ANALYSIS OF REPROCESSING METHODS IN THE MOLTEN SALT BREEDER REACTOR

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:

Madicken Munk
Tomasz Kozlowski

Abstract

Molten salt reactors have had many depletion models generated since being announced as a generation IV reactor. Among these models, one important aspect which is considered is the online reprocessing capabilities of a liquid fueled molten salt reactor. This includes adding fresh fuel and removing fission products from the reactor during operation, allowing for a more efficient neutron economy. In order to simulate reprocessing, there are two different mathematical approaches which are implemented, called batchwise and continuous reprocessing. For online reprocessing, the continuous reprocessing method is more physically reflective of the operation in reality. However, there are many works which have implemented batchwise reprocessing to simulate a continuous reprocessing process. This work shows that continuous reprocessing and batchwise reprocessing methods are not interchangeable, and there is a fairly large difference in the results when using the two different approaches for the same system. This work also shows the computational cost and a depletion step mesh refinement for both continuous and batchwise reprocessing. From those results, it is determined that continuous reprocessing allows for significantly larger steps without large increases in error, which enables drastically reduced computational cost on the order of one magnitude. The effect of density increase when using the Serpent2 code due to continuous reprocessing is also investigated, and it is determined that for typical operations, the density change is roughly 0.12%.

Acknowledgments

This material is based upon work supported under an Integrated University Program Graduate Fellowship.

Table of Contents

List of Tables . . . . . **vi**

List of Figures . . . . . **vii**

Chapter 1 Introduction **1**
1.1 Energy 1
1.2 Nuclear Energy 2
1.3 Molten Salt Reactors 3
1.4 Reprocessing Methods 4

Chapter 2 Literature Review **5**
2.1 Molten Salt Reactors 5
2.1.1 Online Reprocessing 5
2.1.2 DNP Movement 10
2.2 Depletion 11
2.3 The Molten Salt Breeder Reactor 11
2.4 MSR Modeling Approaches 18

Chapter 3 Methods **24**
3.1 Batchwise Reprocessing 24
3.1.1 Bulk 24
3.1.2 Steady 26
3.1.3 Batch Approaches Summary 27
3.2 Continuous Reprocessing 34
3.2.1 Constant 34
3.2.2 Decay 34
3.2.3 Step 42
3.3 Mass Balancing 43
3.4 Effects of Delayed Neutrons on Depletion 44
3.5 Serpent MSBR Model 45
3.5.1 Reprocessing Structure 45

Chapter 4 Results **47**
4.1 Depletion Step Mesh Refinement 47
4.1.1 Batchwise Reprocessing 48
4.1.2 Continuous Reprocessing 53
4.1.3 Variable Depletion Step Size 59
4.1.4 Computational Cost Analysis 60
4.2 Advanced Protactinium Decay Model 61
4.3 Comparisons of Methods 64
4.3.1 Continuous Methods 65
4.3.2 Continuous and Batchwise Methods 66
4.3.3 Overall Differences 72

4.4 Mass Balancing 74
4.4.1 Impact Minimization Using Modified Feeds 75
4.4.2 Actual Impact 75

Chapter 5 Conclusions 77
5.1 Comparison of Continuous and Batchwise Reprocessing Methods 77
5.2 Continuous Reprocessing Investigations 77
5.3 Future Work 77
5.4 Derivation of Batchwise Reprocessing Steady State Mass 79

References 82

List of Tables

1.1 Mortality rates for each energy source in deaths per billion kWh produced (recreated from [10]). . . . 2

2.1 Serpent2 Second Setting Approaches . . . 9
2.2 MSBR General Data [37] . . . 13
2.3 MSBR Primary System Salt Inventory [37] . . . 14
2.4 MSBR Online Reprocessing Cycle Times [37] . . . 15
2.5 Molten Salt Reactor Models with Online Reprocessing . . . 20

3.1 Batchwise Reprocessing Methods . . . 27
3.2 Subset of Decay Reprocessing Approaches . . . 40
3.3 Full Set of Decay Reprocessing Approaches for Various Cycle Times . . . 42

4.1 Variable Depletion Step Size Results . . . 60
4.2 Computation Cost Using Constant Depletion Steps . . . 61

List of Figures

1.1 Energy generation in the United States in 2021 from the US Energy Information Administration [46]. 1

2.1 Simplified model of how Bateman equation relates to mass flow rates. 8
2.2 Plots of DNP concentrations in the molten salt reactor experiment [43]. 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. 10
2.3 MSBR core axial slice showing the different regions from [41]. 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. 12
2.4 MSBR protactinium processing scheme from Robertson et al [37]. 16
2.5 MSBR rare earth processing scheme from Robertson et al [37]. 16
2.6 Combined MSBR rare earth and protactinium processing schemes from Robertson et al [37]. 17
2.7 MSBR noble metal flux and off-gas system from Robertson et al [37]. 18

3.1 Thorium feed rate in the MSBR as a function of time while using bulk batchwise reprocessing [39]. . 25
3.2 Uranium feed rate in the MSBR as a function of time while using bulk batchwise reprocessing [39]. . 25
3.3 Thorium feed rate with a depletion time step of 3 days in the MSBR as a function of time while using steady batchwise reprocessing. 26
3.4 Uranium feed rate with a depletion time step of 3 days in the MSBR as a function of time while using steady batchwise reprocessing. 27
3.5 Plot showing how bulk and steady batchwise reprocessing removal works as a function of time for an example with a cycle time of 20 seconds. 28
3.6 Plot showing how bulk and steady batchwise reprocessing are identical while the cycle time is shorter than or equal to the depletion step size. 28
3.7 Plot showing how bulk and steady batchwise reprocessing are different while the cycle time is longer than the depletion step size for a 40 second cycle time and 20 second depletion step (top) and a 20 second cycle time with a 1 second depletion step (bottom). 29
3.8 Plot comparing the values of continuous and batchwise reprocessing methods with a 40 second cycle time. 32
3.9 A comparison of the Direct Linear and Cycle Rate reprocessing constants for different cycle times. . . 40
3.10 Plot of how reprocessing constants for different approaches vary with cycle time. 41
3.11 Simplified reprocessing scheme based on mimicking SaltProc. 45
3.12 Simplified reprocessing scheme based on the phyiscal MSBR processes. 46

4.1 k_{eff} over time using various depletion step sizes with steady batchwise reprocessing. 49
4.2 Thorium-232 mass over time using various depletion step sizes with steady batchwise reprocessing. 49
4.3 Uranium-233 mass over time using various depletion step sizes with steady batchwise reprocessing. 50
4.4 Xenon-135 mass over time using various depletion step sizes with steady batchwise reprocessing. . . 50
4.5 Xenon-136 mass over time using various depletion step sizes with steady batchwise reprocessing. . . 51
4.6 Thorium-232 feed rate over time using various depletion step sizes with steady batchwise reprocessing. 51
4.7 Uranium-233 feed rate over time using various depletion step sizes with steady batchwise reprocessing. 52

