Milestone 2.1 Report: Demonstration of SaltProc

Enabling Load Following Capability in the Transatomic Power MSR

Author
Andrei RYKHLEVSKII

Principal Investigator
Kathryn D. HUFF

UIUC-ARFC-2019-04

June 28, 2019

ADVANCED REACTORS AND FUEL CYCLES
DEPT. OF NUCLEAR, PLASMA, & RADIOLOGICAL ENGINEERING
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

This research is being performed using funding received from the Department of Energy ARPA-E MEITNER Program (award DE-AR0000983) and the Blue Waters sustained-petascale computing project, which is supported by the National Science Foundation (awards OCI-0725070 and ACI-1238993) and the state of Illinois. Blue Waters is a joint effort of the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Applications.

1 Introduction

The University of Illinois, Urbana-Champaign (UIUC) is engaged in work to develop a fuel processing system that enables load-following in Molten Salt Reactors (MSRs), an important ability that allows nuclear power plants to ramp electricity production up or down to meet changing electricity demand. Nuclear reactions in MSRs produce unwanted byproducts (such as xenon and krypton) that can adversely affect power production. In steady, baseload operation, these byproducts form and decay at the same rate. When electricity production is ramped down, however, the byproducts start to be produced at a greater rate than they decay, leading to a buildup within the reactor. When power production must be once again increased, the response rate is slowed by the time needed for the byproducts to reach their equilibrium level (determined by the radioactive decay half-life, which is on the order of hours). Thus, buildup of these unwanted byproducts resulting from ramping down inhibit proper load following for molten salt reactors. Fortunately, MSRs transport fuel in a flowing molten salt fuel loop, which means that a section of the reactor, outside the core, can be leveraged for fuel processing and "cleanup." The team will determine the feasibility of removal of these unwanted byproducts and design a fuel reprocessing system, removing a major barrier to commercialization for molten salt reactors.

Toward this work, we initiated the Fuel Cycle Simulation task (Task 2) in August 2018 to more realistically model the online reprocessing system of the Transatomic Power (TAP) MSR. A Python toolkit, SaltProc v0.1 [1–3], was developed to represent the simplified online fuel salt processing of a Molten Salt Breeder Reactor (MSBR). More recently, an advanced SaltProc version (SaltProc v0.2) was developed to generically simulate complex molten salt fuel reprocessing systems, including the TAP system, incorporating user-parametrized components into the fuel salt processing design. This report summarizes the progress we have made towards milestone **M2.1: Demonstrate SaltProc** and the steps toward the subsequent Task 2 objectives.

2 Milestone objectives

The finalized work plan for this project (DOE ARPA-E MEITNER award DE-AR0000983) formulated the goal of Milestone 2.1 as follows:

> "Initial demonstration of fuel cycle simulation package working together with Monte Carlo to complete full core TAP reactor depletion calculation. SaltProc will use separations efficiencies and dynamics based on work in Task 1 and will be coupled with Serpent 2 where Monte Carlo results will be done to <10% relative error accuracy."

This milestone has been completed through significant open source software development and large scale high performance computing demonstration. In this document, we

will discuss and demonstrate the capabilities of SaltProc v0.2 which together satisfy this milestone goal. In particular SaltProc v0.2.0 can:

1. Read a user-defined Serpent 2 [4] input template file including the reactor model geometry, material composition, total heating power, and boundary conditions.
2. Read a user-defined `.json` input file with key parameters and structure of the fuel salt reprocessing system.
3. Run Serpent 2 in parallel mode to perform depletion calculations on the Blue Waters supercomputer.
4. Read the resulting depleted fuel composition file and store it in an HDF5 output database [5].
5. Remove poisons from the fuel's isotopic composition by passing information through the user-parametrized components of the fuel salt processing system. For demonstration proposes, SaltProc v0.2 used user-defined constant separation efficiencies and can handle variable efficiencies once they are defined via Task 1.
6. Replace discarded fuel salt mass by adding fresh salt with a user-defined isotopic composition (e.g., low-enriched uranium (LEU) 5% and 19.79%, for this work).
7. Record the fuel salt composition after salt reprocessing; waste streams from each component of the reprocessing system; and other major core parameters such as multiplication factor, burnup, total fissile mass, effective delayed neutron fraction, and breeding ratio.

3 The Transatomic Power Molten Salt Reactor concept

The TAP concept is a 1250 MW_{th} MSR with a LiF-based uranium fuel salt [6]. This concept uses configurable zirconium hydride ($ZrH_{1.66}$) rods as the moderator while most MSR designs usually propose high-density reactor graphite. Zirconium hydride can achieve the same degree of thermalization as graphite with a much smaller volume. Compared to graphite, which shrinks and swells over time under irradiation, the cladded zirconium hydride has a much longer lifespan in extreme operational conditions - high temperature, large neutron flux, chemically aggressive salt. Finally, zirconium hydride is a nonporous material that absorbs much fewer neutron poisons (e.g., krypton, xenon) compared to high-density reactor graphite [6–8].

3.1 TAP design description

The TAP design (figure 1) is very similar to the original Molten Salt Reactor Experiment (MSRE) design developed by Oak Ridge National Laboratory (ORNL) [9] but with two

major innovations: the fuel salt composition and the moderator. The LiF-BeF_2-ZrF_4-UF_4 salt used in MSRE has been substituted with a LiF-UF_4 salt allowing the uranium concentration within the fuel salt to be increased from 0.9 to 27.5% while maintaining a relatively low melting point (490°C compared with 434°C for the original MSRE's salt) [8]. The graphite has a very high thermal scattering cross section which would make it an excellent moderator but it has a few major drawbacks. First, due to the low lethargy gain per collision, the core requires a large volume of graphite to reach criticality, leading to a larger core and obstructing the core power density. Second, even special reactor-grade graphite has relatively high porosity, meaning, it holds gaseous Fission Products (FPs) (e.g., tritium, xenon) in its pores. Third, the reactor graphite lifespan in a commercial reactor is only 10 years [10]. To resolve these issues, the TAP concept uses an alternative moderator, zirconium hydride, allowing for a more compact core and a significant increase in power density. These two innovative design choices, together with a configurable moderator (the moderator-to-fuel ratio can be changed during regular maintenance shutdown), facilitate the commercial deployment of this conceptual design viable in the commercially available 5% LEU fuel cycle.

The primary loop of the TAP MSR consists of the reactor core volume moderated by the silicon carbide (SiC) cladded zirconium hydride rods, pumps, and primary heat exchanger. The pumps circulate the LiF-(Act)F_4 fuel salt through the primary loop. The pumps, vessels, tanks, and piping are made of a corrosion resistant nickel-based alloy (similar to Hastelloy-N[1]) in various molten salt environments. Inside the reactor vessel, near to the zirconium hydride moderator rods, the fuel salt is in a critical configuration and generates heat. Table 1 contains details of the TAP system design taken from the technical white paper [6], the neutronics overview [7], and the ORNL analysis of the TAP design [8, 11].

Table 1: Summary of principal data for the TAP MSR (reproduced from [6, 11]).

Thermal power	1250 MW_{th}
Electric power	520 MW_e
Gross thermal efficiency	44%
Outlet temperature	620°C
Fuel salt components	LiF-UF_4
Fuel salt composition	72.5-27.5 mole%
Uranium enrichment	5% ^{235}U
Moderator	Zirconium Hydride ($ZrH_{1.66}$) rods (with silicon carbide cladding)
Neutron spectrum	Thermal/Epithermal

[1] Hastelloy-N is very common in reactors now but have been studied and developed at ORNL in a program that started in 1950s.

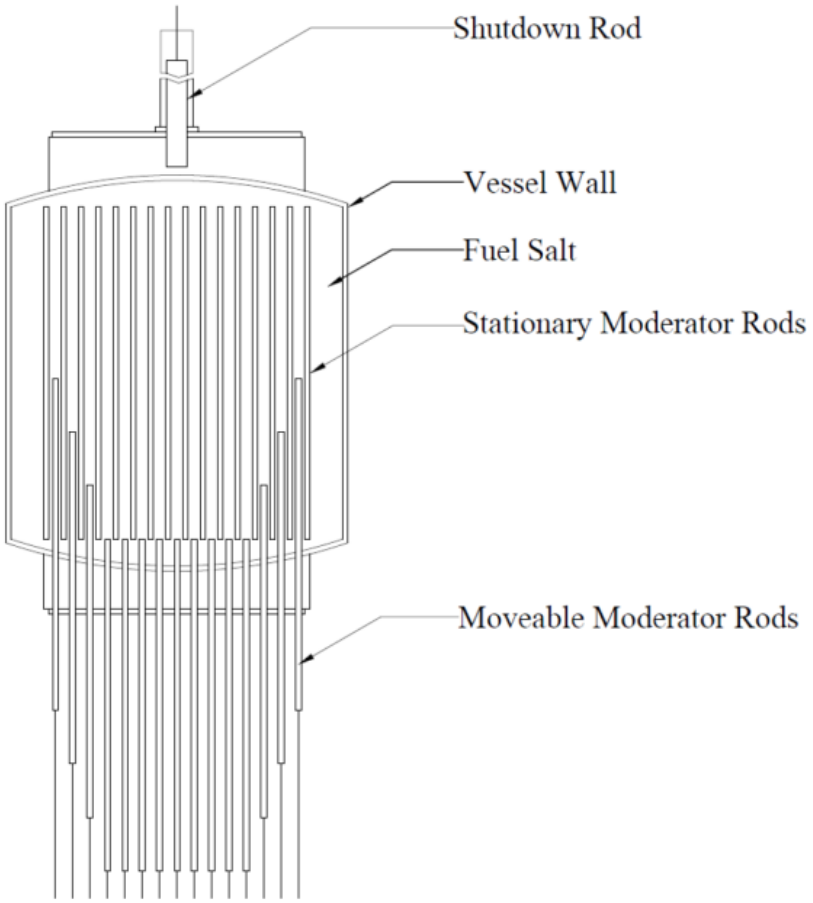

Figure 1: Schematic view of the TAP MSRshowing the movable moderator rod bundles and the shutdown rod (figure reproduced from Transatomic Power White Paper [6]).

3.2 TAP core design

In the TAP core (Figure 2), lattices of SiC clad moderator rods form the moderator assemblies around which the fuel salt flows (Figure 1). The TAP reactor pressure vessel is a cylinder with an inner radius of 150 cm, a height of 350 cm, and a wall thickness of 5 cm. The moderator-to-fuel ratio, or salt volume fraction (SVF), in the core can be varied during operation to shift the spectrum from intermediate to thermal energies to maximize fuel burnup. Intermediate energies are used at Beginning of Life (BOL) and are shifted to thermal at End of Life (EOL). During operation the SVF can be varied by inserting fixed-sized moderator rods from the bottom of the reactor vessel, similarly to moving the control rods in a Boiling Water Reactor (BWR), as shown in Figure 1. For the TAP reactor, EOL occurs when the maximum number of moderator rods are inserted into the core and further injection of fresh fuel salt does not change criticality. Unmoderated salt flowing in the annulus between the core and the vessel wall provides potential reduction of fast neutron flux at the vessel structural material [7].

