Turn off the colored outlines around hyperlinks with the hidelinks option in the hyperref package (I think).

# FUEL PROCESSING SIMULATION TOOL FOR LIQUID-FUELED NUCLEAR REACTORS

BY

ANDREI RYKHLEVSKII

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Nuclear, Plasma, and Radiological Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2020

Urbana, Illinois

Doctoral Committee:

Assistant Professor Kathryn Huff, Chair
Associate Professor Tomasz Kozlowski
Professor James Stubbins
Professor Luke Olson

# Table of Contents

# Chapter 1

# Introduction

## 1.1 Motivation

Humankind has only a few ways to generate reliable, non-intermittent baseload power: fossil fuels, hydropower, geothermal power, and nuclear energy. Because of increasing global climate change concerns, sources with negligible $CO_2$ footprints are crucial measures for global temperature control. Thus, from an environmental viewpoint, hydro and nuclear power are preferable ways to generate reliable power. However, local geographical conditions limit the potential for hydropower; hence, the only option left is nuclear power. Nuclear power plants provided 10% of the global electricity supply in 2018 [1]. Moreover, nuclear share in energy generation is projected to stay constant through 2040, while electricity demand will increase by 30% [2]. <Need a sentence here motivating the connection to the thesis. "This work pursues simulation tools with the potential to advance this vital option."

The Generation IV International Forum (GIF) chose Molten Salt Reactors (MSRs) among the six advanced reactor concepts for further research and development. MSRs offer significant improvements "in the four broad areas of sustainability, economics, safety and reliability, and proliferation resistance and physical protection" [3]. To achieve the goals formulated by the GIF, MSRs simplify the reactor core and improve inherent safety by using liquid coolant, which is also a fuel[1]. In a thermal spectrum MSR, liquid fuel consists of carrier salt (i.e., LiF, LiF-$BeF_2$, or LiF-NaF-KF) and fluorides of fissile and/or fertile materials (i.e., $UF_4$, $PuF_3$ and/or $ThF_4$). The fuel salt circulates in a loop-type primary circuit [4]. This innovation leads to immediate advantages over traditional, solid-fueled reactors. These

[1] Herein MSRs are assumed to be reactors with liquid fuel, which simultaneously serves as a coolant.

include near-atmospheric pressure in the primary loop, relatively high coolant temperature, outstanding neutron economy, a high level of inherent safety, reduced fuel preprocessing, and the ability to continuously remove fission products and add fissile and/or fertile elements without shutdown [5]. The possibility of continuously removing neutron poisons increases the potential fuel burnup and thus improves the resource utilization of MSRs. Finally, ~~the~~ MSRs also could be employed for the transmutation of spent fuel from current Light Water Reactors (LWRs) [6].

Recently, interest in MSRs has resurged, with multiple new companies pursuing commercialization of MSR designs[2] Both domestically and internationally. China's MSR program was initiated in 2011 and promises to ~~startup~~ start up a 2MW$_{th}$ liquid-fueled test MSR in 2020, a 10MW$_{th}$ demonstration reactor in 2025, and a gigawatt-level commercial reactor in 2050 [7]. The European Union funds the Safety Assessment of the Molten Salt Fast Reactor (SAMOFAR) project, in which several European research institutes and universities are developing various molten salt reactor prototypes such as the Molten Salt Fast Reactor (MSFR) [8] and the Molten Salt Actinide Recycler and Transmuter (MOSART) [9]. To advance these MSR concepts, particularly concerning their strategies for online reprocessing and refueling, we need computational analysis methods capturing their unique reactor physics, fuel reprocessing mechanics, and chemistry.

The context of the Ph.D. dissertation is the development and assessment of an advanced neutronics tool for fuel depletion calculations in circulating-fuel nuclear reactors. The present work introduces the open-source reprocessing simulation package, SaltProc [10], which couples with the continuous-energy Monte Carlo depletion calculation code, Serpent 2 [11], for fuel composition dynamics analysis in various MSRs taking into account a realistic, physics-driven model of an online fuel reprocessing system.

[2] Examples include liquid-fueled MSR designs from Terrapower, Terrestrial, ThorCon, Flibe, Copenhagen Atomics, Elysium, etc.

## 1.2 Fuel burnup and online reprocessing

Xe and Kr are "noble" and not "volatile"

All liquid-fueled MSR designs involve various levels of online fuel processing. Minimally, ~~volatile~~ gaseous fission products (e.g., Kr, Xe) escape from the fuel salt during routine reactor operation and must be captured. Other systems might be used to enhance the removal of those elements. Most designs also call for the removal of rare earth metals from the core since these metals act as neutron poisons. Some designs suggest a more elaborate list of elements to process (Figure 1.1), including the temporary removal of protactinium from the salt or other regulation of the actinide inventory in the fuel salt [12]. Fresh fuel salt with dissolved fissile and/or fertile material (e.g., $^{233}$U, $^{232}$Th, low-enriched uranium (LEU), a transuranic vector from LWR spent nuclear fuel (SNF)) make up the salt mass loss caused by poison removal and conserves the total mass in the primary loop.

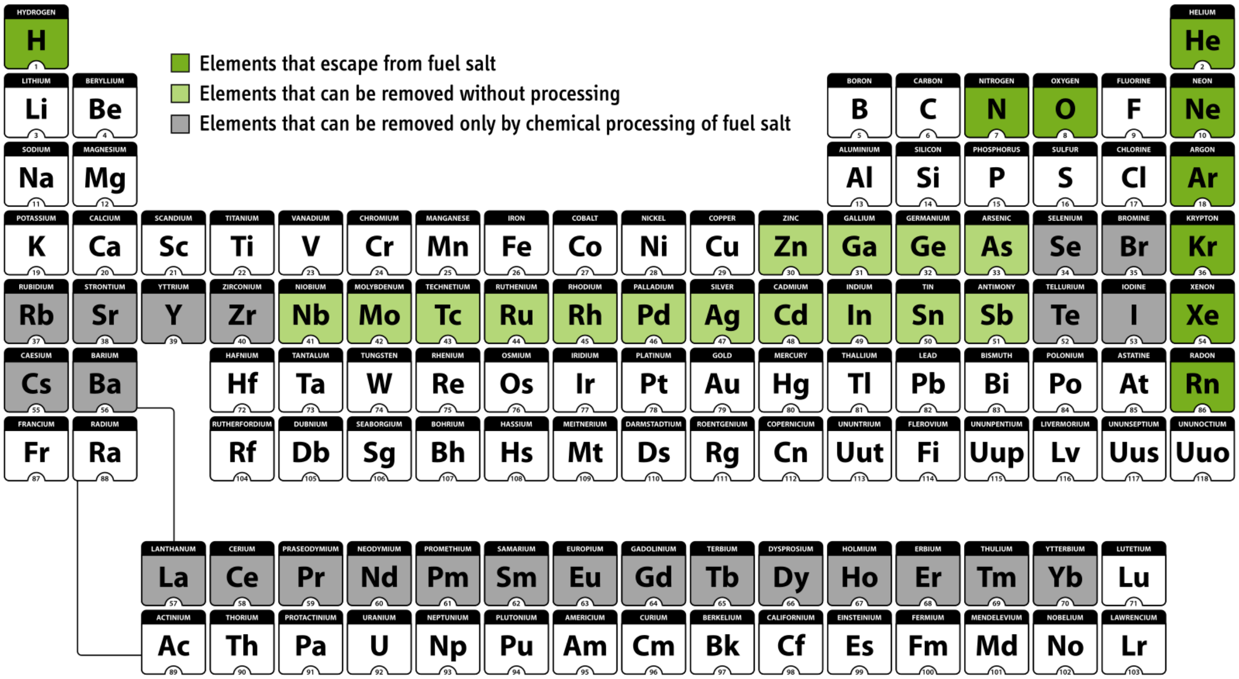

Elements that escape from fuel salt
Elements that can be removed without processing
Elements that can be removed only by chemical processing of fuel salt

Figure 1.1: Processing options for MSR fuels (reproduced from Ahmed *et al.* [12]).

? non-intermittent = continuous

Most liquid-fueled nuclear reactor concepts adopt non-intermittent separations and feeds: the core material is circulated to or from the core at all times (continuously) or specific intervals (batch-wise). In contrast, in a solid-fueled reactor, fission products and actinides remain

Perhaps "throughout its time in the core" ?

within the initial fuel material [during and after the operation until reprocessing.] The ability to perform online fuel salt reprocessing improves the potential neutronics performance of liquid-fueled reactors. First, liquid-fueled reactors can operate with ~~a~~ relatively ~~small~~ low excess reactivity because fissile material ~~is~~ can be continuously ~~being~~ added to the core. Second, continuously removing fission products, including strong absorbers (poisons), ~~should~~ Can significantly improve fuel utilization and decrease parasitic neutron absorption. Third, online reprocessing decreases the amount of decay heat, dissipating after shutdown. Finally, for a breeder[3] excess of fissile material might be continuously extracted from the core and used to startup new reactors. Nevertheless, the removal of each element from the liquid fuel salt presents a unique challenge in terms of chemical separation, storage, and disposal of the separated materials.

This sentence could begin a new paragraph.

Perhaps → Contemporary nuclear fuel depletion software lacks continuous fuel salt reprocessing modeling.

~~Continuous fuel salt reprocessing prevents the usage of most contemporary nuclear reactor fuel burnup software~~. To handle ~~the~~ material flows ~~and~~ in potential online removal and feed of liquid-fueled systems, early MSR simulation methods at Oak Ridge National Laboratory (ORNL) integrated neutronics and fuel cycle codes (i.e., Reactor Optimum Design (ROD) [13]) into operational plant tools (i.e., Multiregion Processing Plant (MRPP) [14]) for MSR fuel reprocessing system design. Extensive research ~~efforts~~ in fast and thermal MSR analysis has yielded useful tools for burnup calculations in liquid-fueled nuclear systems [15, 16, 17, 18, 19, 20, 21]. Table 1.1 presents a list of recent efforts, along with the main features of the employed methods and software.

with the current phrasing, this sentence seems to contradict the first sentence in the paragraph. Consider describing these tools as "specialized"?

Two main online reprocessing simulation approaches ~~are commonly used~~ have been demonstrated in the literature: batch-wise and continuous. In the batch-wise approach, the burnup simulation stops at a given time and restarts with a new liquid fuel composition (after removal of discarded materials and addition of fissile/fertile materials).

ORNL researchers have developed ChemTriton, a Python script for SCALE/TRITON,

[3] conversion ratio (CR) ≡ fissile generated/fissile consumed: if CR < 1, the reactor is a "converter"; CR ≡ 1, an "isobreeder"; CR > 1, a "breeder."

Table 1.1: Tools and methods for liquid-fueled MSR fuel salt depletion analysis.

| | Nuttin *et al.*, 2005 [22] | Aufiero *et al.*, 2013 [17] | Betzler *et al.*, 2018 [23] | Present work |
|---|---|---|---|---|
| Neutronics software | MCNP<br>REM<br>stochastic | Serpent 2<br><br>stochastic | SCALE6.2<br>ORIGEN-S<br>deterministic | Serpent 2<br><br>stochastic |
| Geometry | unit cell | full-core 3D | unit cell | full-core 3D |
| Removal/feed | continuous | continuous | batch-wise | batch-wise |
| Separation efficiency | fixed, must be defined by user before simulation | | | function of many parameters |
| Fuel reprocessing plant | single component, "black" box model | | | realistic multi-component model |
| Reactivity control | continuous adjustment of fissile material injection | | batch injection of fissile material | periodical adjustment of geometry and fissile material injection |
| Safety parameters evolution | thermal feedback | not considered | thermal feedback | thermal feedback, control rod worth, axial offset |

which employs the batch-wise approach to simulate a continuous reprocessing and refill for either single or multiple fluid designs. ChemTriton models salt treatment, separations, discharge, and refill using SCALE/TRITON depletion simulation over small time steps to simulate continuous reprocessing and deplete the fuel salt [23, 24].

Be clear that you are no longer talking about ChemTRITON

Accounting for continuous removal or addition of material presents a greater challenge since it requires adding a term to the Bateman equations. Both ORIGEN [25] ~~or~~ and the Serpent burnup routine [26] solves a set of the Bateman equations using one-group averaged flux and transmutation cross sections obtained from a transport calculation. The Bateman equations describe the rate of change of each isotope, $i$, due to neutron induced reactions and decay processes [27]:

$$\frac{dN_i}{dt} = \underset{(1)}{\sum_{m=1}^{M} l_{im}\lambda_m N_m} + \underset{(2)}{\phi\sum_{m=1}^{M} f_{im}\sigma_m N_m} - (\underset{(3)}{\lambda_i} + \underset{(4)}{\phi\sigma_i} + \underset{(5)}{r_i} - \underset{(6)}{f_i})N_i + F_i\Big|i \in [1, M] \tag{1.1}$$

where

$N_i$ = atom density of nuclide $i$ $[cm^{-3}]$

$M$ = number of nuclides $[-]$

$l_{im}$ = fraction of decays of nuclide $m$ that result in formation of nuclide $i$ $[-]$

$\lambda_i$ = radioactive decay constant of nuclide $i$ $[s^{-1}]$

$\phi$ = neutron flux, averaged over position and energy $[cm^{-2}s^{-1}]$

$f_{im}$ = fraction of neutron absorption by nuclide $m$ leading to the formation of nuclide $i$ $[-]$

$\sigma_m$ = average neutron absorption cross section of nuclide $m$ $[cm^2]$

$r_i$ = continuous removal rate of nuclide $i$ from the system $[s^{-1}]$

$f_i$ = continuous feed rate of nuclide $i$ $[s^{-1}]$

$F_i$ = production rate of nuclide $i$ directly from fission $[cm^{-3} \cdot s^{-1}]$.

The terms on the right-hand side of the equation represent:

(1) production of species $i$ as a result of the decay of all the nuclides present;

(2) production of species $i$ as a result of neutron capture by all nuclides present;

(3) loss of nuclide $i$ through its own decay;

(4) loss of nuclide $i$ as a result of neutron capture;

(5) loss of nuclide $i$ through continuous removal from the system;

(6) gain of nuclide $i$ as a result of continuous feed to the system.

expand acronym?

Nuttin *et al.* developed an in-house depletion code called REM, which directly couples with MCNP [28] to simulate fuel salt material evolution in a simplified Molten Salt Breeder Reactor (MSBR)-like liquid-fueled system. That work directly integrated the Bateman differential equations using neutron flux from ~~the~~ MCNP, tracking all the isotopes available in the data library, and ~~control~~ controlling reactivity to maintain ~~the~~ reactor ~~critical~~ criticality [22].

In a similar vein, Aufiero *et al.* extended Serpent 2 for continuous reprocessing simulations by explicitly introducing "reprocessing" time constants into the system of Bateman equations and adding effective decay and transmutation terms for each nuclide [17]. The developed extension directly accounts for the effects of online fuel reprocessing on depletion calculations and features a reactivity control algorithm. The extended version of Serpent 2 was assessed against a dedicated version of the deterministic ERANOS-based EQL3D procedure in [15] and applied to analyze the MSFR fuel salt isotopic evolution.

It may be helpful to give a definition of the time constants. (in an equation)

More recently, Betzler *et al.* added to SCALE/TRITON continuous removals capability for depletion simulation [21]. Similar to Aufiero *et al.* ~~effort~~, ~~the~~ this extended SCALE/TRITON directly adds feed and removal terms in the burnup matrix and solves it using existing ORIGEN capabilities. TRITON's continuous reprocessing capability was validated against the batch-wise script ChemTriton for the unit-cell Molten Salt Reactor Experiment (MSRE)-like model. Unlike ChemTriton, this new capability will be available for all SCALE users in the 6.3 release. However, at the moment, it is undergoing extensive testing and validation procedures and unavailable for external users.

You use "the" as if the reader should know about this model already.

Most of the existing tools in the literature represented the fuel salt reprocessing plant as an invariable "black box" model, which removes target elements all at once with a fixed efficiency, determined by the user before starting the depletion simulation. Typically, such a "black box" model is characterized by a vector of removing elements and their extraction efficiencies:

$$\begin{bmatrix} N_0^{in} \\ \vdots \\ N_e^{in} \\ \vdots \\ N_E^{in} \end{bmatrix} \times \begin{bmatrix} \epsilon_0 \\ \vdots \\ \epsilon_e \\ \vdots \\ \epsilon_E \end{bmatrix} = \begin{bmatrix} N_0^{out} \\ \vdots \\ N_e^{out} \\ \vdots \\ N_E^{out} \end{bmatrix} \tag{1.2}$$

put these in displaymode?

where $N^{in/out}$ is the number density [$cm^{-3}$], and $\epsilon$ is the extraction efficiency [$-$] for all elements $e$ in $(0, E)$. The ~~main issues related to~~ static “black box” model assumptions in the literature ~~include:~~ neglect:

Time varying extraction:

~~**Time-independent separation efficiency vector.**~~ Realistically, long-term reactor operation will require a time-dependent extraction efficiency vector. The current tools treat separatation efficiency as constant.

~~**The separation efficiency is independent of the reactor’s operational parameters.**~~ The impact of operational parameters on separation efficiency. In reality, the extraction efficiency depends on temperature, power level, current fuel salt isotopic composition, and material mass flow rate. <more detail here>

Discrete component performance and dynamics in the multi-component system.

**All reprocessing plant components are treated as a single “black box” component** in existing simulation tools. However, the fuel salt in a reprocessing plant undergoes many separate components (e.g., helium bubbling, nickel mesh filter, etc.) that target specific elements. Some of these components can be connected in series, parallel, or series-parallel. The “black box” model (only single process) requires extensive pre-simulation analytic work from the user to calculate the lumped separation efficiency vector before a simulation is run and cannot be adjusted during the simulation. Additionally, treating the processing system as a single “black box” ~~may lose~~ neglects dynamics related to relative component flow rates. Finally, the discrete waste streams from each component ~~cannot be~~ are not tracked separately in “black box” tools. ~~which~~ However, this information is necessary for fuel reprocessing system optimization.

This paragraph needs to be separated from the above list.

maybe move this to appear above the list?

Some of the tools listed in Table 1.1 used significant approximations that may lead to inaccurate fuel evolution predictions and others unavailable for external users. This work introduces an open-source simulation package, SaltProc, which expands the capability of the continuous-energy Monte Carlo Burnup calculation code, Serpent 2, for depletion calculations with user-specified liquid-fueled MSR parameters to determine fuel salt composition evolution.

this paragraph needs a sentence saying that SaltProc does not make these approximations.

## 1.3 Operational and safety parameter~~s~~ evolution

In contrast with conventional solid-fueled reactors ~~in which~~ with in-core fuel residence ~~time is~~ averaging 4-5 years[4], ~~the~~ on initial MSR fuel salt batch stays in the MSR~~'s~~ primary loop ~~during the whole~~ throughout the reactor lifetime. Therefore, the fuel salt accumulates Fission Products (FPs) not captured by the fuel reprocessing system as well as transuranic elements[5]. Continuous fuel salt composition evolution has a significant influence on the neutron energy spectrum and, consequently, affects the reactor behavior, necessitating additional safety analysis.

Nuttin *et al.* studied the evolution of a key safety parameter, the temperature reactivity feedback coefficient, estimating it for the MSBR at startup and equilibrium. The temperature coefficient of reactivity quantified reactivity changes due to temperature increase in the core and was calculated in that work as:

$$\alpha = \frac{k_{1200} - k_{900}}{\delta T} \tag{1.3}$$

where

$$k_{900}, k_{1200} = \text{multiplication coefficients at 900K and 1200K } [-]$$
$$\delta T = 300\ [K].$$

That work showed that the fuel temperature coefficient (FTC) at startup and equilibrium is $-1.5$ and $-1.0 pcm/K$, respectively. Percent mille ($pcm$) is the unit of reactivity equal to a $k_{eff}$ of $10^{-5}$. Nuttin *et al.* also reported a positive and time-invariant total temperature coefficient ($+0.8 pcm/K$) [22]. Recently, Park and colleagues expanded that approach to a full-core high-fidelity MSBR model and estimated safety parameters evolution over 20 years

This definition could be more precise. $pcm = 10^{-5}$

[4] For the typical 18-month cycle, during refueling personnel removing 1/3 of the fuel assemblies, re-arranging other assemblies, and loading fresh fuel into the core. Thus, each fuel assembly is kept in the core at most $3 \times 18 = 54$ months.

[5] The chemical elements with atomic numbers greater than uranium (92).

of operation [19]. These calculations showed a relatively large negative total temperature coefficient during the 20 year~~s of the~~ reactor operation; During that time, the coefficient magnitude weakens from −3.21 to −1.41$pcm/K$ ~~at~~ from startup ~~and~~ to equilibrium, respectively. Additionally, that work reported a control rod worth deterioration from 2099$pcm$ to 1970$pcm$ due to neutron spectrum hardening during reactor operation.

More recently, Betzler *et al.* [29] reported ~~key~~ safety parameter~~s~~ evolution for the Transatomic Power (TAP) MSR: the fuel reactivity coefficient at Beginning of Life (BOL) and 15 years from BOL ~~is~~ was negative and decreasing slowly over the reactor lifetime; the moderator reactivity coefficient ~~is~~ was small and positive at BOL and becomes negative after 15 years of operation. Overall, thermal feedback seems to be stronger in the TAP reactor and deteriorates insignificantly during the reactor operation. Notably, the authors ignored material density change with temperature to simplify temperature coefficient~~s~~ calculation; thus, only Doppler broadening was taken into account. ~~Finally, t~~The researchers reported the total worth of all control rods in the TAP core only for the startup fuel composition.

numbers here rather than "small"

The evolution of control rod worth in the TAP has not been reported in the literature before. This dissertation work illuminated the evolution of essential safety parameters (fuel, moderator, total temperature coefficient, control rod worth) for the TAP MSR at various moments during the reactor operation. Additionally, I investigated the impact of neutron poison accumulation (e.g., $^{135}$Xe) in the fuel salt during short-term transients (i.e., load following) on major safety characteristics. <here, you could note and cite your own journal articles, pointing to where, in this dissertation, those results appear>

## 1.4 Background Summary

State-of-the-Art software packages for depletion analysis and evolution of safety parameters in the liquid-fueled MSR are reviewed in Section 1.2. Based on this summary, I have identified a few possible directions for the improvement of MSR tools:

**Reproducibility/availability.** Serpent is the only contemporary nuclear reactor physics software that can perform depletion calculations that can take into account online fuel salt reprocessing regimes. However, this built-in online reprocessing routine is undocumented; ~~and~~ the discussion forum for Serpent users is the only useful source of information at the moment. Other mentioned tools are under the closed-source license or available for internal users only. These issues can be a barrier to reuse research software and to reproduce scientific results. Thus, a new, open-source, reproducible tool for fuel processing simulation would assist in the production of reproducible research in the area of liquid-fueled reactor modeling.

