Development of an Application Programming Interface for Depletion Analysis (APIDA)

Daniel Lago and Farzad Rahnema¹

Nuclear & Radiological Engineering and Medical Physics Programs

Georgia Institute of Technology

770 State Street NW

Atlanta, Georgia 30332-745, USA

dlago3@gatech.edu, farzad@gatech.edu

Abstract

A new utility has been developed with extensive capabilities in identifying nuclide decay and transmutation characteristics, allowing for accurate and efficient tracking of the change in isotopic concentrations in nuclear reactor fuel over time when coupled with a transport solution method. This tool, named the Application Programming Interface for Depletion Analysis (APIDA), employs both a matrix exponential method and a linear chain method to solve for the end-of-time-step nuclide concentrations for all isotopes relevant to nuclear reactors. The Chebyshev Rational Approximation Method (CRAM) was utilized to deal with the ill-conditioned matrices generated during lattice depletion calculations, and a complex linear chain solver was developed to handle isotopes reduced from the burnup matrix due to either radioactive stability or a sufficiently low neutron-induced reaction cross section. The entire tool is housed in a robust but simple application programming interface (API). The development of this API allows other codes, particularly numerical neutron transport solvers, to incorporate APIDA as the burnup solver in a lattice depletion code or reactor core analysis code in memory, without the need to write or read from the hard disk. The APIDA code was benchmarked using several decay and transmutation chains. Burnup solutions produced by APIDA were shown to provide material concentrations comparable to the analytically solved Bateman equations - well below 0.01% relative error for even the most extreme cases using isotopes with vastly different decay constants. As a first order demonstration of the API, APIDA was coupled with the transport solver in the SERPENT code for a fuel pin cell depletion problem. Results show APIDA to be effective and efficient in solving lattice depletion problems, in addition to being successful in terms of portability for users to implement via the API.

Keywords: burnup, benchmark, Bateman equations

1. Introduction

The world's energy demands are at historic levels and the need for clean, sustainable, and dependable power is unquestionably at the forefront of modern society. Nuclear power is increasingly becoming the most powerful tool to combat the impending energy crisis, subsequently

making adequate tools to safely operate nuclear reactors and accurately track the changes in fuel over the core lifetime vital to ensuring the advent of a new fleet of nuclear power plants.

The current push in nuclear reactor physics research is to create all-encompassing codes capable of capturing all the multiphysics present in the complicated core of a nuclear reactor – thermal hydraulics, material performance, radiation shielding, and kinetics being the most relevant. This study focused on the creation of a novel burnup module with sufficient modernization to couple with current and future neutron transport solution methods.

The utility developed in this study, the Application Programming Interface for Depletion Analysis (APIDA), is a standalone burnup solver with an interface accessible enough to be coupled with potentially any transport solver to form a lattice depletion tool or reactor core analysis code. The proceeding sections discuss the methods used to solve the burnup equations, the verification of the solver, the construction of an application programming interface, and the initial coupling with a transport solver to produce preliminary results.

2. Development of a Burnup Solver

Development of the burnup solver in the APIDA code involved studies into multiple linear chain methods and matrix exponential methods. Literature review and implementation of proof-of-concept algorithms led to the conclusion that the Chebyshev Rational Approximation Method (CRAM) was the most effective and flexible method for solving burnup equations.

While CRAM, and inherently any numerical matrix solver, handles the requisite transition matrices in burnup problems well, it cannot be applied to nuclides that introduce zeros into the diagonals of said matrices. Consequently, a linear decay chain solver needs to be applied to solve for the time rate of change of nuclide concentration for those isotopes.

The APIDA tool is the result of combining the two preceding methods – a hybrid matrix exponentiation and linear chain solver for burnup problems. The following section briefly outlines the theory and foundation of CRAM. For a more complete analysis, the reader is encouraged to refer to the work done by Pusa and Leppanen (2010) and Pusa (2013).

2.1 Matrix Exponential Solver – CRAM

The phenomena governing the time rate of change of nuclide concentration can be formalized by the following equation.

$$\frac{dN_i}{dt} = \underbrace{\sum_{j=1}^m b_{ij} \lambda_j N_j + \sum_{k=1}^m y_{ik} RR_k N_k}_{\text{Production}} - \underbrace{(\lambda_i + RR_i) N_i}_{\text{Destruction}}$$

$$RR_i = \int dX \phi(X) \sigma_i(X) \quad (1)$$

N_i = atom density of nuclide i

λ_i = radioactive decay constant of nuclide i

$\sigma_i(X)$ = neutron absorption cross-section for nuclide i over phase-space X

$\phi(X)$ = neutron angular flux over phase-space X

b_{ij} = branching ratio of all other nuclides to nuclide i

y_{ik} = branching ratio for neutron absorption by other nuclides that lead to nuclide i

Equation 1 shows the time rate of change of the concentration of nuclide N_i as a balance equation with the net result being the sum of the destruction and production of the nuclide of interest. The necessary assumptions in forming Equation 1 are as follows: a homogenous medium; space-averaged and energy-integrated reaction rate over one energy group; sufficiently small time step to assume a constant flux. In most lattice depletion applications, these assumptions are acceptable and provide accurate solutions if utilized appropriately.

