Enabling Load Following Capability in the Transatomic Power MSR

Modeling-Enhanced Innovations Trailblazing Nuclear Energy Reinvigoration (MEITNER)
DE-FOA-0001798
Task 2 Milestone Report

Author
Andrei RYKHLEVSKII

Principal Investigator
Kathryn D. HUFF

UIUC-ARFC-2019-012

July 22, 2019

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

The authors gratefully acknowledge the support of the MEITNER program, sponsored by Advanced Research Projects Agency - Energy - U.S. Department of Energy.

1 Introduction

We initiated the Fuel Cycle Simulation task (Task 2) of the project in August 2018 to more realistically model an online reprocessing system of the Transatomic Power (TAP) Molten Salt Reactor (MSR). A Python toolkit, SaltProc v1 [1–3], was developed to represent simplified online fuel salt processing of Molten Salt Breeder Reactor (MSBR). More recently, advanced SaltProc version (SaltProc v2.0+) was developed to simulate a complex salt reprocessing system of the TAP incorporating user-parametrized components in the fuel salt processing design. This report summarizes the progress we have made towards milestone **M2.1: Demonstration SaltProc**, the challenges we currently face, and the plans towards ultimate Task 2 objectives.

2 Milestone objectives

ARPA-E Award No. DE-AR0000983 with the Board of Trustees of the University of Illinois Attachment 3 (Technical Milestones and Deliverables) section D (Description of technical tasks, milestones, and deliverables) formulated M2.1 goal 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."
Herein we demonstrated following capabilities of SaltProc v2.0+:

1. Read a user-defined Serpent 2 input template file with the model geometry, material composition, total heating power, and boundary conditions.

2. Read a user-defined *.json* input file with parameters and structure of fuel salt reprocessing system.

3. Run Serpent 2 in parallel mode to perform depletion calculation.

4. Read Serpent 2 the depleted fuel composition file and store it in the HDF5 database [4].

5. Remove poisons from the fuel isotopic composition by passing information throughout user-parametrized components of the fuel salt processing system. For demonstration proposes, SaltProc v2.0+ used user-defined constant separation efficiencies but can handle variable efficiencies based on work in Task 1.

6. Make-up fuel salt mass loss in the primary loop due to poisons extraction by adding fresh salt with a user-defined isotopic composition (e.g., low-enriched uranium (LEU) 5% and 19.79%, for this work).

7. Store fuel salt composition after performing 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 [5]. This concept uses configurable zirconium hydride rods as the moderator while most MSR designs usually propose high-density reactor graphite. Zirconium hydride offers a much higher neutron moderating density than graphite; much less zirconium hydride volume is needed to achieve a thermal energy spectrum similar to one obtained with graphite moderator. Moreover, zirconium hydride has a much longer lifespan in extreme operational conditions (high temperature, large neutron flux, chemically aggressive salt) than reactor graphite. Finally, zirconium hydride is nonporous material and hold up much fewer neutron poisons (e.g., xenon, krypton) comparing with high-density reactor graphite [5–7].

3.1 TAP design description

The TAP design (figure 1) is very similar to original Molten Salt Reactor Experiment (MSRE) design developed by Oak Ridge National Laboratory (ORNL) [8] but has two major innovations: the fuel salt composition and the moderator. The MSRE's LiF-BeF_2-ZrF_4-UF_4 salt has been substituted with LiF-UF_4 salt which allows for an increase in the uranium concentration within the fuel salt 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) [7]. The graphite has a very high thermal scattering cross section which makes it a perfect moderator but has a few major drawbacks: (1) the low lethargy gain per collision requires a large volume of moderator to be present to reach criticality, which leads to a larger core and obstructs the core power density; (2) even special reactor-grade graphite has relatively high porosity, consequently, it holds gaseous Fission Products (FPs) (e.g., tritium, xenon) in pores; (3) the reactor graphite lifespan in a commercial reactor is about 10 years [9]. To resolve these issues, the TAP concept uses another moderator, namely, 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 in the current commercially available 5% LEU fuel cycle.

The TAP MSR primary loop contains the reactor core volume (including the zirconium hydride moderator rods with silicone carbide cladding), pumps, and primary heat exchanger. Pumps circulate the LiF-(Act)F_4 fuel salt through the primary loop. The pumps,

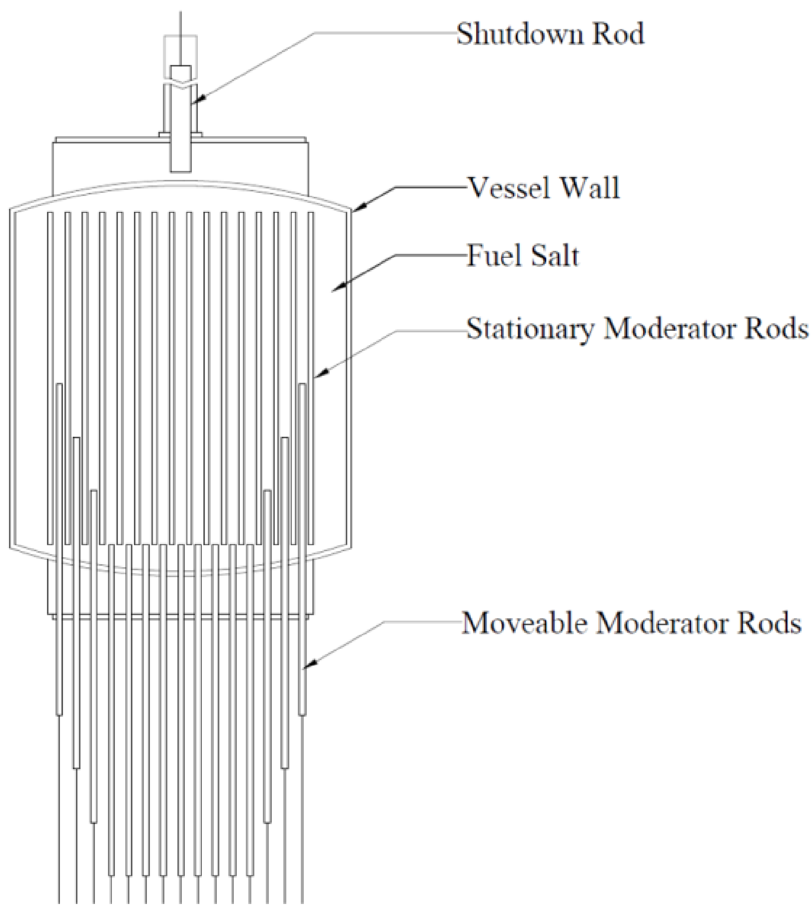

Figure 1: The TAP MSR schematic view showing movable moderator rod bundles and shutdown rod (figure reproduced from Transatomic Power White Paper [5]).

vessels, tanks, and piping are made of a nickel-based alloy (similar to Hastelloy-N[1]), which is highly resistant to corrosion in various molten salt environments. Inside the reactor vessel, in close proximity 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 which are taken from technical white paper [5] and a neutronics overview [6] as well as ORNL analysis of the TAP design [7,10].

Table 1: Summary of principal data for the TAP MSR (reproduced from [5,10]).

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

3.2 TAP core design

In the TAP core (figure 2), fuel salt flows around moderator assemblies consisting of lattices of zirconium hydride rods clad in a corrosion-resistant silicone carbide (figure 1). The TAP reactor pressure vessel is a cylinder with an inner radius 150 cm, height 350 cm, and wall thickness 5 cm made of a nickel-based alloy. 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 (from Beginning of Life (BOL) to End of Life (EOL), respectively) to maximize fuel burnup. In practice, SVF can be varied by inserting fixed-sized moderator rods via the bottom of the reactor vessel (for safety considerations), 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 a criticality. Unmoderated salt is flowing in the annulus between the core and the vessel wall provides for a potential reduction in fast neutron flux at the vessel structural material [6].

