LLNL-TR-496535

LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Assessment of Non-traditional Isotopic Ratios by Mass Spectrometry for Analysis of Nuclear Activities. Annual Report 2011

S. R. Biegalski, B. A. Buchholz

August 30, 2011

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Annual Report – 2011

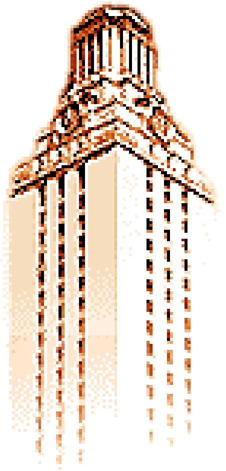
Assessment of Non-traditional Isotopic Ratios by Mass Spectrometry for Analysis of Nuclear Activities

Steven Biegalski, Ph.D., P.E
The University of Texas at Austin
1 University Station, R9000
Austin, TX 78712

Bruce Buchholz, Ph.D.
Center for Accelerator Mass Spectrometry
Lawrence Livermore National Laboratory
Mail stop L-397, P.O. Box 808
7000 East Avenue
Livermore, CA 94551-9900

August 31, 2011

Contract #HDTRA1-08-1-0032



1.0 Objective

The objective of this work is to identify isotopic ratios suitable for analysis via mass spectrometry that distinguish between commercial nuclear reactor fuel cycles, fuel cycles for weapons grade plutonium, and products from nuclear weapons explosions. Methods will also be determined to distinguish the above from medical and industrial radionuclide sources. Mass spectrometry systems will be identified that are suitable for field measurement of such isotopes in an expedient manner.

2.0 Scope

This proposal is in support of the Basic Research Program for Combating Weapons of Mass Destruction, and the assessment of non-traditional isotopic ratios by mass spectrometry for analysis of nuclear activities. Isotopic ratios will be calculated for radionuclides produced in commercial nuclear reactor fuel cycles, fuel cycles for weapons grade plutonium, and nuclear weapons explosions. The isotopic ratios that best identify the source of the radionuclides will be selected. Isotopes with a combination of low production yields and low mass spectrometry detection limits will be removed from consideration. Mass spectrometry techniques will be evaluated as a function of their ability to detect and qualify the radionuclides of concern. A mass spectrometry system design will be identified that has the detection sensitivity necessary for the work and is capable of field operations.

Option Year 1(FY 2010) specifically focused on assessing the uncertainty and range of the results obtained in the first two years of this project. Similar calculations will be performed, but utilizing more advanced codes (such as MCNPX) and different databases. Current commercial and future proposed fuel cycles will be modeled. An effort will also be conducted to develop a test data set to extract forensic content from the test datasets. Both real and simulated data sets will be utilized. Sample preparation methods will be developed for measurements in Option Year 2 of this project.

Option Year 2 (FY 2011) will focus on the quantification capabilities of mass spectrometry methods for the elements of choice. Forensic identification algorithms and a software tool for forensic analysis will also be developed and prototyped. Mass spectrometry measurements of the radioactive isotopes of forensic interest will be conducted.

3.0 Background

Isotopic Ratios

There are many sources for radionuclides in our environment. These include naturalsources, the commercial nuclear industry, nuclear weapons, the medical industry, and other sources. Often times, the source of the radionuclide may be determined through just identification of the radionuclide. If radionuclides are produced through different sources, the identification of the source is complex. In order to ascertain a specific source for attribution, radionuclide ratios are often employed.

Production yields of radionuclides from fission are a function of many variables including: the fissile material, the energy spectrum of the neutron flux, the magnitude of the neutron flux, and the duration of the irradiation. As a result, the ratios of certain radionuclides are highly dependent of these variables and may be utilized to distinguish between radionuclides produced from nuclear weapons, medical waste, short nuclear fuel cycles (e.g. ^{239}Pu production fuel cycles), and long nuclear fuel cycles (e.g., commercial nuclear fuel cycles). While the above is easily stated, the difficult part is to determine which radionuclide ratios should be utilized for best forensic value.

As an added complication, nuclear debris taken for forensic analysis often does not come directly from the source. There is often some type of chemical process or other process that may alter the sample composition. Chemical fractionation issues result and may significantly alter ratios of radionuclides of different elements. To mitigate this problem, it is best to examine isotopic ratios of individual elements since these ratios will be largely unaltered by chemical processes.

Mass Spectrometry

Traditional methods for radionuclide detection depend upon measuring the energy released during radioactive decay. Decay counting is relatively simple, but sample prep and analysis takes time to complete. If short-lived radionuclides have already decayed, traditional counting can be quite slow. Mass spectrometry (MS) techniques often require comparable sample prep to decay counting, but analysis is faster since MS counts atoms rather than waiting for them to decay. Reducing time between sample collection in the field and reliable analytical results requires switching to MS.

There are numerous MS techniques capable of measuring isotopic ratios. The sample size, detection limit, dynamic range, sample prep requirements, and ease of analysis vary widely among the techniques. Some techniques have very simple sample prep, requiring only dissolution in acid or combustion prior to analysis. Others require extensive preprocessing that impedes quick turnaround. In practice, the selected MS technique will need to accurately measure isotope ratios in a range of interest as quickly as possible. To insure speedy analysis, the MS technique should probably be sufficiently robust to be field deployable inside a transportainer.

Results from Current Work

Extensive work was conducted during the first two years of this project. Nuclear reactor fuel cycles were modeled utilizing ORIGEN. Fuels cycles from pressurized water reactors (PWR), boiling water reactors (BWR), and Canadian natural deuterium (CANDU) reactors were all evaluated. Nuclear weapons were modeled by utilizing a bare sphere (k_{eff} =1.0) in MCNPX utilizing the BURN card. The production of every fission product, activation product, and transuranic was recorded and entered into a database.

An R value was calculated for each possible isotopic ratio. This is a metric to evaluate the forensic value. R values greater than 100 or less than 0.01 are considered good. R values are calculated as shown in equation 1.

$$R = \frac{\left({}^{A}_{Z}\mathrm{X}\right)_{\text{unknown reactor}} \Big/ \left({}^{B}_{Z}\mathrm{X}\right)_{\text{unknown reactor}}}{\left({}^{A}_{Z}\mathrm{X}\right)_{\text{reactor}} \Big/ \left({}^{B}_{Z}\mathrm{X}\right)_{\text{reactor}}} \qquad (1)$$

Isotopic ratios were then prioritized by magnitude of the ratio, the absence of possible interferences in field monitoring, and the mass of isotope produced. Table 1 shows the isotopic ratios identified as the best for distinguishing between an unknown reactor type and a known commercial reactor signature. For this study the ratio of ^{134}Cs/^{135}Cs was determined to be the optimum ratio.

Table 1. Top forensic indicators to differentiate between nuclear weapons and commercial nuclear reactors.

				Present in…			A good isotopic indicator at……..				
	Element	**Isotope 1**	**Isotope 2**	**^{235}U sphere**	**Pu sphere**	**^{233}U sphere**	**1 Day**	**7 Days**	**1 Month**	**1 Year**	**10 Years**
1	**Cs**	134	135	Y	Y	Y	Y	Y	Y	Y	Y
2	**Eu**	154	156	Y	Y	Y	Y	Y	Y	Y	N
3	**Pm**	147	148	Y	Y	Y	N	Y	Y	Y	N
4	**Sn**	121	123	Y	Y	Y	N	N	Y	Y	Y
5	**Sm**	146	151	Y	N	Y	Y	Y	Y	Y	Y
6	**Cs**	134	136	Y	Y	Y	Y	Y	Y	Y	N
7	**Pm**	148	149	Y	Y	Y	Y	Y	Y	N	N
8	**Ag**	108	110	N	Y	N	Y	Y	Y	Y	N
9	**Ag**	110	111	Y	Y	Y	Y	Y	Y	N	N
10	**Pm**	148	151	Y	Y	Y	Y	Y	Y	N	N
11	**Tb**	160	161	Y	Y	Y	Y	Y	Y	N	N
12	**Eu**	154	157	Y	Y	Y	Y	Y	N	N	N
13	**Nb**	94	97	Y	Y	Y	Y	Y	N	N	N

R value calculations were also conducted on short and long nuclear fuel cycles in PWR, BWR, and CANDU reactors. The short cycle was defined as one that produces weapons grade Pu. The long fuel cycle was defined as one that was indicative of normal commercial nuclear reactor operation. Similar to above, the ratios were then prioritized by the magnitude of the R value, the absence of possible interferences in field monitoring, and the mass of isotope produced. Table 2 shows the results for the best forensic indicators to determine fuel cycle length. The ratio of ^{146}Sm/^{151}Sm was determined to provide the best forensic value.

Table2. . Top forensic indicators to differentiate between commercial nuclear reactor fuel cycle length.

				Present in…			A good isotopic indicator at……..			
	Element	Isotope 1	Isotope 2	BWR	PWR	CANDU	0 Days	1 Month	1 Year	10 Years
1	Sm	146	151	Y	Y	Y	Y	Y	Y	Y
2	Ba	133	140	Y	Y	N	Y	Y	Y	N
3	Pm	145	147	N	N	Y	Y	Y	Y	Y
4	Cd	109	115	Y	Y	N	Y	Y	Y	N
5	Sm	145	151	Y	Y	Y	Y	Y	Y	N
6	La	137	140	Y	Y	N	Y	Y	Y	N
7	Ag	110	111	Y	Y	Y	Y	Y	N	N
8	Pm	145	151	Y	Y	Y	Y	Y	N	N
9	Pm	145	149	Y	Y	Y	Y	Y	N	N

As a result of the above work, Sm and Cs were identified as the best elements to focus on for nuclear forensics with mass spectrometry.

4.0 Tasks/Scientific Goals

Task 4.1: Utilize MCNPX to calculate isotopic ratios for PWR, BWR, and CANDU reactors. Compare results with those from ORIGEN obtained in the initial Phase of this work. (FY 2010)

The Oak Ridge Isotope Generation and Determination code, or ORIGEN, is frequently used to determine nuclide production and fuel burn up in a nuclear reactor. This work sought to explore the effectiveness of using Monte-Carlo simulations to perform this same task. Specifically, the several reactor designs were run using MCNPX to calculate fuel burnup and fission products, as well as their respective radiogenic daughters.

This work compared nuclide production using ORIGEN and MCNPX calculations in three reactor types: the Westinghouse 17 x 17 pressurized water reactor, the General Electric 8 x 8 boiling water reactor, and the CANDU-37 reactor. Further, a simple sensitivity study was performed using the boiling water reactor to determine what effects small perturbations had on the isotopic ratios calculated.

For this task, nuclide production during fuel irradiation was determined using pre-existing reactor configurations for a simple boiling water reactor, a pressurized water reactor and a CANDU reactor. Nuclide production calculations were made after a fuel burnup of 1 month and 18 months. The cycle ratio, R_C, determined for a number of nuclide pairs and is defined by equation 1 above.

Nuclide production was then determined using probabilistic Monte Carlo simulations, as opposed to the deterministic ORIGEN simulations also performed. This was achieved using the BURN card in MCNPX. MCNP input decks of the same three reactor types used in the ORIGEN calculations: the CANDU-37, the GE 8 x8 BWR, and the Westinghouse PWR, were used in this work. Cycle ratios were then calculated using the output of the MCNPX burnup calculations and then compared to the cycle ratios determined using the ORIGEN code.