4.8 Xenon-135 removal rate over time using various depletion step sizes with steady batchwise reprocessing. 52
4.9 Xenon-136 removal rate over time using various depletion step sizes with steady batchwise reprocessing. 53
4.10 k_{eff} over time using various depletion step sizes with direct linear continuous reprocessing. 54
4.11 Thorium-232 mass over time using various depletion step sizes with direct linear continuous reprocessing. 54
4.12 Uranium-233 mass over time using various depletion step sizes with direct linear continuous reprocessing. 55
4.13 Xenon-135 mass over time using various depletion step sizes with direct linear continuous reprocessing. 55
4.14 Xenon-136 mass over time using various depletion step sizes with direct linear continuous reprocessing. 56
4.15 k_{eff} over time using various depletion step sizes with direct linear continuous reprocessing. 56
4.16 Thorium-232 mass over time using various depletion step sizes with direct linear continuous reprocessing. 57
4.17 Uranium-233 mass over time using various depletion step sizes with direct linear continuous reprocessing. 58
4.18 Xenon-135 mass over time using various depletion step sizes with direct linear continuous reprocessing. 58
4.19 Xenon-136 mass over time using various depletion step sizes with direct linear continuous reprocessing. 59
4.20 k_{eff} over time using various depletion step sizes with direct linear continuous reprocessing. 62
4.21 Thorium-232 mass over time using various depletion step sizes with direct linear continuous reprocessing. 62
4.22 Uranium-233 mass over time using various depletion step sizes with direct linear continuous reprocessing. 63
4.23 Xenon-135 mass over time using various depletion step sizes with direct linear continuous reprocessing. 64
4.24 Xenon-136 mass over time using various depletion step sizes with direct linear continuous reprocessing. 64
4.25 k_{eff} over time comparing different continuous reprocessing methods. 65
4.26 k_{eff} over time comparing bulk batchwise results to continuous methods. 67
4.27 Thorium-232 mass over time comparing bulk batchwise results to continuous methods. 67
4.28 Uranium-233 mass over time comparing bulk batchwise results to continuous methods. 68
4.29 Xenon-135 mass over time comparing bulk batchwise results to continuous methods. 68
4.30 Xenon-136 mass over time comparing bulk batchwise results to continuous methods. 69
4.31 k_{eff} over time comparing steady batchwise results to continuous methods. 69
4.32 Thorium-232 mass over time comparing steady batchwise results to continuous methods. 70
4.33 Uranium-233 mass over time comparing steady batchwise results to continuous methods. 70
4.34 Xenon-135 mass over time comparing steady batchwise results to continuous methods. 71
4.35 Xenon-136 mass over time comparing steady batchwise results to continuous methods. 71
4.36 k_{eff} over time with three day depletion steps for steady batchwise and Direct Linear continuous reprocessing using different uranium-233 feed schemes. 72
4.37 ^{232}Th over time with three day depletion steps for steady batchwise and Direct Linear continuous reprocessing using different uranium-233 feed schemes. 73
4.38 ^{233}U over time with three day depletion steps for steady batchwise and Direct Linear continuous reprocessing using different uranium-233 feed schemes. 73
4.39 ^{135}Xe over time three day depletion steps for steady batchwise and Direct Linear continuous reprocessing using different uranium-233 feed schemes. 74
4.40 ^{136}Xe over time three day depletion steps for steady batchwise and Direct Linear continuous reprocessing using different uranium-233 feed schemes. 74
4.41 Change in net mass of the fuel salt for various methods. 76

Chapter 1

Introduction

1.1 Energy

In modern society, scientific advancements and technological developments have improved quality of life drastically over time. These commonly used technologies, such as air conditioning or food production, require an energy input in order to function properly. To cool the air, energy is required in order to compress, pressurize, and cool the refrigerant. To produce food, energy is required to pump water, create fertilizer, and operate heavy machinery. Energy, typically in the form of electricity, is significant in maintaining and developing quality of life for society.

However, energy is not free, and it takes effort to harvest and convert energy into usable forms. For wind power, this means building large turbines which can be turned by the wind. For solar power, this means building solar panels, which convert photons from the sun into electrical energy. However, for most large-scale electrical generation needs, it is more efficient for the conversion aspect to use boiling water to turn a turbine, which then generates electricity, as can be seen in Figure 1.1. The methods shown which use boiling water account for over 75% of the total energy generation in the United States.

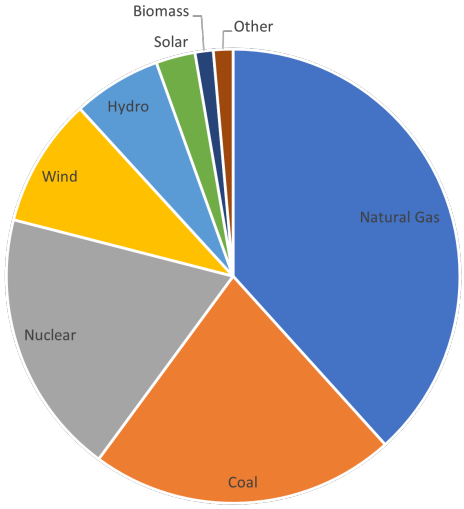

Figure 1.1: Energy generation in the United States in 2021 from the US Energy Information Administration [46].

Figure 1.1 shows the different methods of energy generation in the United States. From this figure, it can be seen that the methods which dominate energy contribution are natural gas, coal, and nuclear. Of the various methods to generate electricity, desirable characteristics are safety, efficiency, consistency, availability, and economics. Nuclear energy contains several of these characteristics: safety can be seen in Table 1.1, thermal efficiency from the high temperatures which are generated, and consistency by providing a constant supply of energy except during scheduled outages. Nuclear energy has many positives, though there is an economic problem due to high up-front costs, which are on the order of $10 billion for a 1GW nuclear power plant [15]. Additionally, availability is lowered due to issues such as siting, public perception, and construction issues.

1.2 Nuclear Energy

Part of the reason nuclear energy has high up-front costs is because the nuclear industry is highly safety conscious and has many regulations in place in order to minimize risks. Although this mindset increases costs, the effort has paid dividends, which can be seen in Table 1.1. This table shows mortality rate normalized to energy produced and shows that nuclear energy is one of the safest energy sources available.

Table 1.1: Mortality rates for each energy source in deaths per billion kWh produced (recreated from [10]).

Energy Source	Mortality Rate (deaths per billion kWh)
Coal (Global Average)	100
Biofuel/Biomass	24
Coal (U.S.)	15
Oil	36
Natural Gas	4
Hydro (Global Average)	1.4
Solar (Rooftop)	0.44
Wind	0.15
Nuclear (Global Average)	0.04

Even though the up-front costs are high, nuclear plants are a long term investment. Nuclear power plants are

licensed to operate for 40 years and can get extensions to extend that time by 10-20 years [9]. An investment at this time scale is difficult to make for a private industry though, as there is less risk when investing in cheaper plants, such as natural gas.

In order to address the economic issue of nuclear power plants, there are many different approaches which are implemented. One of these is to build smaller, cheaper nuclear plants which are then more affordable and are have less up-front cost. Another approach is from the regulation perspective, and involves either taxing negative externalities more harshly or subsidizing carbon-free energy sources. There is also the approach of developing generation IV reactor designs [25]. The goal of these generation IV reactors are sustainability, economics, safety and reliability, and proliferation resistance and physical protection [25].

1.3 Molten Salt Reactors

One of the new generation IV reactor designs is the molten salt reactor, or MSR [25]. The MSR potentially offers a lower cost of electricity when compared to coal and pressurized water reactors, which are a large portion of the nuclear fleet [31]. The liquid fueled MSR design is significantly different from the current light water reactor designs which dominate the nuclear fleet currently in use. Light water reactors include boiling water reactors and pressurized water reactors, both of which are well understood reactor designs. These light water reactors use a solid fuel containing a fissile nuclide, typically ^{235}U, with a water coolant. This water coolant also serves as a moderator to slow down the neutrons to thermal energies. This fuel generates heat as the fissile nuclide fissions. After roughly one year of depletion, it can be reshuffled or removed from the reactor to be placed into a cooling pool [26].

MSRs, instead of a solid fuel, can use a liquid fuel composed of a molten salt mixed with some fissile nuclide. Using a liquid fuel allows for online reprocessing, which is a continuous process which can involve adding fresh fuel to the system as well as chemically removing undesirable products which are generated through fission. The system also allows for batchwise reprocessing, which is a process which occurs in discrete steps, such as salt disposal. Typically, most reprocessing involved in an MSR is continuously performed while the reactor is online.

Online reprocessing is useful because the reactor can operate continuously without needing to shut down for refueling and does not require excess reactivty to maintain operation. In a typical LWR, burnable poisons and control rods are used to lower reactivity initially so the reactor can burn longer. In a liquid fueled MSR, the parasitic absorbers, such as ^{135}Xe for a thermal energy spectrum, can be removed from the reactor. Additionally, fresh fuel salt can be added to the reactor in order to maintain stable operation.