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

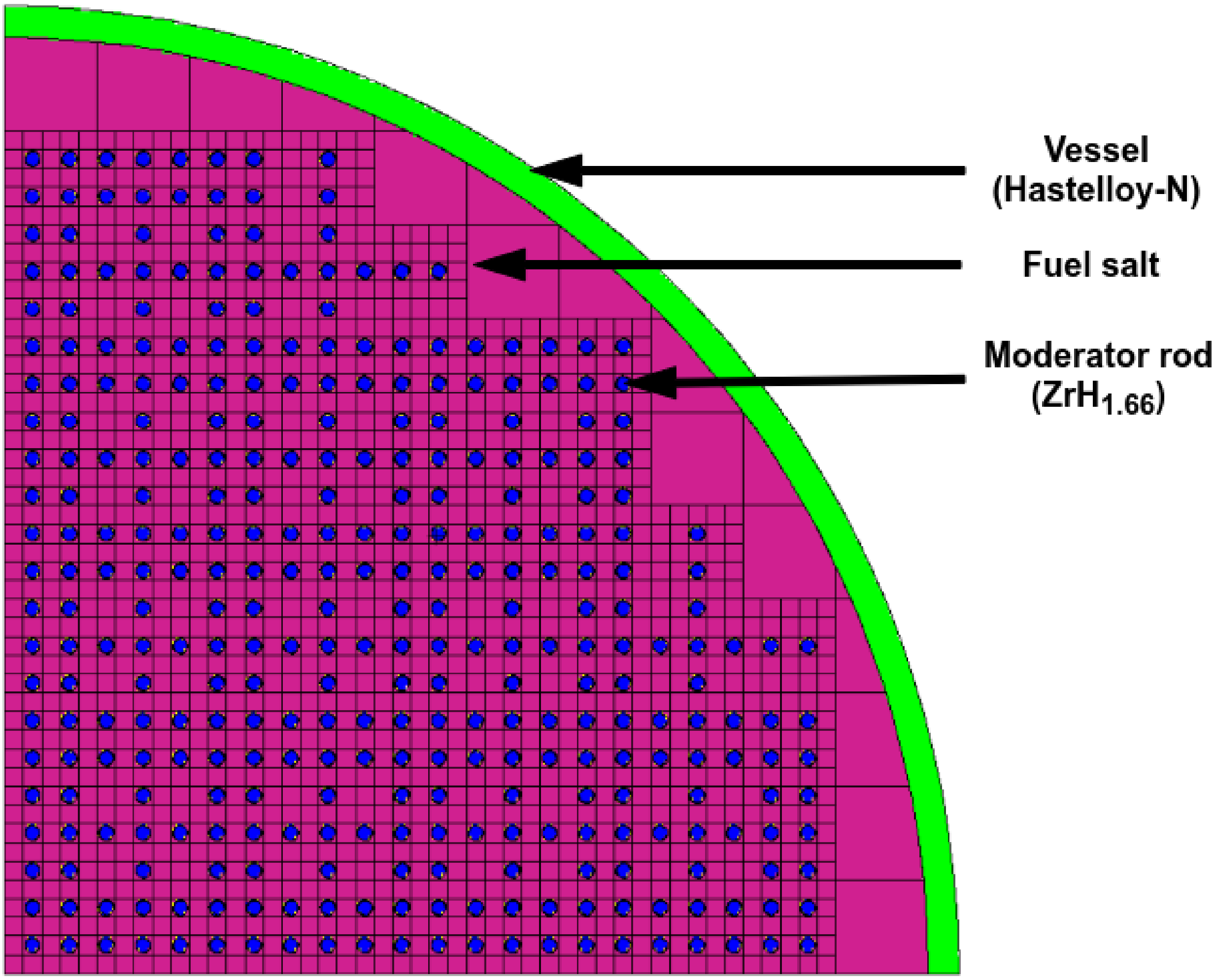

Figure 2: The TAP MSR schematic core view showing moderator rods (figure reproduced from ORNL/TM-2017/475 [10]).

3.3 TAP reprocessing system structure and simulation approach

The TAP nuclear island contains 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. As these byproducts are gradually removed, a small quantity of fresh fuel salt is regularly added to the primary loop. This process conserves a constant fuel salt mass and keeps the reactor critical. In contrast with the MSBR reprocessing system, the TAP does not need a protactinium separation and isolation system because it operates in a uranium-based single-stage fuel cycle. The authors of the TAP concept suggested three distinct fission product removal methods [6]:

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

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

Liquid Metal Extraction: Lanthanides and other non-noble metals stay dissolved in the fuel salt. They 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, lanthanides removal is accomplished via a liquid-metal/molten salt extraction process similar to that developed for MSBR by ORNL [9]. The process converts the dissolved lanthanides into a well-understood oxide waste form, similar to that of Light Water Reactor (LWR) spent nuclear fuel (SNF). This oxide waste comes out of the TAP reprocessing plant in ceramic granules and 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, an off-gas system is also based on a simple process of helium sparging through fuel salt with consequent gas bubbles removed before returning the fuel salt back to the core. Nevertheless, one important difference must be noted: the MSBR gas separation system suggested helium injection and subsequent transport of the voids throughout the primary loop, including the core for at least 10 full loops [9]. It is a significant concern to safe, stable operation because the increase of void fraction in the fuel salt when it enters back to the core would cause unpredictable reactivity change. This drawback can be overcome by using an effective gas separator for stripping helium/xenon bubbles before returning the salt back to a primary loop (Figure 3, blue block).

Noble and semi-noble metal solid fission products tend to plate out onto metal surfaces including piping, heat exchanger tubes, reactor vessel inner surface, etc. Previous research by ORNL [9] reported 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 suggested employing 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 large metal (nickel) surface area. The fuel salt flowing throughout the filter and 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 fuel salt and estimated efficiency of this process.

The TAP project reported detailed list of elements for removal and removal efficiencies (Table 2). We used data from TAP neutronics whitepaper [6] for SaltProc v2.0+ demonstration case without any modifications.

We simulated TAP MSR depletion in SaltProc v2.0+ using reprocessing cycle times

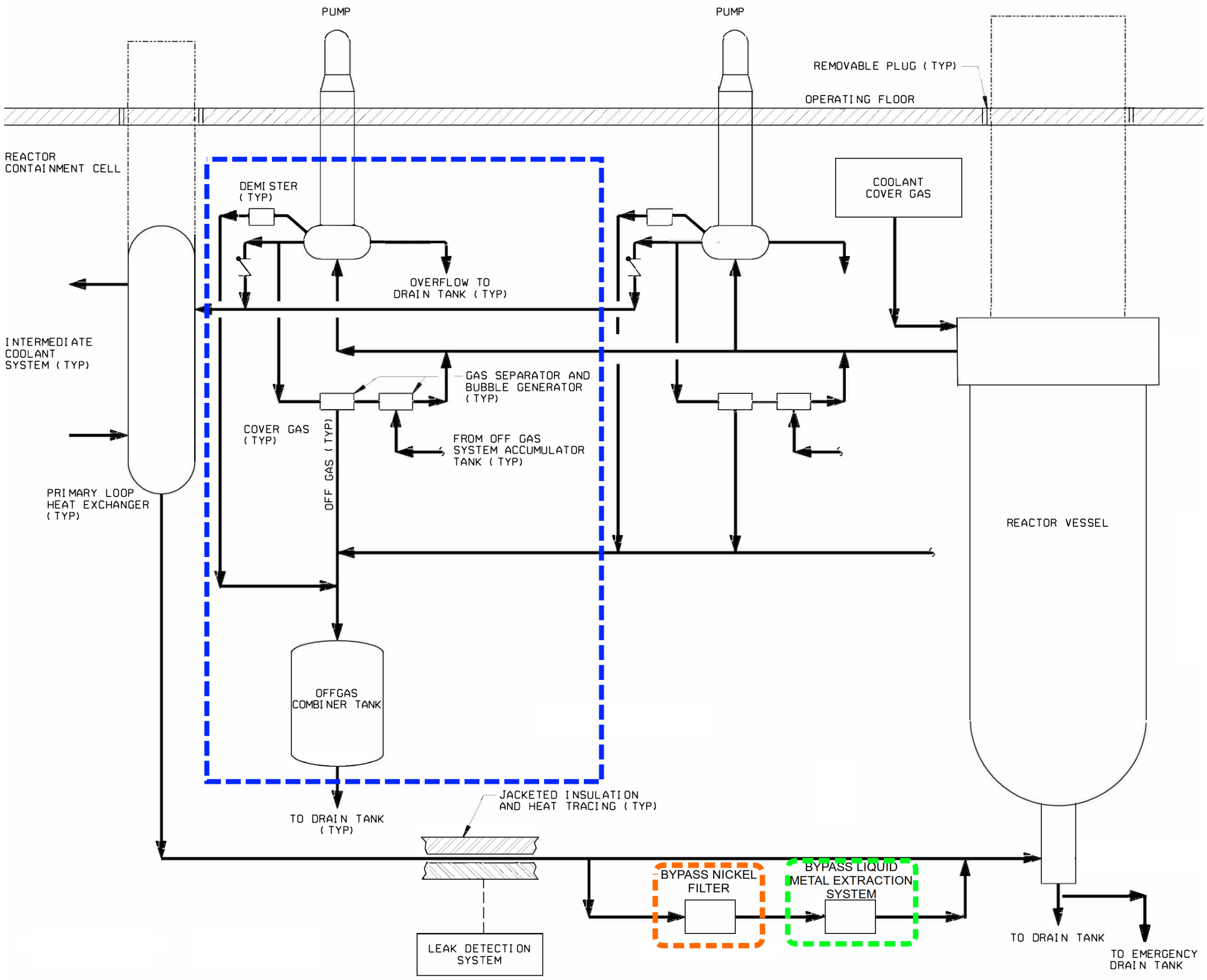

Figure 3: Simplified TAP primary loop design including off-gas system (blue), nickel filter (orange) and liquid metal extraction system (green) (reproduced from [11]).

Table 2: The effective cycle times for fission products removal from the TAP MSR (reproduced from [12] and [6]).