A sensitivity study on what effects small changes in reactor parameters had on the isotope ratios was performed. The sensitivity measurements were conducted using MCNP on the boiling water reactor. Changes in power level, burn time, and initial boron concentrations, were investigated.

In order to minimize discrepancies between the MCNP nuclide production output and the results of the ORIGEN runs, efforts were made to ensure that the reactor parameters used for both the MCNPX and ORIGEN input files were the same. The following tables, tables 3-8, outline the inputted parameters used for the ORIGEN portion in this investigation. The fuel mass, type, enrichment, and also the moderator density were extracted from the respective MCNPX input decks. The parameters stipulated in the BURN card of the MCNPX input deck also matched the burnup and reactor power levels inputted into the ORIGEN run. Figure 1 illustrates the MCNPX fuel assembly models for the BWR, PWR, and CANDU models.

Table 3: Input parameters for ORIGEN simulation of BWR following 1-month burnup

FUEL TYPE	GE 8x8-4
U INITIAL MASS (g)	173,557
ENRICHMENT	3.23 w/o ^{235}U
BURNUP	2952.34 MWd/MTU
CYCLES	1
LIBRARIES	1
COOLING TIME	1 second
MODERATOR DENSITY	0.6 g/cm^3
POWER	17.08 MW
AVERAGE POWER	98.411 MW/MTU

Table 4: Input parameters for ORIGEN simulation of BWR following 18-month burnup

FUEL TYPE	GE 8x8-4
U INITIAL MASS (g)	173,557
ENRICHMENT	3.23 w/o ^{235}U
BURNUP	53142.20 MWd/MTU
CYCLES	1
LIBRARIES	1
COOLING TIME	1 second
MODERATOR DENSITY	0.6 g/cm^3
POWER	17.08 MW
AVERAGE POWER	98.411 MW/MTU

Table 5: Input parameters for ORIGEN simulation of PWR following 1-month burnup

FUEL TYPE	Westinghouse 17x17
U INITIAL MASS (g)	450,030
ENRICHMENT	4.5 w/o ^{235}U
BURNUP	1138.59 MWd/MTU
CYCLES	1
LIBRARIES	1
COOLING TIME	1 second
MODERATOR DENSITY	0.723 g/cm^3
POWER	17.08 MW
AVERAGE POWER	37.953 MW/MTU

Table 6: Input parameters for ORIGEN simulation of PWR following 18-month burnup

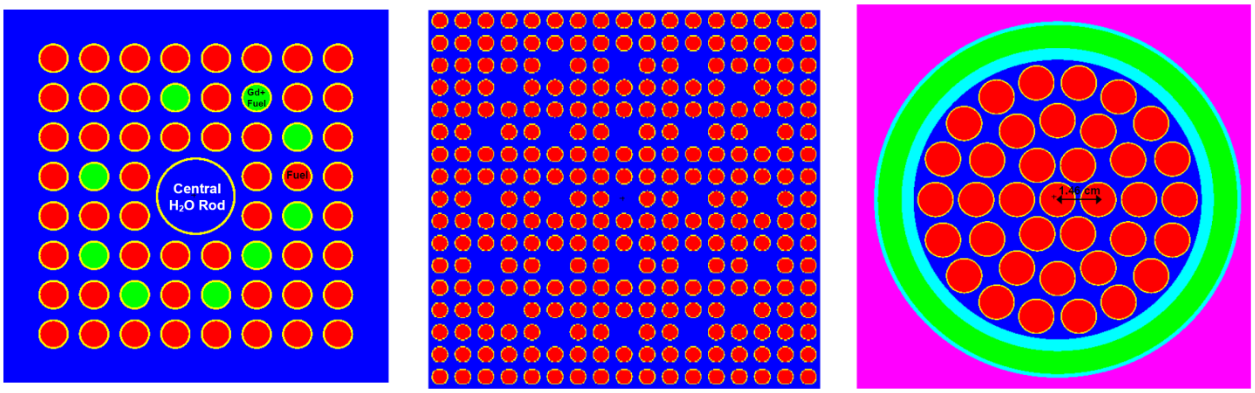
FUEL TYPE	Westinghouse 17x17
U INITIAL MASS (g)	450,030
ENRICHMENT	4.5 w/o ^{235}U
BURNUP	20494.63 MWd/MTU
CYCLES	1
LIBRARIES	1
COOLING TIME	1 second
MODERATOR DENSITY	0.723 g/cm^3
POWER	17.08 MW
AVERAGE POWER	37.953 MW/MTU

Table 7: Input parameters for ORIGEN simulation of CANDU reactor following 1-month burnup

FUEL TYPE	CANDU-37
U INITIAL MASS (g)	19,832
ENRICHMENT	Natural
BURNUP	756.35 MWd/MTU
CYCLES	1
LIBRARIES	1
COOLING TIME	1 second
MODERATOR DENSITY	0.8121 g/cm^3
POWER	0.5 MW
AVERAGE POWER	2.88 MW/MTU

Table 8: Input parameters for ORIGEN simulation of CANDU reactor following 1-month burnup

FUEL TYPE	CANDU-37
U INITIAL MASS (g)	19,832
ENRICHMENT	Natural
BURNUP	1555.68 MWd/MTU
CYCLES	1
LIBRARIES	1
COOLING TIME	1 second
MODERATOR DENSITY	0.8121 g/cm^3
POWER	0.5 MW
AVERAGE POWER	2.88 MW/MTU



BWR – 8x8 PWR – Westinghouse 17x17 CANDU - 37

Figure 1. MCNPX models for BWR, PWR, and CANDU fuel assemblies.

The first goal of this work was to determine whether R values, as described above, have reasonable agreement for the radionuclides of interest shown in Tables 1 and 2 of this report. In addition, a sensitivity study was conducted to investigate how the R values change over a range of operational conditions. It was shown that the R values compare between the MCNPX and the ORIGEN codes. In addition, the largest operational variable that appears to affect the R values was the boron concentration in the moderator of the PWR.

Task 4.2: Develop fuel cycle models that predict forensic signatures from known generic fuel cycles

To identify isotopic ratios that could be used to differentiate short (low burnup) from long (high burnup) fuel cycles, ORIGEN-ARP within SCALE 6 was used to model several fuel types and generate expected nuclide compositions that would arise from a low and a high burnup case.

Typical enrichments, specific power, and typical burnup values for commercial applications were gathered from available sources.[1,2] From the range of typical burnup values upon discharge, representative values for a low and a high burnup case were estimated. Additionally, some simplifications to the sample parameters were made. For example, the specific power, were different values were used for the BWR and PWR classes rather than each different assembly type. All simulations were performed using “Express Mode.” The sample space is summarized in Table 9.

Table 9. Summary of Samples simulated in ORIGEN-ARP. Note, the names of the fuel types follow the convention/format of ORIGEN-ARP

Fuel Type	Reactor Type	Specific Power [MW/MTU]	Initial Enrichment [w/o]	Burnup Values [MWd/kgU]
Abb8x8-1	BWR	23	2.9	1, 27
Atrium9-9	BWR	23	2.9	1, 27
Atrium10-9	BWR	23	2.9	1, 27
CE14x14	PWR	32	2.8	1, 32
CE16x16	PWR	32	2.8	1, 32
GE7x7-0	BWR	23	2.9	1, 27
GE8x8-4	BWR	23	2.9	1, 27
GE9x9-7	BWR	23	2.9	1, 27
GE10x10-8	BWR	23	2.9	1, 27
S14x14	PWR	32	2.8	1, 32
Svea64-1	BWR	23	2.85	1, 27
Svea100-0	BWR	23	2.85	1, 27
Vver440(3.6)	PWR	32	3.60	1, 32
Vver440(3.82)	PWR	32	3.82	1, 32
Vver440(4.25)	PWR	32	4.25	1, 32
Vver440(4.38)	PWR	32	4.38	1, 32
W14x14	PWR	32	2.8	1, 32
W15x15	PWR	32	2.8	1, 32
W17x17	PWR	32	2.8	1, 32

After simulation, the output files, which each contained the top 200 nuclides (by mass at discharge) were each processed to compute all isotopic ratios of the form

${}^{A}_{Z}X \Big/ {}^{A'}_{Z}X$, where $A \neq A'$. Then, these ratios as defined in the statement of work were compared between the low and high burnup cases for each fuel to find R-values:

[1] Knief, Ronald Allen, *Nuclear Engineering: Theory and Technology of Commercial Nuclear Power,* 2nd Ed., American Nuclear Society, IL: 2008

[2] http://wp.ornl.gov/sci/scale/pubs/152495.pdf

$$R = {}^{R_{low}}\big/_{R_{high}}$$

Lastly, R-values greater than 100 or less that 0.01, indicating two orders of magnitude change in the isotopic ratio between the two burnup values, were sought. Isotopic ratios with such R-values are identified as being good candidates for differentiating a low and high burnup fuel cycle. These identified R-values are shown in Table 10.

Task 4.3: Inverse calculations to assess forensic capabilities from test data sets

This task is comprised of three major steps: simulate fuel cycles and assemble nuclide inventories into a database, write an algorithm to compare test cases against entries in the database, assigning a figure of merit to convey their similarity and determine the best-fit entry of the database, and simulate test cases to test the identification methodology's ability to identify a known fuel cycle that best matches the unknown.

Creating the database

To run the large number of samples required for the assembly of the desired database, a MATLAB function was written that could take in parameters such as file name, fuel type, enrichment, cooling time, output units, etc. and write an ORIGEN-S input file (.inp) reflecting these parameters. Additionally, a script was written to call this function for every point in the sample space and write the required batch file to run all of the generated input files. These functions are "makeOrigenInp.m" and "makeInpsAndBat.m" (see Appendix). These files were then run with ORIGEN-S within SCALE 6. All samples were simulated at four different cooling times: 1 minute, 1 day, 30 days, and 1 year. Table 11 summarizes the sample space that was simulated. Note, the MATLAB notation for an array of values is used for denoting the burnup values: 600:690:13000 denotes the set from 600 to 13000 in steps of 690. The step sizes were chosen to correspond to roughly 1 month of operation. As in Task 4.2, these parameters were estimated from available literature to simulate some of the major differences between these different reactor/assembly types, while keeping the simulations simple.

Table 10. Summary of identified isotopic ratios useful for differentiation between low and high burnup cases. Marked are ratios having R-values larger than 100 or less than 0.01 for a particular fuel type (continued on next page).