This fresh fuel feed can include fissile nuclides, such as ^{233}U or ^{235}U, fertile nuclides, such as ^{238}U or ^{232}Th, or some combination of both. Adding a fissile nuclide to the reactor will allow for the reactor to continue fission when

the atom is fissioned by a neutron. For the fertile nuclides, the neutron must first be absorbed and the product decays, leading to a fissile nuclide. This process can be seen in Equation (1.1), which is the breeding process used in the Molten Salt Breeder Reactor, or MSBR [37].

$$^{232}_{90}\mathrm{Th} + ^{1}_{0}\mathrm{n} \longrightarrow ^{233}_{90}\mathrm{Th} \longrightarrow ^{233}_{91}\mathrm{Pa} \longrightarrow ^{233}_{92}\mathrm{U} \tag{1.1}$$

The half life of ^{232}Th via beta decay is on the order of tens of minutes, while the half life of ^{233}Pa via beta decay is roughly 27 days. Because the beta decay of ^{233}Pa is needed in order to generate more fissile material for the core, allowing the ^{233}Pa to capture neutrons negatively impacts the reactor performance. Specifically, the thermal energy capture cross section of ^{233}Pa is roughly five times greater than that of ^{232}Th, meaning that the ^{233}Pa will have a non-negligible loss due to neutron capture [13].

By continuously moving, or reprocessing, the ^{233}Pa from the core into a separate decay tank, these losses can be avoided. The Molten Salt Breeder Reactor continuously chemically extracts the protactinium from the fuel salt, allowing it to decay in a separate tank [37]. While it decays, another continuous chemical process strips uranium which is formed and brings the uranium back into the reactor, continuing the process [37].

1.4 Reprocessing Methods

For MSR reprocessing, the two different physical approaches are continuous and batchwise. These reprocessing schemes can be online, while the reactor is operating, or offline, when the reactor is shutdown. Of particular importance for MSR behavior is online reprocessing, which changes the fuel composition while the reactor is operating.

In order to model online reprocessing, there are two different computational methods which can be implemented: continuous and batchwise. These methods are named after the physical processes which they represent. However, it is fairly common for MSR analysis works which implement a phyiscally continuous reprocessing scheme to simulate this scheme using batchwise mathematical models.

This work seeks to investigate the continuous and batchwise computational reprocessing methods. This investigation involves effects of depletion step size, validity of batchwise methods approximating continuous methods, and different sub-methods within both batchwise and continuous methods which can be implemented.

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 use batchwise and continuous reprocessing models.

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 half lives 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.

2.1.1 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 online reprocessing is by using batchwise reprocessing and continuous reprocessing models. A purely batchwise modeling approach will not fit the reprocessing design used by a reactor, as online reprocessing is carried out continuously. The batchwise reprocessing models can accurately model a reprocessing scheme which has batchwise reprocessing, such as a scheme which has salt entirely replaced after some given amount of time.

Codes such as SaltProc and older versions of ChemTriton implement batchwise reprocessing by running the simulation, stopping, and then adding and removing materials [41, 8]. 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 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) [4, 24]. The benefits of using continuous reprocessing are that the model is more physically accurate and it allows for larger time steps to be used without increasing error. 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 [41]. Alternatively, the volume of the system can be adjusted so that constant density is maintained in order to keep cross section data consistent [36]. Another approach is to move excess fuel salt into a bleed-off tank [36].

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, while altering the input script by changing volumes or removing mass to a bleed-off tank will cause the time step to be run twice.

2.1.1.1 Batchwise Reprocessing

Batchwise reprocessing starts with some removal rate where some fraction, f, of a given element is removed from the fuel salt in the system over some time period, T_{cyc}, referred to as the cycle time. Over a given depletion simulation time step, Δt, a different fraction, γ, of the element's mass is removed, where γ corresponds to f according to Equation (2.1). The scaling term on f 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 fraction removed, γ, to be greater than 1, the amount removed remains at 100% so as to not have negative mass. This method is called steady batchwise reprocessing since it uses a steady removal amount during each depletion step.

An alternative approach is to establish a depletion time step Δt such that the efficiency based time value T_{cyc} is a multiple of Δt and only have the f batchwise removal occur at the steps when Δt is a multiple of T_{cyc}, thus removing the need for scaling. This method is called bulk batchwise reprocessing because it performs reprocessing all at once in a bulk manner.

$$\gamma = f \frac{\Delta t}{T_{cyc}} \tag{2.1}$$

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 et al in the ORNL report [37, 41]. 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 steady 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.

$$f = 100\%, T = 20s, \Delta t = 3d \rightarrow \gamma > 100\% \rightarrow 100\% \tag{2.2}$$

$$f = 100\%, T = 60d, \Delta t = 3d \rightarrow \gamma = 5\% \tag{2.3}$$

2.1.1.2 Continuous Reprocessing

The continuous reprocessing method involves directly modifying the Bateman equation, shown in Equation (2.4). One of the continuous reprocessing methods used in Serpent2 continuous reprocessing, an extension developed by Aufiero et al, is henceforth refered to as "decay" continuous reprocessing [4]. Equation (2.4) shows the Bateman equation with production and loss exclusively from fission, absorption, and decay; and Equation (2.5) shows the Bateman equation with the removal from reprocessing subtracted and addition from feed sources added.

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

$$\frac{dN_j}{dt}_{net} = \frac{dN_j}{dt}_{base} - \lambda_{r,j} N_j + \sum_{feed} \lambda_{f,j} N_j \tag{2.5}$$

The symbols given in the equations are defined as follows [27]:

N_j is the atomic density of isotope j.

$\gamma_{i\to j}$ 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.

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

$\sigma_{i\to j}$ is the microscopic transmution cross section of reaction $i \to j$.

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.

$\lambda_{f,j}$ is the feed constant for feed of isotope i.

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 transfered 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 feeds. Figure 2.1 shows how these reprocessing and feed constants fit into the reactor operation. Additionally, the figure shows that these terms relate to reprocessing and feed mass flow rates of $\dot{m}_{r,j}$ and $\dot{m}_{f,j}$, respectively.

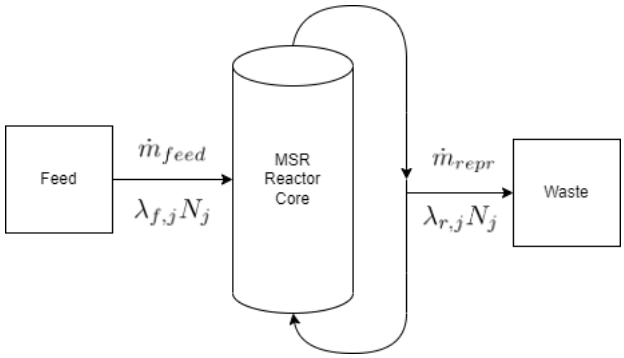

Figure 2.1: Simplified model of how Bateman equation relates to mass flow rates.

Serpent2 has three different settings for continuous reprocessing which alter the effect on the Bateman equations. The zeroth setting does not conserve mass and will not be discussed in this work.

The first setting, which adds a decay-like term to the Bateman equation, is shown in Equation (2.5). This approach of adding a decay-like term to the Bateman equation has been done with other codes [24, 38]. For this setting, three different approaches are implemented in order to compare their results, and are discussed later in this work. This setting is the most important for this work, and is the only setting implemented.

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 2.6, where the C term represents the constant value being removed by continuous reprocessing.

$$\frac{dN_j}{dt} = \sum_{i \neq j} \left[\left(\gamma_{i \to j} \sigma_{f,i} \Phi + \lambda_{i \to j} + \sigma_{i \to j} \Phi \right) N_i \right] - \left(\lambda_j + \sigma_j \Phi \right) N_j - C \tag{2.6}$$

This approach has a flaw which can be directly demonstrated through an example if the reprocessing constants are defined for an element and not for each isotope. Imagine an element exists with two isotopes α and β. 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	Both Fit	α Fit	β Fit	Zeroing Fit
α [g]	1000	900	949.5	900	999	990
β [g]	10	9	-40.5	-90	9	0
Net [g]	1010	909	909	810	1008	990

For the continuous reprocessing, however, the same constant removal rate, C, is defined for both of the isotopes and is the same value. To perform 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 β becomes negative, as can be seen in the "Both Fit" 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 α and β Fit 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 "Zeroing Fit" column, and it also does not work very well.

