

© 2022 by Luke Seifert. All rights reserved.

SEIFERT THESIS

BY

LUKE SEIFERT

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Nuclear, Plasma, Radiological Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2022

Urbana, Illinois

Master's Committee:

Abstract

Abstract.

Acknowledgments

Acks.

Table of Contents

List of Tables	v
List of Figures	vi
Chapter 1 Introduction	1
Chapter 2 Literature Review	2
2.1 Molten Salt Reactors	2
2.2 Online Reprocessing	2
2.2.1 Batchwise Reprocessing	3
2.2.2 Continuous Reprocessing	4
2.2.3 DNP Movement	7
2.3 Depletion	8
2.4 The Molten Salt Breeder Reactor	9
2.5 MSR Modeling Approaches	11
References	17

List of Tables

2.1	Serpent2 Second Setting Approaches	7
2.2	MSBR Online Reprocessing Cycle Times [22]	11
2.3	Molten Salt Reactor Models with Online Reprocessing	12

List of Figures

2.1	Plots of DNP concentrations in the molten salt reactor experiment [26]. The left side of each image shows concentration with no flow, while the right shows the concentration with a 1200 gallon per minute flow rate. Top Left) Longest lived DNP group. Top Right) Third longest lived DNP group. Bottom) Shortest lived DNP group.	8
2.2	MSBR core axial slice showing the different regions from [24]. Zones II-A and II-B are where the spectrum is harder and there is increased breeding. Zone I is where there is more fission and a softer spectrum. The yellow is fuel salt, the purple is graphite, and the cyan is the reactor vessel.	10
2.3	MSBR simplified flow diagram from [22] showing part of primary side.	10

Chapter 1

Introduction

Introduction [?].

Chapter 2

Literature Review

2.1 Molten Salt Reactors

Modeling and simulation of liquid fueled molten salt reactors (MSRs) differs from solid fueled reactors in two main areas. The first is in the fresh fuel feed and fission product removal streams used during operation in MSRs, called online reprocessing. Modeling an MSR without online reprocessing will cause very different results during depletion calculations due to the lack of fresh fuel and buildup of fission products. In order to simulate this reprocessing functionality, software can either use batchwise reprocessing or continuous reprocessing.

The second difference is in the movement of the fuel salt, which causes delayed neutron precursors (DNPs) to move in the core. Because the DNPs have different decay times before the delayed neutrons are born, the effect of the fuel movement on each group varies. However, the overall result is more delayed neutrons in less important regions, such as external piping, which means there is a reduced effective delayed neutron fraction in the core.

2.2 Online Reprocessing

The online reprocessing functionality of MSRs is simulated in different ways depending on the particular software used, and depending on the methods employed by that software. The two main ways to approach the online reprocessing problem is by either using batch-wise reprocessing or continuous reprocessing. There are also different approaches in terms of modeling the full core and using a unit-cell model.

Codes such as SaltProc and older versions of ChemTriton implement batchwise reprocessing by running the simulation, stopping, and then adding and removing materials [24, 6]. The process is then iterated. This method is useful because it is fairly straightforward to implement, easy to customize, and makes mass balancing of the core straightforward. Some issues with the batchwise approach are that using large time steps will make the results more inaccurate, it has increased computational cost with smaller time steps, it has to rerun each time step for pre-reprocessing and post-reprocessing, and it is an approximation of the actual physical process of reprocessing. It is an approximation because, in online reprocessing of a reactor, the fission products are continuously removed

while any feed flows are continuously fed into the reactor.

Codes such as Serpent2 and newer versions of ChemTriton implement continuous reprocessing by adding terms to the Bateman equation similar to an extra decay term, as can be seen in Equation (2.5) [2, 14]. The benefits of using continuous reprocessing are that the model is more physically accurate, it doesn't impact computational cost as much as a batchwise process, and it allows for larger time steps to be used. However, the maximum time step size cannot become too large, or the simulation will become less accurate. This is because the neutron spectrum and cross section data does not update to the new material compositions until a new depletion step begins.

Another aspect of online reprocessing that must be considered is mass balancing. For batchwise reprocessing, mass balancing can be solved in several different ways. One method is to have the net mass of the feed rate over some depletion step be equivalent to the mass removed over that same depletion step, which is the current approach used by SaltProc [24]. Alternatively, the volume of the system can be adjusted so that constant density is maintained in order to keep cross section data consistent [21]. Another approach is to move excess fuel salt into a bleed-off tank [21]. For continuous reprocessing, the depletion time steps can be much larger. This means that during the interval between steps, the mass will not be balanced. However, the same batchwise methods can be used with continuous reprocessing at new depletion steps. Implementing these approaches increases computational cost when performed in Serpent2. This is because using a single input script with depletion steps will cause each depletion step to be run once, but altering the input script by changing volumes or removing mass to a bleed-off tank will cause the time step to be run twice.