Processing group	Nuclides	Removal Rate (s^{-1})	Cycle time (at full power)
Elements removed in MSBR concept and adopted for the TAP [9]			
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 [6, 12]			
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

from this section (Table 2), an online reprocessing system design details, and 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 for calculating MSR fuel composition evolution taking into account an online reprocessing system was developed in 2018 as a part of M.S. thesis [2, 3]. The tool was designed to expand Serpent 2 depletion capabilities for modeling liquid-fueled MSR with online fuel reprocessing system. SaltProc v1 uses HDF5 [4] to store data and uses the PyNE Nuclear Engineering Toolkit [13] for Serpent 2 output file parsing and nuclide naming. SaltProc v1 is an open-source Python package that uses a batch-wise approach to simulate continuous feeds and removals in MSRs.

SaltProc v1 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 appreciable time out of the core, in the primary loop pipes and the heat exchanger. This approach works well for fast-removing elements (gases, noble metals) which should be removed each depletion step. Unfortunately, for the elements with longer cycle times (i.e. rare earths should be removed every 50 days) this simplified approach leads to oscillatory behavior of all major parameters [1].

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

We completely re-factored SaltProc v1 using Object-Orienting Programming (OOP) to create a comprehensive generic 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 v2.0+ architecture

The SaltProc v2.0+ Python toolkit coupled directly with Serpent 2 input and output files, to allow the reprocessing system couples to depletion calculation. Existing PyNE interfaces are employed for Serpent output parsing as well as newly developed interfaces for input and output handling. Python 3 OOP standard features is used to create a flexible, user-friendly tool with great potential for further improvement and collaboration. Figure 4 shows the SaltProc v2.0+ class structure which includes 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.

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 allows to easily convert the units of isotopic composition vector (e.g., from atomic density provided by Serpent to a mass fraction or absolute mass in desired units), decay material (i.e. model the MSBR protactinium decay tank), calculate decay heat, activity, and dose. The main idea of the *MaterialFlow* object is to pass detailed information about the salt starting at the MSR vessel outlet throughout reprocessing components (*Processes*), which modify the *MaterialFlow* object before depleting the material in the next Serpent burnup step.

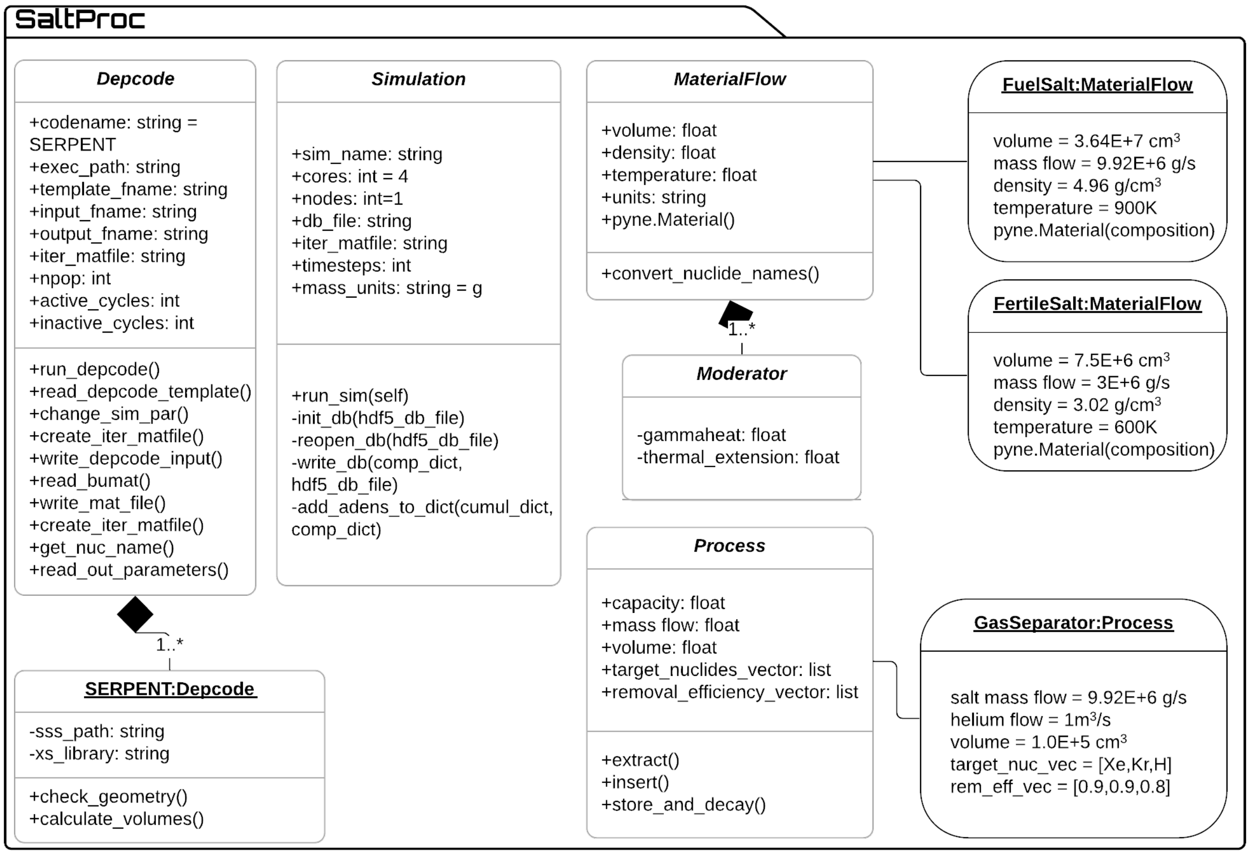

Figure 4: SaltProc v2.0+ python package class diagram in UML notation and examples of object instances.

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, and contribute back to the code package using GitHub (https://github.com/arfc/saltproc).

4.2 SaltProc v2.0+ flowchart

Figure 5 illustrates the online reprocessing simulation algorithm coupling SaltProc v2.0+ and Serpent. To perform a depletion step, SaltProc v2.0+ reads a user-defined Serpent template file. This file contains input parameters such as geometry, material, isotopic composition, neutron population, criticality cycles, total heating power, and boundary conditions. SaltProc v2.0+ fills in the template file and runs Serpent single-step depletion. After the depletion calculation, SaltProc v2.0+ reads the depleted fuel composition file into *MaterialFlow* object (*core_outlet* in figure 5). This 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 target elements with specified extraction efficiency. Afterward, the removed material mass is compensated by fresh fuel salt to maintain the salt inventory in a primary loop. Finally, resulting isotopic composition after reprocessing is stored in HDF5 database and dumped in a new composition file for the next Serpent depletion run. SaltProc v2.0+ also stores in database isotopic composition before reprocessing and waste stream from each fuel processing component.

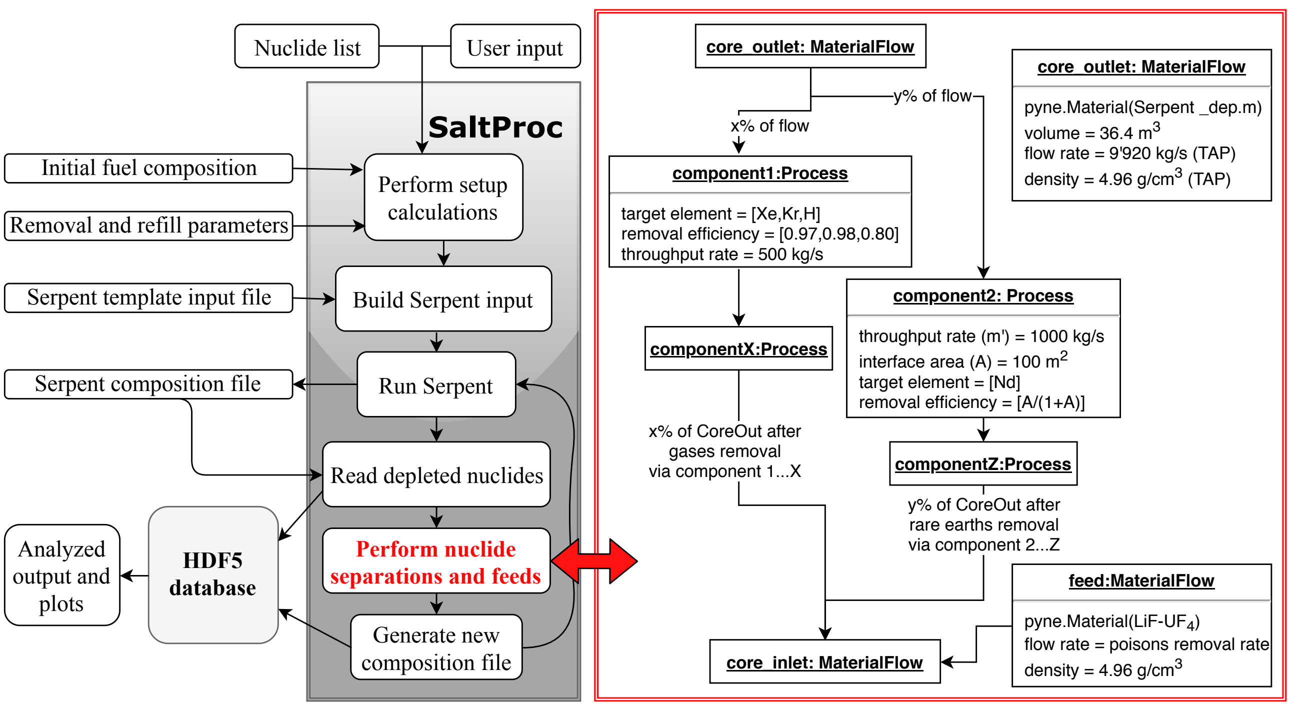

Figure 5: SaltProc v2.0+ python package flow chart.