	G E1 0	G E 9	G E7	Atrium 10	Atrium 9	G E8	Abb 8	Svea 100	Svea 64	W 17	W 15	W 14	Vver 3.82	S 14	Vver 3.6	Vver 4.38	Vver 4.25	CE 16	CE 14
Pu239/Pu241	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Pu239/Pu242	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Pu239/Pu238	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Pu240/Pu242	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Nd143/Nd142	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Nd145/Nd142	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Nd146/Nd142	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Nd148/Nd142	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Nd150/Nd142	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Mo100/Mo96	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Mo98/Mo96	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Mo97/Mo96	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Mo95/Mo96	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Ba134/Ba140		X	X	X	X	X	X			X	X	X	X	X	X	X	X	X	X
Ba136/Ba140		X	X	X	X	X	X			X	X	X	X	X	X	X	X	X	X
Sr89/Sr86		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Sr89/Sr87		X	X	X	X	X	X	X	X	X	X	X		X				X	X
Te127m/Te122		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Eu154/Eu151		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Gd157/Gd152		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Cd110/Cd113m		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Nd144/Nd142			X			X		X	X	X	X	X	X	X	X	X	X	X	X
Xe134/Xe128								X	X					X					X
Xe131/Xe128								X	X	X	X	X	X	X	X	X	X	X	X
Te127m/Te124								X	X	X	X	X	X	X	X	X	X	X	X

Cd110/Cd115m										X	X	X	X	X	X	X	X	X	X
Am241/Am243	X															X			
Gd156/Gd157		X	X	X	X	X	X	X	X							X	X		
Xe136/Xe128								X	X										
Ba138/Ba134								X	X										
Cd111/Cd113								X	X										
Eu151/Eu152																X	X		
Gd155/Gd152																X	X		

Table 11. Summary of samples simulated in ORIGEN-ARP. Note the names of the fuel types follow the convention/format of ORIGEN-ARP.

Fuel Type	Reactor Type	Specific Power [MW/MTU]	Initial Enrichment [w/o]	Burnup Values [MWd/kgU]
Abb8x8-1	BWR	23	2.9	600:690:13000
Atrium9-9	BWR	23	2.9	600:690:13000
Atrium10-9	BWR	23	2.9	600:690:13000
CANDU28	PHWR	22	0.711	600:690:13000[3]
CANDU37	PHWR	22	0.711	600:690:13000
CE14x14	PWR	32	2.8	600:950:18000
CE16x16	PWR	32	2.8	600:950:18000
GE7x7-0	BWR	23	2.9	600:690:13000
GE8x8-4	BWR	23	2.9	600:690:13000
GE9x9-7	BWR	23	2.9	600:690:13000
GE10x10-8	BWR	23	2.9	600:690:13000
S14x14	PWR	32	2.8	600:950:18000
Svea64-1	BWR	23	2.85	600:690:13000
Svea100-0	BWR	23	2.85	600:690:13000
Vver440(3.6)	PWR	32	3.60	600:950:18000
Vver440(3.82)	PWR	32	3.82	600:950:18000
Vver440(4.25)	PWR	32	4.25	600:950:18000
Vver440(4.38)	PWR	32	4.38	600:950:18000
Vver1000	PWR	32	2.8	600:950:18000
W14x14	PWR	32	2.8	600:950:18000
W15x15	PWR	32	2.8	600:950:18000
W17x17	PWR	32	2.8	600:950:18000

To form the database, a master list of nuclides present in the 1628 samples was formed. This list includes 308 nuclides. A 308 x 1628 matrix, D, was then created, where each row corresponds to a nuclide in the master list and each column a sample. The matrix was populated as follows:

$$D_{ij} \equiv \textit{Mass of Nuclide i in Sample j}$$

where the nuclide index *i* is taken from the master list. If sample *j* did not have nuclide *i* in its top-200 list, a zero was placed in that entry of D (see “makeDTRAdatabase.m” in Appendix).

[3] These burnup values are too high for CANDU reactors, but for simplicity, the same values as used for other reactors with a similar power density were used. The effects of heavy-water moderation, cross sections, etc. are still simulated.

Writing an identification algorithm

To identify an unknown fuel cycle from its top-200 nuclide list, an algorithm was written that sorts the unknown cycle's list to agree with the master nuclide list described above (inserting zeros as appropriate), then computes a figure of merit (FOM) to describe the similarity between the resulting column vector and the 1628 columns of D. This was done by three methods:

$$FOM_1 = \sum_{n=1}^{308} (x_n - \bar{x}_n)^2$$

$$FOM_2 = \sum_{n=1}^{308} |x_n - \bar{x}_n|$$

$$FOM_3 = \sum_{n=1}^{308} \frac{|x_n - \bar{x}_n|}{\bar{x}_n}$$

Here $\bar{x}_n$ is the *nth* entry of the database matrix for a particular sample and x_n is the *nth* entry of the unknown sorted nuclide list. These three FOM's each have a motivation: 1 is standard least-squares approach that is commonly used in many "best fit" applications, 2 adjusts the approach of 1, which may over penalize a dissimilar entry, and three adjusts this second approach to examine the relative difference in each entry in order to more fairly weight the contributions to the FOM from low concentration nuclides. In practice, an FOM is generated for each of the 1628 columns of D, and the fuel cycle (fuel type, cooling time, and burnup) with the smallest entry in the resulting vector is found to be the best fit. Additionally, an estimate of the burnup of the unknown is made by doing a linear interpolation between two samples nearest to the best-fit entry in D (which was organized to place column vectors of nuclide inventories of simulations of the same reactor and cooling time next to each other in order of burnup). The FOM was used as the notion of distance (see "IDfuelCycle.m" in Appendix).

To verify the quality of the code, samples from the database were put into the identification algorithm in order to verify that the correct column was returned as the best fit and the corresponding FOM value was 0 as would be expected. This was done using "checkDatabase.m." In this process, it was realized that the third FOM was ill-defined (the inclusion of zeros in the database causes the FOM column vectors to have undefined entries). Rather than change the database formulation, this third FOM option was not further used.

Testing the method

To test the method, a test set of ORIGEN simulations was performed using the aforementioned MATLAB routines. These test cases featured burnup values that were not exactly represented in the database, i.e., falling in between burnup values of sample populating the database. Similar perturbations were also made on the enrichment of the fuel, cooling time of the fuel, and in the various combinations of these three variables. Table 12 summarizes. These samples were fed into the identification function for comparison against the database matrix and the results analyzed.

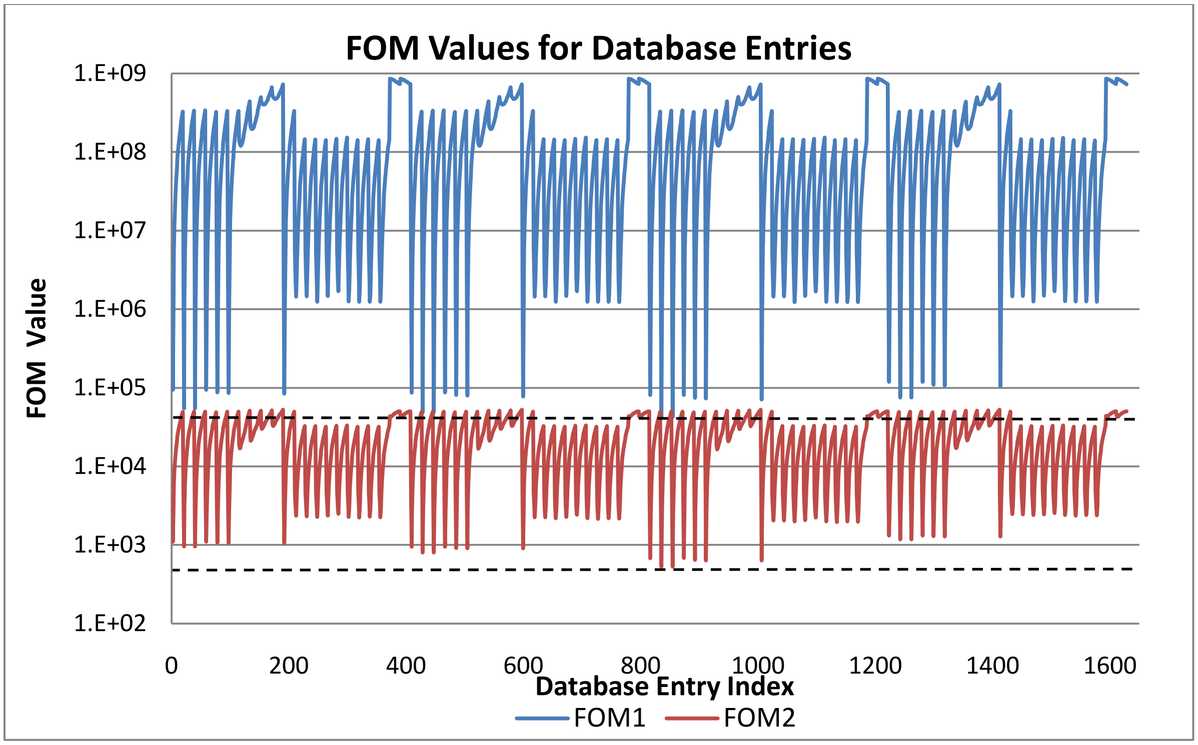
Table 12. Summary of the test cases used for testing the identification algorithm.

Fuel	Enrichment [w/o]	Cooling Time	Burnup [MWd/MTU]
CANDU28	0.71	1m, 7d, 30d, 1a	1400, 5000, 11000
CANDU28	0.71	3m, 9d, 2a	6120, 5000
CE 16x16	2.8	1m, 7d, 30d, 1a	1700, 8700, 17000
CE 16x16	2.8	3m, 9d, 2a	8700, 9150
CE 16x16	3.1	7d, 9d	8700, 9150
GE 7x7-0	2.9	1m, 7d, 30d, 1a	2000, 7200, 10800
GE 7x7-0	2.9	3m, 9d, 2a	7200, 8880
GE 7x7-0	3.2	7d, 9d	7200, 8880

The function “runTestCases.m” performed these tests and recorded the results. The names of the samples were parsed to find the parameters used in the simulation, and these parameters (burnup, enrichment, cooling time) were examined to determine which were perturbed versus calibrated values. The function then passed one of the test cases into the identification algorithm. Relative errors in the estimation of burnup, both from the best-fit entry of the database matrix and from the linear interpolation method, were calculated for each test case. Lastly, successes in the estimation of the fuel assembly type and cooling time were recorded. Successes in the estimation of the assembly type were defined as getting the type exactly correct, i.e., guessing that a test sample was a CE 16x16 rather than CE 14x14 was not considered a success. For a cooling time estimation to be declared successful, it had to be the closest possible guess (of which the algorithm can only make 4: 1 minute, 1 day, 30 days, and 1 year) to the actual cooling time of the unknown.

Figure 1 shows the values of FOM_1 and FOM_2 for different entries in the database, generated for one of the test cases. In this example, the simulated fuel assembly was a CE 16x16 type, with an initial enrichment of 2.8 w/o, a burnup value of 1700 MWd/MTU, and a cooling time of one year. The algorithm is able to correctly identify the type of reactor (see Figure 2.a) and then the nearest burnup value represented in the database (see Figure 2.b). Both of these identifications are indicated by a minimum in the FOM values. Note, while these FOMs are discrete functions, they are displayed with a line for ease of reading the figures. The interpolated burnup value produced by the second FOM produces a better estimate of the burnup of the test case versus the best-fit entry from the database, with relative errors of 3.24% and 8.82%, respectively.

These and the rest of the results for test cases that only altered the burnup are given in Table 12. The complete results, as described above, became very difficult to examine, and further work is needed to fully understand underlying trends, sensitivities of the method to the different sources of variance introduced in this study, and to alter the method to better deal with these additional sources of variance. The complete results may be found in Tables A.1 and A.2 within the Appendix.