3.3 TAP reprocessing system structure and simulation approach

The TAP nuclear island contains a FP removal system. Gaseous FPs are continuously removed using an off-gas system while liquid and solid FPs are extracted via a chemical processing system. A small quantity of fresh fuel salt is regularly added to the primary loop as byproducts are gradually removed. This process maintains a constant fuel salt mass and keeps the reactor critical. In contrast with the MSBR reprocessing system, the TAP does not require a protactinium separation and isolation system because it operates in a single-stage uranium-based fuel cycle. The authors of the TAP concept detailed three distinct fission product removal methods [7]:

Off-Gas System: Gaseous fission products such as krypton and xenon are removed, compressed, and stored temporarily until they have decayed to background radiation levels. Trace amounts of tritium are also removed and bottled in a liquid form via the same process. The off-gas system also removes a small fraction of the noble metals.

Metal Plate-Out/Filtration: Removes solid noble and semi-noble metal fission products as they plate out onto a nickel mesh filter located in a side stream of the primary loop.

Liquid Metal Extraction: Lanthanides and other non-noble metals stay dissolved in the fuel salt. These elements generally have a lower capture cross section and thus absorb fewer neutrons than ^{135}Xe but their extraction is essential to ensuring normal operation. In the TAP reactor, lanthanide removal is accomplished via a liquid-metal/molten salt extraction process similar to that developed for MSBR by ORNL [10]. The process converts the dissolved lanthanides into a well-understood oxide waste form, similar to that for Light Water Reactor (LWR) spent nuclear fuel (SNF). This oxide waste comes out of

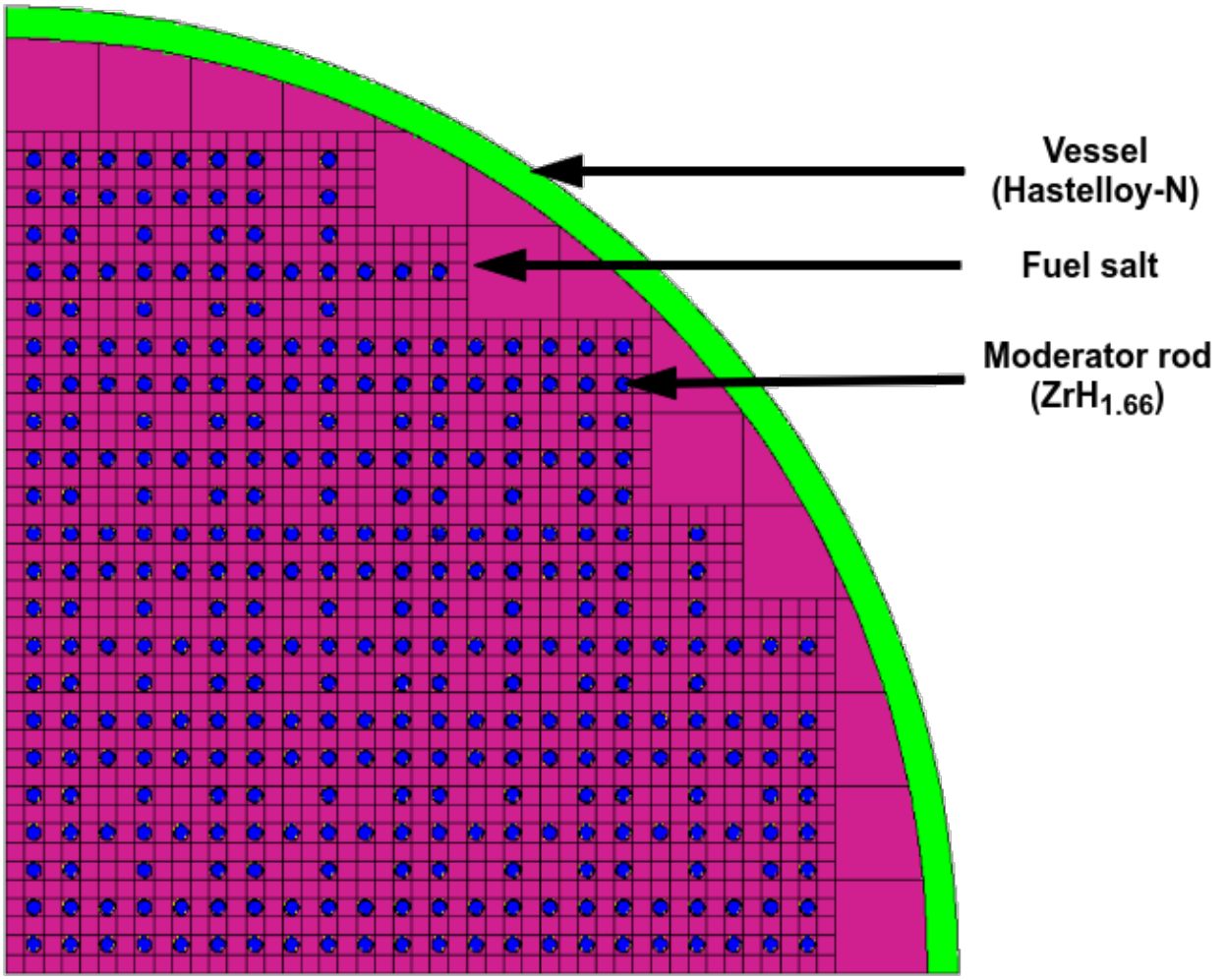

Figure 2: The TAP MSR schematic core view showing moderator rods in the BOL position defined by their point design (figure reproduced from ORNL/TM-2017/475 [11]).

the TAP reprocessing plant in ceramic granules, which can be sintered into another convenient form for storage.

Figure 3 shows a principal design of the TAP primary loop including an off-gas system, nickel mesh filter, and lanthanide chemical extraction facility. Similarly to MSBR, the off-gas system is based on a simple process of helium sparging through the fuel salt with consequent gas bubbles removed before returning the fuel salt back to the core. One very notable difference is the MSBR gas separation system helium injection and subsequent transport of the voids run throughout the primary loop, including the core, for at least 10 full loops [10]. This system presents a significant concern to the safety and stability of operation due to the increase of void fraction in the fuel salt when it enters back to the core, causing unpredictable changes in reactivity. This drawback can be overcome by using an effective gas separator to strip helium/xenon bubbles before returning the salt back to a primary loop (Figure 3, blue block).

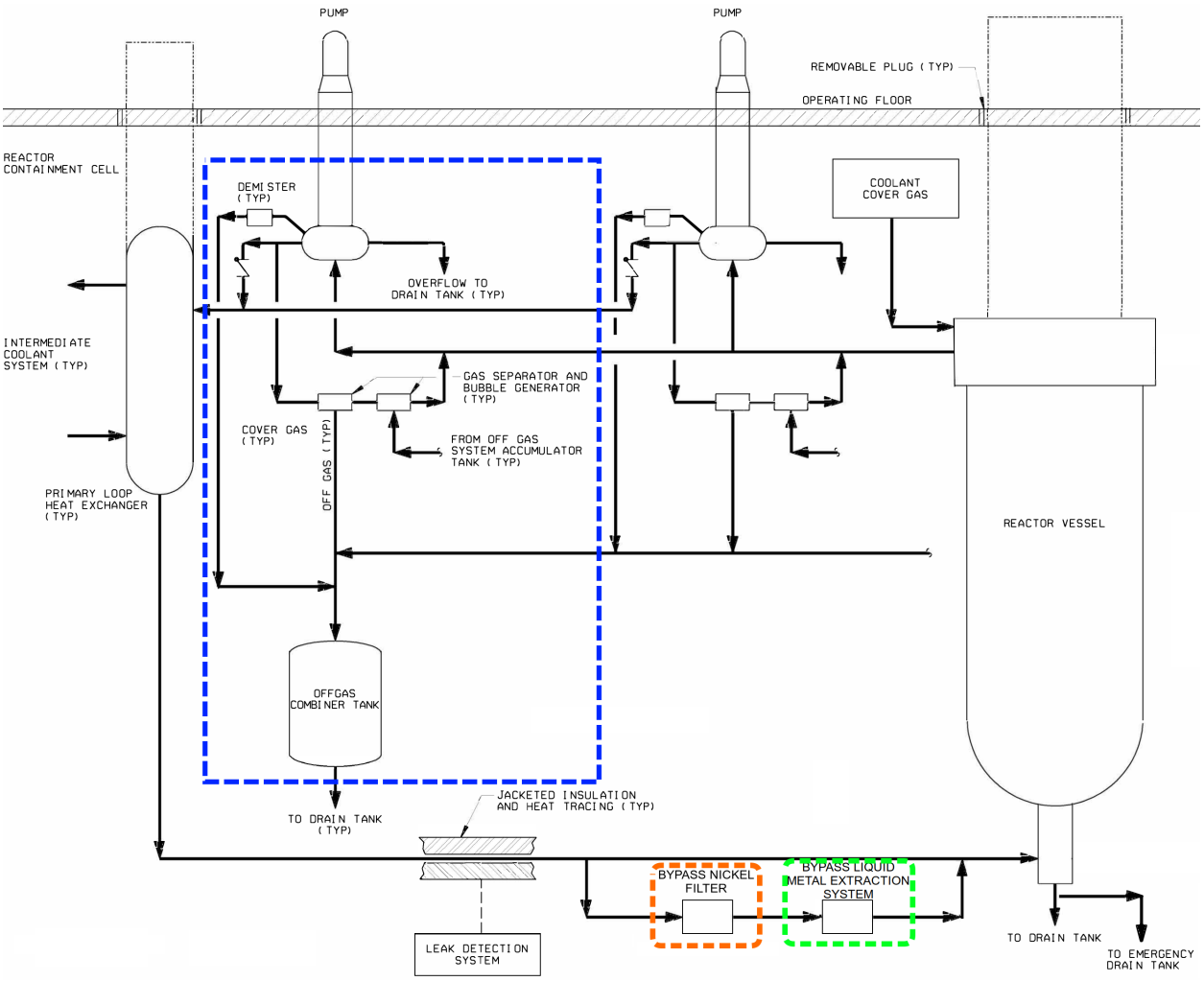

Figure 3: Simplified TAP primary loop design including off-gas system (blue), nickel filter (orange) and liquid metal extraction system (green) (reproduced from [12]).

Solid noble and semi-noble metal fission products tend to plate out onto the metal surfaces including piping, heat exchanger tubes, reactor vessel inner surface, etc. Previous research by ORNL [10] concluded that about 50% of noble and semi-noble metals would plate out inside MSBR systems without any special treatment. To improve the extraction efficiency of these fission products, the TAP concept employs a nickel mesh filter located in a bypass stream in the primary loop (Figure 3, orange block). The main idea of this filter is to create a maze with a large metal (nickel) surface area. The fuel salt flows throughout the filter and the noble metals plate-out on the filter internal surface.

This Liquid Metal Extraction process for the TAP concept has been adopted from the MSBR. The MSRE demonstrated a liquid-liquid extraction process for removing rare earths and lanthanides from the fuel salt and estimated its efficiency.

The TAP project reported a detailed list of elements for removal and removal efficiencies (Table 2). We used data from TAP neutronics whitepaper [7] for the SaltProc v0.2 demonstration case without any modifications.

Table 2: The effective cycle times for fission products removal from the TAP MSR (reproduced from [13] and [7]).