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 via left branch and 10% through a 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 via a cascade of *Processes* to separate selected chemical elements with specific efficiency. Finally, the left-hand-side branch *MaterialFlow* object is merged with the right-hand-side and similarly to the previous case, fresh fuel salt feed compensate the loss of mass in separation facilities and keep fuel salt mass in a primary loop constant.

The class diagram (Figure 4) allows to model the operation of a complex, multi-zone, multi-fluid MSR and is sufficiently general to represent myriad reactor systems. The refactored version of SaltProc stores and edit the isotopic composition of the fuel stream, which makes 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 v2.0+ is an open-source tool (but a user needs Serpent installed to use SaltProc v2.0+), available on Github. It leverage unit and continuous tests crucial for sustainable development [14]. It will also have documentation generated through Sphinx, a documentation generator, for ease of use [15]. In summary, the development approach of SaltProc v2.0+ 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 v2.0+ modeling and simulation tool is demonstrated for TAP MSR with static core geometry, LEU 5% startup composition [6] and following fueling scenarios: (1) no FPs removal and feed (Serpent only); (2) a 5% LEU online feed; (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 [16,17].

5.1 Serpent 2 full-core model

Advanced geometry surfaces and transformation capabilities of Serpent [16] are employed to represent TAP core. Figure 6 shows the XY section of whole-core configuration at the expected reactor operational level when all control rods are fully withdrawn. Figures 7 and 8 show a longitudinal section of the reactor. This model contains the moderator rods with silicon carbide cladding, pressure vessel, and inlet and outlet plena (Table 3). 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 parameter similar to wide-used moderator-to-fuel ratio and can be defined as:

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

where

$$V_F = \text{the fuel volume}$$
$$V_M = \text{the moderator volume}$$
$$V_M/V_F = \text{the moderator-to-fuel salt ratio}$$

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

To represent 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 [10]; (3) air inside guide tubes and control rods. Control rods 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 easily change the control rods position during simulation. Herein we assumed that all control rods are fully withdrawn from the core (figure 8) but for future investigation control rods position may vary. In this report, all figures of the core were generated using the built-in Serpent plotter.

Table 3: Geometric parameters for the full-core 3D model of TAP (reproduced from Betzler *et al.* [10]).

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×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

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

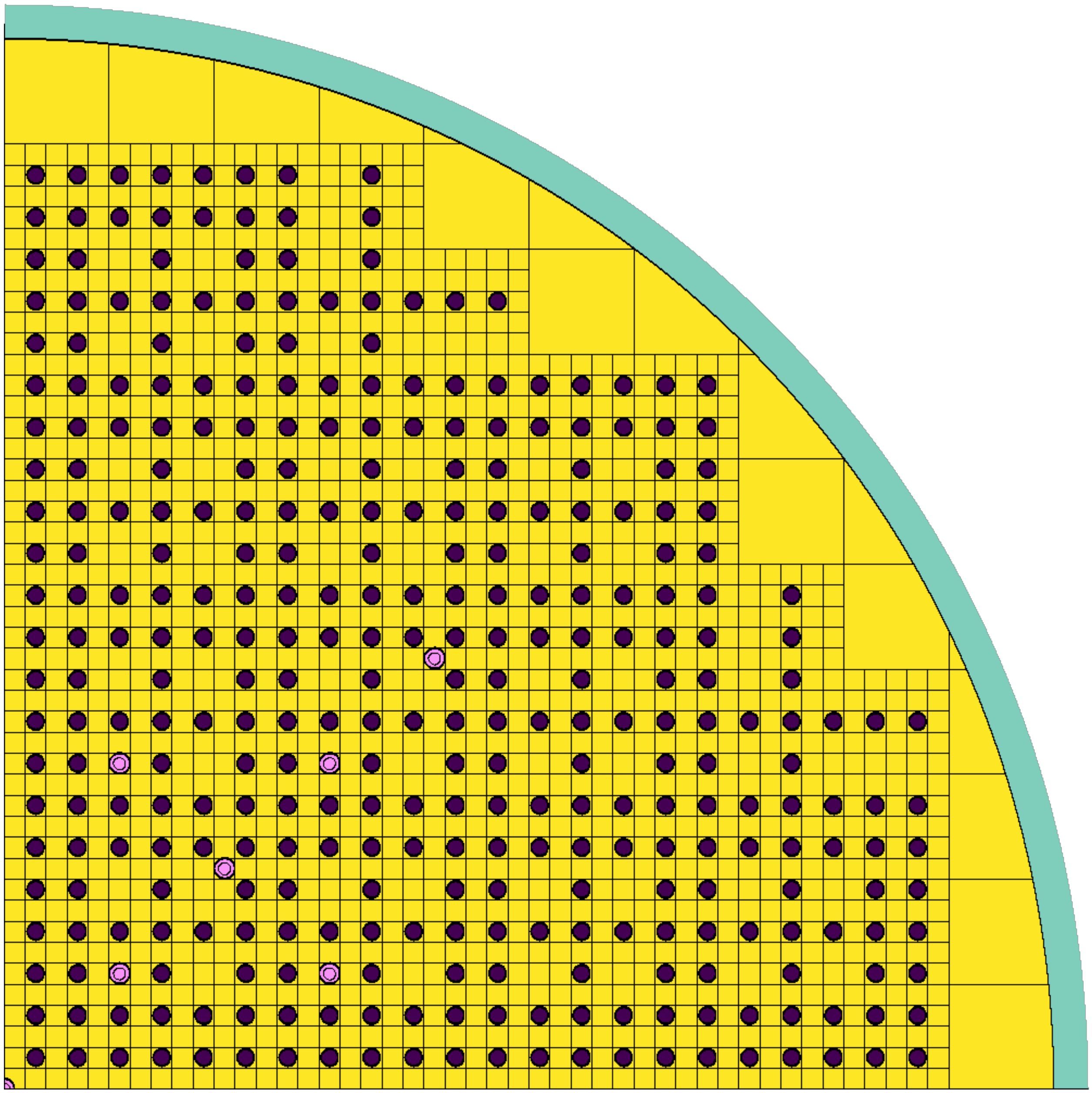

Figure 6: An XY section of the TAP model at horizontal midplane with fully withdrawn control rods at BOL (SVF= 0.907268). The violet color represents zirconium hydride, and the yellow represents fuel salt. The blue color shows Hastelloy-N, a material used for the vessel wall, and the white color is the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