Production of N_i can result from the decay of another nuclide N_j into N_i with the probability of said reaction expressed by its associated branching ratio b_{ij} , or production can come from nuclide N_k participating in a reaction under the influence of a flux resulting in the production of nuclide N_i . Destruction of N_i is determined by two factors, both of which are dependent on the type of problem and isotope. If the nuclide is unstable, its decay probability (λ) determines the removal rate. If there is irradiation, implying a flux (ϕ), then the nuclide's reaction rate determines the removal rate. If both principles are appropriate, then the decay probability and reaction rate are summed to constitute the removal coefficient for the nuclide N_i .

In regards to the reaction types represented by the branching ratios in Equation 1, the initial implementation of APIDA considered the reactions shown in Table 1. While not all encompassing, these equations constitute most of the neutron-induced reactions occurring in a nuclear reactor. Future work for APIDA will address implementation of all neutron-induced reactions.

Table 1: Decay modes and their associated reactions.

Decay Mode	Reaction
$(n, 2n)$	${}_Z^AX + n \rightarrow {}_Z^{A-1}X + 2n$
$(n, 3n)$	${}_Z^AX + n \rightarrow {}_Z^{A-2}X + 3n$
$(n, 4n)$	${}_Z^AX + n \rightarrow {}_Z^{A-3}X + 4n$
(n, γ)	${}_Z^AX + n \rightarrow {}_Z^{A+1}X + \gamma$

(n, p)	${}_Z^AX + n \rightarrow {}_{Z-1}^AY + p$
(n, d)	${}_Z^AX + n \rightarrow {}_{Z-1}^{A-1}Y + d$
(n, t)	${}_Z^AX + n \rightarrow {}_{Z-1}^{A-2}Y + t$
$(n, {}^3He)$	${}_Z^AX + n \rightarrow {}_{Z-2}^{A-2}Y + {}^3He$
(n, α)	${}_Z^AX + n \rightarrow {}_{Z-2}^{A-3}Y + \alpha$
(n, f)	${}_Z^AX + n \rightarrow \text{Fission Products} + \text{neutrons}$

Historically, matrix exponential methods have been applied with varied success in multiple fuel decay and transmutation codes.

For a chosen set of isotopes, Equation 1 can be set up in matrix notation as follows,

$$N' = \mathbf{A}N \quad (2)$$

$\mathbf{A}$ = transition matrix containing coefficients for decay and transmutation

N = nuclide concentration vector, *s.t.* $N = N_i$ for all i

N' = first derivative in time of nuclide concentration

The solution can be given in the form of an exponential as follows,

$$N = \exp(\mathbf{A}t)N(0) \quad (3)$$

$N(0)$ = initial nuclide concentration vector

The matrices produced in burnup problems exhibit one unique property which can be exploited to produce more accurate answers relatively quickly – the eigenvalues of the burnup matrix have been found to be bounded near the negative real axis Pusa (2011). Physically, this is a natural outcome of generating the burnup matrix; the diagonal elements represent the removal coefficients for a particular isotope and are always negative. For methods requiring solutions near the origin (the Taylor series expansion of the matrix exponential), this presents a mathematical hurdle.

CRAM takes advantage of this property and allows burnup matrices to be solved accurately without the removal of short-lived isotopes. For a full description on the derivation of CRAM, refer to Pusa et. al. (2010, 2011, 2014).

2.2 Linear Chain Solver

Burnup problems produce strictly structured and highly sparse matrices. Specifically, the resulting transition matrix containing all the coefficients corresponding to the set of ordinary differential equations governing the problem can be generated with recursive logic.

For the problem presented below,

$$N'(t) = AN(t), \quad (4)$$

where N is the vector of nuclide concentrations and A is the transition matrix holding all the relevant coefficients. The elements of A are defined as follows,

$$a_{ij} = \begin{cases} l_{ij}\lambda_j + (f_{ij}\sigma_j^f + nn_{ij}\sigma_j^{nn})\phi, & i \neq j \\ -(\lambda_i + \sigma_i\phi), & i = j \end{cases}, \quad (5)$$

where, l_{ij} is the decay branching ratio of nuclide j to nuclide i , λ_j is the disintegration constant of nuclide j , f_{ij} is the yield fraction of the fission of nuclide j yielding nuclide i , σ_j^f is the fission cross section of nuclide j , σ_j^{nn} is the cross section for non-fission neutron reactions for nuclide j , nn_{ij} is the branching ratio of a non-fission neutron reaction with nuclide j yielding nuclide i , σ_j is the total cross section of nuclide j , and ϕ is the scalar neutron flux.

