	$\left(\frac{2}{2-t}\right)^2$	1.1080	1.1052	−0.28%	1.1080	0	1.1080	0
x	$\frac{1}{1-t}$	1.1111	1.1052	−0.53%	1.1109	−0.02%	1.1110	−0.01%
x^2	$\frac{1}{\sqrt{1-2t}}$	1.1180	1.1052	−1.3%	1.1169	−0.1%	1.1175	−0.04%
x^5	$(1-5t)^{-\frac{1}{5}}$	1.1487	1.1052	−3.8%	1.1370	−1.0%	1.1422	−0.7%

The predictor-corrector method consistently generates values closest to the exact solution. As such, most modern time-dependent depletion isotopic computer codes use this method because of its proven accuracy. In addition, while this method requires twice as many depletion calculations, this has a negligible effect on the total run time due to the small computational cost of ORIGEN compared to MCNP. It should be noted that, as seen in the results, the other two methods can behave accurately for certain transition matrix behavior, but when $a(x)$ has a large change over the time step, these methods quickly lose accuracy and require smaller timesteps. The predictor-corrector method assures reasonable accuracy regardless.

2.5 Selecting Nuclides and their Reaction Types

In order for MCNP to calculate each nuclide's reaction rates, it must be told for which reaction type library to look. COUPLE and ORIGEN classify three groups of nuclides:

- Light Elements – light elements, cladding, construction and structural materials in a reactor, and al of their activation products;
- Actinides – heavy metal nuclides with $Z \geq 90$, including fissile isotopes and other actinides, and their decay daughters; and
- Fission Products – all nuclide products of fission, including their decay and activation products.

MCODE is concerned with the union of reaction types applicable to these three nuclide types shown in table 2.4. Actinides can contribute capture, fission, $(n,2n)$, and $(n,3n)$ reactions, and light elements and fission products can contribute capture, $(n,2n)$, (n,α), and (n,p) reactions.

In MCODE, all data regarding irradiation nuclides of a given material are stored in a vector, `nuclxs`. Reaction types are assigned to each nuclide in each material's `nuclxs` vector when it is initialized with

```
initnuclxs(mt, n, tracklist)
```

The above function opens the ORIGEN cross section library assigned to that material by the MCODE input and reads which of the six reaction types each nuclide can undergo. Then, when MCODE writes the MCNP input file, it specifies which reaction type libraries to read by writing the appropriate MCNP reaction numbers, as shown in table 2.4, in the FMn tally multiplier for each nuclide.

Table 2.4. MCNP reaction types and corresponding reaction numbers for FMn tally multiplier.

Reaction Type	MCNP Reaction Number
Capture (n,γ)	102
Fission (n,f)	-6
$(n,2n)$	16
$(n,3n)$	17
(n,α)	107
(n,p)	103

Due to the extent of depletion possibilities, ORIGEN outputs a large number of nuclides. Tallying all of these nuclides in MCNP would be computationally intense, so certain nuclide data is not stored in the `nuclxs` vector after each depletion. MCODE classifies which nuclides are important by comparing their fractional neutron absorption. If a nuclide does not contribute neutron absorption within a user-specified threshold, then it will have very little effect on neutron transport compared to a nuclide whose absorption fraction is within the threshold. Therefore, only those nuclides with higher absorptions are tracked. The threshold is declared for each material in the MCODE input file's material specification entries as an importance value. MCODE then reads the absorption fractions of the depleted nuclides, sorts them into descending order, and sums them until the threshold is reached. Therefore, for the default importance value of 0.999, only nuclides that account for the greatest fractions of the 99.9% of the material's total neutron absorption will be tracked. The sensitivity of this value will be discussed in Chapter 5. This selection is performed by the function

```
selectimpnuclides(P, dt, mt, n)
```

The `selectimpnuclides` function is called after the predictor depletion calculation. It is called to select nuclides from both the P (predictor) and N (next-step) depletions. The next-step depletion is a complete set of ORIGEN depletions, which use material and flux parameters from the current depletion point since next step values are not available yet, but next-step depletion point parameters such as relative power, depletion duration in days or burnup, and micro time steps. Without the material composition for the next depletion point, MCODE cannot calculate each material's power fraction, so next-step calculations are always constant flux depletions.

The purpose of the next-step depletion is to give MCODE an idea of which nuclides may be important after the current depletion point. The nuclide concentration data from the next-step depletion output tables is not used; MCODE is only concerned with the absorption information. Without this feature, MCODE could potentially miss flux data of nuclides that, will contribute significant neutron absorption during the next depletion. If a nuclide will be important in future timesteps, MCODE wants to calculate reaction rate data ahead of time even if the nuclide is not important now, so that the final concentration value will be more accurate.

The process of selecting and updating important nuclide concentrations involves reinitializing each materials `nuclxs` array using the `initnuclxs` function. After each depletion points P and N depletions, `initnuclxs` is called, which resets all of the nuclide concentration data to their initial values as described in the MCODE input file. This allows MCODE to remove nuclides that may not be important anymore. After reinitializing, MCODE calls `selectimpnuclides` for both P and N depletions, which adds the important nuclides into the `nuclxs` array with a zero concentration value. Then, `readorgout` function is called to read the condensed output file for every irradiation material's predictor ORIGEN run and to update the nuclide concentrations to their end-of-depletion values. These processes are described in more detail in Appendix B.

3 Updates in Version 3

3.1 ORIGEN-S

In MCODE-3, ORIGEN-S has replaced ORIGEN-2.2 as the burnup depletion and decay code. ORIGEN provides time-dependent concentration data through generation and depletion contributed by transmutation, fission, and radioactive decay reactions. It uses the matrix exponential method to calculate these data. While ORIGEN-S boasts additional features over its predecessor, MCODE-3 uses the same capabilities as MCODE-2 used in ORIGEN-2.2. ORIGEN-S, however, has the latest nuclear data, and is supported by ORNL, while ORIGEN-2.2 is not.

While externally very similar, MCODE-3 requires large changes to use ORIGEN-S in place of ORIGEN-2.2. This is primarily due to the input file formatting, as well as the I/O file structure. ORIGEN-S is the decay and depletion module of SCALE-6, and thus follows the format of other SCALE inputs. This means that the entire SCALE-6 package must be installed for use of ORIGEN-S, and thus for use of MCODE-3, but it allows MCODE-3 to take advantage of other SCALE features, such as COUPLE. The incorporation of COUPLE is discussed further in section 3.2.

The input file formatting is similarly compact in the newer version of ORIGEN, but it adopts several important concepts that are important to understand. ORIGEN-S discontinues the usage of the three-letter data card structure. Instead, numbered data arrays are used to specify the characteristics of each run. These arrays are organized into six data blocks, which are read sequentially, and they all follow the FIDO input system, which is described in section M10 of the SCALE manual. An array is identified by its number, and by two "$" signs or asterisks, indicating that the array entries will be integers or floating point values, respectively. Similar to ORIGEN-2.2, each array has a specific number of data entries. These entries are described in detail in section F07 of the SCALE manual.

The other notable difference between the use of ORIGEN-2.2 and ORIGEN-S is the input and output file structure. ORIGEN-2.2 used outdated "TAPE" files, which held distributed input and output information regarding the original materials, the depletion options, the depleted materials, and the output tables. ORIGEN-2.2 also used "PCH" files to store output material compositions and to input materials into future depletions. These "PCH" files made continuing depletion after an MCNP calculation easy, because ORIGEN could accept the material compositions in the exact same format as it outputted them in the previous iteration. However, as ORIGEN-S reads material compositions from the actual input file, these "PCH" files are not necessary for depletion input. While ORIGEN no longer uses them, MCODE-3 still needs a way to keep track of all the nuclides that do not meet the importance threshold. Thus, while the central ORIGEN output file condenses every material's neutron-important nuclides into a single location, a "PCH" file is kept for every material at every timestep that stores nuclide ID, type, and concentration in grams for every nuclide.

While by default ORIGEN-S, and COUPLE, remain invisible to the user, MCODE-3 includes a new option to keep all of the SCALE input and output files and store them in the temporary directory. Unless the user has a specific need for a single material's input files, this option is not recommended. The SCALE input format is more complex than the commonly used MCNP system, and, with a large number of irradiation materials or depletion points in the MCODE run, the storage of these files can take significant drive space. This option is explained further in Appendix A.

3.2 COUPLE

ORIGEN-S and COUPLE are both modules in the SCALE-6.1 software package from ORNL, which makes for the natural incorporation of COUPLE into MCODE-3. COUPLE is designed to automatically update libraries used by ORIGEN-S from cross-section and flux data, as seen in figure 3.1. It effectively removes a complex function of MCODE-2, allowing MCODE to simply provide COUPLE with reaction rate and flux data from the MCNP calculation, and uses COUPLE to actually update cross-section libraries for ORIGEN-S. Without editing these libraries, ORIGEN-S would be forced to use data from a standard library as ORIGEN-2.2 does, which may not be indicative of the exact flux spectrum in the target environment. Because MCNP is able to calculate the exact spatially-dependent reaction rate data and multi-group flux, MCODE does not have to compromise for a less accurate data library.

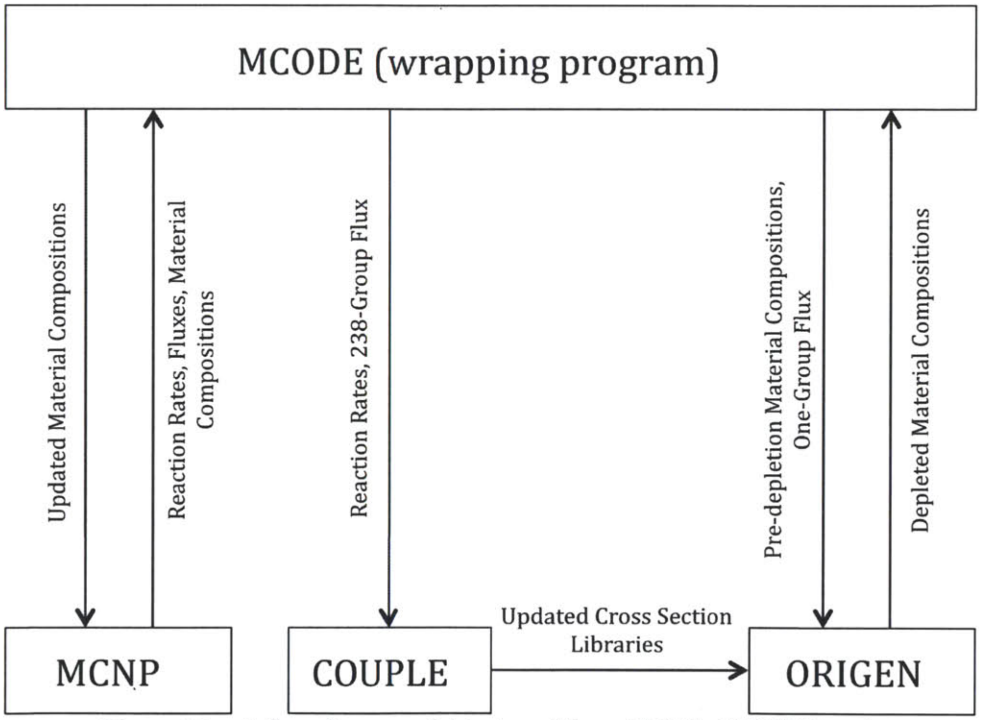

MCODE (wrapping program)
Updated Material Compositions
Reaction Rates, Fluxes, Material Compositions
Reaction Rates, 238-Group Flux
Pre-depletion Material Compositions, One-Group Flux
Depleted Material Compositions
Updated Cross Section Libraries
MCNP
COUPLE
ORIGEN

Figure 3.1. A flow diagram of data to and from MCNP, COUPLE, and ORIGEN within MCODE-3.

Beyond reducing the complexity of MCODE-3, COUPLE also provides an additional feature that bolsters the accuracy of the depletion calculation. In Version 2.2, MCNP would supply one-group reaction rate and flux data for nuclides that were selected as important at beginning-of-timestep, based on their neutron absorption fraction. This practice reduces error considerably, and is continued in MCODE-3. However, while this supplies exact cross-section values to ORIGEN for the important nuclides at beginning-of-timestep, the other beginning-of-timestep nuclides, as well as those that are newly generated during depletion, are forced to use values from the standard cross-section libraries that are not informed by the nuclear characteristics of the specified environment. Thus, MCODE-3 performs an additional MCNP tally, which measures the 238-group flux averaged over all the irradiation materials. COUPLE then uses this flux to update the cross-section libraries for the remaining nuclides, those with neutron absorption below the user-specified threshold at beginning-of-timestep and those that are formed during depletion. While the results on the most important nuclides are often barely affected, this method can have large impact on nuclides with small concentrations, as illustrated in Chapter 4.

This approach supplies depletion data that is a better representation of the specific environment of the MCNP model. However, it could be improved further. As detailed in section 6.2, this approach can lose accuracy in a model with irradiation materials that experience very different 238-group flux spectra. Because MCODE-3 currently averages the 238-group flux over all irradiation materials, data is lost when the spectra vary significantly. Supplying material-specific 238-group flux spectra would be the most accurate approach, but would add considerable runtime to MCNP.

3.3 Python

From a development standpoint, the biggest change between Versions 2 and 3 of MCODE is the rewriting of the entire `main` function and function library into the Python scripting language, using Python Version 2.7.3. The other three programs included with MCODE, `preproc`, `mcodeout`, and `mcnpxs`, have not been changed from ANSI C.

This change offers several advantages over the original language, ANSI C. The driving motivation for the change was the need for the MCODE source code to be more accessible in anticipation of future updates. While executables compiled from C are often heralded as being faster, the language itself can be difficult to read and understand, especially compared to a more user-friendly language such as Python. Just as the need arose to update MCODE to Version 3, there will be a future need to update MCODE again. Whether this means simply adding an option or small edits, to file formatting or a large overhaul of the main linkage program, it will be easier for future work to be done in the Python scripting language than in ANSI C. Furthermore, MIT students have historically been those who have rewritten MCODE, and, depending on their degree program, may be asked to make changes to MCODE with little previous experience with and understanding of the MCODE functions. Because of this, it is important for future editors of MCODE to be able to quickly gain a clear understanding of how MCODE works, such that they may have more time to implement their intended improvements.

With that objective, the MCODE-3 revision was programmed with an intentional abundance of clear and consistent comments. This represents a paradigm shift from MCODE-2, in which the comments were primarily written to assist the original programmer, to MCODE-3, in which the comments are intended to assist those unfamiliar to the code. The same methodology has been applied to other aspects of the MCODE language. For instance, MCODE-2 often compressed multiple commands into a single written line, which, while it shortens the file length, results in a disruption of the normal reading flow of the code, making it harder for a user to parse the commands. MCODE-3 only executes a single command per line, making it clearer to follow.

3.4 User Input

The user input has remained largely unchanged, and a full manual can be found in Appendix A. There are several additional input options in MCODE-3, as well as some removals.

The first is an option to keep or remove the individual SCALE input and output files for both predictor and corrector depletion runs. The option, ORGF, accepts values of 0 and 1. The default value, 0, causes MCODE to erase all the SCALE I/O files. For runs with a large number of materials, storing these files in the temporary directory takes considerable memory and can add unnecessary confusion to a standard MCODE run. The absorption and density data are stored in the [P]org[dt] (i.e. "Porg001", "Corg004", etc.) files regardless; activating this option will only allow the user to see the input and output data in its original form. Entering a 1 for ORGF and thus issuing the alternative command is only recommended when the user has need to see the actual I/O files to check statistics or run an alternative calculation for small numbers of materials.

