Nuclide Depletion Capabilities in the Shift Monte Carlo Code☆

Gregory G. Davidson[a,1,*], Tara M. Pandya[a,1], Seth R. Johnson[a,1], Thomas M. Evans[a,1], Aarno E. Isotalo[b], Cole A. Gentry[a,2], William A. Wieselquist[a,2]

[a] *Oak Ridge National Laboratory, 1 Bethel Valley Rd., Oak Ridge, TN 37831 U.S.A.*
[b] *Aalto University, P.O. Box 14100, FI-00076, Aalto, Finland*

Abstract

A new depletion capability has been developed in the Exnihilo radiation transport code suite. This capability enables massively parallel domain-decomposed coupling between the Shift continuous-energy Monte Carlo solver and the nuclide depletion solvers in ORIGEN to perform high-performance Monte Carlo depletion calculations. This paper describes this new depletion capability and discusses its various features, including a multi-level parallel decomposition, high-order transport-depletion coupling, and energy-integrated power renormalization. Several test problems are presented to validate the new capability against other Monte Carlo depletion codes, and the parallel performance of the new capability is analyzed.

Keywords: Monte Carlo, burnup, parallel scaling, transport-depletion coupling

☆Notice: This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

*Corresponding Author. Tel: +1 865 574 3921

Email addresses: davidsongg@ornl.gov (Gregory G. Davidson), pandyatm@ornl.gov (Tara M. Pandya), johnsonsr@ornl.gov (Seth R. Johnson), evanstm@ornl.gov (Thomas M. Evans), gentryca@ornl.gov (Cole A. Gentry), wieselquiwa@ornl.gov (William A. Wieselquist)

[1]HPC Methods and Applications Team, Reactor and Nuclear Systems Division

[2]Reactor Physics Group, Reactor and Nuclear Systems Division

1. Introduction

Early Monte Carlo depletion codes have typically coupled a stand-alone Monte Carlo transport code (such as MCNP [1]) to a stand-alone depletion code (such as ORIGEN [2] or CINDER [3]). The coupling module operated by creating the input files necessary to run the Monte Carlo code, extracting the various tallies from the output, and then creating the input file for the depletion code. The updated nuclide number densities were then extracted from the depletion output, a new transport input file was created, and the process would begin again for the next depletion step. Codes using this type of file-based transport-depletion coupling system include VESTA [4] and Monteburns [5].

There are numerous drawbacks to this type of Monte Carlo depletion coupling. The most significant is that requiring each module to write its input and output to disk for each depletion step can be a significant performance bottleneck and, especially, limits the parallel scaling of the Monte Carlo depletion algorithm. Scaling such a system to $O(100k)$ cores would be difficult given typical disk throughput.

More recently, nuclide depletion capabilities have been directly integrated into Monte Carlo simulation codes. Examples include MC21 [6], MCNP6 [7], and Serpent [8]. By coupling a depletion calculation inline to the Monte Carlo solver, expensive writes to disk can be avoided. Moreover, the depletion package can easily exploit any multilevel parallel decomposition present in the Monte Carlo package.

This paper discusses a new nuclide depletion capability that has been implemented into the Exnihilo radiation transport suite Exnihilo includes the Shift Monte Carlo solver [9] (containing both continuous-energy and multigroup physics), as well as the Denovo deterministic solver [10] (containing S_N, SP_N, and method of characteristics (MOC)-based deterministic transport). Currently, the depletion package has only been coupled to the continuous-energy Monte Carlo transport solver in Shift. However, the depletion package was designed to make coupling to a multigroup physics package (either Denovo or multigroup Shift) straightforward. The depletion package in Exnihilo was built to scale well for large problems on high-performance supercomputers.

Two applications have provided most of the motivation behind the development of the continuous-energy Monte Carlo depletion capability in Exnihilo. The first, as part of the Consortium for Advanced Simulation of Light Water Reactors (CASL) energy innovation hub [11], is to provide verification calculations for lattice depletion simulations performed using MPACT [12]. Examples of these calculations are provided in Sections 7.2 and 7.3. The other primary motivating application for this capability is faster fuel-cycle depletion analysis for the High Flux Isotope Reactor (HFIR) facility at the Oak Ridge National Laboratory (ORNL) [13]. Therefore, most of the methods and results contained in this paper are directed toward these applications.

The nuclide depletion module couples the transport solvers in Exnihilo to ORIGEN [2]. ORIGEN is a general purpose nuclide depletion solver supporting fuel depletion, target activation, and decay calculations. Therefore, while development efforts have been directed towards problems of interest for CASL and HFIR, this capability is expected to be used for a wide variety of applications in the future.

The purpose of this paper is to provide a complete description of the current capabilities of the nuclide depletion package implemented into Exnihilo, to demonstrate and compare the accuracy, efficiency, and performance of these capabilities within a common code framework on several test problems, and to collect into one place a pedagogical description of Monte Carlo-depletion coupling. The numerical methods described herein have been previously published elsewhere [9, 14, 15, 16, 17, 18], as have some of the numerical results [18, 19]. Results not previously published include fuel cycle simulations using Exnihilo for a detailed HFIR model, as described in Sections 7.3.2 and 7.3.3, as well as the parallel scaling study in Section 7.4.

The remainder of this document is organized as follows: in Section 2 the coupled radiation transport and nuclide depletion equations are derived; in Section 3 the multilevel parallelism used in the depletion package within Exnihilo is discussed; in Section 4 the various transport-depletion coupling methods implemented in Exnihilo are presented; Section 5 described the calculation of reaction rates, and Section 6 discusses the various power normalization methods that have been implemented; Section 7 presents simulation results demonstrating the accuracy of the Exnihilo depletion capability and parallel scaling; and Section 8 concludes and discusses future work.

2. Theory and Background

Monte Carlo nuclear reactor depletion calculations solve a coupled quasi-static system of equations involving the k-eigenvalue radiation transport equation:

$$\begin{aligned}\hat{\boldsymbol{\Omega}} \cdot \nabla\psi(\boldsymbol{x}, \hat{\boldsymbol{\Omega}}, E, t) + \Sigma_t(\boldsymbol{x}, E, t)\psi(\boldsymbol{x}, \hat{\boldsymbol{\Omega}}, E, t) \\ = \int_0^\infty \int_{4\pi} \Sigma_s(\boldsymbol{x}, \hat{\boldsymbol{\Omega}}' \cdot \hat{\boldsymbol{\Omega}}, E' \to E, t)\psi(\boldsymbol{x}, \hat{\boldsymbol{\Omega}}', E', t)\,\mathrm{d}\Omega'\mathrm{d}E' \\ + \frac{1}{k_{\text{eff}}(t)} \frac{\chi(\boldsymbol{x}, E, t)}{4\pi} \int_0^\infty \nu\Sigma_f(\boldsymbol{x}, E', t)\phi(\boldsymbol{x}, E', t)\,\mathrm{d}E'\,, \quad \text{(1a)}\end{aligned}$$

and the Bateman equations:

$$\begin{aligned}\frac{\partial}{\partial t} N_j(\boldsymbol{x}, t) = \sum_{\substack{k \in J \\ k \neq j}} \left[\sum_r \gamma_{r,k,j}\sigma_{r,k}(\boldsymbol{x}, t)\phi(\boldsymbol{x}, t) + b_{k,j}\lambda_k \right] N_k(\boldsymbol{x}, t) \\ - \left[\sum_r \sigma_{r,j}(\boldsymbol{x}, t)\phi(\boldsymbol{x}, t) + \lambda_j \right] N_j(\boldsymbol{x}, t)\,, \qquad j \in J\,. \quad \text{(1b)}\end{aligned}$$

Equation (1a) is an eigenvalue equation describing the population of neutrons in a multiplying medium such as a nuclear reactor. The fundamental unknowns in Eq. (1a) are $\psi(\boldsymbol{x}, \hat{\boldsymbol{\Omega}}, E, t)$, the eigenfunction, which represents the angular neutron flux at position $\boldsymbol{x}$ traveling in direction $\hat{\boldsymbol{\Omega}}$ in energy E at time t; and $k_{\text{eff}}(t)$, the eigenvalue, which physically represents the factor by which the neutron population is multiplied each generation. A k_{eff} of unity represents a constant population. $\Sigma_t(\boldsymbol{x}, E, t)$ is the total macroscopic cross section at position $\boldsymbol{x}$ and energy E at time t, while $\Sigma_s(\boldsymbol{x}, \hat{\boldsymbol{\Omega}}' \cdot \hat{\boldsymbol{\Omega}}, E' \to E, t)$ is the macroscopic scattering cross section for neutrons scattering through angle $\hat{\boldsymbol{\Omega}}' \cdot \hat{\boldsymbol{\Omega}}$ from energy E' to E at position $\boldsymbol{x}$ and time t. $\chi(\boldsymbol{x}, E, t)$ is the probability of a fission neutron being emitted at energy E at position $\boldsymbol{x}$ at time t, ν is the number of neutrons emitted per fission, and $\Sigma_f(\boldsymbol{x}, E', t)$ is the macroscopic fission cross section. Finally, $\phi(\boldsymbol{x}, E, t)$ is the scalar flux, defined as

$$\phi(\boldsymbol{x}, E, t) = \int_{4\pi} \psi(\boldsymbol{x}, \hat{\boldsymbol{\Omega}}, E, t)\,\mathrm{d}\Omega\,. \quad (2)$$

The quasistatic approximation is valid when the change in nuclide concentrations is very slow relative to a neutron lifetime. Equation (1a) is also accompanied with associated boundary conditions, which for simplicity we have omitted.

Equation (1b) is the general Bateman equation, representing the evolution of the atomic density of a given nuclide j. $N_j(\boldsymbol{x}, t)$ represents the atomic density of nuclide j at position $\boldsymbol{x}$ and time t. The first term in Eq. (1b) represents the production of nuclide j from nuclide k for various reactions r, where the yield $\gamma_{r,k,j}$ represents the probability that a reaction r will transform nuclide k into nuclide j, $\phi(\boldsymbol{x}, t)$ represents the energy-integrated scalar flux at position $\boldsymbol{x}$ and time t, and $\sigma_{r,k}(\boldsymbol{x}, t)$ represents the energy-integrated, flux-weighted cross section for reaction r and nuclide k at position $\boldsymbol{x}$ and time t, i.e.,

$$\sigma_{r,k}(\boldsymbol{x}, t) = \frac{\int_0^\infty \sigma_{r,k}(E)\phi(\boldsymbol{x}, E, t)\,\mathrm{d}E}{\int_0^\infty \phi(\boldsymbol{x}, E, t)\,\mathrm{d}E}\,. \tag{3}$$

All of the reactions modeled by ORIGEN are given in Appendix A. Quantity $b_{k,j}$ represents the branching ratio from nuclide k into nuclide j, i.e., the probability that nuclide j will be produced in the decay of nuclide k, and λ_k is the decay constant for nuclide k. The second term in Eq. (1b) represents losses of nuclide j through various reactions r as well as losses due to decay. Other loss mechanisms are possible, such as loss through chemical separation in the case of molten-salt reactors. However, in this paper we assume a solid-fueled reactor with only the loss mechanisms described above. J represents the set of all nuclides to be simulated. Equation (1b) also has associated initial conditions which have been omitted for brevity, and the yields $\gamma_{r,k,j}$ in Eq. (1b) are represented as energy-independent for simplicity, though ORIGEN supports energy-dependent yields.

Equations (1a) and (1b) are coupled through the neutron flux and cross sections. The macroscopic cross sections in Eq. (1a) are defined as:

$$\Sigma_t(\boldsymbol{x}, E, t) = \sum_{j\in J} N_j(\boldsymbol{x}, t)\left[\sigma_{a,j}(E) + \sigma_{s,j}(E)\right]\,, \tag{4a}$$

$$\Sigma_s(\boldsymbol{x}, \hat{\boldsymbol{\Omega}}' \cdot \hat{\boldsymbol{\Omega}}, E' \to E, t) = \sum_{j\in J} N_j(\boldsymbol{x}, t)\sigma_{s,j}(\hat{\boldsymbol{\Omega}}' \cdot \hat{\boldsymbol{\Omega}}, E' \to E)\,, \tag{4b}$$

$$\Sigma_f(\boldsymbol{x}, E, t) = \sum_{j\in J} N_j(\boldsymbol{x}, t)\sigma_{f,j}(E)\,, \tag{4c}$$

where $\sigma_{a,j}(E)$ and $\sigma_{s,j}(E)$ are the microscopic absorption and scattering cross sections for nuclide j, respectively.

Equations (2), (3) and (4) describe how Eqs. (1a) and (1b) are coupled. The macroscopic cross sections described in Eq. (4) and used in Eq. (1a) are coupled to the nuclide number densities calculated using Eq. (1b). The energy-integrated flux calculated using Eqs. (1a) and (2) and the flux-weighted cross sections described by Eq. (3) are used in Eq. (1b).

2.1. Spatial Discretization

Both Eqs. (1a) and (1b) are continuous in space. A spatial discretization must be applied to solve these equations. Spatial regions of interest within the problem domain (termed depletion regions) are defined in which the neutron flux and average nuclide atomic densities are calculated. The average number density for nuclide j within a depletion region i defined by R_i is defined as

$$N_{i,j}(t) = \frac{1}{V_i} \int_{\boldsymbol{x} \in R_i} N_j(\boldsymbol{x}, t) \, \mathrm{d}^3\boldsymbol{x} \,, \tag{5}$$

where V_i is the volume of depletion region i. It is assumed that depletion regions do not span material boundaries. Operating on Eq. (1b) by $\frac{1}{V_i} \int_{\boldsymbol{x} \in R_i} (\cdot) \, \mathrm{d}^3\boldsymbol{x}$, we obtain

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} N_{i,j}(t) = \sum_{\substack{k \in J \\ k \neq j}} \left[\sum_r \gamma_{r,j,k} \sigma_{r,i,k}(t) \phi_i(t) + b_{k,j} \lambda_k \right] N_{i,k}(t) \\ - \left[\sum_r \sigma_{r,i,j}(t) \phi_i(t) + \lambda_j \right] N_{i,j}(t) \,, \end{aligned} \tag{6}$$

where we have used

$$\frac{1}{V_i} \int_{\boldsymbol{x} \in R_i} \sigma_{r,j}(\boldsymbol{x}, t) \phi(\boldsymbol{x}, t) N_j(\boldsymbol{x}, t) \, \mathrm{d}^3\boldsymbol{x} \approx \sigma_{r,i,j}(t) \phi_i(t) N_{i,j}(t) \,, \tag{7}$$

where $\sigma_{r,i,j}(t)$ is the average cross section for reaction r and nuclide j at time t in depletion region i, and $\phi_i(t)$ is the average scalar flux in depletion region i at time t. This approximation is exact when the neutron flux is spatially flat across depletion region i. In practice, it is accurate as long as each depletion region is small enough that the gradient of the flux across the region is modest.