2.1.2 DNP Movement

The movement of DNPs in liquid fueled molten salt reactors impacts the operation of the reactor directly since the decay of the precursors occurs at a different location compared to a static reactor [47, 3]. There are several effects which this movement can have on the reactor. Since the delayed neutron precursors move, there is a probability of decay in non-core regions of the reactor, meaning there are fewer delayed neutrons in the core. This results in a reduced effective delayed neutron fraction, which reduces controllability of the reactor. Additionally, since delayed neutrons have a softer spectrum than prompt neutrons, there is a difference in spectrum in a reactor which has no flow against a reactor which has a circulating fuel. 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 2.2 [43].

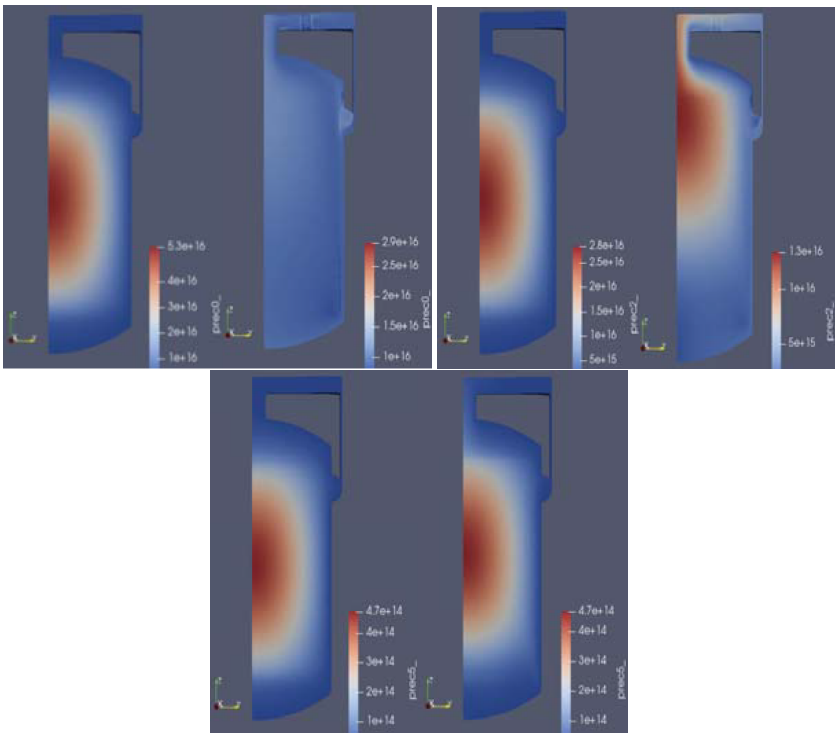

Figure 2.2: Plots of DNP concentrations in the molten salt reactor experiment [43]. The left side of each image shows concentration with no flow, while the right shows the concentration with a 1200 gallon per minute flow rate. Top Left) Longest lived DNP group. Top Right) Third longest lived DNP group. 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. 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.

2.2 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 [27]. 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 [28]. Other codes may use external software to compute depletion, such as REM used by MCNP and ORIGEN-S used by KENO-VI [4]. Another code, aside from Serpent2, that contains built-in depletion solvers is ERANOS [4].

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.

2.3 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 [37]. More specifically, the inner zone has a softer spectrum, a higher fission rate, and a 13% fuel-to-graphite ratio [34]. 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 [41]. The differences in the regions can also be seen in Figure 2.3, which allows for the difference in the fuel-to-graphite ratio to be visually noticeable.

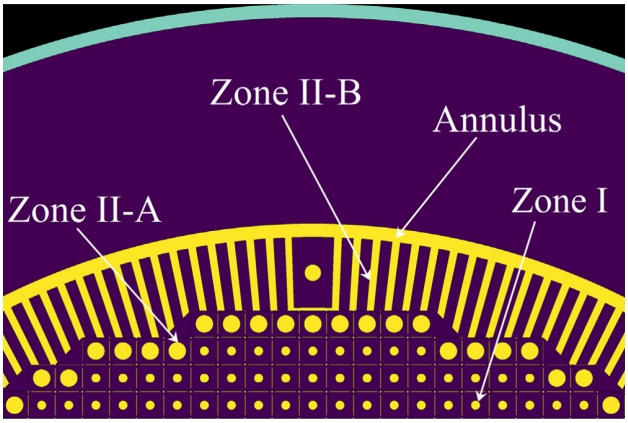

Figure 2.3: MSBR core axial slice showing the different regions from [41]. Zones II-A and II-B are where the spectrum is harder and there is increased breeding. Zone I is where there is more fission and a softer spectrum. The yellow is fuel salt, the purple is graphite, and the cyan is the reactor vessel.

Table 2.2 has general information on the MSBR, while Table 2.3 has specific information on the distribution of the fuel salt during operation. It can be seen in Table 2.2 that the salt inventory cycle time is 10 days, which means that for the entire salt inventory to be cycled takes 10 days. However, the loop cycle time is 11 seconds [37].

Table 2.2: MSBR General Data [37]

Component	Data
Thermal Capacity	2250 MW_{th}
Vessel Inner Diameter	6.77 m
Core Height	3.96 m
Vessel Pressure	0.52 MPa
Inner Salt Fraction	0.13
Outer Salt Fraction	0.37
Maximum Flow Velocity (Core)	2.6 $\frac{m}{s}$
Fuel Salt (Vessel)	30.4 m^3
Fuel Salt (Primary System)	48.7 m^3
Thorium Inventory	68,100 kg
Breeding Ratio	1.06
Processing Rate	1 gpm
Salt Inventory Cycle Time	10 days
Salt Components	$^7LiF-BeF_2-ThF_4-UF_4$
Salt Composition	71.7-16-12-0.3 mole %

Table 2.3 shows that the salt inventory in the core is approximately 19 m^3 due to each core zone. However, SaltProc uses the volume of the primary system for its calculation [41]. This is because the other parts of the MSBR outside of the core are not modeled, but the geometry implemented in Serpent2 uses the correct dimensions. This allows cross sections to remain as if the correct volume were implemented, but also allows the depletion results to be closer to the expected result if the fuel salt were fully utilized.

Table 2.3: MSBR Primary System Salt Inventory [37]

Region	Volume [m^3]
Fuel Salt (Primary System)	48.7
Reactor	-
Core Zone I	8.2
Core Zone II	10.8
Plenums, Inlets, Outlets	6.2
Annulus	3.8
Reflectors	1.4
Primary Heat Exchangers	-
Tubes	7.6
Inlets, Outlets	0.8
Pump Bowls	5.2
Piping (including drain line)	4.1
Off-gas bypass loop	0.3
Tank heels and miscellaneous	0.3

Table 2.4: MSBR Online Reprocessing Cycle Times [37]

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	3 days
Higher Nuclides	Np, Pu	16 years

Shown in Table 2.4 are the cycle times for different elements in the MSBR. According to Robertson et al, the protactinium and rare earth processing has the largest impact on neutronics and performance [37].

The fuel processing of the MSBR is defined to operate continuously with processing methods [37]. For protactinium removal, fluorination is first used to remove uranium before protactinium isolation. Next is countercurrent bismuth with lithium and thorium for stripping any remaining uranium and the protactinium. This is followed by hydrofluorination to separate the uranium and protactinium. This process can be seen in Figure 2.4. For a 10 day protactinium cycle time, a fuel salt flow rate of 0.88 gallons per minute (gpm) is used. The processed salt is processed for rare earth fission products before being fed back into the reactor.

