NE255 Project - Final Report

Vergari, Lorenzo

December 12, 2019

1 Introduction

In recent years, an increasing number of innovative nuclear reactor concepts have been proposed by universities and industries, inducing an increasingly larger effort in devising and performing simulations that can reliably forecast reactor physics, safety and operations. Simulations focused on studying the properties of the fuel and its behavior after irradiation are among those studies, and are particularly concerned with evaluating features such as reactivity, decay heat and radiation dose. Since these features will depend on the composition of fuel at reactor discharge and on the geometry of the systems where newly-spent fuel is stored, it is useful to develop a computational sequence capable of estimating fuel composition at reactor discharge, studying neutron transport in the spent fuel and evaluating the evolution of decay heat and dose with time. One of the aims of this NE255 final project is to tackle part of this task, by building a transport-depletion sequence on the platform SCALE [1] for the PB-FHR fuel [2] , quantifying its sub-criticality margin when extracted from reactor and assessing decay heat evolution.

In order to perform such analyses, several steps and multiple modules of the platform SCALE will be required. Provided an initial equilibrium composition as a known input, a transport calculation will be required to compute the multi-group cross sections and and to collapse them into a one-group structure. Such one-group cross sections will be used as parameters in a depletion calculation to evaluate the fuel composition change due to irradiation. Finally, having assuming the geometry and materials of spent fuel storage systems, the sub-criticality of the configuration will be verified through an additional transport calculation, and the decay heat evolution with time will be monitored through another depletion sequence.

This report also aims at providing a description of the mathematical formulation of each phase and of how the phases are interconnected. With this regards, a particular focus will be dedicated to the depletion sequence. Indeed, in addition to introducing and defining the constitutive equations, as done for the other phases, the report will provide details on the methods and algorithms used for its resolution.

This document is organized as follows. Section 2 will introduce the mathematical modelling of each phase of the problem. Section 3 will deep dive the depletion sequence and focus on its solution algorithms. Section 4 will describe SCALE, the computational suite used for the analysis, and its depletion module ORIGEN [1]. Section 5 will present the results of the depletion sequence to a case study of interest and, finally, Section 6 will include the conclusions to the report.

2 Mathematics

The compositions of spent nuclear fuel and of fuel under irradiation in a reactor evolve as a result of nuclear reactions and radioactive decays. A model used to describe this evolution as a function of time was formulated by Rutherford in 1905 [3] and solved by Harry Bateman in 1908 [4]. In this model, the compositions N_i of each isotope are collected in the vector $\boldsymbol{N}$(t) and are evolved according to Equation 1.

$$\frac{dN_i}{dt} = \sum_{j \neq i}^{N_{isotopes}} (I_{ij}\lambda_j + f_{ij}\sigma_j\Phi)N_i(t) - (\lambda_i + \sigma_i\Phi)N_i(t) + S_i(t) \tag{1}$$

Here, λ_i is the decay constant of nuclide i, I_{ij} is the fractional yield of nuclide i from decay of nuclide j, σ_i is the spectrum-averaged removal cross section for nuclide i, f_{ij} is the fractional yield of nuclide i from neutron-induced removal of nuclide j, Φ is the one speed scalar flux and S_i is a time-dependent source.

The Bateman equation, can be visualized as a balance to the concentration of the nuclides. Source terms are represented by the decay and reactions of other nuclides having the production of nuclide j as one of their branches. On the other side, decay and induced reactions on nuclide j act as loss terms in the balance.

Selected parameters of the Bateman equation, such as decay constants and decay branching ratios, can be retrieved from nuclear libraries. However, other parameters, such as removal cross sections and reaction branching ratios, depend on the neutron spectrum of the system under study.

Indeed, as the spectrum is influenced by core composition, the cross sections and reaction fractional yields in the Bateman equation would not be constant parameters, but would be depending on the flux, which in turn depends on the concentrations N_i, making the equation non-linear.