2.2.1 Batchwise Reprocessing

Batchwise reprocessing starts with some removal rate where $X\%$ of some element is removed from the fuel salt in the system over some time period T . Over a depletion step τ , $\gamma\%$ of the element's mass is removed, where γ corresponds to X according to Equation (2.1). The scaling term γ for the removal efficiency is a linear approximation based on the time step adjustment in Equation (2.1). If the time step used causes the removal fraction to be greater than 1, the amount removed remains at 100% so as to not have negative mass.

An alternative approach is to establish a depletion time step τ such that the efficiency based time value T is a multiple of τ and only have the $X\%$ batchwise removal occur at the steps when τ is a multiple of T , thus removing the need for scaling.

$$\gamma = X \frac{\tau}{T} \quad (2.1)$$

For the Molten Salt Breeder Reactor (MSBR), SaltProc performs batchwise reprocessing every 3 days using a

linear approximation of the cycle times given by Robertson in the ORNL report [22, 24]. The xenon and krypton removals in SaltProc use a gas separation system equation, while the protactinium uses a modeled liquid-liquid reductive extraction process. These efficiencies differ from the cycle time values, causing slight differences in those terms. The MSBR also has two varying feed inputs of ^{232}Th and ^{233}U .

Examples of the batchwise method used in SaltProc can be seen in Equations (2.2) and (2.3). Both examples show what the resulting mass removal per step, or γ , term should be for different cycle time values. The first shows a cycle time of 20s, which is shorter than the step time of 3 days. This results in a removal rate over 100%, which is set to 100%. The second shows a cycle time of 60 days, longer than the step size. This results in a 5% removal per step. An alternative approach to the second example, which is implemented in the earlier versions of SaltProc, is to perform no removal until 60 days have elapsed and then remove 100% of material.

$$X = 100\%, T = 20\text{s}, \tau = 3d \rightarrow \gamma > 100\% \rightarrow 100\% \quad (2.2)$$

$$X = 100\%, T = 60d, \tau = 3d \rightarrow \gamma = 5\% \quad (2.3)$$

2.2.2 Continuous Reprocessing

The continuous approach manipulates the Bateman equation by adding a reprocessing term, which contains a reprocessing constant and the concentration of the isotope. This is one of the methods used in Serpent2 continuous reprocessing, an extension developed by Aufiero et al [2]. Equation (2.4) shows the generic form of the Bateman equation, while Equation (2.5) shows the terms added to Equation (2.4), which are the reprocessing constants, λ_r . These terms represent removal of the current isotope as well as feed of the current isotope from other sources.

$$\begin{aligned} \frac{dN_j}{dt}_{base} = & \sum_{i \neq j} [(\gamma_{i \rightarrow j} \sigma_{f,i} \Phi + \lambda_{i \rightarrow j} + \sigma_{i \rightarrow j} \Phi) N_i] \\ & - (\lambda_j + \sigma_j \Phi) N_j \end{aligned} \quad (2.4)$$

$$\frac{dN_j}{dt}_{net} = \frac{dN_j}{dt}_{base} - \lambda_{r,j} N_j + \sum_{mat} \lambda_{r,i \rightarrow j} N_i \quad (2.5)$$

The symbols given in the equations are defined as follows from Equation [15]:

N_j is the atomic density of isotope j .

$\gamma_{i \rightarrow j}$ is the fractional fission product yield of j in the fission of isotope i .

$\sigma_{f,i}$ is the microscopic fission cross section of isotope i .

Φ is the spectrum-averaged scalar flux in the fuel region.

$\lambda_{i \rightarrow j}$ is the decay constant of decay $i \rightarrow j$.

$\sigma_{i \rightarrow j}$ is the microscopic transmutation cross section of reaction $i \rightarrow j$.

N_i is the atomic density of isotope i .

λ_j is the decay constant of isotope j .

$\lambda_{r,j}$ is the reprocessing constant for removal of isotope j .

σ_j is the microscopic total transmutation cross section of isotope j .

$\lambda_{r,i \rightarrow j}$ is the reprocessing constant for feed of material $i \rightarrow j$.

Equation (2.5) shows that the reprocessing removal has the same mathematical operation as the decay rate. Unlike decay, the isotopes removed by reprocessing instead are transferred to a different material, which operates as a feed for that material. This can be seen in the summation term, which sums over the different materials and moves them from material i to material j .

Serpent2 has three different settings for continuous reprocessing which alter the effect on the Bateman equations. The zeroth setting does not conserve mass, so it will not be discussed. The first setting, which adds a decay-like term to the Bateman equation, is shown in Equations (2.4) and (2.5). This approach of adding a decay-like term to the Bateman equation has been done with other codes [14, 23]. For this setting, three different approaches are implemented in order to compare their results.