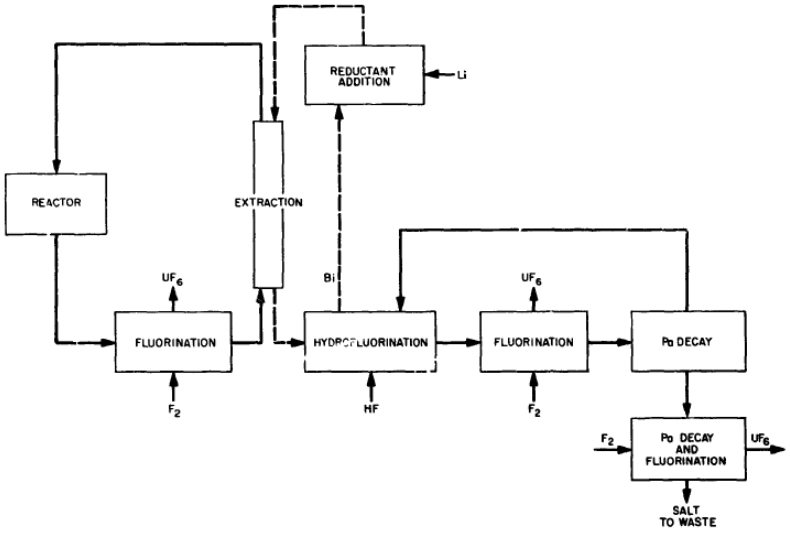

Figure 2.4: MSBR protactinium processing scheme from Robertson et al [37].

Rare earth fission product removal is performed using the metal-transfer process, which uses lithium chloride and bismuth containing a reductant. More specifically, bismuth containing a reductant of thorium and lithium is used to strip the rare earth fission products from the fuel salt. The rare earth fission products are then transported to the lithium chloride acceptor salt, though lithium bromide or a mix of both could be used. Figure 2.5 shows this process, and Figure 2.6 shows the protactinium and rare earth processing schemes together.

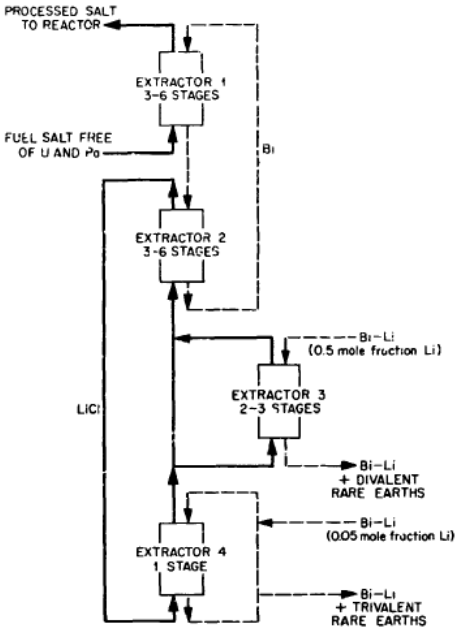

Figure 2.5: MSBR rare earth processing scheme from Robertson et al [37].

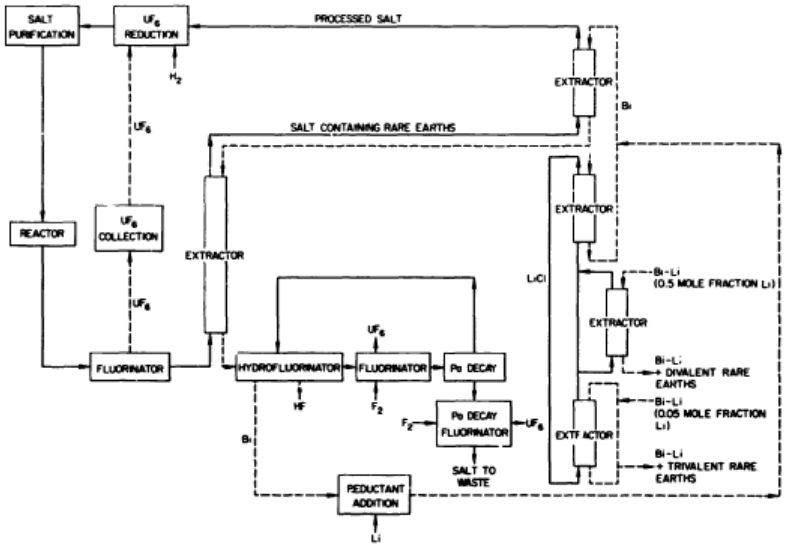

Figure 2.6: Combined MSBR rare earth and protactinium processing schemes from Robertson et al [37].

To strip gaseous fission products, a bubble generator introduces helium sparging gas in 15 to 20 mil bubbles. 10 percent of the flow is redirected to the bypass loop which contains a gas separator. This gas separator strips the fission products with approximately 100 percent efficiency. The off-gas system also includes an approximately 2 hour holdup during which noble metals deposit on the fuelsalt drain tank surface. Figure 2.7 shows this process, beginning with the number per time, or flux, of the noble gasses in the salt migrating to the graphite voids and helium sparging bubbles. This is followed by the volume holdup into more processing to handle the noble metal fission product cleanup.

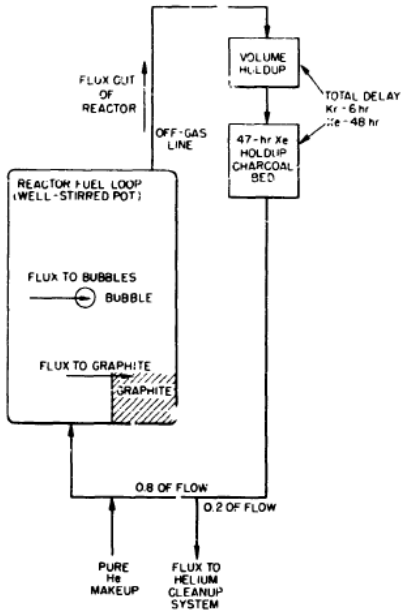

Figure 2.7: MSBR noble metal flux and off-gas system from Robertson et al [37].

Periodic discard of salt at a rate of 0.1 cubic feet per day occurs due to buildup of non-volatile fluorides during fluorination accumulating in the decay tank, such as from zirconium and corrosion product nickel.

2.4 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 aspects such as DNP movement [16, 43] or reactor dynamics [45, 12, 5, 14, 44]. This type of model typically is focused on short time scales of seconds.

The second type of model is one which depletes the fuel, focusing on time scales ranging from days to years. Because the online reprocessing of an MSR has such large effects on its fuel composition during a long depletion time, these 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 [49]. The movement of DNP can have a noticeable effect on reactor performance by affecting neutron energy spectra and distribution of delayed neutrons, though they do not have an impact when looking at longer time scales [7]. Most depletion simulations, however, focus primarily on depletion and do not consider the DNP movement during depletion, as can be seen in Table 2.5. One method used to handle spatial dependence of fuel salt without modeling the piping of the core is the scaled flux method [6].

For all of the continuous reprocessing models given in Table 2.5, all of them use the "fictitious decay constant"

approach, which is shown in Equation (2.5). For the batchwise models, most use the steady batchwise approach shown in Equation (2.1) and remove a fraction at each depletion step rather than bulk removal at certain steps.

The "Reprocessing" column of Table 2.5 shows the mathematical approach taken to the online reprocessing of the work, and does not necessarily represent the reactor reprocessing approach. However, since online reprocessing is considered in the table, offline batchwise reprocessing is not included, such as in the work by Zou et al [51]. The works shown in the table all use continuous reprocessing schemes, but some approximate the continuous physical process using batchwise methods.

The purpose of this table is to show that batchwise approximations to continuous reprocessing schemes have been implemented historically and recently. These methods have been implemented for a range of various reactors, various model types and dimensions, use various depletion step sizes, and cover various net depletion times. Additionally, tracking delayed neutron precursors is not common in depletion calculations, primarily due to their negligible impact on depletion results.

Table 2.5: Molten Salt Reactor Models with Online Reprocessing

Publication	Reprocessing	Model	Reactor	DNP	Year
Valdez et al [24]	Continuous	Unit Cell	N/A	No	2020
Zhuang et al [50]	Continuous	Full 3D	MSFR	No	2020
Xia et al [48]	Continuous	Full 3D	N/A	No	2019
Zhou et al [49]	Continuous	Full 3D	MSBR	Yes	2018
Aufiero et al [4]	Continuous	Full 3D	MSFR	No	2013
Rodriguez-Vieitez [38]	Continuous	Unit Cell	ADNA	No	2002
Rykhlevskii [41]	Batchwise	Full 3D	MSBR	No	2019
Betzler et al [8]	Batchwise	Unit Cell	MSBR	No	2017
Ridley and Chvala [36]	Batchwise	Full 3D	Toy	No	2017
Ahmad et al [2]	Batchwise	Unit Cell	DMSR	No	2015
Park et al [34]	Batchwise	Full 3D	MSBR	No	2015
Sheu et al [42]	Batchwise	Full 3D	MOSART	No	2013
Nuttin et al [33]	Batchwise	Full 3D	MSBR	No	2005
Fiorina et al [17]	Mixed	2D Slice	MSFR	No	2012
Nagy et al [32]	Mixed	Unit Cell	Toy	Yes	2008
Merle-Lucotte et al [30]	Mixed	Full 3D	TMSR	No	2007
Bauman et al [21]	Mixed	2D Slice	N/A	Yes	1971

For the various works using continuous reprocessing, many different codes have been used. One common aspect between all of the different works is implementation of long depletion step sizes on the order of months and long net depletion times on the order of years.