Observing the elements along the diagonal matrix, there are scenarios in which the condition number of A could prohibitively large or even infinite in the case of an extremely small magnitude along the diagonal or a zero element (i.e. a stable isotope with little to no reactions). In this case, the transition matrix is ill-conditioned and needs to be reduced – the row and column associated with the isotope in question must be removed from the matrix. This allows the transition matrix to be full and amenable to exponentiation. Consequently, the isotopes reduced from the transition matrix need to have their final concentrations solved for in a different manner.

In order to find the final concentrations for the reduced nuclides, a linear chain method was utilized in APIDA. For each nuclide not included in the transition matrix, solutions provided by the appropriate Bateman equations are applied to each nuclide. For cases where there is no irradiation and no external sources of production of nuclide N_i , the following equation is used,

$$N_i(t) = N_1^0 \lambda_1 \lambda_2 \dots \lambda_{i-1} \sum_{j=1}^i \frac{e^{-\lambda_j t}}{\prod_{\substack{k=1 \\ k \neq j}}^i (\lambda_k - \lambda_j)} + N_i^0, \quad (6)$$

where λ_j is the disintegration constant of nuclide j and N_1^0 is the initial concentration of the first parent nuclide of the chain. In the case of neutron irradiation and the production of nuclide N_i with precursors, the following equation is used,

$$N_i(t) = \sum_{l=1}^{i-1} \left[N_l^0 \eta_l \eta_{l+1} \dots \eta_{i-1} \sum_{j=1}^i \frac{e^{-\mu_j t}}{\prod_{\substack{k=l \\ k \neq j}}^i (\mu_k - \mu_j)} \right] + N_i^0, \quad (7)$$

where η_i is the chain-linking precursor decay constant of nuclide N_i (including decay from other nuclides and reaction transmutation) and μ_i is the effective decay constant of nuclide i (total removal rate including decay and reactions).

In both cases described above, the branching ratio calculator in the APIDA code is utilized to cycle backwards through each nuclide's decay scheme and enumerate the linear chains for each chain. Expectedly, the resulting set of decay constants and reaction rates are strongly associated with the coinciding coefficients in the transition matrix.

2.3 Verification of the burnup solver

Initial benchmarking of APIDA was done with analytical solutions generated in the Wolfram Alpha tool Mathematica (Wofram Research, 2016). Several decay chains were chosen, with and without reactions and fission yields, over a wide range of decay probabilities. The problems were chosen with the purpose of challenging the methods in APIDA to ensure accuracy of solutions for any given set of isotopes. Figures 1-5 briefly illustrate the problems used to verify the burnup solver. A thorough description and the development of these benchmark problems can be found in Lago and Rahnema (2016).

Figure 1. Description of Benchmark #1.

Figure 2. Description of Benchmark #2.

Figure 3. Description of Benchmark #3.

Figure 4. Description of Benchmark #4.

Figure 5. Description of Benchmark #5.

3. Development of an Application Programming Interface (API)

Traditionally, lattice depletion codes use some transport solution method of choice (discrete ordinates, Monte Carlo, collision probability method, etc.) and an associated burnup solver specifically designed and tailored for the prescribed neutronics code. Historically, these methods are inextricably linked – specifically, the burnup solver cannot be extracted and implemented with another transport solver. Widely used lattice depletion codes like HELIOS (Simenov, 2003), SERPENT Lepannen (2008), CASMO (Edenius, 1995), and PARAGON (Ouisloumen and Tahara, 1998) all use self-developed depletion modules that are difficult or impossible to utilize without their associated transport codes. This presents numerous hurdles, chiefly the inability to independently validate the burnup solver and the impracticality of attempting to integrate the module with a novel transport solution method. APIDA presents a significant step in providing a universal burnup solver capable of integration with any code via a simple Application Programming Interface (API).

3.1 Communication between transport solver and burnup module

The basic procedure of any lattice depletion method is first to determine the physical characteristics (flux) at one discrete time with a transport solver, and then to use that information to calculate the generation, destruction, and change in nuclide concentrations over one or more

prescribed time steps with a burnup solver. Once the new nuclide concentrations in the fuel have been calculated, the information is relayed back to the transport solver to calculate the new physical characteristics (flux, eigenvalue, etc.).

Efficient iteration between the transport and burnup modules in a lattice depletion code is key to minimizing its computation time. Historically, these modules were coupled using text files. In essence, the two codes were treated as “black boxes” in relation to one another. Figure 6 illustrates a typical iterative cycle between a transport solver and a burnup module.

Figure 6. Flowchart of communication between transport solver and burnup module using text inputs.

While this method has allowed for relatively seamless independent code development, it presents some serious problems in modern implementations. The most prominent issue this setup presents is difficulty with large-scale embedding. Cluster computing has become an integral part of reactor modeling, especially in regards to multi-physics applications. The need for high performance computing has forced developers to abandon the model in Figure 6, since the “black box” model requires writing an input to the hard disk. The preferred option in coupling two codes is “in-memory” with the use of an Application Programming Interface (API). Rather than generating text inputs to be read by both code modules, an API allows the “parent” program to call upon the secondary code module in-memory without the need to stop and generate a text file or touch the hard disk.