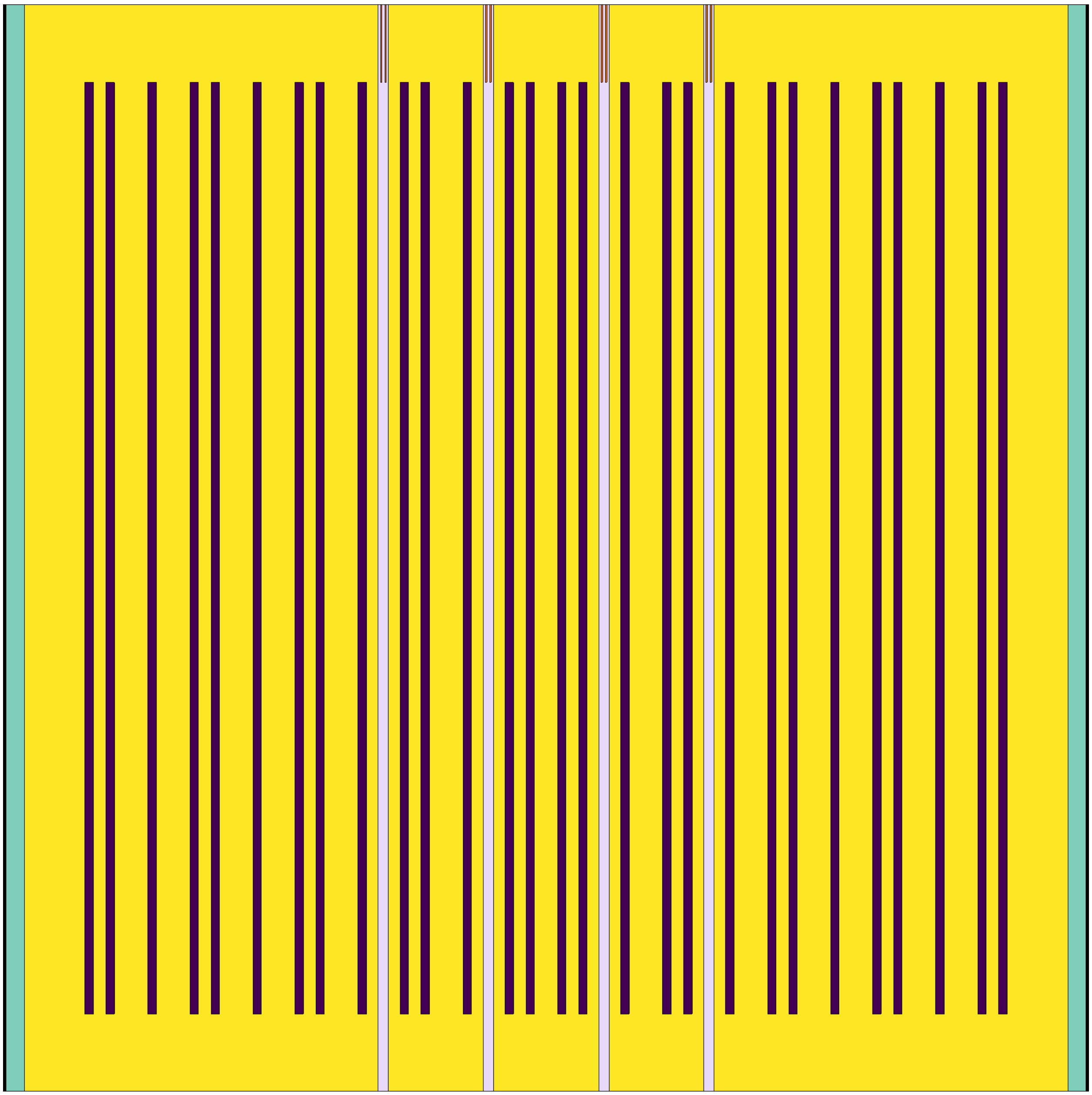

Figure 7: An XZ section of the TAP model.

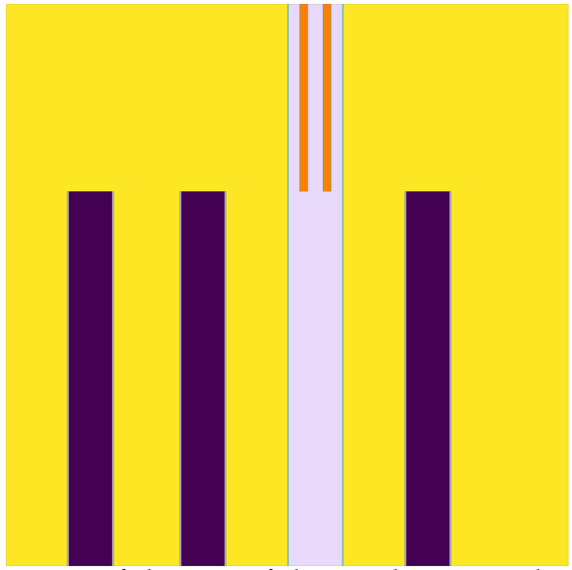

Figure 8: Zoomed XZ section of the top of the moderator rods and guide tubes for TAP model. The orange color shows 70–30% Gd_2O_3–Al_2O_3 ceramic absorbers used for control rods.

5.2 Simulated fuel reprocessing system

We thoroughly analyzed the original TAP reprocessing system design (figure 3) and neutron poisons removal rates (table 2) to determine suitable reprocessing scheme for SaltProc v2.0+ demonstration (figure 9).

The gas removal components (the sparger and entrainment separator) are located in-line because estimated full loop time for the fuel salt is about 18 sec and 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 exceptional efficiency. For the demonstration case herein to achieve required cycle time 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 v2.0+ 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 rest 90% of the flow is directed from the nickel filter to heat exchanger without performing

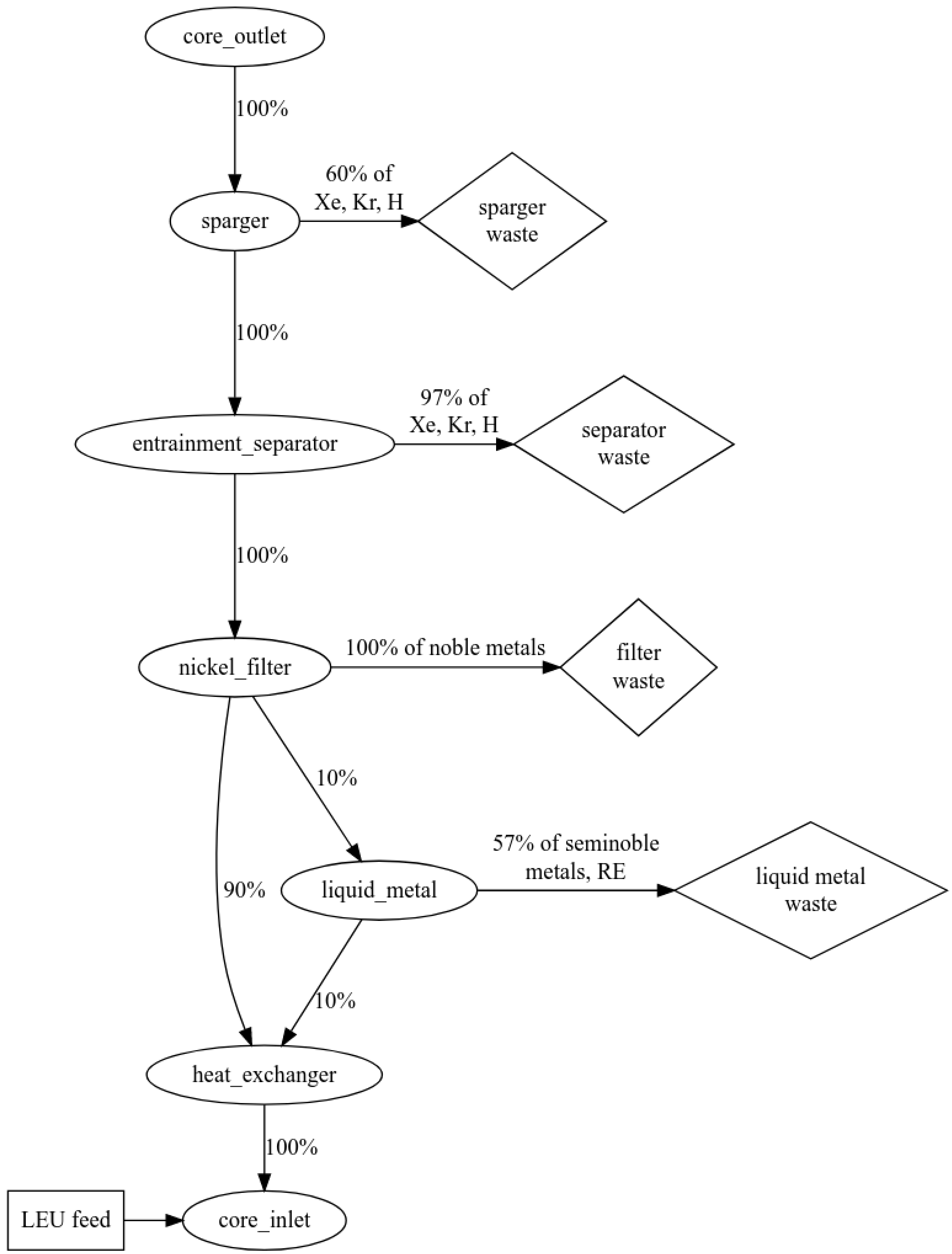

Figure 9: TAP reprocessing scheme flowchart used for SaltProc v2.0+ demonstration. Arrows represent material flows; percents - fraction of total mass flow rate; ellipses - fuel reprocessing system components; diamonds - waste streams; box shows refuel material flow.

any fuel salt treatment.

The removal rates vary among 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 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 time was selected based on Betzler *et al.* timestep refinement study [10]. For longer, lifetime-long depletion simulations, 30-day timestep size will be applied.

6 Results

The SaltProc v2.0+ online reprocessing simulation package is demonstrated for analyzing 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 by setup neutron population 15'000, the number of active cycle 400, and the number of inactive cycle 200. 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 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 v2.0+ 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 used online reprocessing strategy.

Loading initial fuel salt composition with 5% LEU into the TAP core leads to a supercritical configuration with an excess of reactivity about 1900pcm (figure 10). Without performing any fuel salt reprocessing the core became subcritical after 30 days of operation (figure 11). We obtained this result using naked Serpent without introducing any FP extraction and refueling. For the beginning of the TAP lifetime uranium enrichment in the feed has a minor effect because a 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 either with LEU 5% or LEU 19.79% feed.