Same with Serpent. Serpent is not under an open license

**Realistic fuel reprocessing system model.** Significant approximations in fuel reprocessing parameters deteriorate fuel salt composition predictions since the evolution of safety parameter accuracy is strongly dependent on fuel salt composition. A realistic fuel reprocessing system model will allow reprocessing component parameter optimization, increase the fidelity of fuel and waste stream composition calculations, and advance reprocessing system design.

If not all, list those that do and those that don't

**Variable extraction efficiency.** Most ~~of the~~ research efforts in the literature assume~~d~~ ideal 100% extraction efficiency of all removed elements, which stayed constant during the whole reactor lifetime. Realistically the efficiency is time-dependent and changes with respect to operational parameters: temperature, power level, salt composition, etc. Thus, the ability to set up dynamic separation efficiency must be added in MSR simulation tools to advance depletion calculations.

**Reactivity control.** Reconfigurable moderator configuration in the TAP core presents a challenge because of the core geometry changes with time. The reactivity control module, which adjusts the core geometry to maintain criticality, would be an exceptional capability for simulating new, more advanced MSR concepts and short-term transients.

**Safety characteristics evolution during reactor operation.** The MSR fuel salt accumulates FP's \glspl{FP} and transuranic elements, which significantly shift the neutron energy spectrum. Neutron energy hardening might worsen the core safety during operation. The impact of the fuel salt evolution on the MSR safety parameters must be carefully investigated and reported.

This sentence, following the previous one, implies that FPs harden the spectrum... I thought they soften the spectrum. New fuel hardens the spectrum. (and TRUs)

This work aims to overcome these issues and demonstrate the tool capabilities for a two promising MSR concepts.

## 1.5 Objectives and outline of the work

Most of the existing MSR depletion simulators usually assume ideal efficiency (100% of the target nuclide is being removed) of the neutron poison removal process (see Section 1.2). The main goal of this dissertation is to develop a generic open-source tool, SaltProc, capable of simulating a wide range of liquid-fueled systems — including multi-fluid and multi-region designs — and validate it against existing modeling efforts. Additionally, SaltProc enables poison extraction simulation based on a realistic physics-based fuel processing model.

The structure of the thesis is as follows. Chapter 1 serves as a literature review, providing background on fuel burnup, online fuel reprocessing approaches, safety parameter evolution during reactor operation, and how these concepts have been applied to a wide range of MSRs in the literature. Chapter 2 ~~covers~~ details ~~modeling on~~line reprocessing ~~details~~ modeling and the proposed computation tool architecture. In an attempt to avoid the pitfalls of a "black box" understanding and to identify method limitations at an early stage, governing equations and working principles are stated and discussed. Chapter 3 presents equilibrium-seeking results for the MSBR as well as essential operational and safety parameters for both the initial and equilibrium states.

Additionally, the benefits of continuous fission product removal for a thermal MSR are evaluated at the end of chapter 3. Chapter 4 covers SaltProc demonstration and validation needs a comma

efforts with a focus on the TAP MSR, taking into account adjustable moderator configuration. Chapter 5 gives the safety parameter overview and its evolution during the TAP lifetime-long reactor operation. Moreover, the safety parameters dynamics during short-term transients have been evaluated at the end of chapter 5. The final chapter summarizes this work's contribution to the nuclear community, and a conclusion is offered together with an outlook for future work on the topic.

# Chapter 2

# Online reprocessing modeling approach

## 2.1 Fuel salt reprocessing overview

this has a religious meaning in american english. Perhaps "intentional design" or "design optimization"

Removing specific chemical elements from a molten salt is a complicated task that requires intelligent design (e.g., chemical separations equipment design, fuel salt flows to equipment). This section contains a brief overview of a generic MSR fuel salt reprocessing system; modeling such systems is the focus of the current dissertation.

### 2.1.1 Gas separation system

Gaseous fission products (e.g., Xe) must be removed from the fuel salt to avoid reactor poisoning, especially during startup and power maneuvering. This is particularly true for $^{135}$Xe, with its ~~extensive~~ strong neutron capture cross section ($\approx 10^6 \ldots 10^7$ b in a thermal energy range). $^{135}$Xe is produced directly from fission in about 0.3% of $^{235}$U fissions ($\gamma_{^{135}Xe}$), but an even larger fraction of $^{135}$Xe is produced by the decay of $^{135}$I and $^{135}$Te (Table 2.1). $^{135}$I and $^{135}$Te yields from fission are $\gamma_{^{135}I} = 3.6\%$ and $\gamma_{^{135}Te} = 2.5\%$, respectively. Thus, the total $^{135}$Xe production from fission is about 6.4% of fissions (of $^{235}$U), most of this is from $^{135}$I and $^{135}$Te decay. Noble gases (e.g., tritium, xenon, and krypton) can be removed from the fuel salt as follows:

(a) a bubble generator injects helium bubbles in the salt stream;

(b) noble gases migrate ~~promptly~~ to the helium bubbles ~~because of~~ due to their ~~extreme~~ insolubility in the salt [30];

(c) and a gas separator discharges the fission-product-rich bubbles from the salt to the off-gas system.

Table 2.1: $^{135}$Xe production sources and principal rate constants involved (reproduced from Kedl *et al.* [31]).

| $^{135}$**Xe gain mechanism** | **Principal rate parameters involved** |
|---|---|
| Direct from fission<br>yield $\gamma_{{}^{135}Xe}$= 0.003 | $\Sigma_f \gamma_{{}^{135}Xe} \phi$ (for $^{235}$U fission) |
| $^{135}$I decay<br>yield $\gamma_{{}^{135}Xe}$= 0.036, it decays to $^{135}$Xe with $\tau_{1/2}$ = 6.68 h | $\Sigma_f \gamma_{{}^{135}I} \phi$ (for $^{235}$U fission) |
| $^{135}$Te decay<br>yield $\gamma_{{}^{135}Xe}$= 0.025, it decays to $^{135}$I with $\tau_{1/2}$ = 19 s | $\Sigma_f \gamma_{{}^{135}Te} \phi$ (for $^{235}$U fission) |

Figure 2.1 shows the key pathways for xenon production, accumulation, and removal in a typical MSR. Figure 2.2 shows the conceptual design of the MSBR gas separation system. In that system, helium ~~Helium~~ bubbles of a specific size are introduced in a salt stream via the primary pump bowl. These bubbles absorb noble gases before being separated from the salt by a gas separator. ORNL suggested that the MSBR off-gas system would inject $d$ = 0.508mm helium bubbles in the pump bowl, redirect 10% of the fuel salt flow through a bubble separator to remove the bubbles, and then return the flow into the pump suction. Robertson *et al.* reported that the helium bubble size was approximately 25% of the throat width (blue circle on Figure 2.3) and was independent of the gas flow rate [30]. Consequently, it is possible to regulate the helium bubble size by changing the throat width in the bubble generator.

did ORNL give a rate?

To realistically model the gas separation system, we need a mathematical model that describes noble gas extraction efficiency during reactor operation. Particularly, a model of xenon extraction efficiency as a function of sparger design parameters is needed to accurately model the $^{135}$Xe removal in a fuel salt depletion simulation. The gain and loss terms for $^{135}$Xe dissolved in the fuel salt are listed in Tables 2.1 and 2.2. The removal efficiency for the xenon in the pump bowl was measured during Molten Salt Reactor Experiment (MSRE)

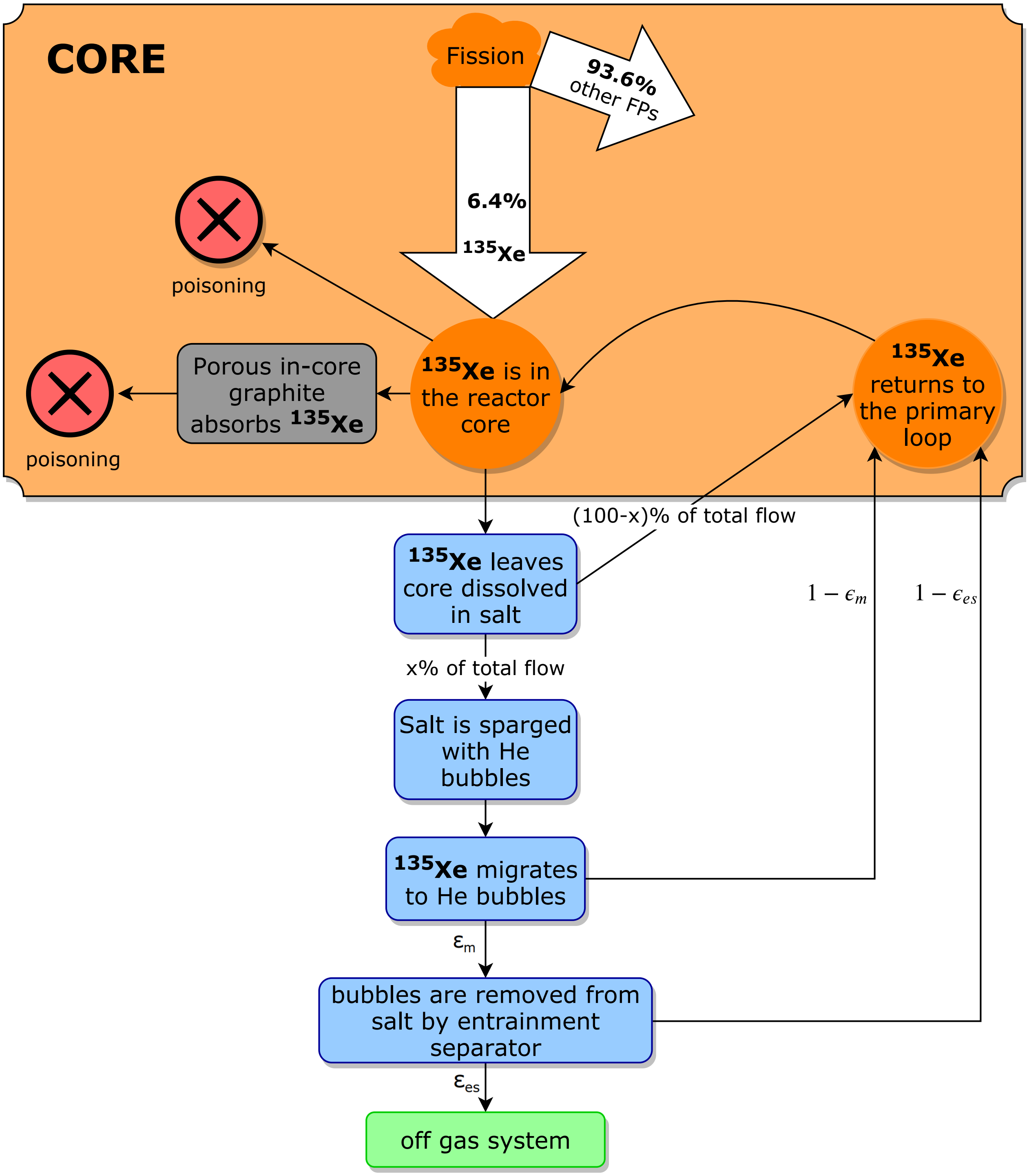

CORE
Fission
93.6% other FPs
6.4% $^{135}$Xe
poisoning
poisoning
Porous in-core graphite absorbs $^{135}$Xe
$^{135}$Xe is in the reactor core
$^{135}$Xe returns to the primary loop
(100-x)% of total flow
$^{135}$Xe leaves core dissolved in salt
$1 - \epsilon_m$
$1 - \epsilon_{es}$
x% of total flow
Salt is sparged with He bubbles
$^{135}$Xe migrates to He bubbles
$\epsilon_m$
bubbles are removed from salt by entrainment separator
$\epsilon_{es}$
off gas system

Figure 2.1: Schematic of $^{135}$Xe circulation in a generic MSR. $x$ is the fraction of fuel salt flow from the pump discharge redirected to the gas separation system, while $\epsilon_m$ and $\epsilon_{es}$ are the efficiencies of migration (of $^{135}$Xe to the helium bubbles in the sparger) and separation (of gas in the entrainment separator), respectively. The orange color represents the fuel salt in the primary loop, the blue color represents the gas separation system, and the gray color is the moderator in the core. Fission yields assume $^{235}$U fission only.

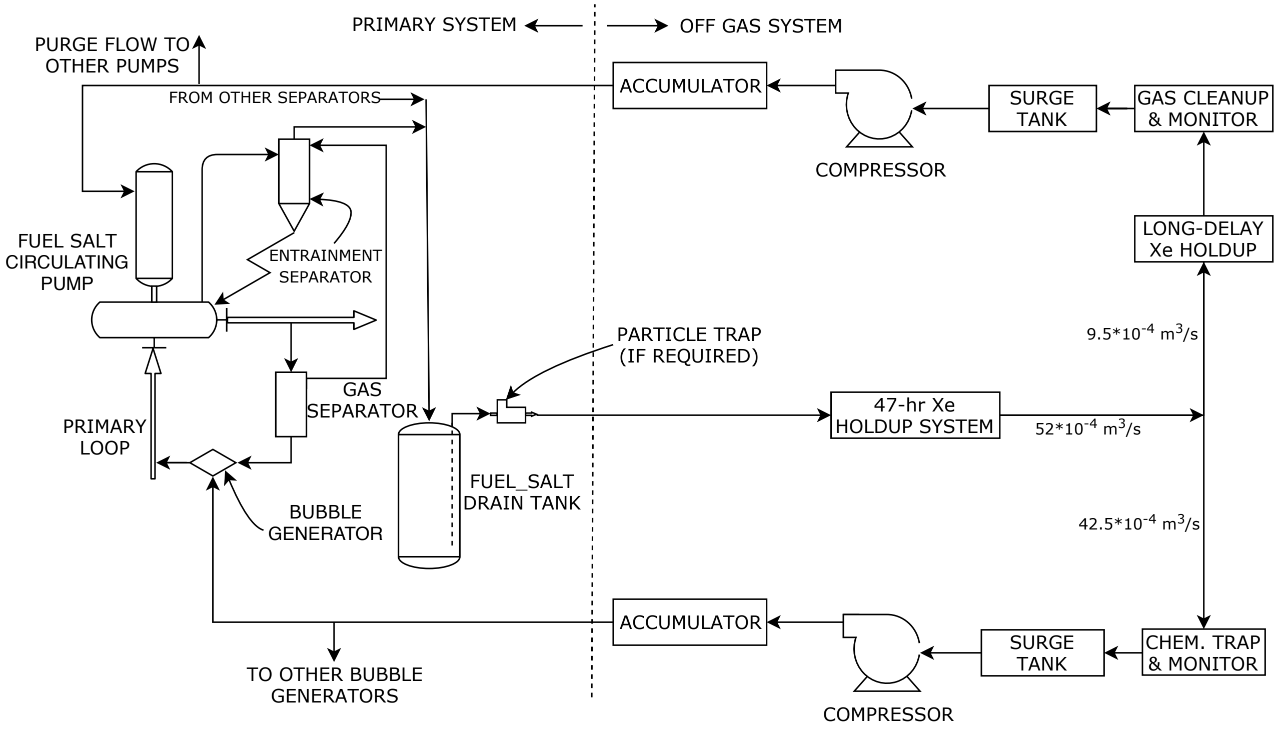


Figure 2.2: Schematic flow diagram of the MSBR gas separation system (reproduced from Robertson *et al.* [30]).

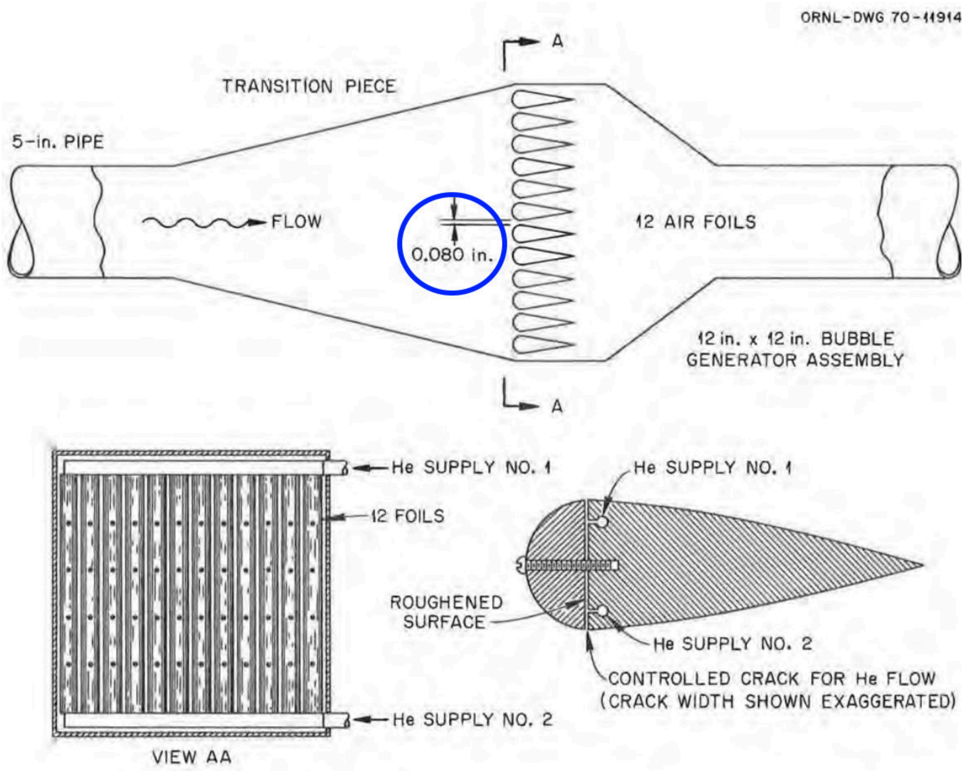


Figure 2.3: Preliminary concept of an MSBR bubble generator (reproduced from Robertson *et al.* [30]). The blue circle shows throat width, which determines bubble size.

Table 2.2: $^{135}$Xe loss terms and principal rate constants involved (reproduced from Kedl *et al.* [31]).

| **$^{135}$Xe loss mechanism** | **Principal rate parameters involved** |
|---|---|
| Decay of dissolved $^{135}$Xe ($\tau_{1/2}$ = 9.1 h) | Decay constant ($\lambda$) |
| $^{135}$Xe burnup<br>dissolved xenon-135 burnup as it passes through the core | Neutron flux ($\phi$) |
| $^{135}$Xe migrated to helium bubbles | Removal efficiency ($\epsilon_m$) |
| $^{135}$Xe transferred into circulating He bubbles; this xenon will eventually be burnup, decay, or stripped via bubble separator | Mass transfer coefficient ($h$), decay constant ($\lambda$), neutron flux ($\phi$), bubble removal efficiency ($\epsilon_{es}$) |

operation. However, the technical report ORNL-4069 by Kedl-Houtzeel only stated its range (from 50 to 100%) and concluded, "It is probably a complex parameter like the circulating-void fraction and depends on many reactor operational variables" [31]. ($^{135}$Xe burnup and decay rates are well known.) this sentence seems unneccessary.

Peebles *et al.* in ORNL-TM-2245 has reported xenon removal efficiency ($\epsilon_{Xe}$) in a gas separation system as a function of many parameters [32]:

$$\epsilon_{Xe} = \frac{1 - e^{-\beta}}{1 + \alpha} \tag{2.1}$$

where

$$\alpha = \frac{RTQ_{salt}}{HQ_{He}} \tag{2.2}$$

$$\beta = \frac{K_L a A_C L(1 + \alpha)}{Q_{salt}} \tag{2.3}$$

$$R = \text{universal gas constant } [L \cdot Pa \cdot mol^{-1} \cdot K^{-1}]$$

$$T = \text{salt temperature } [K]$$

$$Q_{salt} = \text{volumetric salt flow rate } [m^3/s]$$

$$Q_{He} = \text{volumetric helium flow rate } [m^3/s]$$

consider reporting the variables in order of their appearence in the equation.

$H$ = Henry's law constant for solute gas $[Pa \cdot mol^{-1} \cdot L]$

$a$ = gas-liquid interfacial area per unit volume $[m^{-1}]$

$A_C$ = contactor cross section $[m^2]$

$L$ = contactor length $[m]$

$K_L$ = liquid phase mass transfer coefficient $[m/s]$.

Most of the input parameters for that correlation are obvious and easy to obtain from the system component design. The mass transfer coefficient for transferring xenon into helium bubbles ($K_L$) can be estimated experimentally, but published information is currently insufficient to inform an accurate mathematical model appropriate for Computational Fluid Dynamics (CFD). Thus, Peebles *et al.* reported the mass transfer coefficient correlation for the MSBR salt (LiF-BeF$_2$-ThF$_4$-UF$_4$) but for a limited case. While it is out of the scope of this work to accurately estimate mass transfer coefficient, this work seeks to provide a tool which would allow the user to specify any mathematical model for a separation efficiency.

it feels like this should be expanded... Provided a mass transfer coefficient, the user can incorporate it into the model.

Equation 2.1 would apply to other noble gases (e.g., Kr), but Henry's law constant ($H$) and the mass transfer coefficient ($K_L$) would be different. Current effort at the University of Illinois at Urbana-Champaign, namely "Enabling Load Following Capability in the Transatomic Power MSR," [33] has a goal to determine mass transfer coefficients for various gaseous fission products (Ar, Kr, Xe) using experiments, enabling CFD and multi-physics simulations of such reactors. As a result, the obtained mathematical model for gas removal efficiency might be employed to inform a realistic physics-based fuel reprocessing model in SaltProc.

"Vary by element"

### 2.1.2 Insoluble fission product filtering

~~The decay chain of~~ Approximately 40% of FPs ~~has~~ can have gaseous elements in ~~it~~ their decay chains. Some of the non-gaseous FPs produced in the MSR core (e.g., noble and semi-noble metals) have negligible solubility in the molten salt. Some fraction of noble and semi-noble solid FPs plate out onto the internal surfaces of the primary loop equipment, complicating their removal [34]. The remaining noble and semi-noble metals can be removed along with corrosion products using a mechanical filtration system, which "consists largely of a high surface area mechanical filter, likely a nickel mesh, to promote deposition of suspended, undissolved fission and corrosion products," ~~stated~~ according to Holcomb *et al.* [35]. The filter is manufactured from porous metal and ? located on a recirculating side stream of the side. The filter has limited capacity, needs periodic replacement, and the dose rate on the used filter is huge due to the undissolved FPs and residual fuel salt remaining on the filter [36].