There is also an equivalent option for MCNP, the MCF input card. If set to 0, MCODE will remove the input, output, and tally MCNP files after each MCNP calculation. However, MCODE writes a condensed at the end of each MCNP run that stores nuclide density, flux, and reaction rate data for every irradiation material. If the user may want to see an input or output file directly, the MCF card can be set to 1, which signals MCODE to store all of the MCNP input, output, and tally files in the temporary directory. MCF is set to 0 by default because of the large amount of memory necessary to store these files.

The third additional option in MCODE-3 is the TRCK option. This card can be called to force MCODE to track user-specified nuclides through both transport and depletion options, even if they do not contribute neutron absorption fractions within the specified importance level. Otherwise, if a nuclides absorption value is below the threshold, MCODE discards its density data at that timestep. This option allows a user to force MCODE to keep certain nuclides in every MCNP and ORIGEN run, and should be used when the accuracy of the quantity, no matter how small, of a certain nuclide is desired. However, in order to fully control the accumulation and removal of a nuclide, the nuclides which contribute largely to the buildup of the target nuclide should also be tracked.

The fourth option added in MCODE-3 is the ORGP card. MCODE-3 comes with the capability to run COUPLE and ORIGEN-S in parallel threads. By default, ORGP is set to 1 and only a single process runs at a time, but if the card is entered in the MCODE input file, the following value will specify how many processes to run simultaneously. This value must be a positive integer greater than or equal to 1. Because COUPLE generates a complete cross section library for every irradiation material, it makes the depletion step significantly more computationally expensive than with ORIGEN-2.2. Therefore, it is highly recommended that the user spreads the depletion processes over multiple cores when possible, particularly for problems with a large number of irradiation materials.

With the upgrade to using ORIGEN-S in the SCALE-6 software package, the ORGE card, which allowed the user to declare the ORIGEN-2.2 executable in MCODE-2, now refers to the SCALE-6 call command. By default this is `batch6.1`. For continuity, the MCODE-3 input card will inherit the name ORGE. SCALE-6 also provides gamma and decay libraries internally, so the ORGL option has also been removed.

Finally, it is worth noting in this section that the iterative flux option of MCODE-2.2 is not currently available in Version 3. This is discussed further in Chapter 6 under future work.

3.5 Output Files

A fundamental change in the I/O file structure of ORIGEN-2.2 and ORIGEN-S has given way to an overhaul of the output files that MCODE leaves in the temporary directory folder. By default, these files are:

- [P]org[dt] – stores the depletion output data for the important nuclides of all materials, condensed into a single output file per iteration for predictor and corrector calculations. Displays the concentration in grams and the absorption fraction by nuclide of every material.
- [P][m]_[dt].PCH – stores all nuclide data, including unimportant nuclides, for every material after each depletion, including corrector. At the end-of-timestep, the predictor and corrector PCH files are averaged and stored under an identical filename. If the corrector depletion is not activated, the predictor PCH file is renamed without the "P" prefix.
- mc[dt] – stores the one-group reaction rate and density in atoms/b-cm for each nuclide, the total one-group flux for every material, and the k_{eff}, ν, and 238-group flux from the P-flux MCNP calculation.

In addition, the ORGF option can cause MCODE to store all of the SCALE-6 I/O files for every material, for every timestep. Similarly, activiating the MCF option will signal MCODE to store MCNP I/O files. If the C-flux option is activated through the COR option or because of significant gadolinium presence, the four P-flux MCNP files are renamed with a "P" inserted at the front of their filename (i.e. "Pmc001"). The four C-flux MCNP files are then stored in the temporary directory under the original P-flux file names (i.e. "mc001").

4 Testing & Benchmarks

4.1 Introduction

In order to adopt a new computing code, significant testing and benchmark studies are crucial to validating its accuracy. While MCODE-2 has seen widespread use at MIT since it was released in 2006, MCODE-3 will require extensive testing in order to assure correct performance.

This work evaluates the performance of MCODE-3 in three distinct tests: (1) a standalone comparison of a single depletion of ORIGEN-S vs. ORIGEN-2.2; (2) a 52-step pin cell burnup compared with MCODE-2 and the newest version of the widely used LWR lattice code, CASMO5 [11]; and (3) a full MIT Reactor core under operating power containing both fresh and partially depleted fuel irradiation materials.

4.2 Standalone ORIGEN Comparison

With COUPLE's capability to generate a complete, model-specific cross section library from the 238-group flux and the Monte Carlo generated reaction rates for the more absorbent nuclides, it is expected that the results of equivalent depletions with ORIGEN-2.2 and ORIGEN-S will differ slightly. Over the course of a full MCODE depletion, these differences will grow in size and complexity. In order to isolate those differences, a standalone ORIGEN comparison of Version 2.2 and Version S is performed.

This standalone comparison is performed using an MIT Reactor core geometry. A single randomly selected material is studied, which is part of the MIT356 assembly in the B7 position. The fuel in the assembly is split into eight axial regions and six radial regions. The target material is in the fourth radial node and seventh axial node, meaning that it represents the region between 7.104 cm and 14.208 cm of the 56.8325 cm fuel plate, of ninth, tenth, eleventh, and twelfth fuel plates, as displayed in figure 4.1. The target material depletion is performed at constant power of 8.254 kW for one day. Default isotopic importance 0.999 is used.

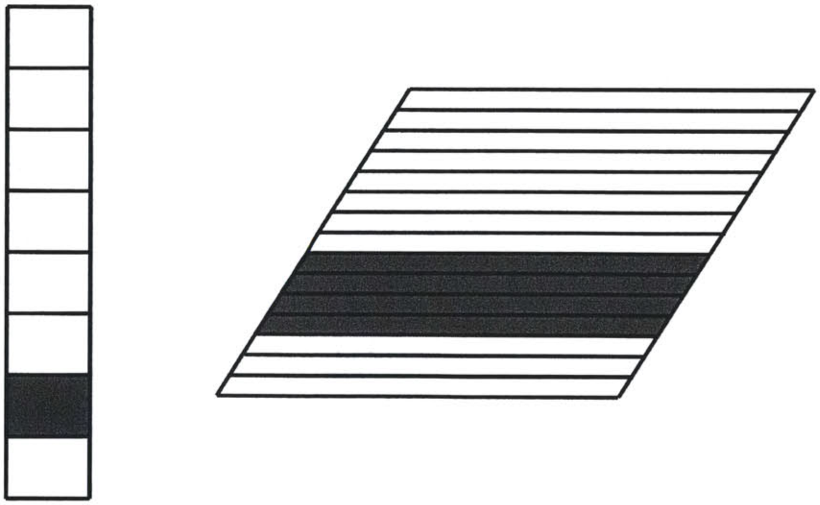


Figure 4.1. Axial and radial identification of the target material in the MITR rhombus fuel elements.

Using libraries updated from the same MCNP model, the ORIGEN-2.2 and ORIGEN-S results are largely similar, although there are a few significant and distinct differences. The eight actinide densities tallied are statistically identical, as were several of the fission products. Most of the other fission products showed minor inconsistencies of approximately 0.1%. However, six nuclides, Rh-105, Cd-113, In-115, Xe-135, Pm-148m, and Gd-147, all showed significantly more variance, up to 5.5%, as can be seen in figure 4.2.

In order to understand these discrepancies, the way that ORIGEN-S and ORIGEN-2.2 find their cross section data must be reviewed. As explained in section 3.2, ORIGEN-2.2 uses pre-compiled one-group cross section data from a number of standard libraries, typically representative of different types of reactors. MCODE-2 uses MCNP to calculate reaction rate data for the important nuclides, which are used to update the reactions associated with those nuclides; however, the rest of the cross section data is left unaltered from the original reactor-specific library.

MCODE-3, on the other hand, uses COUPLE to compile a complete situation-specific cross section library. Instead of using the important nuclide reaction rate data to update a preset library, COUPLE writes a new library from scratch, in which every known cross section is calculated from the 238-group flux supplied by MCNP. Then, it overrides the values for the important nuclides' reactions with the reaction rate data from the MCNP calculation. This creates an entirely model-specific cross-section library, with more up-to-date nuclear data like fission yield and branching ratios, as well as material-specific cross section data for important nuclides.

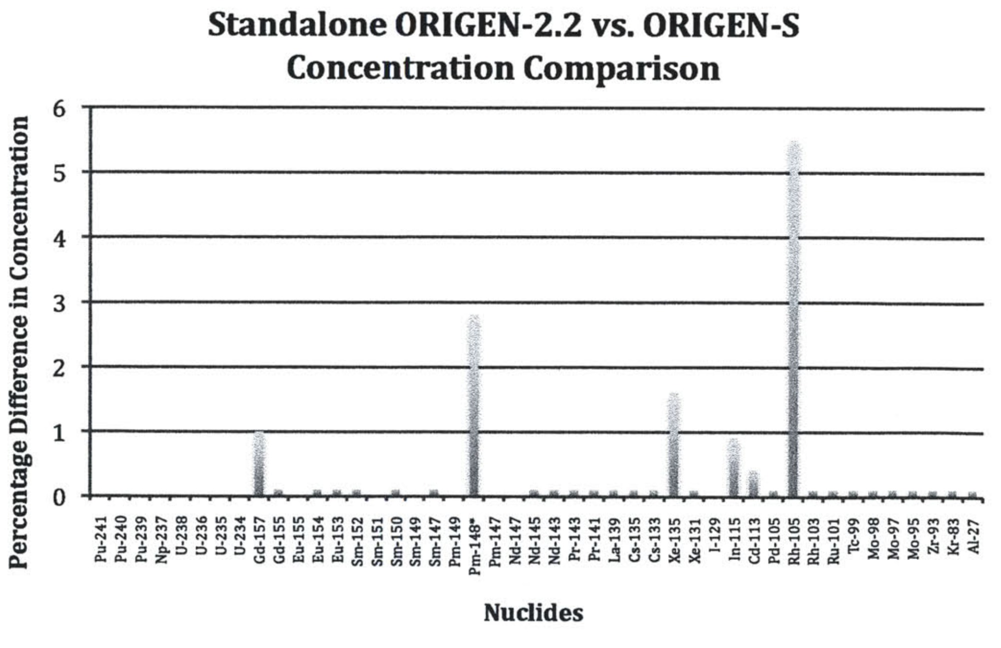


Figure 4.2. End-of-depletion nuclide concentration differences between ORIGEN-2.2 depletion and ORIGEN-S depletion.

With these differences in the cross section libraries, slight end-of-timestep depletion discrepancies are expected. There are many reaction and decay chains that affect each individual nuclide's concentration, and while the important nuclide selection process helps lock many of the most impactful reactions into exactly accurate values, others may have a noticeable or even significant effect on the outcome.

As an example, let us focus on Gd-157. At beginning-of-timestep, Gd-157 and Gd-155 are both selected nuclides, meaning their cross section values will be calculated exactly by MCNP tallies in MCODE-2 and MCODE-3. However, because Gd-156 is not selected during the timestep as being neutron important, its cross section values come from different places for ORIGEN-2 and ORIGEN-S. ORIGEN-2 uses a pre-compiled one-group library, in this case `pwrue.lib`, whereas ORIGEN-S gets cross section values from the completely model-specific library that COUPLE generates using the regionally-averaged 238-group flux. The Gd-156 neutron capture cross section is 3.823 barns in ORIGEN-2.2, and 3.666 barns, significantly smaller, in the ORIGEN-S library. While neither of these cross sections are extremely large, Gd-155's large nuclide concentration and neutron capture cross section, approximately 2000 barns, is enough to generate a significant amount of Gd-156 during the timestep, which in turn produces Gd-157. Due to the reduced cross section that ORIGEN-S is using for Gd-156's neutron capture reaction, a smaller end-of-timestep Gd-157 concentration is expected.

This is the phenomenon that is responsible for practically all the divergence between ORIGEN-2.2 and ORIGEN-S, and consequently MCODE-2 and MCODE-3: one or more nuclides that are not absorbent enough to be selected as important, still contribute significantly to the concentration of the nuclide in question. Since these nuclides are not important, their reactions are determined by the general library cross sections, which differ between the two versions.

This is particularly apparent in nuclides with smaller densities, because neutron absorption scales with nuclide concentration. Thus, the most divergent nuclides are often some of the last nuclides to count within the importance threshold, and other nuclides that might have similar importance to them but are just above the threshold can still have a significant effect on the original nuclide's evolution in time. Furthermore, if a nuclide's density is small, a small difference in a decay value or cross section of a parent nucleus will have more of a proportional effect. This is particularly evident in the standalone comparison, in which the six distinctly divergent nuclides are also the six nuclides with the smallest relative density to have been selected. In fact, their average mass, 5.26E-05 grams, is more than four orders of magnitude smaller than the average mass of all the important nuclides, 9.35E-01 grams.

By selecting an importance criteria value, the user has the ability to limit this divergence at the cost of computing time. As the criteria is raised closer to unity, there will still exist edge cases like the one above, in which the concentration of a nuclide that just meets the selection limit is largely dependent on the reaction rates of one or more nuclides that do not meet the limit. However, as the threshold is raised, those edge cases will only exist for nuclides of smaller and smaller neutron importance, and will thus have an increasingly insignificant effect on the outcome of multiple depletion steps. As shown above, a criteria of 0.999 can limit significant concentration divergence to nuclides with orders of magnitude smaller mass than average.

4.3 CASMO Pin Cell Comparison

For the second benchmark, an infinite lattice of representative UO_2 pin cells taken from a standard Westinghouse 17x17 PWR assembly is used. As seen in figure 4.3, the model consists of four cells, a cylindrical UO_2 fuel cell, a cylindrical oxygen gas gap, a cylindrical Zircalloy cladding cell, and a volume of light water, bounded by reflective surfaces to simulate the infinite lattice. The geometry and material data are described below in tables 4.1 and 4.2. The pin cell is irradiated over 52 timesteps to 50 MWd/kg at 300 K, with a power of 30W/gU. As described in section 2.3, COUPLE generates a completely unique cross section library from the regionally-averaged 238 group flux spectrum, whereas MCODE-2 uses a pre-specified PWR library updated with the reaction rates for select nuclides. The CASMO run was performed using primarily ENDF cross section data with Pu-239 cross section data from JENDL libraries. The MCODE and CASMO inputs can be found in Appendix C.

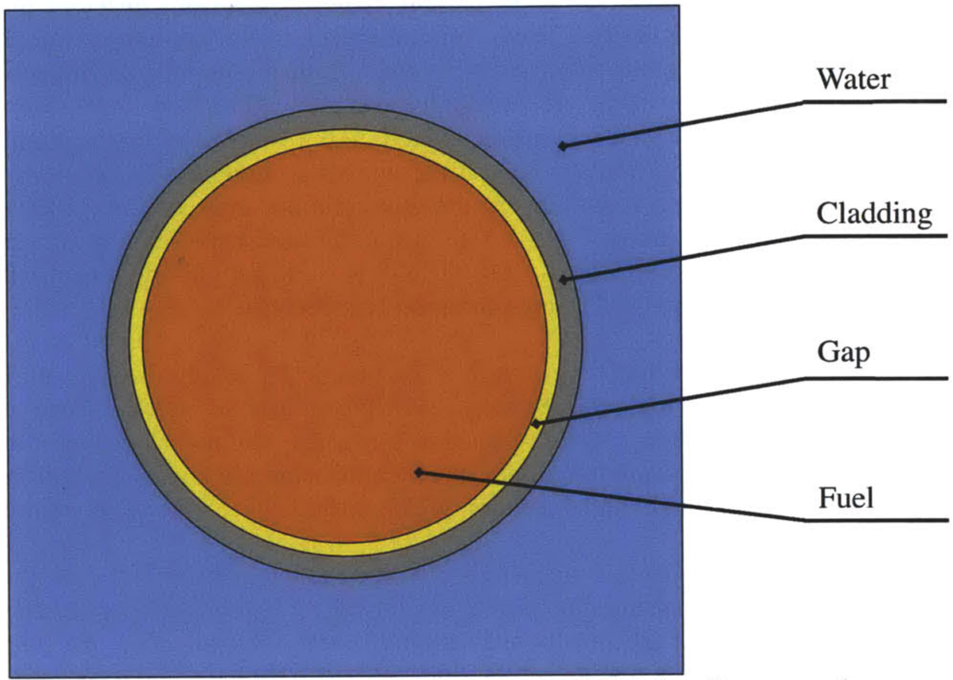


Figure 4.3. XY – plane of a representative PWR pin cell, not to scale.