One strategy to preserve the linearity of the equation is to adopt for the cross sections equilibrium values, i.e. values that are representative for all the irradiation history, with a space- and time-

averaged neutron spectrum. With this scope in mind, let's consider the energy dependent transport equation.

$$\begin{aligned}&\frac{1}{v}\frac{\partial\psi}{\partial t}(\vec{r},E,\hat{\Omega},t)+\hat{\Omega}\cdot\nabla\psi(\vec{r},E,\hat{\Omega},t)+\Sigma_t(\vec{r},E,N(t))\psi(\vec{r},E,\hat{\Omega},t)=\\&\int_0^\infty\int_{4\pi}\Sigma_s(\vec{r},E'\to E,\hat{\Omega}'\to\hat{\Omega},N(t))\psi(\vec{r},E',\hat{\Omega}',t)d\hat{\Omega}'dE'\\&+\frac{\chi_p(E)}{4\pi}\int_0^\infty\int_{4\pi}\nu(E')\Sigma_f(\vec{r},N(t),E')\psi(\vec{r},E',\hat{\Omega}',t)d\hat{\Omega}'dE'\\&+S(\vec{r},E,\hat{\Omega},t).\end{aligned}\tag{2}$$

Here, in addition to the cross sections Σ, ψ is the angular flux, $\hat{\Omega}$ is the unit direction vector, v and E are the neutron velocity and energy, χ_p and ν are the fission energy spectrum and yield and S is an external source term. In the equation, the cross sections Σ can be decomposed as $\Sigma = N(t,\vec{r})\sigma(E)$, where σ is the microscopic cross sections and N is the density.

As the microscopic cross section is a function of the neutron energy, one can write:

$$\sigma=\frac{\int_0^T\int_V\int_E\sigma(E)\Phi(\vec{r},E,t)}{\int_0^T\int_V\int_E\Phi(\vec{r},E,t)}\tag{3}$$

where *[0,T]* is the irradiation time and V the system volume.

Thus, in order to compute the average microscopic cross section to be used in the Bateman equation, one should perform a time-dependent transport simulation covering the whole irradiation time. Alternatively, one can resort to the integral mean value theorem, and express this denominator as the space- , time- and energy- averaged scalar flux Φ_{STE} multiplied by the phase space volume VTE. Therefore, if the time averaged composition is known, one could consider a time-independent transport equation using such time averaged compositions, as those would lead to a time averaged neutron flux. As a result, a depletion problem can be modelled as a two step sequence:

1. Time-independent transport equation with equilibrium compositions to define the energy-averaged cross sections;

2. Time evolution of the composition vector with the Bateman equation.

As the main focus of this report is the study of the depletion sequence, a discussion of the discretization strategy and of the solving algorithms for step 1 is left to other references. Interested readers might consult [5] and [6]. For our purpose, it is sufficient to introduce the one-group cross sections to be used in the Bateman equation for each nuclide as [7]:

$$\sigma_i=\frac{\sum_g\sigma_i^g\Phi_{eq}^g}{\sum_g\Phi_{eq}^g}\tag{4}$$

Where the superscript g addresses the energy group and Φ_{eq}^g is the scalar flux of the g-th energy group considering an equilibrium core composition. An analogus expression defines the one-group fractional yields f_{ij} .

Having defined the energy and time averaged parameters that will appear in the depletion sequence, let us consider the second step of the sequence, described by $N_{isotopes}$ Eq. 1 (one for each nuclide). These equations constitute a system of linear ODEs in time which can be written as:

$$\frac{d\vec{N}}{dt}=\mathbf{A}\vec{N}(t)+\vec{S}(t)\tag{5}$$

Here, $\mathbf{A}$ represents the transition matrix, whose matrix element A_{ij} defined as in Eq. 6 and $\vec{S}$ is a source vector.

$$A_{ij}\doteq\begin{cases}l_{ij}\lambda_i+f_{ij}\sigma_i\Phi_{eq}, & \text{if } i\neq j\\-\lambda_i-\sigma_i\Phi_{eq}, & \text{otherwise}\end{cases}\tag{6}$$

The source term is non-zero in case of an external feed, like in a molten salt reactor. For all other cases, including the FHR test case that will be described in this report, the term is null, and will therefore be neglected in the following sections.

3 Methods and Algorithms

The ODE system of Eq. 5 in absence of a source term, has an analytical solution in the form of a matrix exponential:

$$\vec{N}(t) = e^{\mathbf{A}t}\vec{N}(0) \tag{7}$$

$$e^{\mathbf{A}t} \doteq \mathbf{I} + \mathbf{A}t + \frac{(\mathbf{A}t)^2}{2} + ... = \sum_{k=0}^{\infty} \frac{(\mathbf{A}t)^k}{k!} \tag{8}$$

The algorithm that leads to the solution of the Bateman equation using the exponential matrix of Eq. 8 is named MATREX. An alternative method to MATREX is CRAM (Chebyshev Rational Approximation Method), an algorithm based on LU decomposition [8][9][10].