2.2. Time Discretization

While Eq. (1a) describes a transport problem that can be solved directly for any given time t, Eq. (6) is a first-order ordinary differential equation in time. Equation (6) can be rewritten in matrix form as

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{N}_i(t) = \mathsf{A}_i(t)\boldsymbol{N}_i(t)\,, \tag{8}$$

where $\boldsymbol{N}_i(t)$ is the vector of all nuclide number densities in region i and $\mathsf{A}_i(t)$ is the transition matrix in depletion region i. Thus, the element (j,k) corresponding to nuclides j and k in the transition matrix $\mathsf{A}_i(t)$ is defined as

$$\mathsf{A}_{i,j,k}(t) = \begin{cases} -\sum_r \sigma_{r,i,j}(t)\phi_i(t) - \lambda_j\,, & j = k\,, \\ \sum_r \gamma_{r,i,j}\sigma_{r,i,j}(t)\phi_i(t) + b_{k,j}\lambda_k\,, & j \neq k\,. \end{cases} \tag{9}$$

Consider a time step m bounded by times t_m and t_{m+1}. Coupled transport-depletion problems are typically solved by computing transport solutions at the beginning and ending of time steps, i.e., at times $t_0, \ldots, t_{M+1}$ for time steps $m = 0, \ldots, M$. Furthermore, time step m is subdivided into S substeps, such that substep s within time step m is bounded by times $t_{m,s}$ and $t_{m,s+1}$, where $t_{m,0} = t_m$ and $t_{m,S+1} = t_{m+1}$. Depletion solutions are computed over each substep. Many predictor-corrector coupling methods (see Section 4.3), as well as methods employing substep power renormalization (see Section 6), benefit from improved accuracy when using multiple substeps per time step.

The objective is to solve Eq. (8) over substep s in time step m. For coupling methods such as the fully explicit method (see Section 4.1) and the middlestep method (see Section 4.2), the transition matrix is approximated as constant through the time step, and thus Eq. (8) can be rewritten as

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{N}_i(t) = \bar{\mathsf{A}}_i^m \boldsymbol{N}_i(t)\,, \qquad t_m \leq t \leq t_{m+1}\,, \tag{10}$$

where $\bar{\mathsf{A}}_i^m$ represents an average transition matrix for time step m approximated according to the assumptions of the transport-depletion coupling method. Equation (10) can be solved to find

$$\boldsymbol{N}_i^{m,s+1} = \exp\left[\bar{\mathsf{A}}_i^m \Delta t_{m,s}\right] \boldsymbol{N}_i^{m,s}\,, \tag{11}$$

where $\boldsymbol{N}_i^{m,s} = \boldsymbol{N}_i(t_{m,s})$. The methods used to approximate $\bar{\mathsf{A}}_i^m$ will be described in Sections 4.1 and 4.2.

For predictor-corrector methods employing non-constant interpolations and extrapolations (described in Section 4.3), the transition matrices are not assumed to be constant over the substep. In this case Eq. (8) does not directly reduce to Eq. (10). There is no closed-form solution to Eq. (8), but the solution can be expressed as an infinite Magnus series [20]:

$$\boldsymbol{N}_i^{m,s+1} = \exp\left[\sum_{k=1}^{\infty} \Omega_{i,k}(t_{m,s+1}, t_{m,s})\right] \boldsymbol{N}_i^{m,s}\,, \tag{12}$$

where the first three terms of the Magnus series are given by [21]:

$$\Omega_{i,1}(t_{m,s+1}, t_{m,s}) = \int_{t_{m,s}}^{t_{m,s+1}} \mathsf{A}_i(\tau)\,\mathrm{d}\tau\,, \tag{13a}$$

$$\Omega_{i,2}(t_{m,s+1}, t_{m,s}) = \frac{1}{2}\int_{t_{m,s}}^{t_{m,s+1}} \int_{t_{m,s}}^{\tau_1} [\mathsf{A}_i(\tau_1), \mathsf{A}_i(\tau_2)]\,\mathrm{d}\tau_1\mathrm{d}\tau_2\,, \tag{13b}$$

$$\begin{aligned}\Omega_{i,3}(t_{m,s+1}, t_{m,s}) = \frac{1}{6}\int_{t_{m,s}}^{t_{m,s+1}} \int_{t_{m,s}}^{\tau_1} \int_{t_{m,s}}^{\tau_2} \big(&[\mathsf{A}_i(\tau_1), [\mathsf{A}_i(\tau_2), \mathsf{A}_i(\tau_3)]] \\ &+ [\mathsf{A}_i(\tau_3), [\mathsf{A}_i(\tau_2), \mathsf{A}_i(\tau_1)]] \big)\,\mathrm{d}\tau_1\mathrm{d}\tau_2\mathrm{d}\tau_3\,,\end{aligned} \tag{13c}$$

where $[\mathsf{A}, \mathsf{B}] \equiv \mathsf{AB} - \mathsf{BA}$ is the matrix commutator of matrices A and B.

When solving Eq. (8), we choose to truncate the Magnus series at the first term. Then the approximate solution to Eq. (8) can be written as

$$\boldsymbol{N}_i^{m,s+1} \approx \exp\left[\bar{\mathsf{A}}_i^{m,s} \Delta t_{m,s}\right] \boldsymbol{N}_i^{m,s}\,, \tag{14}$$

where $\bar{\mathsf{A}}_i^{m,s}$ is the transition matrix for depletion region i averaged over substep s in time step m. For reasonably small substeps and good approximations of $\bar{\mathsf{A}}_i^{m,s}$, Eq. (14) is an accurate solution to Eq. (8). By comparing Eq. (14) with Eq. (11), we can see that Eq. (14) is the approximate solution to Eq. (8) and the exact solution to Eq. (10). The effect of the higher terms of the Magnus series on accuracy and practical substep lengths is deferred to future work.

Equation (14), together with initial conditions $\boldsymbol{N}_i^0 = \boldsymbol{N}_i(t_0)$, enable us to find the nuclide number densities in any depletion region i. Equation (14) is a matrix-exponential equation, and ORIGEN provides two different methods for solving this equation: the so-called Matrix Exponential Method

(MATREX) [2], and a new Chebyshev Rational Approximation Method (CRAM) [22]. Details of these solvers are provided in the references.

The substep-averaged transition matrix in Eq. (14) is calculated by substituting substep-averaged scalar fluxes and reaction cross sections into Eq. (9). For coupling methods that allow time-varying transition matrices, this implies an additional approximation:

$$\overline{\sigma_{r,i,j}^{m,s}\,\phi_i^{m,s}} \approx \bar{\sigma}_{r,i,j}^{m,s}\bar{\phi}_i^{m,s}\,. \tag{15}$$

This approximation is not strictly necessary; the reaction rates could be averaged instead of the cross sections and fluxes. However, this reflects the current implementation in Exnihilo, and the difference is negligible [14]. The various schemes implemented in Exnihilo to calculate approximate substep-averaged fluxes and reaction cross sections are described in Section 4.

There are two methods for calculating the region-averaged flux-weighted reaction cross sections defined by Eq. (3). The reaction rate can be tallied directly during Monte Carlo transport, and from this the cross section can be calculated. Conversely, a fine-bin flux can be estimated during transport, from which the cross section can be calculated. The former method uses less memory and has lower statistical variance, but is more computationally expensive. This cost is due to the required cross section interpolations necessary at every step during the random walk. For this reason, the results in this paper use the fine-bin flux method, as described in Section 5. A direct comparison of the methods will be the subject of a future study.

When performing fuel-cycle depletion problems, the transport equation given by Eq. (1a) is used. However, as an eigenvalue equation, it only produces the shape of the angular flux, but does not produce its magnitude. The magnitude of the flux is given by an auxiliary equation relating the flux to a user-specified reactor power. The reactor power equation, and its use in conjunction with coupled radiation transport-nuclide depletion equations, is described in Section 6.

3. Multilevel Parallelism in the Depletion Package

The Shift Monte Carlo code within the Exnihilo transport code suite provides a multilevel parallelism scheme through the Multiple-Set Overlapping Domain (MSOD) algorithm [23, 9]. The problem domain is decomposed into a Cartesian grid of (potentially) overlapping blocks. Then the block-decomposed geometry is replicated into "sets" over which particle histories are parallelized. The total number of domains (processes) is then

$$N_d = N_s \times N_b \,, \tag{16}$$

where N_s is the number of sets, and N_b is the number of blocks. Each set simulates $N_{p,s}$ histories, where

$$N_{p,s} = \frac{N_p}{N_s} \,, \tag{17}$$

and where N_p is the number of requested particles. This is illustrated in Figure 1.

During a k-eigenvalue transport calculation there are three types of communication [9]:

- intra-set communication of particles across domain boundaries during transport,
- intra-set and intra-block communication of fission sites between cycles, and
- intra-set and intra-block communication of mesh-based tallies at the end of the transport solve.

Reference [9] provides more details of the MSOD algorithm.

The depletion package within Exnihilo does not attempt to maximize parallelism by simply evenly distributing the depletion regions amongst all available processes since this would require that the communication of the depletion results be global. Rather, the depletion package exploits the multilevel parallelism of the MSOD algorithm to reduce the amount of memory and communication required during the solution of the depletion equations. The processors on a spatial block need only perform the depletion calculation on the depletion regions within their block. Within a block, the depletion regions are distributed evenly to each set. This maximizes the parallelism within a block while minimizing block-to-block communication. During the depletion calculation, there are two types of communication:

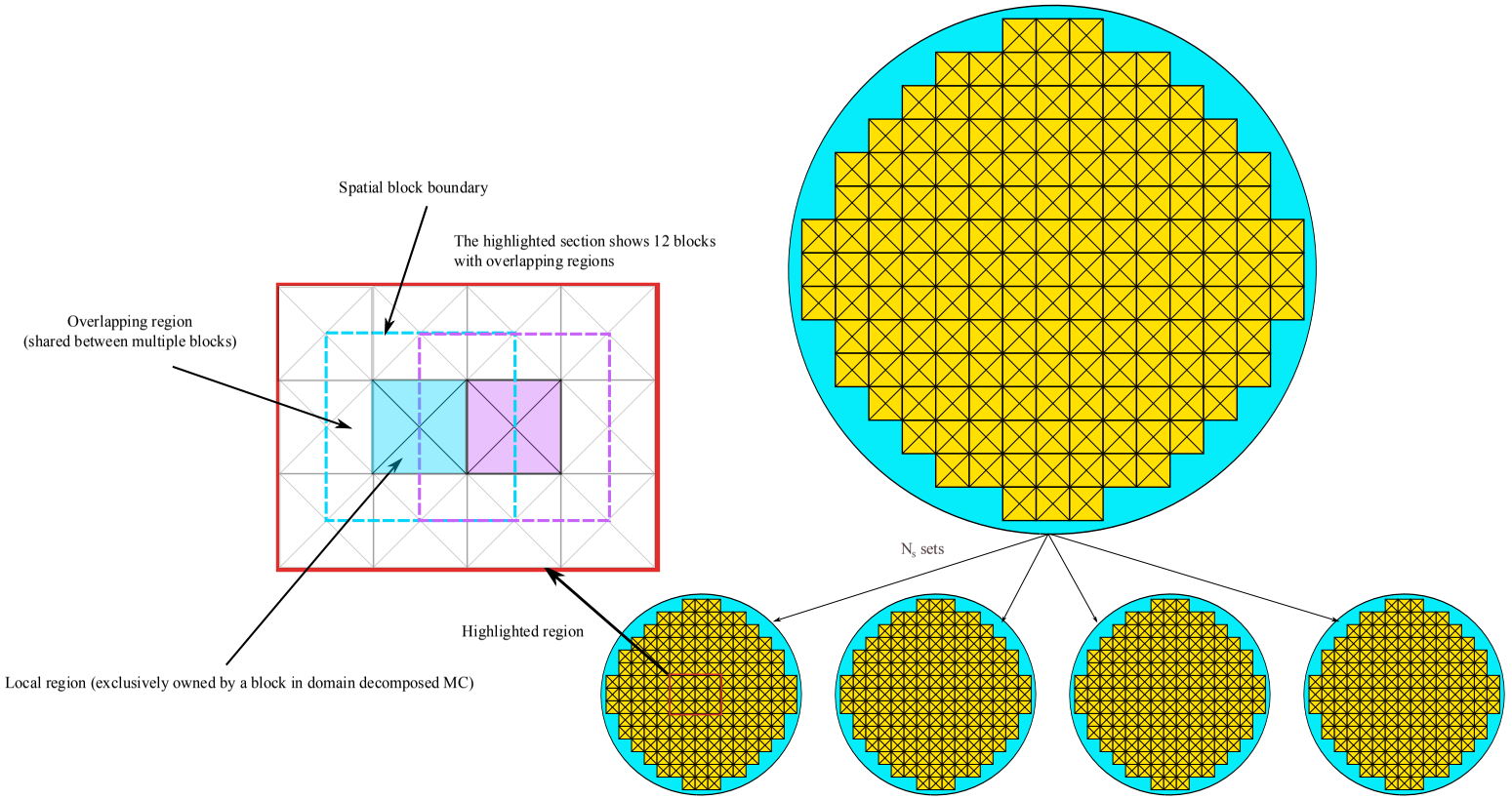

Figure 1. Example of MSOD geometry, in which $N_s = 4$ and the entire geometry is represented within each set [23].

- intra-set and intra-block communication during power normalization (see Section 6), potentially at every depletion substep, and
- intra-block communication to update material number densities at the completion of a depletion solve.

Two aspects of the depletion communication should be noted. First, while the communication during power normalization is a global communication across all blocks and sets and must (potentially) be performed at each depletion substep, only a single number (the global reactor power) is communicated. Thus, this communication is inexpensive and limited only by the network latency. Updating the number densities at the completion of a depletion solve requires the communication of substantially more data. However, this occurs only at the conclusion of a depletion solve (after all substeps have been performed), and it only requires an intra-block communication (no communication between blocks). Thus the distribution of work within a block is maximized, and the communication across blocks is minimized.

The depletion package should scale well if (1) the number of depletion regions is the same on each block, and (2) there is at least one depletion region per set on each block. The first condition is generally true for reactors since depletion regions are evenly distributed throughout the core. For modestly sized problems, it is possible to have more sets than depletion regions so the second condition may not be met. While this will reduce the parallel scalability of the depletion solve, the cost of the depletion solve under this condition will be very low relative to the cost of the transport solve. This is demonstrated in Section 7. We note that in this paper we limit ourselves to a single block with multiple sets. In future work the parallel scaling of the depletion package over multiple blocks will be demonstrated.

4. Transport-Depletion Coupling Methods

As stated in Section 2, time-approximations must be applied to calculate the scalar fluxes and reaction cross sections necessary to solve the depletion equations. A method is needed to form the substep-averaged transition matrix $\bar{\mathsf{A}}_i^{m,s}$ by calculating substep-averaged cross sections $\bar{\sigma}_{r,i,j}^{m,s}$ and scalar fluxes $\bar{\phi}_i^{m,s}$ and using Eq. (9).

The goal of any transport-depletion coupling method is to optimize time to solution by minimizing the number of transport calculations while maintaining accuracy. Because the computational cost of solving the depletion equations is typically small relative to the cost of solving the transport equation, it is usually advantageous to choose a coupling method that enables long time steps and fewer transport calculations, even if this potentially results in more depletion solves.

Several types of fuel cycle depletion coupling methods are supported in Exnihilo. These fall into three major classes: the fully explicit method (described in Section 4.1), the middlestep method (described in Section 4.2), and predictor-corrector coupling methods (described in Section 4.3). Comparisons of the accuracy of the various methods available in Exnihilo can be found in Refs. [24] and [25].