Table 4.1. Geometric data of the representative PWR pin cell.

Parameters	Values [cm]
Fuel Pellet Outer Radius	0.4096
Gas Gap Outer Radius	0.4178
Cladding Outer Radius	0.4750
Pin Cell Pitch	1.26

Table 4.2. Initial material data of the representative PWR pin cell.

Material	Nuclide	Number Density [cm^{-3}]
Fuel 10.198 g/cm^3	U-235	$6.910x10^{20}$
	U-238	$2.2062x10^{22}$
	O-16	$4.5510x10^{22}$
Gas	O-16	$3.764x10^{19}$
Clad 6.55 g/cm^3	O-16	$3.082x10^{20}$
	Cr	$7.585x10^{19}$
	Fe	$9.535x10^{19}$
	Ni	$3.696x10^{19}$
	Zr	$4.243x10^{22}$
	Sn	$4.818x10^{20}$
Coolant (H_2O)	H-1	$2.23x10^{22}$
	O-16	$1.12x10^{22}$

The effectiveness of MCODE-3 is evaluated by two methods. The first is by comparing the system eigenvalue, k-infinity, at each timestep for each code. As can be seen in figure 4.4, CASMO5, MCODE-2, and MCODE-3 calculate very similar eigenvalues throughout the depletion. The largest difference is noted beyond approximately 30 MWd/kg, where the MCODE-2 and MCODE-3 eigenvalues rise above the CASMO5 eigenvalue. This end-of-burnup difference is expected, as the differences between each codes' depletion method compounds during every timestep. In figure 4.5, the MCODE-3 eigenvalue can be seen to grow to approximately 1300 pcm above the CASMO5 calculation, whereas the MCODE-2 – CASMO5 difference reaches approximately 500 pcm.

The second evaluation method is to compare nuclide concentrations at each timestep. Before specific nuclide concentrations are discussed, the sum of the difference between both MCODE-2 and MCODE-3 and CASMO5 nuclide densities is measured over every timestep, and can be seen in figure 4.6. In order to calculate the total difference for a timestep, the difference between the calculated concentrations for each version of MCODE and CASMO5 for every nuclide were summed, and that total was divided by the CASMO5 total mass and converted to a percentage:

$$Total\%Difference = 100\% \bullet \frac{\sum_{j=1}^{m} C_{j,meas} - C_{j,ref}}{\sum_{j=1}^{m} C_{j,ref}} \tag{4.1}$$

where n = the total number of nuclides;
$C_{j,meas}$ = the measured concentration of nuclide j; and
$C_{j,ref}$ = the reference concentration of nuclide j from CASMO.

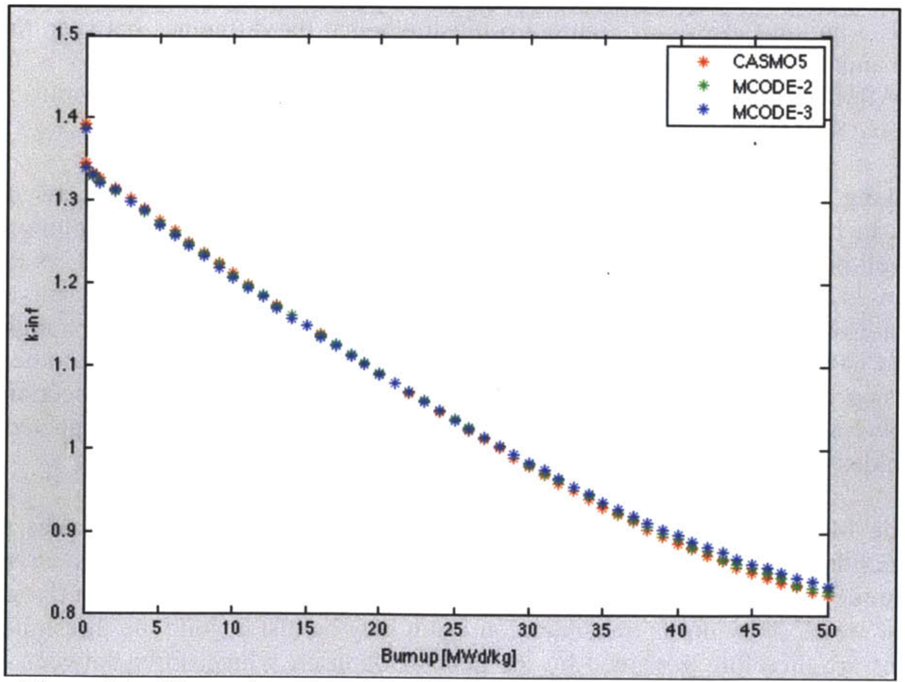


Figure 4.4. Eigenvalue comparison for 50 MWd/kg burnup of a representative PWR pin cell.

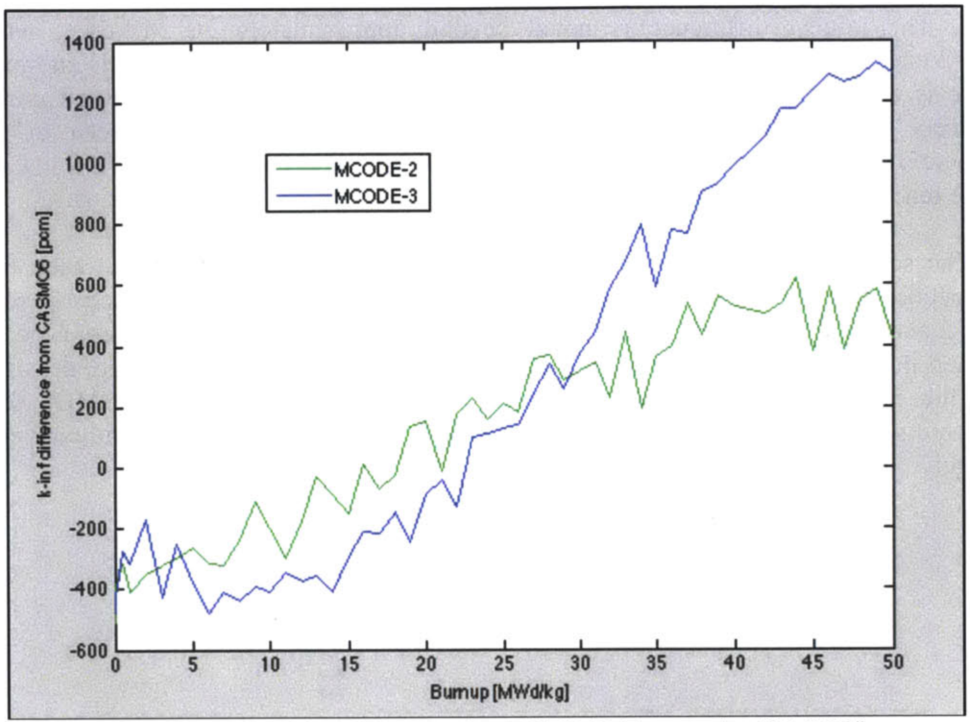


Figure 4.5. Eigenvalue difference from CASMO5 in pcm for 50 MWd/kg burnup of a representative PWR pin cell.

This is effectively a measurement of the weighted divergence of nuclide densities in MCODE-2 and MCODE-3 and in CASMO5. Figure 4.6 illustrates that the difference between the MCODE calculations and CASMO5 grows steadily with burnup. However, MCODE-3 is consistently closer to CASMO5 than MCODE-2, and as burnup increases MCODE-2 results move even farther from CASMO5.

Looking at specific nuclides, the actinides up to curium are compared between the three codes, and can be seen in figures 4.7 – 4.11. The uranium, neptunium, and plutonium match extremely well between the CASMO5, MCODE-2, and MCODE-3 calculations, as do Am-241, Am-242, Am-242m, and Cm-242 isotopes. The only actinides that see particular divergence are Am-243, Cm-244, and Cm-245. Unsurprisingly, these particular nuclides have the highest atomic numbers, and have thus been passed farther along transmutation chains then the other actinides. Since each step of their transmutation chains are governed by a set of cross section data, and MCODE-2 and MCODE-3 use different general cross section libraries, it is expected that these heavier actinides might diverge more than lighter actinides.

While MCODE strives to mitigate this problem by solving reaction rates exactly for important nuclides, two phenomena can bypass this system. First, as discussed in section 4.2, when an actinide builds up enough concentration to meet the neutron absorption threshold, other actinides that are close by on its transmutation chain may be just beyond the threshold, and their cross sections are therefore governed by the general libraries, which differ between MCODE-2 and MCODE-3. Secondly, because the accumulation of these heavy actinides is dependent on

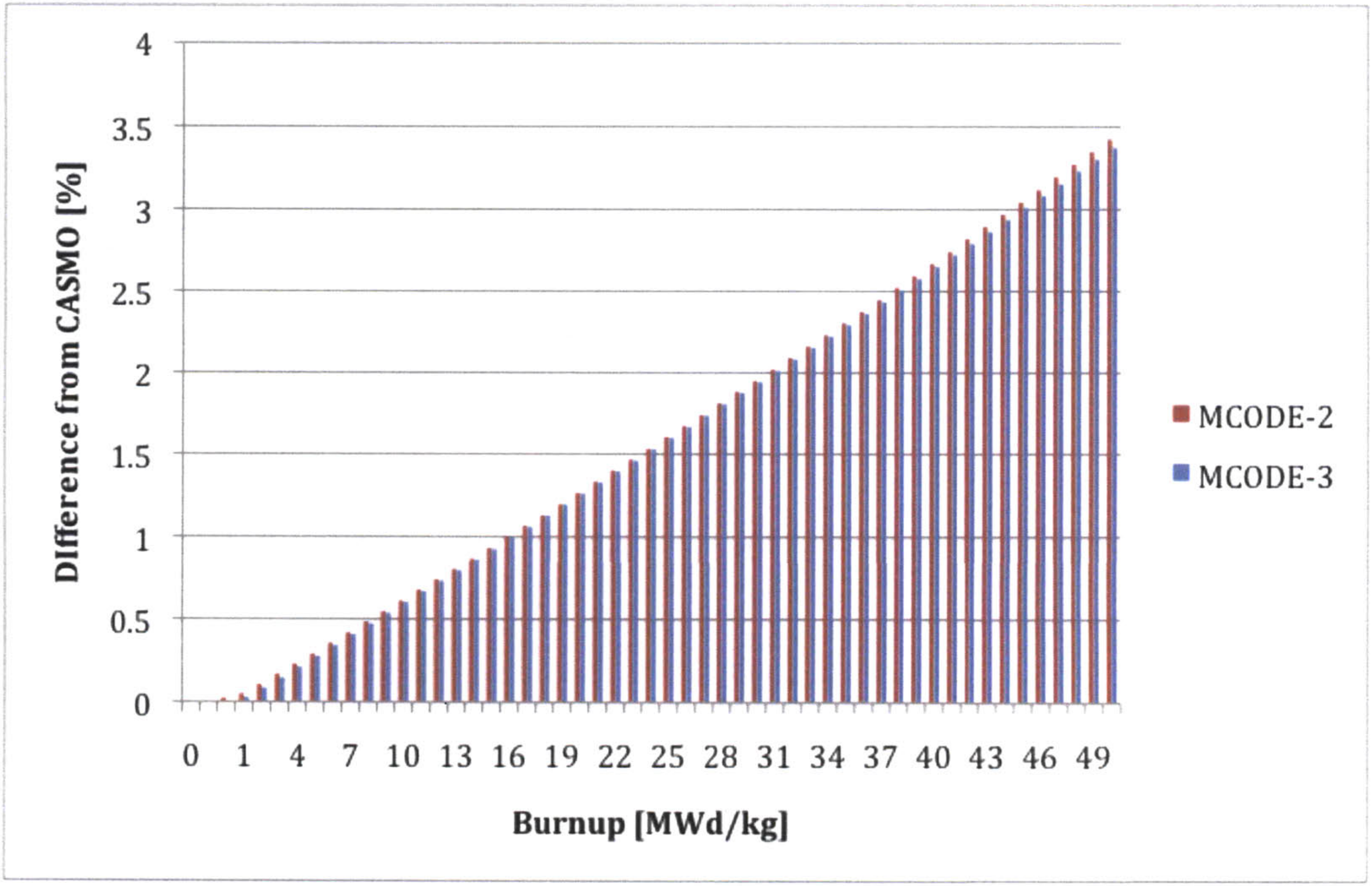
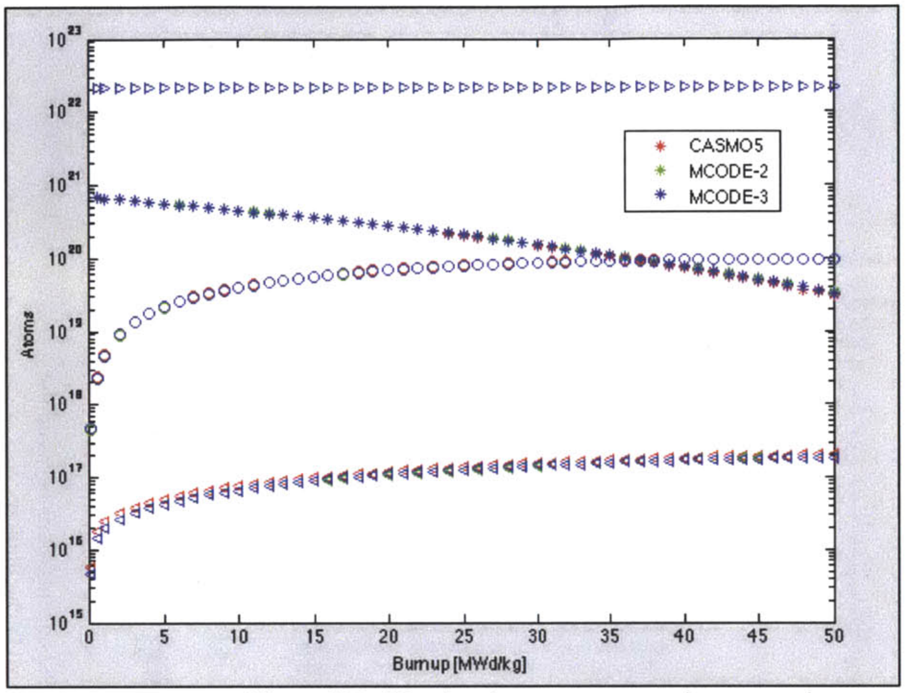


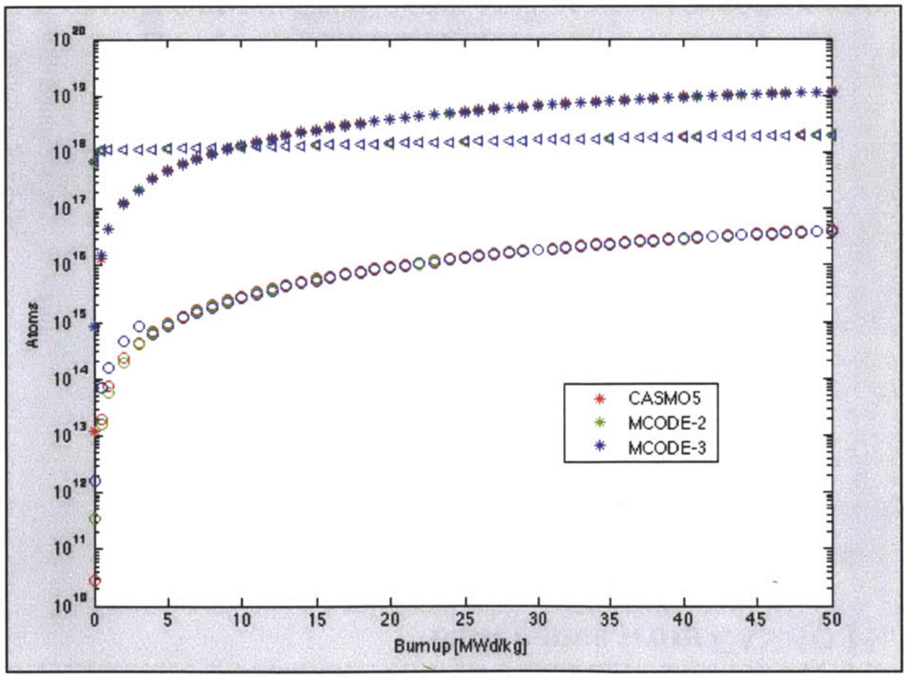
Figure 4.6. The weighted difference in nuclide concentrations between the MCODE calculations and the CASMO5 calculation.

these different libraries until they are selected, the timestep at which they do meet the absorption importance criteria will differ.