The historic MSRE program provided basic information on the design and performance of the large mechanical filter. Figure 2.4 shows the piping layout of the filter, storage, and processing tanks. The filter pressure vessel is made of high-nickel alloy (Inconel) and accommodates 40-$\mu m$ pore size sintered Inconel fibers. This large molten salt filter had a total filtering area of $0.8m^2$ and was designed to filter approximately 1 kg of the molten salt per minute, but the removal efficiency has never been reported. Also, the design of the filter, the filter holder, and the remotely operated equipment for the filter replacement for commercial-scale MSR designs presents a significant engineering challenge [36].

In this work, we assumed ideal and an constant separation efficiency in the filtering system. However, in the future, a physics-driven mathematical formula can be used when the experimental data or analytical model ~~will be~~ becomes available.

### 2.1.3 Fuel chemical processing facility

In addition to noble gases, noble metals, and semi-noble metals, the fuel salt reprocessing system must extract other FPs such as the lanthanides. These absorb ~~fewer~~ neutrons than $^{135}$Xe, but their removal is crucial to guarantee normal operation. Meanwhile, lanthanides have relatively high solubility in the carrier salt and must be removed by chemical extraction.

maybe not the right use of "Meanwhile"

Perhaps "Unfortunately"

In thorium-fueled MSR designs, $^{232}$Th in the fuel salt absorbs thermal neutrons and pro-

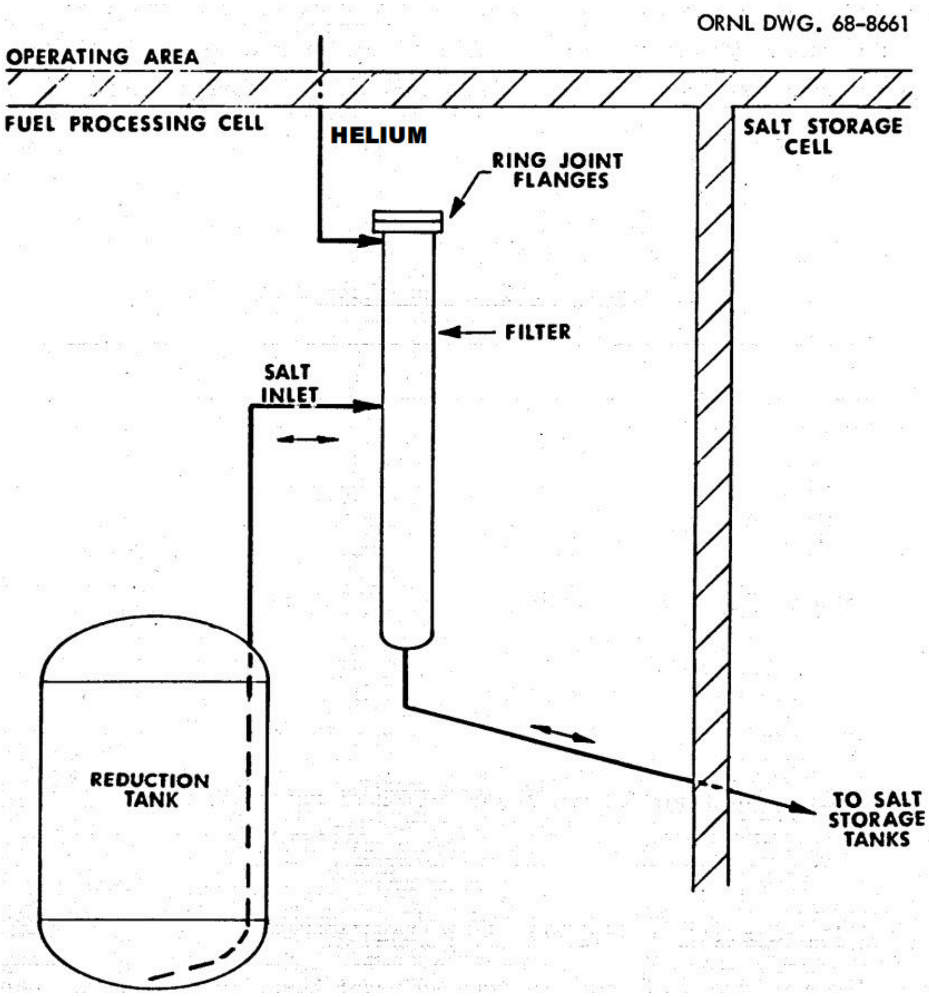
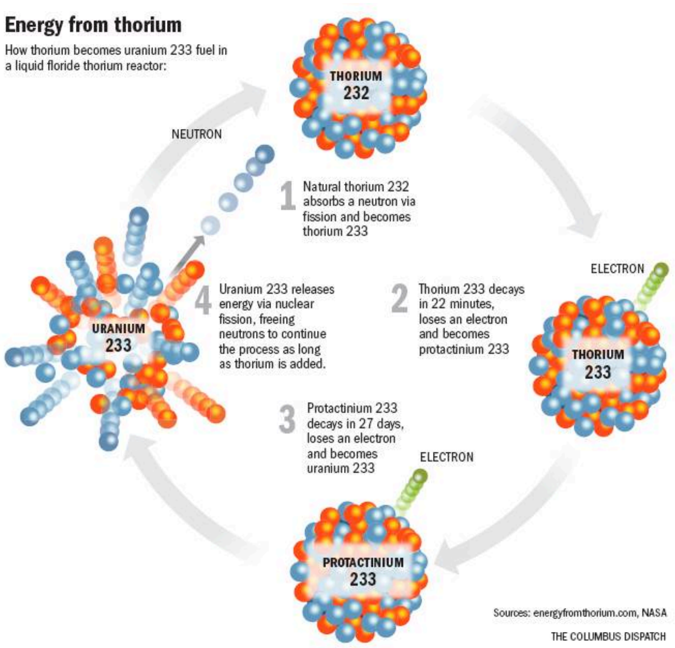


Figure 2.4: Schematic flow diagram of the large molten salt mechanical filter designed and operated during the MSRE (reproduced from Lindauer *et al.* [37]).

duces $^{233}$Pa, which then decays into the fissile $^{233}$U (Figure 2.5). Protactinium presents a challenge since it has a large absorption cross section in the thermal energy spectrum. Accordingly, $^{233}$Pa is continuously removed from the fuel salt into a ~~protactinium decay~~ tank in which ~~to allow~~ $^{233}$Pa decays ~~to~~ decay to $^{233}$U without poisoning the reactor. This feature allows the thorium-fueled MSR to avoid neutron losses to protactinium, keeps FPs on a trace level, and increases the efficiency of $^{233}$U breeding.


Energy from thorium
How thorium becomes uranium 233 fuel in a liquid floride thorium reactor:
THORIUM 232
NEUTRON
1 Natural thorium 232 absorbs a neutron via fission and becomes thorium 233
ELECTRON
2 Thorium 233 decays in 22 minutes, loses an electron and becomes protactinium 233
THORIUM 233
URANIUM 233
4 Uranium 233 releases energy via nuclear fission, freeing neutrons to continue the process as long as thorium is added.
3 Protactinium 233 decays in 27 days, loses an electron and becomes uranium 233
ELECTRON
PROTACTINIUM 233
Sources: energyfromthorium.com, NASA
THE COLUMBUS DISPATCH

nice figure!

Figure 2.5: Production of $^{233}$U from $^{232}$Th (reproduced from Sorensen [38]).

Many authors report that a liquid-liquid reductive extraction process is the best option for removing protactinium and soluble FPs from molten fluoride salts [39, 40, 41]. In that process, the protactinium or lanthanides can be selectively stripped from the salt into liquid bismuth due to different chemical potentials. Moreover, the MSRE experience indicated that the extraction could be carried out rapidly and continuously [42].

The principal scheme of the MSBR reprocessing facility concept is shown in Figure 2.6. The fuel salt is first temporarily stored for cooling and decay of the shortest-lived fission products, then it is directed to the primary fluorinator. There, most of the uranium is removed by fluorination to $UF_6$. After that, the salt is routed to an extraction column where it is combined with a mixture containing metallic bismuth, lithium, and thorium reductants. The remaining uranium and protactinium are reductively extracted to a bismuth solution, leaving a salt that only contains fission products dissolved in carrier salt (base composition LiF-$BeF_2$-$ThF_4$). The salt then goes through a reduction column where $UF_6$ is reduced to $UF_4$, preparing it for return to the reactor. $BeF_2$ and $ThF_4$ are also added and all residual bismuth is removed from the salt. After a final cleanup step and valence adjustment, the purified salt returns to the reactor [43, 38].

The bismuth, accommodating some uranium and protactinium, is routed to a hydrofluorination column where metallic solutes in the bismuth are oxidized into their fluoride forms in the presence of a decay salt[1]. The decay salt, containing $UF_4$, $PaF_4$, and $ThF_4$, passes into a decay tank where $^{233}Pa$ decays to $^{233}U$. The uranium generated by protactinium decay is removed through fluorination to $UF_6$ and directed to the reduction column to refuel the purified fuel salt. A hydrofluorinator and a fluorinator can remove approximately 95% of the uranium from the stream [30].

reverse the order of clauses in this sentence to increase clarity & brevity.

The fully processed salt, on its way back to the reactor, has uranium added from the protactinium decay tank at the rate required to maintain or adjust the uranium concentration in the reactor (and, consequently, control the reactivity). Adding fissile material is performed by sparging the salt with $UF_6$ and hydrogen to produce $UF_4$ in the salt and HF gas [30].

After these separation steps, the fuel salt stream from the protactinium isolation system contains only traces of protactinium and uranium but contains practically all of the rare earths. A fraction of this salt stream is redirected to a reductive extraction process for removing rare earths. The principal scheme of a rare earth removal system is shown in Figure 2.7. A molten salt flow that contains rare earth fluorides is fed to the center of an extraction column. The salt flows countercurrent to a liquid bismuth stream, which contains thorium and lithium. In the upper part of the column, the rare earths are reduced and

[1] The decay salt contains $UF_4$, $PaF_4$, $ThF_4$ and FPs. Uranium produced after $^{233}Pa$ decay is extracted and directed back into the reactor. Decay salt is the precursor for the waste salt as it was periodically discarded every 220 days [30].

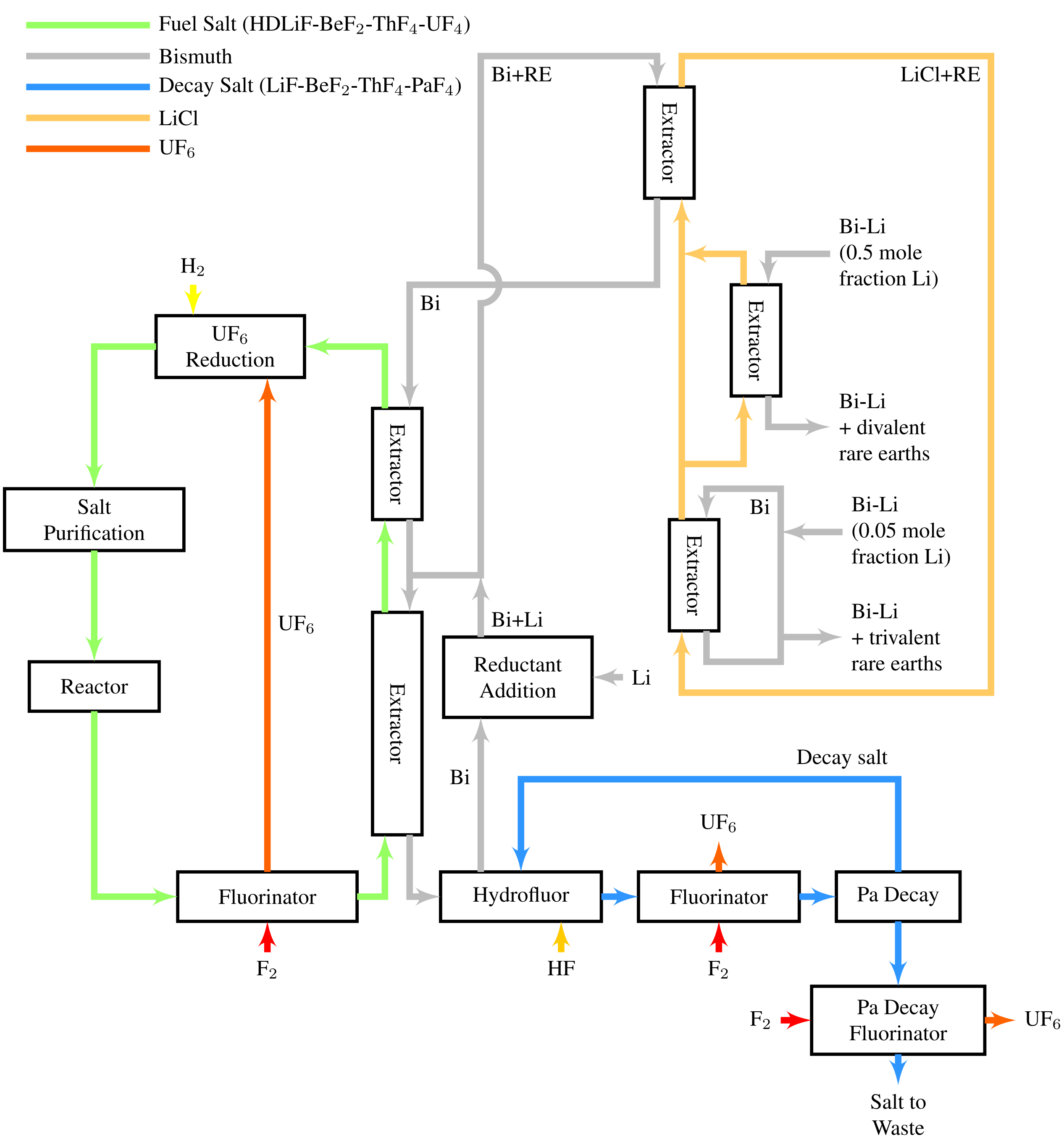


Figure 2.6: Simplified block diagram of chemical processing scheme for a single-fluid MSBR (reproduced from Sorensen [38]). *RE* represents the rare earth elements extracted from the salt.

transferred to the downflowing liquid metal stream. Below the feed point, the rare earth concentration is increased in the salt and metal streams in order to produce a concentration high enough for disposal [39].

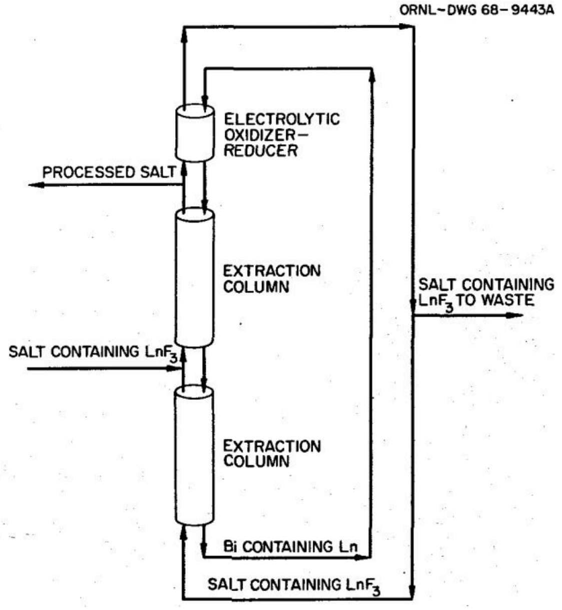


Figure 2.7: Rare earth removal from a fuel salt by reductive extraction (reproduced from Briggs *et al.* [39]).

While it is out of the scope of this work to derive the accurate chemistry-based mathematical formula for rare earths and protactinium separation efficiency, this work seeks to provide a flexible tool that is able to simulate chemical processes in significant detail concerning vital system design parameters.

## 2.2 Serpent overview

Serpent is a continuous-energy Monte Carlo neutronics software capable of solving the neutron transport problem by tracking individual neutrons within the problem geometry and using the stochastic method to determine the chain of events for each neutron [11]. Serpent is under active development at the VTT Technical Research Centre of Finland since 2004, where it was initially conceived as a tool to simplify group constant generation in a high-fidelity Monte Carlo environment. Serpent is now ~~a~~ widely used ~~transport code, used~~ by more than 500 registered individuals in 155 organizations located in 37 countries around the world. The burnup calculation capability in Serpent is based on built-in calculation routines without using any external solvers. A restart feature enables fuel shuffling simulation or applying any modifications to the input by dividing the calculation into several parts, which is crucial for online reprocessing simulations.

The latest version, Serpent 2, supports advanced geometries and has advanced burnup capabilities, including online refueling capabilities that are necessary for neutronic computations of pebble-bed reactors and liquid-fueled MSRs [17]. Unfortunately, built-in online refueling features are still under active development, undocumented, and the discussion forum for Serpent users is the only useful source of information at the moment [17]. Additionally, multi-physics simulations using Serpent 2 have been demonstrated, including calculations with thermal-hydraulics, CFD, and fuel performance codes [44].

This has appeared before.

Serpent 2 can be effectively run in parallel on computer clusters and multi-core workstations. Parallelization is handled by thread-based OpenMP, which enables all processors to use shared memory space. Calculations can be divided into several nodes by distributed-memory Message Passing Interface (MPI) parallelization. Serpent 2 is an improvement upon Serpent 1 and contains a complete redesign of memory management using hybrid OpenMP [45] + MPI parallelization. This hybrid parallelization is substantial for depletion calculations using computer clusters with multiple nodes and allows us to achieve significant speed-up in depletion calculations on computer clusters with more than 4,000 cores [11].

Simulations herein were performed using Serpent 2 version 2.1.31 on both the National Center for Supercomputing Applications' Blue Waters and Idaho National Laboratory's Falcon supercomputers. The JEFF-3.1.2 [46] and ENDF/B-VII.1 [47] libraries provided nuclear data for all calculations in this dissertation.

Is Falcon spelled FALCON or Falcon?

"my" ?

## 2.3 Simulation tool design and capabilities

The first version of the SaltProc tool for calculating MSR fuel composition evolution, taking into account an online reprocessing system, was developed in 2018 as a part of the M.S. thesis [10, 48]. The tool was designed to expand Serpent 2 depletion capabilities for modeling liquid-fueled MSRs with online fuel reprocessing systems. SaltProc v0.1 uses HDF5 [49] to store data and uses the PyNE Nuclear Engineering Toolkit [50] for Serpent 2 output file parsing and nuclide naming. 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 at the end of the specific "cycle time"[2]. 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) [51]. The SaltProc v0.1 architecture and the principal structure were not designed for flexible implementation of sophisticated online reprocessing systems, including realistic variable extraction efficiencies.

In

For the current work, SaltProc v0.1 was completely refactored using Object-Oriented Programming (OOP) to create a comprehensive generic tool to realistically model complex MSR fuel reprocessing systems while taking into account variable extraction efficiencies, time-dependent core geometry, and the mass balance between the core and the reprocessing plant.

### 2.3.1 Software architecture

**This section needs to be updated when the structure will be finalized.**

The SaltProc v1.0 Python toolkit couples directly with Serpent 2 input and output files, to couple the reprocessing system to depletion calculation. Python 3 OOP standard features

[2] The MSBR program defined "cycle time" as the time required to remove 100% of a target nuclide from a fuel salt [30].

are used to create a flexible, user-friendly tool with great potential for further improvement and collaboration. Figure 2.8 shows the SaltProc v1.0 class structure which includes 4 main classes:

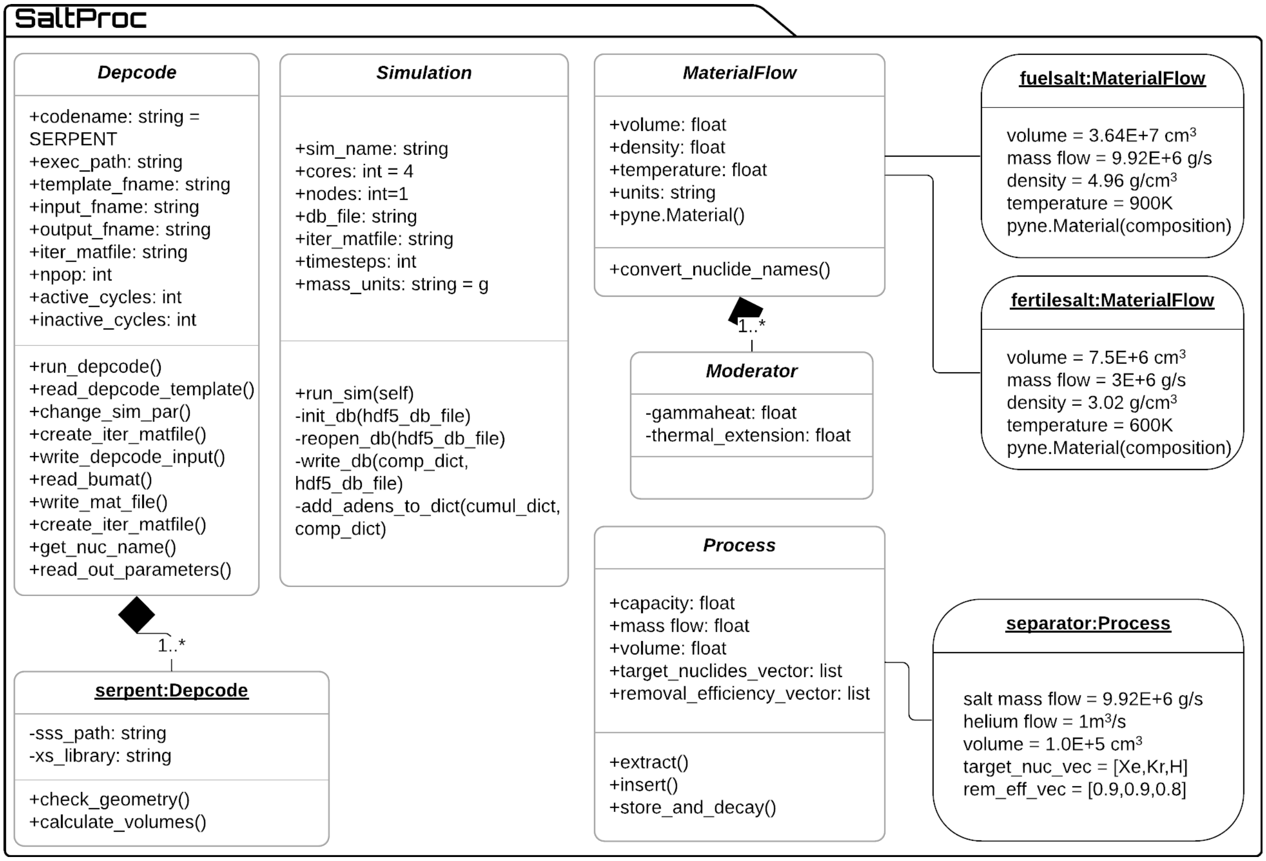


Figure 2.8: SaltProc v1.0 Python package class diagram in UML notation with examples of object instances.