3.1 MATREX

Solving the Bateman equation by simply computing the exponential matrix as in Eq. 8 would require storage of the entire transition matrix $\mathbf{A}$. However, for an accurate analysis, many thousands of nuclides needs to be considered, with significant demands for memory storage. A recursive relation has therefore been defined to provide an alternative of lower computational cost. To describe this alternative, let's replace Eq. 8 in Eq. 7 and write the scalar equation for the i-th nuclide, expanding the inner products and the powers of $\mathbf{A}$:

$$\begin{aligned} N_i(t) =& N_i(0) + t\sum_j Aij N_j(0) + \frac{t}{2}\sum_k \left[A_{ik} t \sum_j A_{kj} N_j(0) \right] \\ &+ \frac{t}{3}\sum_m \left\{ A_{im} \frac{t}{2} \sum_k \left[A_{mk} t \sum_j A_{kj} N_j(0) \right] \right\} + ... \end{aligned} \tag{9}$$

Here, N_i is computed as a summation of a series of terms that arise from the multiplication of the transition matrix with the vector of concentration increments produced from the previous terms. In other words, by defining the *step concentrations*:

$$C_i^0 = N_i(0), \quad C^1 = t\sum_j A_{ij} C_j^0, \quad C^{n+1} = \frac{t}{n+1}\sum_j A_{ij} C_j^n, \tag{10}$$

the concentration of nuclide i can be written as

$$N_i(t) = \sum_{n=0}^{n_{term}} C_i^n + \epsilon_{trunc} \tag{11}$$

where n_{term} is the number of terms kept in the summation and ϵ_{trunc} is the series truncation error. Therefore, this strategy requires storage of only two vectors, $\vec{C}^n$ and $\vec{C}^{n+1}$, in addition to the current value of the solution. A simplified representation of the MATREX algorithm is provided in Appendix A.

Key to obtain an accurate solution with MATREX is to split the nuclides in two groups; those who are short-lived, for which this methodology is followed, and those who are short-lived, for which a more accurate solution is given in an alternate fashion [1]. A description of such methodology is in Appendix B.

3.2 CRAM

The Chebyshev Rational Approximation Method approximates the generic exponential function e^x by means of the rational function $\hat{r}_{k,l} = \frac{\hat{p}_k}{\hat{q}_k}$ that minimizes the largest absolute error (i.e. the infinity norm) on the negative real axis.

$$\epsilon_{k,k} \doteq sup_{x\in\mathbb{R}^-} |\hat{r}_{k,k}(x) - e^x| \tag{12}$$

In the specific case of the depletion equation, it has been shown that the exponential matrix can be defined as a contour integral in the complex plane $\mathbb{C}$ and that the eigenvalues of the matrix are confined to a region near the negative real axis [11] . The calculation of the exponential matrix can therefore be replaced by a calculation of a contour integral, which can be approximated with

quadrature formulas [12]. Such quadrature formulas can, in turn, be associated with rational functions, whose poles and residues correspond to the nodes and weights of the quadrature [13]. As a result, our exponential matrix can be approximated with a rational fraction $\hat{r}_{k,k}$ that can be selected according to the criterion in Eq. 12. Moreover, the rational function can be expressed in terms of its limit at infinity α_0, its poles θ_i (which come in conjugate pairs) and their corresponding residues α_i [9]:

$$\hat{r}_{k,k}(x) = \alpha_0 + Re\left(\sum_{i=1}^{k/2} \frac{\alpha_i}{(x-\theta_i)}\right) \tag{13}$$

Thanks to the Carathèodory-Fejer theorem, poles and residues θ_i and α_i can be computed from the coefficients of the Chebyshev rational function, which have been reported in literature [14], as performed in [13]. From these, the concentration vector can be approximated as follows:

$$\mathbf{N}(t) = e^{\mathbf{A}(t)}\mathbf{N}(0) \approx \hat{r}_{k,k}(-\mathbf{A}t)\mathbf{N}(0) = \alpha_0\mathbf{N}(0) - Re\left(\sum_{i=1}^{k/2}(\theta_i\mathbf{I} + \mathbf{A}t)^{-1}\alpha_i\mathbf{N}(0)\right) \tag{14}$$

As a result, the concentration vector can be calculated by solving k/2 linear systems. An efficient strategy for solving these systems consists in forming the depletion matrix $\mathbf{A}$ by indexing the nuclides in ascending order with respect to their mass number (so that non zero elements are concentrated around the diagonal), calculating the LU factorization of $\mathbf{A}$ [15] and performing a Gaussian factorization [16]. A simplified representation of the algorithm is provided in Appendix C.