The TAP core is 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 initial 1.018 to 0.88 for the 19.79%

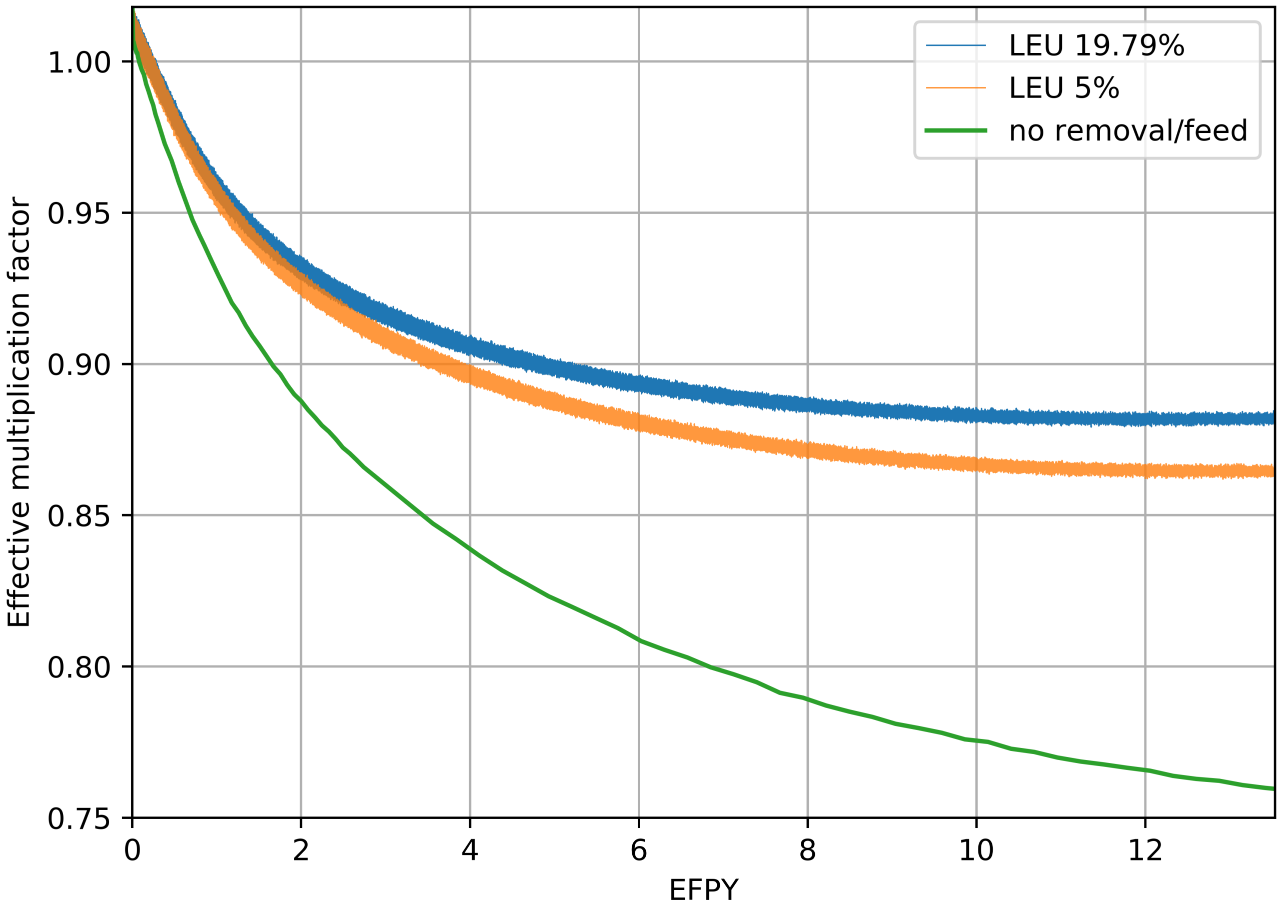

Figure 10: Effective multiplication factor dynamics for full-core TAP model for different fueling scenarios over a 13-year reactor operation. Confidence interval $\pm\sigma = 28pcm$ is shaded. Clearly, the reactor went subcritical too fast and further investigation needed to overcome this issue. Possible 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., TRU elements from spent LWR fuel); (4) adjust SVF on-the-fly by moving moderator assemblies during operation [5] or adding moderator rods only at regular intervals during shutdown for reactor maintenance [18].

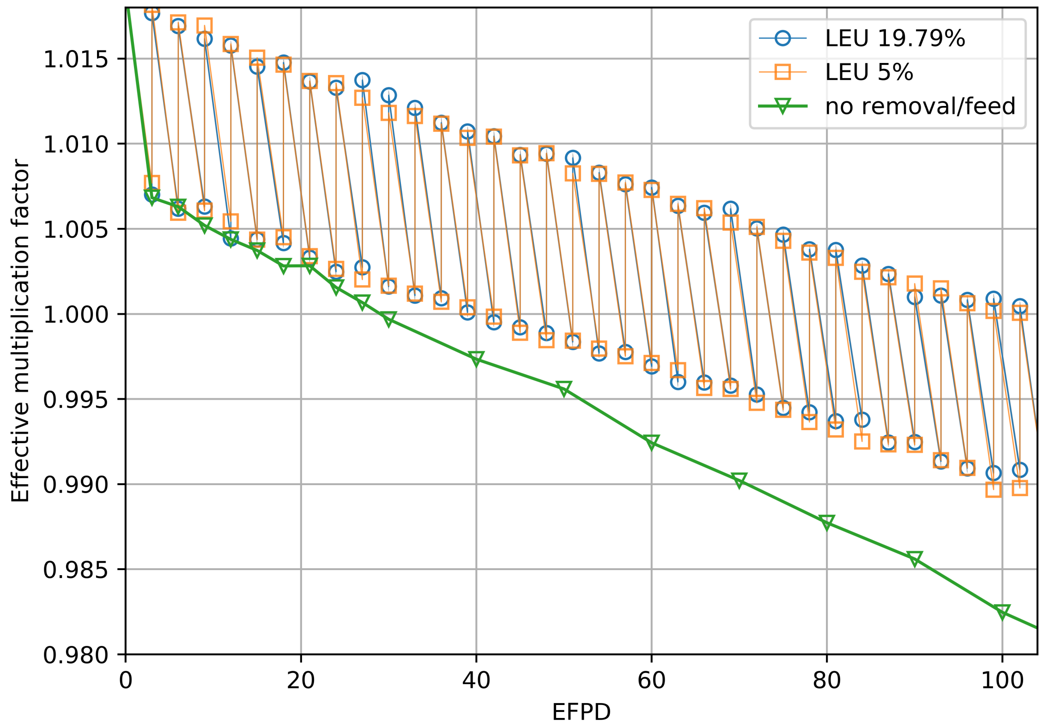

Figure 11: Zoomed effective multiplication factor for the first 104 EFPD after startup.

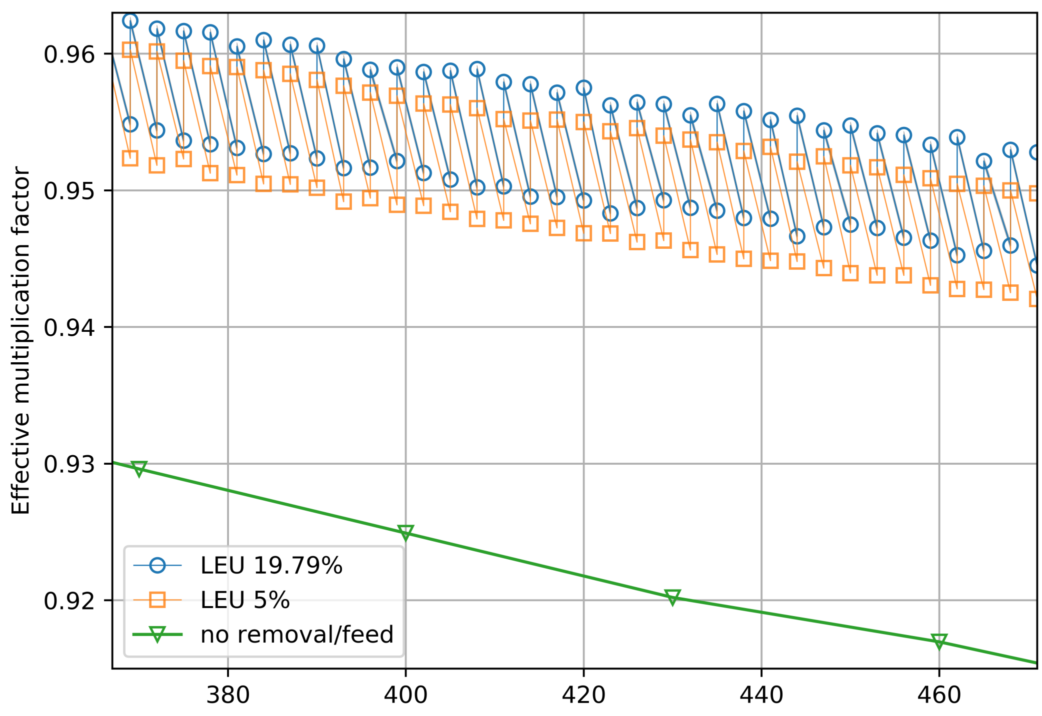

Figure 12: Zoomed effective multiplication factor for the time interval from 367 to 471 EFPD after startup.