**Depcode.** *Depcode* class 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.** *Simulation* class runs a Serpent depletion step, creates and writes an 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 (Figure 2.9). All instances of this class contain an isotopic composition vector stored in PyNE Material object, mass flow rate, temperature, density, volume, and void fraction. Existing PyNE Material capabilities convert the units of the isotopic composition vector (e.g., from the 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). 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.

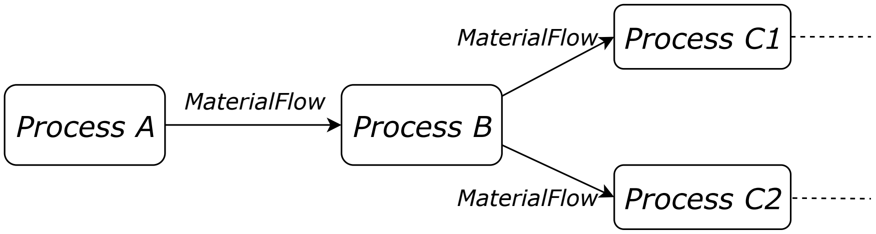


Figure 2.9: Schematic for passing material data between fuel processing system components.

Add a figure with configured processes & flows.
( core, fuel, sparger.... )

**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 process-specific parameters. The 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).

Such a class structure provides outstanding flexibility in simulating various MSR fuel processing system designs. I created 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, nickel filter) object examples to help a user to create a model of a desired reprocessing scheme quickly. At runtime, the user should connect *Process* objects in series, parallel, or both with

Only for directed graphs?

Anyway, explain how this problem is a directed graph.

*MaterialFlow* objects to form a comprehensive reprocessing system. To make the reprocessing system definition self-explanatory and straightforward, I employed a standardized graph description language, *dot*, which is widely used in Computer Science [52]. The reprocessing plant structure described with *dot* can be simply plotted using Graphviz [53] and those plots can be used for analysis, optimization, and publication purposes. The user also had the flexibility to create custom objects with desired attributes and methods and contribute back to the code package using GitHub (https://github.com/arfc/saltproc).

no need to capitalize

### 2.3.2 Tool flowchart

Consider splitting up these sentences into depletion software parameters and reprocessing system parameters.

this seems like an output... not an input.

What exactly do you mean by this?

Figure 2.10 illustrates the online reprocessing simulation algorithm coupling SaltProc v1.0 and Serpent. A *json*-compatible user input file for SaltProc contains parameters such as paths to the depletion software executable, neutron population and number of criticality cycles, depletion history, total heating power, and a list of files with the core geometry definition. To perform a depletion step, SaltProc v1.0 reads a user-defined Serpent template file. This file contains input parameters such as the path to a nuclear data library, material isotopic composition at startup, burnup calculation parameters, and boundary conditions. SaltProc v1.0 fills in the template file and runs Serpent single-step depletion.

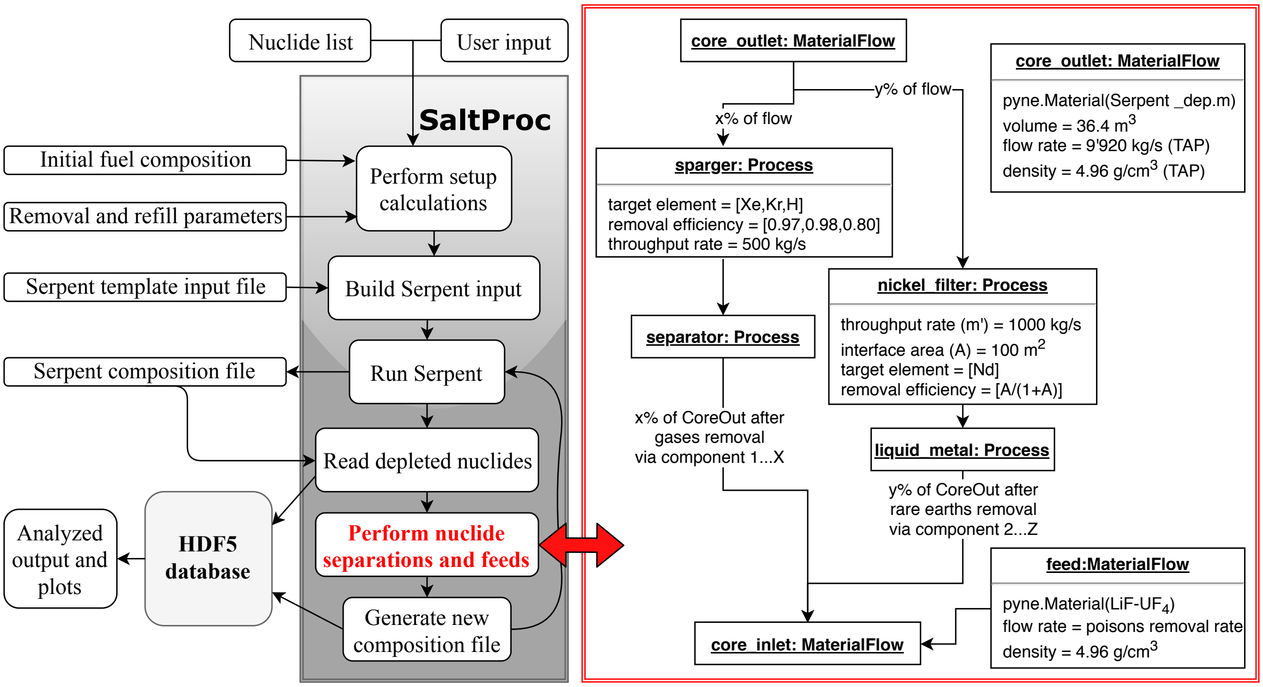
maybe give symbolic names to all of these ($\rho$, T, $\vec{m}$, $|\vec{m}|$, $m_i$, V, $\varepsilon$ ...)

"connected in series"

After the depletion calculation, SaltProc v1.0 reads the depleted fuel composition file into the *MaterialFlow* object (*core_outlet* in Figure 2.10). 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, if 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~~ flows sequentially between *Processes*, and each *Process* ~~is removing~~ removes 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, the resulting isotopic composition after reprocessing is stored in the HDF5 database and dumped in a new composition file for the next Serpent depletion

composition math: cite PyNE?

run. SaltProc v1.0 also stores the isotopic composition before reprocessing and waste stream from each fuel processing component in the HDF5 database.



Nuclide list
User input
SaltProc
Initial fuel composition
Removal and refill parameters
Perform setup calculations
Serpent template input file
Build Serpent input
Serpent composition file
Run Serpent
Read depleted nuclides
Analyzed output and plots
HDF5 database
Perform nuclide separations and feeds
Generate new composition file
core_outlet: MaterialFlow
x% of flow
y% of flow
sparger: Process
target element = [Xe,Kr,H]
removal efficiency = [0.97,0.98,0.80]
throughput rate = 500 kg/s
core_outlet: MaterialFlow
pyne.Material(Serpent _dep.m)
volume = 36.4 m$^3$
flow rate = 9'920 kg/s (TAP)
density = 4.96 g/cm$^3$ (TAP)
nickel_filter: Process
throughput rate (m') = 1000 kg/s
interface area (A) = 100 m$^2$
target element = [Nd]
removal efficiency = [A/(1+A)]
separator: Process
x% of CoreOut after gases removal via component 1...X
liquid_metal: Process
y% of CoreOut after rare earths removal via component 2...Z
feed:MaterialFlow
pyne.Material(LiF-UF$_4$)
flow rate = poisons removal rate
density = 4.96 g/cm$^3$
core_inlet: MaterialFlow

Figure 2.10: Flow chart for the SaltProc v1.0 Python package.

For a more general case with multiple concurrent extraction processes, ~~a~~ separate *MaterialFlow* ~~object is~~ objects are 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% throughout a right branch). The total mass and isotopic composition vector for each *MaterialFlow* object are 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 compensates for mass losses ~~the loss of mass~~ in the Processes ~~separation facilities~~ and keeps the fuel salt mass in the ~~a~~ primary loop constant.

Mass flow rate has dimensions [mass/time] but your variable has dimensions of %. Perhaps "mass flow branching ratio" or "mass flow branching percentage"

The UML class diagram (Figure 2.8) allows the user to model a complex, multi-zone, multi-fluid MSR operation and is sufficiently general to represent myriad reactor systems. SaltProc v1.0 only stores and changes 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 v1.0 is an open-source tool (but a user needs Serpent 2.1.31 installed to use SaltProc v1.0) available on GitHub. It leverages unit tests and continuous ~~tests~~ crucial for ~~sustainable development~~ Software sustainability [54, 55]. The documentation is automatically generated using Sphinx [56] and available here: *https://arfc.github.io/saltproc/*. In summary, the development approach of SaltProc v1.0 is focused ~~on producing~~ a generic, flexible and expandable tool ~~to give~~ that extends 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.

make this its own sentence

<integration

use href or url command.

### 2.3.3 Reactivity control module

Simulation of specific MSR concepts requires changing the reactor core geometry during lifetime-long operation modeling. For instance, the TAP concept aims to increase the core lifetime by using continuous fresh fuel feeds, removing FPs, and reconfiguring moderator rod assemblies to compensate for negative reactivity insertion due to fissile material burnup. The concept proposes maintaining reactivity in the long term by replacing stationary moderator assemblies with ~~more highly populated~~ denser lattices to increase the moderator-to-fuel ratio [29]. SaltProc v1.0 can switch from one file containing the core geometry to another core geometry (e.g., with larger moderator-to-fuel ratio) if the effective multiplication factor, $k_{eff}$, falls below a specific limit (e.g., 1.002). This unique capability allows SaltProc v1.0 to analyze the fuel cycle performance of any liquid-fueled MSR system, including advanced designs with a moving moderator.

An ordinary reader may need more detail.

## 2.4 Concluding remarks

~~In this chapter, the~~ This chapter presented an overview of ~~the~~ fuel salt reprocessing ~~plant has first been presented~~ in MSRs. It described various components of the plant and the physical or chemical mechanism responsible for neutron poison extraction from the salt. General core physics aspects and Serpent 2 depletion software capabilities have then been discussed. ~~I~~ It also introduced SaltProc, a Python package developed and used to simulate continuous feeds and removals in various MSR designs.

In the following chapters, SaltProc v1.0 will be demonstrated and validated for two liquid-fueled MSR designs.

# Chapter 3

# Tool demonstration: Molten Salt Breeder Reactor

This chapter describes the fuel cycle analysis of the MSBR obtained using the open-source Python package, SaltProc. The development was initially started as a part of my master thesis [48] in 2017. ~~In this~~ This effort, for verification purposes, ~~I~~ assumed ideal extraction efficiency (e.g., 100% of target isotope mass extracted) because all results available in the literature also rely on this assumption.

The main results presented in this chapter have been published in: 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*, 128 (2019): 366–379. The high-fidelity, full-core MSBR model has been presented at the 2017 American Nuclear Society (ANS) Winter Meeting in Washington, D.C. The fuel salt composition evolution has been presented at the 2018 Blue Waters Symposium in Sunriver, OR. The obtained results relevant to MSBR analysis have been compared against those obtained by Benjamin R. Betzler and colleagues for a simplified unit cell model, adopting the in-house code ChemTriton.

I think you can cite this.

cite

## 3.1 Introduction

The thorium-fueled MSBR was developed in the early 1970s by ORNL, specifically to explore the promise of the thorium fuel cycle, which uses natural thorium fertile feed material instead of enriched uranium fissile fuel. With continuous fuel reprocessing, the MSBR realizes the advantages of the thorium fuel cycle because the $^{233}$U bred from $^{232}$Th is almost instantly[1] recycled back into the

[1] The fertile $^{232}$Th is transmuted into the $^{233}$Th after capturing a neutron. Next, this isotope decays to the $^{233}$Pa ($\tau_{1/2}$=21.83m), which finally decays to the $^{233}$U ($\tau_{1/2}$=26.967d).

"good" feels like a value judgement. "beneficial"?

core [57]. The chosen fuel salt, LiF-BeF$_2$-ThF$_4$-UF$_4$, has a melting point of 499°C, low vapor pressure at operating temperatures, and good flow and heat transfer properties [30].

In this work, we analyzed the MSBR neutronics and fuel cycle to establish its equilibrium core composition. Additionally, we compared predicted operational and safety parameters of the MSBR at both the initial and equilibrium states to characterize the evolution of its safety case over time. Moreover, these depletion simulations determined the appropriate $^{232}$Th feed rate for maintaining criticality and enabled analysis of the overall MSBR fuel cycle performance. Finally, the benefits of online fission product removal in the thermal spectrum MSBR were identified.

## 3.2 Molten Salt Breeder Reactor design and model description

diagram of all this? Mention it.

The MSBR vessel has a diameter of 680 cm and a height of 610 cm. It contains a molten fluoride fuel-salt mixture that generates heat in the active core region and transports that heat to the primary heat exchanger by way of the primary salt pump. In the active core region, the fuel salt flows through channels in moderating and reflecting graphite blocks. Fuel salt at 565°C enters the central manifold at the bottom via four 40.64-cm-diameter nozzles and flows upward through channels in the lower plenum graphite. The fuel salt exits at the top at about 704°C through four equally spaced nozzles, which connect to the salt-suction pipes leading to primary circulation pumps. The fuel salt drain lines connect to the bottom of the reactor vessel inlet manifold.

Figure 3.1 shows the configuration of the MSBR vessel, including the "fission" (zone I) and "breeding" (zone II) regions inside the vessel. The core has two radial zones bounded by a solid cylindrical graphite reflector and the vessel wall. The central zone, zone I, in which 13% of the volume is fuel salt and 87% is graphite, is composed of 1,320 graphite cells, 2

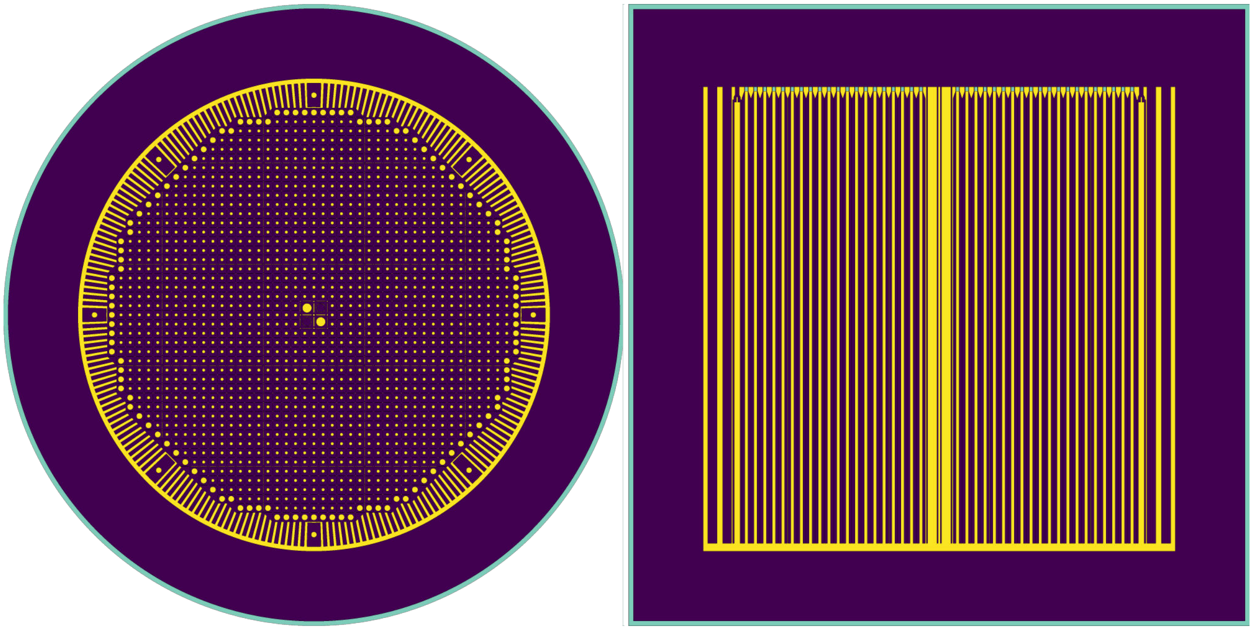


Figure 3.1: *XY* (left) and *XZ* (right) views of a Serpent MSBR model (reproduced from Rykhlevskii *et al.* [51]).

graphite control rods, and 2 ~~safety~~[2] emergency shutdown rods. The under-moderated zone, zone II, in which 37% of the volume is fuel salt and 63% is graphite, and radial reflector, surrounds the zone I core region and serves to diminish neutron leakage. Zones I and II are surrounded radially and axially by fuel salt (Figure 3.2); this space for fuel is necessary for the injection and flow of molten salt.

Since reactor graphite experiences significant dimensional changes due to neutron irradiation, the reactor core was designed for periodic replacement. Based on the experimental irradiation data from the MSRE, the core graphite lifetime is about 4 years, and the reflector graphite lifetime is 30 years [30].

The core design also has eight symmetric graphite slabs with a width of 15.24 cm in zone II, one of which is illustrated in Figure 3.2. The holes in the centers are for the core lifting rods used during the core replacement operations. These holes also allow a portion of the fuel salt to flow to the top of the vessel for cooling the top head and axial reflector. Figure 3.2 also shows the 5.08-cm-wide annular space between the removable core graphite in zone II-B

[2] ~~These rods needed for emergency shutdown only.~~

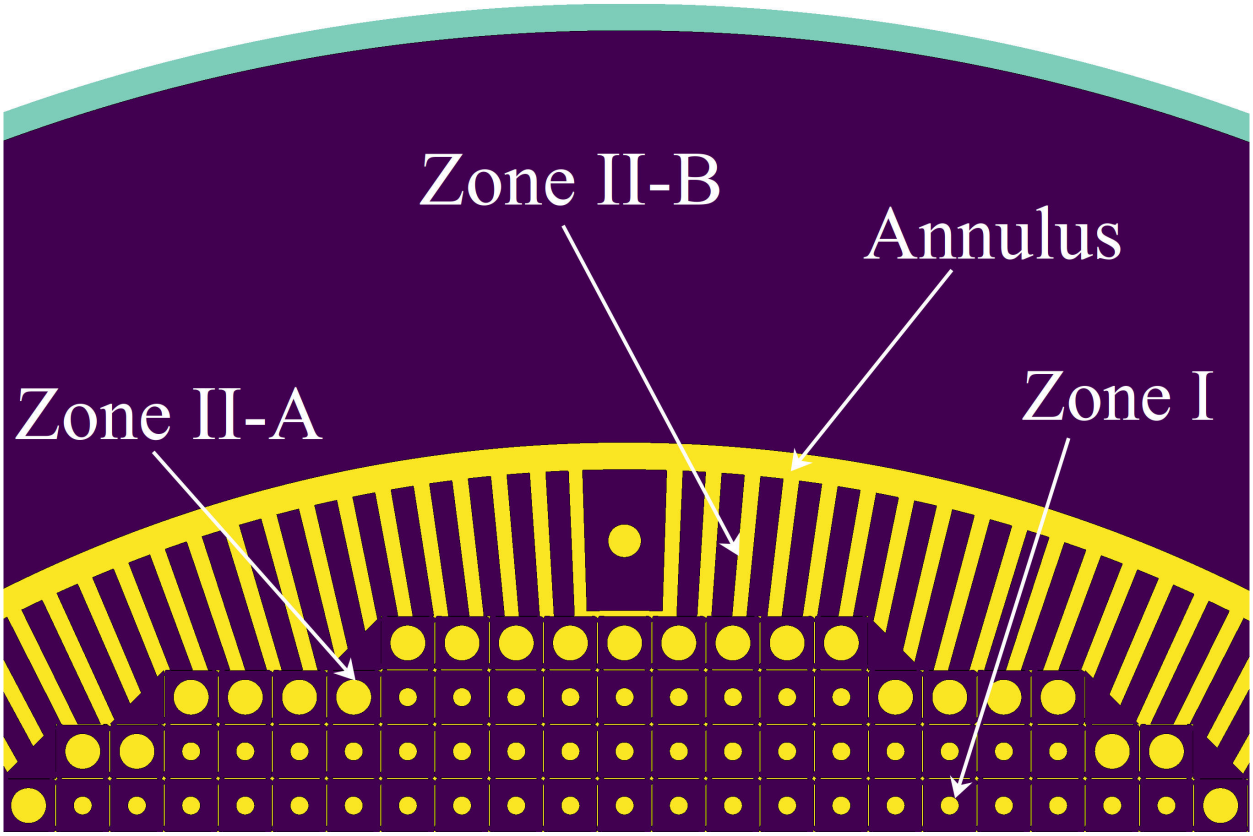


Figure 3.2: Detailed view of the MSBR two-zone model. Yellow represents fuel salt, purple represents graphite, and aqua represents the reactor vessel (reproduced from Rykhlevskii *et al.* [51]).

and the permanently mounted reflector graphite. This annulus consists entirely of fuel salt, provides space for moving the core assembly, helps compensate for the elliptical dimensions of the reactor vessel, and serves to reduce the damaging flux at the surface of the graphite reflector blocks.

$^{135}$Xe is a strong neutron poison, and some fraction of this gas is absorbed by graphite during MSBR operation. ORNL calculations showed that for unsealed commercial graphite with a helium permeability of $10^{-5}$ cm$^2$/s, the calculated poison fraction is less than 2% [30]. This parameter can be improved by using experimental graphite types or by applying sealing technology. The effect of the gradual poisoning of the core graphite with xenon is out of the scope if this work.

outside or "beyond"

do you mean 2% of all generated $^{135}$Xe is trapped in the graphite? All poisons?