The first approach is labeled the Cycle Time Decay (CTD) approach. It takes half the value of the cycle times and treats them as half life values. A similar method of treating removal periods for reprocessing can be seen in Brovchenko et al [7]. Equation (2.6) shows how this approach works with a cycle time of 20 seconds. The operation cuts the cycle time in half and then calculates the decay constant, or reprocessing constant, using the half life from the cycle time.

$$20s \rightarrow \tau = 10s \rightarrow \lambda_r = \frac{\ln(2)}{\tau} \approx 6.93E-2 \quad (2.6)$$

The next approach is labeled the SaltProc Cycle Rate (SPCR) approach. It takes the same efficiency rates used by SaltProc, which uses 3 day depletion steps for the MSBR, and directly converts those rates to continuous reprocessing rates. Taking the inverse of the cycle time is a common way to generate the removal rate [24, 18].

Equation (2.7) shows how this approach works with a cycle time of 20s. This operation uses the SaltProc value, which in this case becomes 100% removal over 3 days, converts it to be per second, and then converts that to the rate form usable in Serpent2.

$$20s \rightarrow \frac{1}{3d} \rightarrow X = \frac{4E-6}{1s} \rightarrow \lambda_r = \ln\left(\frac{1}{1-X}\right) \approx 4E-6 \quad (2.7)$$

This natural logarithm in this approach comes from the differential equation solution based on the removal rate provided. The method used to derive this equation can be seen in Equations (2.8), (2.9), and (2.10). These equations assume that the X term, or the fractional removal rate, is in units of per second.

$$\frac{dN}{dt} = -\lambda_r N \quad (2.8)$$

$$N(t) = (1 - X)N_{cur} = N_{cur} e^{-\lambda_r} \quad (2.9)$$

$$-\ln(1 - X) = \lambda_r = \ln\left(\frac{1}{1 - X}\right) \quad (2.10)$$

The final approach is labeled the Cycle Rate (CR) approach. It takes the cycle times and uses a linear approximation to generate the efficiency rate directly. Equation (2.11) shows how this approach works using a cycle time of 20 seconds. This operation takes the cycle time, determines the efficiency rate for 100% removal over 20 seconds, and then uses that value to calculate the Serpent2 reprocessing constant.

$$20s \rightarrow X = \frac{1}{20s} \rightarrow \lambda_r = \ln\left(\frac{1}{1 - X}\right) \approx 5.12E-2 \quad (2.11)$$

The second continuous reprocessing setting used in Serpent2 removes a constant value from each isotopic Bateman equation for an element. This approach can be seen in Equation 2.12, where the C term represents the constant value being removed by continuous reprocessing.

$$\frac{dN_j}{dt} = \sum_{i \neq j} [(\gamma_{i \rightarrow j} \sigma_{f,i} \Phi + \lambda_{i \rightarrow j} + \sigma_{i \rightarrow j} \Phi) N_i] - (\lambda_j + \sigma_j \Phi) N_j - C \quad (2.12)$$

This approach has flaws which can be directly demonstrated through an example. Imagine an element exists with two isotopes A and B . There is a removal rate of 10% per second defined for the element. A batchwise process could remove 10% of each isotopes relative abundance after a second by using a one second time step, which is physical in terms of fractional removal. This can be seen directly in Table 2.1, where 10% of both isotopes and the

net count are removed.

Table 2.1: Serpent2 Second Setting Approaches

Labels	Initial	Batch	Net	A	B	Null
A	1000	900	949.5	900	999	990
B	10	9	-40.5	-90	9	0
Net	1010	909	909	810	1008	990

For the continuous reprocessing, the first attempt might be to try and remove 10% of the net by splitting it amongst the isotopes evenly. The issue with this approach is immediately noticeable, which is that the concentration of isotope *B* becomes negative, as can be seen in the *Net* column. A logical next approach would be to try and make one of the isotopes exactly correct while bringing the other along with it, which can be seen in the *A* and *B* columns. The results of this approach are seen in the table, and it can be seen that only by looking at the isotope with the smallest concentration and using that to determine the amount to remove can the result be ensured to be non-negative. One final approach to consider is to set the smallest concentration to 0, and bring the other isotopes along. This approach can be seen in the *Null* column, and it also does not work very well.

2.2.3 DNP Movement

The movement of DNPs in liquid fueled molten salt reactors impacts the operation of the reactor directly by reducing the number of neutrons which are present in the important regions of the reactor [28]. In a solid fueled reactor, the flux and the concentration of delayed neutron precursors have the same profile. However, a moving fluid fuel causes the precursors to have a shifted profile based on the fluid flow rate, mixing, and decay rate of the precursors. This was shown very clearly by Jun Shi and Massimiliano Fratoni in their work, which can be seen in Figure 2.1.