4.1. The Fully Explicit Method

The fully explicit transport-depletion coupling method is a forward Euler time discretization (also known as the constant-extrapolation method) that requires a single calculation of the transport solution per time step. Equation (1a) is evaluated at the beginning of the time step, and the resulting flux and cross sections are assumed to remain constant through the time step as the depletion equations are solved for each substep (although substep power normalization may cause the scalar flux to change somewhat between substeps; see Section 6 for details). In other words, we set

$$\bar{\sigma}_{r,i,j}^{m,s} = \sigma_{r,i,j}^{m} \equiv \sigma_{r,i,j}(t_m)\,, \tag{18a}$$

$$\bar{\phi}_i^{m,s} = \phi_i^{m,s} \equiv \phi_i(t_m)\,, \tag{18b}$$

and thus the substep-averaged transition matrix is assumed constant through the depletion step:

$$\bar{\mathsf{A}}_i^{m,s} = \mathsf{A}_i^{m} \equiv \mathsf{A}_i(t_m)\,. \tag{19}$$

While this is the simplest coupling method implemented in Exnihilo and requires only a single transport solve per time step, accurate solutions may require small time steps and therefore would result in more transport calculations.

4.2. Middlestep Method

The fully explicit coupling scheme given in Section 4.1 assumes that the fluxes and cross sections computed at the beginning of the time step hold constant throughout the time step. This requires problems in which the flux and number densities change slowly as a function of time, or problems in which the time steps are small. A more accurate method to calculate the time step-averaged cross sections and fluxes is the middlestep method. This method can be split into two parts. In the first part, we wish to calculate accurate time step-averaged fluxes and cross sections. In the second part, those fluxes and cross sections are used to deplete through the time step.

To calculate more accurate fluxes and cross sections, Eq. (1a) is solved at the beginning of time step m at time t_m, using the beginning-of-step number densities $\boldsymbol{N}_i^m$ to calculate the macroscopic cross sections using Eqs. (4). From the transport solution, *predictor* time-step averaged fluxes and cross sections using the methods given in Section 5 are calculated:

$$\bar{\sigma}_{r,i,j}^{m,\dagger} = \sigma_{r,i,j}^{m} \,, \tag{20a}$$

$$\bar{\phi}_{i}^{m,\dagger} = \phi_{i}^{m} \,. \tag{20b}$$

The fluxes are then normalized to $\bar{P}_R^m$, the reactor power over time step m using the methods given in Section 6, and the *predictor* transition matrices $\mathsf{A}_i^{m,\dagger}$ are calculated using Eq. (9). Depletion is then performed over substeps $s = 0, \ldots, (S_m/2-1)$ using Eq. (14) until the midpoint of time step m at time $t_{m+1/2}$ is reached, at which point middle-of-step *predictor* number densities $\boldsymbol{N}_i^{m+1/2,\dagger}$ have been calculated.

At this point, Eq. (1a) is solved again, using macroscopic cross sections evaluated with middle-of-step nuclide number densities $\boldsymbol{N}_i^{m+1/2,\dagger}$ and Eqs. (4). Using the transport solution at time $t_{m+1/2}$, *corrector* fluxes and cross sections are calculated using the methods given in Section 5, and these resultant fluxes and cross sections are then treated as time-step average quan-

tities:

$$\bar{\sigma}_{r,i,j}^{m} = \sigma_{r,i,j}^{m+1/2}\,, \tag{21a}$$

$$\bar{\phi}_{i}^{m} = \phi_{i}^{m+1/2}\,. \tag{21b}$$

Once again the fluxes are normalized to the reactor power and then used in Eq. (9) to find the time step-averaged transition matrix. Then the depletion over the whole time step is performed over all substeps $s = 0, \ldots, (S_m - 1)$ until time t_{m+1} is reached. This method requires two transport solves per time step. However, the improved estimate of the time step-averaged fluxes and cross sections enables much longer time steps for many problems.

4.3. Predictor-Corrector Methods

Exnihilo also features a suite of predictor-corrector coupling methods [14, 15]. These methods all require two transport calculations per time step. The complete set of predictor-corrector coupling methods available in Exnihilo is given in Table 1.

Coupling Method	**Predictor Extrapolation Method**	**Corrector Interpolation Method**
CE	constant	*none*
LE	linear	*none*
CE/LI	constant	linear
CE/QI	constant	quadratic
LE/LI	linear	linear
LE/QI	linear	quadratic

Table 1. Predictor-corrector transport-depletion coupling methods available in Exnihilo.

During the predictor step, a transport calculation is performed at the beginning of time step m using known nuclide number densities $\boldsymbol{N}_i^m$ to calculate the macroscopic cross sections given in Eqs. (4). From this transport solution fluxes and reaction cross sections at time t_m are calculated using the method given in Section 5. The fluxes and cross sections are then extrapolated from t_m (and t_{m-1} from the previous time step for linear extrapolation)

according to the particular extrapolation method for this predictor-corrector coupling scheme. For each depletion substep, the substep-averaged *predictor* fluxes $\bar{\phi}_i^{m,s,\dagger}$ and cross sections $\bar{\sigma}_{r,i,j}^{m,s,\dagger}$ are calculated, and then the *predictor* substep-averaged transition matrices $\bar{\mathsf{A}}_i^{m,s,\dagger}$ are calculated from Eq. (9). Then the *predictor* end-of-substep nuclide number densities $\boldsymbol{N}_i^{m,s+1,\dagger}$ can be calculated using Eq. (14). This is performed over each substep until the end of the time step at time t_{m+1} is reached. The predictor step ends with the execution of a predictor transport calculation using the predictor end-of-timestep nuclide number densities $\boldsymbol{N}_i^{m+1,\dagger}$ to calculate the macroscopic cross sections. This transport solution is used to calculate end-of-time step cross sections $\sigma_{r,i,j}^{m+1,\dagger}$ and fluxes $\phi_i^{m+1,\dagger}$, which must be normalized to the power level $\bar{P}_R^m$ using the methods in Section 6.

If the particular predictor-corrector method being used does not have a corrector step, we let $\boldsymbol{N}_i^{m+1} = \boldsymbol{N}_i^{m+1,\dagger}$, $\sigma_{r,i,j}^{m+1} = \sigma_{r,i,j}^{m+1,\dagger}$ and $\phi_i^{m+1} = \phi_i^{m+1,\dagger}$ and continue on to the next time step. Alternatively, if there is a corrector step, the fluxes and cross sections at the beginning of the time step are used, as well as the *predictor* fluxes and cross sections at the end of the time step, to re-deplete over the time step, interpolating substep-averaged fluxes and cross sections over each substep. If using quadratic interpolation, fluxes and cross sections from the previous time step at time t_{m-1} are also used. For each depletion substep, the substep-averaged *corrector* fluxes $\bar{\phi}_i^{m,s}$ and cross sections $\bar{\sigma}_{r,i,j}^{m,s}$ are calculated. Then the substep-averaged transition matrices are calculated, and end-of-substep nuclide number densities are obtained using Eq. (14) until the end of time step m is reached at time t_{m+1}. At this point we will have obtained end-of-time step nuclide number densities $\boldsymbol{N}_i^{m+1}$. The linear extrapolation/quadratic interpolation (LE/QI) predictor and corrector steps are illustrated in Figures 2 and 3, respectively.

4.4. Other Coupling Methods Featured in Exnihilo

Two additional coupling methods are available in Exnihilo. The first of these is called the TRITON [26] coupling method, because this method was first featured in the TRITON code, although it is also sometimes called the Ce/CM method in the literature. This method is similar to the middlestep method, but it does not perform a corrector transport calculation at the end of the time step. Instead, the fluxes and cross sections from the previous time step are used to deplete to the step midpoint. Then, a transport calculation is performed, and the resultant fluxes and cross sections are then used to

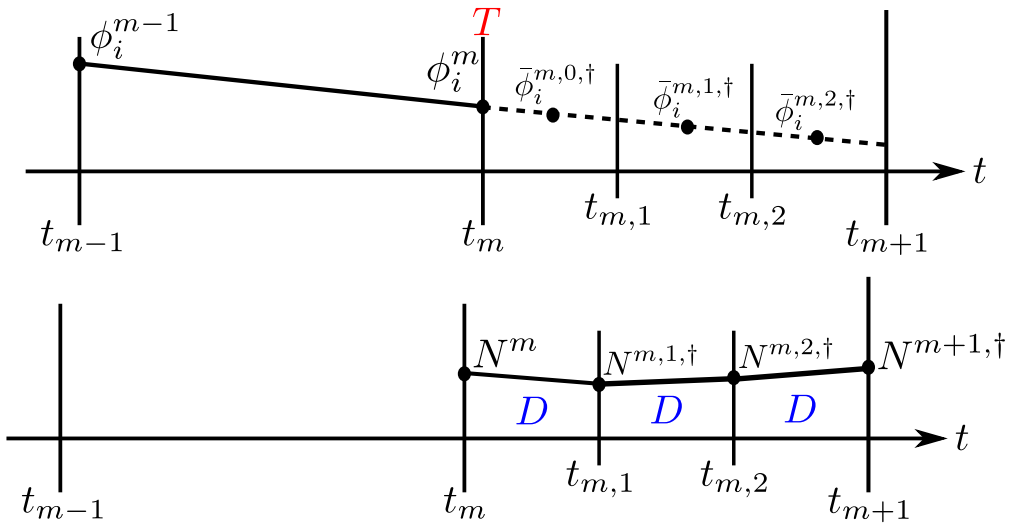

Figure 2. Illustration of the linear extrapolation/quadratic interpolation (LE/QI) algorithm for the predictor step, showing the transport component (*top*) and depletion component (*bottom*). Transport is performed at the beginning of the time step at time t_m, and flux ϕ_i^m is calculated along with cross sections $\sigma_{r,i,j}^m$ (not shown). Fluxes are then extrapolated from time t_{m-1} and t_m over time step m, and substep-averaged *predictor* fluxes $\bar{\phi}_i^{m,s,\dagger}$ are calculated for each substep. Similar extrapolations are performed to calculate substep-averaged *predictor* reaction cross sections $\bar{\sigma}_{r,i,j}^{m,s,\dagger}$ (not shown). These are used to perform depletion over each substep to find *predictor* number densities $N^{m,s,\dagger}$ until the end of the time step is reached. T represents a transport solve, and D represents a depletion solve.

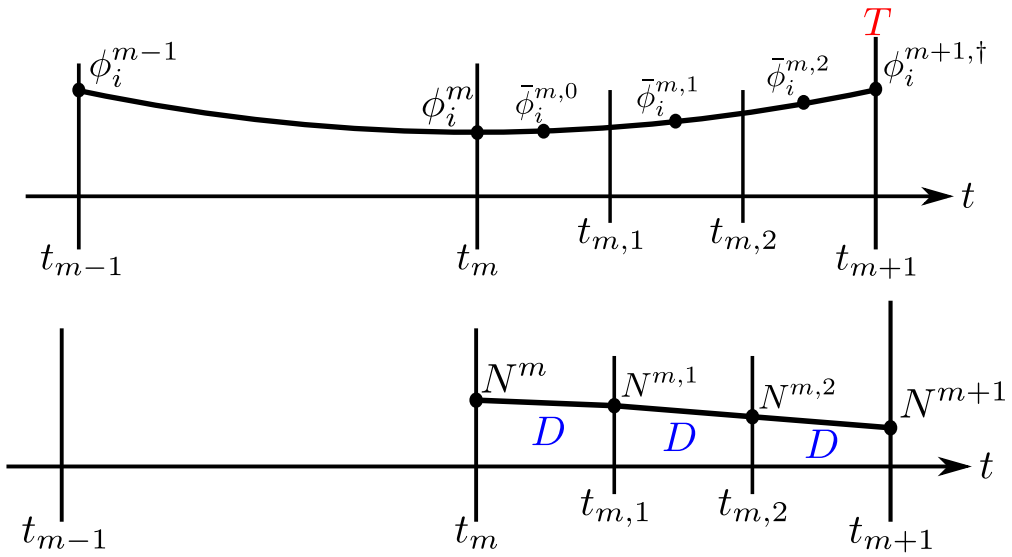

Figure 3. Illustration of the linear extrapolation/quadratic interpolation (LE/QI) algorithm for the corrector step, showing the transport component (*top*) and depletion component (*bottom*). Transport is performed at the end of the time step at time t_{m+1} to calculate end-of-step *predictor* fluxes and cross sections. These are then interpolated over the time step using values at times t_{m-1}, t_m and the predictor values at time t_{m+1}. Substep-averaged *corrector* fluxes $\bar{\phi}_i^{m,s}$ are calculated for each substep, as are substep-averaged *corrector* reaction cross sections $\bar{\sigma}_{r,i,j}^{m,s}$ (not shown). These are used to perform depletion over each substep to find number densities $N^{m,s}$ until the end of the time step is reached. T represents a transport solve, and D represents a depletion solve.

re-deplete from the beginning of the time step through to the end of the time step.

The second additional coupling method is featured in a variety of transport-depletion codes, including the Polaris [27] code, and is another type of predictor-corrector method. For the predictor step, a constant extrapolation is performed. For the corrector step, transport is solved at the end of the time step, and then the end-of-time-step fluxes and cross sections are used to re-deplete over the time step. The final nuclide number densities at the end of the time step are the averages of the number densities calculated during the predictor solve and the corrector solve.

5. Calculating Flux-Weighted Reaction Cross Sections

From the methods presented in Section 4, the necessary substep-averaged fluxes and reaction cross sections can be calculated given fluxes and cross sections evaluated at time step boundaries t_m. These substep-averaged fluxes and reaction cross sections are then used to calculate substep-averaged transition matrices using Eq. (9) to calculate depletion solutions using Eq. (14).

As mentioned at the end of Section 2, all of the reaction rates for all of the reactions in Table A.9 in all depletion regions of the problem can be directly tallied during the transport calculation, and then these reactions rates, together with the tallied scalar flux, can be used to calculate the reaction cross sections. However, for continuous-energy Monte Carlo simulations, there are reasons to avoid doing this. Solving the transport equation using Monte Carlo is by far the most expensive part of a coupled transport-depletion simulation, and the cost of the tallies is a significant portion of the cost of the transport solve. Since directly calculating all of the reaction rates requires a large number of tallies, this approach will substantially increase the cost of the transport solution.

5.1. Fine-Group Flux Approach

Fortunately, there is an alternative to directly tallying the reaction rates, but it requires an additional approximation. Instead of directly tallying the reaction rates, a fine-group region-averaged scalar flux can be tallied and used to approximate a flux-weighted energy-integrated reaction cross section. This is the same method used by VESTA [4] to calculate cross sections.