### 3.2.1 Core zone I

The central region of the core, called zone I, is made up of graphite elements, each 10.16cm× 10.16cm×396.24cm. Zone I has 4 channels for control rods: two for graphite rods, which both regulate and shim during normal operation, and two for backup safety rods consisting of boron carbide clad to assure sufficient negative reactivity for accidents.

These graphite elements have a mostly rectangular shape with lengthwise ridges at each corner that leave space for salt flow elements. Various element sizes reduce the peak damage flux and power density in the center of the core to prevent local graphite damage. Figure 3.3 shows the elevation and plan views of graphite elements of zone I [30] and their Serpent model [58].

### 3.2.2 Core zone II

Zone II, which is undermoderated, surrounds zone I. Combined with the bounding radial reflector, zone II serves to diminish neutron leakage. Two kinds of elements form this zone: large-diameter fuel channels (zone II-A) and radial graphite slats (zone II-B).

Zone II has 37% fuel salt by volume, and each element has a fuel channel diameter of 6.604cm. The graphite elements for zone II-A are prismatic, with elliptical dowels running axially between the prisms. These dowels isolate the fuel salt flow in zone I from that in zone II. Figure 3.4 shows the shapes and dimensions of these graphite elements and their Serpent model. Zone II-B elements are rectangular slats spaced far enough apart to provide the 0.37 fuel salt volume fraction. The reactor zone II-B graphite 5.08cm-thick slats vary in the radial dimension (average width is 26.67cm) as shown in Figure 3.2. Zone II serves as a blanket to achieve the best performance: a high breeding ratio and a low fissile inventory. The harder neutron energy spectrum in zone II enhances the rate of thorium resonance capture relative to the fission rate, thus limiting the neutron flux in the outer core zone and reducing the neutron leakage [30].

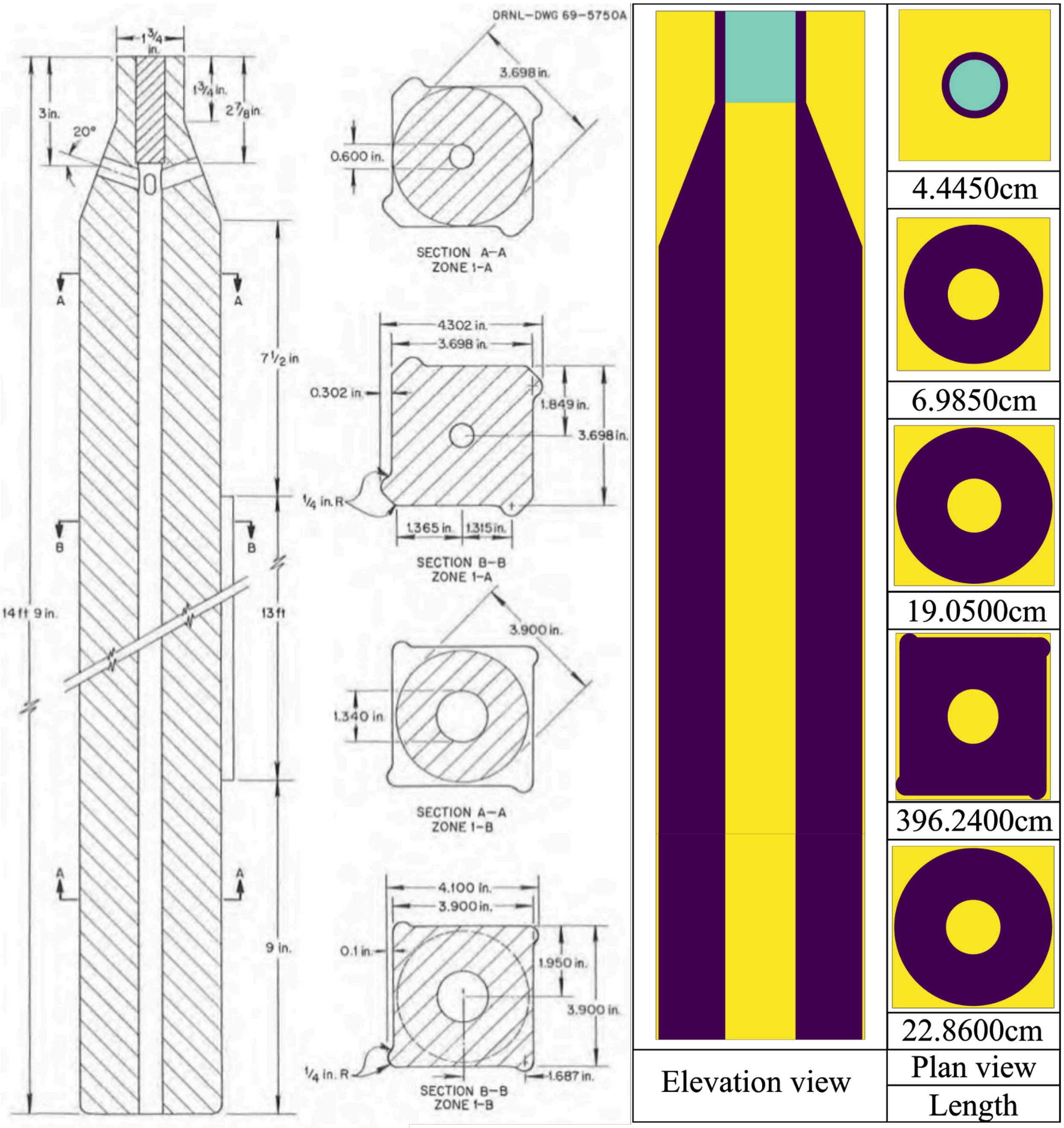


Figure 3.3: Graphite moderator elements for zone I: reference design (left) [30] and Serpent model (right) [58]. Yellow represents fuel salt, purple represents graphite, and aqua represents the reactor vessel ()reproduced from Rykhlevskii *et al.* [51]).

The sophisticated, irregular shapes of the fuel elements challenge an accurate representation of zone II-B. The suggested design [30] of zone II-B has eight irregularly-shaped graphite elements as well as dozens of salt channels. These graphite elements were simplified into "did not impact" or "did not affect"

right-circular cylindrical shapes with central channels. Figure 3.2 illustrates this core region in the Serpent model. The volume of fuel salt in zone II was kept exactly at 37% so this simplification unaffected the core neutronics. Simplifying the eight edge channels was the only simplification made to the MSBR geometry in this work.

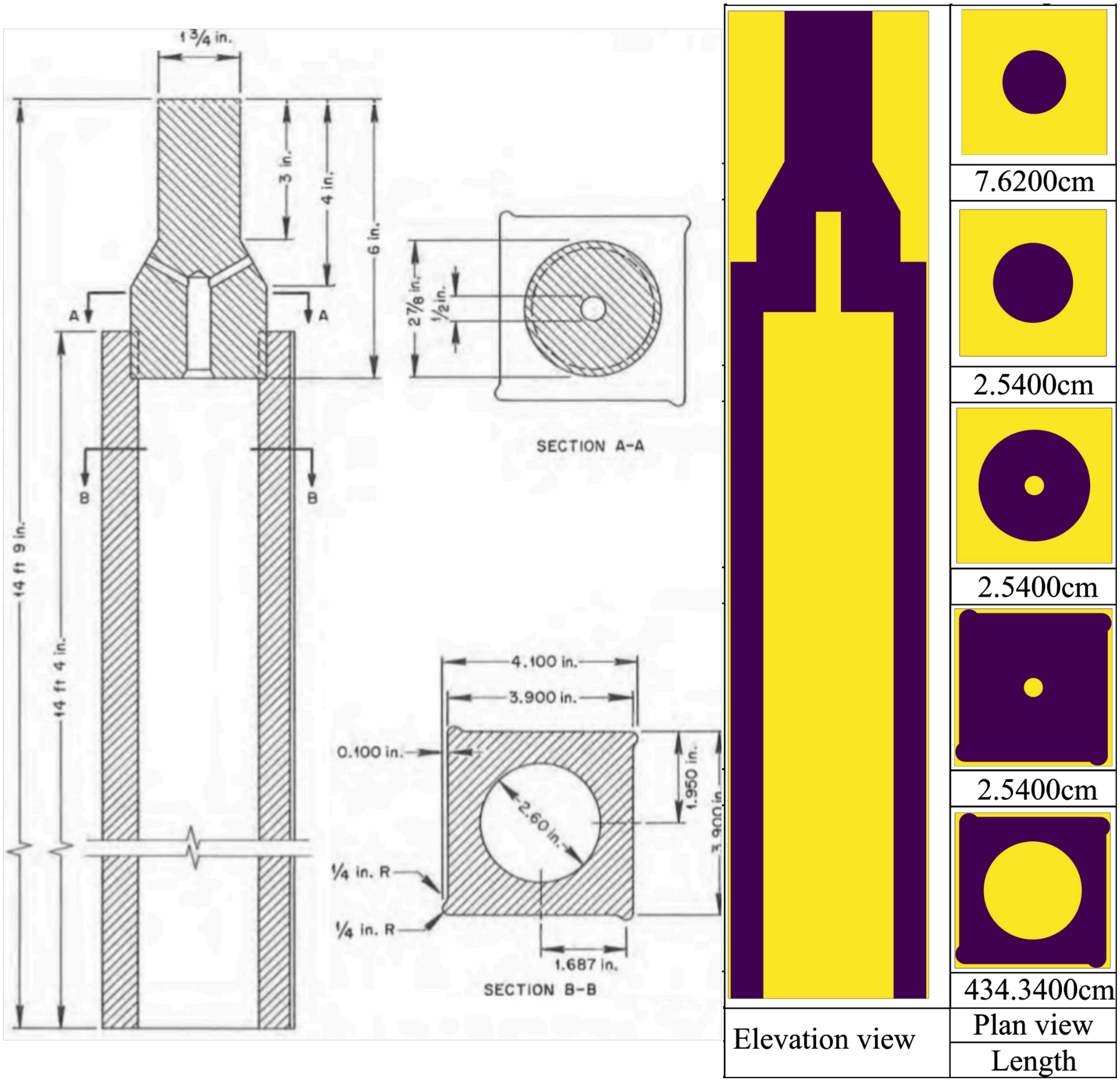


Figure 3.4: Graphite moderator elements for zone II-A: reference design (left) [30] and Serpent model (right) [58]. Yellow represents fuel salt, purple represents graphite, and aqua represents the reactor vessel (reproduced from Rykhlevskii *et al.* [51]).

### 3.2.3 Material composition and normalization parameters

"invented" "conceived"?

The fuel salt, reactor graphite, and modified Hastelloy-N are all materials created at ORNL specifically for the MSBR. The initial fuel salt used the same density (3.35 g/cm$^3$) and composition LiF-BeF$_2$-ThF$_4$-$^{233}$UF$_4$ (71.75-16-12-0.25 mole %) as the MSBR design [30]. The lithium in the molten salt fuel is fully enriched to 100% $^7$Li because $^6$Li is an extremely strong neutron poison and becomes tritium upon neutron capture.

confusing. The initial salt for the MSBR is the same as the MSBR? Something is missing. Do you mean that the salt you modeled is the same as the design?

This is impossible physically. In fuel cycle literature, we typically assume "five nines" (99.999%) or greater but not 100%

The specific temperature was fixed for each material and stays constant during ~~the~~ reactor operation. The isotopic composition of each material at the initial state was described in detail in the MSBR conceptual design study [30] and has been applied to the Serpent model without any modification. Table 3.1 is a summary of the major MSBR parameters used to inform the Serpent model [30].

Table 3.1: Summary of principal data for the MSBR (reproduced from Robertson *et al.* [30]).

| | |
|---|---|
| Thermal power | 2250 MW$_{th}$ |
| Electric power | 1000 MW$_e$ |
| Gross thermal efficiency | 44.4% |
| Salt volume fraction in central zone I | 0.13 |
| Salt volume fraction in outer zone II | 0.37 |
| Fuel salt inventory (Zone I) | 8.2 m$^3$ |
| Fuel salt inventory (Zone II) | 10.8 m$^3$ |
| Fuel salt inventory (annulus) | 3.8 m$^3$ |
| Total fuel salt inventory | 48.7 m$^3$ |
| Fissile mass in fuel salt | 1303.7 kg |
| Fuel salt components | LiF-BeF$_2$-ThF$_4$-$^{233}$UF$_4$ |
| Fuel salt composition | 71.75-16-12-0.25 mole% |
| Fuel salt density | 3.35 g/cm$^3$ |

consider right alignment? All this space is too much.

As mentioned in section 2.1, the MSBR design requires online reprocessing to "fully"? "completely"? remove neutron gaseous FPs (Xe, Kr) and noble metals (e.g., Se, Nb, and Mo) every 20 seconds. The $^{232}$Th in the fuel absorbs thermal neutrons and produces $^{233}$Pa, which then decays into the fissile $^{233}$U. Protactinium presents a challenge since it has a large absorption cross section in the thermal energy spectrum. Moreover, $^{233}$Pa left in the core ~~would produce~~ produces $^{234}$Pa and $^{234}$U, neither of which are useful as fuel. Accordingly, $^{233}$Pa is continuously removed from the fuel salt into a ~~protactinium decay~~ temporary storage tank to allow $^{233}$Pa to decay to $^{233}$U without the corresponding negative neutronic impact. The reactor chemical processing system must separate $^{233}$Pa from the molten salt fuel over 3 days, hold it while $^{233}$Pa decays into $^{233}$U, and return it to the primary loop. This feature allows the reactor to avoid neutron losses to protactinium, lowers in-core fission product inventory, and increases the efficiency of $^{233}$U breeding.

Table 3.2 summarizes a full list of nuclides and their cycle time used for modeling salt treatment and separations [30]. The removal rates vary among chemical elements in this reactor concept and dictate the necessary resolution of depletion calculations. If the depletion time intervals are 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 interval was selected for depletion calculations to correlate with the removal interval of $^{233}$Pa, and $^{232}$Th was continuously added to maintain the initial mass fraction of $^{232}$Th.

impact on what? Expand this sentence slightly for clarity.

this seems like it should be its own sentence.

Table 3.2: The cycle times for protactinium and fission product~~s~~ removal from the MSBR (reproduced from Robertson *et al.* [30]).

| Processing group | Nuclides | Cycle time (at full power) |
|---|---|---|
| Rare earths | Y, La, Ce, Pr, Nd, Pm, Sm, Gd | 50 days |
| | Eu | 500 days |
| Noble metals | Se, Nb, Mo, Tc, Ru, Rh, Pd, Ag, Sb, Te | 20 sec |
| Semi-noble metals | Zr, Cd, In, Sn | 200 days |
| Gases | Kr, Xe | 20 sec |
| Volatile fluorides | Br, I | 60 days |
| Discard | Rb, Sr, Cs, Ba | 3435 days |
| Protactinium | $^{233}$Pa | 3 days |
| Higher nuclides | $^{237}$Np, $^{242}$Pu | 16 years |

## 3.3 Fuel salt isotopic composition dynamics and equilibrium search

The SaltProc online reprocessing simulation package is demonstrated in four applications: (1) analyzing the MSBR neutronics and fuel cycle to find the equilibrium core composition and fuel salt depletion, (2) demonstrating that in a single-fluid two-region MSBR conceptual design the undermoderated outer core zone II works as a virtual "blanket", reduces neutron leakage, and improves breeding ratio due to neutron energy spectral shift, (3) studying operational and safety parameters evolution during MSBR operation, and (4) determining the effect of fission product removal on the core neutronics. This section discusses the first two applications.

Transition more smoothly into this paragraph. Clarify that these are Serpent parameters (not SaltProc)

The neutron population per cycle and the number of active/inactive cycles were chosen to compromise between reasonable uncertainty for a transport problem ($\leq$ 15 pcm[3] for effective multiplication factor) and computational time. The MSBR depletion and safety parameter computations were performed on 64 Blue Waters XK7 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 fuel salt depletion during 60 years of operation was approximately 9,900 node-hours (18 core-years.)

At first glance this footnote seems like pcm cubed. Consider stating just "$\leq$15pcm or equivalently $10^{-5}$ Δk/k ~~stuff~~ stuff"

### 3.3.1 Effective multiplication factor dynamics

Figures 3.5 and 3.6 show the effective multiplication factors obtained using SaltProc v0.1 and Serpent. The effective multiplication factors were calculated after removing fission products listed in Table 3.2 and adding the fertile material at the end of each depletion step (3 days). The effective multiplication factor fluctuates significantly as a result of the batch-wise nature of this online reprocessing strategy.

First, Serpent calculates the effective multiplication factor for the beginning of the cycle.

[3] 1 $pcm = 10^{-5}\Delta k_{eff}/k_{eff}$

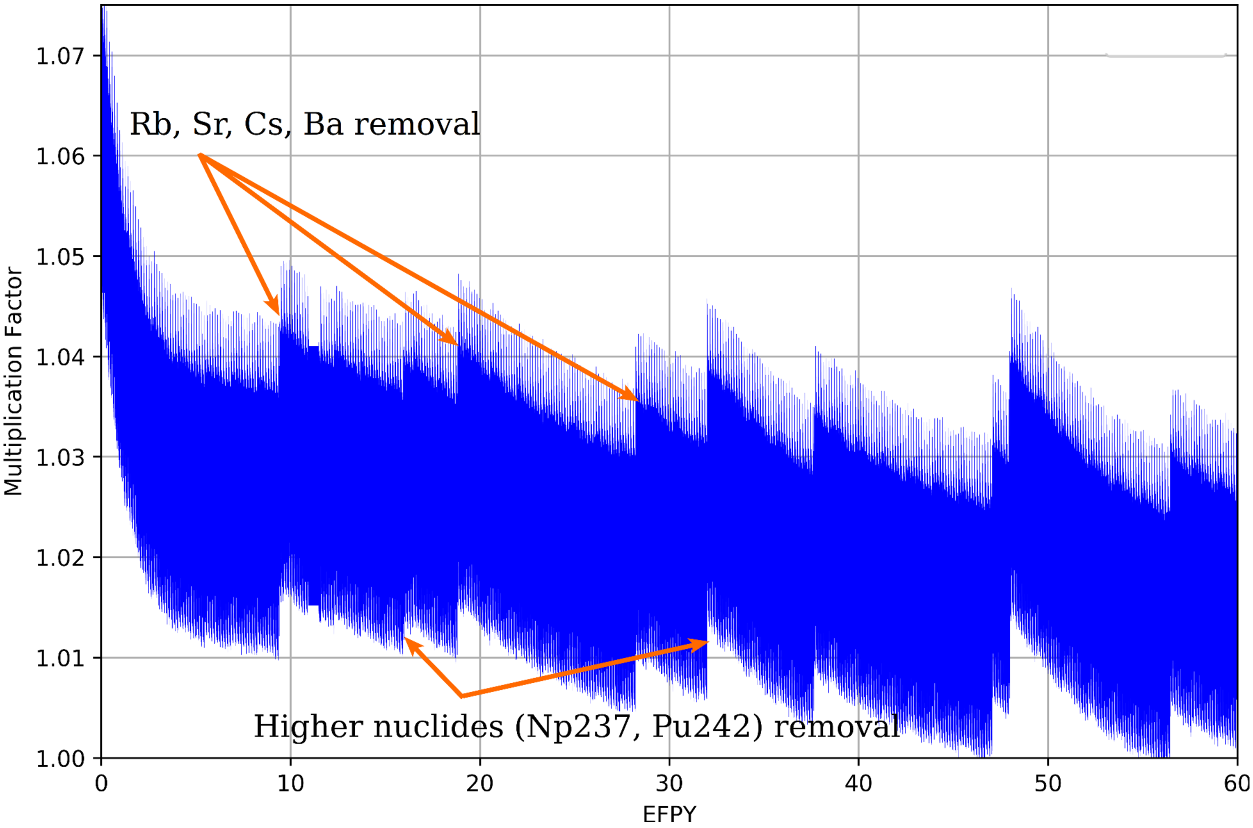


Figure 3.5: Effective multiplication factor dynamics for full-core MSBR model over a 60-year reactor operation lifetime (reproduced from Rykhlevskii *et al.* [51]).

Next, it computes the new fuel salt composition at the end of a 3-day depletion. The corresponding effective multiplication factor is much smaller than the previous one. Finally, Serpent calculates $k_{eff}$ for the depleted composition after applying feeds and removals. The $k_{eff}$ increases accordingly since major reactor poisons (e.g., Xe, Kr) are removed, while fresh fissile material ($^{233}$U) from the protactinium decay tank is added.

Additionally, the presence of rubidium, strontium, cesium, and barium in the core are disadvantageous to reactor physics. Overall, the effective multiplication factor gradually decreases from 1.075 to ≈1.02 at equilibrium after approximately 6 years of irradiation.

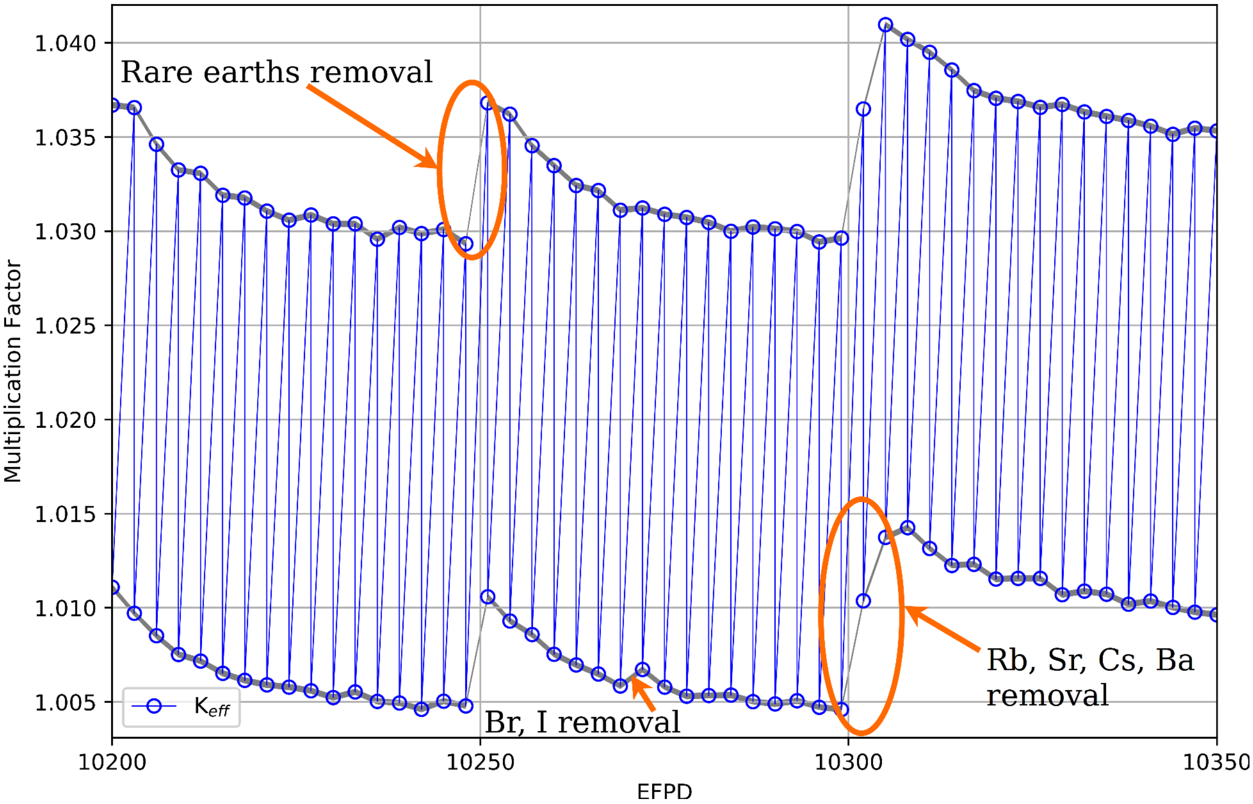


Figure 3.6: Zoomed effective multiplication factor for a 150-EFPD time interval (reproduced from Rykhlevskii *et al.* [51]).