Processing group	Nuclides	Removal Rate (s^{-1})	Cycle time (at full power)
Elements removed in MSBR concept and adopted for the TAP [10]			
Volatile gases	Xe, Kr	5.00E-2	20 sec
Noble metals	Se, Nb, Mo, Tc, Ru, Rh, Pd, Ag, Sb, Te	5.00E-2	20 sec
Seminoble metals	Zr, Cd, In, Sn	5.79E-8	200 days
Volatile fluorides	Br, I	1.93E-7	60 days
Rare earths	Y, La, Ce, Pr, Nd, Pm, Sm, Gd	2.31E-7	50 days
	Eu	2.32E-8	500 days
Discard	Rb, Sr, Cs, Ba	3.37E-9	3435 days
Additional elements removed [7,13]			
Volatile gases	H	5.00E-2	20 sec
Noble metals	Ti, V, Cr, Cu	3.37E-9	3435 days
Seminoble metals	Mn, Fe, Co, Ni, Zn, Ga, Ge, As	3.37E-9	3435 days
Rare earths	Sc	3.37E-9	3435 days
Discard	Ca	3.37E-9	3435 days

We simulated TAP MSR depletion in SaltProc v0.2 using reprocessing cycle times from Table 2. The online reprocessing system design details, and a full-core reactor Serpent model (section 5.1) to capture the dynamics of fuel composition evolution during reactor operation.

4 The SaltProc modeling and simulation code

The first version of the SaltProc Python tool was developed in 2018 as a part of M.S. thesis to calculate MSR fuel composition evolution taking into account an online reprocessing system [2, 3]. The tool was designed to expand depletion capabilities of Serpent 2 for modeling liquid-fueled MSR with an online fuel reprocessing system. SaltProc v0.1 uses HDF5 [5] to store data and uses the PyNE Nuclear Engineering Toolkit [14] to parse Serpent 2 output. SaltProc v0.1 is an open-source Python package that uses a batch-wise approach to simulate continuous feeds and removals in MSRs.

SaltProc v0.1 only allows 100% separation efficiency for either specific elements or groups of elements (e.g., Processing Groups as described in Table 2) at the end of the specific cycle time. This simplification neglects the reality that the salt spends an appreciable amount of time out of the core, in the primary loop pipes and heat exchanger. This approach works well for fast-removing elements (gases, noble metals) which should be removed after each depletion step. Unfortunately, for the elements with longer cycle times (i.e. rare earths which should be removed every 50 days) this simplified approach leads to oscillatory behavior of all major parameters [1].

The capabilities of the SaltProc, paired with the Monte Carlo software Serpent 2, were demonstrated using the full-core MSBR design for a simplified case using ideal removal efficiency (100% of mass for target elements removed) [1]. The preliminary version of the SaltProc architecture and principal structure were not designed for flexible implementation of sophisticated online reprocessing systems, including realistic physics/chemistry-based extraction efficiencies.

We completely re-factored SaltProc v0.1 using Object-Orienting Programming (OOP) to create a generic, comprehensive tool to realistically model any MSR reprocessing plant while taking into account non-ideal or variable extraction efficiencies and mass balance between the core and processing plant.

4.1 SaltProc v0.2 architecture

The SaltProc v0.2 Python toolkit couples directly with the Serpent 2 input and output files, to allow the reprocessing system couples to depletion calculation. Existing PyNE interfaces are employed to parse Serpent output while newly developed interfaces handle input. The standard OOP features of Python 3 are used to create a flexible, user-friendly tool with increased potential for further improvement and collaboration. Figure 4 illustrates the SaltProc v0.2 class structure, consisting of 4 main classes:

Depcode. Contains attributes and methods for reading the user's input file for the depletion software, initial material (e.g., fuel and/or fertile salt) composition, principal parameters for burnup simulation (e.g., neutron population and number of cycles for Monte Carlo neutron transport), and running the depletion code.

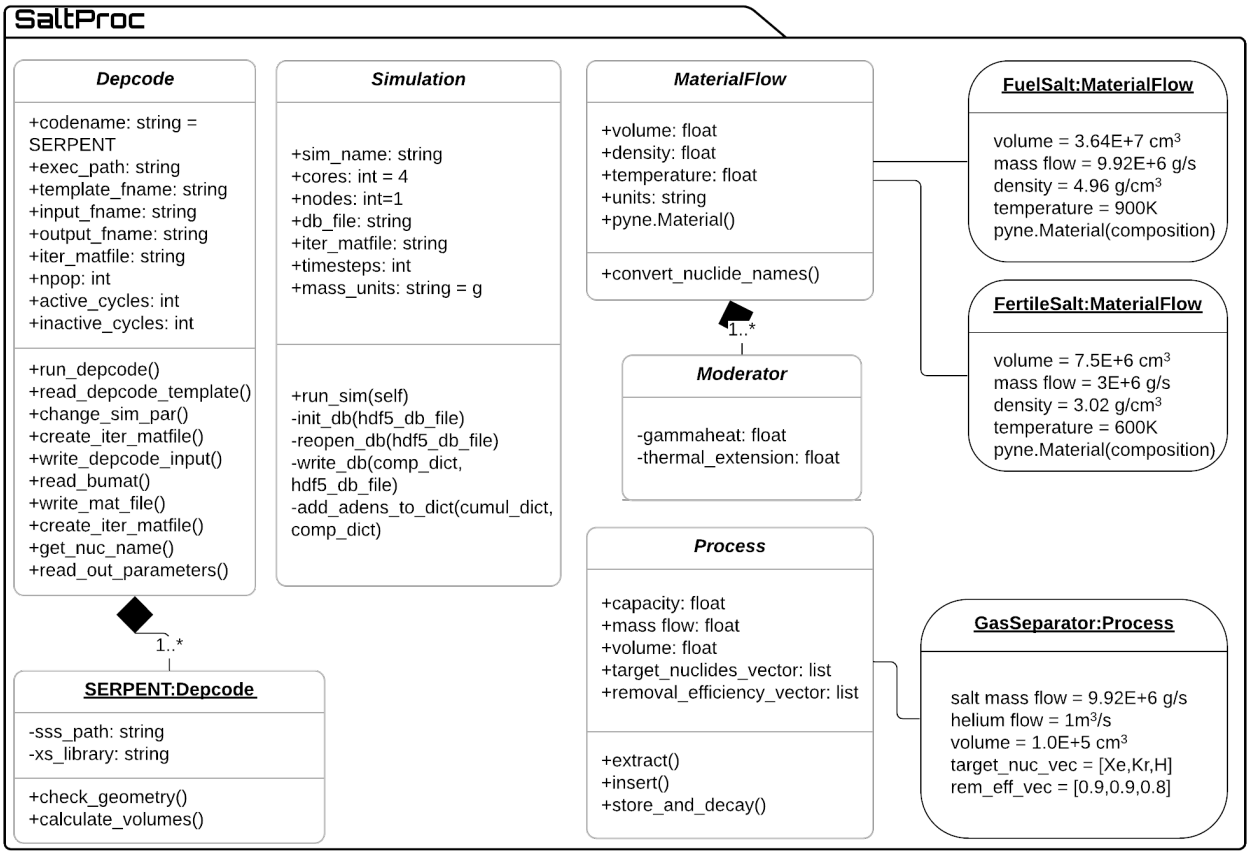

Figure 4: SaltProc v0.2 python package class diagram in UML notation with examples of object instances.

Simulation. Runs Serpent depletion step, creates and writes HDF5 database, tracks time and converts isotopic composition vector nuclide names from Serpent to human-readable format.

MaterialFlow. Each *MaterialFlow* object represents the material flowing between *Process* objects. All instances of this class contain an isotopic composition vector (PyNE Material object initialized from Serpent output file **dep.m**), mass flow rate, temperature, density, volume, and void fraction. Existing PyNE Material capabilities allow us to easily convert the units of the isotopic composition vector (e.g., from atomic density provided by Serpent to a mass fraction or absolute mass in desired units) and decay the material (i.e. model the MSBR protactinium decay tank), calculate decay heat, activity, and dose. The main purpose of the *MaterialFlow* object is to pass detailed information about the salt - starting at the MSR vessel outlet - throughout reprocessing components (*Processes*). These processes modify the *MaterialFlow* object before depleting the material in the next Serpent

burnup step.

Process. Each *Process* object represents a realistic fuel processing step characterized by its throughput rate, volumetric capacity, extraction efficiency for each target element (can be a function of many parameters), waste streams, and other parameters specific to the particular process. Feed *Process* injects fresh fuel salt *MaterialFlow* directly into the reactor core (e.g., adding fissile material with a specific mass flow rate to *MaterialFlow* after performing all removals).

The proposed class structure provides outstanding flexibility in simulating various MSR fuel processing system designs. A library of various *MaterialFlow* (e.g., fuel salt flow, fertile salt flow, refueling salt flow) and *Process* (e.g., helium sparging facility, gas separator, lanthanide removal component) objects will be created to allow a user to quickly create a model of a desired reprocessing scheme. At runtime, the user will connect *Process* objects in series or parallel with *MaterialFlow* objects to form a comprehensive reprocessing system. The user will also be able to create custom objects with desired attributes and methods as well as contribute them back to the code package using GitHub (https://github.com/ arfc/saltproc).

4.2 SaltProc v0.2 flowchart

Figure 5 illustrates the online reprocessing simulation algorithm, coupling Serpent with SaltProc v0.2. To perform a depletion step, SaltProc v0.2 reads a user-defined Serpent template file. The template contains input parameters such as geometry, material, isotopic composition, neutron population, criticality cycles, total heating power, and boundary conditions. SaltProc v0.2 fills in the template file and runs the Serpent single-step depletion. After the depletion calculation, SaltProc v0.2 reads the depleted fuel composition file into the *MaterialFlow* object (*core_outlet* in Figure 5). This *MaterialFlow* object contains an isotopic composition vector, total volume of material, total mass, mass flow rate, density, temperature, void fraction, etc. For the simplest reprocessing case, when all fuel processing components are located in-line (100% of total material flow goes through a chain of separation components), the *core_outlet* object is flowing sequentially between *Processes*, and each *Process* is removing a mass fraction of the target elements with specified extraction efficiency. Afterward, the removed material mass is replenished by fresh fuel salt to maintain the salt inventory in a primary loop. Finally, the resulting isotopic composition after reprocessing is stored in the HDF5 database and dumped in a new composition file for the next Serpent depletion run. SaltProc v0.2 also stores isotopic composition before reprocessing and waste stream from each fuel processing component in a database.