4 Code Use

4.1 SCALE suite

The depletion sequence described in the previous sections has been performed using several multiple of SCALE [1], a modeling and simulation suite for nuclear applications developed at Oak Ridge National Laboratories (ORNL). SCALE includes several computational modules, including nuclear libraries, three Monte Carlo solvers and a depletion solver. For the scope of this project, three different modules of SCALE have been used:

- KENO Monte Carlo solver is used to perform a transport simulation on an equilibrium FHR cell and retrieve the one-group cross sections and fractional yields.
- ORIGEN depletion module is used to simulate irradiation of fuel in the reactor and decay of spent fuel.
- CSAS-6 sequence (based on KENO) is used to perform criticality analysis of the spent fuel.

In what follows, a fairly detailed description of ORIGEN input and output will be provided. Readers interested in other modules shall consult SCALE code system manual [1].

4.2 ORIGEN input

The depletion calculation described in the previous sections has been performed using SCALE depletion module ORIGEN [1]. ORIGEN uses the Scale Object Notation (SON) language for its input. ORIGEN input decks are divided in blocks that can be filled with the required information. The basic structure of an input deck is shown in Fig. and described as follows.

- **Bounds.** The "bounds" block defines the energy groups to be used for emission spectra and source calculation.
- **Solver.** The "solver" block controls the choice of the solver, either MATREX or CRAM, and their parameters, such as the number of expansion terms in MATREX and the order of the method in CRAM.
- **Options.** The "options" block allow to specify multiple options regarding the routine (such as whether to use a fixed fission energy instead of the nuclide specific values) or the output to be displayed.

```
'SCALE comment
=origen

     % ORIGEN comment %

bounds{ … }
solver{ … }
options{ … }

case(A){
      time=[31 365] % days
       …
}

case(B){
      …
}

% more cases?

end
```

Figure 1: ORIGEN basic input deck structure

- **Case.** The "case" blocks are the core of the ORIGEN input decks. A case represents a depletion calculation and is described by a set of parameters ("cards") that need to be specified within the block:
  - *Lib.* The "lib" card points to ORIGEN the transition matrix to be used during calculation.
  - *Mat.* The "mat" card includes the list of material and their compositions to be depleted.
  - *Time.* Such card defines the length of each of the discrete time instants to simulate.
  - *Flux/ Power.* This card defines the power at which the materials are irradiated during in each time instant.
  - *Print.* The "print" card specifies which case quantities to be plotted in the output.

The physical quantities that shall be provided as inputs to ORIGEN are the initial composition of each component N_i , the cross-sections, branching and transition rates composing the matrix $\mathbf{A}$, the power P_{eq} or flux level Φ_{eq} that will be used for the depletion (SCALE is capable of automatically converting the two quantities) and the length of irradiation/ decay period. In addition to such physical quantities, the user shall provide parameters on how to perform the simulation (including which solution method to adopt) and options on how to visualize results.

In some cases, the user does not have to manually insert one of the required inputs, but it is enough to address another file that contains such information, if available. For instance, the parameters of the transition matrix $\boldsymbol{A}$, which have been calculated before with a steady state transport simulation, are stored in an existing SCALE file (*.f33* extension) that can be read by the routine [1].

4.3 ORIGEN output

ORIGEN output is collected in a binary file (*.f71* extension). Possible results to be included in this output are compositions of depleted material at different time instants and radiation spectra for different radiation types. Such outcomes can be printed in human readable formats (*.txt* files) or can be visualized graphically through OPUS, SCALE module for data plotting and visualization [1].

5 Test Problem and Results

5.1 Case Study

One of the objectives of the project is to apply the mathematical background described in the previous sections to a practical case study based on the FHR report [2]. SCALE is generally

capable of automatically coupling the transport and depletion calculations in a sequence: it would compute the group constants in a first step, build the transition matrix and use them to deplete fuel in a second one. However, the fuel used for such depletion would be the same fuel used for building the group constants. On the other hand, for this project, the composition used for the generation of group constants should be the equilibrium composition, whereas the depletion step would regard the fresh fuel composition.

One way to do so is to set-up a depletion calculation for the equilibrium composition and to retrieve its group constants at time 0, disregarding and/or not performing the following step. At this point, a second depletion calculation (this time on the fresh fuel composition) can be performed using such constants.