### 3.3.2 Fuel salt composition dynamics

The analysis of the fuel salt composition evolution provides more comprehensive information about the equilibrium state. Figure 3.7 shows the number densities of major nuclides, which have a strong influence on the reactor core physics. The concentration of $^{233}$U, $^{232}$Th, $^{233}$Pa, and $^{232}$Pa in the fuel salt change insignificantly after approximately 2500 days of operation. In particular, the $^{233}$U number density fluctuates by less than 0.8% between 16 and 20 years of operation. Hence, a quasi-equilibrium state was achieved after 16 years of reactor operation.

In contrast, a wide variety of nuclides, including fissile isotopes (e.g., $^{235}$U) and non-fissile strong absorbers (e.g., $^{234}$U), kept accumulating in the core. Figure 3.8 demonstrates the production of fissile isotopes in the core. At the end of the considered operational time, the core contained significant $^{235}$U ($\approx 10^{-5}$ atoms/b-cm), $^{239}$Pu ($\approx 5 \times 10^{-7}$ atoms/b-cm), and

$^{241}$Pu ($\approx 5 \times 10^{-7}$ atoms/b-cm). Meanwhile, the equilibrium number density of the target fissile isotope $^{233}$U was approximately 7.97×$10^{-5}$ atoms/b-cm. Small dips in neptunium and plutonium number density every 16 years are caused by removing $^{237}$Np and $^{242}$Pu (included in Processing group "Higher nuclides", see Table 3.2) which decay into $^{235}$Np and $^{239}$Pu, respectively. Thus, the production of new fissile materials in the core, as well as $^{233}$U breeding, made it possible to compensate for the negative effects of strong absorber accumulation ($^{234}$U) and keep the reactor critical.

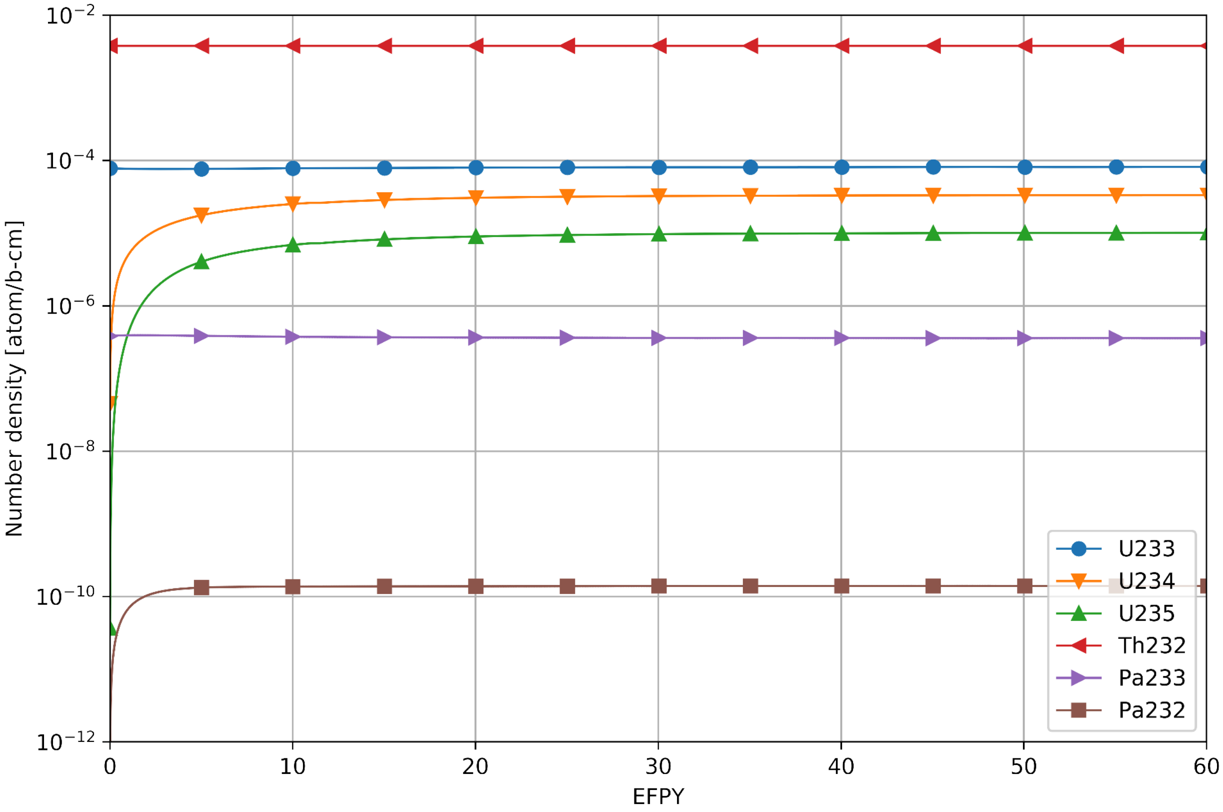


Figure 3.7: The number density of major nuclides during 60 years of reactor operation (reproduced from Rykhlevskii *et al.* [51]).

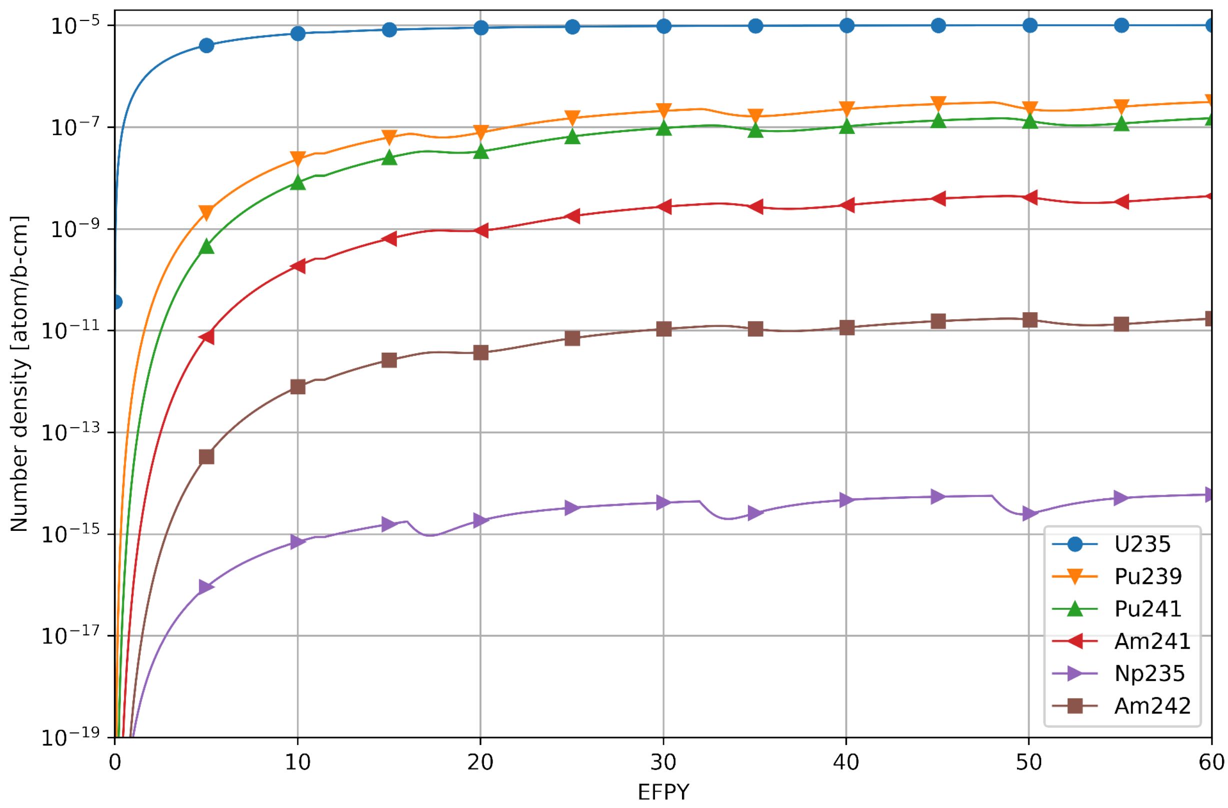


Figure 3.8: The number density of *fissile in epithermal spectrum* nuclides during 60 years of the reactor operation (reproduced from Rykhlevskii *et al.* [51]).

### 3.3.3 Neutron spectrum

Figure 3.9 shows the normalized neutron flux spectrum for the full-core MSBR model in the energy range from $10^{-8}$ to 10 MeV. The neutron energy spectrum at equilibrium is harder than at startup due to plutonium and other strong absorbers accumulating in the core during reactor operation.

Figure 3.10 shows that zone I produced more thermal neutrons than zone II, corresponding to a majority of fissions occurring in the central part of the core. In the undermoderated zone II, the neutron energy spectrum is harder, which leads to more intensive neutron capture by $^{232}$Th and helps achieve a relatively high breeding ratio. Moreover, the (n,$\gamma$) resonance energy range in $^{232}$Th is from $10^{-4}$ to $10^{-2}$ MeV. Therefore, the moderator-to-fuel ratio for zone II was chosen to shift the neutron energy spectrum in this range. Furthermore, in the central core region (zone I), the neutron energy spectrum shifts to a harder spectrum over 20 years of reactor operation; meanwhile, in the outer core region (zone II), a similar spectral shift takes place at a reduced scale. These results are in good agreement with the original ORNL report [30] and the most recent whole-core steady-state study [19].

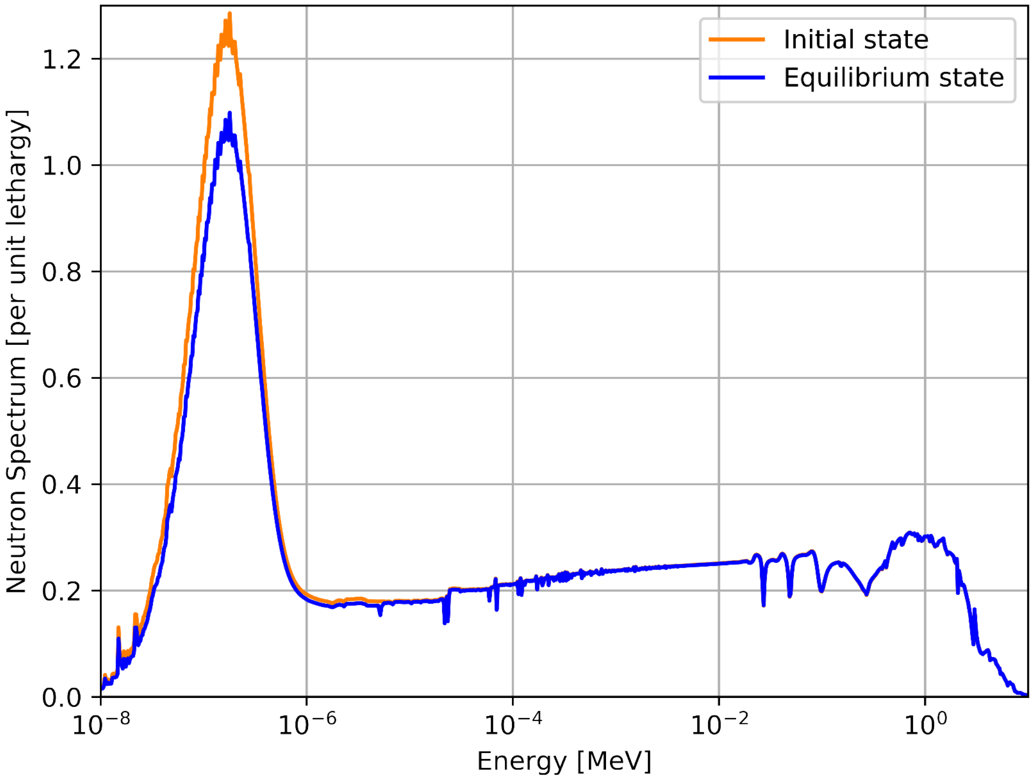


Figure 3.9: The neutron flux energy spectrum for initial and equilibrium state normalized by unit lethargy (reproduced from Rykhlevskii *et al.* [51]).

It is important to obtain the epithermal and thermal spectra to produce $^{233}$U from $^{232}$Th because the radiative capture cross section of thorium decreases monotonically from $10^{-10}$ MeV to $10^{-5}$ MeV. Hardening the spectrum tends to significantly increase resonance absorption in thorium and decrease absorptions in fissile and construction materials.

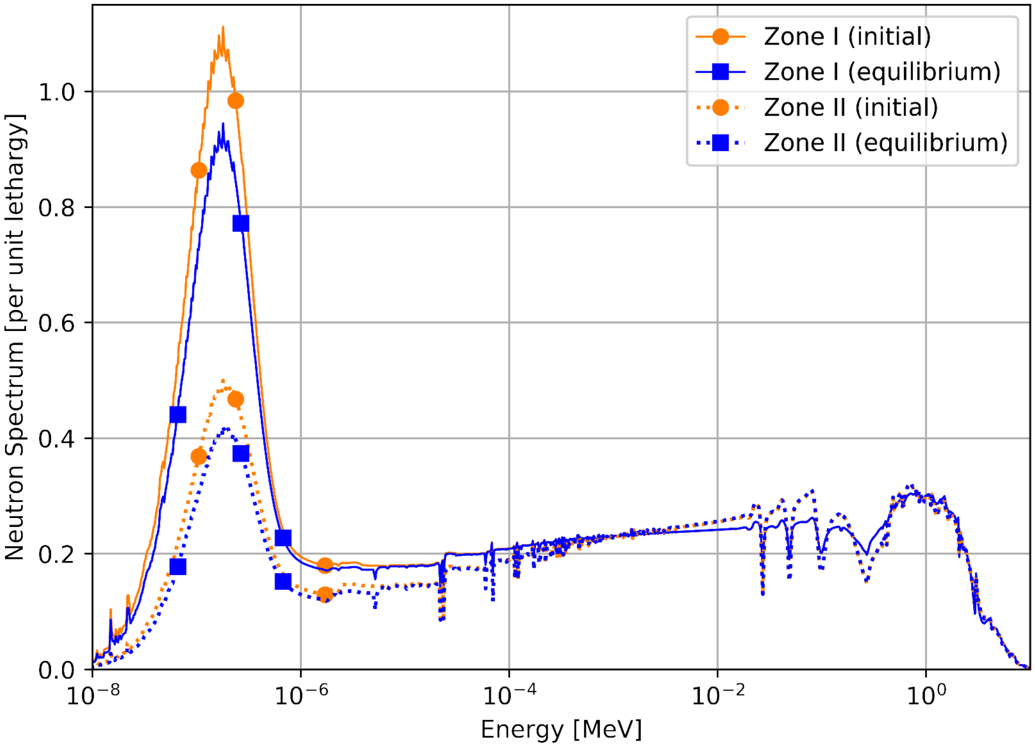


Figure 3.10: The neutron flux energy spectrum for initial and equilibrium state normalized by unit lethargy (reproduced from Rykhlevskii *et al.* [51]).

### 3.3.4 Neutron flux

this is a little confusing.

The spectrum includes all neutrons.

Figure 3.11 shows the radial distribution of fast and thermal neutron flux for initial and equilibrium compositions. The neutron fluxes have similar shapes for both compositions, but the equilibrium case has a harder spectrum. A significant spectral shift was observed in the central region of the core (zone I). In ~~while~~ for the outer region (zone II), spectral shif ~~it~~ is negligible for fast neutrons but notable for thermal neutrons. These neutron flux radial distributions agree with the fluxes in the original ORNL report [30]. Overall, spectrum hardening during MSBR operation should be carefully studied when designing the reactivity control system.

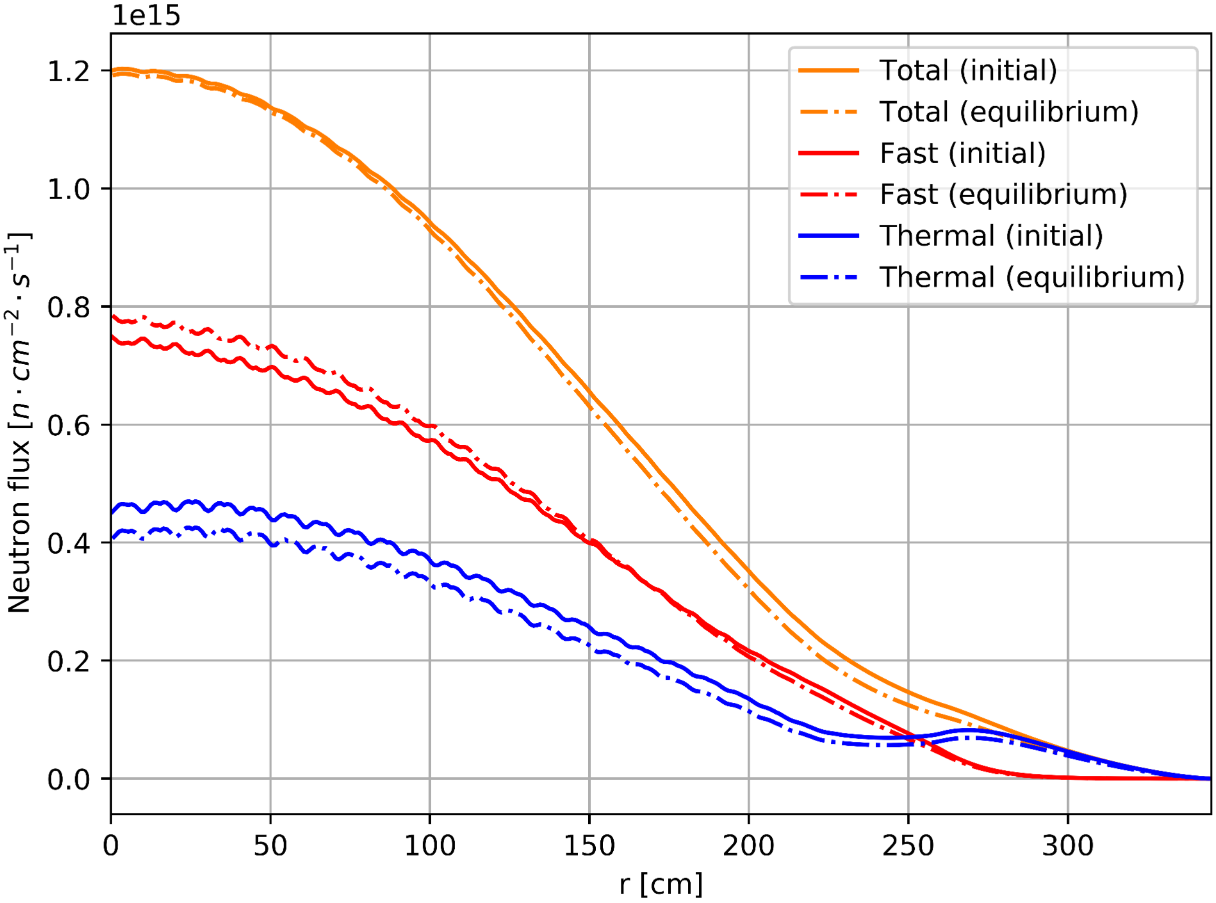


Figure 3.11: Radial neutron flux distribution for initial and equilibrium fuel salt compositions (reproduced from Rykhlevskii *et al.* [51]).

### 3.3.5 Power and breeding distribution

Table 3.3 shows the power fraction in each zone for initial and equilibrium fuel compositions. Figure 3.12 reflects the normalized power distribution of the MSBR quarter core for equilibrium fuel salt composition. For both the initial and equilibrium compositions, fission primarily occurs in the center of the core, namely zone I. The spectral shift during reactor operation results in slightly different power fractions at startup and equilibrium, but most of the power is still generated in zone I at equilibrium (Table 3.3).