Valdez et al [24] shows the updated version of ChemTriton 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.

Zhuang et al [50] 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.

Xia et al [48] demonstrates the MOlten salt reactor specific DEpletion Code (MODEC). This code uses continuous reprocessing, and has been used to analyze the Molten Chloride salt Fast Reactor (MCFR) by Liaoyuan et al [29]. The MCFR analysis uses SCALE6.1 for transport and MODEC for depletion.

Zhou et al [49] covers 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 analyzes the molten salt fast reactor over 200 operational years and the molten salt breeder reactor for 20 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.

Aufiero et al [4] 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.

Rodriguez-Vieitez [38] 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 works which implement batchwise reprocessing implement depletion step sizes on the order of several days, and use net depletion times on the order of years. Because the net depletion time is similar to that of continuous reprocessing while implementing much smaller depletion step sizes, this results in a higher computational cost.

Rykhlevskii [41] 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.

Betzler et al [8] is on 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 [35].

Ridley and Chvala [36] 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.

Ahmad et al [2] 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 [18].

Park et al [34] 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 et al [37]. This model uses 3 day time steps and models the fuel composition over 20 years.

Sheu et al [42] 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.

Nuttin et al [33] 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.

Finally, a mixture of batchwise and continuous methods can be implemented if the reprocessing scheme calls for a physically batchwise process, or if the continuous method cannot handle the exact scheme as well as the batchwise method.

Fiorina et al [17] 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 at extend out to 30 year steps for a 200 year depletion. Additionally, the radiotoxicity analysis extends out to 1 million years.

Nagy et al [32] uses a toy model to go through different reprocessing schemes based on the MSBR and MOSART reprocessing designs [37, 23]. SCALE-5 [1] is used for neutron transport and ORIGEN-S is used for depletion. To account for DNPs, the authors make a few approximation which lead to reducing the calculated eigenvalues by 90 pcm. Depeletion steps are performed every 5 days, during which continuous reprocessing was implemented for either only gaseous fission product removal or more, depending on the reprocessing scheme chosen. Additionally, the batchwise reprocessing was performed by altering the binary output of ORIGEN, as is performed for the feed fuel and more, depending on the reprocessing scheme chosen.

Merle-Lucotte et al [30] covers depletion of the Thorium Molten Salt Reactor (TMSR) using MCNP4 [20] 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.

Bauman et al [21] 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.

Chapter 3

Methods

3.1 Batchwise Reprocessing

SaltProc is used to handle batchwise reprocessing in this work because it provides a batchwise reprocessing scheme which can be customized fairly easily. There are different versions of SaltProc which have slightly different methods in how the depletion data is stored as well as how reprocessing rates are calculated.

3.1.1 Bulk

SaltProc version 0.1 is the first full version of SaltProc which was publicly released, and provides all the functionality necessary for modeling the MSBR with batchwise reprocessing. This version uses 3 day steps and does not perform fractional reprocessing at each step, but instead performs reprocessing according to the cycle times, which is termed as the "Bulk" approach to batchwise reprocessing. This is the same approach used by Gehin and Powers for an analysis of the MSBR [19].

For example, a 30 day cycle time would be implemented as no reprocessing during the first 27 days, then full 100% removal of the target at the 30 day mark. This works well for reactor processes which are performed batchwise, such as adding uranium to the reactor which can be performed as a batch process. However, for approximating online reprocessing, this approach is not ideal, as it is less able to capture the frequency of the reprocessing.

For the refueling of the reactor, the thorium feed rate was set to maintain the thorium mass in the reactor. The uranium feed rate was set to be equivalent to the protactinium removal rate. This carries an implicit assumption that there is already a backlog of decayed protactinium which has transmuted into uranium which is ready to be pumped back into the core. Both the thorium feed and protactinium to uranium refueling processes occur every 3 days. These refueling rates can be seen in Figures 3.1 and 3.2, which are recreated from the data from Rykhlevskii [39].

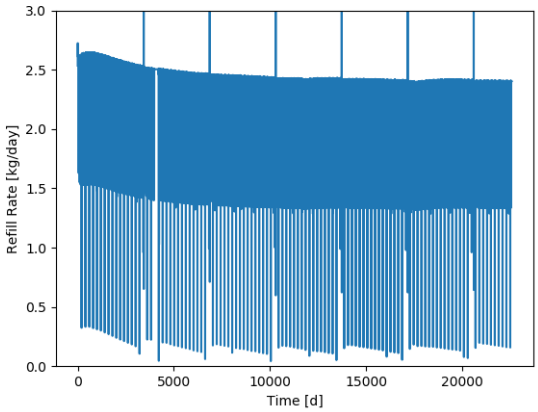

Figure 3.1: Thorium feed rate in the MSBR as a function of time while using bulk batchwise reprocessing [39].

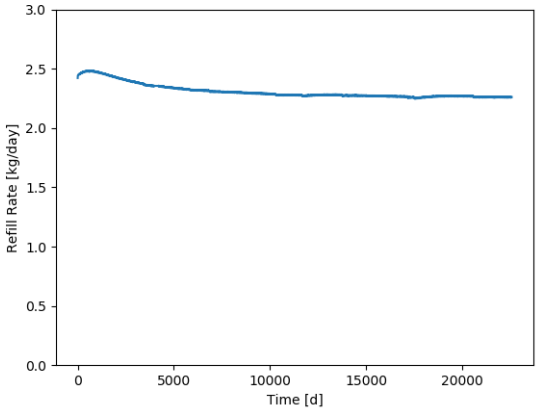

Figure 3.2: Uranium feed rate in the MSBR as a function of time while using bulk batchwise reprocessing [39].

The large jumps in the thorium feed rate are due to the bulk removal of absorbers, leading to spikes in the effective multiplication factor [39]. The uranium feed rate is smooth because it is based purely on the outflow of protactinium and does not fluctuate as rapidly as the thorium, which is the only external feed in the MSBR system. The average thorium-232 feed rate is 2.38 kg/day, while the average uranium-233 feed rate is 2.31 kg/day. These average values are used as the baseline for feed rates for the other batchwise method and for the continuous methods.

Another difference is that this version makes use of the h5py Python package to handle the hdf5 data files which contain depletion data. Within the hdf5 file, the atom density of each isotope before and after batchwise reprocessing is recorded as well as different reactor parameters, such as the neutron multiplication factor.

3.1.2 Steady

SaltProc version 0.3 includes an example MSBR which users can run as soon as SaltProc is installed, though some of the run parameters such as neutrons per generation and run time have to be altered to generate a more valid model. This model also uses 3 day steps, but instead of performing the batch removal only at the cycle time value, this model removes a fraction every 3 days, which is called the "Steady" approach for batchwise reprocessing.

For example, a 30 day cycle time would result in 10% removal every 3 days. This more accurately models online reprocessing, which is primarily what the MSBR employs in its reprocessing scheme. However, this steady version of SaltProc also makes some minor changes to the reprocessing scheme by adding in efficiency terms. These changes act as a multiplier on the reprocessing constant, where 100% efficiency makes no change and 0% makes the reprocessing constant become 0. These changes are as follows: xenon uses 91.2% removal over three days instead of 100%, krypton uses 91.5% removal over three days instead of 100%, protactinium uses 9.5% removal over three days instead of 100%, and the discard has a removal of 0.9% over three days instead of 0.09%.