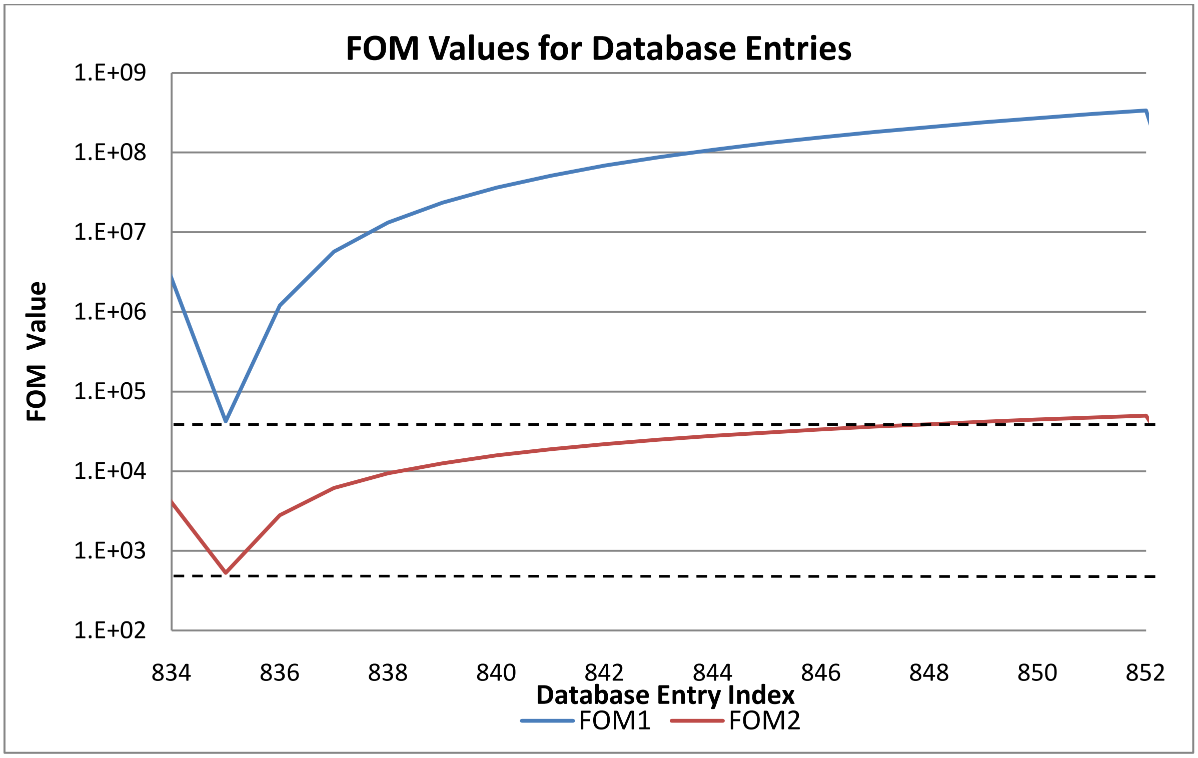


Figure 2. Values of FOM for entries in the database. **1.a:** (Top) The area around the minimum (black line) identifies the reactor type and cooling time. **1.b:** (Bottom) Shows the FOM values for entries in the database corresponding to CE 16x16 with 1 year cooling time and minimum vale for the two FOM's, identifying the best guess for the burnup of the test case.

Table 7. Shows the results of test cases where only burnup was altered. The results shown here were obtained with the least squares value figure of merit. The case corresponding to Figure 1 is highlighted. (Next page) Shows the same test case results as obtained using the absolute value figure of merit.

Reactor Type Guess	Simulated Reactor	Success	Best-Fit Burnup [MWd/MTU]	Error [%]	Interpolated Burnup [MWd/MTU]	Error [%]	Simulated Burnup [MWd/MTU]	Cooling Time Guess	Simulated Cooling Time	Success
CANDU28	CANDU 28	X	1290	7.86	1552	10.86	1400	1a	1a	X
CANDU 37	CANDU 28		1290	7.86	1522	8.71	1400	7d	1m	
CANDU 28	CANDU 28	X	1290	7.86	1547	10.50	1400	30d	7d	
CANDU 28	CANDU 28	X	1290	7.86	1552	10.86	1400	30d	30d	X
CANDU 37	CANDU 28		4740	5.20	5318	6.36	5000	1a	1a	X
CANDU 37	CANDU 28		4740	5.20	5301	6.02	5000	7d	1m	
CANDU 37	CANDU 28		4740	5.20	5313	6.26	5000	30d	7d	
CANDU37	CANDU 28		4740	5.20	5318	6.36	5000	30d	30d	X
CANDU 28	CANDU 28	X	10950	0.45	11046	0.42	11000	1a	1a	X
CANDU 28	CANDU 28	X	10950	0.45	11047	0.43	11000	1m	1m	X
CANDU28	CANDU 28	X	10950	0.45	11047	0.43	11000	7d	7d	X
CANDU 28	CANDU 28	X	10950	0.45	11047	0.43	11000	30d	30d	X
CE 16x16	**CE**	**X**	**1550**	**8.82**	**1909**	**12.29**	**1700**	**1a**	**1a**	**X**

	16x16									
CE 16x16	CE 16x16	X	1550	8.82	1889	11.12	1700	7d	1m	
CE 16x16	CE 16x16	X	1550	8.82	1905	12.06	1700	30d	7d	
CE 16x16	CE 16x16	X	1550	8.82	1909	12.29	1700	30d	30d	X
VVER 1000	CE 16x16		9150	5.17	8307	4.52	8700	1a	1a	X
VVER 1000	CE 16x16		9150	5.17	8317	4.40	8700	1m	1m	X
VVER 1000	CE 16x16		9150	5.17	8334	4.21	8700	1m	7d	
VVER 1000	CE 16x16		9150	5.17	8309	4.49	8700	30d	30d	X
W 14x14	CE 16x16		16750	1.47	17198	1.16	17000	1a	1a	X
W 14x14	CE 16x16		16750	1.47	17206	1.21	17000	1m	1m	X
W 14x14	CE 16x16		16750	1.47	17196	1.15	17000	30d	7d	
W 14x14	CE 16x16		16750	1.47	17201	1.18	17000	30d	30d	X
GE 7x7-0	GE 7x7-0	X	1980	1.00	2062	3.10	2000	1a	1a	X
GE 7x7-0	GE 7x7-0	X	1980	1.00	2054	2.70	2000	1m	1m	X
GE 7x7-0	GE 7x7-0	X	1980	1.00	2060	3.00	2000	7d	7d	X
GE 7x7-0	GE 7x7-0	X	1980	1.00	2061	3.05	2000	30d	30d	X
GE 8x8-4	GE 7x7-0		7500	4.17	6922	3.86	7200	1a	1a	X

GE 8x8-4	GE 7x7-0		7500	4.17	6913	3.99	7200	1m	1m	X
GE 8x8-4	GE 7x7-0		7500	4.17	6932	3.72	7200	1m	7d	
GE 8x8-4	GE 7x7-0		7500	4.17	6938	3.64	7200	1m	30d	
GE 7x7-0	GE 7x7-0	X	10950	1.39	10666	1.24	10800	1a	1a	X
GE 7x7-0	GE 7x7-0	X	10950	1.39	10658	1.31	10800	1m	1m	X
GE 7x7-0	GE 7x7-0	X	10950	1.39	10662	1.28	10800	7d	7d	X
GE 7x7-0	GE 7x7-0	X	10950	1.39	10669	1.21	10800	7d	30d	

Reactor Type Guess	Simulated Reactor	Success	Best-Fit Burnup [MWd/MTU]	Error [%]	Interpolated Burnup [MWd/MTU]	Error [%]	Simulated Burnup [MWd/MTU]	Cooling Time Guess	Simulated Cooling Time	Success
CANDU 28	CANDU 28	X	1290	7.86	1441	2.93	1400	1a	1a	X
CANDU 28	CANDU 28	X	1290	7.86	1411	0.79	1400	7d	1m	
CANDU 28	CANDU 28	X	1290	7.86	1442	3.00	1400	7d	7d	X
CANDU 28	CANDU 28	X	1290	7.86	1441	2.93	1400	30d	30d	X
CANDU 28	CANDU 28	X	4740	5.20	5188	3.76	5000	1a	1a	X
CANDU 37	CANDU 28		4740	5.20	5127	2.54	5000	7d	1m	
CANDU 28	CANDU 28	X	4740	5.20	5194	0.88	5000	7d	7d	X
CANDU	CANDU	X	4740	5.20	5192	3.84	5000	30d	30d	X

28	28									
CANDU 28	CANDU 28	X	10950	0.45	11006	0.05	11000	1a	1a	X
CANDU 28	CANDU 28	X	10950	0.45	11014	0.13	11000	1m	1m	X
CANDU 28	CANDU 28	X	10950	0.45	11012	0.11	11000	7d	7d	X
CANDU 28	CANDU 28	X	10950	0.45	11010	0.09	11000	30d	30d	X
CE 16x16	**CE 16x16**	**X**	**1550**	**8.82**	**1755**	**3.24**	**1700**	**1a**	**1a**	**X**
CE 16x16	CE 16x16	X	1550	8.82	1733	1.94	1700	7d	1m	
CE 16x16	CE 16x16	X	1550	8.82	1743	2.53	1700	30d	7d	
CE 16x16	CE 16x16	X	1550	8.82	1756	3.29	1700	30d	30d	X
CE 14x14	CE 16x16		9150	5.17	8634	0.76	8700	1a	1a	X
CE 14x14	CE 16x16		9150	5.17	8647	0.61	8700	7d	1m	
CE 14x14	CE 16x16		9150	5.17	8613	1.00	8700	30d	7d	
CE 14x14	CE 16x16		9150	5.17	8674	0.30	8700	30d	30d	X
CE 16x16	CE 16x16	X	16750	1.47	17098	0.58	17000	1a	1a	X
CE 16x16	CE 16x16	X	16750	1.47	17042	0.25	17000	30d	1m	
CE 16x16	CE 16x16	X	16750	1.47	17084	0.49	17000	30d	7d	
CE 16x16	CE 16x16	X	16750	1.47	17129	0.76	17000	30d	30d	X

GE 7x7-0	GE 7x7-0	X	1980	1.00	2018	0.90	2000	1a	1a	X
GE 7x7-0	GE 7x7-0	X	1980	1.00	2005	0.25	2000	1m	1m	X
GE 7x7-0	GE 7x7-0	X	1980	1.00	2012	0.60	2000	7d	7d	X
GE 7x7-0	GE 7x7-0	X	1980	1.00	2016	0.80	2000	30d	30d	X
Atrium 10-9	GE 7x7-0		7500	4.17	7004	2.72	7200	1a	1a	X
GE 9x9-7	GE 7x7-0		6810	5.42	7306	1.47	7200	1m	1m	X
Atrium 10-9	GE 7x7-0		7500	4.17	7023	2.46	7200	1m	7d	
Atrium 10-9	GE 7x7-0		7500	4.17	7036	2.28	7200	7d	30d	
GE 7x7-0	GE 7x7-0	X	10950	1.39	10772	0.26	10800	1a	1a	X
GE 7x7-0	GE 7x7-0	X	10950	1.39	10714	0.80	10800	1m	1m	X
GE 7x7-0	GE 7x7-0	X	10950	1.39	10770	0.28	10800	1m	7d	
GE 7x7-0	GE 7x7-0	X	10950	1.39	10783	0.16	10800	7d	30d	

Some Conclusions

The second figure of merit (absolute value) was more successful in identifying the reactor type, with 27 successes versus 19 successful identifications made by the first FOM, while the first FOM performed slightly better at predicting cooling time, the first and second FOM's yielding 25 and 24 successful cooling time guesses respectively. The average relative errors in predicting the burnup of the test cases using the best-fit entry from the database was similar for the two FOM's, the least squares averaging 3.95% and the absolute value averaging 3.99% relative error. The error in this burnup prediction is a function of the difference between the burnup values simulated for the formation of the database. Unsurprisingly, the absolute value FOM performed much better predicting the burnup of the test cases via linear interpolation, yielding an average relative error in prediction of 1.38% versus the 4.72% average produced using the first figure of merit. Thus, overall the second figure of merit performed better.