Table 3.3: Power generation fraction in each zone for initial and equilibrium state (reproduced from Rykhlevskii *et al.* [51]).

| Core region | Initial | Equilibrium |
|---|---:|---:|
| Zone I | 97.91% | 98.12% |
| Zone II | 2.09% | 1.88% |

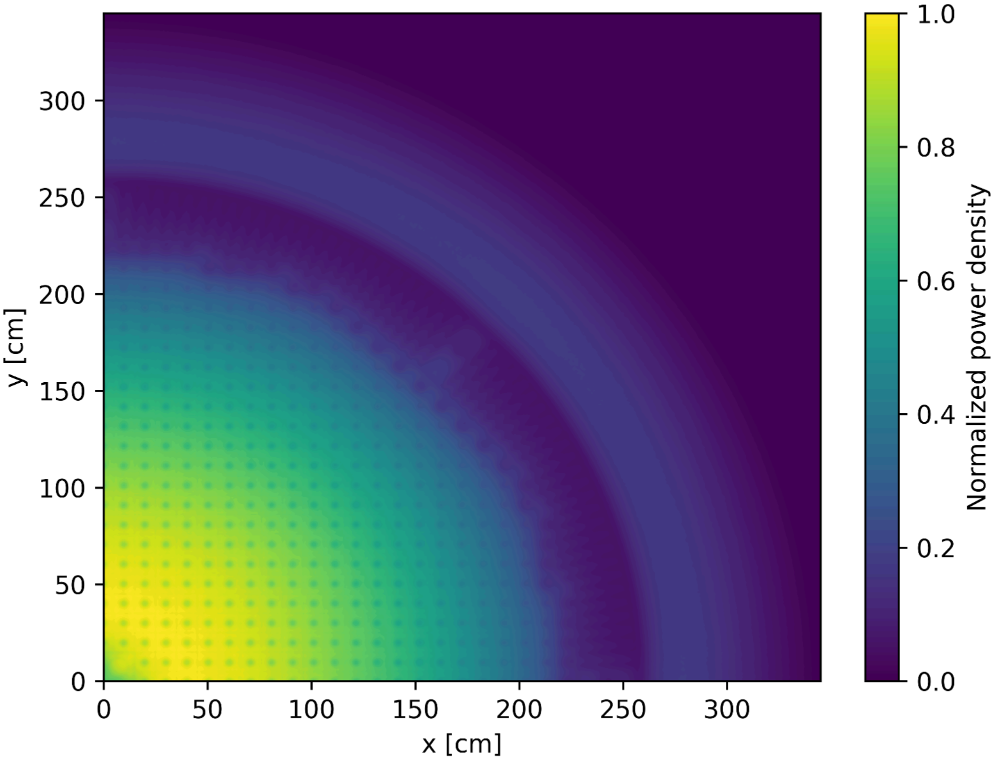


Figure 3.12: Normalized power density for equilibrium fuel salt composition (reproduced from Rykhlevskii *et al.* [51]).

Figure 3.13 shows the distribution of the neutron capture reaction rate for $^{232}$Th normalized by the total neutron flux for the initial and equilibrium states. The distribution reflects the spatial distribution of $^{233}$U production in the core. $^{232}$Th neutron capture produces $^{233}$Th, which then $\beta$-decays to $^{233}$Pa, the precursor for $^{233}$U production. Accordingly, this characteristic represents the breeding distribution in the MSBR core. The power and breeding distribution remained almost constant during the reactor operation. Even after 20 years of operation, most of the power is still generated in zone I.

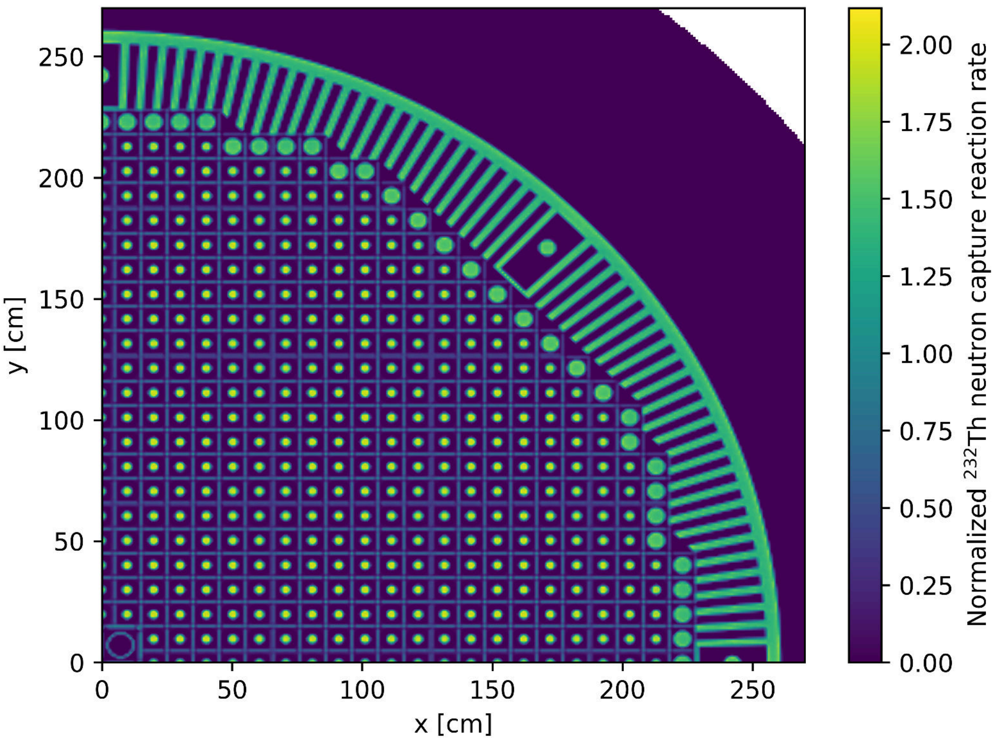


Figure 3.13: $^{232}$Th neutron capture reaction rate normalized by total flux for equilibrium fuel salt composition (reproduced from Rykhlevskii *et al.* [51]).

### 3.3.6 Thorium refill rate

In the MSBR, the only external feed material flow is $^{232}$Th. Figure 3.14 shows the $^{232}$Th feed rate calculated over 60 years of reactor operation. The $^{232}$Th feed rate fluctuates significantly as a result of the batch-wise nature of this online reprocessing approach. Figure 3.15 shows a zoomed thorium feed rate for a short 150-EFPD interval. Note that the large spikes of up to 36 kg/day in a thorium consumption occur every 3435 days. Those spikes happened due to strong absorbers' (Rb, Sr, Cs, Ba) removal at the end of the effective cycle (100% of these elements removing every 3435 days of operation). The corresponding effective multiplication factor increase (Figure 3.5) and breeding intensification leads to additional $^{232}$Th consumption.

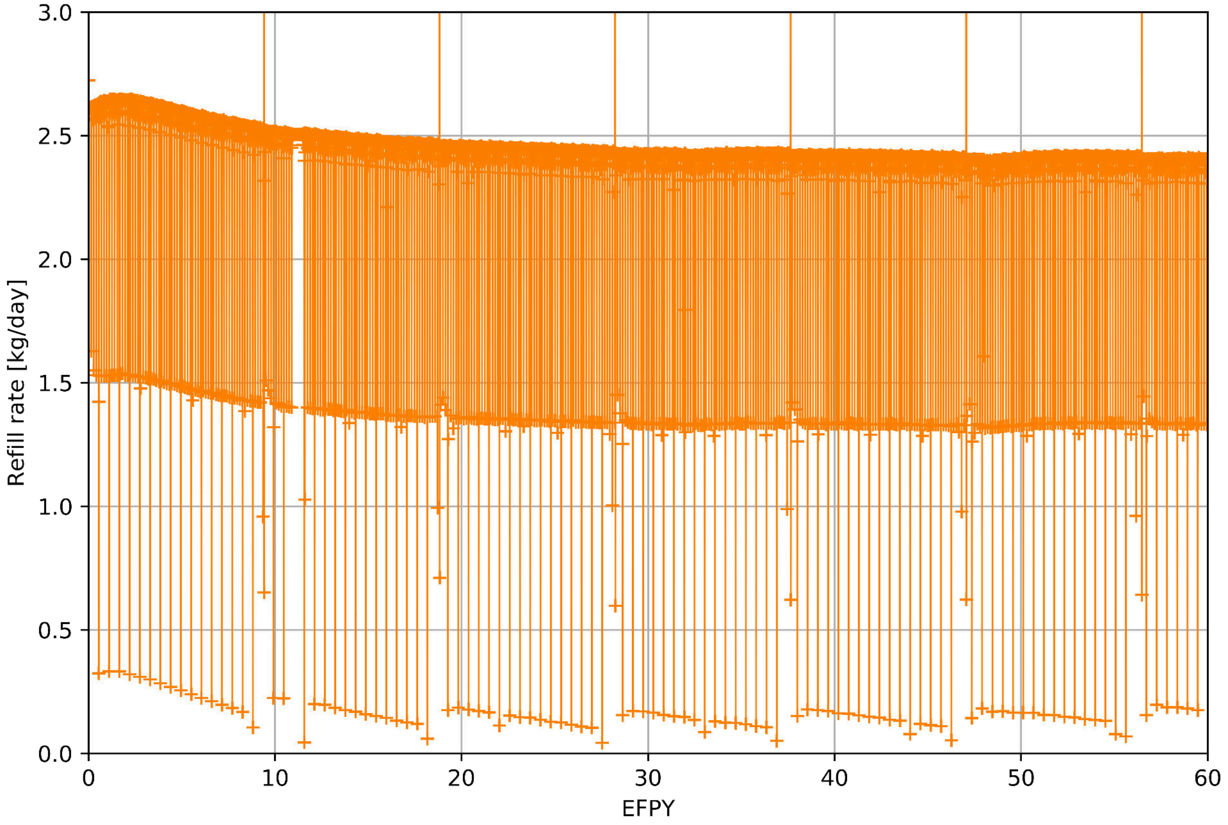


Figure 3.14: $^{232}$Th feed rate over 60 years of the MSBR operation (reproduced from Rykhlevskii *et al.* [51]).

The average thorium feed rate increases during the first 500 days of operation and steadily decreases due to spectrum hardening and accumulation of absorbers in the core. As a result, the average $^{232}$Th feed rate over 60 years of operation is about 2.40 kg/day. This thorium consumption rate is in good agreement with a recent online reprocessing study by ORNL [20]. At equilibrium, the thorium feed rate is determined by the reactor power, the energy released per fission, and the neutron energy spectrum.

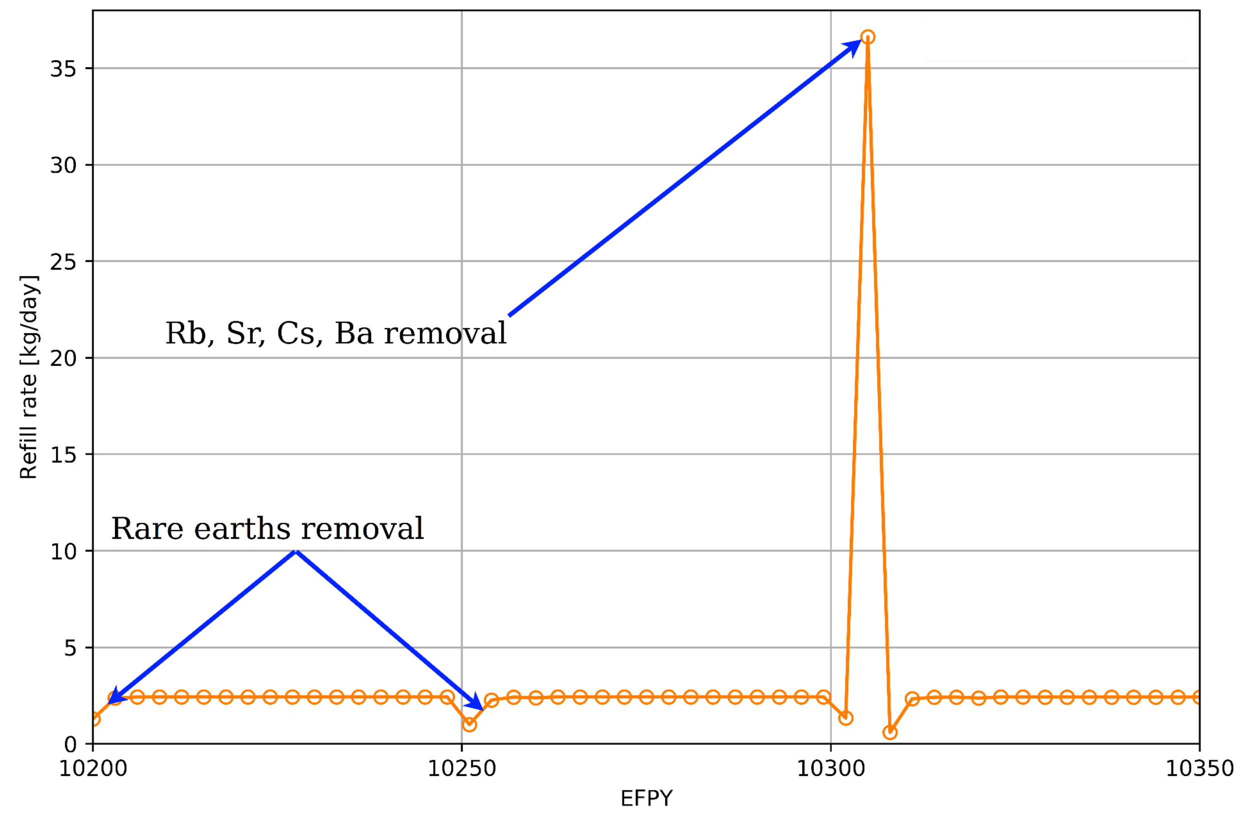


Figure 3.15: Zoomed $^{232}$Th feed rate for a 150-EFPD time interval (reproduced from Rykhlevskii *et al.* [51]).

## 3.4 Operational and safety parameter~~s~~ evolution

In Section 3.3, we reported how fuel salt composition changes during MSBR operation. The number density of the most important heavy isotopes, $^{233}$U and $^{232}$Th, was stable while transitioning from startup to equilibrium composition (Figure 3.7). At the same time, a number of different actinides is being produced in the reactor core. Most of these nuclides ($^{234}$U, $^{239}$Pu, $^{241}$Pu) have a much larger absorption cross section than $^{233}$U and $^{232}$Th loaded initially into the core, which causes significant neutron energy spectrum hardening. In the current section, we analyze how such neutron spectrum shift affects major operation and safety parameters such as temperature coefficients of reactivity and reactivity worth of the control rods.

### 3.4.1 Temperature coefficient of reactivity

Table 3.4 summarizes temperature effects on reactivity calculated in this work for both initial and equilibrium fuel compositions, compared to the original ORNL report data [30]. By propagating the $k_{eff}$ statistical error provided by Serpent, uncertainty for each temperature coefficient was obtained and appears in Table 3.4. Other sources of uncertainty are neglected, such as cross section measurement error and approximations inherent in the equations of state, providing both the salt and graphite density dependence on temperature. The main physical principle underlying the reactor temperature feedback is an expansion of heated material. If the fuel salt temperature increases, the density of the salt decreases; at the same time, the total volume of fuel salt in the core remains constant because the graphite bounds it. If the graphite temperature increases, the density of graphite decreases, creating additional space for fuel salt. To determine the temperature coefficients, the cross section temperatures for the fuel and moderator were changed from 900K to 1000K. Three different cases were considered:

1. Temperature of fuel salt rising from 900K to 1000K.
2. Temperature of graphite rising from 900K to 1000K.
3. Whole reactor temperature rising from 900K to 1000K.

Table 3.4: Temperature coefficients of reactivity for the initial and equilibrium states (reproduced from Rykhlevskii *et al.* [51]).

| Reactivity coefficient | Initial [pcm/K] | Equilibrium [pcm/K] | Reference [30] (initial) [pcm/K] |
|---|---|---|---|
| Doppler in fuel salt | −4.73 ± 0.038 | −4.69 ± 0.038 | −4.37 |
| Fuel salt density | +1.21 ± 0.038 | +1.66 ± 0.038 | +1.09 |
| Total fuel salt | −3.42 ± 0.038 | −2.91 ± 0.038 | −3.22 |
| Graphite spectral shift | +1.56 ± 0.038 | +1.27 ± 0.038 | |
| Graphite density | +0.14 ± 0.038 | +0.23 ± 0.038 | |
| Total moderator (graphite) | +1.69 ± 0.038 | +1.35 ± 0.038 | +2.35 |
| Total core | −1.64 ± 0.038 | −1.58 ± 0.038 | −0.87 |

In the first case, changes in the fuel temperature only impact fuel density. In this case, the geometry is unchanged because the fuel is a liquid. However, if the moderator heats up, both the density and the geometry change due to the thermal expansion of the solid graphite blocks and reflector. Accordingly, the new graphite density was calculated using a linear temperature expansion coefficient of $1.3\times10^{-6}$K$^{-1}$ [30]. A new geometry input for Serpent, which takes into account the displacement of graphite surfaces, was created based on this information. For calculation of displacement, it was assumed that the interface between the graphite reflector and vessel is immobile and the vessel temperature is constant. This is the most reasonable assumption for the short-term reactivity effects because inlet salt cools the graphite reflector and the inner surface of the vessel.

The fuel temperature coefficient (FTC) is negative for both initial and equilibrium fuel compositions due to thermal Doppler broadening of the resonance capture cross sections in the thorium. A small positive effect of fuel density on reactivity increases from +1.21 pcm/K at reactor startup to +1.66 pcm/K for equilibrium fuel composition, which has a negative effect on FTC magnitude during the reactor operation; this is in good agreement with earlier research [30, 19]. The moderator temperature coefficient (MTC) is positive for the startup composition and decreases during reactor operation because of spectrum hardening with fuel depletion. Finally, the total temperature coefficient of reactivity is negative for both cases but decreases in magnitude during reactor operation due to spectral shift. In summary, even after 20 years of operation, the total temperature coefficient of reactivity is relatively large and negative during reactor operation (comparing with conventional PWR which has temperature coefficient about -1.71 pcm/°F ≈ -3.08 pcm/K [59]), despite positive MTC, and affords excellent reactor stability and control.

### 3.4.2 Reactivity control system rod worth

Table 3.5 summarizes the reactivity control system worth. During normal operation, the control (graphite) rods are fully inserted, and the safety ($B_4C$) rods are fully withdrawn. To insert negative reactivity into the core, the graphite rods are gradually withdrawn from the core. In an accident, the safety rods would be dropped down into the core. The integral rod worths were calculated for various positions to separately estimate the worth of the control rods[4], the safety rods, and the whole reactivity control system. Control rod integral worth is approximately 28 cents and stays almost constant during reactor operation. The safety rod integral worth decreases by 16.2% during 20 years of operation because of neutron spectrum hardening and absorber accumulation in proximity to reactivity control system rods. This 16% decline in control system worth must be taken into account in MSBR accident analysis and safety justification.

Table 3.5: Control system rod worth for the initial and equilibrium fuel compositions (reproduced from Rykhlevskii *et al.* [51]).

| Reactivity parameter | Initial[¢] | Equilibrium[¢] |
|---|---:|---:|
| Control (graphite) rod integral worth | 28.2 ± 0.8 | 29.0 ± 0.8 |
| Safety rod integral worth | 251.8 ± 0.8 | 211.0 ± 0.8 |
| Total reactivity control system worth | 505.8 ± 0.7 | 424.9 ± 0.8 |

### 3.4.3 Six Factor Analysis

The effective multiplication factor can be expressed using the following formula:

$$k_{eff} = \eta f p \epsilon P_f P_t \tag{3.1}$$

where

$\eta$ = neutron reproduction factor [−]

$f$ = thermal utilization factor [−]

$p$ = resonance escape probability [−]

$\epsilon$ = fast fission factor [−]

[4] In [30], the graphite rods are referred to as “control” rods.

$P_f$ = fast non-leakage probability $[-]$

$P_t$ = thermal non-leakage probability $[-]$.

Table 3.6 summarizes the six factors for both the initial and equilibrium fuel salt compositions. Using Serpent and SaltProc, these factors and their statistical uncertainties have been calculated for both the initial and equilibrium fuel salt compositions (see Table 3.1). The fast and thermal non-leakage probabilities remain constant despite the evolving neutron spectrum during operation. In contrast, the neutron reproduction factor ($\eta$), resonance escape probability ($p$), and fast fission factor ($\epsilon$) are considerably different between startup and equilibrium. As indicated in Figure 3.9, the neutron spectrum is softer at the beginning of reactor life. Neutron spectrum hardening causes the fast fission factor to increase through the core lifetime; the opposite is true for the resonance escape probability. Finally, the neutron reproduction factor decreases during reactor operation due to the accumulation of fissile plutonium isotopes.

Table 3.6: Six factors for the full-core MSBR model for the initial and equilibrium fuel compositions (reproduced from Rykhlevskii *et al.* [51]).

| Factor | Initial | Equilibrium |
|---|---:|---:|
| Neutron reproduction factor ($\eta$) | 1.3960 ± .000052 | 1.3778 ± .00005 |
| Thermal utilization factor (f) | 0.9670 ± .000011 | 0.9706 ± .00001 |
| Resonance escape probability (p) | 0.6044 ± .000039 | 0.5761 ± .00004 |
| Fast fission factor ($\epsilon$) | 1.3421 ± .000040 | 1.3609 ± .00004 |
| Fast non-leakage probability ($P_f$) | 0.9999 ± .000004 | 0.9999 ± .000004 |
| Thermal non-leakage probability ($P_t$) | 0.9894 ± .000005 | 0.9912 ± .00005 |

## 3.5 Benefits of fission products removal

To investigate how online fuel salt processing described in Chapter 2 affects the reactor performance, the separate effect of each poison group removal was studied in this section.

### 3.5.1 The effect of removing fission products from the fuel salt

Loading the initial fuel salt composition into the MSBR core leads to a supercritical configuration (Figure 3.16). After reactor startup, the effective multiplication factor for the case with volatile gases and noble metals removal is approximately 7500 pcm higher than for the case without fission product removal. This significant impact on the reactor core lifetime is achieved due to the immediate removal (20 sec cycle time) and the high absorption cross sections of Xe, Kr, Mo, and other noble metals removed. The effect of rare earth element removal is significant in a few months after startup and reached approximately 5500 pcm after 10 years of operation. The rare earth elements were removed at a slower rate (50-day cycle time). Moreover, Figure 3.16 demonstrates that batch-wise removal of strong absorbers every 3 days unnecessarily leads to fluctuation in results, but rare earth element removal every 50 days causes an approximately 600 pcm jump in reactivity.

The effective multiplication factor of the core reduces gradually over operation time because the fissile material ($^{233}$U) continuously depletes from the fuel salt due to fission while fission products accumulate in the fuel salt simultaneously. Eventually, without fission products removal, the reactivity decreases to the subcritical state after approximately 500 and 1300 days of operation for cases with no removal and volatile gases & noble metals removal, respectively. The time when the simulated core becomes subcritical ($k_{eff}$ <1.0 for full-core model) is called the core lifetime. Therefore, removing fission products provides significant neutronic benefits and enables a longer core lifetime.