3.2 Object-oriented framework in APIDA

APIDA is a code written in the C++ programming language and contains numerous features for users interested in all aspects of burnup calculations. All the features are housed within classes, each with their own API suited to the particular information being retrieved.

The key to the portability and utility of APIDA in terms of seamless integration with other codes is the object-oriented framework of the API. The data pertinent to the burnup problem is stored in exposed data containers where the user only needs to provide the data necessary to run a calculation. Programming these structures in C++ allows for these objects to be instantiated simultaneously in a massively parallel environment (multiple processors over multiple nodes). The C++ programming language allows for memory to be allocated on the heap, making it possible to instantiate multiple instances without threatening the integrity of the data with overwrites.

Discussed in the next section, the APIDA code is separated into multiple objects representing the important variables in a burnup calculation. The data for each important variable is encapsulated within different objects available to the user with “get” and “set” functions. Each object, or “class,” contains multiple functions publicly available to the user, allowing for the manipulation of private data.

3.3 Classes and features in APIDA

As mentioned in the previous section, the APIDA framework is broken down into multiple classes, each responsible for handling the requisite steps for a burnup calculation. The main classes inherently instantiated for a burnup calculation are the following:

- ‘apida.h’ – container for the front-end user API.
- ‘library_builder.h’ – container for library data (decay and fission yields).
- ‘tran_mat.h’ – container for transition matrix.
- ‘depletion.h’ – container for solver methods for burnup problems.
- ‘output_proc.h’ – container for output processing methods.

Each class contains a suite of methods used to handle the operations required by each step of a burnup calculation. The ‘apida’ class serves as the front-end interface, allowing user to run a calculation with only the data absolutely required to complete a calculation – the list of nuclides being tracked, their corresponding initial concentrations, the reaction rates of each type by nuclide, and the time step. If desired, users can access burnup data explicitly (half-lives, fission yields, decay modes, etc.) via the methods available in each class.

3.4 Implementation of APIDA

The purpose of APIDA is to provide a simple and straightforward way to incorporate a burnup solver using the C++ interface. Future work for APIDA includes incorporating compatibility with more programming languages (C, FORTRAN, etc.) using “wrappers” to make the code interface amenable to other languages. In order to run the calculation, the steps are as follows:

1. Initialize the instance of the 'apida' class.
2. Provide a vector containing the ZAID of the isotopes to be tracked.
3. Provide a vector containing the initial concentrations of the isotopes to be tracked.
4. Provide a vector with the time steps (cumulative) for the calculation.
5. Provide the reaction rates for each isotope (description given in comments of code example).
6. Initialize the library to gather data for the calculation.
7. Set the initial concentrations, time steps, and reaction rates.
8. Run the calculation.
9. Retrieve the final concentrations for each nuclide at each time step.

APIDA contains numerous other features to allow for more robust calculations or to access specific data for other applications.

3.5 Testing the API implementation

As a first order application to test the methods in APIDA, a single fuel pin cell was modeled and depleted. The parameters for the pin cell were based on the AREVA European Pressurized Reactor design (AREVA, 2013-1 and 2013-2). It is a standard pressurized water reactor (PWR) fuel pin design – a cylindrical fuel pin surrounded by zirconium-based cladding with a small gap, encapsulated in a moderator (water) with boron. The fuel pin design is illustrated in Figure 7 and the geometric parameters for the pin cell are shown in Table 2.

Figure 7. Illustration of fuel pin cell depletion problem.

One fuel pin was modeled with UO_2 enriched to 3.5wt%. The fuel parameters for the pin cell problem are shown in Table 3.

In addition to the results generated by SERPENT and the results generated by coupling APIDA to serpent, HELIOS was also used to produce pin cell depletion results for further comparison. HELIOS is an extensively validated lattice depletion code utilizing method of collision probability (CPM) solver in 2D general geometry for transport solutions and a linearized chain method for burnup.

Table 2. Geometric parameters for fuel pin cell.

Pin Pitch	1.259840 cm
Cladding Outer Radius	0.474980 cm
Cladding Inner Radius	0.417957 cm
Fuel Pellet Radius	0.409575 cm
Boron Concentration	1000 ppm
Moderator Density	0.7 g/cm ³
Fuel Density	10.4 g/cm ³
Fuel Temperature	900 K
Moderator Temperature	600 K

Table 3. Fuel parameters for UO₂ pin cell.