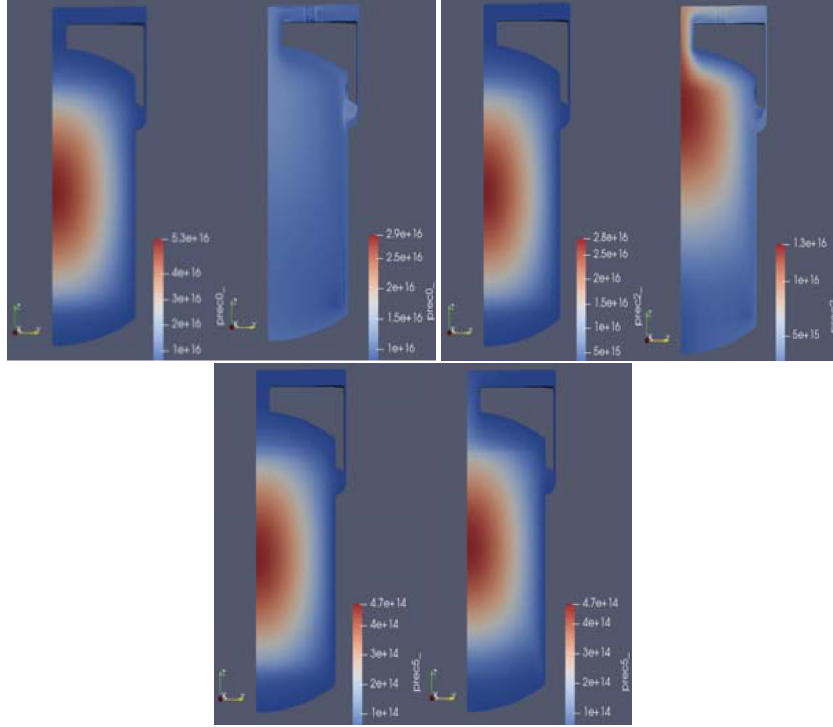


Figure 2.1: Plots of DNP concentrations in the molten salt reactor experiment [26]. The left side of each image shows concentration with no flow, while the right shows the concentration with a 1200 gallon per minute flow rate. Top Left) Longest lived DNP group. Top Right) Third longest lived DNP group. Bottom) Shortest lived DNP group.

From these images, it can be seen that for the long shortest lived DNPs, the movement of the fuel does not have any significant impact upon the DNP distribution when compared to static fuel. This is reasonable since the precursors do not live for very long in this group, so they do not have much chance to travel. For the intermediate DNP group, there is a clear shift in the direction of the fuel flow, which causes more precursors to produce delayed neutrons in the less important piping and upper plenum regions. The longest lived DNP group seems to be spread almost equally throughout the entire reactor, which means that those delayed neutrons are contributing significantly less to the overall neutron economy. Additionally, this reduces the overall effective delayed neutron fraction since the delayed neutrons are in less important regions. This directly affects the controllability of the reactor through the reactor period, which is heavily dependent upon the delayed neutrons.

2.3 Depletion

Depletion is the process of burning the fuel and simulating the changes to the materials this causes in the system. Depletion requires specific information to run properly, such as decay constants, fission yields, and fission and transmutation cross section data. The fission and transmutation cross section data can come from a transport

calculation, while the other data can be read from a data library [15]. Additionally, running depletion involves solving the Bateman equations, a generic version of which can be seen in Equation 2.4.

There are several different approaches to solving these equations, such as the Transmutation Trajectory Analysis (TTA) method and the Chebyshev Rational Approximation Method (CRAM). These are the two different methods used in Serpent2, and are both built into Serpent2 [16]. Other codes may use external software to compute depletion, such as REM used by MCNP and ORIGEN-S used by KENO-VI [2]. Another code, aside from Serpent2, that contains built-in depletion solvers is ERANOS [2].

The purpose behind solving these equations and having depletion models is to generate an accurate representation of the composition of the target materials during reactor operation. This is important in determining how long the reactor can run, how the safety of the reactor develops over time, and how operation may have to change to adjust to the new reactor state. For molten salt reactors, depletion also allows for information regarding fresh fuel feed rate and fission product removal rates, as well as how variations of those parameters can affect reactor performance and behaviour.

2.4 The Molten Salt Breeder Reactor

The molten salt breeder reactor (MSBR) is a useful design to analyze for several reasons. Because it is a molten salt reactor, online reprocessing is used within the design. In addition, it contains an inner and outer zone, each of which has different neutronic behaviour [22]. More specifically, the inner zone has a softer spectrum, a higher fission rate, and a 13% fuel-to-graphite ratio [19]. The outer zone has a harder spectrum, a higher breeding rate, and a 37% fuel-to-graphite ratio. The inner and outer zones prove to be an issue when accurately modeling using a unit-cell or one-region approach, as those models are unable to capture the different characteristics of each region [24]. The differences in the regions can also be seen in Figure 2.2, which allows for the difference in the fuel-to-graphite ratio to be visually noticeable.