We define the scalar flux averaged over depletion region i and evaluated at time t_m as

$$\phi_i^m(E) = \frac{1}{V_i} \int_{\boldsymbol{x} \in R_i} \phi(\boldsymbol{x}, E, t_m) \, \mathrm{d}^3\boldsymbol{x} \,. \tag{22}$$

Substituting this into Eq. (3), we obtain the reaction cross section averaged over depletion region i and evaluated at time t_m as

$$\sigma_{r,i,j}^m = \frac{\int_0^\infty \sigma_{r,j}(E) \phi_i^m(E) \, \mathrm{d}E}{\int_0^\infty \phi_i^m(E) \, \mathrm{d}E} \,. \tag{23}$$

Now we introduce an energy discretization, in which the energy-axis is subdivided into many fine energy bins, as shown in Figure 4. Assuming the

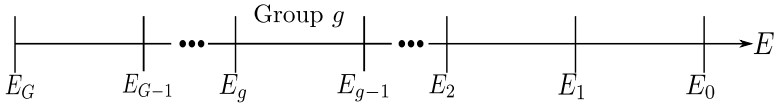

Figure 4. Energy grid showing multiple tally bins.

energies $E_G \leq E \leq E_0$ span all of the relevant physics, Eq. (23) can be rewritten without aproximation as

$$\sigma_{r,i,j}^m = \frac{\sum_{g=1}^G \int_{E_g}^{E_{g-1}} \sigma_{r,j}(E)\phi_i^m(E)\,\mathrm{d}E}{\sum_{g=1}^G \int_{E_g}^{E_{g-1}} \phi_i^m(E)\,\mathrm{d}E}\,. \tag{24}$$

To calculate $\sigma_{r,i,j}^m$, the following approximation is made:

$$\begin{aligned}\sigma_{r,i,j}^m &\approx \frac{\sum_{g=1}^G \left(\frac{\int_{E_g}^{E_{g-1}} \sigma_{r,j}(E)\,\mathrm{d}E \int_{E_g}^{E_{g-1}} \phi_i^m(E)\,\mathrm{d}E}{\int_{E_g}^{E_{g-1}} \mathrm{d}E} \right)}{\sum_{g=1}^G \int_{E_g}^{E_{g-1}} \phi_i^m(E)\,\mathrm{d}E} \\ &\approx \frac{\sum_{g=1}^G \left(\frac{\phi_{i,g}^m}{\Delta_g} \right) \sigma_{r,j,g}}{\sum_{g=1}^G \phi_{i,g}^m}\,, \end{aligned} \tag{25}$$

where $\Delta_g = E_{g-1} - E_g$ and

$$\sigma_{r,j,g} = \int_{E_g}^{E_{g-1}} \sigma_{r,j}(E)\,\mathrm{d}E\,, \tag{26a}$$

$$\phi_{i,g}^m = \int_{E_g}^{E_{g-1}} \phi_i^m(E)\,\mathrm{d}E\,. \tag{26b}$$

This approximation is accurate as long as the bin size is small enough such that the flux and/or cross section is nearly constant over the bin.

5.2. Calculating Cross Section Bin Integrals

We now discuss how to calculate Eq. (26a) in order to find $\sigma_{r,j,g}$. Consider the energy bin in Figure 5. Cross sections are defined as a series of energy points and values, and are assumed linear between points, as shown in the

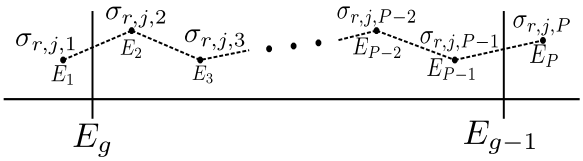

Figure 5. A single tally bin showing multiple energy points and associated cross section values. The cross section is assumed linear between points.

figure. Although this is an approximation, it is the same approximation that is imposed by the nuclear data evaluation. Therefore,

$$\begin{aligned}\sigma_{r,j,g} &= \sum_{p=1}^{P-1} \int_{E_{B_p}}^{E_{T_p}} \sigma_{r,j}(E)\,\mathrm{d}E \\ &= \sum_{p=1}^{P-1} \bar{\sigma}_{r,j,p}\,, \end{aligned} \tag{27}$$

where the integral bounds are defined as

$$E_{T_p} = \min(E_{p+1}, E_{g-1})\,, \tag{28a}$$

and

$$E_{B_p} = \max(E_p, E_g)\,, \tag{28b}$$

and $\bar{\sigma}_{r,j,p}$ is the integral of the cross section between E_{B_p} and E_{T_p}. E_{B_p} and E_{T_p} are defined to handle the case where the energy bin bound falls between two tabulated cross section data points.

Given the linearity of the cross sections between energy points, the cross section within $[E_p, E_{p+1}]$ is given by

$$\sigma_{r,j}(E) = \sigma_{r,j,p} + \left(\frac{E - E_p}{E_{p+1} - E_p}\right)\left(\sigma_{r,j,p+1} - \sigma_{r,j,p}\right)\,, \qquad E_p \le E \le E_{p+1}\,, \tag{29}$$

where $\sigma_{r,j,p} = \sigma_{r,j}(E_p)$. Therefore, integrating Eq. (29), we obtain

$$\begin{aligned}\bar{\sigma}_{r,j,p} = &\left(E_{T_p} - E_{B_p}\right)\sigma_{r,j,p} \\ &+ \left(\frac{\sigma_{r,j,p+1} - \sigma_{r,j,p}}{E_{p+1} - E_p}\right)\left[\frac{E_{T_p}^2 - E_{B_p}^2}{2} - E_p\left(E_{T_p} - E_{B_p}\right)\right]\,. \end{aligned} \tag{30}$$

Defining $\Delta_p = E_{T_p} - E_{B_p}$, this can be simplified to

$$\bar{\sigma}_{r,j,p} = \Delta_p \sigma_{r,j,p} + \left[\frac{\sigma_{r,j,p+1} - \sigma_{r,j,p}}{E_{p+1} - E_p}\right] \left[\frac{\Delta_p}{2}\left(E_{T_p} + E_{B_p} - 2E_p\right)\right] . \tag{31}$$

For the case in which the fine-bin energies E_p and E_{p+1} are known to be within the tally bin bounds E_g and E_{g-1}, so that $E_{B_p} = E_p$ and $E_{T_p} = E_{p+1}$, then $\Delta_p = E_{p+1} - E_p$ and Eq. (31) can be further simplified to

$$\bar{\sigma}_{r,j,p} = \frac{\Delta_p}{2}\left(\sigma_{r,j,p+1} + \sigma_{r,j,p}\right) . \tag{32}$$

5.3. Discussion of Cross Section Collapse

We conclude this section with a discussion of the use of Eqs. (27), (31), and (32) to calculate the one-group cross section from Eq. (25). By default, Shift uses the same tally bin structure as VESTA [4], which contains a total of 43 000 tally bins and has been tuned to produce accurate results in problems dominated by ^{238}U. Nuclear systems in which the primary actinide is something other than ^{238}U may not have accurate results from this bin structure. The tally bin structure used by default is given in Table 2. Users may substitute their own group structure if the default is not appropriate for their application.

This method assumes that the cross section is known exactly at the given evaluated energy points. However, in the unresolved resonance region the cross section is known only in a statistically averaged sense. This can produce errors that do not diminish with increasingly fine tally bins. There exists in the literature methods to compensate for this [28], but these methods have not yet been implemented into Shift.

Finally, the method described above is used by Shift to collapse the cross sections given by the ENDF/B-VII library (either VII.0 or VII.1). However, only a small fraction (approximately a quarter) of the depletion nuclides in ORIGEN have corresponding cross sections in this library. ORIGEN also uses multigroup cross section data from JEFF-3.1 to calculate certain reaction rates. Therefore, in addition to the one-group cross sections, Shift also passes a multigroup flux to ORIGEN, which is calculated by collapsing the fine-bin flux to the group structure of the JEFF-3.1 library.

Energy bounds [eV]	**Number of equilethargy bins**
$2.0 \times 10^{7} - 1.0 \times 10^{7}$	1000
$1.0 \times 10^{7} - 1.0 \times 10^{6}$	1000
$1.0 \times 10^{6} - 1.0 \times 10^{5}$	4000
$1.0 \times 10^{5} - 1.0 \times 10^{4}$	4000
$1.0 \times 10^{4} - 1.0 \times 10^{3}$	10 000
$1.0 \times 10^{3} - 1.0 \times 10^{2}$	10 000
$1.0 \times 10^{2} - 1.0 \times 10^{1}$	4000
$1.0 \times 10^{1} - 1.0 \times 10^{0}$	4000
$1.0 \times 10^{0} - 1.0 \times 10^{-1}$	1000
$1.0 \times 10^{-1} - 1.0 \times 10^{-2}$	1000
$1.0 \times 10^{-2} - 1.0 \times 10^{-3}$	1000
$1.0 \times 10^{-3} - 1.0 \times 10^{-4}$	1000
$1.0 \times 10^{-4} - 1.0 \times 10^{-5}$	1000

Table 2. Default tally bin structure used by Shift.

6. Power Normalization

The transport equation given by Eq. (1a) is an eigenvalue equation, so it specifies only the space-energy shape of the neutron flux, but does not specify its magnitude. The magnitude is determined by the user-specified power level in the reactor. The total power of the modeled system is given by

$$\begin{aligned} P(t) &= P_d(t) + P_r(t) \\ &= \sum_i \sum_j V_i N_{i,j}(t) \lambda_j \kappa_{d,j} + \sum_i \sum_j \sum_r V_i N_{i,j}(t) \kappa_{r,j} \sigma_{r,i,j}(t) \phi_i(t) \,, \end{aligned} \quad (33)$$

where $P_d(t)$ is the power due to nuclide decay, $P_r(t)$ is the power generated by neutron-induced reactions at time t, $\kappa_{d,j}$ is the average recoverable energy released by the decay of nuclide j, $\kappa_{r,j}$ is the average recoverable energy released by nuclide j undergoing reaction r, $\sigma_{r,i,j}(t)$ is the cross section for nuclide j undergoing reaction r in region i at time t, and $\phi_i(t)$ is the average neutron flux in region i at time t. In practice, the $\kappa_{d,j}$ are typically set to zero, and the $\kappa_{r,j}$ are adjusted to approximately include the energy released in decay reactions. Moreover, many codes, such as ORIGEN 6.2, only use $\kappa_{f,j}$ and $\kappa_{c,j}$, for fission and capture reactions, respectively. Therefore, the power is calculated as

$$P(t) = \sum_i \sum_j V_i N_{i,j}(t) \phi_i(t) \left[\kappa_{f,j} \sigma_{f,i,j}(t) + \kappa_{c,j} \sigma_{c,i,j}(t) \right] . \quad (34)$$

Some codes, such as Serpent, only use $\kappa_{f,j}$. By default, Exnihilo uses the $\kappa_{f,j}$ and $\kappa_{c,j}$ provided by ORIGEN 6.2. Exnihilo also allows for user-input $\kappa_{f,j}$ and $\kappa_{c,j}$ values, which are used for comparison to other codes. Currently, Exnihilo does not consider power produced by activation of materials outside of the specified depletion regions.

The flux shape is normalized by requiring that the evaluated power be equal to a user-specified input reaction power P_R^m over time step m, such that

$$\begin{aligned} P(t) &= \sum_i \sum_j V_i N_{i,j}(t) \left[\kappa_{f,j} \sigma_{f,i,j}(t) + \kappa_{c,j} \sigma_{c,i,j}(t) \right] \phi_i(t) \\ &= P_R^m \,, \qquad t_m \le t \le t_{m+1} \,. \end{aligned} \quad (35)$$

Setting

$$\phi_i(t) = c\phi_i^*(t)\,, \qquad t_m \leq t \leq t_{m+1}\,, \tag{36}$$

where $\phi_i^*(t)$ is the result of solving Eq. (1a) during time step m, and solving for c gives

$$\phi_i(t) = \left[\frac{P_R^m}{\sum_{i'}\sum_j V_{i'} N_{i',j}(t)\left[\kappa_{f,j}\sigma_{f,i',j}(t) + \kappa_{c,j}\sigma_{c,i',j}(t)\right]\phi_{i'}^*(t)}\right]\phi_i^*(t)\,,$$
$$t_m \leq t \leq t_{m+1}\,. \tag{37}$$

Because the nuclide number densities are changing continuously over the course of a fuel depletion calculation, the flux corresponding to a fixed power level is changing continuously as well. Many codes, such as Serpent 2.1.24, normalize the flux once at the beginning of each depletion step. This allows the flux to "drift" from the correct normalized flux as each substep is calculated during the depletion step; this "drift" can become significant for certain transport-depletion coupling methods (such as the fully explicit method) and long time steps. Ref. [17] describes several different substep flux normalization schemes implemented in Exnihilo, which are summarized here for completeness.

Exnihilo includes the options of normalizing only at the beginning of a depletion step (i.e., no substep normalization), at the middle of each substep, or to use an energy-integrated normalization scheme. Investigations using beginning-of-substep normalization were conducted and resulted in systematic overrepresentation of nuclide concentrations at early times [17].

Middle-of-substep flux normalization is accomplished by performing a depletion calculation to the middle of the substep. The resultant number densities are then used together with Eq. (37) to compute the appropriate flux normalization at the middle of the substep. Then the depletion is performed over the entire substep. This requires two depletion calculations per substep, but as shown in Ref. [17], this can improve accuracy and thereby enable longer time steps and thus fewer transport calculations.

Energy-based substep flux normalization is performed by augmenting the depletion system of equations to include calculations for the time-integrated total energy released during the substep [16]. This calculated total energy is compared to the total energy that should be released for the given power level to determine the normalization factor. Then, the flux is normalized by

this factor, and the substep is then re-depleted using the renormalized flux. The energy-based substep flux normalization also requires two depletion calculations per substep, and each depletion calculation is approximately 10% more expensive [16], due to the expense of solving the augmented system, for a total increase in the cost of the depletion of 220% compared to using no substep flux normalization, given a fixed number of substeps. However, as shown in Ref. [17], this produces more accurate depletion results than any of the other methods with fewer substeps, making the method an overall performance and accuracy improvement. Moreover, as will be shown in Section 7.4, the cost of the depletion is small relative to the cost of the Monte Carlo transport solve.

7. Results

This section presents results for the Exnihilo depletion capabilities discussed in the previous sections on three sets of demonstration problems. The first demonstration is a comparison between Shift and Serpent on an axial slice of a Westinghouse 17×17 pressurized water reactor (PWR) assembly. This is presented in Section 7.1. The second demonstration is on a series of Westinghouse AP1000® lattices, where we will show Shift results against both Serpent and MPACT [12]. This is presented in Section 7.2. Finally, Section 7.3 presents a demonstration of Shift depletion performance on several full-core simulations of the High Flux Isotope Reactor (HFIR) and compares the results against VESTA.

7.1. Westinghouse 17×17 Lattice

Our first test case is an axial slice of a representative Westinghouse 17×17 PWR assembly. These results were first presented in Ref. [18]. This test problem has been used elsewhere [24] as a depletion demonstration problem. This lattice features 248 regular fuel rods containing UO_2 with a ^{235}U enrichment of 4.2 at.%, and 16 gadolinium rods enriched to 2.7 at.%. The boundaries are all reflecting, including the top and bottom axial boundaries. In the regular fuel rods, the entire fuel pin is treated as a single depletion region, while in the gadolinium rods each fuel pin is divided into ten equal-volume concentric rings. These results were produced with ENDF/B-VII.0 cross sections.

The LE/QI method was employed for both Shift and Serpent 2.1.23, using the burnup lengths given in Table 3. To enable direct comparison between Shift and Serpent, no substep flux normalization was used (since this is not implemented in Serpent 2.1.23). To calculate the power at each depletion step the same $\kappa_{f,j}$ values used in Serpent were used in Shift, as shown in Appendix B. Both Shift and Serpent were run with 20 000 particles per cycle, 20 inactive cycles, and 250 active cycles.

Figure 6 shows the difference in k_{eff} between Shift and Serpent over the depletion steps. It can be seen that the difference in k_{eff} between Shift and Serpent is less than 100 pcm for all depletion steps. It should be noted that the error bars in Figure 6, which represent one standard deviation, only consider the variance in the transport solution; neither Shift nor Serpent propagate statistical uncertainties through the depletion calculation.

Figure 7 shows the relative percent differences in the lattice-averaged atom density for several selected actinides (*left*) and fission products (*right*).

Burnup length [MWd/MT]	Depletion steps	Length of each step [MWd/MT]
38.6	1	38.6
77.2	1	77.2
154.4	1	154.4
270.2	1	270.2
540.4	1	540.4
13 896.0	15	926.4
1524.7	1	1524.7
2084.4	1	2084.4
2875.7	1	2875.7
3242.4	1	3242.4
37 056.0	8	4632.0

Table 3. Burnup step details for the Westinghouse 17×17 problem.