Therefore, as a first step, a coupled KENO-ORIGEN input was written down for an equilibrium FHR cell. Instead of performing a reactor wide simulation and averaging, a fictitious face-centered cubic fuel-coolant cell representative of the different burn-up states of the fuel pebbles was built and periodical boundary conditions were set [2]. The *.f33* file produced by this simulation was then used as a *lib* input in a ORIGEN depletion calculation of a fresh pebble.

The composition at reactor discharge predicted by ORIGEN was then used within a CSAS-6 input deck to simulate how many depleted pebbles would need to be stacked (in a cubic basket submerged in FLiBe) to reach a critical mass. Finally, the decay of a pebble was simulated for 20 years (again with ORIGEN) in order to assess the evolution of the decay heat with time. All input decks are provided in a supplementary folder alongside this report.

5.2 Results

5.2.1 Reactor Irradiation

In MK1-PB-FHR, each pebble contains 1.5 grams of 19.9 percent enriched Uranium, 0.05 grams of Oxygen and about 23 grams of Carbon. Pebbles are designed to circulate in the core and to receive irradiation for 1.52 years, prior to discharge [2]. With a total FHR power of 236 MW_{th}, produced by 470,000 fuel pebbles, the average power density is 22.7 W_{th}/cm^3. The ORIGEN simulation was performed using these parameters, and the evolution of the concentrations (computed using MATREX algorithm with the default number of terms in the series expansions) is shown in Figure 2.

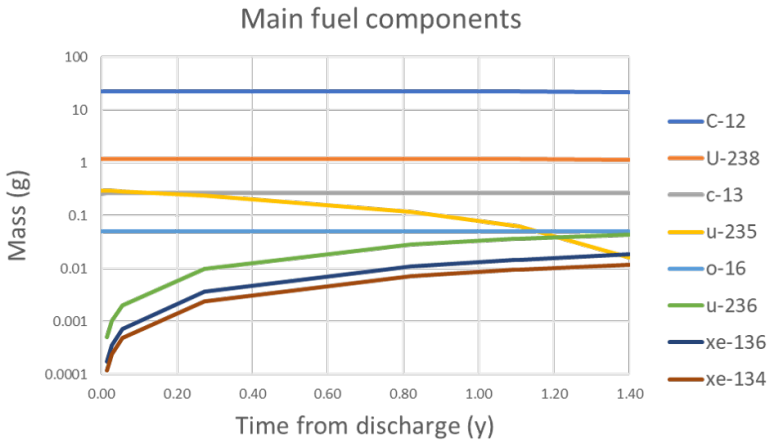

Figure 2: Mass of pebble components (g) vs. time (years)

5.2.2 Criticality Safety

The minimum number of spent fuel pebbles required to achieve a critical mass was evaluated by assembling a variable size cube containing an FCC lattice of spent pebbles and submerging it in FLiBe. For any given cube size, a steady state CSAS-6 transport simulation was performed and the multiplication factor k_{eff} was computed. The result of this parametric analysis is that it is impractical to amass enough fuel pebbles for a unitary k_{eff}. For instance, even considering a total number of 147 millions of pebbles (about 314 times larger than the number of pebbles contained in a FHR unit), the resulting k_{eff} is only 0.50186. Hence, it is reasonable to conclude that wet storage

will not pose the threat of spent fuel criticality, therefore, wet storage design features and limits will depend only on thermal-hydraulic and radiation protection considerations.

5.2.3 Spent Fuel Decay

ORIGEN was used to simulate the evolution of fuel composition due to radioactive decay. A period of 20 years was taken into consideration and the decay power production per pebble was computed. A display of such each source evolution with time is displayed in Figure 3.

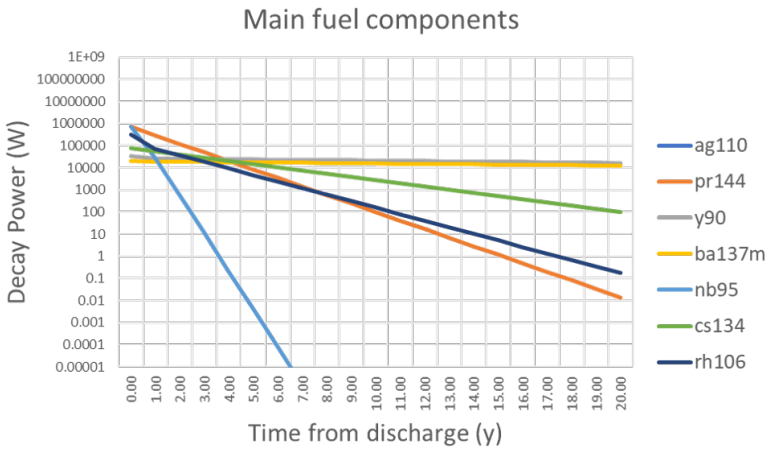

Figure 3: Decay power per pebble (MW) vs. time (years)