The reactor refueling is performed slightly differently from v0.1 as well. This version assumes a constant ratio between the uranium refueling and the thorium refueling. The net refueling for each 3 day depletion step is set to maintain the mass of the system. Thus, the average values of 2.38 kg/day and 2.31 kg/day for thorium-232 and uranium-233, respectively, are used to determine the ratio of uranium to thorium which is implemented.

Figures 3.3 and 3.4 show the refueling rate of thorium-232 and uranium-233, respectively, using a depletion step size of 3 days over a net time of 90 days. The average feed rates of the thorium-232 and uranium-233 are 1.24 and 1.21 kg/day. The reasons for the difference are because the smaller efficiencies cause less mass loss, a shorter time frame is covered, and the protactinium removal is much less, resulting in a much smaller uranium-233 feed.

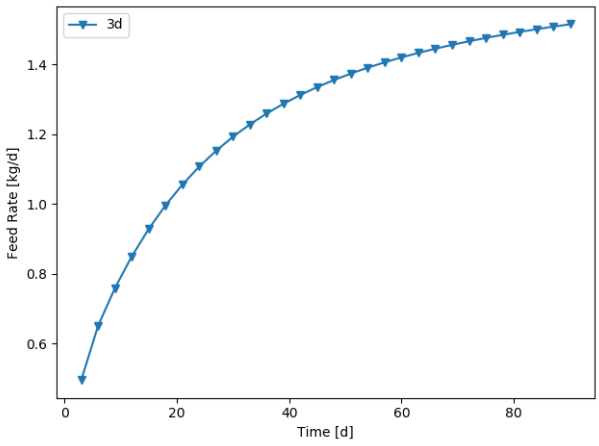

Figure 3.3: Thorium feed rate with a depletion time step of 3 days in the MSBR as a function of time while using steady batchwise reprocessing.

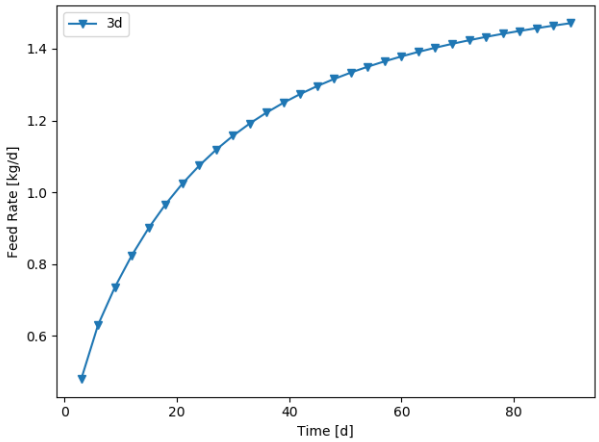

Figure 3.4: Uranium feed rate with a depletion time step of 3 days in the MSBR as a function of time while using steady batchwise reprocessing.

Additionally, this version of SaltProc uses the PyTables Python package for handling the hdf5 data files. The data files are structured similarly to v0.1 of SaltProc, but instead of atom density they use mass, which makes analysis of the files slightly more user friendly.

3.1.3 Batch Approaches Summary

Table 3.1: Batchwise Reprocessing Methods

Approach	Δt [s]	T_{cyc} [s]	Fractional Removal Rate [s^{-1}]	Step Removal
Bulk	1	20	-	0/1*
Bulk	10	20	-	0/1*
Bulk	40	20	-	1
Steady	1	20	0.05	0.05
Steady	10	20	0.05	0.5
Steady	40	20	0.025	1

* Bulk removal is 0 until the depletion step is equal to the cycle time, at which point it removes 100%.

Table 3.1 reflects the information shown in Figure 3.5. The table shows how the bulk method will only remove 100% at the cycle time, and otherwise removes nothing. Additionally, the bulk method will remove 100% as soon as

possible if the depletion step size, Δt, is larger than the cycle time, T_{cyc}.

The table also describes how the steady method fractional removal is constant up until the depletion step size becomes larger than the cycle time, at which point the removal at each step is 100%, causing the steady method to then be equivalent to the bulk method. However, for shorter times, the steady method allows for a consistent fractional removal rate by adjusting the removal at each step.

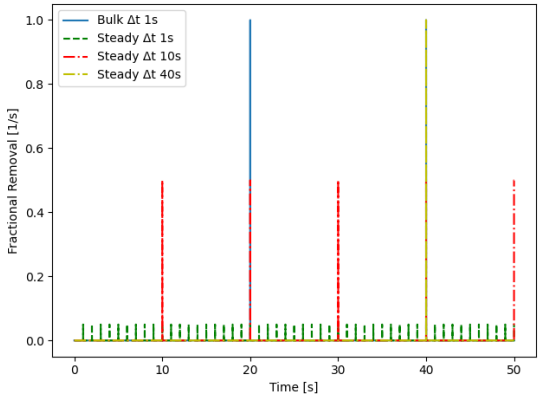

Figure 3.5: Plot showing how bulk and steady batchwise reprocessing removal works as a function of time for an example with a cycle time of 20 seconds.

Figure 3.5 shows an example fractional removal scheme for a fission product with a cycle time of 20 seconds. The Bulk method simply extracts 100% of the product every 20 seconds, no matter how small the depletion step size is. The steady method removes some fraction every depletion step, which allows a semi-continuous process as the depletion step size becomes smaller and smaller. This can be seen in how a 10 second depletion step results in 50% removal every 10 seconds, whereas a 1 second step allows for 5% removal per depletion step.

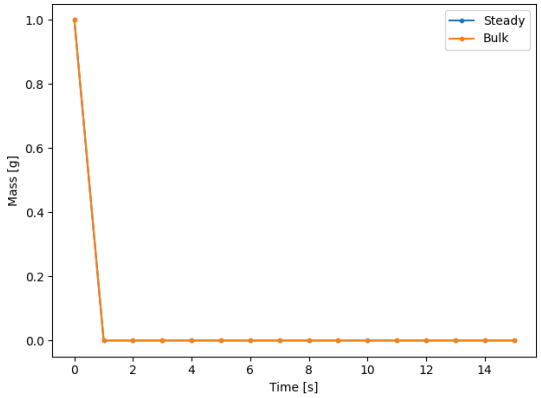

Figure 3.6: Plot showing how bulk and steady batchwise reprocessing are identical while the cycle time is shorter than or equal to the depletion step size.

A useful piece of information to note is that if the cycle time is shorter than or equal to the depletion step size,

then the steady and bulk batchwise methods will be identical to each other, as demonstrated in Figure 3.6 which shows a simple model case with 1 second depletion steps and 1 second cycle times for an imaginary isotope. For the MSBR simulation, this means that the noble metals, gasses, and protactinium reprocessing will remain the same for either batchwise method employed. However, the longer cycle time groups such as the rare earths and volatile fluorides have cycle times which are 10-20 times longer than the depletion step size.

Figure 3.7 shows a simulation of this for a situation where an isotope is generated at a rate of 1 gram per second, and is simulated with some combination of cycle time and depletion step size where the depletion step size is smaller than the cycle time.

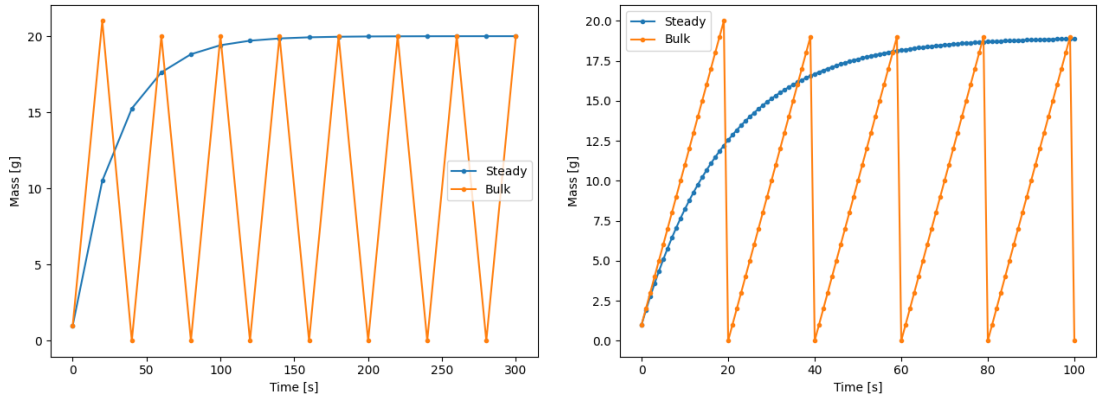

Figure 3.7: Plot showing how bulk and steady batchwise reprocessing are different while the cycle time is longer than the depletion step size for a 40 second cycle time and 20 second depletion step (top) and a 20 second cycle time with a 1 second depletion step (bottom).