For a more general case with multiple concurrent extraction processes, a separate *MaterialFlow* object is created for each branch with a user-defined mass flow rate (e.g. 90% of total mass flow rate flows through the left branch and 10% through the right branch). The total mass and isotopic composition vector for each *MaterialFlow* object is calculated as a fraction of incoming *core_outlet* flow. Then, each *MaterialFlow* object is passed through

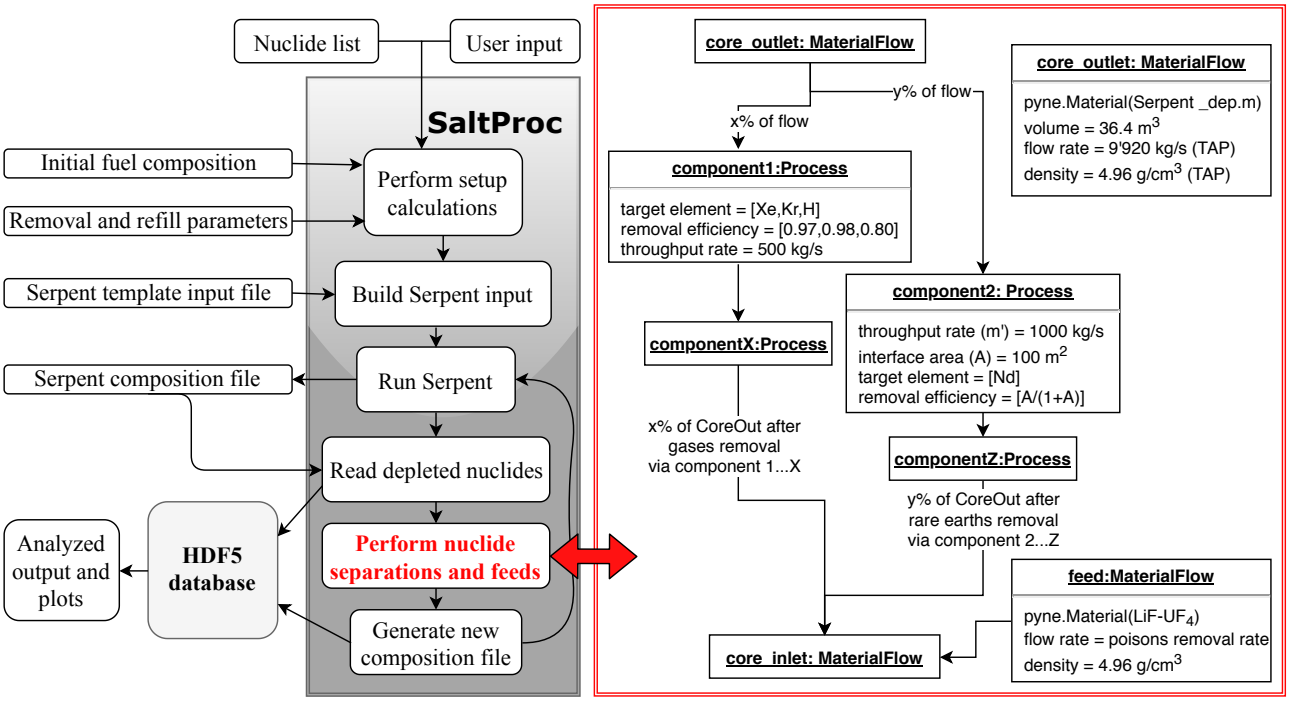

Figure 5: SaltProc v0.2 python package flow chart.

a cascade of *Processes* to separate selected chemical elements with a specific efficiency. Finally, the *MaterialFlow* object from the left branch is merged with the right-hand-side and just like the previous case, a fresh fuel salt feed compensates the loss of mass in the separation facilities and keeps the fuel salt mass in the primary loop constant.

The class diagram (Figure 4) allows user to model the operation of a complex, multi-zone, multi-fluid MSR and is sufficiently generalized to represent numerous reactor systems. The refactored version of SaltProc stores and edits the isotopic composition of the fuel stream, making it a flexible tool to model any geometry: an infinite medium, a unit cell, a multi-zone simplified assembly, or a full core. This flexibility allows the user to perform simulations of varying fidelity and computational intensity. SaltProc v0.2 is an open-source tool (though having Serpent installed is required to use SaltProc v0.2) available on GitHub. It leverages unit and continuous tests crucial for sustainable development [15]. Clickable documentation is available through Sphinx, a documentation generator, for ease of use [16]. In summary, the development of SaltProc v0.2 is focused on producing a generic, flexible and expandable tool to give the Serpent 2 Monte Carlo code the ability to conduct advanced in-reactor fuel cycle analysis as well as simulate many online refueling and fuel reprocessing systems.

5 SaltProc demonstration case

The SaltProc v0.2 modeling and simulation tool is demonstrated for the TAP MSR with static core geometry, LEU 5% startup composition [7] and the three following fueling scenarios: (1) no FP removal or feed (Serpent only); (2) a 5% LEU online feed; and (3) a 19.79% LEU online feed. The primary focus and the bulk of the analysis herein has been on the last fueling scenario using 19.79% LEU. All calculations are run with Serpent version 2.1.31 and the JEFF-3.1.2 nuclear data library [17,18].

5.1 Serpent 2 full-core model

The advanced geometric surfaces and transformation capabilities of Serpent [17] are employed to represent the TAP core. Figure 6 shows the XY section of the whole-core configuration at the expected operational level of the reactor with all control rods fully withdrawn. Figures 7 and 8 depict a longitudinal section of the reactor. This model contains the moderator rods with silicon carbide cladding, inlet/outlet plena, and the pressure vessel (Table 3). The fuel salt flows around rectangular moderator assemblies consisting of lattices of small-diameter zirconium hydride rods in a corrosion-resistant material. The salt volume fraction (SVF) in the core is a parameter similar to the widely-used moderator-to-fuel ratio and is defined as:

$$SVF = \frac{V_F}{V_F + V_M} = \frac{1}{1 + V_M/V_F} \tag{1}$$

where

$$\begin{aligned} V_F &= \text{in-core fuel salt volume} \\ V_M &= \text{in-core moderator volume} \\ V_M/V_F &= \text{in-core moderator-to-fuel-salt ratio} \end{aligned}$$

The SVF for model herein is 0.907268 which means the modeled core is under-moderated and has an intermediate/fast spectrum.

To represent the reactivity control system the model has: (1) control rod guide tubes made of nickel-based alloy; (2) control rods represented as hollow 70-30% Gd_2O_3-Al_2O_3 cylinders with a thin Hastelloy-N coating [11]; (3) the air inside the guide tubes and control rods. The control rod assembly design has yielded a cluster of 25 rods that provide a total reactivity worth of 1121pcm[2].

The control rod cluster is modeled using the **TRANS** Serpent 2 feature, which allows easy change of the control rods position during simulation. All figures of the core in this report were generated using the built-in Serpent plotter.

[2] 1 pcm = $10^{-5}\Delta k_{eff}/k_{eff}$.

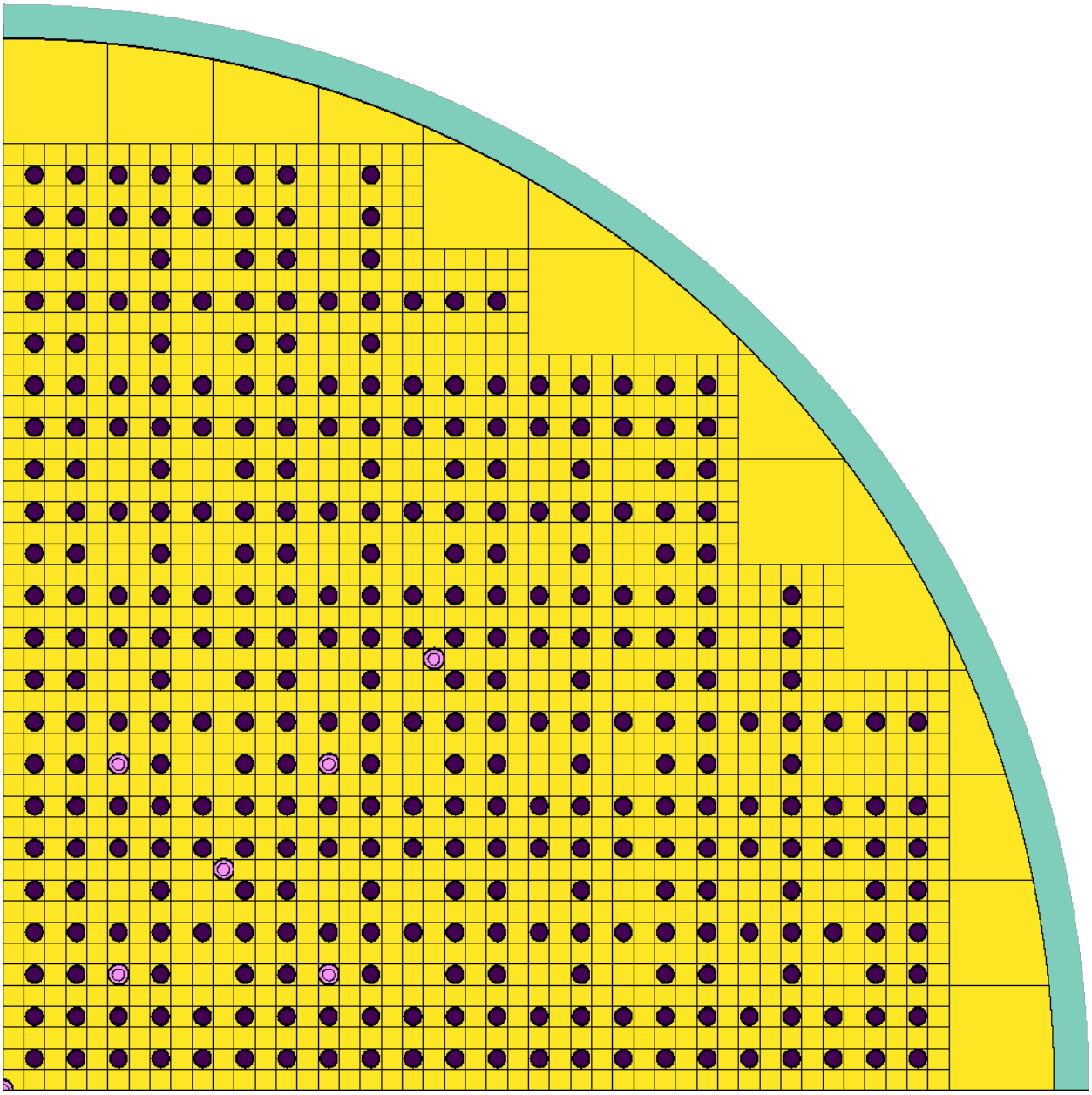

Figure 6: An *XY* section of the TAP model at the horizontal midplane with fully withdrawn control rods at BOL (SVF= 0.907268). The violet color represents zirconium hydride, and the yellow represents fuel salt. The blue color shows Hastelloy-N, the alloy used for the vessel wall, and the white color is the air.