* = U-235
o = U-236
< = U-237
> = U-238

Figure 4.7. Uranium isotope concentration comparison.



* = Np-237
o = Np-238
< = Np-239

Figure 4.8. Neptunium isotope concentration comparison.

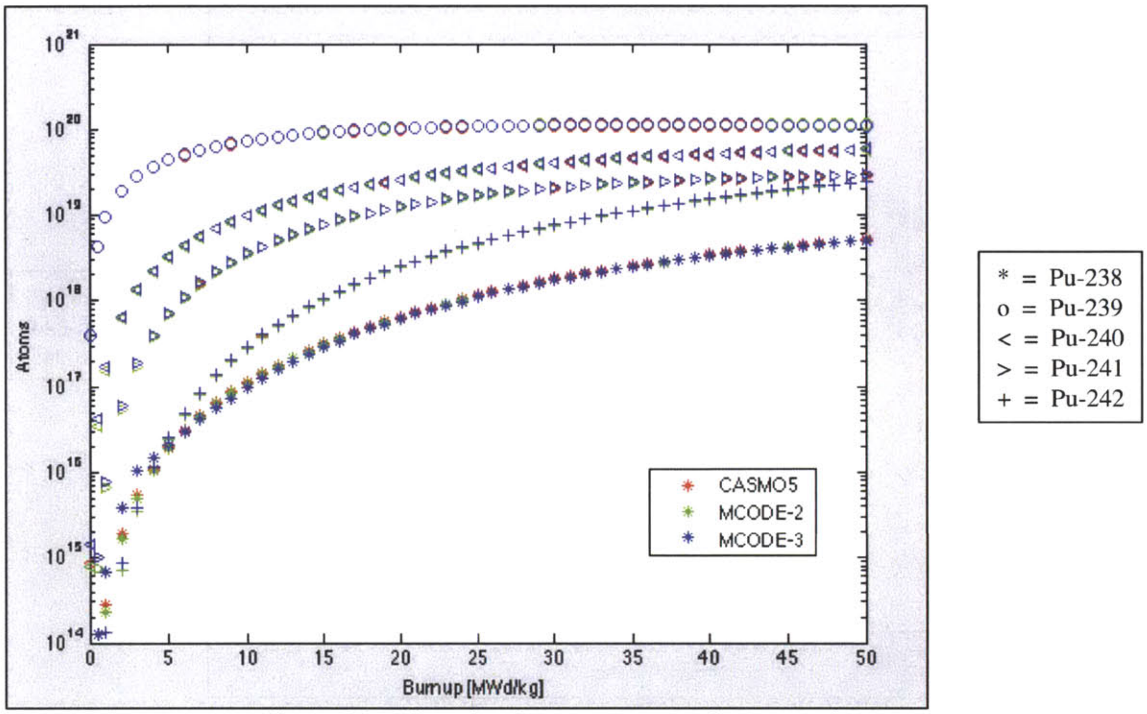


Figure 4.9. Plutonium isotope concentration comparison.

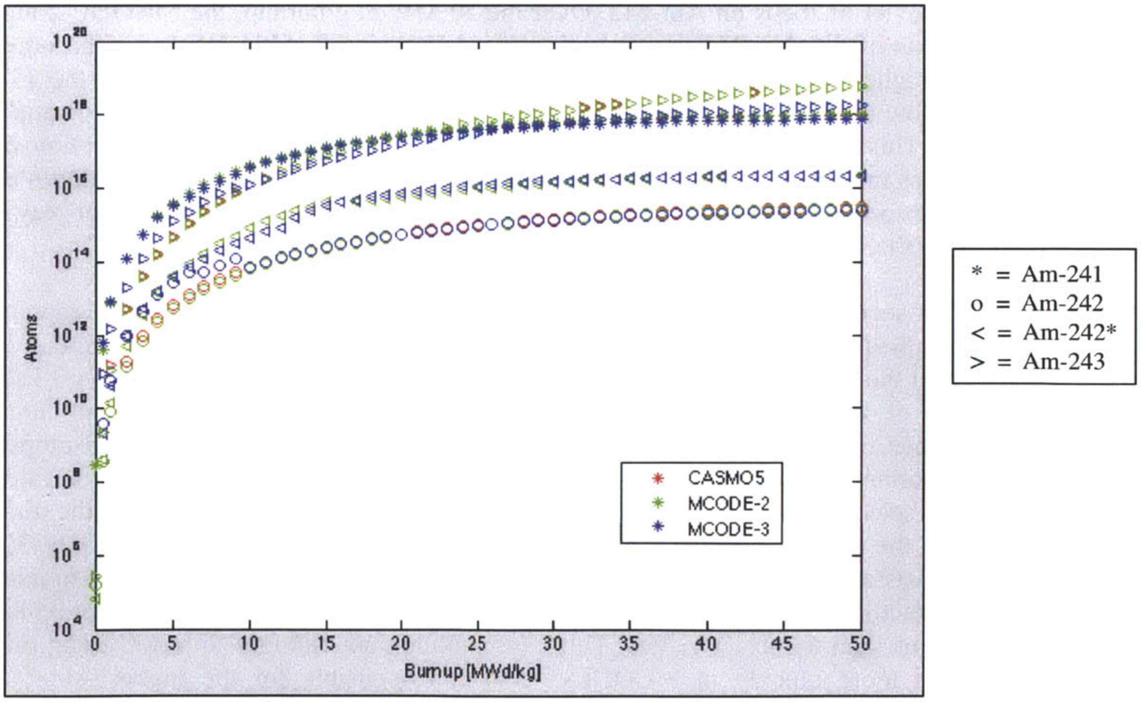


Figure 4.10. Americium isotope concentration comparison.

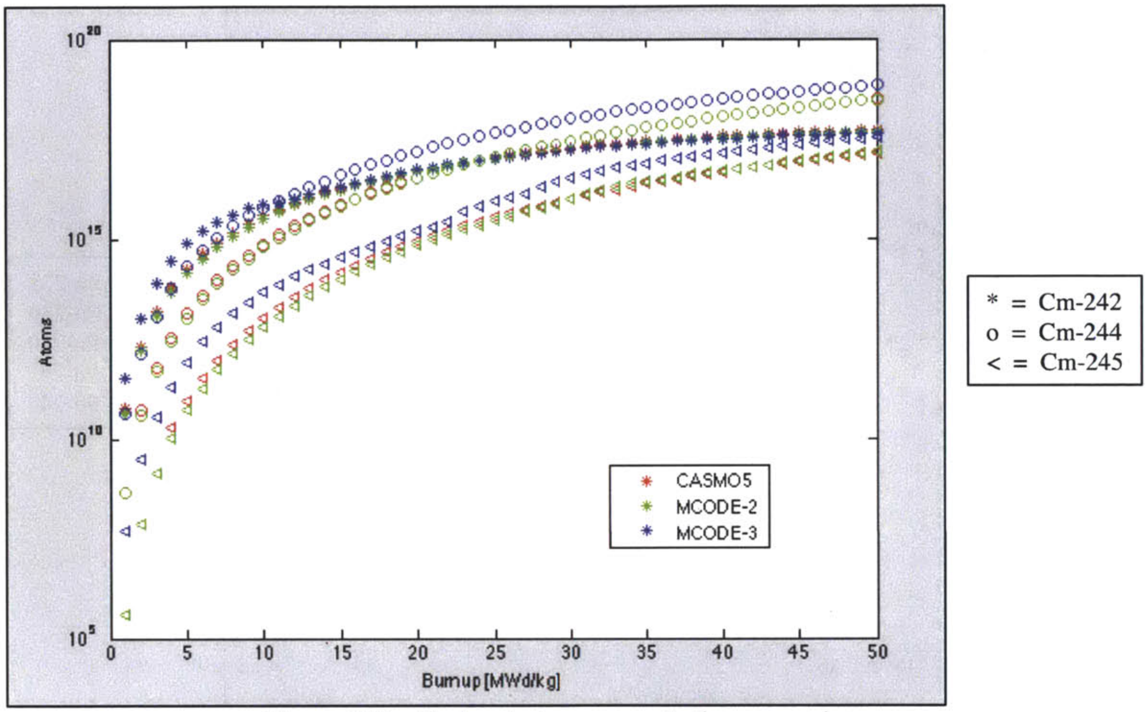


Figure 4.11. Curium isotope concentration comparison.

For example, let us focus on Am-243. Over the 50 MWd/kg burnup, the MCODE-2 and CASMO5 projections of the Am-243 concentration match fairly well. MCODE-3, on the other hand, calculates a higher Am-243 concentration in the beginning of the burnup, but at the 13th timestep, it falls below the MCODE-2 and CASMO5 predictions, both of which end significantly above MCODE-3. This evolution can be explained by two main events: (1) the selection of plutonium isotopes as important neutron absorbers; and (2) the selection of americium isotopes as important neutron absorbers. While branching ratios differ between the codes, they do not play a significant effect on the Am-243 concentration.

As can be seen in figure 4.12, Am-243 accumulates either from Am-242 capture reactions or Pu-243 **ß-** decays, the latter typically being more significant. Therefore its buildup is highly dependent on the neutron capture cross sections of the plutonium isotopes. Table 4.3 lists the cross sections of these four reactions used by MCODE-2 and MCODE-3. The large discrepancies between timesteps, particularly in MCODE3, occur when the plutonium isotopes are selected as important neutron absorbers and MCODE begins to measure their reaction rates exactly. Pu-239 is selected in the initial depletion point, Pu-240 is selected during the first timestep, Pu-241 in the third, and Pu-242 in the eleventh. The library generated by MCODE-3's 238-group flux consistently provides higher neutron capture cross sections for the plutonium isotopes than the exact reaction rate solutions. Conversely, MCODE-2's default library provides smaller cross sections than the exact values. This causes plutonium isotopes to move along this transmutation chain more quickly in MCODE-3, and is responsible for the higher Am-243 concentration seen in approximately the first 12 timesteps, until all the plutonium isotopes are selected as important neutron absorbers.

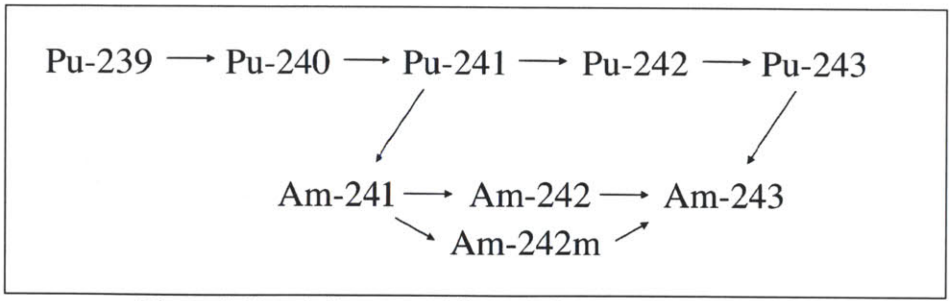


Figure 4.12. A fragment of the actinide transmutation chain.

Table 4.3. Plutonium (*n,γ*)cross section values for MCODE-2 and MCODE-3.

	MCODE-2 xs-values [barns]				MCODE-3 xs-values [barns]			
Timestep	Pu-239	Pu-240	Pu-241	Pu-242	Pu-239	Pu-240	Pu-241	Pu-242
1	64.91	213.5	33.63	32.90	65.01	7121	192	88.96
2	63.96	255.8	33.63	32.90	63.98	257.6	193.1	89.31
3	63.48	257.4	33.63	32.90	63.43	257.9	193.7	89.89
4	63.03	256.2	48.69	32.90	63.06	258.1	48.79	90.01
…	…	…	…	…	…	…	…	…
10	59.26	204.9	47.4	32.90	59.50	211.5	47.43	90.08
11	58.82	196.7	47.3	36.49	59.22	203.3	47.34	36.31

This causes MCODE-3's initial increase in Am-243 concentration over MCODE-2, but once all the plutonium isotopes are selected as important neutron absorbers and their neutron capture cross sections are calculated directly by MCNP, the Am-243 concentration growth evens out. However, within the next few timesteps Am-242m reaches the neutron importance threshold. For MCODE-2, the MCNP-calculated neutron capture cross section for Am-242m is more than twice as large as the default library cross section, so production of Am-243 from Am-242m doubles. Am-242m has a significantly higher neutron capture cross section than Am-242, and, as is seen in figure 4.10, has a consistently larger concentration than Am-242, so this has a massive affect on MCODE-2's calculation of Am-243 production. On the other hand, MCODE-3's Am-242m neutron capture cross section is virtually unchanged between when it's generated by the 238-group flux and when it's calculated specifically by MCNP. This causes a consistent increase in MCODE-2's calculation of Am-243 accumulation over MCODE-3's.

Ultimately the divergence observed between the MCODE-2 and MCODE-3 pin cell runs stems from the difference between the cross section library that COUPLE generates from the model specific 238-group flux and the generic PWR library that MCODE-2 uses. As seen in the nuclide concentration figures, the generic library augmented by MCODE's neutron importance selection function handles most nuclides well. However, several nuclides, particularly when on the border of the neutron importance criteria, can diverge significantly between burnup codes.

4.4 MIT Reactor Core Problem

The third test serves as the most complex verification, designed to measure an MCODE-3 calculation against an MCODE-2 calculation with an extensive MCODE model. To suit this purpose, an MIT Reactor core input file is used. This core has 21 fuel elements at varying degrees of burnup, and 3 elements with fresh fuel. Each element has 15 plates and 8 axial fuel regions in each plate, resulting in 2880 fuel cells, however the materials in a fuel element are split into 6 radial regions across the 15 plates, giving a total of 1152 unique irradiation materials. The input is based on the MITR skeleton input MCNP model developed by Everett Redmond II in 1989 [13], and is modeled by MCNP in figure 4.13.