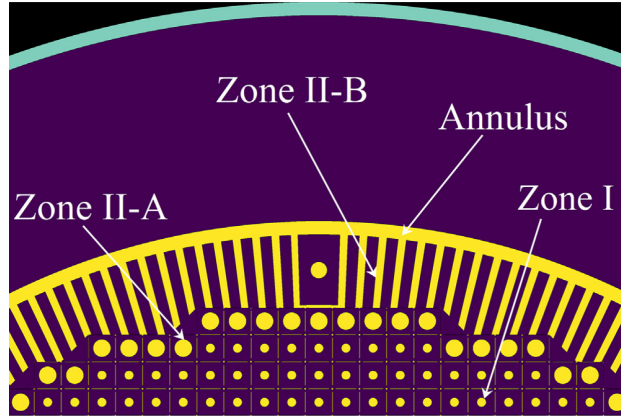


Figure 2.2: MSBR core axial slice showing the different regions from [24]. Zones II-A and II-B are where the spectrum is harder and there is increased breeding. Zone I is where there is more fission and a softer spectrum. The yellow is fuel salt, the purple is graphite, and the cyan is the reactor vessel.

For the online reprocessing aspect of the reactor, there are several mechanisms that contribute, as can be seen in Figure 2.3. This figure shows how the reactor (1) links with the fuel salt pump (3), entrainment separator (14), and gas separator (13). The elements which are to be extracted, as well as their cycle times, are shown in Table 2.2.

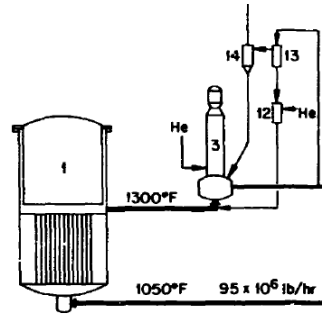


Figure 2.3: MSBR simplified flow diagram from [22] showing part of primary side.

Table 2.2: MSBR Online Reprocessing Cycle Times [22]

Reprocessing Group	Element(s)	Cycle Time (Full Power)
Rare Earths	Y, La, Ce, Pr, Nd, Pm, Sm, Gd	50 days
Rare Earths	Eu	500 days
Noble Metals	Se, Nb, Mo, Tc, Ru, Rh, Pd, Ag, Sb, Te	20 seconds
Seminoble Metals	Zr, Cd, In, Sn	200 days
Gases	Kr, Xe	20 seconds
Volatile Fluorides	Br, I	60 days
Discard	Rb, Sr, Cs, Ba	3435 days
Salt Discard	Th, Li, Be, F	3435 days
Protactinium	Pa (233)	3 days
Higher Nuclides	Np (237), Pu (242)	16 years

From SaltProc's MSBR example, the removal of each group can be attributed to a particular process within the physical system [24]. The entrainment separator is used to remove the gases; a nickel mesh is used to remove noble metals, seminoble metals, and volatile fluorides; and various processes given by Robertson are lumped together into the liquid metal process which removes rare earths, discard, and protactinium [22]. The higher nuclides are removed at such an infrequent rate that they are not included within the SaltProc processes. The MSBR design removes the protactinium with the intent that the protactinium-233 will not absorb neutrons and will instead be fed out of the core to decay into uranium-233, which can then be fed back into the core.

2.5 MSR Modeling Approaches

For modeling of MSRs, there are generally two different models which are used. The first is a model which takes a given fuel composition and analyzes the performance of the reactor given that fuel composition. This may include focus on DNP movement [9, 26] or reactor dynamics [27, 8, 3].

The second type of model is one which depletes the fuel. Because the online reprocessing of an MSR has such

large effects on its fuel composition, depletion models of MSRs typically incorporate online reprocessing. These models can also account for other factors, such as DNP movement, as shown by Zhou et al [29]. The movement of DNP can have a noticeable effect on reactor performance by affecting neutron energy spectra and distribution of delayed neutrons [5]. Most, however, focus primarily on depletion and do not consider the DNP movement during depletion, as can be seen in Table 2.3. In order to handle spatial dependence of fuel salt without modeling the piping of the core, models can use the scaled flux method [4].

For all of the continuous reprocessing models given in Table 2.3, all of them use the "fictitious decay constant" approach, which is shown in Equation (2.5). For the batchwise models, they typically use the approach shown in Equation (2.1) and remove a fraction at each depletion step rather than bulk removal at certain steps.