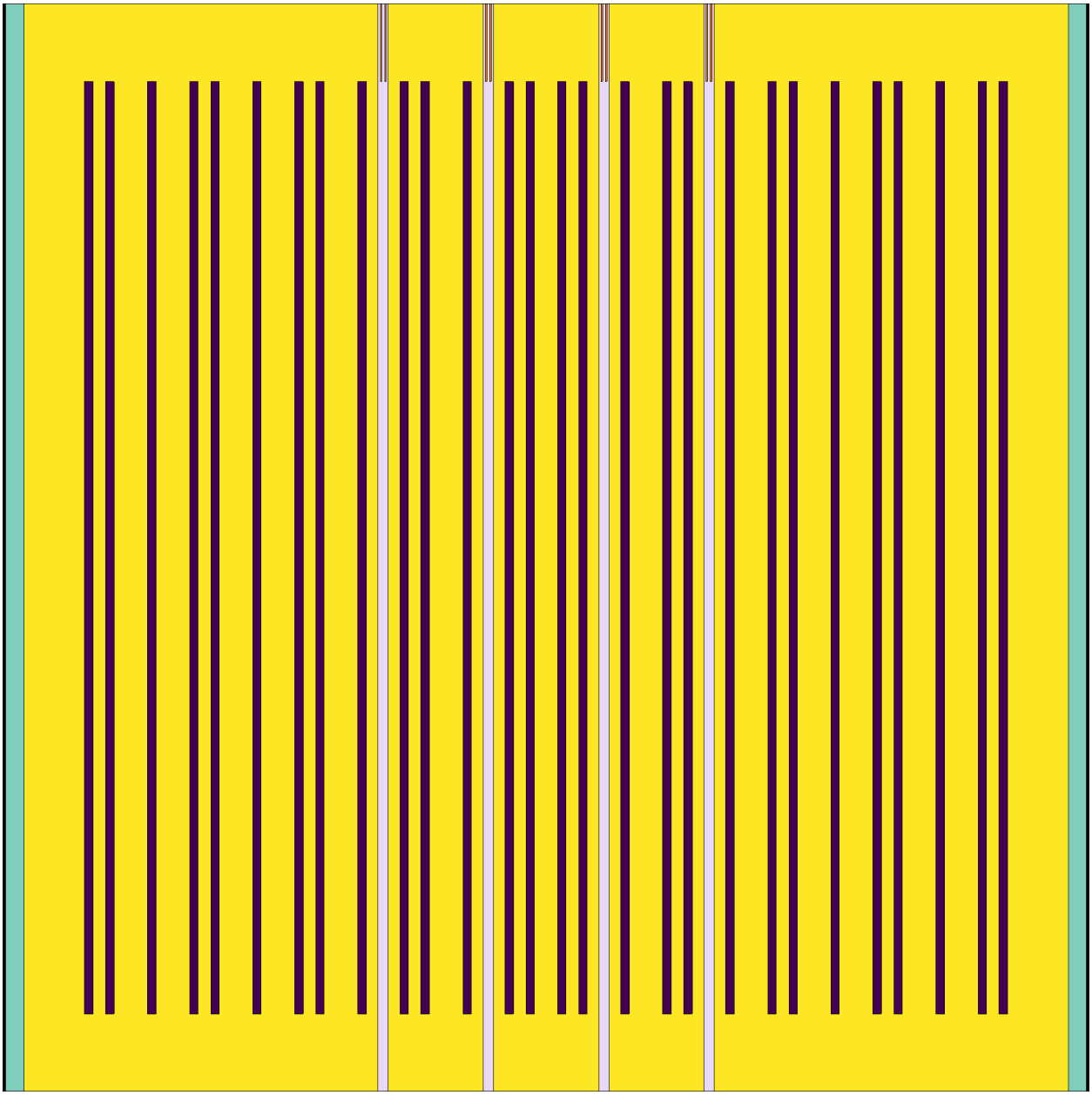

Figure 7: An XZ section of the TAP model.

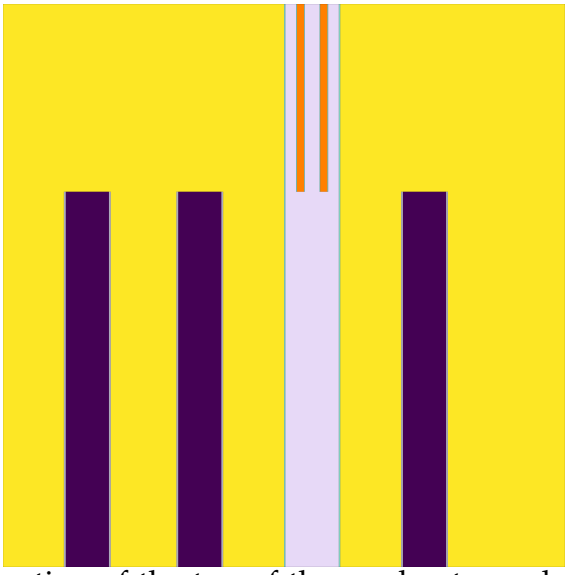

Figure 8: Zoomed XZ section of the top of the moderator rods and guide tubes for the TAP model. The orange color shows 70–30% Gd_2O_3–Al_2O_3 ceramic absorbers used for the control rods.

Table 3: Geometric parameters for the full-core 3D model of TAP (reproduced from Betzler *et al.* [11]).

Component	**Parameter**	Value	Unit
Moderator rod	Cladding thickness	0.10	cm
	Radius	1.15	cm
	Length	3.0	m
	Pitch	3.0	cm
Moderator assembly	Array	5×5	rods$\times$rods
	Pitch	15.0	cm
Core	Assemblies	268	assemblies/core
	Inner radius	1.5	m
	Plenum height	25.0	cm
	Vessel wall thickness	5.0	cm

5.2 Simulated fuel reprocessing system

We thoroughly analyzed the original TAP reprocessing system design (figure 3) and neutron poison removal rates (table 2) to determine a suitable reprocessing scheme for the SaltProc v0.2 demonstration (figure 9).

The gas removal components (the sparger and entrainment separator) are located inline because the estimated full loop time for the fuel salt is about 18 sec and has an approximately equal cycle time (table 2). To remove all volatile gases every 20 sec, the fuel reprocessing system must operate with 100% of the core throughout flow rate and an exceptional efficiency. To achieve required cycle time for the demonstration case herein

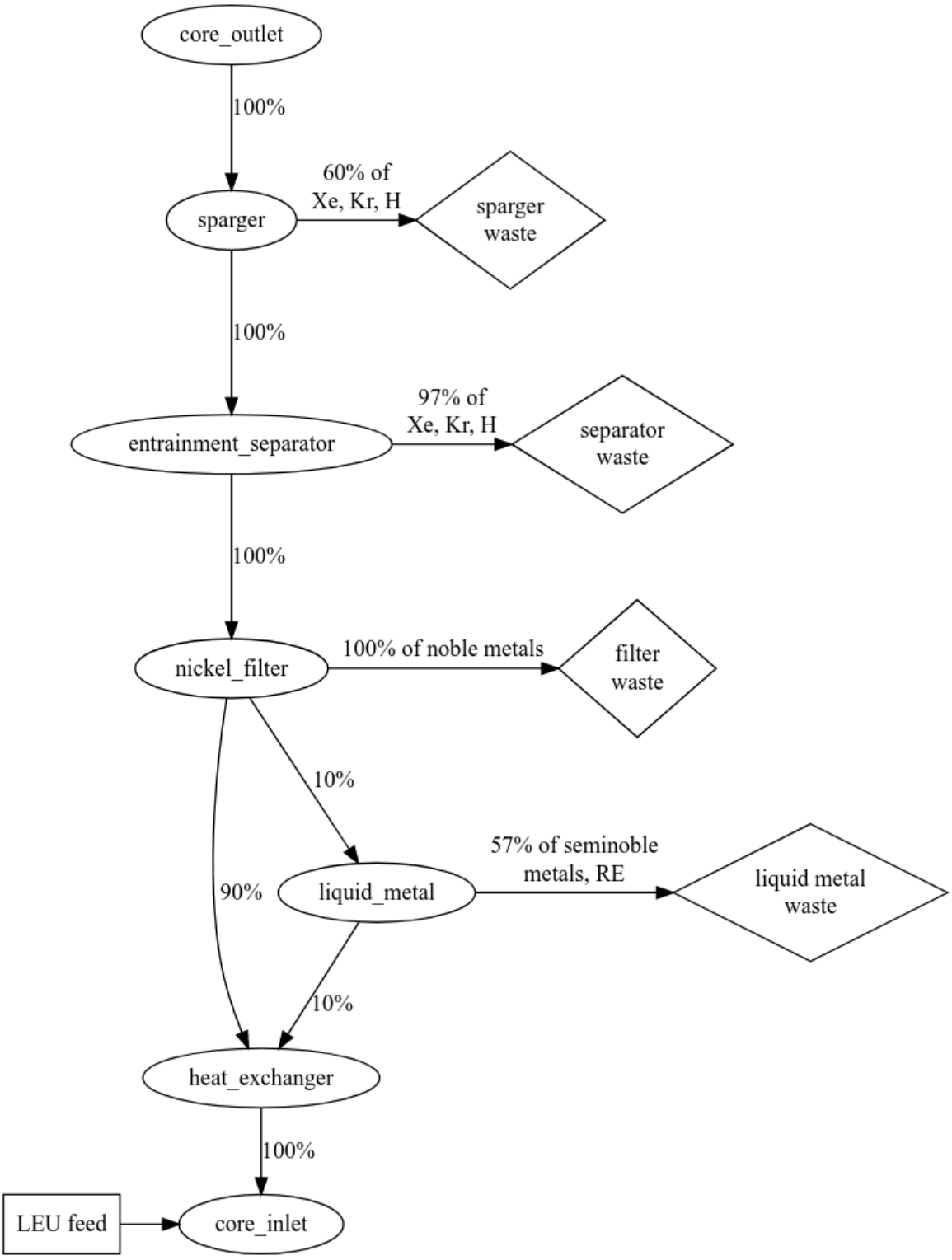

Figure 9: TAP reprocessing scheme flowchart used for SaltProc v0.2 demonstration. Arrows represent material flows; percents - fraction of total mass flow rate; ellipses - fuel reprocessing system components; diamonds - waste streams; the box shows refuel material flow.

we assumed xenon, krypton, and hydrogen extraction efficiencies for the sparger and entrainment separator are equal 60% and 97%, respectively.

The nickel filter in the TAP concept is designed to extract noble metals and volatile fluorides. Similarly to volatile gases, noble metals must be removed every 20 sec and, hence, the filter should also be able to operate in-line. The nickel filter removes a wide range of elements with various efficiencies. We calculated these efficiencies for SalProc v0.2 input from removal rates reported in table 2.

Lanthanides and other non-noble metals generally have a lower capture cross-section and absorb fewer neutrons than gases and noble metals. These elements can be removed via a liquid-metal/molten salt extraction process with relatively low removal rates (cycle time > 50 days). This is accomplished using small fuel salt flow rate (10% of the core throughout flow rate) via liquid-metal/molten salt component, where lanthanides are removed with specific extraction efficiency to match required cycle time (table 2). The remaining 90% of the flow is directed from the nickel filter to heat exchanger without performing any fuel salt treatment.

The removal rates vary among the nuclides in this reactor concept, which dictate the necessary resolution of depletion calculations. If the depletion time intervals are very short, an enormous number of depletion steps are required in order to obtain the equilibrium composition. On the other hand, if the depletion calculation time interval is too long, the impact of short-lived fission products is not captured. To compromise, a 3-day interval was selected based on Betzler *et al.* timestep refinement study [11]. For longer, lifetime-long depletion simulations, a 30-day timestep size will be applied.

6 Results

The SaltProc v0.2 online reprocessing simulation package is demonstrated for analyzing the TAP MSR neutronics and fuel cycle to find the equilibrium core composition and core depletion. The neutron population per cycle and the number of active/inactive cycles were chosen to obtain a balance between reasonable uncertainty for a transport problem (25 pcm for effective multiplication factor) and computational time. We accomplished it with a neutron population of 15,000, 400 active cycles, and 200 inactive cycles. The TAP depletion was performed on 64 Blue Waters XE6 nodes (two AMD 6276 Interlagos CPU per node, 16 floating-point Bulldozer core units per node or 32 "integer" cores per node, nominal clock speed is 2.45 GHz). The total computational time for calculating the equilibrium composition one time was approximately 9000 node-hours ($\approx$16 core-years).

6.1 Effective multiplication factor

Figures 10, 11, 12 demonstrate the effective multiplication factors obtained using SaltProc v0.2 and Serpent. We obtained the effective multiplication factors after removing fission products and adding feed material at the end of each depletion step (3 days for this work).