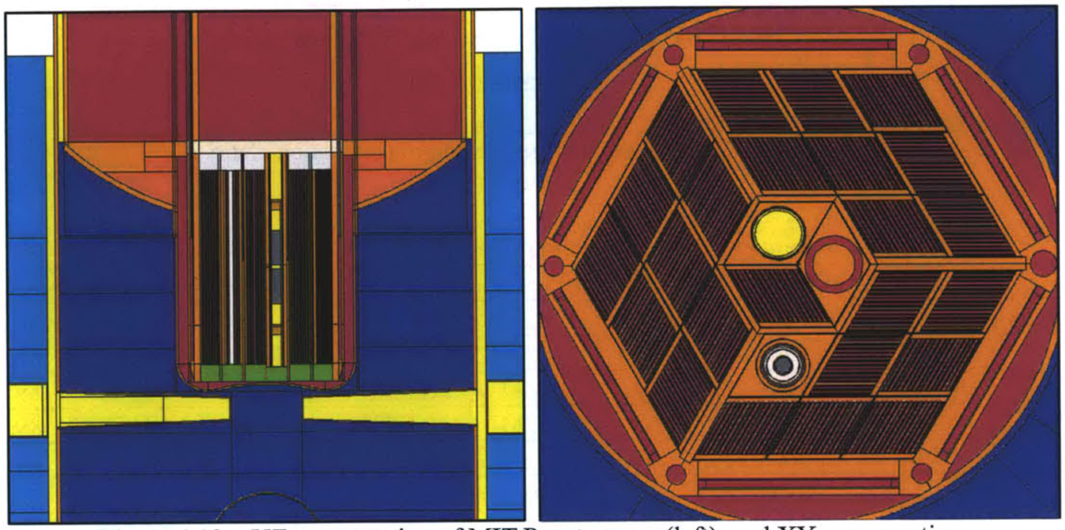


Figure 4.13. YZ cross section of MIT Reactor core (left), and XY cross section of 24 fuel assemblies and 3 test apparatuses in MIT Reactor core (right).

This is a vastly more complex geometric system than the pin cell test in section 4.3, and pushes the functionality of MCODE to its full extent. In each timestep, MCODE must handle over a thousand materials and several hundred nuclides of significant density. The MCNP calculations are considerably more time consuming, due to the increased geometric complexity as well as the tens of thousands of flux tallies it must perform to calculate reaction rates for selected nuclides. Unsurprisingly, even a few timesteps of this calculation take longer to execute than the 52 timesteps of the pin cell experiment.

Nonetheless, MCODE-3 performs as expected. After a 32 day depletion split across four timesteps at full reactor power equal to 5.0859 MW, similarities and discrepancies between MCODE-2 and MCODE-3 nuclide concentrations comparable to the pin cell experiment can be seen. The end of life percentage differences of MCODE-3 actinide concentrations from MCODE-2 actinide concentrations for a single irradiation material are shown in figure 4.14. While some concentrations, such as U-235, are within 0.1%, others exhibit differences on the order of 1%, with U-237 at a difference of 4.7%. These occur for the same reasons as in the pin cell experiment, stemming from the difference in cross section libraries used by ORIGEN-2.2 and ORIGEN-S.

Let us investigate the nuclide with the smallest concentration, Np-238. As with the standalone and pin cell tests, this difference can be accredited to the difference between the MCODE-3 238-group flux cross section library and the MCODE-2 one-group library. With Np-238, the final concentration is approximately 2% less in MCODE-3 than in MCODE-2. First, it should be noted that Np-237 is selected as an important nuclide in each step and Np-238 is not, for both codes. As an actinide, the reactions with the biggest effects on nuclide concentration are generally fission and neutron capture. Because Np-237 is always selected, its neutron capture and fission cross sections should be similar in both MCODE-2 and MCODE-3, which, from reading the cross sections, they are. The Np-238 cross sections do differ, however, because they come from the background library. The Np-238 neutron capture and fission cross sections are approximately eight times larger in MCODE-3's cross section library then in MCODE-2's. Thus, while Np-238 is produced at roughly the same rate between the codes, it is removed at a much faster rate in MCODE-3, which is why a smaller end-of-life concentration is observed.

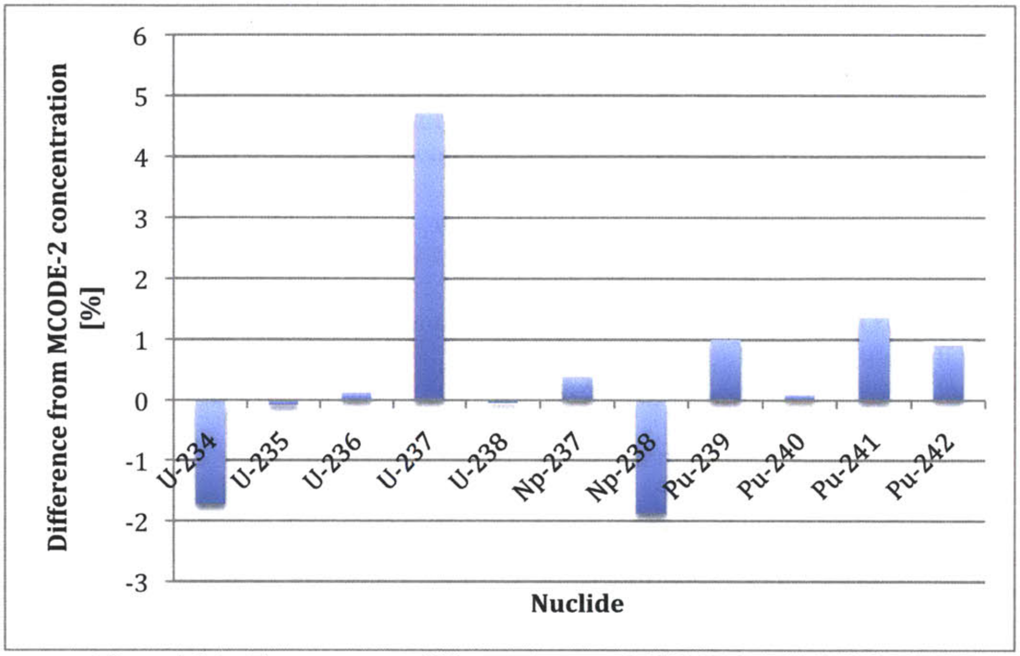


Figure 4.14. The percentage difference between MCODE-3 and MCODE-2 concentrations for actinides of significant mass for material 1371.

The rest of the other nuclides show a similar slight divergence of the 32 day burnup. From the weighted concentration difference seen in equation 4.1, the MCODE-3 nuclide densities are shown to diverge by 0.291%. The eigenvalue calculation differs by 0.054%, which is just outside the combined statistical error of 0.039%, and the flux by 0.495%. These data simultaneously demonstrate that MCODE-3 can handle complex geometries with a multitude of irradiation materials, and that MCODE-3 generates distinct results from MCODE-2. The inconsistencies are a result of the individual nuclide concentration differences, which, as has been explained throughout this chapter, are caused by the incorporation of COUPLE's 238-group flux cross section library.

5 IMPORTANCE CRITERIA SENSITIVITY EXPERIMENT

5.1 Problem Statement

As discussed in detail in Chapters 3 and 4, the primary shift MCODE-3 makes from other neutron transport and depletion linkage codes is its use of a non-standard, model specific 238-group cross section library during depletion. This is made possible by COUPLE, a module in the SCALE-6 software package. COUPLE uses the 238-group flux calculated by MCNP to create a unique library of cross sections that detail every known nuclide's removal cross section and contribution cross sections to other nuclides. Historically, a precompiled one-group library is used in an attempt to approximate the model's neutron environment as one of several reactor types. It is expected that the 238-group library generated by COUPLE will give more accurate results.

Both MCODE-2 and MCODE-3 make an additional step to improve cross section accuracy. After using the `selectimpnuclides()` function to compile a list of the most neutron absorbent nuclides, both codes use MCNP to calculate nuclide density specific reaction rates for these important nuclides, and override the corresponding entries of the "base" libraries with these exact cross section values. This approach helps to produce accurate results.

The number of nuclides that are selected as important is dependent on the user-defined importance threshold for each irradiation nuclide. This importance value is set with a trade-off in mind; increasing the threshold will allow more nuclides to be selected and their reaction rates to be calculated individually by MCNP, but this causes an increase in MCNP tallies and has a significant effect on computation time. The default importance value is 0.999, which means that the nuclides that make up 99.9% of the total neutron absorption are selected, counting the largest absorbers first.

While this approach works well, it still requires a large amount of computational power, especially as burnup increases and more nuclides are selected under the importance threshold. If COUPLE is effectively creating 238-group cross section libraries that are specific to the irradiation environment, the base library should be considerably more accurate than the precompiled one-group libraries used in the past. As such, it may be possible to reduce the importance criteria, thus reducing computation time, without sacrificing significant depletion accuracy because the "unimportant" cross sections will be model-specific from the 238-group flux.

5.2 Sensitivity Analysis

In order to provide a preliminary investigation into the accuracy of the 238-group cross section library and the dependence of depletion results on the specific reaction rates computed by MCNP for important nuclides, a comparison of three MCODE-3 runs is made. These MCODE-3 runs have successively reduced importance criteria: the first uses the default value of 0.999, the second a value of 0.990, and the third a value 0.950. For simplicity's sake, these calculations are performed on the unit pin cell defined in section 4.3, a representative UO_2 pin cell taken from a standard Westinghouse 17x17 PWR assembly. A diagram of the pin cell and tables describing its geometry and initial material compositions can be found in figure 4.3 and tables 4.1 and 4.2.

During the 50 MWd/kg burnup, the results diverge significantly as the importance threshold is lowered. At completion of the run, the calculated k-infinity value of the 99% importance calculation is 0.480% above the default 99.9% importance value, and the 95%

importance run is 2.80% above the 99.9% run. Using the weighted concentration difference formula described in equation 4.1 to measure the differences of all the nuclides, the 99% and 95% importance runs rise steadily to an end-of-life total weighted nuclide difference of 0.0912% and 0.369%, respectively, above the 99.9% importance run. These data can be seen in table 5.1, and the weighted nuclide divergence as a function of burnup is illustrated by figure 5.1. If the 238-group cross section library matched perfectly with the specific reaction rates calculated by MCNP, the concentration differences would be almost 0, especially over the first several depletion calculations. Some variance would be expected over many timesteps, as MCNP will overestimate k-eff and flux when fewer neutron absorbers are accounted for, compounding the divergence, as explained below.

Table 5.1. End-of-life flux and k-eff values for the default 99.9% importance run, and the percentage difference of the 99% and 95% importance runs from them.

	99.9% Importance (value)	99% Importance (% difference)	95% Importance (% difference)
k-eff	0.8354	0.480	2.80
Weighted Nuclide Difference	2.393E+21	0.0912	0.369

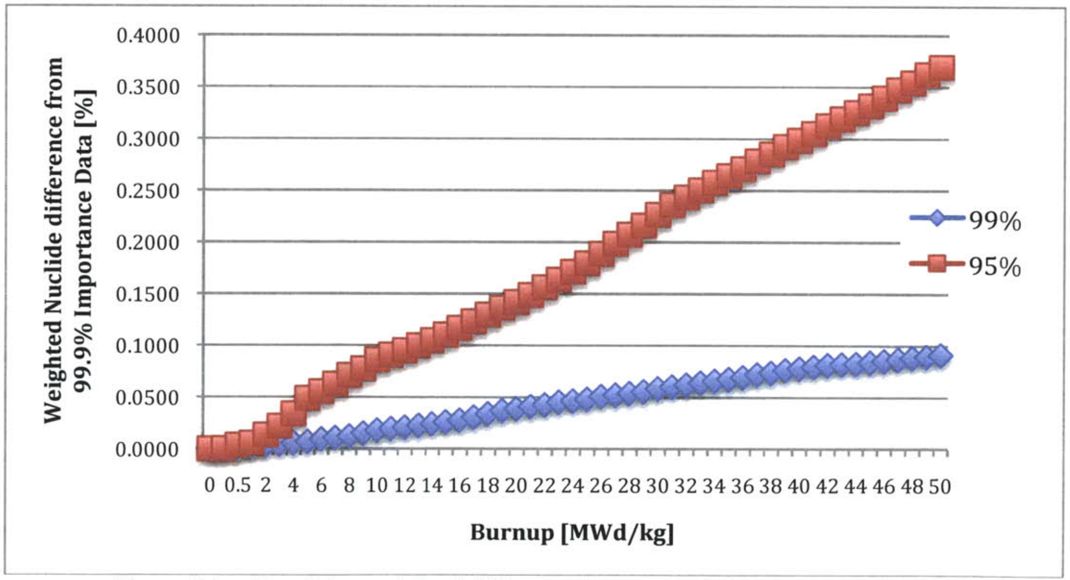


Figure 5.1. The rising weighted difference between nuclide concentrations during burnup.

Individual nuclide concentrations are also observed. Figure 5.2 and 5.3 display the final timestep percentage concentration differences from the 99.9% importance calculation of individual actinides and fission products, respectively. Just as in Chapter 4, many of the nuclides with high difference percentages have low total nuclide densities, and thus contribute little to the weighted concentration difference. Because of their relatively low concentrations, small differences in their accumulation can appear as large percentage differences. Of the selected figures, those nuclides are Am-241, Am-242m, Cm-245, Sm-149, Sm-151, Gd-155, and Gd-157.

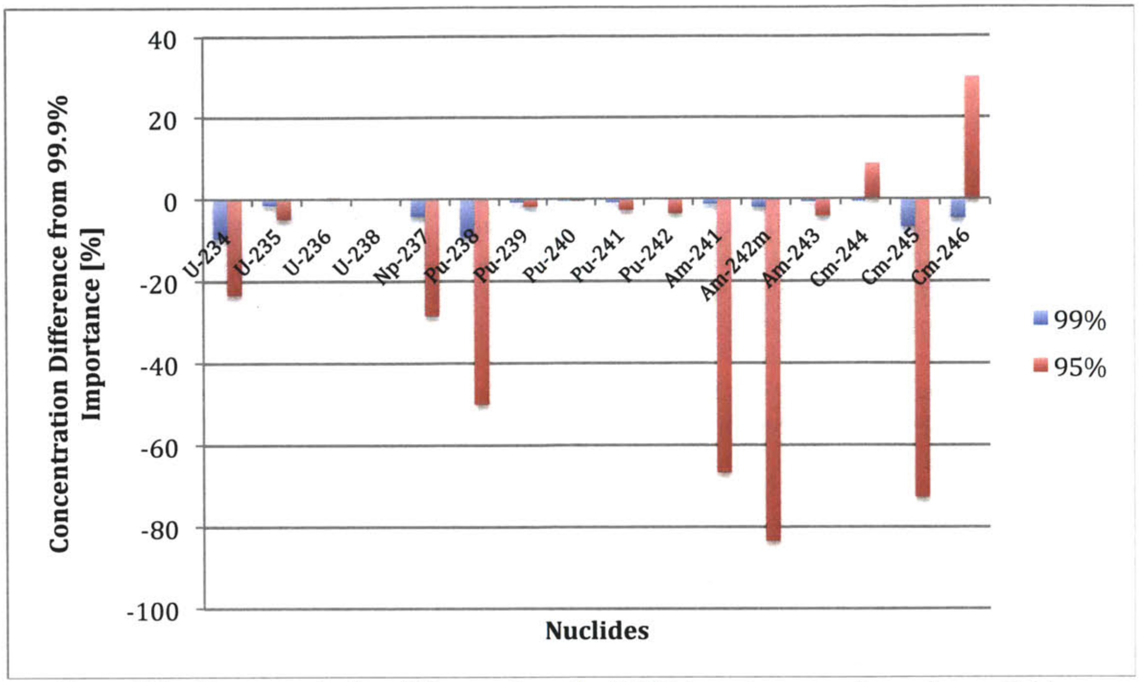


Figure 5.2. The concentration difference from the 99.9% importance run for various actinides.