As the full set of test cases and results becomes much more complex to analyze, further work is suggested to more fully understand the performance of the method outlined in this work and develop ways to overcome its shortcomings. Possible solutions could include using different figures of merit and/or combinations of different FOM's for the various parts of the analysis. Optimization of the database, e.g., including more simulations or concentrating database entries at points in the sample space (reactor type, burnup values, enrichment, etc.) where the output is most sensitive to changes in these input parameters, could also improve the ability of the method to make accurate predictions. Similarly, the sensitivity of each nuclide in the output files relative to these input parameters could be studied and this knowledge used to optimize the method. Lastly, such a sensitivity study could also be used to make estimates of the uncertainty in the various predictions made.

In summation, the present work demonstrates the efficacy of a simple method of analyzing more of the data available from measurements of spent nuclear fuel for the purposes of fuel cycle identification, and further work is warranted to develop such methods as a complement to currently-employed methods that rely on data reduction via the use of isotopic ratios.

Task 4.4: Develop chemistry for analysis of samarium.

The chemical form of samarium required for analysis varies for different mass spectrometry techniques. Inductively Coupled Plasma Mass Spectrometry (ICPMS) requires a liquid sample for introduction into the plasma. Accelerator mass spectrometry requires solid targets for negative ion production. We previously examined ion production from samarium oxide (Sm_2O_3) and samarium fluoride (SmF_3) in the LLNL ion source. The fluoride produced an order of magnitude more negative ions and was selected as the best option for samarium AMS sample material. Samarium is readily soluble in nitric acid and commercial elemental liquid standards contain nitric acid. Liquid Sm standards are generally produced by dissolving high purity solid Sm_2O_3 in high purity nitric acid.

Production of SmF_3 for AMS samples was investigated starting with a Sm liquid standard. This material does not have elemental interferences and is a good starting material for developing sample chemistry. The LLNL AMS facility uses the fluorides of Ca and Sr currently, so we have lab facilities suitable for working with hydrofluoric acid. The general procedure for making SmF_3 is as follows: Start with Sm solution, e.g. 4 mL solution containing 2 mg Sm in 5-10% nitric acid. Add concentrated ammonium hydroxide but keep solution acidic. Add 3 mL 40% hydrofluoric acid and allow samarium fluoride to precipitate overnight. Centrifuge to form a pellet and remove the supernatant. Add 1 mL of dilute hydrofluoric acid, then transfer samarium fluoride and rinse to a new 2 mL centrifuge tube. Centrifuge again to form a pellet, remove the supernatant, and dry overnight on a heating block. Place dry SmF_3 in a watertight vial until ready to use.

Samarium fluoride can be precipitated by adding a soluble fluoride salt such as ammonium fluoride rather than hydrofluoric acid. Ammonium fluoride has traces of hydrofluoric acid , however, so the same safety procedures are used. Hydrofluoric acid has been shown to work better when solutions are less pure (e.g., Ca), however, so we decided to stick with it.

Task 4.5: Conduct presentations/meetings at times and places specified in the contract schedule.

A presentation was delivered on July 22, 2011 in Springfield, VA at the DTRA program review.

Task: 4.6: Write Option Year 1 report

It is written and submitted by the September 1, 2011 deadline.

Appendix

Table A1. Results of the analysis of the various test cases using FOM_1

Different Enrichment	Different Burnup	Different Cooling Time	Reactor Type Guess	Simulated Reactor	Success	Best-fit Burnup [MWd/MTU]	Error [%]	Interpolated Burnup [MWd/MTU]	Error [%]	Simulated Burnup [MWd/MTU]	Cooling Time Guess	Simulated Cooling Time	Success
	1		'candu28'	'CANDU28'	1	1290	7.86%	1552	10.86%	1400	'1years'	'1years'	1
	1		'candu37'	'CANDU28'		1290	7.86%	1522	8.71%	1400	'7days'	'1minutes'	
	1		'candu28'	'CANDU28'	1	1290	7.86%	1547	10.50%	1400	'30days'	'7days'	
	1		'candu28'	'CANDU28'	1	1290	7.86%	1552	10.86%	1400	'30days'	'30days'	1
	1		'candu37'	'CANDU28'		4740	5.20%	5318	6.36%	5000	'1years'	'1years'	1
	1		'candu37'	'CANDU28'		4740	5.20%	5301	6.02%	5000	'7days'	'1minutes'	
	1		'candu37'	'CANDU28'		4740	5.20%	5313	6.26%	5000	'30days'	'7days'	
	1		'candu37'	'CANDU28'		4740	5.20%	5318	6.36%	5000	'30days'	'30days'	1
	1		'candu28'	'CANDU28'	1	10950	0.45%	11046	0.42%	11000	'1years'	'1years'	1
	1		'candu28'	'CANDU28'	1	10950	0.45%	11047	0.43%	11000	'1minutes'	'1minutes'	1
	1		'candu28'	'CANDU28'	1	10950	0.45%	11047	0.43%	11000	'7days'	'7days'	1
	1		'candu28'	'CANDU28'	1	10950	0.45%	11047	0.43%	11000	'30days'	'30days'	1
		1	'candu37'	'CANDU28'		6120	0.00%	6129	0.15%	6120	'1years'	'2years'	1
		1	'candu37'	'CANDU28'		6120	0.00%	6129	0.15%	6120	'1minutes'	'3minutes'	1
		1	'candu37'	'CANDU28'		6120	0.00%	6130	0.16%	6120	'7days'	'9days'	1
	1	1	'candu37'	'CANDU28'		4740	5.20%	5318	6.36%	5000	'1years'	'2years'	1
	1	1	'candu37'	'CANDU28'		4740	5.20%	5302	6.04%	5000	'7days'	'3minutes'	
	1	1	'candu37'	'CANDU28'		4740	5.20%	5315	6.30%	5000	'30days'	'9days'	
	1		'ce16x16'	'CE16'	1	1550	8.82%	1909	12.29%	1700	'1years'	'1years'	1
	1		'ce16x16'	'CE16'	1	1550	8.82%	1889	11.12%	1700	'7days'	'1minutes'	
	1		'ce16x16'	'CE16'	1	1550	8.82%	1905	12.06%	1700	'30days'	'7days'	
	1		'ce16x16'	'CE16'	1	1550	8.82%	1909	12.29%	1700	'30days'	'30days'	1
	1		'vver1000'	'CE16'		9150	5.17%	8307	4.52%	8700	'1years'	'1years'	1
	1		'vver1000'	'CE16'		9150	5.17%	8317	4.40%	8700	'1minutes'	'1minutes'	1
	1		'vver1000'	'CE16'		9150	5.17%	8334	4.21%	8700	'1minutes'	'7days'	
	1		'vver1000'	'CE16'		9150	5.17%	8309	4.49%	8700	'30days'	'30days'	1
	1		'w14x14'	'CE16'		16750	1.47%	17198	1.16%	17000	'1years'	'1years'	1
	1		'w14x14'	'CE16'		16750	1.47%	17206	1.21%	17000	'1minutes'	'1minutes'	1
	1		'w14x14'	'CE16'		16750	1.47%	17196	1.15%	17000	'30days'	'7days'	
	1		'w14x14'	'CE16'		16750	1.47%	17201	1.18%	17000	'30days'	'30days'	1
	1	1	'vver1000'	'CE16'		9150	5.17%	8308	4.51%	8700	'1years'	'2years'	1
	1	1	'vver1000'	'CE16'		9150	5.17%	8317	4.40%	8700	'1minutes'	'3minutes'	1
	1	1	'vver1000'	'CE16'		9150	5.17%	8313	4.45%	8700	'7days'	'9days'	1
		1	'ce16x16'	'CE16'	1	9150	0.00%	9154	0.04%	9150	'1years'	'2years'	1
		1	'ce16x16'	'CE16'	1	9150	0.00%	9164	0.15%	9150	'1minutes'	'3minutes'	1
		1	'ce16x16'	'CE16'	1	9150	0.00%	9152	0.02%	9150	'30days'	'9days'	
1	1		'ge10'	'CE16'		8880	2.07%	8467	2.68%	8700	'7days'	'7days'	1
1	1	1	'ge10'	'CE16'		8880	2.07%	8464	2.71%	8700	'30days'	'9days'	
1			'ge10'	'CE16'		8880	2.95%	9125	0.27%	9150	'30days'	'7days'	
1		1	'ge10'	'CE16'		8880	2.95%	9126	0.26%	9150	'30days'	'9days'	
	1		'ge7'	'GE7'	1	1980	1.00%	2062	3.10%	2000	'1years'	'1years'	1
	1		'ge7'	'GE7'	1	1980	1.00%	2054	2.70%	2000	'1minutes'	'1minutes'	1
	1		'ge7'	'GE7'	1	1980	1.00%	2060	3.00%	2000	'7days'	'7days'	1
	1		'ge7'	'GE7'	1	1980	1.00%	2061	3.05%	2000	'30days'	'30days'	1
	1		'ge8'	'GE7'		7500	4.17%	6922	3.86%	7200	'1years'	'1years'	1
	1		'ge8'	'GE7'		7500	4.17%	6913	3.99%	7200	'1minutes'	'1minutes'	1
	1		'ge8'	'GE7'		7500	4.17%	6932	3.72%	7200	'1minutes'	'7days'	
	1		'ge8'	'GE7'		7500	4.17%	6938	3.64%	7200	'1minutes'	'30days'	
	1		'ge7'	'GE7'	1	10950	1.39%	10666	1.24%	10800	'1years'	'1years'	1
	1		'ge7'	'GE7'	1	10950	1.39%	10658	1.31%	10800	'1minutes'	'1minutes'	1
	1		'ge7'	'GE7'	1	10950	1.39%	10662	1.28%	10800	'7days'	'7days'	1
	1		'ge7'	'GE7'	1	10950	1.39%	10669	1.21%	10800	'7days'	'30days'	
		1	'ge7'	'GE7'	1	8880	0.00%	8910	0.34%	8880	'1years'	'2years'	1
		1	'ge7'	'GE7'	1	8880	0.00%	8900	0.23%	8880	'1minutes'	'3minutes'	1
		1	'ge7'	'GE7'	1	8880	0.00%	8906	0.29%	8880	'7days'	'9days'	1
	1	1	'ge8'	'GE7'		7500	4.17%	6923	3.85%	7200	'1years'	'2years'	1
	1	1	'ge8'	'GE7'		7500	4.17%	6913	3.99%	7200	'1minutes'	'3minutes'	1
	1	1	'ge8'	'GE7'		7500	4.17%	6933	3.71%	7200	'1minutes'	'9days'	
1			'ge10'	'GE7'		8190	7.77%	8252	7.07%	8880	'1minutes'	'7days'	
1		1	'ge10'	'GE7'		8190	7.77%	8255	7.04%	8880	'1minutes'	'9days'	
1	1		'ge10'	'GE7'		6120	15.00%	6637	7.82%	7200	'7days'	'7days'	1
1	1	1	'ge10'	'GE7'		6120	15.00%	6638	7.81%	7200	'7days'	'9days'	1