## 3.6 Concluding remarks

This chapter introduces the first ever version of the open-source MSR simulation package SaltProc v0.1. The main goal of this work has been to demonstrate SaltProc's capability to find the equilibrium fuel salt composition (the number densities of major isotopes vary by less than 1% over several years). A secondary goal has been to compare predicted operational

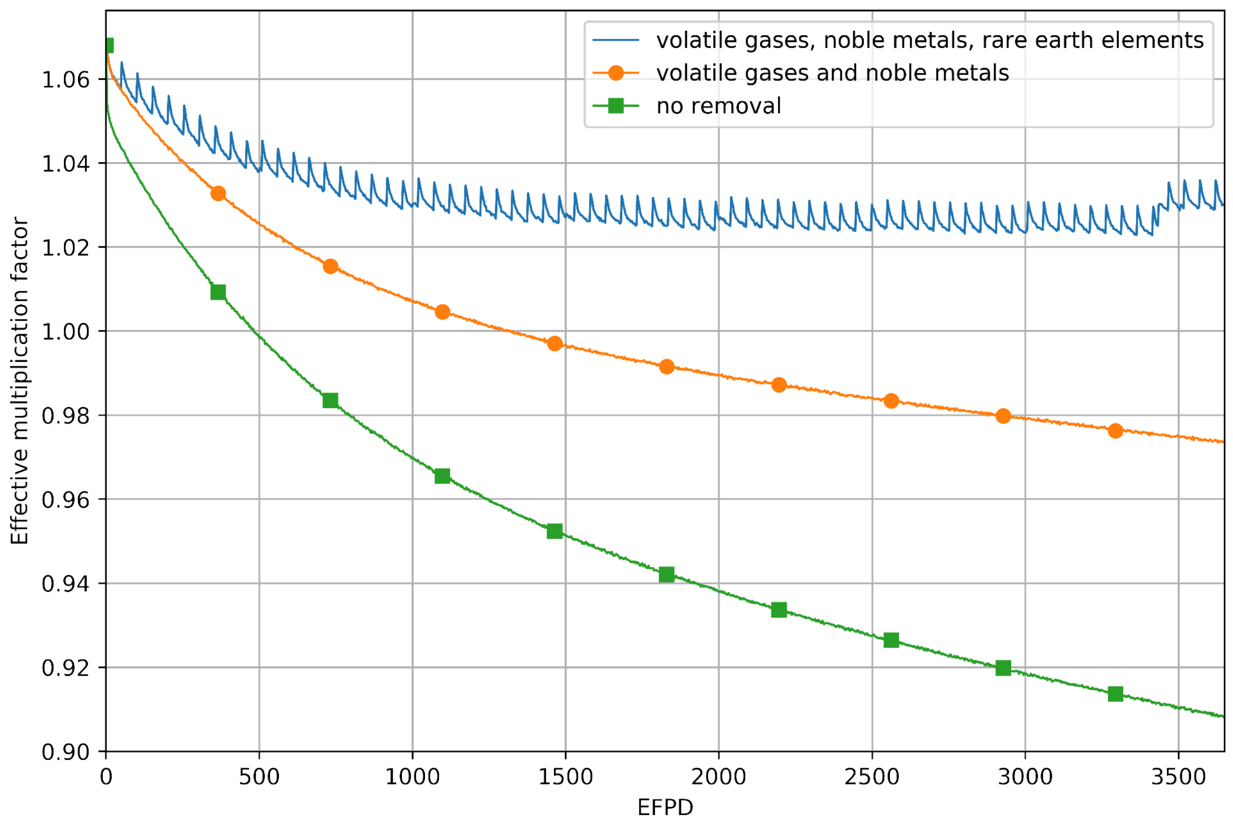


Figure 3.16: Calculated effective multiplication factor for the full-core MSBR model with the removal of various fission product groups over 10 years of operation (reproduced from Rykhlevskii *et al.* [51]).

and safety parameters (e.g., neutron energy spectrum, power and breeding distribution, temperature coefficients of reactivity) of the MSBR at startup and equilibrium states. A tertiary goal has been to demonstrate the benefits of continuous fission product removal for thermal MSR design.

To achieve these goals, a full-core high-fidelity benchmark model of the MSBR was created in Serpent 2. The full-core model was used instead of the simplified single-cell model [20, 60, 23] to precisely describe the two-region MSBR concept design sufficiently to represent breeding in the outer core zone accurately. When running depletion calculations, the most critical fission products and $^{233}$Pa are removed, while fertile and fissile materials are added to the fuel salt every 3 days. Meanwhile, the removal interval for the rare earths, volatile fluorides, and semi-noble metals was greater than one month (50 days), which caused significant $k_{eff}$ fluctuation.

The results in this chapter indicate that $k_{eff}$ slowly decreases from 1.075 and reaches 1.02 at equilibrium after approximately 6 years of operation. At the same time, the concentrations of $^{233}$U, $^{232}$Th, $^{233}$Pa, and $^{232}$Pa stabilized after approximately 2500 days of operation. Particularly, $^{233}$U number density equilibrates[5] after 16 years of operation. Consequently, the core reaches the quasi-equilibrium state after 16 years of operation. However, a wide variety of actinides, including fissile isotopes (e.g., $^{233}$U and $^{239}$Pu) and non-fissile strong absorbers ($^{234}$U), continue accumulating in the core.

Those actinides cause neutron energy spectrum hardening as the core approaches equilibrium. Moreover, the neutron energy spectrum in the central core region is much softer than in the outer core region due to the lower moderator-to-fuel ratio in the outer zone, and this distribution remains stable during reactor operation. Finally, the epithermal or thermal spectrum is needed to effectively breed $^{233}$U from $^{232}$Th because the radiative capture cross section of thorium-232 monotonically decreases from $10^{-10}$ MeV to $10^{-5}$ MeV. A harder spectrum in the outer core region tends to significantly increase resonance absorption in thorium and decrease the absorptions in fissile and structural materials.

The spatial power distribution in the MSBR shows that 98% of the fission power is generated in the central zone I, and the neutron energy spectral shift has zero effect on the power distribution. The spatial distribution of neutron capture reaction rate for fertile $^{232}$Th, corresponding to breeding in the core, confirms that most of the breeding occurs in an outer, undermoderated, region of the MSBR core. Finally, the average $^{232}$Th refill rate throughout 60 years of operation is approximately 2.40 kg/day or 100 g/GWh$_e$.

We compared the safety parameters at startup and state using the Serpent Monte Carlo code. The total temperature coefficient is large and negative at startup and equilibrium, but the magnitude decreases throughout reactor operation from −3.10 to −0.94 pcm/K as the spectrum hardens. The moderator temperature coefficient is positive and also decreases

[5] fluctuates less than 0.8%

during fuel depletion. The reactivity control system efficiency analysis showed that the safety rod integral worth decreases by approximately 16.2% over 16 years of operation, while the graphite rod integral worth remains constant. Therefore, neutron energy spectrum hardening during fuel salt depletion has an undesirable impact on MSBR stability and controllability and should be taken into consideration in further analysis of transient accident scenarios.

Finally, we proved that the MSBR core performance benefits from the removal of volatile gases, noble metals, and rare earths from the fuel salt. Immediate removal of volatile gases (e.g., xenon) and noble metals increased reactivity by approximately 7500 pcm over a 10-year timeframe. In contrast, the effect of relatively slower removal of rare earth elements (every 50 days cycle instead of 3 days) has less impact (5500 pcm) on the core reactivity after 10 years of operation. An additional study is needed to establish neutronic and economic tradeoffs of removing each element.

This chapter's results also helped identify the main directions of SaltProc v0.1 improvement. Firstly, the poison removal efficiency is not ideal, as was discussed in Chapter 1; consequently, the user should be able to simulate the fuel salt reprocessing system using a variable, non-ideal extraction efficiency. Secondly, SaltProc v0.1 entirely removes elements with longer residence times (semi-noble metals, volatile fluorides, Rb, Sr, Cs, Ba, Eu) at the end of cycle time (e.g., 3435 days for rubidium) which causes significant jumps in $k_{eff}$ due to the removal of large batches of the poison at once. In SaltProc v1.0, this drawback has been eliminated by removing a fraction of the target element with longer residence time at each depletion step. In the following chapters, improved SaltProc v1.0 capabilities will be demonstrated.

# Chapter 4

# Tool demonstration: Transatomic Power MSR

## 4.1 Transatomic Power Molten Salt Reactor concept design and model description

## 4.2 Fuel salt isotopic composition dynamics

## 4.3 Reactor load following analysis

## 4.4 Prototype design for the xenon removal system

## 4.5 Concluding remarks

# Chapter 5

# TAP MSR Safety analysis

## 5.1 Safety and operational parameters

## 5.2 Fuel salt composition influence on safety

## 5.3 Concluding remarks

# Chapter 6

# Error propagation in depletion calculations

## 6.1 Statistical uncertainty in depleted fuel composition

## 6.2 Nuclear data related uncertainty in depleted fuel composition

## 6.3 Concluding remarks

# Chapter 7

# Conclusions and future work

# References

[1] IEA. Nuclear Power in a Clean Energy System. Technical report, IEA, May 2019.

[2] World Energy Outlook 2017. Technical report, IEA, November 2017.

[3] U. S. DoE. A technology roadmap for generation IV nuclear energy systems. In *Nuclear Energy Research Advisory Committee and the Generation IV International Forum*, pages 48–52, 2002.

[4] Paul N. Haubenreich and J. R. Engel. Experience with the Molten-Salt Reactor Experiment. *Nuclear Technology*, 8(2):118–136, February 1970.

[5] David LeBlanc. Molten salt reactors: A new beginning for an old idea. *Nuclear Engineering and Design*, 240(6):1644–1656, June 2010.

[6] Massimiliano Fratoni, David Barnes, Ehud Greenspan, and Augusto Gandini. Design and Analysis of Molten Salt Reactor Fueled by TRU from LWR. *PHYSOR 2004*, 2004.

[7] Dalin Zhang, Limin Liu, Minghao Liu, Rongshuan Xu, Cheng Gong, Jun Zhang, Chenglong Wang, Suizheng Qiu, and Guanghui Su. Review of conceptual design and fundamental research of molten salt reactors in China. *International Journal of Energy Research*, 42(5):1834–1848, 2018.

[8] Carlo Fiorina. *The molten salt fast reactor as a fast spectrum candidate for thorium implementation*. PhD, Politecnico Di Milano, March 2013.

[9] V. Ignatiev, O. Feynberg, I. Gnidoi, A. Merzlyakov, A. Surenkov, V. Uglov, A. Zagnitko, V. Subbotin, I. Sannikov, A. Toropov, V. Afonichkin, A. Bovet, V. Khokhlov, V. Shishkin, M. Kormilitsyn, A. Lizin, and A. Osipenko. Molten salt actinide recycler and transforming system without and with Th-U support: Fuel cycle flexibility and key material properties. *Annals of Nuclear Energy*, 64(Supplement C):408–420, February 2014.

[10] Andrei Rykhlevskii, Jin Whan Bae, and Kathryn Huff. arfc/saltproc: Code for online reprocessing simulation of molten salt reactor with external depletion solver SERPENT. *Zenodo*, July 2018.

[11] Jaakko Leppanen, Maria Pusa, Tuomas Viitanen, Ville Valtavirta, and Toni Kaltiaisenaho. The Serpent Monte Carlo code: Status, development and applications in 2013. *Annals of Nuclear Energy*, 82:142–150, August 2014.

[12] Ali Ahmad, Edward B. McClamrock, and Alexander Glaser. Neutronics calculations for denatured molten salt reactors: Assessing resource requirements and proliferation-risk attributes. *Annals of Nuclear Energy*, 75:261–267, January 2015.

[13] H. F. Bauman, G. W. Cunningham III, J. L. Lucius, H. T. Kerr, and C. W. Jr Craven. Rod: A Nuclear and Fuel-Cycle Analysis Code for Circulating-Fuel Reactors. Technical Report ORNL-TM-3359, Oak Ridge National Lab., Tenn., January 1971.

[14] C. W. Kee and L. E. McNeese. MRPP: multiregion processing plant code. Technical Report ORNL/TM–4210, Oak Ridge National Lab., 1976.

[15] Carlo Fiorina, Manuele Aufiero, Antonio Cammi, Fausto Franceschini, Jiri Krepel, Lelio Luzzi, Konstantin Mikityuk, and Marco Enrico Ricotti. Investigation of the MSFR core physics and fuel cycle characteristics. *Progress in Nuclear Energy*, 68:153–168, September 2013.

[16] R. J. Sheu, C. H. Chang, C. C. Chao, and Y. W. H. Liu. Depletion analysis on long-term operation of the conceptual Molten Salt Actinide Recycler & Transmuter (MOSART) by using a special sequence based on SCALE6/TRITON. *Annals of Nuclear Energy*, 53:1–8, March 2013.

[17] M. Aufiero, A. Cammi, C. Fiorina, J. Leppnen, L. Luzzi, and M. E. Ricotti. An extended version of the SERPENT-2 code to investigate fuel burn-up and core material evolution of the Molten Salt Fast Reactor. *Journal of Nuclear Materials*, 441(13):473–486, October 2013.

[18] D. Heuer, E. Merle-Lucotte, M. Allibert, M. Brovchenko, V. Ghetta, and P. Rubiolo. Towards the thorium fuel cycle with molten salt fast reactors. *Annals of Nuclear Energy*, 64:421–429, February 2014.

[19] Jinsu Park, Yongjin Jeong, Hyun Chul Lee, and Deokjung Lee. Whole core analysis of molten salt breeder reactor with online fuel reprocessing. *International Journal of Energy Research*, 39(12):1673–1680, October 2015.

[20] Benjamin R. Betzler, Jeffrey J. Powers, and Andrew Worrall. Molten salt reactor neutronics and fuel cycle modeling and simulation with SCALE. *Annals of Nuclear Energy*, 101(Supplement C):489–503, March 2017.

[21] Benjamin R. Betzler, Kursat B. Bekar, William Wieselquist, Shane W. Hart, and Shane G. Stimpson. Molten Salt Reactor Fuel Depletion Tools in SCALE. In *Proc. of GLOBAL International Fuel Cycle Conference*, Seattle, WA, United States, September 2019. American Nuclear Society.

[22] A. Nuttin, D. Heuer, A. Billebaud, R. Brissot, C. Le Brun, E. Liatard, J. M. Loiseaux, L. Mathieu, O. Meplan, E. Merle-Lucotte, H. Nifenecker, F. Perdu, and S. David. Potential of thorium molten salt reactorsdetailed calculations and concept evolution with a view to large scale energy production. *Progress in Nuclear Energy*, 46(1):77–99, January 2005.

[23] Benjamin R. Betzler, Sean Robertson, Eva E. Davidson, Jeffrey J. Powers, Andrew Worrall, Leslie Dewan, and Mark Massie. Fuel cycle and neutronic performance of a spectral shift molten salt reactor design. *Annals of Nuclear Energy*, 119:396–410, September 2018.

[24] J. J. Powers, T. J. Harrison, and J. C. Gehin. A new approach for modeling and analysis of molten salt reactors using SCALE. In *Transactions of the American Nuclear Society*, Sun Valley, ID, USA, July 2013. American Nuclear Society, 555 North Kensington Avenue, La Grange Park, IL 60526 (United States).

[25] Ian C. Gauld, Georgeta Radulescu, Germina Ilas, Brian D. Murphy, Mark L. Williams, and Dorothea Wiarda. Isotopic Depletion and Decay Methods and Analysis Capabilities in SCALE. *Nuclear Technology*, 174(2):169–195, May 2011.

[26] Jaakko Leppnen and Maria Pusa. Burnup calculation capability in the PSG2/Serpent Monte Carlo reactor physics code. *Proc. M&C*, pages 3–7, 2009.

[27] Nicholas Tsoulfanidis. *The Nuclear Fuel Cycle.* American Nuclear Society, La Grange Park, Illinois, USA, 2013. 00177.

[28] C. J. Werner. MCNP - A General Monte Carlo N-Particle Transport Code, 2017.

[29] Benjamin R. Betzler, Sean Robertson, Eva E. Davidson, Jeffrey J. Powers, Andrew Worrall, Leslie Dewan, and Mark Massie. Assessment of the Neutronic and Fuel Cycle Performance of the Transatomic Power Molten Salt Reactor Design. Technical Report ORNL/TM-2017/475 CRADA/NFE-16-06345, Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States), September 2017.

[30] R. C. Robertson. Conceptual Design Study of a Single-Fluid Molten-Salt Breeder Reactor. Technical Report ORNL–4541, ORNL, January 1971.

[31] R. J. Kedl and A. Houtzeel. DEVELOPMENT OF A MODEL FOR COMPUTING Xe-135 MIGRATION IN THE MSRE. Technical report, Oak Ridge National Lab., Tenn., 1967.

[32] F. N. Peebles. Removal of Xenon-135 from Circulating Fuel Salt of the MSBR by Mass Transfer to Helium Bubbles. Technical Report ORNL-TM-2245, Oak Ridge National Laboratory, Oak Ridge, TN, United States, 1968.

[33] Kathryn Huff. Enabling Load-Following Capability in the TAP MSR, December 2018.

[34] R. B. Briggs. Molten-Salt Reactor Program semiannual progress report for period ending July 31, 1964. Technical Report Archive and Image Library ORNL-3708, Oak Ridge National Laboratory, Oak Ridge, TN, United States, 1964.

[35] David Eugene Holcomb, Roger A. Kisner, and Sacit M. Cetiner. Instrumentation Framework for Molten Salt Reactors. Technical report, Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States), 2018.

[36] Joanna Mcfarlane, Paul Allen Taylor, David Eugene Holcomb, and Willis Poore III. Review of Hazards Associated with Molten Salt Reactor Fuel Processing Operations. Technical report, Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States), 2019.

[37] R. B. Lindauer and C. K. McGlothlan. Design, Construction, and Testing of a Large Molten Salt Filter. *Oak Ridge National Laboratory, Tech. Rep. ORNL-TM-2478*, 1969.

[38] Kirk Sorensen. One-Fluid MSBR Chemical Processing -, 2006.

[39] R. B. Briggs. Molten-salt reactor program. Semiannual progress report. Technical Report ORNL-4396, Oak Ridge National Lab., Tenn., February 1969.

[40] Sylvie. Delpech, Cline Cabet, Cyrine Slim, and Grard S. Picard. Molten fluorides for nuclear applications. *Materials Today*, 13(12):34–41, December 2010.

[41] X. Doligez, D. Heuer, E. Merle-Lucotte, M. Allibert, and V. Ghetta. Coupled study of the Molten Salt Fast Reactor core physics and its associated reprocessing unit. *Annals of Nuclear Energy*, 64(Supplement C):430–440, February 2014.

[42] M. E. Whatley, L. E. McNeese, W. L. Carter, L. M. Ferris, and E. L. Nicholson. Engineering development of the MSBR fuel recycle. *Nuclear Applications and Technology*, 8(2):170–178, 1970.

[43] W. L. Carter and E. L. Nicholson. DESIGN AND COST STUDY OF A FLUORINATION–REDUCTIVE EXTRACTION–METAL TRANSFER PROCESSING PLANT FOR THE MSBR. Technical Report ORNL-TM-3579, Oak Ridge National Lab. (ORNL), Oak Ridge, TN (United States), January 1972.

[44] Jaakko Leppanen, Ville Hovi, Timo Ikonen, Joona Kurki, Maria Pusa, Ville Valtavirta, and Tuomas Viitanen. The Numerical Multi-Physics project (NUMPS) at VTT Technical Research Centre of Finland. *Annals of Nuclear Energy*, 84:55–62, October 2015.

[45] L. Dagum and R. Menon. OpenMP: an industry standard API for shared-memory programming. *IEEE Computational Science and Engineering*, 5(1):46–55, January 1998.

[46] OECD/NEA. The JEFF-3.1.2 Nuclear Data Library. Technical Report JEFF Report 24, OECD/NEA Data Bank, OECD/NEA, 2014.

[47] M. B. Chadwick. ENDF/B-VII.1 Nuclear Data for Science and Technology: Cross Sections, Covariances, Fission Product Yields and Decay Data. *Nuclear Data Sheets*, 112(12):2887–2996, December 2011.

[48] Andrei Rykhlevskii. Advanced online fuel reprocessing simulation for Thorium-fueled Molten Salt Breeder Reactor. Master's thesis, University of Illinois at Urbana-Champaign, Urbana, IL, April 2018.

[49] The HDF Group. Hierarchical data format, version 5, 1997.

[50] Anthony Scopatz, Paul K. Romano, Paul P. H. Wilson, and Kathryn D. Huff. PyNE: Python for Nuclear Engineering. In *Transactions of the American Nuclear Society*, volume 107, San Diego, CA, USA, November 2012. American Nuclear Society.

[51] Andrei Rykhlevskii, Jin Whan Bae, and Kathryn D. Huff. Modeling and simulation of online reprocessing in the thorium-fueled molten salt breeder reactor. *Annals of Nuclear Energy*, 128:366–379, June 2019.

[52] Eleftherios Koutsofios and Stephen C. North. Drawing graphs with dot. *AT&T Bell Laboratories*, 1996.

[53] John Ellson, Emden R. Gansner, Eleftherios Koutsofios, Stephen C. North, and Gordon Woodhull. Graphviz and dynagraph  static and dynamic graph drawing tools. In *Graph Drawing Software*, pages 127–148. Springer-Verlag, 2003.

[54] Holger Krekel, Bruno Oliveira, Ronny Pfannschmidt, Floris Bruynooghe, Brianna Laugher, and Florian Bruhin. pytest: Python testing tool, 2004.

[55] Travis. travis-ci/travis-api, 2016.

[56] Georg Brandl. Sphinx: Python Documentation Generator, 2009.

[57] Benjamin R. Betzler, J. J. Powers, and A. Worrall. Modeling and simulation of the start-up of a thorium-based molten salt reactor. In *Proc. Int. Conf. PHYSOR*, May 2016.

[58] Andrei Rykhlevskii, Alexander Lindsay, and Kathryn D. Huff. Full-core analysis of thorium-fueled Molten Salt Breeder Reactor using the SERPENT 2 Monte Carlo code. In *Transactions of the American Nuclear Society*, Washington, DC, United States, November 2017. American Nuclear Society.

[59] Benoit Forget, Kord Smith, Shikhar Kumar, Miriam Rathbun, and Jingang Liang. Integral Full Core Multi-Physics PWR Benchmark with Measured Data. Technical report, Massachusetts Institute of Technology, 2018.

[60] Andrei Rykhlevskii, Alexander Lindsay, and Kathryn D. Huff. Online reprocessing simulation for thorium-fueled molten salt breeder reactor. In *Transactions of the American Nuclear Society*, Washington, DC, United States, November 2017. American Nuclear Society.