Fuel Composition	UO ₂
Fuel Enrichment	3.5 wt% ²³⁵ U
Fuel Density	10.4 g/cm ³
Cladding Composition	Natural Zr
Cladding Density	6.514 g/cm ³
Moderator Density	0.7 g/cm ³
Soluble Boron Concentration	1000 ppm

4. Results

4.1 Burnup Solver Verification

The results (final concentration in number of atoms) for each benchmark problem are presented in Tables 4 through 9. For each problem, the generated results from *Mathematica* using the ‘DSolve’ function are shown as the reference solutions. Additionally, the results generated from the matrix exponential function in MATLAB (‘expm’) are also presented for the reader (MATLAB, 2006). The relative differences between the Mathematica results and the APIDA results were calculated using Equation 8 also shown in the table.

$$\text{APIDA R.D. (\%)} = \frac{|N_{\text{Mathematica}} - N_{\text{APIDA}}|}{N_{\text{Mathematica}}} * 100 \quad (8)$$

Table 4. Final concentration results for Benchmark #1.

	Mathematica	MATLAB	APIDA	APIDA R.D. (%)
U-238	856027125	856076082	856077093	5.84E-03
Th-234	0.012641889	0.012642	0.012642	3.35E-03

Table 5. Final concentration results for Benchmark #2.

	Mathematica	MATLAB	APIDA	APIDA R.E. (%)
Np-237	9.89808E+11	9.89808E+11	9.89808E+11	0.00E+00
Pa-233	34096	34096	34096	0.00E+00
U-233	9519333638	9519333605	9519333605	3.47E-07

Table 6. Final concentration results for Benchmark #3.

	Mathematica	MATLAB	APIDA	APIDA R.E. (%)
Pb-211	407570893	407570892	407570892	2.45E-07
Bi-211	25683207	25683207	25683207	0.00E+00
Tl-207	65963040	65780982	65780982	2.76E-01
Pb-207	9500792870	9500782861	9500782861	1.05E-04

Table 7. Final concentration results for Benchmark #4.

	Mathematica	MATLAB	APIDA	APIDA R.E. (%)
U-235	176886799	176886902	176886902	5.82E-05

U-236	1528302359	1528302835	1528302835	3.11E-05
U-237	9.22519E+11	9.22519E+11	9.22519E+11	6.37E-08
Np-237	75775364619	75775364627	75775364627	1.06E-08

Table 8. Final concentration results for Benchmark #5.

	Mathematica	MATLAB	APIDA	APIDA R.E. (%)
U-238	1768869	1768869	1768869	0.00E+00
U-239	450504	450504	450504	0.00E+00
Np-239	7765299776	7765299776	7765299776	0.00E+00
Pu-239	273085486	273085486	273085486	0.00E+00
Pu-240	279959954	279959954	279959954	0.00E+00
Pu-241	282115515	282115515	282115515	0.00E+00
Pu-242	21477	21477	21477	0.00E+00
Pu-243	275858289	275858289	275858289	0.00E+00
Am-241	529647302	529647302	529647302	0.00E+00
Am-243	172381411	172381411	172381411	0.00E+00
Am-244	312220315	312220315	312220315	0.00E+00
Cm-244	107192026	107192026	107192026	0.00E+00

As seen by the relative differences, the APIDA results are accurate to a very high degree for a wide range of benchmark problems. For the results with relative differences between APIDA and Mathematica approaching 0.01%, the concentrations are prohibitively small and consequently don't represent cause for concern. The results from APIDA and MATLAB are exact to numerical precision for every case and thus aren't compared directly.

4.2 API Implementation

The UO₂ fuel pin cell described in the previous section was used in the multi-step depletion calculation to compare APIDA to SERPENT and HELIOS. Results were generated with both SERPENT and HELIOS up to a total burnup of 10,000 MWD/MTU at intervals of 500 MWD/MTU, with the first two steps at 250 MWD/MTU for higher fidelity. Two sets of SERPENT results are shown. The transport solutions from SERPENT were used at each step by APIDA to generate the new material concentrations for the next step. The power was fixed at 34 W/g and calculations were run without xenon equilibrium and with predictor-corrector turned off. The results using SERPENT were generated running 15000 histories per cycle with 1000 total cycles and 200 inactive cycles. Figure 8 shows the change in the eigenvalue (k-effective) over these burnup steps using SERPENT, HELIOS, and APIDA, with APIDA results generated

with 738 isotopes tracked and 1049 isotopes tracked. Because the implementation of APIDA as the burnup solve gives the user the power to choose how many isotopes to track, this is an area of interest for future work. The specific isotopes tracked in each calculation are outlined in Appendix A.

Figure 8. Eigenvalue (k-eff) as a function of burnup resulting from SERPENT, HELIOS, and APIDA calculations.

The depletion curve generated by APIDA shows good agreement with SERPENT, especially when tracking 1049 isotopes. The curve generated when tracking 738 isotopes seems to diverge as the fuel pin is burned more, indicating added importance to some of the less dominant actinides in terms of pathways of transmutation. One noticeable difference between the burnup methods used in SERPENT and APIDA is the number of reaction types tracked. For this initial testing, only the 10 most relevant neutron-induced reactions were tracked by APIDA. Additionally, the standard deviation of the eigenvalue was on the order of 100 pcm. The statistical nature of Monte Carlo calculations, especially when it's not clear if the source has converged to the correct solution, also contributes to some of the differences in the eigenvalue.