Table 2.3: Molten Salt Reactor Models with Online Reprocessing

Index	Reprocessing	Model	Reactor	DNP	Ref.
1	Batchwise	Full 3D	MSBR	No	[24]
2	Continuous	Full 3D	MSFR	No	[2]
3	Batchwise	Full 3D	MSBR	No	[19]
4	Batchwise	Unit Cell	MSBR	No	[6]
4*	Continuous	Unit Cell	N/A	No	[14]
5	Batchwise	Inf Lattice	DMSR	No	[1]
6	Continuous	Full 3D	MSBR	Yes	[29]
7	Batchwise	Full 3D	MSBR	No	[18]
8	Batchwise	Full 3D	MOSART	No	[25]
9	Continuous	Inf Lattice	ADNA	No	[23]
10	Mixed	2D Slice	MSFR	No	[10]
11	Continuous	Full 3D (?)	MSFR	No	[30]
12	Batchwise	Full 3D	Toy	No	[21]
13	Mixed	2D Slice	N/A	Yes	[13]
14	Mixed	Full 3D	TMSR	No	[17]

The first entry in Table 2.3 is SaltProc's MSBR example, which uses Serpent2, a full 3D core, 3 day time steps, and runs for 60 years. The batchwise reprocessing in SaltProc includes both the removal and feed processes.

The second entry is by Aufiero et al, and displays the Serpent2 newly developed continuous reprocessing functionality. A full 3D simplified core layout of the molten salt fast reactor (MSFR) is used, and continuous reprocessing is used to operate the removal and feed processes. It evaluates the model at various scales, from steady state uranium concentrations at over 60 years to radiotoxicity over 10^8 years. The step sizes vary as well, though not specifically mentioned, appearing very small at BOC and expanding to around 10 years at SS for one of the figures presented.

The third entry by Park et al is very similar to the MSBR example in SaltProc, with the primary difference being MCNP6 is used for transportation calculations instead of Serpent2. The depletion was performed using CINDER90 and custom Python scripts. Another difference is that SaltProc uses adjusted values for the removal efficiency rates of xenon, krypton, and protactinium, while the work by Park et al uses the cycle time values from Robertson [22]. This model uses 3 day time steps and models the fuel composition over 20 years.

The fourth entry is for ChemTriton, a tool designed for SCALE to perform online reprocessing of MSRs. It contains work on multiple reactors, such as the MSBR and the MSRE. The tool iteratively runs SCALE/TRITON over small time steps to simulate continuous reprocessing. For the MSBR case, 3 day time steps are used, similar to SaltProc and Park et al. The reason for not using shorter time steps is that using shorter time steps increases computational cost while also providing very little improvements to calculated eigenvalue and fuel cycle metrics, discussed by Betzler et al as well as by Powers et al [20].

ChemTriton has been updated after TRITON implemented new tools for continuous material feeds and removals. The work discusses in depth how the removal rates for isotopes can be calculated using a well mixed approximation and accounting for non-modeled piping by introducing a correcting decay factor. Although it does not account for DNP movement, it does account for neutron important regions without having to model the piping regions.

The fifth entry by Ahmad et al analyzes single fluid and two fluid denatured molten salt reactors (DMSRs). This work uses MCNP5 and ORIGEN2 as the codes to perform transport and depletion over 30 years. Although the steps at which fuel is adjusted is at 5 year intervals, the burnup intervals are actually set to 6 months. The reason such large steps were used is because the work analyzes proliferation and resource requirements for the reactors. This means that the neutronics and safety parameters were not as important, and so the model did not have to be as precise in that regard. Additionally, only gaseous fission products or fission products which are removed via sparging are included in the reprocessing scheme, no other chemical extraction takes place. This is because the authors wanted the system to be as simple as possible. ORIGEN provides continuous reprocessing capability, however for this work it would not have been necessary since the authors wanted to use a simpler approach for the

model [11].

The sixth entry by Zhou et al is for FAMOS (Fuel cycle Analysis code for MOlten Salt reactors). This code uses OpenMC to generate homogenized cross section data followed by the DIF3D diffusion solver along with extensions to model DNP movement, thermal hydraulics, depletion, and continuous reprocessing. Their work also analyzes the molten salt fast reactor over 200 operational years. The depletion step size is not specifically given in the work, but from analysis of the plots, small time steps seem to have been used at the beginning of cycle and then larger time steps of 100 days are then used for the rest of the steps.

The seventh entry by Nuttin et al analyzes the MSBR using a slightly simplified geometry with MCNP. The reprocessing approach used here is also the same as the approach used in SaltProc, which is to take the inverse of the cycle time to determine the removal rate. Depletion is performed in this work by coupling MCNP to an evolution program and the NJOY code. However, this work uses 10 day steps instead of 3 day steps that SaltProc uses. It also runs for 100 years instead of 20.

The eighth entry by Sheu et al analyzes the Molten Salt Actinide Recycler & Transmuter (MOSART) reactor and uses custom scripts to implement batchwise reprocessing. The codes used are SCALE6 for transport and TRITON for depletion. The step sizes used include 15, 30, and 60 days, in which the authors found that 30 days gave the best results in terms of accuracy and run time balancing for the 30 year simulation.