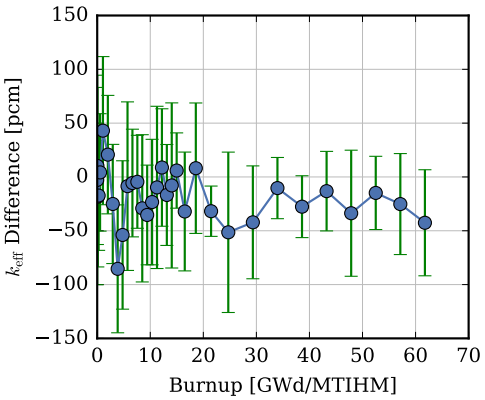

Figure 6. Difference in k_{eff} between Shift and Serpent for the Westinghouse 17×17 lattice.

All nuclides are well within 1.0%, except ^{235}U, which climbs to about 1.3%

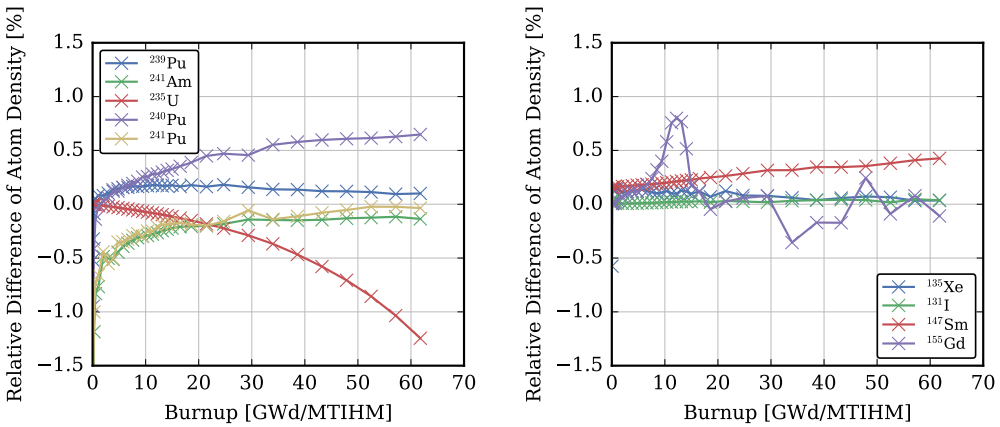

Figure 7. Relative differences between Shift and Serpent for select actinides (*left*) and fission products (*right*) for the Westinghouse 17×17 lattice.

by the end of the simulation. To understand why this occurs, the percentage of ^{235}U and ^{238}U consumed over the course of the simulation is shown in Figure 8 (*left*), and the percent difference in the uranium consumption as calculated by Shift and Serpent in Figure 8 (*right*). From the left image in Figure 8 it can be seen that over 90% of the ^{235}U is consumed over the simulation, and from the right image it can be seen that the difference in the consumption between Shift and Serpent peaks at about 0.13%. Thus, the relative difference between Shift and Serpent for ^{235}U grows to over 1% by the end of the depletion because there is a relatively small amount of ^{235}U remaining, but the absolute differences are very small.

7.2. AP1000® Lattices

The next series of test problems is for the Westinghouse AP1000® reactor. Similar results verifying MPACT vs. Shift and Serpent 2.1.24 were demonstrated in Ref. [19]. For these simulations all codes used the same $\kappa_{f,j}$ values as those used in Serpent, as shown in Appendix B, and use ENDF/B-VII.0 cross sections.

The AP1000® has an 18-month operating cycle and achieves very low neutron leakage using a highly heterogeneous fuel loading. Assembly enrich-

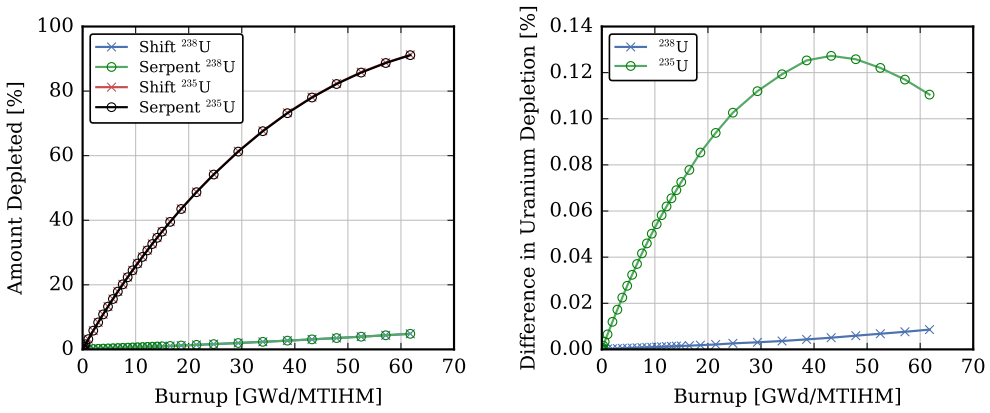

Figure 8. Fraction of ^{235}U and ^{238}U consumed during the simulation (*left*) and difference in uranium consumption between Shift and Serpent (*right*).

ments vary from nearly 5 wt.% ^{235}U to natural uranium. The core also features intra-assembly enrichment zoning, and contains integral fuel burnable absorbers (IFBAs) as well as short and long wet annular burnable absorbers (WABAs) to help support a radially flat reactivity.

Seven different axial slices of various assemblies were examined. Each lattice is 1/8 radially symmetric, and the layout of the pins within each lattice is illustrated in Figure 9. The pincells within these lattices are illustrated in Figure 10, with each material within the pin labeled. The composition and radii of the various materials labeled in Figure 10 are described in Table 4. Additional model details are given in Table 5, and a detailed description of the AP1000® core and simulation models can be found in Refs. [29, 30]. Each lattice was run for three initial depletion steps of 0.1, 0.4, and 0.5 GWd/MT, followed by a series of steps of approximately 1 GWd/MT. The total number of burn steps and the final burnup for each lattice are presented in Table 6.

Because these runs were used in Ref. [19] to validate MPACT, Shift and Serpent were run with high-fidelity, but not identical, settings. Serpent was run with a CE/LI predictor-corrector transport-depletion coupling method, with two substeps for the predictor and one for the corrector. All simulations except Region 3 and 4 were run with 3.5×10^6 particles per cycle, with 286 active cycles and 100 inactive cycles, while the Region 3 and 4 simulations

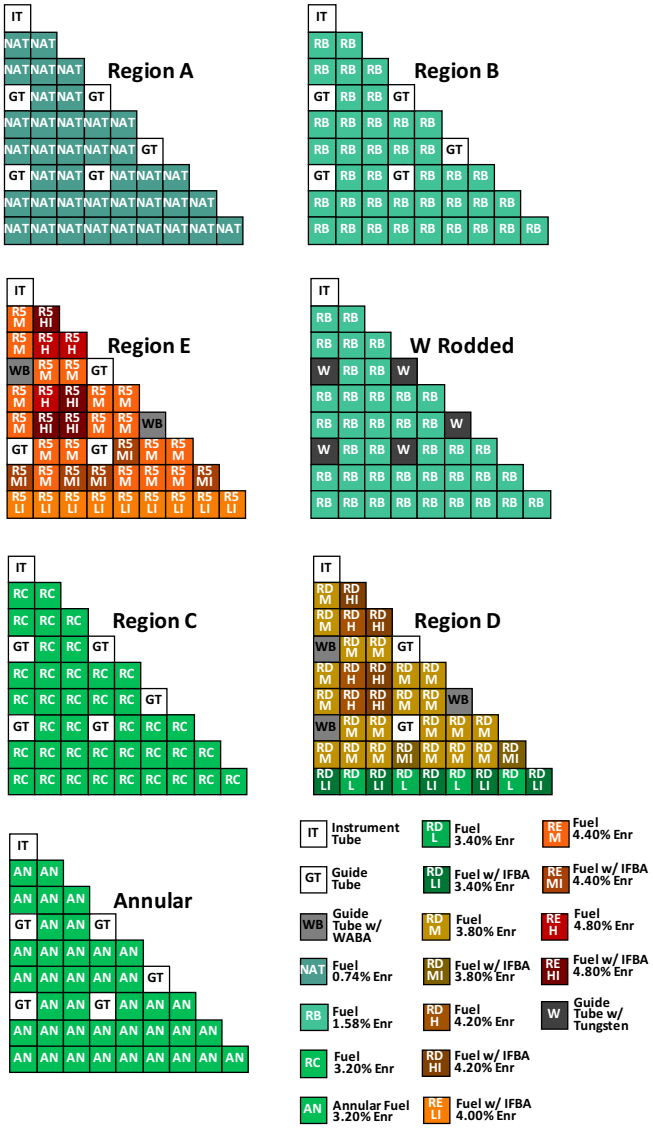

Figure 9. Layout of the seven different AP1000® assembly lattices under examination.

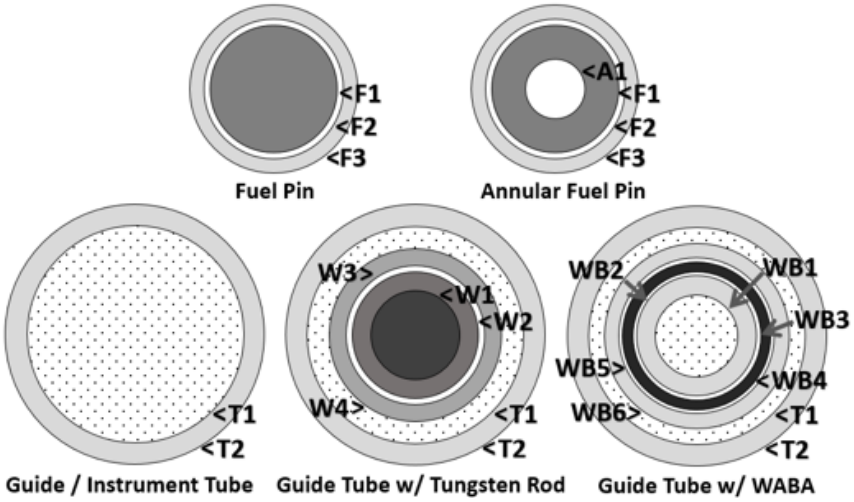

Figure 10. Illustration of the five different pins within the AP1000® assembly lattices.

Material Zone	Material	Radii [cm]
A1	Helium	0.19685
F1	Fuel	0.409575
F2	Helium	0.41783
F3	Zirlo	0.47498
W1	Tungsten	0.25019
W2	Inconel	0.3937
W3	Helium	0.4191
W4	Stainless Steel	0.48387
T1	Coolant	0.56134
T2	Zirlo	0.61214
WB1	Coolant	0.29
WB2	Zircaloy	0.34
WB3	Helium	0.35
WB4	Alumina Boron-Carbide	0.40386
WB5	Helium	0.41783
WB6	Zircaloy	0.48387

Table 4. AP1000® pin radii and material descriptions.

Parameter	Value
Fuel temperature	900 K
Non-fuel temperature	600 K
Coolant density	0.744 g/cc
Boron concentration	1321 ppm
IFBA layer thickness	0.000508 cm
Assembly pitch	21.5 cm
Pin pitch	1.26 cm
Lattice power	0.05075 MW
Lattice height	1 cm

Table 5. Additional model specifications for the AP1000® lattices.

Lattice	Depletion Steps	Final Burnup [GWd/MT]
Region A	64	62
Region B	72	70
Region C	72	70
Region D	71	69
Region E	71	69
W Rodded	56	54
Annular	73	70

Table 6. Number of depletion steps and total burnup for each AP1000® lattice.

were performed with 4×10^6 particles per cycle, with 250 active and 100 inactive cycles. The Shift simulations were performed with an LE/QI transport-depletion coupling method, with one predictor substep and four corrector substeps, and with energy-based substep flux normalization. Shift was also run with 4×10^6 particles per cycle, with 250 active and 100 inactive cycles for the first time step. Subsequent timesteps, when the fission source was presumed to be close to convergence, used only 10 inactive cycles. MPACT uses a predictor-corrector coupling method that is constant in both the predictor and the corrector. A method-of-characteristics ray spacing of 0.05 cm was used, as well as a 51-group multigroup library specifically designed for PWR simulations [31]. Finally, we note that both Serpent and Shift used one depletion region per pin, while MPACT used three equal-volume depletion regions per pin.

Figure 11 shows the difference in k_{eff} between Shift and Serpent (*top*) and between Shift and MPACT (*bottom*). Shift and Serpent match within 100 pcm at all depletion steps, except for two steps towards the end of the simulation of Region E, where the difference slightly exceeds 100 pcm. A slight tilt in the relative difference in k_{eff} between Shift and Serpent can be observed for all of the AP1000® lattice problems, with Serpent producing a small but consistently lower k_{eff} by end of cycle. The reasons for this are not clear. Possible explanations could include minor differences in the way energy-dependent fission yields are modeled by Serpent and Shift, differences in the ENDF/B-VII.0 decay data used in Serpent versus the ENDF/B-VII.1 decay data used by ORIGEN, slight differences due to the different transport-depletion coupling methods, or the use of substep flux normalization in Shift. Further investigation is necessary to definitively explain the difference. Nevertheless, we find good agreement in k_{eff} between Shift and Serpent for these lattice problems.

The differences are greater between Shift and MPACT, although they are still within 150 pcm. The greater differences are likely due to the fact that, as a deterministic code, MPACT applies a multigroup approximation to the cross sections.

Figure 12 shows the maximum percent pin power difference between Shift and Serpent (*top*) and between Shift and MPACT (*bottom*). The pin powers calculated by Shift differ from those calculated by Serpent by less than 0.21% for all depletion steps. The differences are greater when comparing to pin powers from MPACT, but are still well within 0.5%.

Figure 13 shows the root mean square (RMS) percent pin power difference

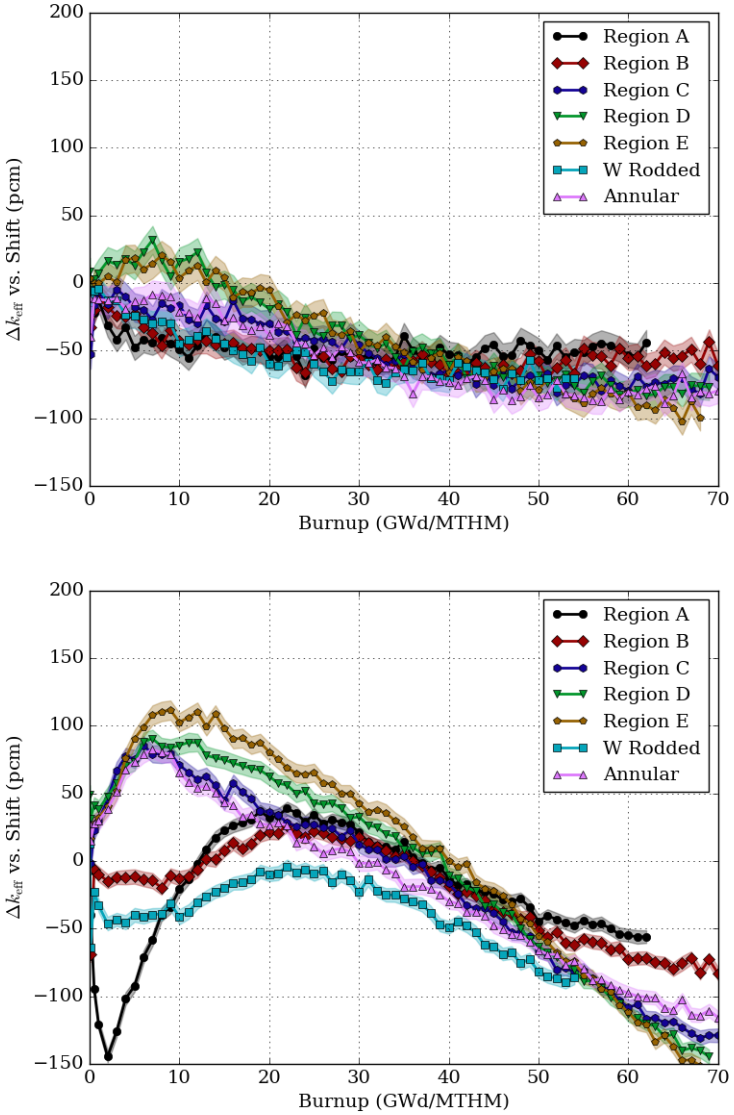

Figure 11. Difference in k_{eff} between Shift and Serpent (*top*) and Shift and MPACT (*bottom*). Shaded regions indicate 2σ uncertainty in the transport solution. Uncertainties are not propagated through the solution of the depletion equations.