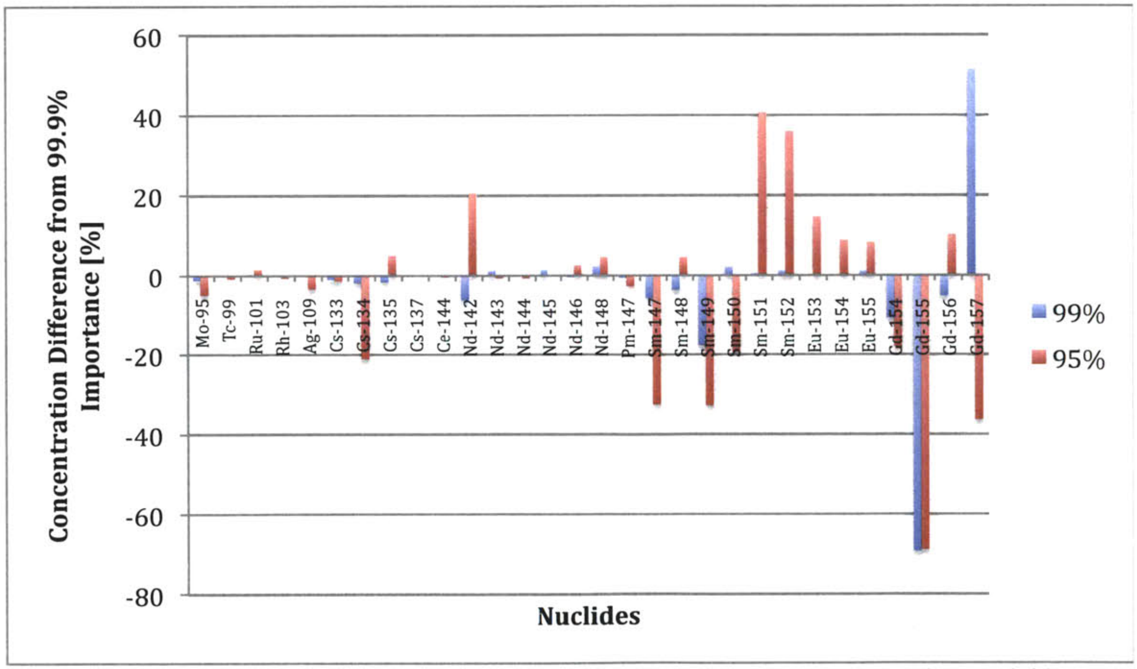


Figure 5.3. The concentration difference from the 99.9% importance run for various actinides.

The individual nuclide divergence is indicative of the importance selection process. If a nuclide diverges in either the 95% or 99% importance run, that is because the nuclides that are not selected due to the lowered importance threshold have significantly different cross section values in the base 238-group COUPLE-generated library and from the specific MCNP-calculated reaction rates. Similarly, if a nuclide in the 99% importance run matches well with the 99.9% importance run, but diverges significantly in the 95% importance run, then a cross section that has a significant effect on this particular nuclide's accumulation is in the 95%-99% importance range, and this cross section differs significantly between the COUPLE-generated library cross section and the specific MCNP-generated cross section. In both cases, the deviating cross section may belong to another nuclide responsible for contribution to the divergent nuclide, or it may belong to the nuclide itself, and thus govern removal of that nuclide.

In addition, these concentrations are affected by an increase in flux. By removing neutron absorbers from the MCNP input, k-eff and flux values will increase, and a higher flux causes overestimated depletion. This is seen clearly for the actinides in figure 5.2; almost all of their concentrations are decreased by lowering the neutron absorption criteria. This is also true for the selected fission products in figure 5.3, although increased fission yield due to higher flux competes with increased removal and some see net increases.

This effect is hard to quantify, and is an issue that must be addressed before a lowered importance criteria can be relied on for accurate results, perhaps by a corrective flux factor that is dependent on the importance threshold. However, if the desired number of timesteps is small and the user is only concerned with the important actinides with large initial concentrations, the results are fairly constant. The potential for computation reduction is also tremendous: the default 99.9% importance run took 10.31 hours spread over 8 cores, while the 95% importance run only took 3.38 hours.

6 Conclusions and Future Work

6.1 Conclusions

MCODE-3 has been developed as an upgraded version of MCODE-2, which replaces the ORIGEN-2.2 depletion module with its updated version, ORIGEN-S, because ORIGEN-2.2 is no longer supported by ORNL. In addition to upgrading depletion modules, MCODE-3 has successfully introduced the use of COUPLE, which generates a completely model-specific cross section library based on the 238-group flux spectrum supplied by MCNP. MCODE-3 has also been written in the Python scripting language instead of ANSI C, a change that will allow easier accessibility of the source code for future updates without significant cost to MCODE runtime. The appendices include the entire MCODE-3 script, compatible with the UNIX operating system, as well as an input manual, development guide, and collection of benchmark input files.

Beyond these major changes, MCODE-3 incorporates several expansions to the code's functionality. Through the use of the ORGF and MCF input file options, the user can now customize the output files to include SCALE input and output files, as well as MCNP input and output files. The user can also use the TRCK card to specify nuclides to track throughout the depletion, forcing them onto the list of selected nuclides. This allows for more accurate calculations of nuclides with small densities, which otherwise may have been grossly miscalculated due to their insignificant concentrations. Furthermore, MCODE-3 includes the option for parallel computing during the depletion calculation. The number of processes to be run simultaneously can be specified by the ORGP input card.

The three verification tests performed in Chapter 4 demonstrate how the results of MCODE-3 calculations compare both to MCODE-2 and CASMO5. While the most important nuclides match extremely well, even for long depletions, these comparisons illustrated the differences that can be expected of MCODE-3 results due to the difference between the 238-group cross section library generated by COUPLE and used by ORIGEN-S and the precompiled one-group libraries used by ORIGEN-2.2. These tests also served to verify the functionality of MCODE-3 and its application to both simple and complex input systems.

With the model-specific "base" cross section library that COUPLE provides, the exact reaction rates from MCNP might not be as necessary. Reducing the number of specific nuclides tallied in MCNP would significantly reduce MCODE runtime, which is already significant. Thus, a sensitivity study was performed to test how MCODE-3 results changed with different importance threshold values. Isotope importance settings of 99% and 95% were compared with the default importance of 99.9% for the pin cell model used in section 4.3. The results indicated that there is still a significant difference between cross sections from the 238-group library and from the specific MCNP tallies, and that the increased flux caused by decreased neutron absorption has a large impact on burnup. Further testing is needed to conclusively determine the accuracy of relying more heavily on the COUPLE-generated 238-group cross section library and of lowering the isotopic importance threshold and its impact with its impact on reactivity and flux.

As illustrated by this work, the primary difference between MCODE-3 and similar burnup isotopic codes is its use of complete COUPLE-generated cross section libraries unique to the input model. Past codes have relied on selecting a pre-compiled cross section library, usually representative of a common reactor environment, which might have large differences with the actual cross section values of the specified system. MCODE-2 attempts to mitigate this with its selection of important nuclides based on neutron absorption and its calculation of exact reaction rates for those nuclides. It even performs an additional depletion at each timestep, just to see what nuclides might be important in the following timestep, but it still leaves the vast majority of cross sections up to the pre-compiled library. As shown in Chapter 4, this is usually successful, but can sometimes have a significant effect on nuclide concentrations at each step. MCODE-3's use of model-specific libraries provides a more precise calculation.

6.2 Future Work

One addition that should be made to MCODE-3 is an iterative flux option for non-default constant flux depletions. If the constant flux depletion is chosen over the default power depletion, the FMF is calculated for beginning-of-timestep only. Using this FMF throughout the depletion may result in a distinctly different burnup step size than desired. The iterative flux design would calculate the end-of-timestep burnup step size, compare it to the expected burnup step size, adjust the FMF by a step factor, and repeat the process until the difference between the calculated and expected depletion burnup step size is within a certain amount of error.

While MCODE-3 is complete and functioning as designed, there is still extensive testing that needs to be performed, as well as work that could and should be done to improve its functionality and accuracy. Most importantly, this code should be benchmarked against known, real-world nuclear samples, in order to truly test the impact of using the 238-group cross section library versus a precompiled one-group library.

MCODE-3 has the capacity for substantial improvement in burnup accuracy over MCODE-2.2. That improvement stems from the difference in the cross section libraries used for depletion in MCODE-2 and MCODE-3. As introduced in section 3.2, MCODE-3's use of COUPLE to generate a complete cross section library from the 238-group flux allows for model specific depletion. COUPLE then takes the exact reaction rates for the most neutron absorbent nuclides and overrides those specific cross sections of the library that it just wrote. Similarly, MCODE-2 uses the reaction rates of neutron important nuclides to update a cross section library, but the base library that it is updates is a generic pre-compiled library, representative of one of several reactor types. Thus, all neutron unimportant nuclides are left with old and non-model-specific cross section values. The library that MCODE-3 uses, on the other hand, is model specific, even for the unimportant nuclides.

However, this step could be taken one step further. In this version of MCODE-3, MCNP makes a single 238-group flux measurement; when there is more than one irradiation material, MCNP tallies the average 238-group flux of all of them. This still provides model-specific cross section values, albeit not exact for each material, and essentially maintans the MCNP computing time. Also, COUPLE is given the 238-group flux by the fraction of the total flux that each bin contributes, which it combines with the specific material's total flux value, so even if the total flux varies between irradiation materials this approach is accurate as long as the flux distribution is similar between materials. However, in a model in which the 238-group flux spectrum varies drastically between irradiation materials, the accuracy of this approach decreases.

This could be easily remedied by having MCNP run a 238-group flux tally for each irradiation material. Then, COUPLE would have irradiation material specific 238-group flux spectra and its generated cross section library would be even more precise. MCODE-3 already runs COUPLE to generate a specific library for every irradiation material, so this would not add computation time in the depletion step, and the increase in the MCODE overhead time for processing multiple 238-group flux spectra would be negligible. However, this approach would cause an increase in MCNP runtime, which is already the bottleneck for MCODE-3 overall runtime. Presumably this effect would increase with more irradiation materials. If this method is considered in the future, significant testing would need to be performed in order to determine if the increase in result accuracy is worth the increase in runtime.

With the increase in cross section accuracy through the 238-group flux tally, huge runtime benefits are available by reducing the isotopic importance criteria, as discussed in Chapter 5. However, before these are adopted, a method to correct for the increase in flux and criticality due to decreased neutron absorption in the MCNP calculation must be developed.

Beyond investigating these improvements, other future work includes maintaining and updating the code, and integrating MCODE-3 into the MITR neutronics practices. With the help of the new language and increase in code documentation, along with this manual, these processes should be natural.

REFERENCES

[1] Z. Xu, P. Hejzlar, M.S. Kazimi, *MCODE, Version 2.2 – An MCNP-ORIGEN Depletion Program*. February 2006. Center for Advanced Nuclear Energy Systems. Cambridge, MA.

[2] Z. Xu, P. Hejzlar, M.J. Driscoll, and M.S. Kazimi. *An Improved MCNP-ORIGEN Depletion Program (MCODE) and Its Verification for High-Burnup Applications*. October 2002. *PHYSOR-2002*, Seoul, Korea.

[3] X-5 Monte Carlo Team. *MCNP – A General Monte Carlo N-Particle Transport Code, Version 5*. April 2003. LA-UR-03-1987. Los Alamos National Laboratory.

[4] *Scale: A Comprehensive Modeling and Simulation Suite for Nuclear Safety Analysis and Design*. June 2011. ORNL/TM-2005/39. Version6.1. Oak Ridge National Laboratory.

[5] A. G. Croff. *A Users Manual for the ORIGEN2 Computer Code*. July 1980. ORNL/TM-7175. Oak Ridge National Laboratory.

[6] R.L. Moore, B.G. Schnitzler, C.A. Wemple, R.S. Babcock and D.E. Wessol. *MOCUP: MCNP-ORIGEN2 Coupled Utility Program*. September 1995. INEL-95/0523. Idaho National Engineering Laboratory.

[7] H.R. Trellue and D.I. Poston. *User's Manual, Version 2.0, for MONTEBURNS, Version 2.0*. September 1999. LA-UR-99-4999. Los Alamos National Laboratory.

[8] *VESTA 2.0.3: A Generic Monte Carlo Code and Depletion Module Interface*. March 2011. NEA-1856/01. Institut de Radioprotection et de Sûreté Nucléaire, Cedex, France.

[9] W. Haeck, B. Verboomen. *ALEPH 1.1.2 – A Monte Carlo Burn Up Code*. January 2006. SCK-CEN-BLG-1003 Rev. 0. Studiecentrum voor Kernenergie - Centre d'Etude de l'Energie Nucléaire, Belgium.

[10] E. Shwageraus, E. Friedman, E. Abramsky, and A. Galperin. *BG-CORE – A Comprehensive Package for Reactor Core and Fuel Storage Analysis. Proc. The 23rd Conferences of the Nuclear Societies in Israel.* Dead Sea, Israel. 2005.

[11] J. Rhodes, R. Ferrer, N. Gheorghiu. "CASMO5/ A Fuel Assembly Burnup Program Methodology Manual." 2011. SSP-08/405.

[12] C. M. Kang and R. O. Mosteller. "Incorporation of a Predictor-Corrector Depletion Capability into the CELL-2 Code." November 1983. *Trans. Am. Nucl. Soc.* 729-731.

[13] E. Redmond, J. Yanch, and O. Harling. "Monte Carlo Simulation of the MIT Research Reactor." 1994. *Nuclear Technology*, vol. 106.

APPENDIX A: USER INPUT MANUAL

A.1 Introduction

The MCODE-3 input manual is nearly identical to the input manual of Version 2. The only significant difference is the inclusion of the MCF, ORGF, ORGP, and TRCK card descriptions, and the removal of the ORGL card.

The MCODE input file was design to cater towards the nuclear community's more prevalent understanding of the MCNP input format over the ORIGEN format. With that objective, most of the input file is simply the MCNP model, including a cell block with 3-D geometric model and initial material densities, a surface block that defines all surfaces used in the cell block, and a data block with initial material compositions, transport physics parameters, and source and tally specifications. Standard MCNP formatting applies, as described by the MCNP input manual. It should be noted that the MCODE-3 input file cannot contain the optional MCNP message card at the beginning of the file. This message card can be used to change MCNP file names and run options, but MCODE will do this itself. Furthermore, MCODE-3 only supports self-sustained fissile burnup problems, so only MCNP criticality calculations can be performed. As such, a kcode card should be present in the third paragraph, the data block.

The MCODE data cards come after the standalone MCNP input file in a fourth data block or paragraph, separated from the first three MCNP blocks by a blank line delimiter. The MCODE paragraph follows the general input rules:

- Each line may contain up to 80 characters;
- A $ sign (dollar sign) terminates the data entry and anything following the $ is interpreted as comments;
- Blank lines are used as terminators of the current paragraph;
- Data entries are separated by one or more spaces (tabs and hidden characters should be avoided);
- Comment lines must start with a C or c anywhere in columns 1-5 followed by at least one space; and
- Blanks in the first 5 columns indicate a continuation of the data from the last card.

One exception worth noting is that & characters should not be used as a continuation signal.

Besides a text input file, an optional MCNP source file can be used. Generated by MCNP as a binary file, the source file must be located up one directory from the input. It is able to reduce the number of inactive cycles of MCNP runs.

A.2 Input Cards

A full list of MCODE-3 input cards (to be used in the fourth paragraph of MCODE-3 input) is described in this section in table A.1. The irradiation material starts with the material number as defined in MCNP materials block. All other cards start with an alphabetical string consisting of three or four characters. The letters in the card identifier can be a mixture of capital and small letters. Not all cards are mandatory, and default values are recommended. In particular, it is highly recommended that the environment-dependent cards (MCE, MCXS, and ORGE) should be avoided.