Table A2. Results of the analysis of the various test cases using FOM$_2$

Different Enrichment	Different Burnup	Different Cooling Time	Reactor Type Guess	Simulated Reactor	Success	Best-fit Burnup [MWd/MTU]	Error [%]	Interpolated Burnup [MWd/MTU]	Error [%]	Simulated Burnup [MWd/MTU]	Cooling Time Guess	Simulated Cooling Time	Success
	1		'candu28'	'CANDU28'	1	1290	7.86%	1441	2.93%	1400	'1years'	'1years'	1
	1		'candu28'	'CANDU28'	1	1290	7.86%	1411	0.79%	1400	'7days'	'1minutes'	
	1		'candu28'	'CANDU28'	1	1290	7.86%	1442	3.00%	1400	'7days'	'7days'	1
	1		'candu28'	'CANDU28'	1	1290	7.86%	1441	2.93%	1400	'30days'	'30days'	1
	1		'candu28'	'CANDU28'	1	4740	5.20%	5188	3.76%	5000	'1years'	'1years'	1
	1		'candu37'	'CANDU28'		4740	5.20%	5127	2.54%	5000	'7days'	'1minutes'	
	1		'candu28'	'CANDU28'	1	4740	5.20%	5194	3.88%	5000	'7days'	'7days'	1
	1		'candu28'	'CANDU28'	1	4740	5.20%	5192	3.84%	5000	'30days'	'30days'	1
	1		'candu28'	'CANDU28'	1	10950	0.45%	11006	0.05%	11000	'1years'	'1years'	1
	1		'candu28'	'CANDU28'	1	10950	0.45%	11014	0.13%	11000	'1minutes'	'1minutes'	1
	1		'candu28'	'CANDU28'	1	10950	0.45%	11012	0.11%	11000	'7days'	'7days'	1
	1		'candu28'	'CANDU28'	1	10950	0.45%	11010	0.09%	11000	'30days'	'30days'	1
		1	'candu37'	'CANDU28'		6120	0.00%	6142	0.36%	6120	'1years'	'2years'	1
		1	'candu37'	'CANDU28'		6120	0.00%	6140	0.33%	6120	'1minutes'	'3minutes'	1
		1	'candu37'	'CANDU28'		6120	0.00%	6144	0.39%	6120	'7days'	'9days'	1
	1	1	'candu28'	'CANDU28'	1	4740	5.20%	5192	3.84%	5000	'1years'	'2years'	1
	1	1	'candu37'	'CANDU28'		4740	5.20%	5127	2.54%	5000	'7days'	'3minutes'	
	1	1	'candu28'	'CANDU28'	1	4740	5.20%	5142	2.84%	5000	'30days'	'9days'	
	1		'ce16x16'	'CE16'	1	1550	8.82%	1755	3.24%	1700	'1years'	'1years'	1
	1		'ce16x16'	'CE16'	1	1550	8.82%	1733	1.94%	1700	'7days'	'1minutes'	
	1		'ce16x16'	'CE16'	1	1550	8.82%	1743	2.53%	1700	'30days'	'7days'	
	1		'ce16x16'	'CE16'	1	1550	8.82%	1756	3.29%	1700	'30days'	'30days'	1
	1		'ce14x14'	'CE16'		9150	5.17%	8634	0.76%	8700	'1years'	'1years'	1
	1		'ce14x14'	'CE16'		9150	5.17%	8647	0.61%	8700	'7days'	'1minutes'	
	1		'ce14x14'	'CE16'		9150	5.17%	8613	1.00%	8700	'30days'	'7days'	
	1		'ce14x14'	'CE16'		9150	5.17%	8674	0.30%	8700	'30days'	'30days'	1
	1		'ce16x16'	'CE16'	1	16750	1.47%	17098	0.58%	17000	'1years'	'1years'	1
	1		'ce16x16'	'CE16'	1	16750	1.47%	17042	0.25%	17000	'30days'	'1minutes'	
	1		'ce16x16'	'CE16'	1	16750	1.47%	17084	0.49%	17000	'30days'	'7days'	
	1		'ce16x16'	'CE16'	1	16750	1.47%	17129	0.76%	17000	'30days'	'30days'	1
	1	1	'ce14x14'	'CE16'		9150	5.17%	8635	0.75%	8700	'1years'	'2years'	1
	1	1	'ce14x14'	'CE16'		9150	5.17%	8647	0.61%	8700	'7days'	'3minutes'	
	1	1	'ce14x14'	'CE16'		9150	5.17%	8622	0.90%	8700	'30days'	'9days'	
		1	'ce16x16'	'CE16'	1	9150	0.00%	9147	0.03%	9150	'1years'	'2years'	1
		1	'ce16x16'	'CE16'	1	9150	0.00%	9143	0.08%	9150	'7days'	'3minutes'	
		1	'ce16x16'	'CE16'	1	9150	0.00%	9139	0.12%	9150	'30days'	'9days'	
1	1		'ge10'	'CE16'		8880	2.07%	8719	0.22%	8700	'7days'	'7days'	1
1	1	1	'ge10'	'CE16'		8880	2.07%	8730	0.34%	8700	'7days'	'9days'	1
1			'ge10'	'CE16'		8880	2.95%	9460	3.39%	9150	'7days'	'7days'	1
1		1	'ge10'	'CE16'		8880	2.95%	9484	3.65%	9150	'7days'	'9days'	1
	1		'ge7'	'GE7'	1	1980	1.00%	2018	0.90%	2000	'1years'	'1years'	1
	1		'ge7'	'GE7'	1	1980	1.00%	2005	0.25%	2000	'1minutes'	'1minutes'	1
	1		'ge7'	'GE7'	1	1980	1.00%	2012	0.60%	2000	'7days'	'7days'	1
	1		'ge7'	'GE7'	1	1980	1.00%	2016	0.80%	2000	'30days'	'30days'	1
	1		'atrium10'	'GE7'		7500	4.17%	7004	2.72%	7200	'1years'	'1years'	1
	1		'ge9'	'GE7'		6810	5.42%	7306	1.47%	7200	'1minutes'	'1minutes'	1
	1		'atrium10'	'GE7'		7500	4.17%	7023	2.46%	7200	'1minutes'	'7days'	
	1		'atrium10'	'GE7'		7500	4.17%	7036	2.28%	7200	'7days'	'30days'	
	1		'ge7'	'GE7'	1	10950	1.39%	10772	0.26%	10800	'1years'	'1years'	1
	1		'ge7'	'GE7'	1	10950	1.39%	10714	0.80%	10800	'1minutes'	'1minutes'	1
	1		'ge7'	'GE7'	1	10950	1.39%	10770	0.28%	10800	'1minutes'	'7days'	
	1		'ge7'	'GE7'	1	10950	1.39%	10783	0.16%	10800	'7days'	'30days'	
		1	'ge7'	'GE7'	1	8880	0.00%	8901	0.24%	8880	'1years'	'2years'	1
		1	'ge7'	'GE7'	1	8880	0.00%	8856	0.27%	8880	'1minutes'	'3minutes'	1
		1	'ge7'	'GE7'	1	8880	0.00%	8873	0.08%	8880	'7days'	'9days'	1
	1	1	'atrium10'	'GE7'		7500	4.17%	7011	2.63%	7200	'1years'	'2years'	1
	1	1	'ge9'	'GE7'		6810	5.42%	7306	1.47%	7200	'1minutes'	'3minutes'	1
	1	1	'atrium10'	'GE7'		7500	4.17%	7035	2.29%	7200	'1minutes'	'9days'	
1			'ge10'	'GE7'		8880	0.00%	8524	4.01%	8880	'1minutes'	'7days'	
1		1	'ge10'	'GE7'		8880	0.00%	8537	3.86%	8880	'1minutes'	'9days'	
1	1		'ge9'	'GE7'		6810	5.42%	7067	1.85%	7200	'1minutes'	'7days'	
1	1	1	'ge9'	'GE7'		6810	5.42%	6965	3.26%	7200	'7days'	'9days'	1