5.2.4 Difference between Matrex and CRAM outcomes

As a final analysis, the irradiation and decay ORIGEN runs were performed with both CRAM and MATREX algorithm (with default configurations), in order to point out any existing significant difference in outcomes and computational time. In the irradiation run, computational times required by CRAM and MATREX were similar (0.55s and 0.51s, respectively), while the CRAM decay run required about twice as much as MATREX (0.76s vs 0.31s). With regards to concentration, on the other hand outcomes were similar. Indeed, comparing the depleted masses nuclide by nuclide, the largest difference between the outputs from the two algorithms is in the order of 0.24 milligrams, i.e. the 0.001 percent of the total pebble mass. Similarly, the maximum difference in the decay phase amounted to 0.20 milligrams. It is therefore possible to conclude that, at least with regards to the case simulated in this circumstance, there is no ground to consider that the two solvers provides significantly distinct outcomes, but that, limited to the decay phase, CRAM simulation results more computationally expensive.

6 Conclusion

Aim of this report was to provide context and mathematical background to the fuel depletion computational sequence and the Bateman equation, to describe two algorithms for its solution and to apply them to a case study. After having introduced the scope of analysis in Section 1, Section 2 has broken down the depletion sequence in two steps; a first step consisting in a transport simulation to compute the reaction cross section and a second step represented by the actual Bateman equation. It has been shown that the computation of the one-group cross section required by the latter can be performed via a time-independent transport simulation at equilibrium condition. The Bateman equation was subsequently defined, and two solution methods were presented in Section 3. The first of the two, MATREX, solves the exponential matrix by adding up *step concentrations* computed iteratively. The second method, CRAM, exploits contour integration to solve the depletion equation by approximating the exponential fraction with a rational function. Section 4 presents SCALE, the computational suite employed for the implementation of the computational sequence and provides a description of the ORIGEN input-output structure. Finally, the proposed technique is applied to a case study, and the results are displayed in Section 5. Future works may focus on assessing the accuracy of the solution, comparing it with outputs from other Monte Carlo codes (MCNP, Serpent), and on comparing the computational demands of MATREX and CRAM in a larger number of cases.

References

[1] Bradley T Rearden and Matthew Anderson Jessee. *SCALE code system*. Tech. rep. Oak Ridge National Laboratory (ORNL), 2016.

[2] Anselmo Tomas Cisneros. "Pebble Bed Reactors Design Optimization Methods and their Application to the Pebble Bed Fluoride Salt Cooled High Temperature Reactor (PB-FHR)." PhD thesis. UC Berkeley, 2013.

[3] Ernest Rutherford. *Radio-activity*. University press, 1905.

[4] Harry Bateman. "The solution of a system of differential equations occurring in the theory of radio-active transformations." In: *Proc. Cambridge Phil. Soc., 1908* 15 (1908), pp. 423–427.

[5] Elmer Eugene Lewis and Warren F Miller. "Computational methods of neutron transport." In: (1984).

[6] George I Bell and Samuel Glasstone. *Nuclear reactor theory*. Tech. rep. US Atomic Energy Commission, Washington, DC (United States), 1970.

[7] D Kotlyar, E Fridman, and E Shwageraus. "One-Group Cross-Section Generation for Monte Carlo Burnup Codes: Multigroup Method Extension and Verification." In: *Nuclear Science and Engineering* 179.3 (2015), pp. 274–284.

[8] David Ray Vondy. *Development of a General Method of Explicit Solution to the Nuclide Chain Equations for Digital Machine Calculations (Thesis)*. Tech. rep. Oak Ridge National Lab., Tenn., 1962.

[9] Maria Pusa and Jaakko Leppänen. "Computing the matrix exponential in burnup calculations." In: *Nuclear science and engineering* 164.2 (2010), pp. 140–150.

[10] AE Isotalo and PA Aarnio. "Comparison of depletion algorithms for large systems of nuclides." In: *Annals of Nuclear Energy* 38.2-3 (2011), pp. 261–268.

[11] Maria Pusa et al. *Numerical methods for nuclear fuel burnup calculations*. Aalto University, 2013.