In addition to the core eigenvalue, the atom density of certain isotopes is of considerable interest in fuel cycle analysis. One of the most important fission products related to criticality analysis is ^{135}Xe as it acts as a strong neutron absorber, depressing the fission density in the fuel pin. Figure 9 shows the production of ^{135}Xe as calculated by SERPENT and APIDA, as well as the relative difference over several burnup steps.

Figure 9. Production of ^{135}Xe as a function of burnup resulting from SERPENT and APIDA calculations.

Another factor in criticality is the enrichment of ^{235}U in UO_2 fuel. Figure 10 shows the change in ^{235}U atomic density in the fuel pin cell over several burnup steps. The relative difference between SERPENT and APIDA oscillates around zero, bounded by approximately 0.05%.

Figure 10. Change in ^{235}U as a function of burnup resulting from SERPENT and APIDA calculations.

Finally, a parameter with significance to nuclear nonproliferation and national security is the ratio of ^{240}Pu to ^{239}Pu . This ratio is key in determining the viability of plutonium as a special nuclear material for weapons production. Figure 11 shows the ratio of ^{240}Pu to ^{239}Pu as a function of burnup.

Figure 11. Ratio of ^{240}Pu to ^{239}Pu as a function of burnup resulting from SERPENT and APIDA calculations.

Initially divergent, the relative difference starts to approach zero as the fuel pin is burned more. This is a phenomenon which should be investigated further, as this ratio of interest is particularly important during the early stages of burnup when the amount of ^{239}Pu is still dominant and makes the plutonium more amenable for weapons production. The likely culprit in this analysis is the extremely low atomic density at the first time-step (on the order of 1×10^{-10}). As mentioned earlier, the standard deviations associated with the eigenvalue results are also relatively high, indicating relatively large uncertainty in the material with low concentrations. Uncertainties in material concentrations are not calculated by SERPENT – material concentration uncertainty propagation is an area of future interest for applications of APIDA when coupled to stochastic transport methods.

5. Discussion of Results and Concluding Remarks

In this study, a robust, computationally very efficient, and portable burnup tool was developed with the capability to easily couple with any transport solver in memory to efficiently perform lattice depletion calculations. The new code, APIDA, employs a novel hybrid burnup

solver and presents completely new code module capable of seamless integration within independently developed transport solvers.

The APIDA code was numerically validated with Mathematica to benchmark the burnup tool and its ability to calculate material concentrations after multiple time steps. The solutions produced by the APIDA code resulted in a relative percent error of well below 1% for all nuclides, even with the most numerically taxing problem descriptions.

The APIDA framework was built from the ground up as an object-oriented API in the C++ language and the pertinent public classes are available for users for a variety of applications.

The APIDA code was also coupled with the SERPENT code to track the change in eigenvalue of a pin cell over multiple time steps. The fluxes generated by SERPENT were fed into APIDA to generate new material concentrations for the subsequent transport calculation. The results generated by APIDA showed good agreement with the SERPENT reference solution and maintained high accuracy in terms of tracking the change in nuclide concentration over time. APIDA was shown to provide accurate solutions for some of the more important parameters related to criticality and nuclear nonproliferation, namely the buildup of ^{135}Xe and the ratio of ^{240}Pu to ^{239}Pu .

Overall, the capabilities of APIDA were shown to be both powerful and easy to implement. Future work for APIDA includes the implementation of a more general library reader class to handle raw ENDF files as APIDA currently uses the processed ORIGEN data libraries (Chadwick et. al., 2006 and Gauld and Wiarada, 2009). The methods in APIDA can also be expanded to handle some of the lesser occurring but still present neutron-induced reactions not considered in this study. The APIDA code should also be integrated with a massively parallel transport code to validate the thread-safe nature of the API framework in APIDA.

Acknowledgements

The first author would like to acknowledge this research was performed using funding received from the U.S. Department of Energy Office of Nuclear Energy's Nuclear Energy University Programs. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the U.S. Department of Energy Office of Nuclear Energy.

References

AREVA (2013) - U.S. EPR Final Safety Analysis Report - AREVA Design Control Document Rev. 5 - Tier 2 Chapter 04 - Reactor - Section 4.3 Nuclear Design.

AREVA (2013) - U.S. EPR Final Safety Analysis Report - AREVA Design Control Document Rev. 5 - Tier 2 Chapter 01 - Introduction and General Description of the Plant - Section 1.2 General Plant Description.

Chadwick, M. B. and others (2006), “ENDF/B-VII.0: Next Generation Evaluated Nuclear Data Library for Nuclear Science and Technology,” Nuclear Data Sheets, vol.107, pp. 2931–3060.