Following is some of the code in the form of m-files used to perform Task 4.3:

```
function makeOrigenInp(File_Name, Fuel_Type, Enrichment, Burnup, ...
   S_Power, Num_cycles, Mod_Den, Basis, T_c, T_c_units, Out_units,...
   NRank,Location)

   %Kenneth Dayman--University of Texas at Austin, PNNL--May 2011
   %This function takes input parameters and writes a *.inp file for input
   %into OrigenArp (via Scale6) for batch runs


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   %Input parameters:
   %File_Name: Name of input file and the subsequent saved output after
   %          running Origen
   %Fuel_Type: The type of fuel assembly in OrigenArp shorthand,
   %          e.g., ce14x14
   %Enrichment: U-235 enrichment of the fuel in percent, e.g., 5
   %Burnup:    Desired final burnup of the fuel after N cycles
   %S_Power: Power produced per unit fuel (MW/MTU) ~ power level of
   %        reactor
   %Num_Cycles: Number of cycles to use (1 or 3 typically)
   %Mod_Den: Density of the moderator (Non-variable for many reactors),
   %        g/cc
   %Basis: Amount of uranium basis in grams
   %T_c: Cooling time
   %T_c_units: Units of the cooling time parameter (seconds, minutes,
   %          months, years)
   %Out_units: Units of the plot/out, e.g., grams, Curies, etc.
   %NRank: Number of nuclides to be output, max 200
   %Location: Directory where the .inp will be located and where the
   %         Origen output will be saved


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   %Defines derived variables and arrays needed in the .inp file from the
   %input parameters

   %Hard-Wired Variables for Certain Reactors
   if strcmpi(Fuel_Type,'ce14x14')
      Mod_Den = 0.7332;
   elseif strcmpi(Fuel_Type,'ce16x16')
      Mod_Den = 0.71;
   elseif strcmpi(Fuel_Type,'w14x14')
      Mod_Den = 0.7264;
```

```
elseif strcmpi(Fuel_Type,'s14x14')
   Mod_Den = 0.7283;
elseif strcmpi(Fuel_Type,'w15x15')
   Mod_Den = 0.7135;
elseif strcmpi(Fuel_Type,'w17x17')
   Mod_Den = 0.723;
elseif strcmpi(Fuel_Type,'w17x17_ofa')
   Mod_Den = 0.71;
elseif strcmpi(Fuel_Type,'vver440(3.6)')
   Enrichment = 3.6;
   Mod_Den = 0.75;
elseif strcmpi(Fuel_Type,'vver440(3.82)')
   Mod_Den = 0.75;
   Enrichment = 3.82;
elseif strcmpi(Fuel_Type,'vver440(4.25)')
   Mod_Den = 0.75;
   Enrichment = 4.25;
elseif strcmpi(Fuel_Type,'vver440(4.38)')
   Mod_Den = 0.75;
   Enrichment = 4.38;
elseif strcmpi(Fuel_Type,'vver1000')
   Mod_Den = 0.7145;
elseif strcmpi(Fuel_Type, 'agr')
   Mod_Den = 1;
elseif strcmpi(Fuel_Type, 'magnox')
   Mod_Den = 1;
elseif strcmpi(Fuel_Type, 'candu37')
   Mod_Den = 0.8121;
   Enrichment = 0.711;
elseif strcmpi(Fuel_Type, 'candu28')
   Mod_Den = 0.8121;
   Enrichment = 0.711;
end

time_of_cycle = (Burnup/Num_cycles)/S_Power;
power = S_Power*(Basis/1e6);
time_of_cycle_inc = time_of_cycle/10;
tempBasis = 1e6;
factor  = Basis/tempBasis;

%computes the number of atoms of each U isotope, rounding to acheive a
%total of 1e6 atoms
u234 = round(((9e-05)*Enrichment+9e-9)*tempBasis);
u235 = round(Enrichment*tempBasis/100);
u236 = round(0.516854*u234);
%changed to tempBasis to avoid getting a negative amount of u238
```

```
u238 = tempBasis - u236 - u235 - u234;

%rescales the number of each U isotope to get the recover the desired
%basis
u234 = u234*factor;
u235 = u235*factor;
u236 = u236*factor;
u238 = u238*factor;

%Write the *.inp file
%prints the arp block
fid = fopen(strcat(File_Name,'.inp'),'w');
fprintf(fid, '=arp\r\n');
fprintf(fid, '%s',Fuel_Type);
fprintf(fid, '\r\n');
fprintf(fid, '%g',Enrichment);
fprintf(fid, '\r\n');
fprintf(fid, '%d',Num_cycles);
fprintf(fid, '\r\n');

for i = 1:Num_cycles
    if time_of_cycle < 10
        fprintf(fid, '%0.6f',time_of_cycle);
        fprintf(fid,'\r\n');
    elseif time_of_cycle < 100
        fprintf(fid, '%0.5f',time_of_cycle);
        fprintf(fid,'\r\n');
    elseif time_of_cycle < 1000
        fprintf(fid, '%0.4f',time_of_cycle);
        fprintf(fid,'\r\n');
    elseif time_of_cycle < 10000
        fprintf(fid, '%0.3f',time_of_cycle);
        fprintf(fid,'\r\n');
    end
end

for i = 1:Num_cycles
    fprintf(fid,'%g',S_Power);
    fprintf(fid,'\r\n');
end

for i = 1:Num_cycles
    fprintf(fid, '1\r\n');
end

fprintf(fid,'%g',Mod_Den);
fprintf(fid,'\r\n');
fprintf(fid,'ft33f001\r\n');
fprintf(fid,'end\r\n');

%prints the origens block
fprintf(fid,'#origens\r\n');
fprintf(fid,'0$$ a4 33 a11 71 e t\r\n');
fprintf(fid, '%s',Fuel_Type);
fprintf(fid,'\r\n');
fprintf(fid,'3$$ 33 a3 1 27 a16 2 a33 18 e t\r\n');
fprintf(fid,'35$$ 0 t\r\n');
fprintf(fid,'56$$ 10 10 a10 0 a13 4 a15 3 a18 1 e\r\n');
fprintf(fid,'57** 0 a3 1e-05 1 e\r\n');
fprintf(fid,'95$$ 0 t\r\n');
fprintf(fid,'Cycle 1 -');
fprintf(fid,'%s',File_Name);
fprintf(fid,'\r\n');
fprintf(fid, '%g',Basis/1e6);
fprintf(fid,' MTU\r\n');
fprintf(fid,'58** ');
for i = 1:5
    fprintf(fid,'%g',power);
    fprintf(fid,' ');
end
fprintf(fid, '\r\n ');
for i = 1:5
    fprintf(fid,'%g',power);
    fprintf(fid,' ');
end
fprintf(fid,'\r\n');
fprintf(fid,'60** ');
j = 0;
for i = 1:10
    if i == 6
        fprintf(fid,'\r\n ');
    end
    j = j + time_of_cycle_inc;
    if j < 10
        fprintf(fid, '%0.6f',j);
        fprintf(fid,' ');
    elseif j < 100
        fprintf(fid, '%0.5f',j);
        fprintf(fid,' ');
    elseif j < 1000
        fprintf(fid, '%0.4f',j);
        fprintf(fid,' ');
    elseif j < 10000
        fprintf(fid, '%0.3f',j);
        fprintf(fid,' ');
    end
end

if T_c == .1
    num_time_points = 7;
elseif T_c > .1 && T_c <= .3
    num_time_points = 6;
elseif T_c > .3 && T_c <= 1
    num_time_points = 7;
elseif T_c > 1 && T_c <= 3
    num_time_points = 8;
elseif T_c > 3 && T_c <= 10
    num_time_points = 9;
elseif T_c > 10
    num_time_points = 8;
end

fprintf(fid,'\r\n');
fprintf(fid,'66$$ a1 2 a5 2 a9 2 e\r\n73$$ 922340 922350 922360 922380\r\n');
fprintf(fid, '74** ');
fprintf(fid, '%g',u234);
fprintf(fid, ' ');
fprintf(fid, '%g',u235);
fprintf(fid, ' ');
fprintf(fid, '%g',u236);
fprintf(fid, ' ');
fprintf(fid, '%g',u238);
fprintf(fid, '\r\n');
fprintf(fid,'75$$ 2 2 2 2\r\n');
fprintf(fid,'t\r\n');
fprintf(fid,'54$$ a8 1 a11 0  e\r\n');
fprintf(fid,'56$$ a2 ');
fprintf(fid,'%d',num_time_points);
fprintf(fid,' a6 1 a10 10 ');
if strcmpi(T_c_units,'seconds')
    fprintf(fid, 'a14 1 ');
elseif strcmpi(T_c_units, 'minutes')
    fprintf(fid, 'a14 2 ');
elseif strcmpi(T_c_units, 'hours')
    fprintf(fid, 'a14 3 ');
elseif strcmpi(T_c_units, 'years')
    fprintf(fid, 'a14 5 ');
end
```

```
fprintf(fid, 'a15 3 a17 2 e\r\n');
fprintf(fid,'57** 0 a3 1e-05 e\r\n');
fprintf(fid,'95$$ 0 t\r\n');
fprintf(fid,'Cycle 1 Down - ');
fprintf(fid,'%s',File_Name);
fprintf(fid,'\r\n');
fprintf(fid, '%g',Basis/1e6);
fprintf(fid,' MTU\r\n');
fprintf(fid,'60** ');

%CHECK THIS BLOCK
if T_c == .1
   num_time_points = 7;
   fprintf(fid, '0.0001 0.0003 0.001 0.003 0.01 0.03 ');
   fprintf(fid,'%g',T_c);
elseif T_c > .1 && T_c <= .3
   num_time_points = 6;
   fprintf(fid, '0.001 0.003 0.01 0.03 0.1 ');
   fprintf(fid,'%g',T_c);
elseif T_c > .3 && T_c <= 1
   num_time_points = 7;
   fprintf(fid,'0.001 0.003 0.01 0.03 0.1 0.3 ');
   fprintf(fid,'%g',T_c);
elseif T_c > 1 && T_c <= 3
   num_time_points = 8;
   fprintf(fid, '0.001 0.003 0.01 0.03 0.1 0.3 1 ');
   fprintf(fid, '%g',T_c);
elseif T_c > 3 && T_c <= 10
   num_time_points = 9;
   fprintf(fid, '0.001 0.003 0.01 0.03 0.1 0.3 1 3 ');
   fprintf(fid, '%g',T_c);
elseif T_c > 10
   num_time_points = 8;
   fprintf(fid, '0.01 0.03 0.1 0.3 1 3 10 ');
   fprintf(fid, '%g',T_c);
end
%END BLOCK

fprintf(fid,'\r\n');
fprintf(fid,'61** f0.05\r\n');
fprintf(fid,'65$$\r\n');
fprintf(fid,'''Gram-Atoms   Grams   Curies   Watts-All   Watts-Gamma\r\n');
fprintf(fid,' 3z   1   0   0   3z   3z   3z   6z\r\n');
fprintf(fid,' 3z   1   0   0   3z   3z   3z   6z\r\n');
fprintf(fid,' 3z   1   0   0   3z   3z   3z   6z\r\n');
fprintf(fid,'81$$ 2 0 26 1 a7 200 e\r\n');
```

```
    fprintf(fid, '82$$ ');

    for i = 1:num_time_points
        fprintf(fid, '2 ');
    end

    fprintf(fid, 'e\r\n');
    fprintf(fid,'83**\r\n');
    fprintf(fid,' 1.0000000e+07 8.0000000e+06 6.5000000e+06 5.0000000e+06 4.0000000e+06\r\n');
    fprintf(fid,' 3.0000000e+06 2.5000000e+06 2.0000000e+06 1.6600000e+06 1.3300000e+06\r\n');
    fprintf(fid,' 1.0000000e+06 8.0000000e+05 6.0000000e+05 4.0000000e+05 3.0000000e+05\r\n');
    fprintf(fid,' 2.0000000e+05 1.0000000e+05 5.0000000e+04 1.0000000e+04 e\r\n');
    fprintf(fid,'84**\r\n');
    fprintf(fid,' 2.0000000e+07 6.4340000e+06 3.0000000e+06 1.8500000e+06\r\n');
    fprintf(fid,' 1.4000000e+06 9.0000000e+05 4.0000000e+05 1.0000000e+05 1.7000000e+04\r\n');
    fprintf(fid,' 3.0000000e+03 5.5000000e+02 1.0000000e+02 3.0000000e+01 1.0000000e+01\r\n');
    fprintf(fid,' 3.0499900e+00 1.7700000e+00 1.2999900e+00 1.1299900e+00 1.0000000e+00\r\n');
    fprintf(fid,' 8.0000000e-01 4.0000000e-01 3.2500000e-01 2.2500000e-01 9.9999850e-02\r\n');
    fprintf(fid,' 5.0000000e-02 3.0000000e-02 9.9999980e-03 1.0000000e-05 e\r\n');
    fprintf(fid,'t\r\n');

    for i = 1:num_time_points
        fprintf(fid,'56$$ 0 0 a10  ');
        fprintf(fid,'%g',i);
        fprintf(fid,' e t\r\n');
    end
    fprintf(fid,'56$$ f0 t\r\n');
    fprintf(fid,'end\r\n');
    fprintf(fid,'=opus\r\n');
    fprintf(fid,'LIBUNIT=33\r\n');
    fprintf(fid,'TYPARAMS=NUCLIDES\r\n');
    Out_units = upper(Out_units);
    fprintf(fid,'UNITS=');
    fprintf(fid,'%s',Out_units);
    fprintf(fid,'\r\n');
    fprintf(fid,'LIBTYPE=ALL\r\n');
    fprintf(fid,'TIME=');
    T_c_units = upper(T_c_units);
    fprintf(fid,'%s',T_c_units);
    fprintf(fid,'\r\n');
    fprintf(fid,'NPOSITION=');
    for i = 1:num_time_points
        fprintf(fid,'%g',i);
        fprintf(fid,' ');
    end
    fprintf(fid,'end\r\n');
    fprintf(fid,'NRANK=(');
    fprintf(fid,'%d',NRank);
    fprintf(fid,')\r\n');
    fprintf(fid,'end\r\n');
    fprintf(fid,'#shell\r\n');
    fprintf(fid,'copy ft71f001 "');
    fprintf(fid,'%s',Location);
    fprintf(fid,'\\');
    fprintf(fid,'%s',File_Name);