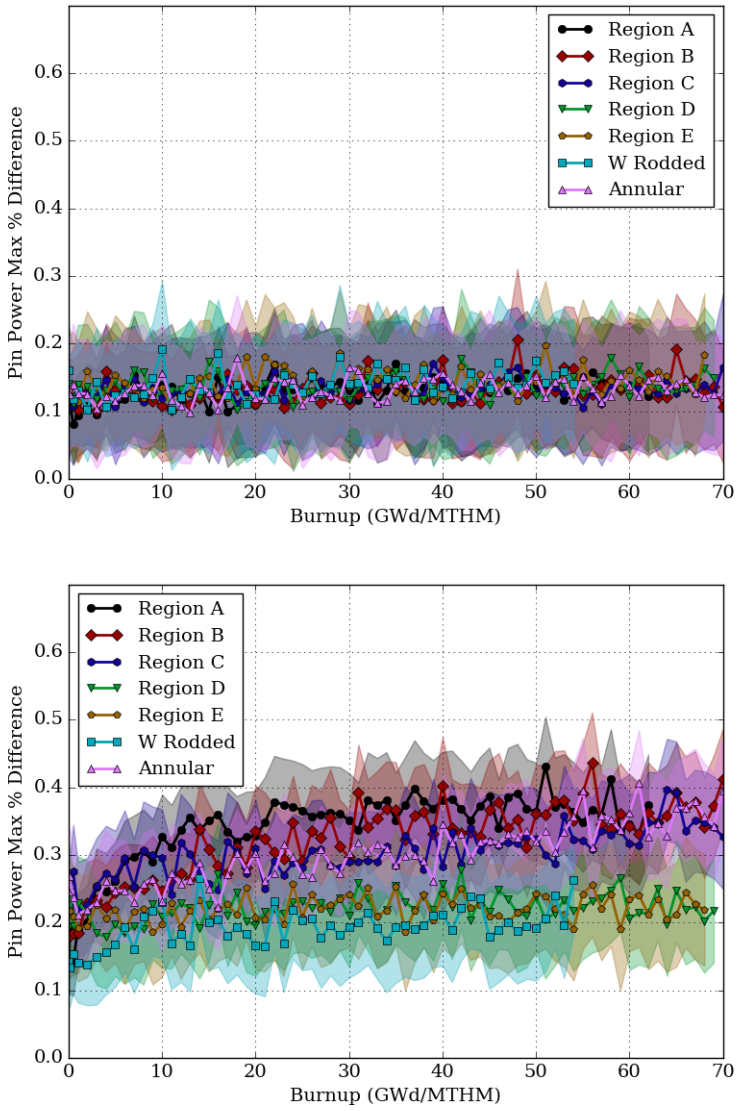

Figure 12. Maximum pin power difference between Shift and Serpent (*top*) and between Shift and MPACT (*bottom*). Shaded regions indicate 2σ uncertainty in the transport solution. Uncertainties are not propagated through the solution of the depletion equations.

between Shift and Serpent (*top*) and between Shift and MPACT (*bottom*). The difference in the RMS pin powers between Shift and Serpent are less than 0.05% for all lattices for almost all depletion steps. Again, the errors are somewhat greater when comparing to MPACT, likely due to the truncation errors introducted by MPACT's deterministic methods. Nevertheless, the RMS pin power difference remains under 0.15%, which is very good agreement.

In all, we consider the solutions generated by the transport-depletion algorithms in Shift, Serpent, and MPACT to be in good agreement for all of the AP1000® lattices under consideration here.

7.3. High Flux Isotope Reactor

The final series of test cases are models of the High Flux Isotope Reactor (HFIR) at Oak Ridge National Laboratory. Currently, HFIR is fueled with highly enriched 93 wt.% UO_2 fuel arranged in two cylindrical fuel elements, an inner and an outer fuel element. These fuel elements are composed of 171 and 369 involute fuel plates, respectively. At beginning-of-cycle, a HFIR core contains 10.1 kg of uranium (9.4 kg ^{235}U). HFIR currently operates with a fuel cycle length of 24 to 26 days at a power of 85 MW. The LEU conversion program is working towards converting HFIR to use low enriched uranium.

Both Exnihilo and VESTA used the ORIGEN 2.2 $\kappa_{f,j}$ values for these HFIR simulations. Actinides with atomic number $Z \geq 80$ are assigned a value using [32]

$$\kappa_{f,j} = 1.29927 \times 10^{-3} Z^2 \sqrt{A} + 33.12\,, \tag{38}$$

where Z is the atomic number and A is the atomic mass of nuclide j. ENDF/B-VII.0 cross sections and the fully-explicit coupling method were used for all simulations.

7.3.1. Simplified HEU Cycle 400 Model

We will begin by demonstrating Shift on the simplified HFIR cycle 400 model. The model used for this demonstration is HFIR cycle 400 with a simplified fuel representation. The fuel, clad, and filler within a fuel involute, along with the interchannel coolant, are all homogenized into axially stacked concentric rings, as shown in Figure 14. We refer the reader to Ref. [13] for details about the model. This model has been shown [13] to provide accurate results for the HEU-fueled HFIR core while reducing the number of depletion regions, and thus the computational expense.

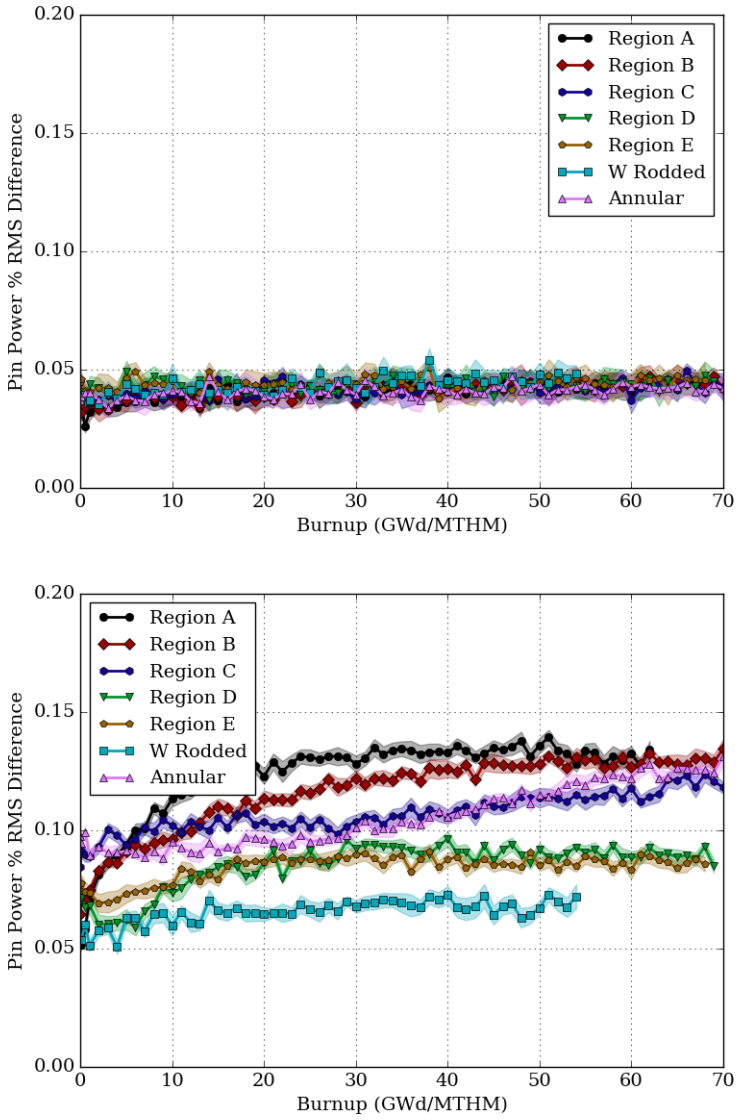

Figure 13. Root mean squared pin power difference between Shift and Serpent (*top*) and between Shift and MPACT(*bottom*). Shaded regions indicate 2σ uncertainty in the transport solution. Uncertainties are not propagated through the solution of the depletion equations.

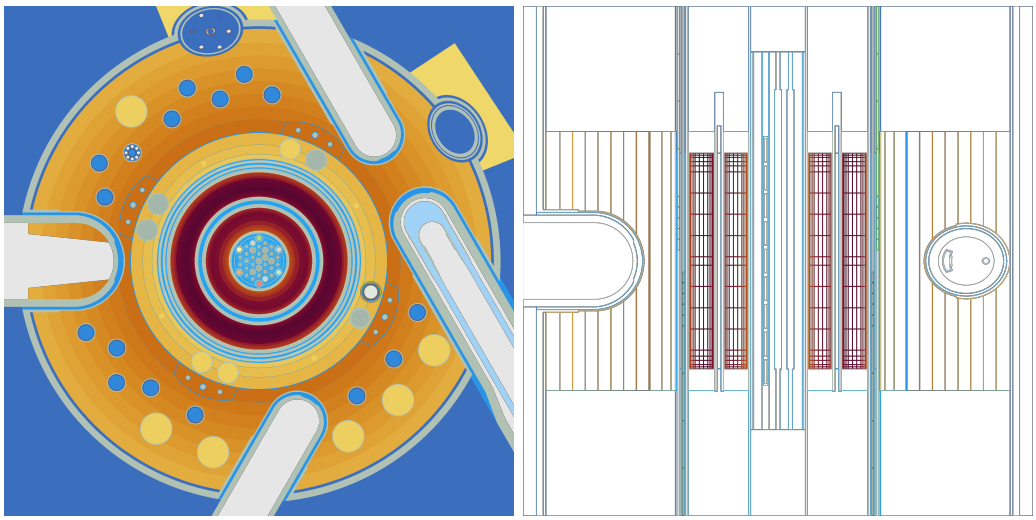

Figure 14. Radial (*left*) and axial (*right*) view of the simplified HFIR model. The fuel, filler, clad, and surrounding moderator (shown in red on the *left*) are homogenized into concentric cylinders axially stacked.

The simulations were run with 1×10^5 particles per cycle, with 50 inactive cycles and 300 active cycles for both Shift and VESTA 2.0.3. Figure 15 shows the calculated k_{eff} for both Shift and VESTA throughout the fuel cycle (*left*) and the difference between the calculate k_{eff} values between Shift and VESTA (*right*). It can be seen that the difference between Shift and VESTA peaks at about 100 pcm near the end of the fuel cycle.

Figure 16 shows the relative difference between Shift and VESTA for select actinides. Near the beginning of the fuel cycle, the amounts of transuranic actinides present in the model are very small, and thus the relative differences are large. As the cycle proceeds and transuranic elements are produced, the relative error becomes small. By the end of the cycle, all of the selected transuranics are within 1%.

Figure 17 shows the relative difference between Shift and VESTA for select fission products. The difference between Shift and VESTA for most fission products are within 1%, with a few exceptions. There is a persistent difference between Shift and VESTA for ^{135}Xe and ^{149}Sm. This is likely due to the fact that the modern version of ORIGEN to which Shift is cou-

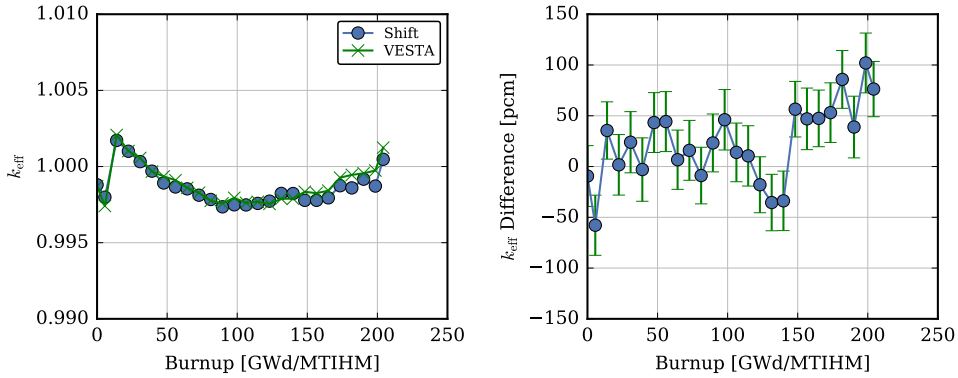

Figure 15. k_{eff} **values (*left*) and** k_{eff} **differences (*right*) calculated by Shift and VESTA over a fuel cycle using the simplified HFIR cycle 400 model.**

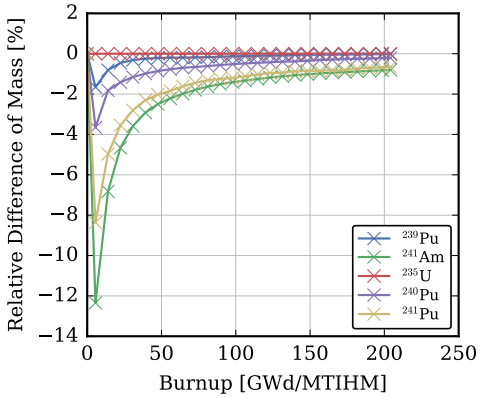

Figure 16. Relative difference in total mass of select actinides calculated by Shift and VESTA for the simplified HFIR cycle 400 model.

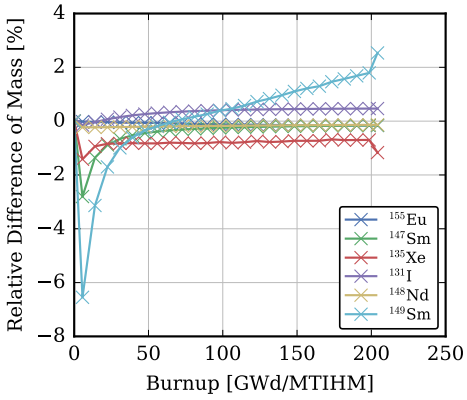

Figure 17. Relative difference in total mass of select fission products calculated by Shift and VESTA for the simplified HFIR cycle 400 model.

pled uses energy-dependent fission yields, while VESTA-2.0.3 is coupled to ORIGEN-2.2 and uses energy-independent fission yields. Overall, the absolute differences between the ^{135}Xe and ^{149}Sm are small, as shown in Figure 18.