Edenius M., Ekberg k., Forssén B., Knott D. (1995), “CASMO-4, A Fuel Assembly Burnup Program, User’s Manual,” STUDSVIK/SOA-95/1, Studsvik of America, Inc.

Gauld, I., Wiarda, D. (2009), “ORIGEN Data Libraries.” Scale manual. Oak Ridge National Laboratory.

Lago, D., Rahnema, F. (2016), “Development of a set of benchmark problems to verify methods for solving burnup equations.” Annals of Nuclear Energy, (<http://dx.doi.org/10.1016/j.anucene.2016.09.004>).

Lepannen, J. (2008), “PSG2/Serpent—A Continuous-Energy Monte Carlo Reactor Physics Burnup Calculation Code,” VTT Technical Research Centre of Finland.

MATLAB version 7.14.0., (2012), Natick, Massachusetts: The MathWorks Inc.

Ouisloumen M. and Tahara Y. (1998), “PARAGON: The New Westinghouse Lattice Code”, Int. Conf. On the Physics of Nuclear Science and Technology, Long Island, NY.

Pusa, M. (2010), “Numerical methods for nuclear fuel burnup calculations.” Ph.D. Thesis. Aalto University. Finland.

Pusa, M. (2011), “Rational Approximations to the Matrix Exponential in Burnup Calculations,” Nuclear Science and Engineering, vol. 169, no. 2, pp. 155–167.

Pusa, M., Leppanen, J. (2010), “Computing the Matrix Exponential in Burnup Calculations.” Nuclear Science and Engineering, 164(2), 140-150.

Simeonov, T., (2003). “Release Notes – Helios System Version 1.8,” Studsvik Scandpower Report, SSP-03/221.

Wolfram Research, Inc. (2016), Mathematica, Version 10.4, Champaign, IL.

Appendix A: List of Isotopes Used in APIDA calculations

Table A1. Isotopes in APIDA-738 calculation.

10010	310780	330830	350870	370930	390960	410970	421080	441120	461120
10020	310790	330840	350880	370940	390961	410971	421090	441130	461130
10030	310800	330850	350890	370950	390970	410980	430980	451030	461140
20030	310810	330860	350900	370960	390980	410981	430990	451031	461150
40090	310820	330870	350910	370970	390981	410990	430991	451040	461160
60120	320700	330880	350920	380840	390990	410991	431000	451050	461170
70140	320720	340760	360800	380860	391000	411000	431010	451051	461180
80160	320730	340770	360820	380870	391010	411001	431020	451060	461190
80170	320740	340780	360830	380880	391020	411010	431021	451061	471070
280610	320750	340790	360831	380890	400900	411020	431030	451070	471090
280620	320760	340791	360840	380900	400910	411021	431040	451080	471091
280640	320770	340800	360850	380910	400920	411030	431050	451081	471101
280660	320771	340810	360851	380920	400930	411040	431060	451090	471110
290630	320780	340811	360860	380930	400940	411041	431070	451100	471111
290650	320790	340820	360870	380940	400950	411050	431080	451101	471120
290670	320791	340830	360880	380950	400960	411060	431090	451110	471130
300690	320800	340831	360890	380960	400970	420920	431100	451120	471131
300720	320810	340840	360900	380970	400980	420940	440980	451130	471140
300740	320820	340850	360910	380980	400990	420950	440990	451140	471150
300750	320830	340860	360920	380990	401000	420960	441000	451150	471151
300760	320840	340870	360930	381000	401010	420970	441010	451160	471160
300770	320860	340880	360940	390890	401020	420980	441020	461020	471161
300780	330750	340890	370850	390900	401030	420990	441030	461040	471170
310690	330760	350790	370860	390901	401040	421000	441040	461050	471171
310710	330770	350810	370870	390910	401050	421010	441050	461060	471180
310720	330780	350820	370880	390911	410930	421020	441060	461070	471181
310730	330790	350830	370890	390920	410931	421030	441070	461080	471190
310740	330800	350840	370900	390930	410940	421040	441080	461090	471200
310750	330810	350841	370901	390931	410950	421050	441090	461100	471210
310760	330820	350850	370910	390940	410951	421060	441100	461110	471221
310770	330821	350860	370920	390950	410960	421070	441110	461111	481060