Table A.1. List of MCODE-2.2 input cards.

Card	Description	Page
—	Irradiation material specifications (volume, temperature, etc.)	48
MCE	MCNP executable name	50
MCXS	Summary file name of MCNP cross section libraries (with directory)	51
MCS	MCNP source file option	52
MCF	*MCNP I/O file storage option	53
ORGE	SCALE executable name	54
ORGF	*ORIGEN I/O file storage option	55
ORGP	*ORIGEN asynchronous processing option	56
TAL	Manual MCNP tally specification	57
TRCK	*List of nuclide IDs to track	58
POW	System power	59
NOR	**ORIGEN depletion normalization option	60
COR	Predictor-corrector option	61
DEP	Depletion specification	62
STA	Start directive	63
END	End directive	64

* New option in Version 3
** Not yet available in Version 3

Irradiation material specification

Format:

mt-num vol ORG-XS-lib TEMP IMP MCNP-XS-opt ntal moved

where mt-num = irradiation material number (defined in the MCNP material block);

vol = occupied volume (in units of cm^3);

ORG-XS-lib = attached ORIGEN one-group cross section library;

TEMP = temperature (an integer in units of K);

IMP = the neutron absorption fraction threshold;

MCNP-XS-opt = MCNP cross section library option

0 = use user-specified MCNP cross section library,

1 = use MCNP cross section library as indicated by `preproc`;

ntal = MCNP F4 tally number (between 0 and 99);

moved = mcodeFM continuation option

0 = treat as normal MCODE run;

1 = mcodeFM supplies initial PCH files.

Default:

TEMP	300
IMP	0.999
MCNP-XS-opt	0
ntal	automatically generated in sequential order
moved	0

Remarks:

1. There is in general no limit on the total number of irradiation materials.
2. Different ORIGEN one-group cross section libraries can be attached for different irradiation materials. Table A.2 shows the list of ORIGEN cross section libraries.
3. Users need to be cautious setting the IMP value. Since the ORIGEN output has only 4 significant digits, it is meaningless to specify IMP greater than 0.9999. Default value of 0.999 is recommended (initiated in C function `void initmt(IRR_MT *mt, const int n)` in `mcodelib.c`). Refer to page 9 for discussions of including ORIGEN isotopes.
4. The MCNP-XS-opt specifies which MCNP library to use. The option 0 will keep the MCNP library as it is specified in the input. The option 1 will instruct MCODE-2.2 to use MCNP library selected by the program, `preproc`.
5. The ntal is the tally number of the material. Two or more different irradiation materials can share the same tally number.

6. In general, the moved variable should be left at default unless being activated by mcodeFM.

Example:

```
210    1.05414      PWRUE.LIB   900  0.999  1  10   $ UO2
220    2.25151  PWRDU3TH.LIB   900  0.999  1  10   $ ThO2
230  2.3522E-2      PWRUE.LIB   900  0.999           $ B-10
```

Table A.2. List of ORIGEN one-group cross section libraries.

Name	NLIB(5)	NLIB(6)	NLIB(7)	NLIB(12)	Description
PWRU.LIB	204	205	206	1	^{235}U enriched UO_2, with a burnup of 33 MWd/kg
PWRPUU.LIB	207	208	209	2	^{235}U enriched UO_2 in a self-generated Pu recycle reactor
PWRPUPU.LIB	210	211	212	3	Pu-enriched UO_2 in a self-generated Pu recycle reactor
PWRDU3TH.LIB	213	214	215	7	ThO_2-enriched with denatured ^{233}U
PWRPUTH.LIB	216	217	218	8	Pu-enriched UO_2
PWRU50.LIB	219	220	221	9	^{235}U enriched UO_2, with a burnup of 50 MWd/kg
PWRD5D35.LIB	222	223	224	10	ThO_2-enriched with makeup, denatured ^{235}U
PWRD5D33.LIB	225	226	227	11	ThO_2-enriched with recycled, denatured ^{233}U
PWRUS.LIB	601	602	603	38	3.2 w/o ^{235}U fuel, 3-cycle PWR to achieve 33 MWd/kg
PWRUE.LIB	604	605	606	39	4.2 w/o ^{235}U fuel, 3-cycle PWR to achieve 50 MWd/kg
BWRU.LIB	251	252	253	4	^{235}U enriched UO_2
BWRPUU.LIB	254	255	256	5	^{235}U enriched UO_2 in a self-generated Pu recycle reactor
BWRPUPU.LIB	257	258	259	6	Pu-enriched fuel in a self-generated Pu recycle reactor
BWRUS.LIB	651	652	653	40	3.0 w/o ^{235}U fuel, 4-cycle BWR to achieve 27.5 MWd/kg axial varying moderator density considered
BWRUS0.LIB	654	655	656	41	3.0 w/o ^{235}U fuel, 4-cycle BWR to achieve 27.5 MWd/kg, constant axial moderator density
BWRUE.LIB	657	658	659	42	3.4 w/o ^{235}U fuel, 4-cycle BWR to achieve 40 MWd/kg
CANDUNAU.LIB	401	402	403	21	CANDU, natural uranium
CANDUSEU.LIB	404	405	406	22	CANDU, slightly enriched uranium
					LMFBR: Early oxide, LWR-Pu/U/U/U:
EMOPUUUC.LIB	301	302	303	18	Core
EMOPUUUA.LIB	304	305	306	19	Axial blanket
EMOPUUUR.LIB	307	308	309	20	Radial blanket
					LMFBR: Advanced oxide, LWR-Pu/U/U/U
AMOPUUUC.LIB	311	312	313	12	Core
AMOPUUUA.LIB	314	315	316	13	Axial blanket
AMOPUUUR.LIB	317	318	319	14	Radial blanket
					LMFBR: Advanced oxide, recycle-Pu/U/U/U
AMORUUUC.LIB	321	322	323	15	Core
AMORUUUA.LIB	324	325	326	16	Axial blanket
AMORUUUR.LIB	327	328	329	17	Radial blanket
					LMFBR: Advanced oxide, LWR-Pu/U/U/Th
AMOPUUTC.LIB	331	332	333	32	Core
AMOPUUTA.LIB	334	335	336	33	Axial blanket
AMOPUUTR.LIB	337	338	339	34	Radial blanket
					LMFBR: Advanced oxide, LWR-Pu/Th/Th/Th
AMOPTTTC.LIB	341	342	343	29	Core
AMOPTTTA.LIB	344	345	346	30	Axial blanket
AMOPTTTR.LIB	347	348	349	31	Radial blanket
					LMFBR: Advanced oxide, recycle ^{233}U/Th/Th/Th
AMO0TTTC.LIB	351	352	353	35	Core
AMO0TTTA.LIB	354	355	356	36	Axial blanket
AMO0TTTR.LIB	357	358	359	37	Radial blanket
					LMFBR: Advanced oxide, 14% denatured ^{233}U/Th/Th/Th
AMO1TTTC.LIB	361	362	363	23	Core
AMO1TTTA.LIB	364	365	366	24	Axial blanket
AMO1TTTR.LIB	367	368	369	25	Radial blanket
					LMFBR: Advanced oxide, 44% denatured ^{233}U/Th/Th/Th
AMO2TTTC.LIB	371	372	373	26	Core
AMO2TTTA.LIB	374	375	376	27	Axial blanket
AMO2TTTR.LIB	377	378	379	28	Radial blanket
FFTFC.LIB	381	382	383	0	LMFBR: Fast Flux Test Facility (FFTF) Pu/U
					Clinch River Breeder Reactor (CRBR):
CRBRC.LIB	501	502	503	0	Core
CRBRA.LIB	504	505	506	0	Axial blanket
CRBRR.LIB	507	508	509	0	Radial blanket
CRBRI.LIB	510	511	512	0	Internal blanket
THERMAL.LIB	201	202	203	0	0.0253-eV cross section library

MCE — MCNP executable name

Format:

MCE MCNP-command

Default:

MCNP-command MCNP_EXE defined in params.h

Remarks:

1. This card specifies the MCNP execution command (including the path).
2. It is recommended to set MCNP_EXE in params.h to the actual MCNP execution command before compiling. Default value can then be used since this card will introduce machine-dependence into the MCODE-2.2 input.
3. On a cluster-type machine with parallel computing capability, there is a trick of running MCNP in a parallel fashion, i.e., a script file, say mcnp.exe, can be used which contains only one line: `mcnp $1 $2 $3 tasks -10x1`

Example:

```
mce  C:\USR\Codes\MCNP\4C3\mcnp.np.s.exe
```

MCXS — Summary file of MCNP cross section libraries

Format:

MCXS MCNP-xs-sum

Default:

MCNP-xs-sum	MCNP_XS_SUM defined in params.h

Remarks:

1. The summary file of MCNP cross section libraries is generated via the program, mcnpxs, with necessary manual revisions. It is required by the program, preproc, to build a one-to-one correspondence between best-suitable MCNP library and ORIGEN nuclide. Users are recommended to use the default value as in the header file, params.h.

Example:

```
mcxs  C:\USR\Codes\MCODE\v2.2\mcnpxs.sum
```

MCS — MCNP source file option

Format:

```
MCS  src-opt  MCNP-src-file
```

where src-opt = MCNP source option
0 = no MCNP source file specified
1 = MCNP source file given, and it is used every timestep
2 = MCNP source file given, but updated every timestep;
MCNP-src-file = MCNP source file name.

Default:

src-opt	0

Remarks:

1. The MCNP source file is a binary file describing fission source distribution. It is generated by the MCNP from previous runs. As a binary file, it can neither be viewed by a text editor nor be used under different platforms.
2. The motivation of use of an MCNP source file is to reduce the number of initial inactive kcode cycles; hence reduce MCNP running time.

Example:

```
mcs  0
```

MCF — MCNP I/O file storage option

Format:

MCF MCNP-files-option

where MCNP-files-option = 0, do not store MCNP I/O files
1, store MCNP I/O files.

Default:

MCNP-files-option	0

Remarks:

1. Entering a 1 will signal MCODE to store the input, output, and tally files for every MCNP run at every time step in the temporary directory. This can require significant storage space, so unless the user seeks to use individual MCNP input or output files, the default is recommended.

Example:

```
orgf 1
```

ORGE — ORIGEN executable name

Format:

ORGE ORIGEN-command

Default:

ORIGEN-command ORG_EXE defined in params.h

Remarks:

1. This card is similar to MCE, but specifying the ORIGEN execution command.

Example:

```
orge C:\USR\Codes\ORIGEN2\CODE\ORIGEN22.EXE
```

ORGF — ORIGEN I/O file storage option

Format:

ORGF ORIGEN-files-option

where ORIGEN-files-option = 0, do not store depletion I/O files
1, store depletion I/O files.

Default:

ORIGEN-files-option	0

Remarks:

1. Entering a 1 will signal MCODE to store the input and output files for every material at every time step in the temporary directory. This can require significant storage space. All relevant output data is stored in each timestep's condensed depletion output file, so unless the user seeks to use individual ORIGEN input files, the default is recommended.

Example:

```
orgf 1
```

ORGP — ORIGEN asynchronous processing option

Format:

ORGP ORIGEN-procnum

where ORIGEN-procnum = number of parallel processes by which to run SCALE.

Default:

ORIGEN-procnum	1

Remarks:

1. It is highly recommended to use parallel computing where possible both for MCNP and SCALE.
2. It is also recommended that the value of ORIGEN-procnum not exceed the number of available processing units.

Example:

```
orgp 1
```

TAL — Manual MCNP tally specification

Format:

TAL mt-num tally-spec

where mt-num = irradiation material number;
tally-spec = tally specification.

Default:

tally-spec = (all cells containing mt-num)

Remarks:

1. This card specifies the F4 tally cells.
2. Default is recommended.
3. It is possible to do a lumped depletion, i.e., one material might exist in multiple spatial regions. The depletion cross sections can be tallied from an average spectrum.

Example:

```
tal 210 (1110 1120)
```

This card means for material 210, the F4 tally is: `f4:n (1110 1120)`

TRCK — List of nuclide IDs to track

Format:

TRCK ID_1 ID_2 ... ID_n

where ID_n = the nth specified nuclide ID

Default:

(None)

Remarks:

1. This card allows for the user to specify nuclides to track across all depletion points. Normally, if a nuclide does not have a significant enough neutron absorption fraction, it is not stored. This option overrides that for specified nuclides.
2. IDs should be specified in ZZAAAM format, as shown below.
3. IDs should be separated by space characters, and can be specified on subsequent continuation lines after the TRCK card.

Example:

```
TRCK 130270 942420 942430
TRCK  130270
       942420
       942430
```

These examples are equivalent, each meaning to track Al-27, Pu-242, and Pu-243.

`POW` — System power

Format:

`POW` power

where power = total system thermal power (in units of Watts).

Default:

None

Remarks:

1. The system thermal power is rated power level for the modeled system.
2. The conversion from specific power is to multiply by the mass of initial heavy metal. The conversion from power density is to multiply by the total volume.

Example:

```
pow 1009.029   $ watts
```

NOR — ORIGEN depletion normalization option

Format:

NOR nor1 nor2

where nor1 = 1, constant flux depletion
2, constant power depletion;
nor2 = 0, iterative flux depletion
1, one-pass flux depletion.

Default:

nor1	2
nor2	0

Remarks:

1. The default normalization is constant power depletion.
2. The option nor2 is only meaningful when nor1 = 1.
3. The constant flux depletion has an error threshold of 0.0001 and a maximum iteration limit of 20.
4. The option of no flux iterations, i.e., one depletion calculation, might have special applications for the research reactor, i.e., run the whole MCNP model (including the core and the test specimen) while doing burnup calculations for the specimen only.

Example:

```
nor   1  0
```

COR — Predictor-corrector option

Format:

COR opt

where opt = 0, no corrector calculation (predictor only)
1, predictor-corrector option on with P-flux calculation only
2, predictor-corrector option on with both P- and C-flux calculations.

Default:

opt	1	if no gadolinium
	2	if Gd-155 or Gd-157 > 10^{-6} atoms/barn-cm

Remarks:

1. The default option is highly recommended. One MCNP calculation is performed each time step if no gadolinium exists.
2. The burnup predictor-corrector option is automatically omitted if the depletion step is ≤ 0.001 MWd/kg (too small).
3. The option 2 requires two MCNP calculations per time step; hence doubles the running time. It can be used to verify the option 1 calculation if desired.

Example:

```
cor   1
```

DEP — Depletion specification

Format:

```
DEP  opt    B(1)   POW(1)   NMD(1)
            B(2)   POW(2)   NMD(2)
            ...    ...      ...
            B(n)   POW(n)   NMD(n)
```

where opt = unis of burnup
D = days (time)
E = MWd/kg (exposure);
B(i) = the i-th burnup point, i=1, 2, ..., n;
POW(i) = relative power for the i-th time step, i=1, 2, ..., n;
NMD(i) = number of ORIGEN micro-depletion time steps for the i-th time step.

Default:

POW(i)	1.0
NMD(i)	20

Remarks:

1. The nomenclatures of depletion points and time steps in MCODE-2.2 are defined as follows. Suppose there are n+1 depletion points: 0, 1, 2, ..., n. There are then n time steps: 1, 2, ..., n. The i-th time step is defined between depletion points i–1 and i.
2. Since the depletion point 0 is known, i.e., time/burnup = zero, only the remaining n depletion points are needed: B(1), B(2), ..., B(n).
3. The parameters of POW(i) and NMD(i) are for the i-th time step.
4. If POW(i) is negative, the decay calculations will be performed. In such a case, the burnup unit has to be days, D, for obvious reasons.
5. Since the DEP input is across multiple lines, the first 5 columns have to be spaces (blanks) for the continuation lines.