The k_{eff} fluctuates significantly as a result of the batch-wise nature of the online reprocessing strategy used.

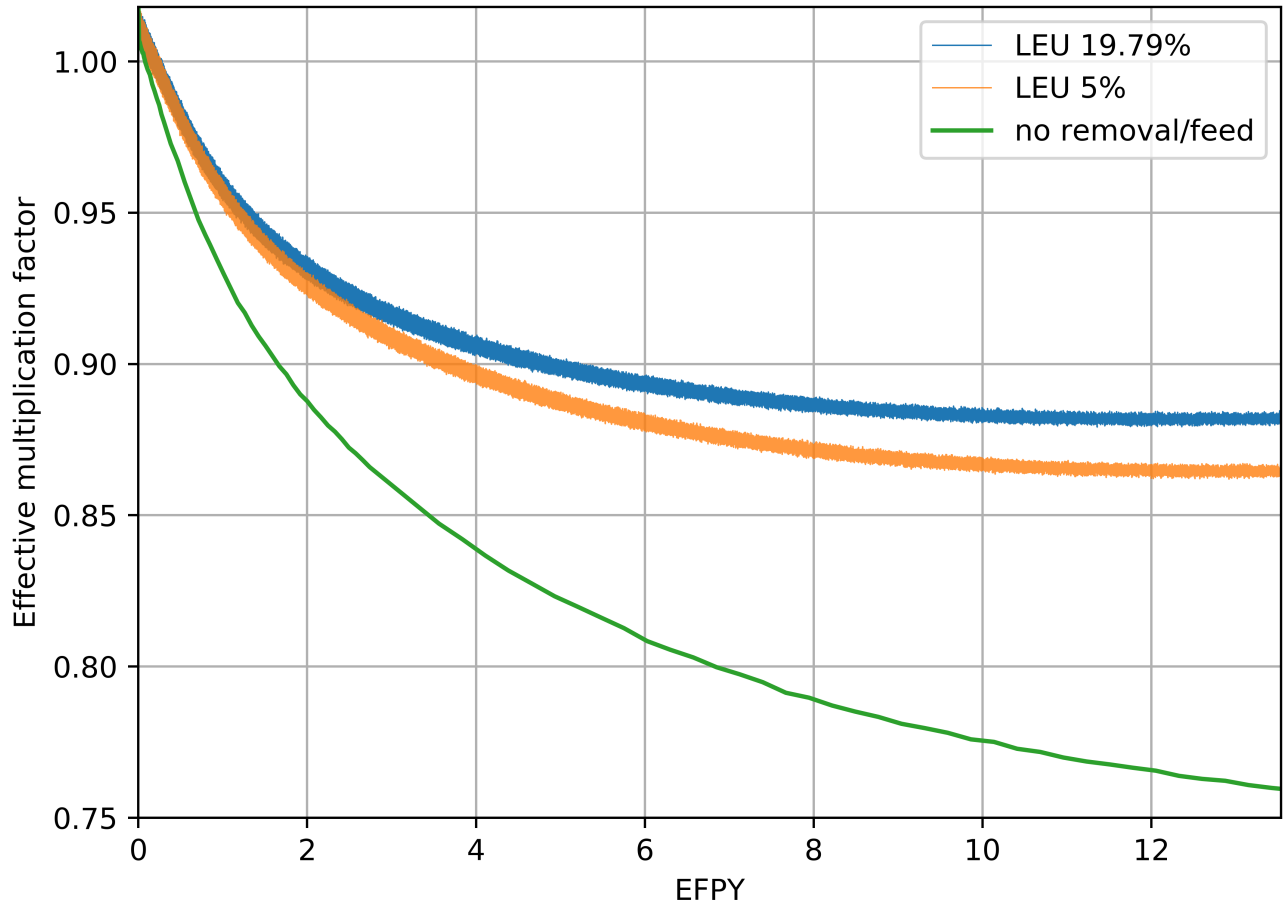

Figure 10: Effective multiplication factor dynamics for full-core TAP model for different fueling scenarios over a 13-year reactor operation. Confidence interval $\pm\sigma = 28pcm$ is shaded.

Figure 10 clearly indicates taht the reactor went subcritical too fast and further investigation is needed to overcome this issue. Possible solutions are: (1) reduce neutron leakage from the core by introducing thick graphite reflector and thermal insulation around vessel to increase effective multiplication factor at the BOL to 1.035; (2) extract poisons with faster removal rate; (3) use another fissile material for the feed (i.e., transuranic (TRU) elements from spent LWR fuel); (4) adjust SVF on-the-fly by moving moderator assemblies during operation [6] or adding moderator rods only at regular intervals during shutdown for reactor maintenance [19].

Loading the initial fuel salt composition with 5% LEU into the TAP core leads to a supercritical configuration with an excess reactivity of about 1900pcm (Figure 10). Without performing any fuel salt reprocessing, the core became subcritical after 30 days of opera-

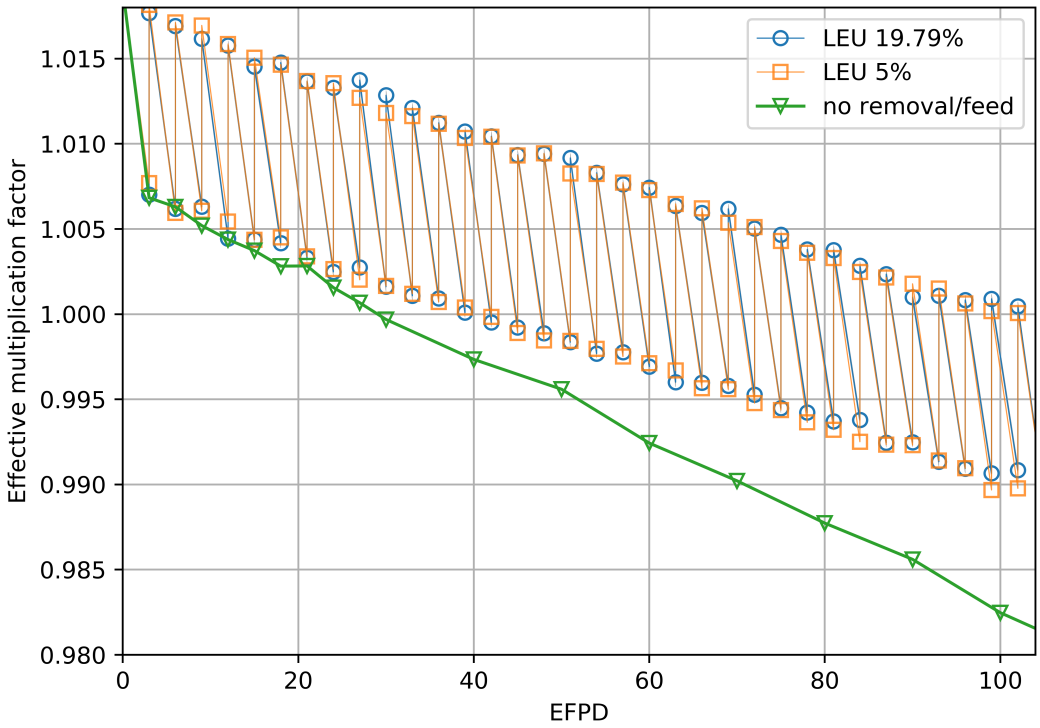

Figure 11: Zoomed effective multiplication factor for the first 104 EFPD after startup.

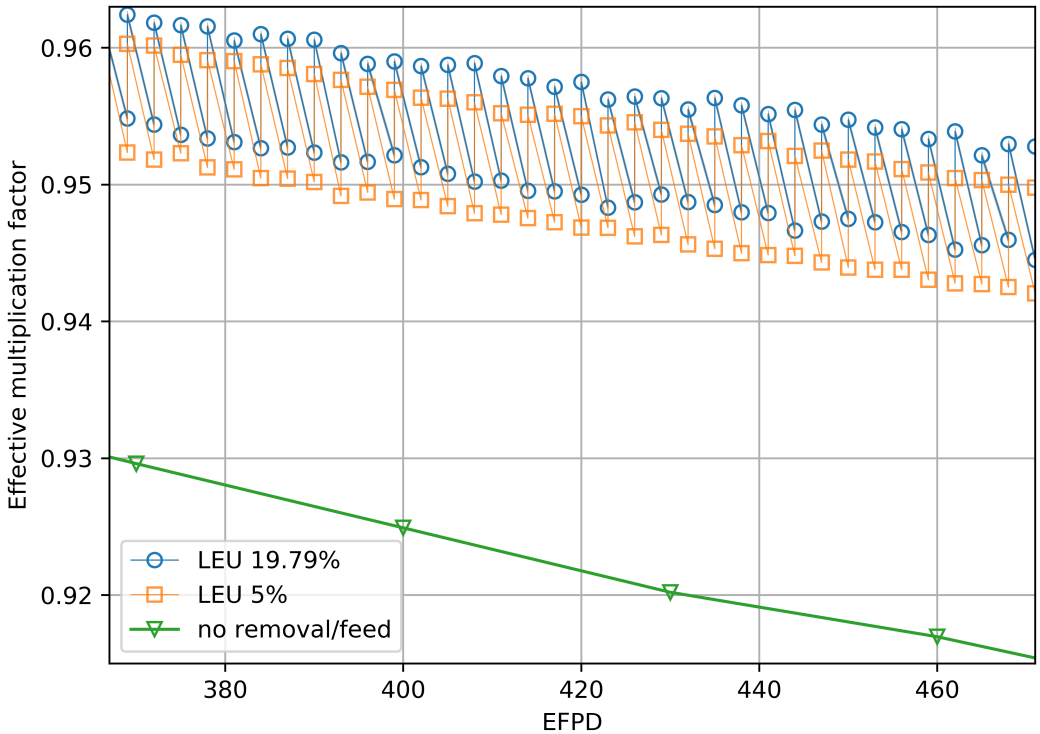

Figure 12: Zoomed effective multiplication factor for the time interval from 367 to 471 EFPD after startup.

tion (Figure 11). We obtained this result using Serpent ONLY without introducing any FP extraction and refueling. For the beginning of the TAP lifetime, uranium enrichment in the feed has a minor effect because the tiny amount of poisons was produced (<1kg/day) and, hence, a small mass of fresh salt was injected. Notably, the core went subcritical after 42 days of operation with either LEU 5% or LEU 19.79% feed.

The TAP core never reached equilibrium fuel salt composition without performing fuel salt reprocessing and refueling. For the fueling scenarios with 5% and 19.79% LEU feed, the reactor achieved the equilibrium state after 10 years of operation. Overall, the effective multiplication factor gradually decreases from the initial 1.018 to 0.88 for the 19.79% LEU feed and 0.86 for the 5% LEU feed, which indicates problems with operating this nuclear reactor design. We will try to overcome this issue by re-optimizing the TAP core and design parameters as well as adding new functionality to SaltProc v0.2.

Acting as a complement to Figure 10, the Figure 13 shows the Shannon entropy of a fission source as a function of the number of inactive cycles and clearly indicates that the Monte Carlo simulation converge with > 200 inactive cycles [20].