481080	491191	501190	511261	521330	541290	551390	571410	591441	611490
481100	491200	501191	511270	521331	541300	551400	571420	591450	611500
481110	491201	501200	511280	521340	541310	551410	571430	591460	611510
481120	491210	501210	511281	521350	541311	551420	571440	591470	611520
481130	491211	501211	511290	521360	541320	551430	571450	591480	611521
481131	491220	501220	511300	521370	541330	551440	571460	591481	611530
481140	491221	501230	511301	521380	541331	551450	571461	591490	611540
481150	491230	501231	511310	521390	541340	551460	571470	591500	611541
481151	491231	501240	511320	521400	541341	561300	571480	591510	611550
481160	491240	501250	511321	531270	541350	561320	571490	591520	611560
481170	491241	501251	511330	531280	541351	561330	571500	591530	611570
481171	491250	501260	511340	531290	541360	561340	581380	591540	611580
481180	491251	501270	511341	531300	541370	561350	581390	601420	621470
481190	491260	501271	511350	531301	541380	561351	581400	601430	621480
481191	491261	501280	511360	531310	541390	561360	581410	601440	621490
481200	491270	501281	511370	531320	541400	561370	581420	601450	621500
481210	491271	501290	521220	531321	541410	561371	581430	601460	621510
481211	491280	501291	521230	531330	541420	561380	581440	601470	621520
481220	491281	501300	521240	531331	541430	561390	581450	601480	621530
481230	491290	501301	521250	531340	541440	561400	581460	601490	621540
481240	491291	501310	521251	531341	551320	561410	581470	601500	621550
481250	491300	501311	521260	531350	551330	561420	581480	601510	621560
481260	491310	501320	521270	531360	551340	561430	581490	601520	621570
481300	501120	501330	521271	531361	551341	561440	581500	601530	621580
491130	501130	501340	521280	531370	551350	561450	581510	601540	621590
491150	501140	511210	521290	531380	551351	561460	581520	601550	621600
491151	501150	511220	521291	531390	551360	561470	581530	601560	631510
491170	501160	511230	521300	531400	551361	561480	591410	611460	631520
491171	501170	511240	521310	531410	551370	571380	591420	611470	631530
491180	501171	511250	521311	541260	551380	571390	591430	611480	631540
491190	501180	511260	521320	541280	551381	571400	591440	611481	631550

631560	671650	922340	962430
631570	671660	922350	962440
631580	671661	922360	962450
631590	671670	922370	
631600	681640	922380	
631610	681660	922390	
641520	681660	922400	
641530	681670	922410	
641540	681680	932350	
641550	681690	932360	
641560	681700	932370	
641570	882230	932380	
641580	882240	932390	
641590	882250	932400	
641600	882260	942360	
641610	892260	942370	
641620	892270	942380	
651590	902270	942390	
651600	902280	942400	
651610	902290	942410	
651620	902300	942420	
651630	902310	942430	
651640	902320	942440	
661580	902330	952410	
661600	902340	952420	
661610	912310	952421	
661620	912320	952430	
661630	912330	952440	
661640	912340	952441	
661650	922320	962410	
661660	922330	962420	

Table A2. Additional Isotopes in APIDA-1049 calculation.

290690	290701	350821	421120	471050	511201	591400	661570	822090	902260
260590	290710	350930	421130	471060	511221	591421	661590	822100	912300
260660	290720	350940	421140	471071	511241	591560	661651	822110	912350
260670	290730	350950	430960	471080	511380	591570	661680	822120	922300
260680	290740	360810	430970	471100	511390	601410	661690	822140	922310
260690	290750	360950	431120	471201	521210	601580	661700	832100	932340
270610	290760	360970	431130	471220	521231	601590	661710	832110	932410
270620	290770	360980	431140	471230	521410	611590	671620	832120	942371
270660	290780	370830	431150	471240	531250	611600	671630	832130	942450
270670	290790	370840	431160	471250	531260	611610	671631	832140	952400
270680	300691	370861	431170	471260	531420	621460	671640	842100	952450
270690	300800	370980	441160	471270	531430	621610	671641	842110	
270700	300810	370990	441170	471280	531440	621620	671680	842120	
270710	300820	380850	441180	481070	541270	621630	671690	842130	
270720	310700	380871	441190	481090	541450	631500	671700	842140	
270730	310741	381010	441200	481111	541460	631521	671701	842150	
280630	310830	381020	451010	481231	551300	631541	671710	842160	
280650	310840	381030	451020	481280	551310	631620	671720	842180	
280670	310850	390891	451041	481290	551470	631630	681630	852170	
280680	320710	391040	451170	481310	551480	631640	681650	852180	
280690	320731	391050	451180	481320	561310	631650	681671	862180	
280700	320751	401060	451190	491110	561361	641510	681710	862190	
280710	320870	401070	451200	491120	561490	641640	681720	862200	
280720	320880	401080	451210	491131	561500	641650	691710	862220	
280730	330890	410920	451220	491140	561510	641660	691720	872210	
280740	330900	410941	461030	491161	571370	641670	802060	872220	
280750	340771	411080	461071	491181	571520	651570	812060	872230	
280760	340910	411090	461091	491330	571530	651580	812070	882220	
290660	340920	411100	461210	491340	581370	651660	812080	882270	
290680	340930	411110	461220	501350	581391	651670	812090	882280	
290681	350800	420930	461230	501360	581540	651680	812100	892250	
290700	350801	421110	461240	511200	581550	651690	822080	892280	