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 v2.0+.

Acting as a complement to Figure 10, the Figure 13 shows Shannon entropy of a fission source as a function of the number of inactive cycles and clearly indicates that the Monte Carlo simulation convergences with a number of inactive cycles >200 [19].

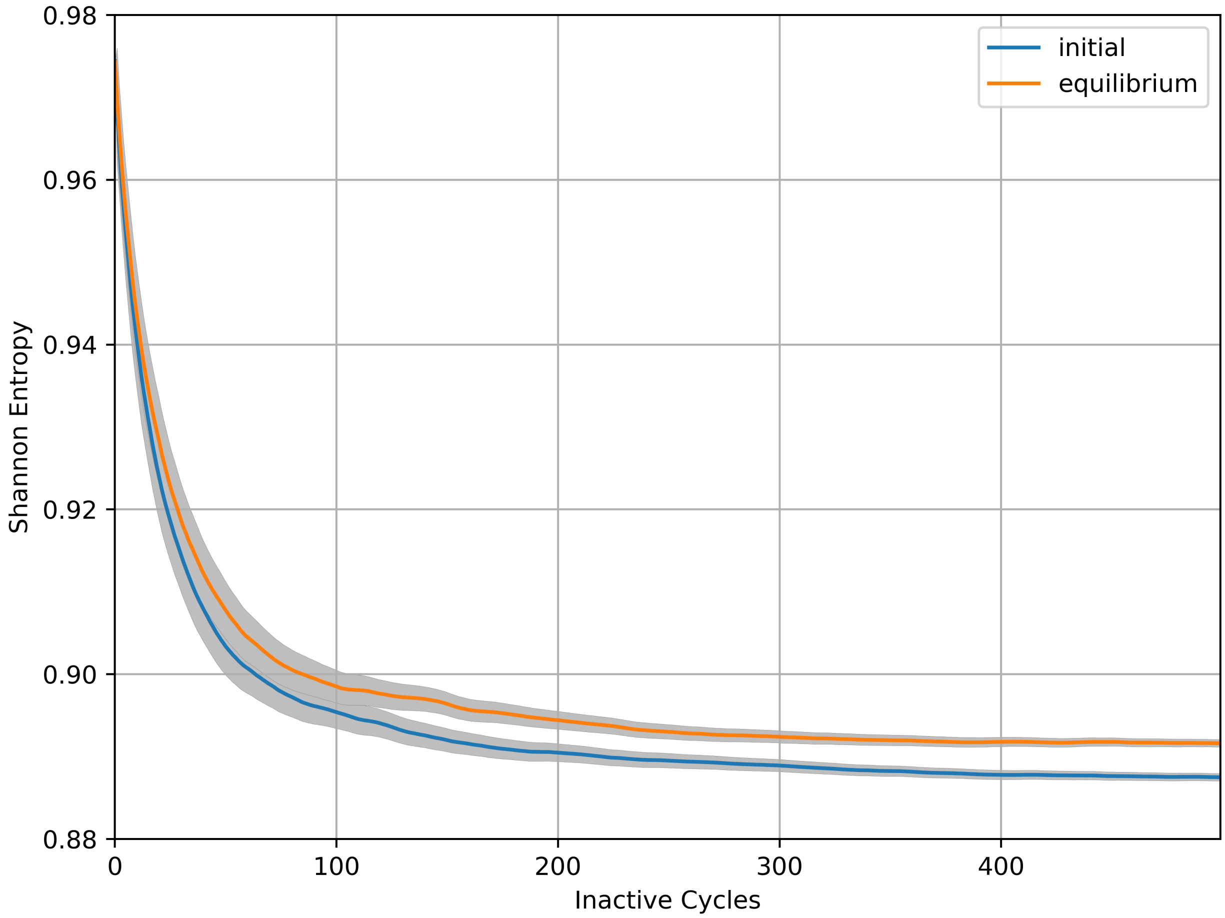

Figure 13: Shannon entropy of a fission source for initial and equilibrium fuel salt composition (19.87% LEU feed) as a function of inactive cycles number for the full core calculations with neutron population $M = 15'000$.

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 plutonium and other strong absorbers accumulating 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 [10].

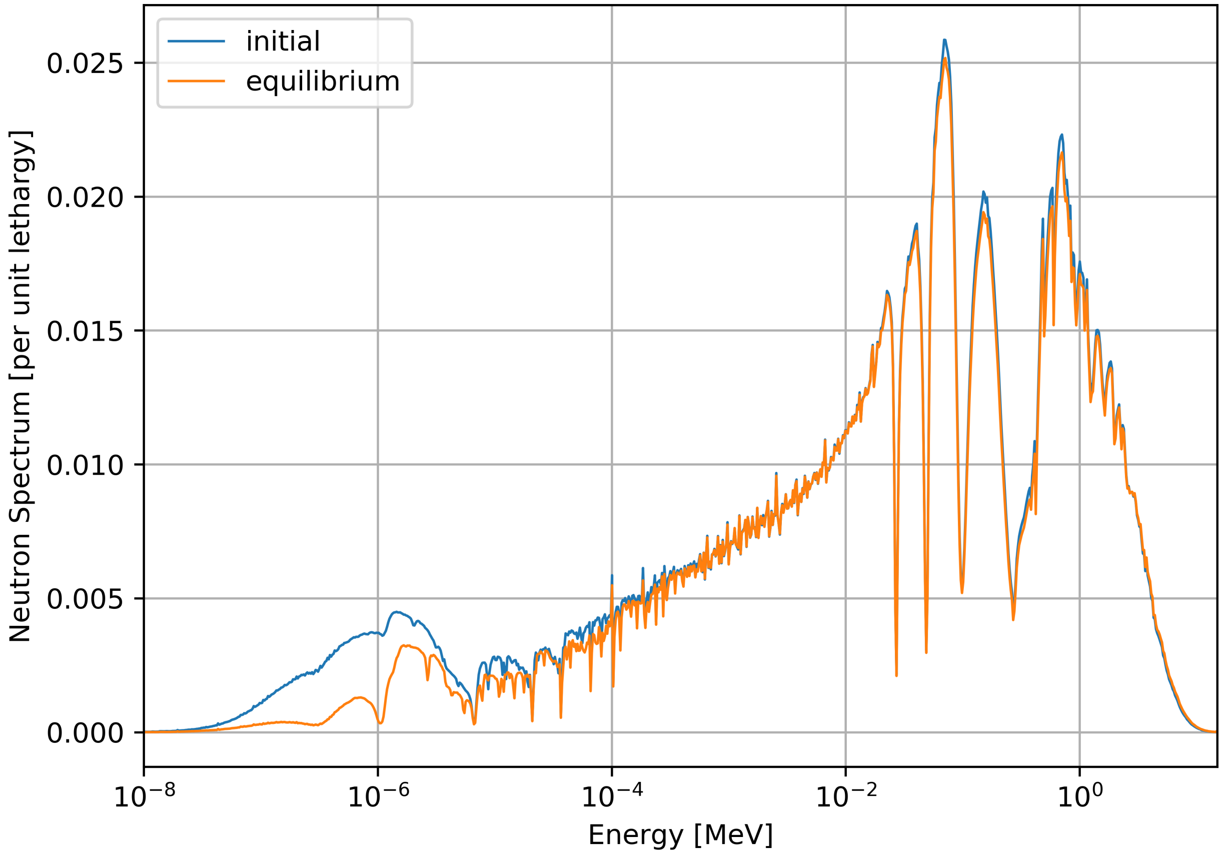

Figure 14: The neutron flux energy spectrum normalized by unit lethargy for initial and equilibrium fuel salt composition.

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 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, rate of breeding fissile ^{239}Pu from ^{238}U even in relatively hard neutron spectrum is not large enough to compensate negative effects of strong absorbers accumulation and keep the reactor critical.