7.3.2. Detailed HEU Cycle 400 Model

We now examine Shift results with a detailed version of the HEU-fueled HFIR cycle 400 model. This features explicitly modeled fuel plates, as shown in Figure 19. For full details about this model please refer to Ref. [13].

These simulations were run with 1×10^5 particles per cycle, with 300 active cycles, and 50 inactive cycles for both Shift and VESTA 2.0.3. Figure 20 shows the k_{eff} calculated by Shift and VESTA (*left*) and the difference between Shift and VESTA (*right*). There is a greater divergence between Shift and VESTA for this model, peaking at just under 150 pcm, although we still consider these differences moderate.

Finally, Figure 21 shows the percent differences between Shift and VESTA for select actinides (*left*) and fission products (*right*). These results are similar to those found for the simplified model.

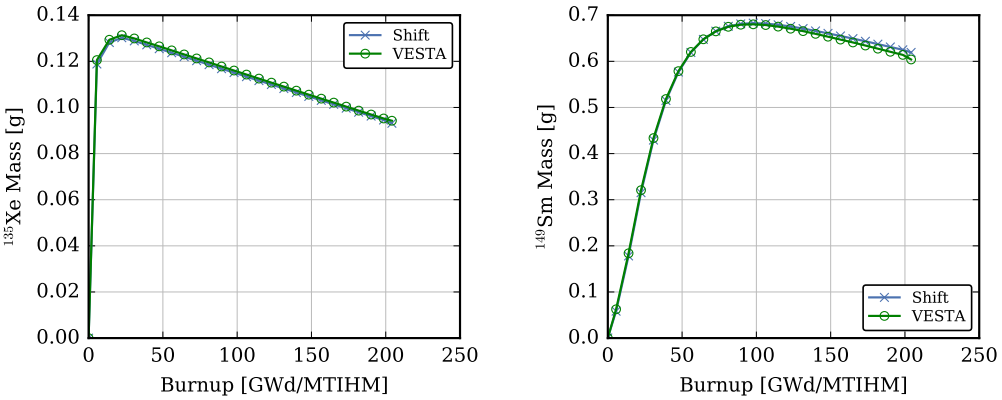

Figure 18. Total mass of ^{135}Xe (*left*) and ^{149}Sm (*right*) in the HFIR core as calculated by Shift and VESTA for the simplified cycle 400 model.

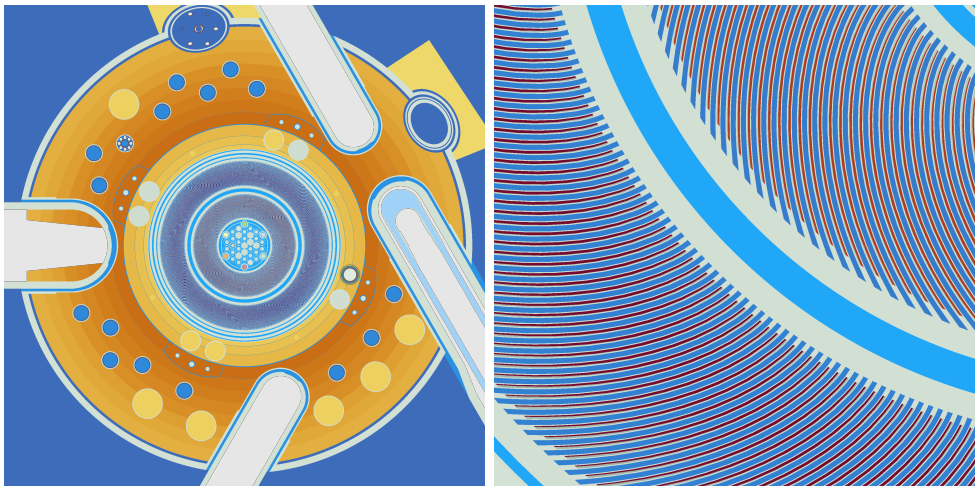

Figure 19. HFIR core model with explicit axial fuel plates in both the outer and inner fuel elements (*left*) and a close-up view of the fuel involutes (*right*).

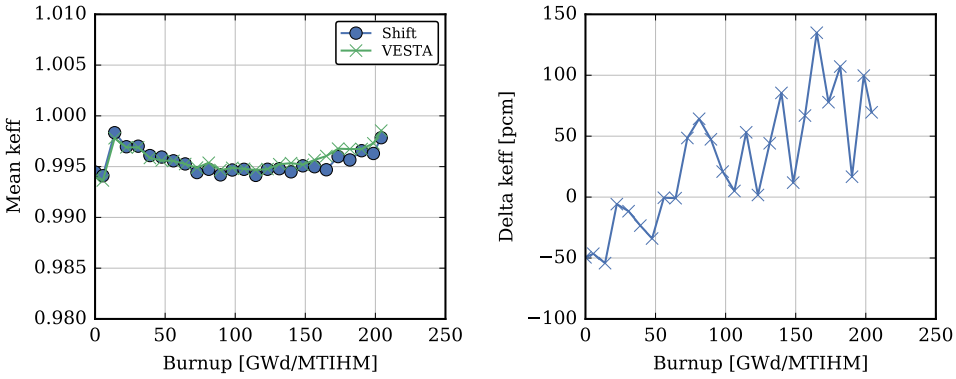

Figure 20. k_{eff} **values (*left*) and** k_{eff} **differences (*right*) calculated by Shift and VESTA over a fuel cycle for the detailed HFIR cycle 400 model.**

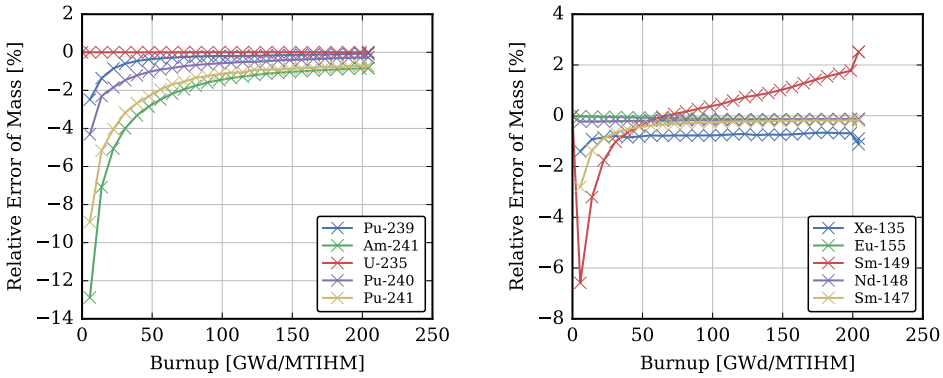

Figure 21. Relative percent differences in total mass between Shift and VESTA for several select actinides (*left*) and fission products (*right*) for the detailed HFIR cycle 400 model.

7.3.3. Detailed LEU Cycle 400 Model

We now examine a representative LEU-fueled HFIR model similar to the HEU-fueled model. The details about the LEU fuel and core are given in Ref. [33]. The particular model simulated also included the experimental target loading. These details are given in Ref. [34].

We ran both the Shift and VESTA 2.0.3 simulations with 1×10^5 particles per cycle, with 300 active and 50 inactive cycles. Figure 22 shows the calculated k_{eff} for Shift and VESTA (*left*) and the difference between Shift and VESTA (*right*). There is very good agreement, with the difference in k_{eff}

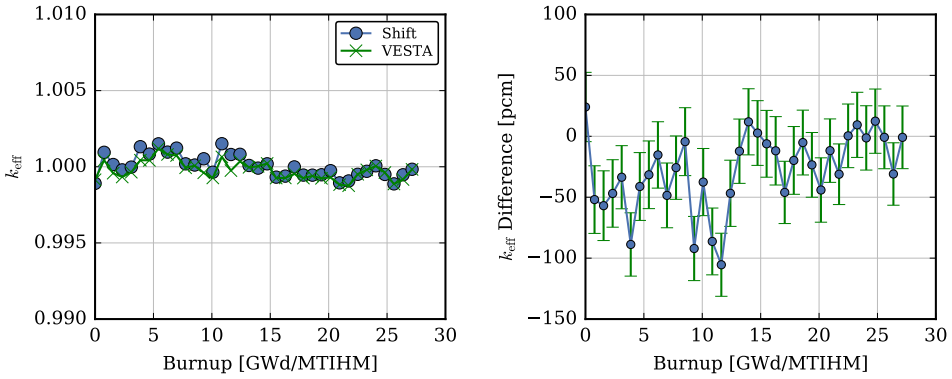

Figure 22. Calculated k_{eff} values for Shift and VESTA (*left*), and the difference between the calculated k_{eff} values (*right*) for the LEU HFIR cycle 400 model.

between Shift and VESTA peaking fairly early in the fuel cycle near 100 pcm and then settling close to zero towards the end of the fuel cycle.

Figure 23 shows the difference between Shift and VESTA for select actinides (*left*) and fission products (*right*). For the actinides, some modest differences between Shift and VESTA can be seen for ^{241}Pu, ^{240}Pu, and ^{241}Am. These differences are largely due to differences in small numbers. Figure 24 shows the mass of ^{241}Pu (*left*) and ^{241}Am (*right*) generated during the fuel cycle as calculated by Shift and VESTA, and it can be observed that the differences are almost indistinguishable. For the fission products, as before, there are differences between Shift and VESTA for ^{135}Xe and ^{149}Sm. Figure 25 shows that the absolute differences between Shift and VESTA for these nuclides are small.

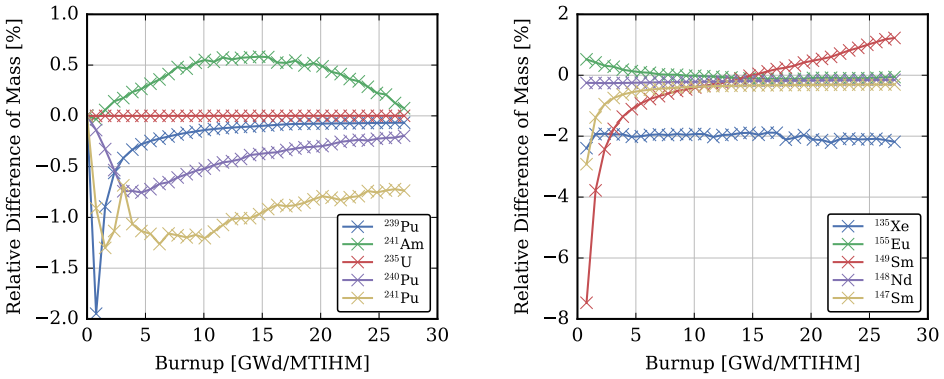

Figure 23. Relative percent difference between Shift and VESTA for several select actinides (*left*) and fission products (*right*) for the LEU HFIR cycle 400 model.

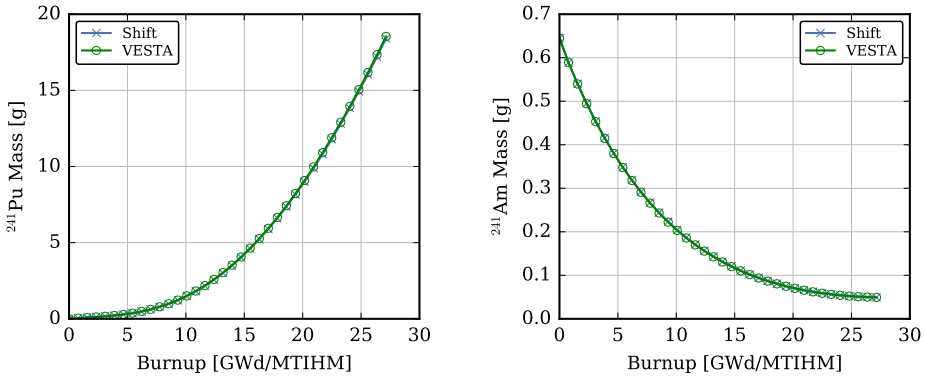

Figure 24. Total mass of ^{241}Pu (*left*) and ^{241}Am (*right*) generated over the fuel cycle as calculated by Shift and VESTA for the LEU HFIR cycle 400 model.

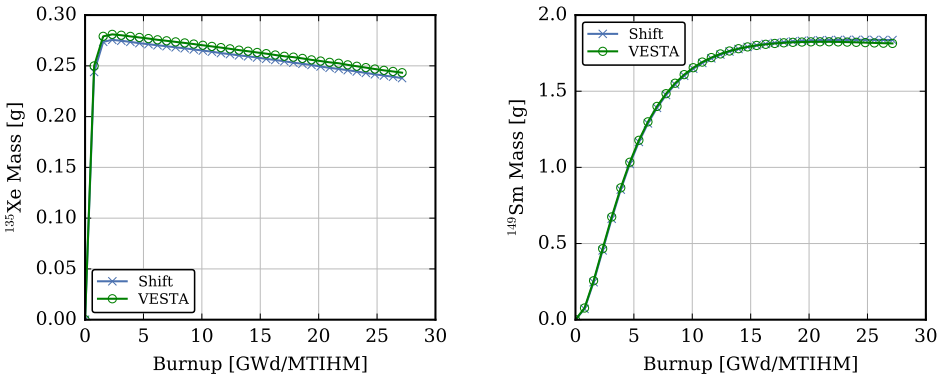

Figure 25. Total mass of ^{135}Xe (*left*) and ^{149}Sm (*right*) generated over the fuel cycle as calculated by Shift and VESTA for the LEU HFIR cycle 400 model.

7.4. Performance of the Shift Depletion Capability

We will now examine the computational performance of this new Shift depletion capability. First, the performance of Shift on the various HFIR models is compared with the traditional HFIR analysis tool, VESTA. Performance numbers comparing Shift and VESTA were first presented in Ref. [18]. However, since that time, improved cross section caching has been implemented in Shift, improving the performance of the Shift Monte Carlo transport solver. This timing study was performed on the Oak Ridge Institutional Cluster (OIC), which features 2.40 GHz Intel Xeon processors. Updated comparisons of Shift performance versus VESTA are shown in Table 7.

Several observations can be made about these timings. First, the reason that only 24 CPUs were used in the VESTA runs is because the limit of scalability available with MCNP-5.1, the transport solver used by VESTA, is two nodes. The OIC has 12 cores per node. Using more than 24 cores causes a reduction in performance. Second, for problems with relatively low numbers of depletion regions, VESTA uses fewer CPU-hours than Shift. This is because the underlying transport solver (MCNP-5.1) is more efficient for small problems. However, because of VESTA's limited scalability, the wallclock time for all Shift runs is much less than for VESTA. The LEU HFIR runtimes are longer for both Shift and VESTA, likely because the presence of low-enriched fuel causes additional scattering and thus longer

HFIR with Simplified Fuel Representation
(498 Depletion Regions)

Code	Wallclock Time [h]	CPUs	CPU hours
Shift	6.8	384	2592.8
VESTA	100.1	24	2402.2

HEU HFIR with Explicit Fuel Plates
(707 Depletion Regions)

Code	Wallclock Time [h]	CPUs	CPU hours
Shift	9.4	384	3712.0
VESTA	205.3	24	4926.2

LEU HFIR with Explicit Fuel Plates
(595 Depletion Regions)

Code	Wallclock Time [h]	CPUs	CPU hours
Shift	10.4	384	4108.8
VESTA	325.0	24	7800.0

Table 7. Performance comparison between Shift and VESTA for various HFIR models.

particle histories. Altogether, these results indicate that as the transport problem becomes more expensive, either because of more depletion regions to tally or because of the underlying physics, Shift is more efficient, both in wallclock time and in CPU hours.