[12] Maria Pusa. "Rational approximations to the matrix exponential in burnup calculations." In: *Nuclear science and engineering* 169.2 (2011), pp. 155–167.

[13] Lloyd N Trefethen, J Andre C Weideman, and Thomas Schmelzer. "Talbot quadratures and rational approximations." In: *BIT Numerical Mathematics* 46.3 (2006), pp. 653–670.

[14] AJ Carpenter, A Ruttan, and RS Varga. "Extended numerical computations on the "1/9" conjecture in rational approximation theory." In: *Rational approximation and interpolation*. Springer, 1984, pp. 383–411.

[15] Robert Endre Tarjan. "Graph theory and Gaussian elimination." In: *Sparse Matrix Computations*. Elsevier, 1976, pp. 3–22.

[16] Donald J Rose and Robert Endre Tarjan. "Algorithmic aspects of vertex elimination on directed graphs." In: *SIAM Journal on Applied Mathematics* 34.1 (1978), pp. 176–197.

[17] Leon Lapidus and Rein Luus. *Optimal control of engineering processes*. Tech. rep. 1967.

[18] OW Hermann and RM Westfall. "ORIGEN-S: SCALE system module to calculate fuel depletion, actinide transmutation, fission product buildup and decay, and associated radiation source terms." In: *Vol. II, Sect. F7 of SCALE: A Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluation, NUREG/CR-0200, Rev* 6 (1998).

Appendix A MATREX simplified algorithm

Algorithm 1 MATREX simplified algorithm

initialize values and vectors (composition vector, step concentration, etc.)
$\vec{C}^0$=$\vec{N}(0)$
for i in nuclides **do**
  for j in nuclides **do**
    if $i \neq j$ **then**

$$A_{ij} = l_{ij}\lambda_i + f_{ij}\sigma_i\Phi_{eq}$$

    else

$$A_{ij} = -\lambda_i - \sigma_i\Phi_{eq}$$

    end if
  end for
end for
for t in time **do**
  while not converged **do**
    for i in nuclides **do**
      for j in nuclides **do**

$$\vec{C}_i^{n+1} = \vec{C}_i^{n+1} + \frac{t}{n+1}A_{ij}\vec{C}_j^n$$

      end for
    end for
    check for step composition convergence
  end while
  $\vec{N}(t)$=$\vec{C}^n$
end for

Appendix B MATREX advanced algorithm

Given a transition matrix $\boldsymbol{A}$, its norm is defined as the minimum between the maximum-row absolute sum and the maximum-column absolute sum [17]

$$|\mathbf{A}| = min\left[max_j \sum_i |a_{ij}|, max_i \sum_j a_{ij}|\right] \tag{15}$$

To maintain precision in the calculation of N by successive approximations, it has been shown that a constraint on the product of the norm of $\boldsymbol{A}$ and t shall be set. Lapidus and Luus [17] have developed a relation showing the number of term required in Equation 11 to have a series truncation error below 0.1 percent:

$$n_{term} = 7|\mathbf{A}|t/2 + 6 \tag{16}$$

By setting the following constraint:

$$|\mathbf{A}|t <= -2ln(0.001) = 13.185, \tag{17}$$

Lapidus and Luus [17] have shown that five-significant figure accuracy can be mantained. Hence, by considering a fixed t, the norm of $\boldsymbol{A}$ shall be confined to 13.185/t, and ORIGEN does that by removing from the transition matrix the short lived nuclides (for which $e^{a_{ii}t} < 0.001$, for which an other solution strategy is followed. For example, in the decay chain A $\rightarrow$ B $\rightarrow$ C, if the decay constant for B is large, a new rate constant is inserted in the matrix for A $\rightarrow$ C and the evolution of B is studied differently. A queue is formed of B and all the short-lived precursors of each long-lived isotope. Such queues extend back up the several chains to the last preceding long-lived precursor and include all nuclides whose transition rates would not allow to satisfy the above constraint. For them, the solution is found via the nuclide chain equations [8]:

$$N_i(t) = N_i(0)e^{A_{ii}t} + \sum_{k=1}^{i-1} N_k(0) \prod_{n=k}^{i-1} \frac{A_{n+1,n}}{-A_{nn}} \left[\sum_{j=k}^{i-1} -A_{jj} \frac{e^{A_{jj}t} - e^{A_{ii}t}}{-A_{ii} + A_{jj}} \prod_{n=i,n\neq j}^{i-1} \frac{-A_{nn}}{-A_{nn} + A_{jj}}\right] \tag{18}$$