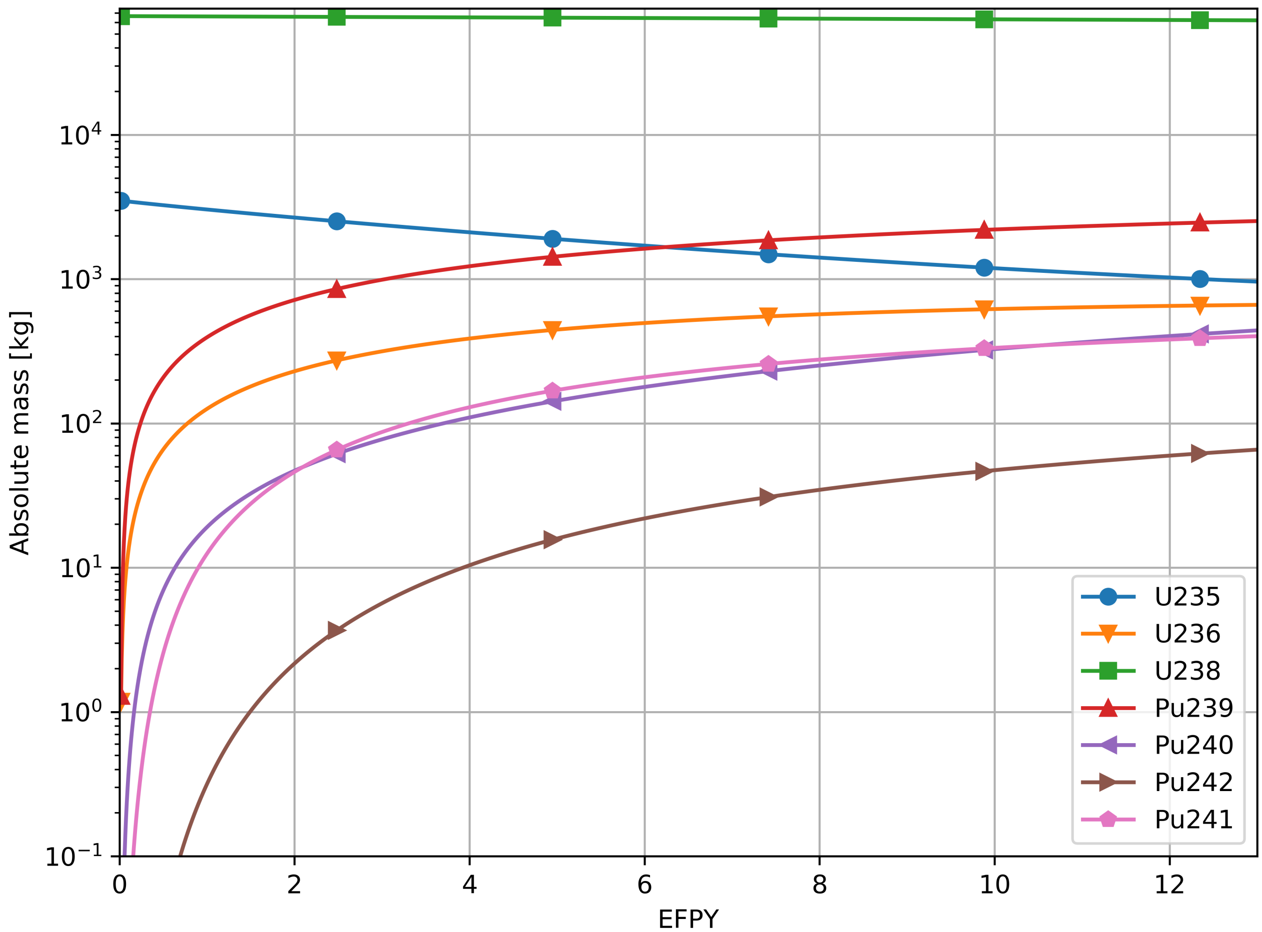

Figure 15: Mass of major nuclides during 13 years of reactor operation with 19.79% LEU feed.

We checked correctness of SaltProc v2.0+ by comparing mass of important for load-following operation isotopes (^{135}Xe, ^{135}I) to expected mass after each depletion step (figure 16). For ^{135}Xe expected mass was calculated as follows:

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

where

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

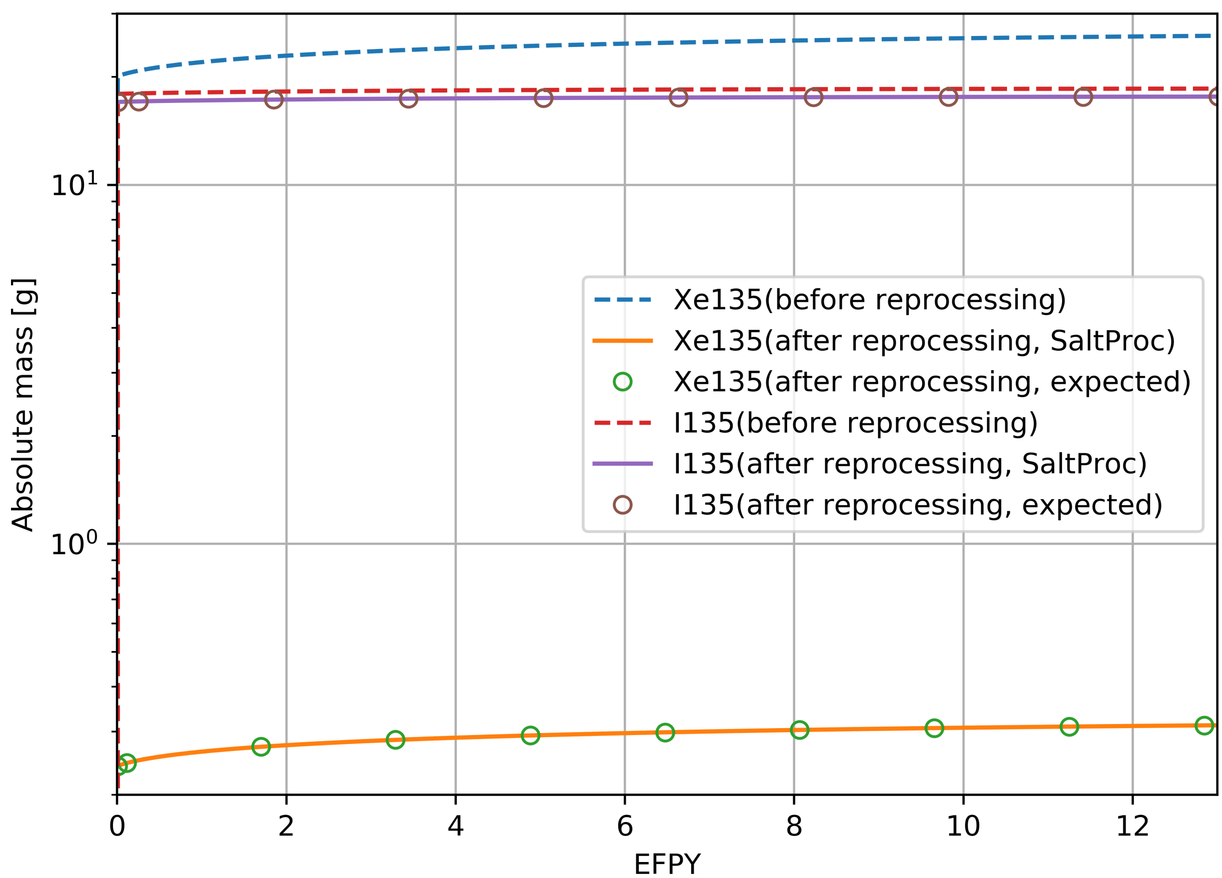

Figure 16: Mass of major neutron poison, ^{135}Xe, and its main precursor, ^{135}I, during 13 years of reactor operation before and after reprocessing.

For the iodine approach is similar, but the extraction efficiency of iodine in the nickel filter is only 5%. Figure 16 shows that SaltProc v2.0+ extraction module correctly removes target isotopes with specified extraction efficiency: SaltProc and expected mass match. Overall, the TAP fuel reprocessing system simulated with SaltProc v2.0+ allows keeping ^{135}Xe inventory in the core during operation on 100% power as low as 1g.

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 [18] reported initial *k* eigenvalue calculated for BOL about 1.035 which is 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 get a larger excess of reactivity at the BOL.

Dynamic moderator-to-fuel ratio: The TAP major feature is the ability to adjust moderator-to-volume, or SVF, ratio during lifetime by changing moderator rods configuration. Adding more moderator in the core thermalizes 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 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 assuming 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) [18]. Additionally, we will add in SaltProc v2.0+ capability to switch from one geometry file to another with a user-defined time interval.

Reprocessing scheme: Extraction efficiencies and refueling strategy of the TAP fuel reprocessing and refueling plant will be revised to make sure that all possible strong poisons are removed with 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] T. H. Group, “Hierarchical data format, version 5,” 1997.

[5] Transatomic Power Corporation, “Technical White Paper,” White Paper 2.1, Transatomic Power Corporation, Cambridge, MA, United States, Nov. 2016.

[6] Transatomic Power Corporation, “Neutronics Overview,” White Paper 1.1, Transatomic Power Corporation, Cambridge, MA, United States, Nov. 2016.

[7] B. R. Betzler, J. J. Powers, A. Worrall, S. Robertson, L. Dewan, 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), 2016.

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

[9] R. C. Robertson, “Conceptual Design Study of a Single-Fluid Molten-Salt Breeder Reactor.,” Tech. Rep. ORNL–4541, comp.; Oak Ridge National Lab., Tenn., Jan. 1971.

[10] 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, 2017.

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

[12] 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.

[13] 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.

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

[15] G. Brandl, “Sphinx: Python Documentation Generator,” 2009.

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

[17] O. D. Bank, “The JEFF-3.1.2 Nuclear Data Library,” Tech. Rep. JEFF Report 24, OECD/NEA Data Bank, 2014.

[18] 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.

[19] 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.