The ninth entry by Rodriguez-Vieitez analyzes the Accelerator-Driven Neutron Applications (ADNA) Tier-1 reactor using MCNP. Although the model used does implement continuous reprocessing in the Bateman equation, it is not used for the purpose of fuel cycle analysis and depletion. Rather, the work iteratively works towards the equilibrium fuel composition in order to analyze the effect of varying different reactor parameters, such as geometry.

The tenth entry by Fiorina et al is on the Molten Salt Fast Reactor (MSFR) and uses a modified version of the ERANOS-based EQL3D procedure. This model makes use of continuous reprocessing for non-soluble fission products, but uses batchwise reprocessing for soluble fission products. The batchwise reprocessing was incorporated for simplicity in the model. The time steps used are shorter at beginning of cycle and extend out to 30 year steps for a 200 year depletion. Additionally, the radiotoxicity analysis extends out to 1 million years.

The eleventh entry by Zhuang et al discusses an OpenMC extension called OpenMCB-MSR to implement continuous reprocessing in an analysis of the MSFR. The depletion step sizes used are not specified in the work, though with continuous reprocessing step sizes could have been set to above single digit days and still maintained high precision.

The twelfth entry by Ridley and Chvala use batchwise reprocessing on a toy MSR using Serpent2 and a python library. The feed rates were set to fluctuate in order to maintain criticality while also accounting for density imbalance in Serpent2 by increasing the volume of the material. The full core 3D model was then analyzed using 7

day depletion steps for 10 years.

The thirteenth entry is on the Reactor Optimum Design (ROD) code, which allows users to use continuous or batchwise reprocessing for the MSR model. ROD operates by taking a one or two dimensional input and determining the equilibrium fuel composition. This is a useful code for determining a prediction of a model, but is less precise than more modern codes which offer more detailed geometry and other features.

The fourteenth entry by Merle-Lucotte et al covers depletion of the Thorium Molten Salt Reactor (TMSR) using MCNP4 [12] for transport and REM for material evolution. Continuous reprocessing is used for gaseous fission product extraction, while the other fission products are processed with batchwise reprocessing.

Appendix

Appendix.

References

- [1] Ahmad, A., McClamrock, E. B., and Glaser, A. (2015). Neutronics calculations for denatured molten salt reactors: Assessing resource requirements and proliferation-risk attributes. *Annals of Nuclear Energy*, 75:261–267.
- [2] Aufiero, M., Cammi, A., Fiorina, C., Leppänen, J., Luzzi, L., and Ricotti, M. (2013). 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(1-3):473–486.
- [3] Aufiero, M., Cammi, A., Geoffroy, O., Losa, M., Luzzi, L., Ricotti, M. E., and Rouch, H. (2014). Development of an OpenFOAM model for the Molten Salt Fast Reactor transient analysis. *Chemical Engineering Science*, 111:390–401.
- [4] Betzler, B. (2021). Liquid-fueled Molten Salt Reactor Depletion Modeling. Technical report, Oak Ridge National Lab. (ORNL), Oak Ridge, TN (United States).
- [5] Betzler, B., Powers, J., and Brown, N. (2017a). Implementation of Molten Salt Reactor Tools in SCALE. *undefined*.
- [6] Betzler, B. R., Powers, J. J., and Worrall, A. (2017b). Molten salt reactor neutronics and fuel cycle modeling and simulation with SCALE. *Annals of Nuclear Energy*, 101:489–503.
- [7] Brovchenko, M., Kloosterman, J.-L., Luzzi, L., Merle, E., Heuer, D., Laureau, A., Feynberg, O., Ignatiev, V., Aufiero, M., Cammi, A., Fiorina, C., Alcaro, F., Dulla, S., Ravetto, P., Frima, L., Lathouwers, D., and Merk, B. (2019). Neutronic benchmark of the molten salt fast reactor in the frame of the EVOL and MARS collaborative projects. *EPJ Nuclear Sciences & Technologies*, 5:2.
- [8] Cervi, E., Lorenzi, S., Cammi, A., and Luzzi, L. (2019). Development of a multiphysics model for the study of fuel compressibility effects in the Molten Salt Fast Reactor. *Chemical Engineering Science*, 193:379–393.
- [9] Fei, T., Feng, B., and Heidet, F. (2020). Molten salt reactor core simulation with PROTEUS. *Annals of Nuclear Energy*, 140:107099.
- [10] Fiorina, C., Cammi, A., Krepel, J., Mikityuk, K., and Ricotti, M. E. (2012). Preliminary Analysis of the MSFR Fuel Cycle Using Modified-EQL3D Procedure. In *Volume 4: Codes, Standards, Licensing, and Regulatory Issues; Fuel Cycle, Radioactive Waste Management and Decommissioning; Computational Fluid Dynamics (CFD) and Coupled Codes; Instrumentation and Co*, page 293, Anaheim, California, USA. ASME.
- [11] Gauld, I. C., Radulescu, G., Ilas, G., Murphy, B. D., Williams, M. L., and Wiarda, D. (2011). Isotopic Depletion and Decay Methods and Analysis Capabilities in SCALE. *Nuclear Technology*, 174(2):169–195.
- [12] Goluoglu, S. and McClure, J. (1998). SOFTWARE QUALIFICATION REPORT for MCNP Version 4B2.
- [13] H. F. Bauman, G. W. Cunnningham III, J. L. Lucius, H. T. Kerr, and C. W. Craven, Jr. (1971). ROD: a nuclear and fuel-cycle analysis code for circulating-fuel reactors.
- [14] Jr. Vicente Valdez, P., Betzler, B., Wieselquist, W., and Fratoni, M. (2020). Modeling Molten Salt Reactor Fission Product Removal with SCALE. Technical Report ORNL/TM-2019/1418, 1608211, ORNL.
- [15] Leppänen, J. (2007). Development of a New Monte Carlo Reactor Physics Code. *VTT Publications*, 640:241.