A scaling study was performed on the detailed LEU-fueled HFIR core on the CADES institutional cluster at Oak Ridge National Laboratory. Figure 26 shows the solve time versus the number of cores for the transport component (*left*) and the depletion component (*right*). Both components scale roughly linearly for low numbers of cores. As expected, as the number of cores exceeds the number of depletion regions, the depletion solve time remains flat. Table 8 shows the fraction of time spent in various ma-

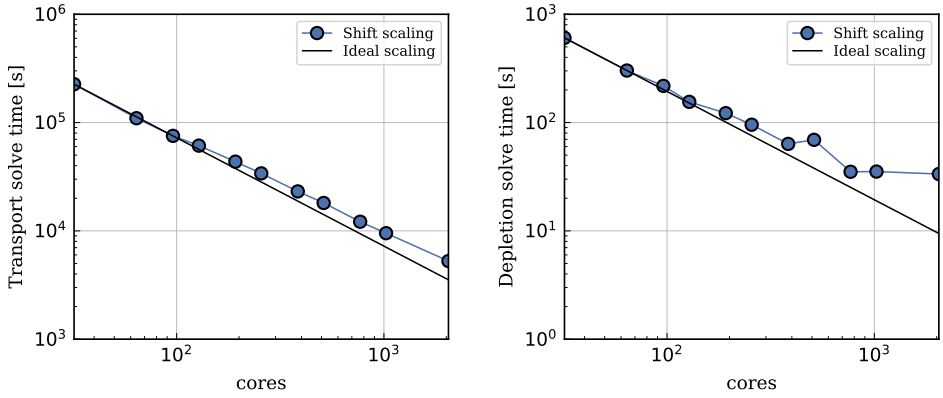

Figure 26. Scaling of the transport component (*left*) and depletion component (*right*) for the HFIR LEU-fueled HFIR simulation.

jor components during the solution of the HFIR LEU fuel cycle. Numbers do not sum to unity because there are other components, such as setup, input/output, etc., that are not shown. As these timings illustrate, the cost of the depletion is small relative to the cost of the transport. Therefore, future work on Shift optimizations will likely focus on further optimizations to the transport solver. As larger problems such as full-core PWR depletion are addressed, the cost of the depletion may become significant.

Cores	XS Collapse (%)	Depletion Solve (%)	Power Norm (%)	MC Solve (%)
32	0.27	0.27	0.16	99.30
64	0.27	0.27	0.15	95.76
96	0.29	0.29	0.17	99.19
128	0.27	0.25	0.17	98.42
192	0.28	0.28	0.18	99.01
256	0.29	0.28	0.22	98.84
384	0.26	0.24	0.21	85.29
512	0.43	0.38	0.36	96.00
768	0.32	0.28	0.33	96.34
1024	0.38	0.36	0.41	94.29
2048	0.59	0.58	0.61	84.72

Table 8. Percent time spent in various solve components, including cross section collapse, the depletion solve, power normalization, and the Monte Carlo solve.

8. Conclusions and Future Work

A new depletion capability has been implemented in the Exnihilo radiation transport suite and coupled to the continuous-energy transport capability in the Shift Monte Carlo code. This new depletion package exploits the multilevel parallelism present in the Shift Monte Carlo code to maximize parallel scalability while minimizing communication. This paper demonstrates this new capability on several lattice problems, as well as full-core simulations of the High Flux Isotope Reactor, and compares results against other depletion codes, including VESTA, Serpent, and MPACT. We have demonstrated that the depletion package has near-linear parallel scaling.

Future work includes coupling the depletion package to the deterministic capabilities in Exnihilo, including the multigroup Monte Carlo solver in Shift, as well as the multigroup S_N, SP_N, and MOC solvers in Denovo. Additional capabilities are planned for development, such as directly tallying reaction rates, as well as optimizations such as selectively omitting unimportant nuclides. Ultimately, the depletion package will be scaled up to $O(100k)$ cores to simulate sub-pin resolved depletion calculations of full-core pressurized water reactors.

Appendix A. Reactions Modeled by the ORIGEN 6.2 Depletion Solver

Table A.9 lists all of the nuclide reactions modeled by the ORIGEN 6.2 depletion solver. The MT number refers to the reaction as specified in the evaluated nuclear data files.

MT Number	Reaction
16	(n, 2n)
17	(n, 3n)
18	(n, fission)
22	(n, n-α)
23	(n, n-3α)
24	(n, 2n-3α)
25	(n, 3n-α)
28	(n, n-p)
29	(n, n-2α)
32	(n, n-d)
33	(n, n-t)
34	(n, n-^{3}He)
37	(n, 4n)
51	(n, nx1)
102	(n, γ)
103	(n, p)
104	(n, d)
105	(n, t)
106	(n, ^{3}He)
107	(n, α)
108	(n, 2α)
111	(n, 2p)
112	(n, p-α)

Table A.9. Reactions modeled by ORIGEN 6.2.

Appendix B. Fission Heating Values used by Serpent

Serpent 2.1.23 and Serpent 2.1.24 use the following fission heating values ($\kappa_{f,j}$). Shift simulations comparing results to Serpent also use these values.

Nuclide	$\kappa_{f,j}$ [**MeV**]	**Nuclide**	$\kappa_{f,j}$ [**Mev**]	**Nuclide**	$\kappa_{f,j}$ [**MeV**]
^{227}Th	208.8261291	^{237}U	187.94445592	^{240}Am	214.92294284
^{228}Th	193.16409111	^{238}U	206.77231386	^{241}Am	210.87378395
^{229}Th	208.8261291	^{239}U	187.94445592	^{242}Am	208.82717324
^{230}Th	199.00864162	^{240}U	206.73890151	^{242m}Am	208.82717324
^{231}Th	186.61495772	^{241}U	187.94445592	^{243}Am	212.60694508
^{232}Th	196.7882867	^{234}Np	199.85084161	^{244}Am	208.8261291
^{233}Th	193.16409111	^{235}Np	208.82717324	^{244m}Am	208.8261291
^{234}Th	193.16409111	^{236}Np	208.8261291	^{240}Cm	219.22676647
^{229}Pa	194.63903744	^{237}Np	205.03696005	^{241}Cm	211.28089253
^{230}Pa	194.42843523	^{238}Np	208.8261291	^{242}Cm	211.4897197
^{231}Pa	208.82717324	^{239}Np	198.38581458	^{243}Cm	208.82717324
^{232}Pa	208.8261291	^{236}Pu	203.60649391	^{244}Cm	208.82717324
^{233}Pa	208.82717324	^{237}Pu	204.54621619	^{245}Cm	208.82717324
^{230}U	198.51821101	^{238}Pu	206.09153727	^{246}Cm	208.82717324
^{231}U	197.32246661	^{239}Pu	207.61994335	^{247}Cm	208.82717324
^{232}U	197.34063458	^{240}Pu	208.27378123	^{248}Cm	217.95292071
^{233}U	199.47171588	^{241}Pu	210.89466667	^{249}Cm	208.82717324
^{234}U	200.30702457	^{242}Pu	210.47690791	^{250}Cm	208.82717324
^{235}U	202.02295745	^{243}Pu	208.82717324		
^{236}U	203.07398462	^{244}Pu	211.80296046		

Table B.10. Values of $\kappa_{f,j}$ used by Serpent.

Acknowledgments

We thank Steven Hamilton for his assistance with the solution of matrix exponential differential equations and the Magnus series.

Research sponsored by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the U.S. Department of Energy and by the Consortium for Advanced Simulation of Light Water Reactors (www.casl.gov), an Energy Innovation Hub (http://www.energy.gov/hubs) for Modeling and Simulation of Nuclear Reactors under U.S. Department of Energy Contract No. DE-AC05-00OR22725. This research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725.

References

[1] X-5 Monte Carlo Team, MCNP - a general Monte Carlo n-particle transport code, version 5, Tech. Rep. LA-UR-03-1987, Los Alamos National Laboratory (2008).

[2] I. C. Gauld, G. Radulescu, G. Ilas, B. D. Murphy, M. L. Williams, D. Wiarda, Isotopic depletion and decay methods and analysis capabilities in SCALE, Nuclear Technology 174 (2) (2011) 169–195.

[3] W. L. Wilson, T. R. England, K. A. V. Riper, Status of CINDER'90 codes and data, in: Proceedings of the Fourth Workshop on Simulating Accelerator Radiatoin Environments (SARE4), Knoxville, TN, 1998.

[4] VESTA User's Manual - Version 2.1.0, Tech. Rep. DSU/SEC/T/2008-331, IRSN, BP 17, 92262 Fontenay-aux-roses, Cedex, France (2008).

[5] H. R. Trellue, D. I. Poston, User's Manual, Version 2.0 for Monteburns, Version 1.0, Tech. Rep. LA-UR-99-4999, LANL (September 1999).

[6] D. P. Griesheimer, D. Gill, B. Nease, T. Sutton, M. Stedry, P. Dobreff, D. Carpenter, T. Trumbell, E. Caro, H. Joo, D. Millman, MC21 v.6.0 - A Continuous-Energy Monte Carlo Particle Transport Code with Integrated Reactor Feedback Capabilities, in: Joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo 2013 (SNA + MC2013), American Nuclear Society, Paris, France, 2013.

[7] J. T. Goorley, M. R. James, T. E. Booth, F. B. Brown, J. S. Bull, Initial mcnp6 release overview - mcnp6 version 1.0, Tech. Rep. LA-UR-13-22934, Los Alamos National Laboratory (2013).

[8] J. Leppanen, Serpent–A Continuous-energy Monte Carlo Reactor Physics Burnup Calculation Code, Tech. rep., VTT Technical Research Centre of Finland (June 2015).

[9] T. M. Pandya, S. R. Johnson, T. M. Evans, G. G. Davidson, S. P. Hamilton, A. T. Godfrey, Implementation, capabilities, and benchmarking of Shift, a massively parallel Monte Carlo radiation transport code, Journal of Computational Physics 308 (2016) 239–272.

[10] T. Evans, A. Stafford, R. Slaybaugh, K. Clarno, DENOVO: A new three-dimensional parallel discrete ordinates code in SCALE, Nuclear Technology 171 (2010) 171–200.

[11] P. J. Turinsky, D. B. Kothe, Modeling and simulation challenges pursued by the consortium for advanced simulation of light water reactors (CASL), Journal of Computational Physics 313 (2016) 367–376.

[12] A. Zhu, B. Collins, B. Kochunas, T. Downar, Assessment of the depletion capability in MPACT, in: PHYSOR 2014, Kyoto, Japan, 2014.

[13] G. Ilas, D. Chandler, B. Ade, E. Sunny, B. Betzler, D. Pinkston, Modeling and Simulations for the High Flux Isotope Reactor Cycle 400, Tech. Rep. ORNL/TM-2015/36, Oak Ridge National Laboratory (March 2015).

[14] A. Isotalo, P. Aarnio, Higher order methods for burnup calculations with bateman solutions, Annals of Nuclear Energy 38 (2011) 1987–1995.

[15] A. E. Isotalo, P. Aarnio, Substep methods for burnup calculations with bateman solutions, Annals of Nuclear Energy 38 (2011) 2509–2514.

[16] A. Isotalo, Calculating time-integral quantities in depletion calculations, Nuclear Science and Engineering 183 (2016) 421–429.

[17] A. Isotalo, G. Davidson, T. Pandya, W. Wieselquist, S. Johnson, Flux renormalization in constant power burnup calculations, Annals of Nuclear Energy 96 (2016) 148–157.

[18] G. G. Davidson, T. M. Pandya, A. E. Isotalo, S. R. Johnson, T. M. Evans, W. A. Wieselquist, Nuclide depletion capabilities in the Shift Monte Carlo code, in: PHYSOR 2016, American Nuclear Society, Sun Valley, Idaho, 2016.

[19] C. Gentry, A. Godfrey, T. Pandya, G. Davidson, F. Franceschini, AP1000 benchmarking of VERA neutronics toolset, in: International Conference on Mathematics & Computational Methods Applied to Nuclear Science and Engineering, American Nuclear Society, Jeju, Korea, 2017.

[20] W. Magnus, On the exponential solution of differential equations for a linear operator, Communications on Pure and Applied Mathematics VII (1954) 649–673.

[21] S. Blanes, F. Casas, J. Oteo, J. Ros, The magnus expansion and some of its applications, Physics Reports 470 (2009) 151–238.

[22] A. E. Isotalo, W. A. Wieselquist, A method for including external feed in depletion calculations with CRAM and implementation into ORIGEN, Annals of Nuclear Energy 85 (2015) 68–77.

[23] J. C. Wagner, S. W. Mosher, T. M. Evans, D. E. Peplow, J. A. Turner, Hybrid and parallel domain-decomposition methods development to enable Monte Carlo for reactor analyses, Progress in Nuclear Science Technology 2 (2011) 815–820.

[24] A. E. Isotalo, V. Sahlberg, Comparison of neutronics-depletion schemes for burnup calculations, Nuclear Science and Engineering 179 (4) (2015) 434–459.

[25] A. E. Isotalo, Comparison of neutronics-depletion coupling schemes for burnup calculations–continued study, Nuclear Science and Engineering 180 (3) (2015) 286–300.

[26] M. A. Jessee, M. D. DeHart, TRITON: A multipurpose transport, depletoin and sensitivity and uncertainty analysis module, Tech. Rep. ORNL/TM-2005/39, Version 6.1, Available from the Radiation Safety Information Computational Center at Oak Ridge National Laboratory as CCC-785 (2011).

[27] M. A. Jessee, et al., POLARIS: A New Two-Dimensional Lattice Physics Analysis Capability for the SCALE Code System, in: PHYSOR 2014 International Conference, American Nuclear Society, Kyoto, Japan, 2014.

[28] D. Kotlyar, E. Fridman, E. Shwageraus, One-group cross-section generation for monte carlo burnup codes: Multigroup method extension and validation, Nuclear Science and Engineering 179 (3) (2015) 274–284.

[29] M. Hone, et al., AP1000 core reference report: Revision 1, Tech. Rep. WCAP-17524-NP-A, Westinghouse Electric Company, LLC (2015).

[30] F. Franceschini, et al., Westinghouse VERA test stand zero power physics test simulations for the AP1000 PWR, Tech. Rep. CASL-U-2014-0012-000, Westinghouse Electric Company, LLC (2014).

[31] K. S. Kim, K. Clarno, C. Gentry, A. Godfrey, M. L. Williams, D. Wiarda, B. Kochunas, Y. Liu, Development of the V4.2m5 and V5.0m0 multigroup cross section libraries for MPACT for PWR and BWR, Tech. Rep. CASL-U-2017-1280-000, ORNL/TM-2017/95 (2017).

[32] A. G. Croff, ORIGEN2: A versatile computer code for calculating the nuclide compositions and characteristics of nuclear materials, Nuclear Technology 62 (1983) 335–352.

[33] B. R. Betzler, D. Chandler, E. E. Davidson (née Sunny), G. Ilas, High Fidelity Modeling and Simulation for a High Flux Isotope Reactor Low-Enriched Uranium Core Design, Nuclear Science and Engineering 187 (1) (2017) 81–99.

[34] D. Chandler, B. R. Betzler, G. Hertz, G. Ilas, E. E. Sunny, Modeling and Depletion Simulations for a High Flux Isotope Reactor Cycle with a Representative Experiment Loading, Tech. Rep. ORNL/TM-2016/23, Oak Ridge National Laboratory (2016).