Here, $N_1(0)$ is the concentration of the first precursor at the beginning of the time interval, $N_2(0)$ that of the second precursor, etc. In the equation, the first product over isotopes n is the fraction of nuclides remaining after the k-th particular sequence of decays and captures (ORIGEN neglects that when it goes below 10^{-6}). Equation 18 is applied to calculate all contributions to the "queue end-of-interval concentrations" of each short-lived nuclide from the initial concentrations of all others in the queue described above. The beginning-of-interval concentrations of long-lived or stable daughter products are augmented by the appropriate contribution from all nuclides of the queue divided by $e^{A_{ii}t}$. End-of-interval concentration of long-lived nuclides can then be computed through the exponential matrix. [18]

This advanced MATREX methodology can be implemented through the following algorithm:

Algorithm 2 MATREX advanced algorithm

initialize values and vectors (composition vector, step concentration, etc.)
$\vec{C}^0$=$\vec{N}(0)$
for i in nuclides **do**
  for j in nuclides **do**
    if $i \neq j$ **then**

$$A_{ij} = l_{ij}\lambda_i + f_{ij}\sigma_i\Phi_{eq}$$

    else

$$A_{ij} = -\lambda_i - \sigma_i\Phi_{eq}$$

    end if
  end for
end for
for t in time **do**
  $T = t - t_{old}$
  $\vec{N}(0)$=$\vec{N}(t_{old})$
  $\vec{N}_{long}(0)$=$\vec{N}(0)$
  for i in nuclides **do**
    if $e^{-}A_{ii}T < 0.001$ **then**
      Save i in vector shortlived
      for j in nuclides **do**
        if $j \notin$ shortlived **then**
          for k in nuclides **do**
            if $k \notin$ shortlived **then**
              Correct $A_{jk} = A_{jk} + A_{ji}A_{ik}/A_{ii}$
            end if
          end for
        end if
      end for
      Remove i-th row and column from **A** and i-th row from vector $\vec{N}_{long}(0)$
    end if
  end for
  $\vec{C}^0$=$\vec{N}_{long}(0)$
  for s in shortlived **do**
    Build s-th queue of concentration vector q_s using the concentrations from $\vec{N}(0)$
    Compute $N_s(T)$ using Equation 18 applied to its queue vector q_s
  end for
  for i in nuclides **do**
    if $i \notin$ shortlived **then**
      Update $N_i(0) = N_i(0) + \left[\sum_{s\in\text{shortlived}} N_s(T)A_{is}\right] / e^{A_{ii}T}$
    end if
  end for
  while not converged **do**
    for i in nuclides **do**
      if $i \notin$ shortlived **then**
        for j in nuclides **do**
          if $j \notin$ shortlived **then**

$$\vec{C}_i^{n+1} = \vec{C}_i^{n+1} + \frac{t}{n+1}A_{ij}\vec{C}_j^{n}$$

          end if
        end for
      end if
    end for
    check for step composition convergence
  end while
  $\vec{N}_{long}(t)$=$\vec{C}^n$
  Build $\vec{N}(t)$ merging $\vec{N}_{long}(t)$ and $N_s(T)$ $\forall s \in$ shortlived
  $t_{old} = t$
end for

Appendix C CRAM algorithm

Algorithm 3 CRAM simplified algorithm

initialize values and vectors (composition vector, step concentration, etc.)
for i in nuclides **do**
  for j in nuclides **do**
    if $i \neq j$ **then**

$$A_{ij} = l_{ij}\lambda_i + f_{ij}\sigma_i\Phi_{eq}$$

    else

$$A_{ij} = -\lambda_i - \sigma_i\Phi_{eq}$$

    end if
  end for
end for
Define order of Chebyshev rational fraction K based on required tolerance [13]
for t in time **do**
  Compute $\hat{r}_{K,K}$ order 1,2, ..., $K/2$ poles a_k and residues θ_k from Chebyshev coefficients using a polynomial root finder [13].
  Initialize $\vec{N}(t) = a_0\vec{N}(0)$
  while k $<$ K/2 **do**
    Build $\tilde{\mathbf{A}} = \mathbf{A}t + \theta_k\mathbf{I}$
    Solve the linear system $\tilde{\mathbf{A}}\vec{x} = a_i\vec{N}(0)$ [1]
    Initialize $\vec{N}(t) = \vec{N}(t) - Re(\vec{x})$
  end while
end for

[1] LU factorization and Gaussian elimination algorithms not included here. Interested readers might consult [15] and [16]