- [16] Leppänen, J., Pusa, M., Viitanen, T., Valtavirta, V., and Kaltiaisenaho, T. (2015). The Serpent Monte Carlo code: Status, development and applications in 2013. *Annals of Nuclear Energy*, 82:142–150.
- [17] Merle-Lucotte, E., Heuer, D., Allibert, M., Ghetta, V., Brun, C. L., Brissot, R., Liatard, E., and Mathieu, L. (2007). THE THORIUM MOLTEN SALT REACTOR: LAUNCHING THE THORIUM CYCLE WHILE CLOSING THE CURRENT FUEL CYCLE. In *European Nuclear Conference*, volume European Nuclear Conference, pages 48–53, European Nuclear Conference.
- [18] Nuttin, A., Heuer, D., Billebaud, A., Brissot, R., Brun, C. L., Liatard, E., Meplan, O., Merle-Lucotte, E., Nifenecker, H., and Perdu, F. (2005). POTENTIAL OF THORIUM MOLTEN SALT REACTORS : DETAILED CALCULATIONS AND CONCEPT EVOLUTION WITH A VIEW TO LARGE SCALE ENERGY PRODUCTION. *Progress in Nuclear Energy*, 46(1):23.
- [19] Park, J., Jeong, Y., Lee, H. C., and Lee, D. (2015). Whole core analysis of molten salt breeder reactor with online fuel reprocessing: Whole core analysis of MSBR with online fuel reprocessing. *International Journal of Energy Research*, pages n/a–n/a.
- [20] Powers, J. J., Harrison, T. J., and Gehin, J. C. (2013). A new approach for modeling and analysis of molten salt reactors using SCALE. Technical report, American Nuclear Society - ANS; La Grange Park (United States).
- [21] Ridley, G. and Chvala, O. (2017). A method for predicting fuel maintenance in once-through MSRs. *Annals of Nuclear Energy*, 110:265–281.
- [22] Robertson, R. (1971). CONCEPTUAL DESIGN STUDY OF A SINGLE-FLUID MOLTEN-SALT BREEDER REACTOR. Technical Report ORNL-4541, 4030941, ORNL.
- [23] Rodriguez-Vieitez, E., Lowenthal, M., Greenspan, E., and Ahn, J. (2002). TRANSMUTATION CAPABILITY OF ONCE-THROUGH CRITICAL OR SUB-CRITICAL MOLTEN-SALT REACTORS. In *National Research Council*.
- [24] Rykhlevskii, A., Bae, J. W., and Huff, K. D. (2019). Modeling and simulation of online reprocessing in the thorium-fueled molten salt breeder reactor. *Annals of Nuclear Energy*, 128:366–379.
- [25] Sheu, R., Chang, C., Chao, C., and Liu, Y.-W. (2013). 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.
- [26] Shi, J. and Fratoni, M. (2021). Gen-foam Model and Benchmark of Delayed Neutron Precursor Drift in the Molten Salt Reactor Experiment. *EPJ Web of Conferences*, 247:06040. JC0007.
- [27] Singh, V., Wheeler, A. M., Upadhyaya, B. R., Chvála, O., and Greenwood, M. S. (2020). Plant-level dynamic modeling of a commercial-scale molten salt reactor system. *Nuclear Engineering and Design*, 360:110457.
- [28] Wooten, D. and Powers, J. J. (2018). A Review of Molten Salt Reactor Kinetics Models. *Nuclear Science and Engineering*, 191(3):203–230.
- [29] Zhou, S., Yang, W. S., Park, T., and Wu, H. (2018). Fuel cycle analysis of molten salt reactors based on coupled neutronics and thermal-hydraulics calculations. *Annals of Nuclear Energy*, 114:369–383.
- [30] Zhuang, K., Li, T., Zhang, Q., He, Q., and Zhang, T. (2020). Extended development of a Monte Carlo code OpenMC for fuel cycle simulation of molten salt reactor. *Progress in Nuclear Energy*, 118:103115.