Example:

```
dep  D        3                   $ For illustration only
             30         1
             90
            120
            540
           1000        -1
```

STA — Start directive

Format:

STA sta-pt

where sta-pt = starting point index (between 0 and n).

Default:

sta-pt	0

Remarks:

1. The burnup points are referenced according to their indices stored in the array. A typical C array index starts a 0.
2. The default start of calculation is from the first depletion point. Restart calculations are possible by setting sta-pt to desired values. Note that for restart runs, this value should be one less that the timestep where you want to begin, e.g., if you ended your last run at timestep 4 and now want to restart and run for timestep 5 to 8, sta-pt should be 4. This is because MCODE-2.2 must perform the ORIGEN burnup calculations between timesteps 4 and 5 before it can perform an MCNP criticality calculation for timestep 5 and beyond.

Example:

```
sta   $ start calculation, default 0
```

END — Start directive

Format:

END end-pt

where end-pt = ending point index.

Default:

end-pt	n (last depletion point)

Remarks:

1. The ending point index should be larger than the starting point index.
2. The ending point index should not exceed the range of input depletion points. The default is recommended.

Example:

```
end
```

A.3 Execution & Input/Output Files

The running of MCODE-2.2 is via command line only. The input/output files of the MCODE-2.2 utility programs are shown in table A.3.

Table A.3. Input/used/output files of the MCODE-2.2 utility programs.

Programs	Input	Output	Used files
mcnpxs	Console inputs	mcnpxs.sum	xsdir, MCNP xs libraries
preproc	MCODE input	ORIGEN xs library skeleton	mcnpxs.sum, ORIGEN xs libraries
mcode3	MCODE input	MCODE log file, MCNP/ORIGEN output files	mcnpxs.sum, MCNP source file, ORIGEN xs library skeleton
mcodeout	MCODE input	MCODE output	All intermediate files

As an example, given an input test.inp, the running command will be:

```
$  preproc   test.inp   -DLC203
$  mcode3      test.inp
$  mcodeout  test.inp
```

Upon completion of the run, one log file, test.log, and one output file, test.out, will be produced. The contents of the log file and the output file are self-explanatory. Note that the output file is generated by the output module mcodeout. The primary information in the output file consists of neutronics and materials information in a statepoint-by-statepoint fashion:

- *neutronics*: the eigenvalue and its standard deviation, fission rate/power fractions in pre-defined zones, various types of reaction rates
- *materials*: averaged material number densities.

The summary of the output file summarizes the neutronics and materials information.

In addition, all intermediate files are under a temporary directory ./tmpdir11. The MCNP results are named as mc[dt], where [dt] is the three-digit depletion point. The condensed ORIGEN results for the predictor and corrector steps are stored in [P]org[dt], where [P] is the predictor-corrector identifier, ether "P" or "C", and [dt] is the three-digit depletion point. And, the usage of each program (except mcnpxs) can be seen by typing the command directly.

APPENDIX B: USER INPUT MANUAL

B.1 Important Variables

The main function of MCODE revolves around the variable, irrmt. irrmt is a list, with an entry for each irradiation material specified by the MCODE input file. Each irradiation material is represented in the list, irrmt, as an instance of the class IRR_MT, which has member variables described in table B.1. irrmt is initialized by the initmt() function (where *n* is the number of irradiation materials), and throughout the run it is never reassigned, although many of the list's entries' member variables are.

Table B.1. Description of IRR_MT member variables.

Member Variable	Description
m	The MCNP material # associated with this instance of IRR_MT
cel	List of cell numbers containing material, m
ncel	Number of cells containing material, m
n	Number of initial nuclides
nucl	List of NUCL instances describing initial nuclides
IHM	Grams of initial heavy metal
ntal	MCNP F$ tally number (0-99)
temp	Temperature
imp	Neutron importance threshold
flux	Flux
nuclxs	List of NUCL_XS instances describing current irradiation nuclide data
orgxslib	The cross section library used by ORIGEN-2.2, used to find reaction type data
moved	For use by mcodeFM, describes if this is a continuation from a previous cycle

One of the member variables of the IRR_MT class is nucl. nucl is also a list, the length of which is the total number of nuclides that compose the material whose IRR_MT instance owns this nucl member variable, as defined in the material card of the MCNP portion of the MCODE input. Intuitively, each entry of the nucl list is an instance of the class, NUCL, which is described in table B.2. This is how MCODE stores data on the initial nuclide composition described by the MCODE input. This data is unchanged throughout the depletion loop; MCODE does not overwrite the number densities stored in nucl by end-of-timestep depleted values.

Table B.2. Description of NUCL member variables.

Member Variable	Description
ID	ZZAAAM ID value, integer type
xs	ZZAAA.nnc cross section library identifier, string type
at	Atomic fraction
wt	Weight fraction
awr	Atomic weight ratio

Not all the nuclides on each nucl list will be depleted, though. MCODE has certain criteria for irradiation nuclides that must be met, otherwise the ID member variable of the NUCL class is set to -1, and MCODE treats this nuclide as burnup-independent, i.e. it does not deplete it with ORIGEN-S. There are two conditions that, if met, cause MCODE to assign NUCL's ID variable to -1 and classify a nuclide as burnup-independent: (1) if the ZZAAAM ID value defines a natural element instead of a specific nuclide (i.e. 920000 instead of 922350), or (2) if the cross section identifier, stored in NUCL member variable xs, ends in a lower case "c" *and* the nuclide is not an actinide (Z<90), not gadolinium (Z!=64), not erbium (Z!=68), and not hafnium (Z!=72).

If MCODE does not assign a nuclide's ID variable to -1 by the criteria above, that nuclide is then entered into the nuclxs list, which is another member variable of IRR_MT. The nuclxs list can be thought of as the active database of nuclide information. It stores the irradiation data information that is used during the depletion calculation. Similar to nucl, nuclxs is a list in which each entry is an instance of the NUCL_XS class, as described in table B.3. NUCL_XS is very similar to NUCL, as they both store data pertinent to a single nuclide.

Table B.3. Description of NUCL_XS member variables.

Member Variable	Description
ID	ZZAAAM ID value, integer type
Xs	ZZAAA.nnc cross section library identifier, string type
M	Material number for this nuclide individually, for RR tallyng
Typ	Nuclide type (1=LE, 2=ACT, 3=FP)
Nd	Number density in atoms/b-cm
Rx	List or applicable reactions, 6 entries for SNG, SNF, N2N, N3N, SNA, and SNP reactions (1=ON, 0=OFF)
Rr	Corresponding list of reaction rates

Because nuclxs varies from timestep to timestep, MCODE handles it differently than nucl. nuclxs is first initialized at the beginning of the initial time step, with the function initnuclxs(). This function, introduced in section 2.5, effectively copies the initial nuclide data from nucl of irradiation nuclides only, as well as data for the "tracked nuclides" in tracklist. These are nuclides that the user has specified to track throughout the burnup calculation. They are added to nuclxs with a zero density unless they are one of the initial nuclides and have an initial concentration.

After its first initialization, nuclxs is reinitialized after every depletion, effectively restoring the initial nuclide composition. After the reinitialization, MCODE calls the selectimpnuclides() function, which selects the nuclides of the immediately preceeding depletion that meet the neutron absorption importance criteria. These selected nuclides are added to the nuclxs list as well. Then, MCODE calls the readorgout() function, which reads the output from the depletion calculation and replaces the number densities of the nuclxs entries with the end-of-depletion values. By restoring nuclxs to the initial nuclide composition at each timestep, MCODE removes the possibility that a nuclide that loses neutron importance in future timesteps lingers in the nuclxs list. At this point, nuclxs stores the information for the union of initial irradiation nuclides, tracked nuclides, and selected nuclides from just the depletion calculation immediately prior.

Before each depletion, MCODE introduces a new list of `IRR_MT` classes, stored in the list `allmt`. The sole purpose of this list is to load the complete library of beginning-of-timestep nuclides and nuclide densities from the end-of-timestep PCH file of the previous step. This list is used when writing the ORIGEN-S input, so that both the predictor *and* corrector depletions use the complete library of beginning-of-timestep nuclide concentration values.

B.2 Important Functions

`averagePCH(dp, mt, n)`

As the name suggests, this function averages the PCH files for the end-of-timestep predictor and corrector depleted nuclide concentrations for timestep `dp`. This function is fairly intuitive, it reads the concentrations from both PCH files, calculates the average of the two, and stores the output in a new PCH file without a prefix for every irradiation material. This function is called after the corrector depletion calculation. If a corrector option is not requested, MCODE will not call this function, and will rename the predictor PCH file without the "P" prefix, representing the end-of-timestep nuclide concentrations.

`detectgad(mt, n)`

This function is responsible for checking if any materials have a significant amount of gadolinium 155 or 157, which are two strong neutron absorbers. If either of these nuclides have a concentration greater than 1.0×10^{-6} atoms per barn cm, the function returns the list of the material numbers containing the significant Gd densities, otherwise it returns 0. If 0 is not returned, MCODE will force the C-flux MCNP calculation, because a timestep with significant neutron absorber concentration can cause a large change in flux over the timestep, and thus a C-flux calculation is necessary to preserve accuracy.

`findType(mti, Fxstal)`

This function is used to return a list of the types associated with the irradiation nuclides of a material, `mti`, which is an instance of `IRR_MT`. These types (light elements are 1, actinides are 2, and fission products are 3) are necessary for the ORIGEN-S input file. This function is only necessary when generating the initial timestep PCH file; afterwards, the types will be stored in the output PCH files from each depletion. It should be noted that the input `Fxstal` is the file variable for the cross section ID library generated by the `preproc` program. While this can be unique for each material, it is often the same, so MCODE-3 saves significant computing time by only opening it once unless the file is different.

`initnuclxs(mt, n, tracklist)`

As mentioned in section 2.5, this function initializes the `nuclxs` list for each material. In order to do so, it opens up the cross section ID library and stores data on every nuclide's applicable reaction types. It uses a similar practice as `findType()` in that it only reopens the library if it's different in order to reduce computing time. Then, the function adds an instance of `NUCL_XS` to `nuclxs` for every nuclide in `nucl` with an `ID` > 0. Each `NUCL_XS` instance is initialized with its initial number density from `nucl`, and with the reaction type data taken from the cross section identifier library. `nuclxs` is then appended with all the nuclides specified in `tracklist` if they haven't already been added to `nuclxs`, and each is assigned a number density, `nd`, of 0 and appropriate reaction type data. `mt` is then returned by the function, now with a new `nuclxs` list that contains initial concentration data as well as reaction type data for irradiation nuclides and reaction type data for tracked nuclides.

readmcmct(dp, mt, n, taldict)

This function reads the condensed tally output data from an MCNP run and stores it in mt. That data includes one-group flux and individual nuclide reaction rate data for every material, as well as regionally averaged 238-group flux. taldict is generated when the MCNP input file is written, and is used to locate and assign the correct reaction rate data to the appropriate nuclides.

readmcout(dp)

This function is called at the same point as readmcmct() and is used to store the keff and nu data for the corresponding MCNP run. These values are return by the function to MCODE.

readorgout(P, dp, mt, n)

This function reads the MCODE-written depletion output file, which stores nuclide concentrations for every important nuclide. It applies these values to each materials nuclxs nuclides, setting their values equal to those in the depletion output file, and thus setting them to their depleted values.

readPCH(P, dp, mt, n)

This function is used to read the end-of-timestep PCH files from the previous timestep, and store the nuclide concentrations in those files to an irradiation material's nuclxs list. This function is used by MCODE before depletions in order to correctly input the beginning-of-timestep nuclide concentrations into the ORIGEN-S input file.

selectimpnuclides(P, dt, mt, n)

As introduced in section 2.5, this function reads through each material's ORIGEN-S output file and selects the nuclides that contribute the most to neutron absorption. It does this by creating a list of nuclides and a corresponding list of their absorption fractions, then summing their absorptions, largest to smallest, until the total fraction reaches the neutron absorption threshold for that material as set by the input file. Any nuclide whose absorption fraction was under the threshold is added to that material's nuclxs vector, and its reaction rates will be tallied specifically by the next MCNP calculation.

writemc(P, dp, mt, n, nu, keff, fl)

After an MCNP calculation, this function writes a file that condenses all the pertinent tally information into a single output file for that timestep. This includes the keff and nu values for that MCNP calculation, each material's flux and total number density, and the number densities and reaction rates of every nuclide for each material. This file is always saved in the temporary directory for each timestep.

writeorg(P, dp, mt, nmd, n, relpow)

This function is analogous to ORIGEN-S as writemc() is to MCNP, and also writes a condensed output file which is always stored in the temporary directory for each timestep. It reads through every material's ORIGEN-S output and stores values on nuclide concentration and absorption for all selected nuclides. It then writes these data into a single output file for each timestep.

writePCH(dp, mt, n)

MCODE only calls this function once, and its purpose is to create initial PCH files that just mimic the initial material compositions specified in the MCODE input file. This allows the first depletion to use the readPCH() function to create the second list of IRR_MT instances that writecouporginp() requires to run.

APPENDIX C: PIN CELL BENCHMARK INPUT FILES

C.1 High Burnup of UO$_2$ Pin Cell

CASMO-5 INPUT:

```
TTL * uniform pin case
SSP 10*1
TFU 300.
TMO 300.
PSC 0
PDE 30.
PWR 3 1.26
PIN 1 0.4096 0.4178 0.4750/'1' * Fuel Pin
FUE 1 10.0196/6.910E20 8000=4.5510E22 92238=2.2062E22
LPI
1
DEP 0.1,0.4,0.5,69*1.0/'DE'
FUM „3
STA
END
```

MCODE-2 and MCODE-3 INPUT:

```
High Burnup U02 (9.75 w/o)
1  1 -10.198  -7 1 -2 imp:n=1
2  2 -1.0E-03  7  -8 1 -2 imp:n=1
3  3 -6.55  8 -9 1 -2 imp:n=1
4  4 -0.99777  9 1 -2 3 -4 5 -6 imp:n=1
5  0  #(1 -2 3 -4 5 -6) imp:n=0

*1  pz  -0.9486374     $ These dimensions make 1 cc of IHM,
*2  pz   0.9486374     $ which makes 10.198g of UO2
*3  px  -0.63
*4  px   0.63
*5  py  -0.63
*6  py   0.63
7   cz  0.4096
8   cz  0.4178
9   cz  0.4750

m1   92235.60c 6.910E20 92238.60c 2.2062E22
     8016.62c 4.5510E22
m2   8016.62c 1
m3   8016.62c -0.125  24000.50c -0.10    26000.50c -0.135
    28000.50c -0.055  40000.66c -98.135  50000.42c -1.45
m4   1001.62c 6.6630E-02 8016.62c 3.3315E-02
mt4  lwtr.01t
ksrc  -0.2 0 0   0 0.2 0   0.2 0 0   0 -0.2 0   0 0 0
      0 0 -0.5   0 0 0.5
```

```
kcode 3000 1.0 20 320
prdmp 200 200 1
print

1  1 pwrue.lib
mce  ../mcnp.sh
orgf 1
mcf 1
POW 269.668   $ From CASMO, 30 W/gU
DEP E   0.1
      0.5
      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
      21.0
      22.0
      23.0
      24.0
      25.0
      26.0
      27.0
      28.0
      29.0
      30.0
      31.0
      32.0
      33.0
      34.0
      35.0
      36.0
      37.0
      38.0
      39.0
      40.0
```

```
    41.0
    42.0
    43.0
    44.0
    45.0
    46.0
    47.0
    48.0
    49.0
    50.0
STA
END
```