6.2 Neutron spectrum

Figure 14 shows the normalized neutron flux spectrum for the full-core TAP core model in the energy range from 10^{-8} to 15 MeV. The neutron energy spectrum at equilibrium is a little bit harder than at startup due to the accumulation of plutonium and other strong absorbers in the core during reactor operation. The TAP spectrum is significantly harder than in a typical LWR and is in a good agreement with ORNL report [11].

6.3 Fuel salt composition

Figure 15 shows the absolute mass of major heavy isotopes which have a strong influence on the reactor core physics. The mass of ^{236}U, ^{238}U, ^{239}Pu, ^{240}Pu, and ^{241}Pu in the fuel salt changes insignificantly after approximately 10 years of operation, which matches stabilization time for the effective multiplication factor. Hence, the quasi-equilibrium state was reached after 10 years of reactor operation. Moreover, the TAP core bred approximately the same amount of fissile ^{239}Pu ($\approx$ 2t) as was initial fissile material (^{235}U) load. A significant amount of non-fissile plutonium builds up during operation and accounts for 50% of the plutonium after 13 years of operation. Overall, the rate of breeding fissile ^{239}Pu from ^{238}U even in a relatively hard neutron spectrum is not sufficient to compensate for the negative effects of strong absorber accumulation to maintain the reactor critical.

We checked the correctness of SaltProc v0.2 by comparing the mass of the important isotopes (^{135}Xe, ^{135}I) for load-following operation to an expected mass after each depletion step (Figure 16). The expected mass of a ^{135}Xe was calculated as follows:

$$m_{after\ reprocessing} = m_{before\ reprocessing} \times \epsilon_{sparger} \times \epsilon_{separator} \tag{2}$$

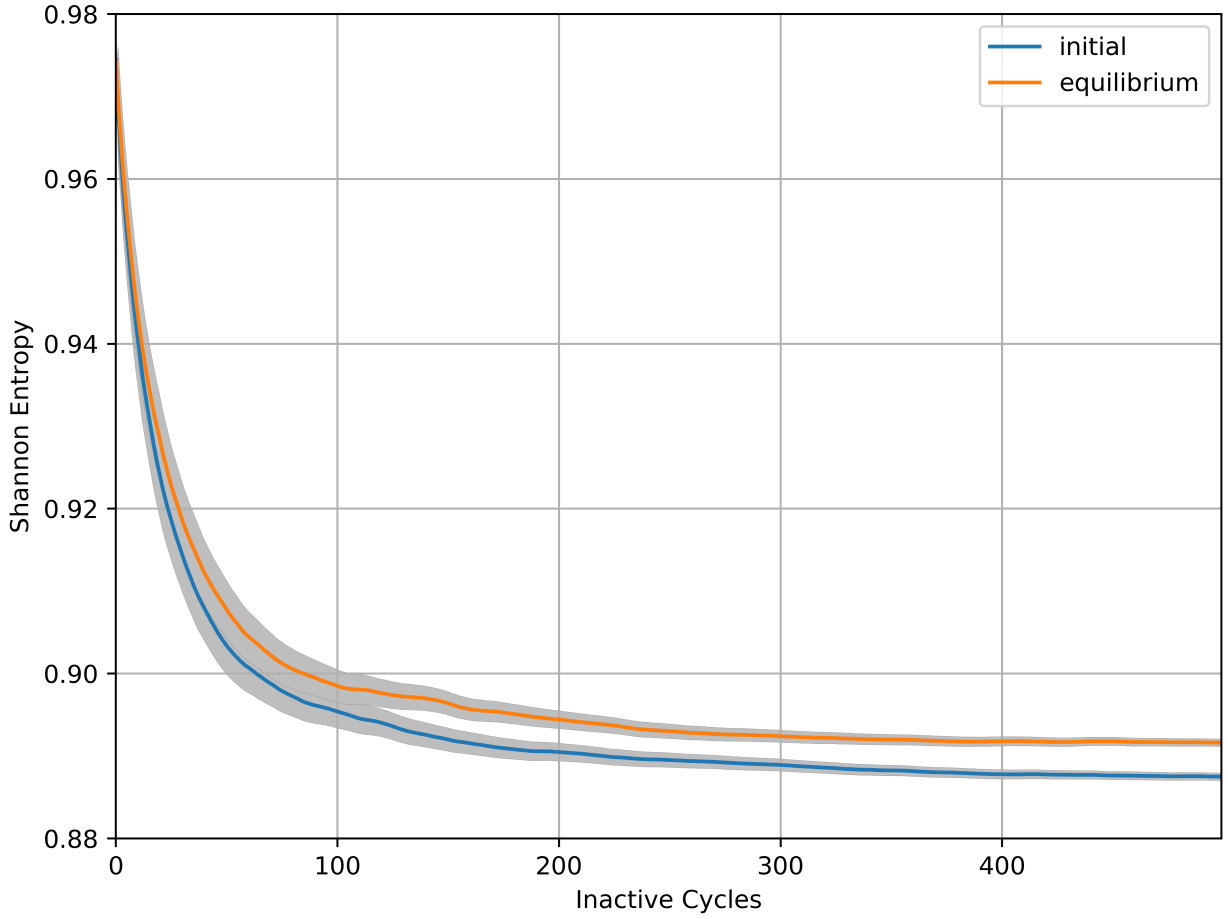

Figure 13: Shannon entropy of a fission source for initial and equilibrium fuel salt composition (19.87% LEU feed) as a function of inactive cycles number for the full core calculations with a neutron population of $M = 15,000$.

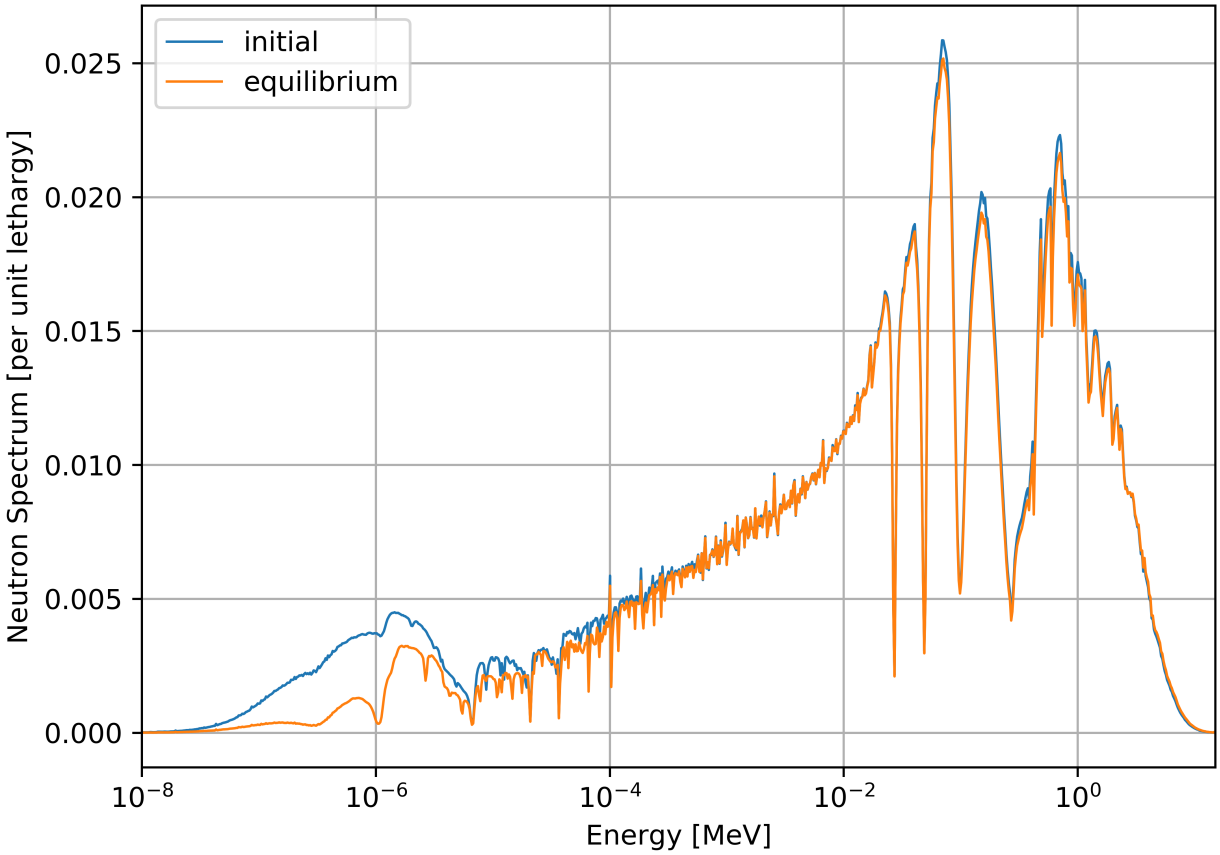

Figure 14: The neutron flux energy spectrum normalized by unit lethargy for initial and equilibrium fuel salt composition.

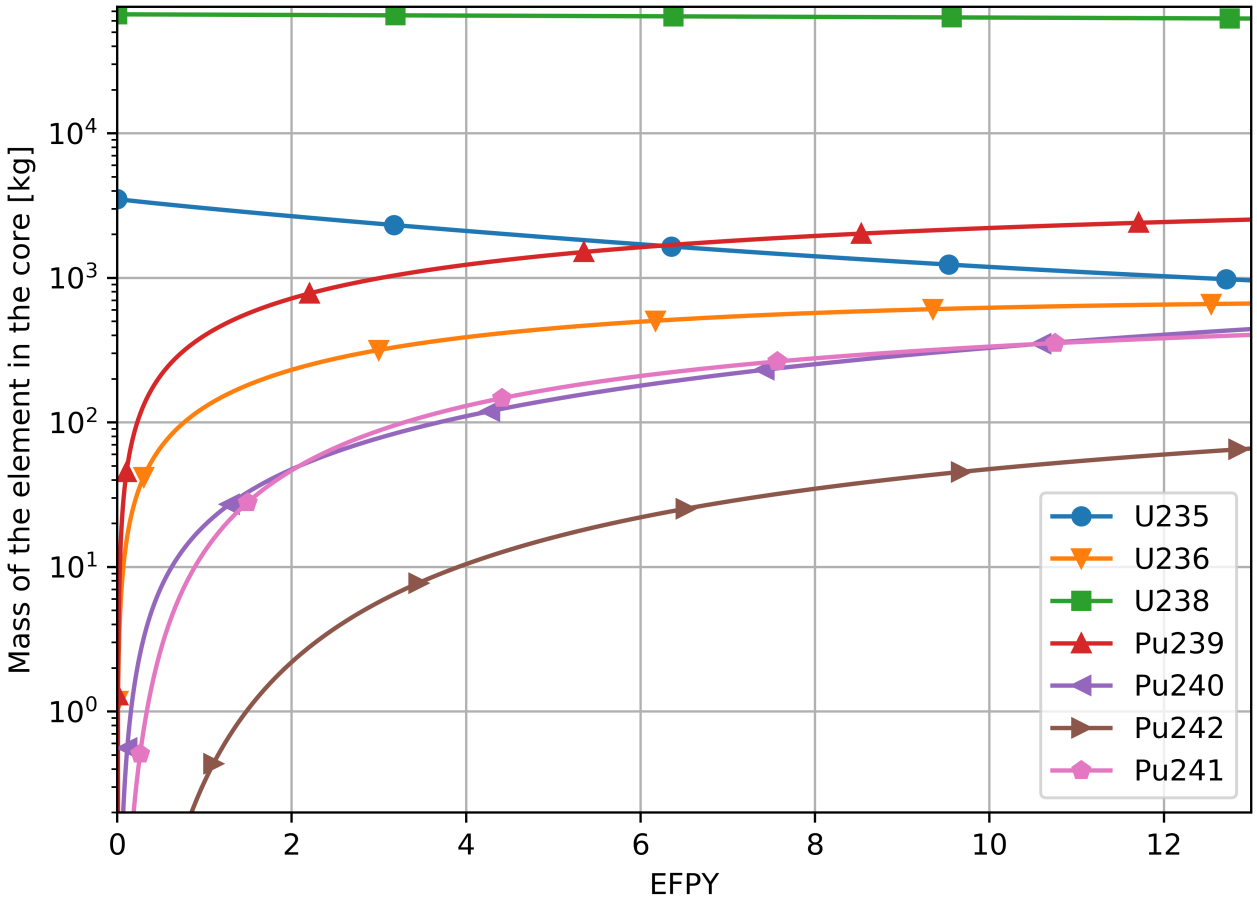

Figure 15: Mass of major nuclides during 13 years of reactor operation with 19.79% LEU feed.