    fprintf(fid,'.f71"\r\n');
    fprintf(fid,'del ft71f001\r\n');
    fprintf(fid,'end');

    fclose(fid);
end


function [Database,MasterNuclide,newNames] = makeDTRAdatabase()
    %Kenneth Dayman -- University of Texas at Austin -- June 2011
    %
    %This function opens each ORIGEN run listed in the batch file, and compiles
    %the nuclide and mass information into a usable database for use in inverse
    %calculations for identifying fuel cycles (defined here as ordered triples:
    %(reactor_type, cooling time, burnup))
    %
    %A MasterNuclideList is compiled from all the nuclides observed in the
    %output of the ORIGEN runs.  The data from each run is then compiled into
    %an (N x 1628) matrix, where N is the number of nuclides in the
    %MasterNuclideList and 1628 is the number of runs.  If a particular run
    %does not exhibit a nuclide in the MasterNuclideList, a zero will be
    %inserted.  The ordering of each run (column) is preserved to ensure that
    %each row only contains data relating to a single nuclide.

    %Form the list of filenames of the ORIGEN output
    fid = fopen('OrigenRuns.txt');
    temp = textscan(fid,'%s %s %s');
    names = temp{3};
    fclose(fid);
    newNames = cell(length(names),1);
```

```
%Declares Useful static variables
NumRuns = length(names);
NumNuclides = 200;

%Reads in the nuclides from each run and appends each unique nuclide not
%already in MasterNuclideList to the end of MasterNuclideList.  First run's
%nuclide list is set equal to MasterNuclideList to start.
longname = strcat(names(1),'._plot000.plt');
fid = fopen(char(longname));
temp = textscan(fid,'%s %*f %*f %*f %*f %*f %*f %*f %*f %f','HeaderLines',6);
RunNuclides = temp{1};
fclose(fid);

%Sets MasterNuclideList to the list from first run, then deletes the last
%one (total) and decrements the NumbMasterNuclides
MasterNuclide = RunNuclides;
NumMasterNuclides = length(MasterNuclide);
MasterNuclide(NumMasterNuclides)='';
NumMasterNuclides = NumMasterNuclides-1;

%Goes through the remaining runs and searches for unique nuclides,
%appending these to the end of the MasterNuclide
for i = 2:NumRuns
    fprintf('%d',i);
    fprintf('\n');
    %opens file and reads in the nuclide names
    longname = strcat(names(i),'._plot000.plt');
    fid = fopen(char(longname));
    if mod(i,4)==1
        temp = textscan(fid, '%s %*f %*f %*f %*f %*f %*f %*f %*f %f','HeaderLines',6);
    elseif mod(i,4)==2
        temp = textscan(fid, '%s %*f %*f %*f %*f %*f %*f %*f %f','HeaderLines',6);
    else
        temp = textscan(fid, '%s %*f %*f %*f %*f %*f %*f %f','HeaderLines',6);
    end
    RunNuclides = temp{1};
    fclose(fid);

    %compares the nuclides in the run to all nuclides in the master list
    for j = 1:NumNuclides %in the run
        for k = 1:NumMasterNuclides %master list
            presentInMaster=0;
            if strcmpi(RunNuclides(j),MasterNuclide(k))
                presentInMaster=1;
                break;
            end
            end
            if presentInMaster==0
                MasterNuclide(NumMasterNuclides + 1) = RunNuclides(j);
                NumMasterNuclides = NumMasterNuclides + 1;
            end
        end
    end

    %Forming Database
    Database = zeros(NumMasterNuclides,NumRuns);

    %the l-loop allows for the loading of the Database matrix to group
    %together all reactor types, then cooling times, then burnup
    t = 0;
    for l = 1:4
        for i = l:4:NumRuns
            t = t + 1;
            fprintf('%d',t);
            fprintf('\n');
            newNames(t) = names(i);
            %opens the run file and loads the nuclide names and masses
            longname = strcat(names(i),'._plot000.plt');
            fid = fopen(char(longname));
            if mod(i,4)==1
                temp = textscan(fid, '%s %*f %*f %*f %*f %*f %*f %*f %*f %f','HeaderLines',6);
            elseif mod(i,4)==2
                temp = textscan(fid, '%s %*f %*f %*f %*f %*f %*f %*f %f','HeaderLines',6);
            else
                temp = textscan(fid, '%s %*f %*f %*f %*f %*f %*f %f','HeaderLines',6);
            end
            RunNuclides = temp{1};
            RunMass = temp{2};
            fclose(fid);

            for j = 1:NumMasterNuclides
                for k = 1:NumNuclides
                    if strcmpi(RunNuclides(k),MasterNuclide(j))
                        Database(j,t) = RunMass(k);
                    end %implicitly if nuclide in master isn't in run, set to 0
                end
            end
        end
    end
end
```

```
function [reactor_type,cooling_time,guessBurnup,interpBurnup,smallFoM] =
IDfuelCycle(unknownFileName,numTimePoints,Database,MasterNuclide,names,FoMtype)
   %Kenneth Dayman -- University of Texas -- June 2011
   %
   %This function takes the ORIGEN ouput of an unknown fuel cycle (defined
   %for this work as an ordered triple (reactor_type, cooling_time,burnup)
   %and makes a best guess against a database of the Mass outputs from a
   %number of known, simulated fuel cycles.  The best-fit estimate will
   %use a least-squares metric (squared Euclidean norm), and then
   %interpolates an estimate for the burnup using the "taxicab" norm as a
   %distance metric from the samples adjacent the Database matrix to the
   %best-fit fuel cycle.
   %
   %unknownFileName = Name of unknown fuel cycle's ORIGEN output (w/o
   %                 file extension)
   %numTimePoints = number of time points in the unknown's ORIGEN output
   %Database = sorted matrix of mass of each nuclide in the MasterNuclide
   %           list for each of the fuel cycles in names. Rows are
   %           nuclides in the MasterNuclide list and Columns are the
   %           fuel cycles in names
   %MasterNuclides = Listing of all the nuclides accounted for in the
   %           Database
   %names = all the fuel cycles accounted for in the Database
   %FoM type = {'square'   => least squares Figure of Merit
   %            'abs'      => absolute value Figure of Merit
   %            'relative' => relative absolute value Figure of Merit
   %                       (possibly ill-defined due to divide by 0's}

   %Reads in the Names and masses of the unknown ORIGEN output
   longname = strcat(unknownFileName,'._plot000.plt');
   fid = fopen(char(longname));
   if numTimePoints == 6
      temp = textscan(fid, '%s %*f %*f %*f %*f %*f %f','HeaderLines',6);
   elseif numTimePoints == 7
      temp = textscan(fid, '%s %*f %*f %*f %*f %*f %*f %f','HeaderLines',6);
   elseif numTimePoints == 8
      temp = textscan(fid, '%s %*f %*f %*f %*f %*f %*f %*f %f','HeaderLines',6);
   elseif numTimePoints == 9
      temp = textscan(fid, '%s %*f %*f %*f %*f %*f %*f %*f %*f %f','HeaderLines',6);
   end

   RunNuclides = temp{1};
   RunMass = temp{2};
   fclose(fid);
   NumMasterNuclides = length(MasterNuclide);
   NumNuclides = 200;
```

```
%Conforms the mass numbers to MasterNuclides order and full listing.
%If a nuclide in MasterNuclides is not in the unkown, then a 0 is
%listed.  If a nuclide is the unknown is not in the MasterNuclides
%(very unlikely), that nuclide will be compared against a 0 external of
%the database in the calculating of the figures of merit
unknownFull = zeros(length(MasterNuclide),1);
NumNotPresent = 0;
NotPresentMasses = zeros(1,1);

for j = 1:NumMasterNuclides
   for k = 1:NumNuclides
      if strcmpi(RunNuclides(k),MasterNuclide(j))
         unknownFull(j) = RunMass(k);
      end
   end
end

%checks for stuff in RunNuclides not in MasterNuclide and saves any
%nonPresentNuclides in NotPresentMasses for use later in calc. FoM's
for k = 1:NumNuclides
   notThere = 1;
   for j = 1:NumMasterNuclides
      if strcmpi(RunNuclides(k),MasterNuclide(j))
         notThere = 0;
         break;
      end
   end
   if notThere == 1;
      NotPresentMasses(NumNotPresent + 1) = RunMass(k);
      NumNotPresent = NumNotPresent + 1;
   end
end

%finds FoM for each known fuel cycle in the Database
FoM = zeros(1,length(names));
for i = 1:length(names)
   if strcmpi(FoMtype,'square')
      FoM(i) = sum((unknownFull-Database(:,i)).*(unknownFull-Database(:,i)));   %FoM
   elseif strcmpi(FoMtype,'abs')
      FoM(i) = sum(abs(unknownFull-Database(:,i)));
   elseif strcmpi(FoMtype,'relative')
      FoM(i) = sum(abs(unknownFull-Database(:,i))./Database(:,i));
   else fprintf('Invalid Figure of Merit type');
   end

   if NumNotPresent ~= 0
```

```
            for j = 1:NumNotPresent
                if strcmpi(FoMtype,'square')
                    FoM(i) = FoM(i) + (NotPresentMasses(j)^2);                        %FoM
                elseif strcmpi(FoMtype,'abs')
                    FoM(i) = FoM(i) + abs(NotPresentMasses(j));
                elseif strcmpi(FoMtype,'relative')
                    fprintf('nuclides not found in Database ignored.  FoM undefined');
                else
                    fprintf('Invalid Figure of Merit type');
                end
            end
        end
    end

   %Finds Minimum FoM and Declares that the best-fit fuel cycle (number i)
   smallFoM = FoM(1);
   winner = 1;
   for i=2:length(FoM)
        if FoM(i)<smallFoM
            smallFoM = FoM(i);
            winner = i;
        end
   end

   %Parses name(i) to give reactor type and cooling time
   tempparts = regexp(names(winner),'_','split');
   parts = tempparts{1};
   reactor_type = parts(1);
   cooling_time = parts(3);
   guessBurnup = str2double(parts(2));

   %Interpolates burnup estimate from the i+1 and i-1 fuel cycles' burnups
   if (guessBurnup ~= 600 && guessBurnup ~= 17700) || (guessBurnup ~= 600 && guessBurnup
~= 12330)
       temptemp1 = regexp(names(winner-1),'_','split');
       temp1=temptemp1{1};
       temptemp2 = regexp(names(winner+1),'_','split');
       temp2=temptemp2{1};
       burn1 = str2double(temp1(2));
       burn2 = str2double(temp2(2));
       d1 = abs(FoM(winner)-FoM(winner-1));
       d2 = abs(FoM(winner)-FoM(winner+1));
       interpBurnup = (d1/(d1+d2))*(burn2-burn1) + burn1;
   else
       interpBurnup = 0;
   end
```

```
end
```