where

m_{after} = the mass of the isotope after applying removals and feeds
m_{before} = the mass of the isotope right before reprocessing
$\epsilon_{sparger}$ = the sparger extraction efficiency
$\epsilon_{separator}$ = the entrainment separator extraction efficiency

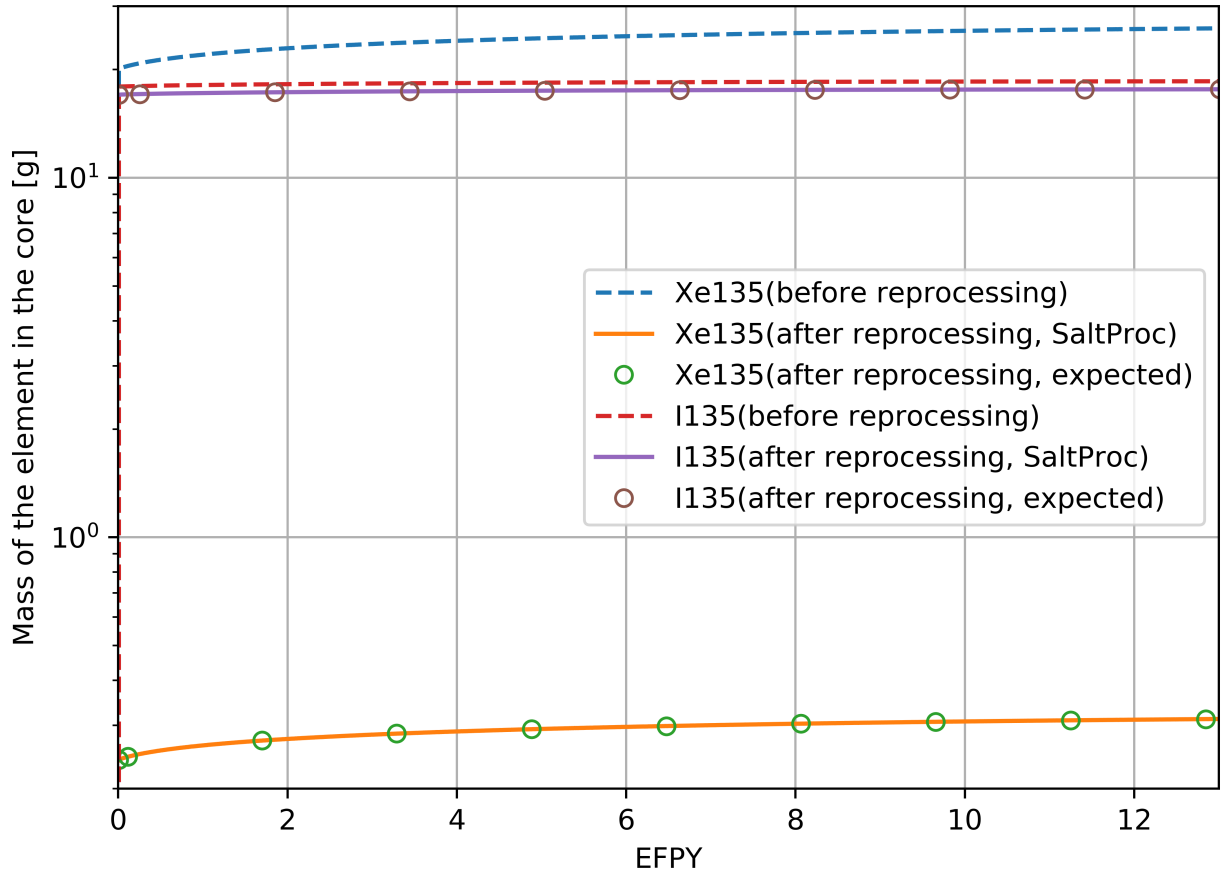

Figure 16: Mass of major neutron poison, ^{135}Xe, and its main precursor, ^{135}I, during 13 years of reactor operation before and after reprocessing.

The ^{135}I approach is similar, but the extraction efficiency of iodine in the nickel filter is only 5%. Figure 16 shows that SaltProc v0.2 extraction module correctly removes target isotopes with the specified extraction efficiency: SaltProc and expected mass match. Overall, the TAP fuel reprocessing system simulated with SaltProc v0.2 allows maintaining ^{135}Xe inventory in the core as low as 1g during operation on 100% power.

7 Future work

The TAP core should be able to maintain a critical state ($k_{eff} \geq 1.0$) for at least 30 years of operation lifetime. We will re-optimize and improve the TAP reactor model by performing the next steps:

k eigenvalue at BOL: The effective multiplication factor is too small at the BOL. The most recent ORNL paper [19] reported the initial k eigenvalue calculated for BOL to be about 1.035, much greater than our result ($1.01909 \pm 23pcm$). We will reduce fast neutron leakage by adding an appropriate reflector and thermal insulation around the vessel to reach a larger excess of reactivity at the BOL.

Dynamic moderator-to-fuel ratio: The notable feature of the TAP is the ability to adjust moderator-to-volume, or SVF, ratio during lifetime by changing the moderator rods configuration. Adding more moderator to the core thermalizes the neutron spectrum and significantly extends the core lifetime. Unfortunately, the TAP White papers and ORNL technical reports lack details about how those configurations are formed. We will create various geometries with various SVF based on the assumption, that the plant personnel is reconfiguring the moderator rods only at regular intervals (i.e., 18 months) during the shutdown for reactor maintenance. That is, we assume that the reactor maintaining the long-term reactivity by periodically replacing stationary zirconium hydride rod assemblies with those containing more rods (e.g., replacement of a four-rod assembly with a nine-rod assembly) [19]. Additionally, we will add in a SaltProc v0.2 capability to switch from one geometry file to another with a user-defined time interval.

Reprocessing scheme: Extraction efficiencies and refueling strategies of the TAP fuel reprocessing and refueling plant will be revised to ensure that all possible strong poisons are removed at an appropriate rate.

References

[1] A. Rykhlevskii, J. W. Bae, and K. D. Huff, "Modeling and simulation of online reprocessing in the thorium-fueled molten salt breeder reactor," *Annals of Nuclear Energy*, vol. 128, pp. 366–379, June 2019.

[2] A. Rykhlevskii, "Advanced online fuel reprocessing simulation for Thorium-fueled Molten Salt Breeder Reactor," Master's thesis, University of Illinois at Urbana-Champaign, Apr. 2018.

[3] A. Rykhlevskii, J. W. Bae, and K. Huff, "arfc/saltproc: Code for online reprocessing simulation of molten salt reactor with external depletion solver SERPENT," *Zenodo*, Mar. 2018.

[4] J. Leppanen, M. Pusa, T. Viitanen, V. Valtavirta, and T. Kaltiaisenaho, "The Serpent Monte Carlo code: Status, development and applications in 2013," *Annals of Nuclear Energy*, vol. 82, pp. 142–150, Aug. 2014.

[5] T. H. Group, "Hierarchical data format, version 5," 1997.

[6] Transatomic Power Corporation, "Technical White Paper," White Paper 2.1, Transatomic Power Corporation, Cambridge, MA, United States, Nov. 2016.

[7] Transatomic Power Corporation, "Neutronics Overview," White Paper 1.1, Transatomic Power Corporation, Cambridge, MA, United States, Nov. 2016.

[8] B. R. Betzler, J. J. Powers, A. Worrall, L. Dewan, S. Robertson, and M. Massie, "Two-Dimensional Neutronic and Fuel Cycle Analysis of the Transatomic Power Molten Salt Reactor," Tech. Rep. ORNL/TM-2016/742, Oak Ridge National Lab.(ORNL), 2017.

[9] P. N. Haubenreich and J. R. Engel, "Experience with the Molten-Salt Reactor Experiment," *Nuclear Technology*, vol. 8, pp. 118–136, Feb. 1970.

[10] R. C. Robertson, "Conceptual Design Study of a Single-Fluid Molten-Salt Breeder Reactor.," Tech. Rep. ORNL–4541, ORNL, Jan. 1971.

[11] B. R. Betzler, S. Robertson, T. E. E. Davidson, J. J. Powers, and A. Worrall, "Assessment of the Neutronic and Fuel Cycle Performance of the Transatomic Power Molten Salt Reactor Design," Tech. Rep. ORNL/TM-2017/475, ORNL, 2017.

[12] T. Power, "Transatomic Reactor Documentation," Mar. 2019. original-date: 2018-11-08T00:08:31Z.

[13] B. R. Betzler, J. J. Powers, N. R. Brown, and B. T. Rearden, "Implementation of Molten Salt Reactor Tools in SCALE," in *Proc. M&C 2017 - International Conference on Mathematics & Computational Methods Applied to Nuclear Science and Engineering*, (Jeju, Korea), Apr. 2017.

[14] A. Scopatz, P. K. Romano, P. P. H. Wilson, and K. D. Huff, "PyNE: Python for Nuclear Engineering," in *Transactions of the American Nuclear Society*, vol. 107, (San Diego, CA, USA), American Nuclear Society, Nov. 2012.

[15] H. Krekel, B. Oliveira, R. Pfannschmidt, F. Bruynooghe, B. Laugher, and F. Bruhin, "pytest: Python testing tool," 2004.

[16] G. Brandl, "Sphinx: Python Documentation Generator," 2009.

[17] J. Leppanen, "Serpent – a Continuous-energy Monte Carlo Reactor Physics Burnup Calculation Code," *VTT Technical Research Centre of Finland, Espoo, Finland*, 2013.

[18] OECD/NEA, "The JEFF-3.1.2 Nuclear Data Library," Tech. Rep. JEFF Report 24, OECD/NEA Data Bank, OECD/NEA, 2014.

[19] B. R. Betzler, S. Robertson, E. E. Davidson, J. J. Powers, A. Worrall, L. Dewan, and M. Massie, "Fuel cycle and neutronic performance of a spectral shift molten salt reactor design," *Annals of Nuclear Energy*, vol. 119, pp. 396–410, Sept. 2018.

[20] F. B. Brown, ""K-effective of the World" and Other Concerns for Monte Carlo Eigenvalue Calculations," *Progress in Nuclear Science and Technology*, vol. 2, pp. 738–742, Oct. 2011.