One interesting aspect to note from this figure is that the steady method approaches a steady state value which is at the peak of the bulk method. This result can be confirmed by checking that the solutions for both the steady and bulk methods are valid. To check the solution, the 40 second cycle time and 20 second depletion step will be used.

For the bulk method, after 100% removal after the first cycle time, or 40 seconds, the mass then oscillates between 20 and 0 grams. This makes sense since 20 grams are generated over 20 seconds while every 40 seconds there is 100% removal.

For the steady method, the initial mass of 1 gram gains 20 grams over 20 seconds, yielding 21 grams. However, the steady removal then requires removal of 50%, previously discussed and shown in Figure 3.5. This 50% removal then drops the mass to 10.5 grams. 20 more grams are added to this value and 50% is removed again, continuing iteratively. This can then be re-written as an infinite sum in order to determine the steady state solution. The first three iterations are shown in Equations (3.1) and (3.2), where the value of 20 is the mass added during each depletion step and the value of 0.5 is the fractional removal performed each depletion step.

$$m_{ss} = (((N_0 + 20)(0.5) + 20)(0.5) + 20)(0.5) + ... \tag{3.1}$$

$$m_{ss} = 0.5^3 N_0 + 0.5^3(20) + 0.5^2(20) + 0.5(20) + ... \tag{3.2}$$

Continuing this form infinitely yields Equation (3.3), where the infinite sum yields 1, resulting in a net value of 20, the result of which is unaffected by the initial mass of the isotope and instead depends on the generation and reprocessing rates.

$$m_{ss} = 20 \sum_{n=1}^{\infty} \frac{1}{2}^n \tag{3.3}$$

In order to generate a generic form for this equation, a short derivation is required, which is provided in Appendix A. From this derivation, Equation (3.4) is generated.

$$m_{n+1} = \left[\left(m_n + \frac{C}{\lambda - \lambda_r} \right) e^{-(\lambda - \lambda_r)\Delta t} + \frac{C}{\lambda - \lambda_r} \right] (1 - \Delta t \lambda_r), \tag{3.4}$$

In this equation, C represents the gain terms in the Bateman equation, Δt represents the depletion step size, λ represents the loss terms in the Bateman equation, λ_r represents the reprocessing constant. This equation assumes a constant C, λ, and λ_r, which is a reasonable assumption at steady state when the changes over time are negligible. This equation is used to generated the steady state equation as shown in Equations (3.5) through (3.9).

$$m_{ss} = \frac{C}{\lambda - \lambda_r} \left(e^{(\lambda - \lambda_r)\Delta t} - 1 \right) \sum_{n=1}^{\infty} \left((1 - \lambda_r \Delta t)\, e^{-(\lambda - \lambda_r)\Delta t} \right)^n \tag{3.5}$$

$$\sum_{n=1}^{\infty} (x)^n = \frac{x}{1-x} \ni |x| < 1 \tag{3.6}$$

$$m_{ss} = \frac{C(e^{(\lambda - \lambda_r)\Delta t} - 1)}{\lambda - \lambda_r} \left(\frac{\eta}{1-\eta} \right) \ni |\eta| < 1, \tag{3.7}$$

$$\eta = (1 - \lambda_r \Delta t) e^{-(\lambda - \lambda_r)\Delta t} \tag{3.8}$$

The constraint listed in Equation (3.7) shows when the equation no longer holds. However, this constraint only shows when the infinite series becomes divergent. The more physical constraint is given in Equation (3.9). If the depletion step size, Δt, is equal to zero, then the simulation will never end, as no progress is made. Thus, the

concept of steady state does not apply. For any depletion step sizes which are greater than the cycle time, the steady state mass will always be zero, as negative masses are non-physical.

$$m_{ss} = \frac{C(e^{(\lambda-\lambda_r)\Delta t} - 1)}{\lambda - \lambda_r}\left(\frac{\eta}{1-\eta}\right) \ni 0 < \lambda_r \Delta t < 1 \tag{3.9}$$

Overall, this shows that the steady and bulk batchwise methods have some similarities, but for elements with longer cycle times, the bulk method will experience oscillations whereas the steady method will level off smoothly. Additionally, the average value of the steady method will be higher than the oscillating bulk method.

3.1.3.1 Comparison with Continuous Reprocessing

3.1.3.1.1 Error Estimation The difference between the bulk and steady batchwise reprocessing methods has been demonstrated, though how these methods compare to the continuous reprocessing methods have not been shown. For a simple example comparison, the steady batchwise method will be compared against the direct linear continuous method, which gives the relationship shown in Equation (3.10).

$$\lambda_r = \frac{1}{T_{cyc}} \tag{3.10}$$

This comparison will assume there is some isotope which is generated at a constant rate, C, and is only removed due to reprocessing.

The relevant continuous method equations are given in Equations (3.11) and (3.12), where the variables are the same as those discussed in Equation (3.9).

$$\frac{dm}{dt} = C - \lambda m \tag{3.11}$$

$$m(t) = \left(m_0 - \frac{C}{\lambda}\right)e^{-\lambda t} + \frac{C}{\lambda} \tag{3.12}$$

The previous example is continued here, using a constant rate of 1, an initial concentration of 1, a cycle time of 40 seconds, and a varying depletion step size. The results of this comparison can be seen in Figure 3.8.

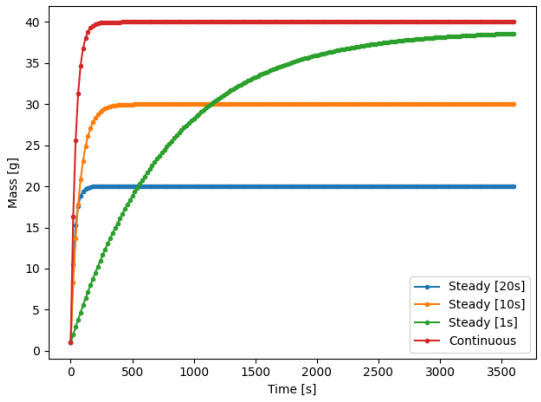

Figure 3.8: Plot comparing the values of continuous and batchwise reprocessing methods with a 40 second cycle time.

It can be seen in this figure that the direct linear continuous method will result in a steady state result which is significantly different from the steady batchwise method unless the depletion step size is infinitely small.

$$m_{ss} = \frac{C}{\lambda} \tag{3.13}$$

Equation (3.13) shows the steady state mass which will appear when using continuous reprocessing when there is a constant growth term. This can be set equal to the steady batch steady state term in order to solve for the depletion time step needed to match the result from continuous reprocessing at steady state. The result of this can be seen in Equations (3.14) and (3.15), which reinforces what can be seen in Figure 3.8.

$$\frac{C}{\lambda} = \frac{C(e^{(\lambda-\lambda_r)\Delta t}-1)}{\lambda-\lambda_r}\left(\frac{\eta}{1-\eta}\right) \tag{3.14}$$

$$\Delta t = 0 \tag{3.15}$$

Correspondingly, this means that the steady state error associated with implementing the steady batchwise method to approximate a continuous reprocessing scheme is directly proportional to the magnitude of the depletion step size used. Specifically, as shown in Equations (3.16) and (3.17), the error at steady state due to reprocessing directly scales based on the ratio of the depletion time step and the cycle time.

$$|E| = \frac{|m_{ss}^{batch} - m_{ss}^{continuous}|}{m_{ss}^{continuous}} \tag{3.16}$$

$$|E| = \frac{\lambda}{C}\left|\frac{C}{\lambda} - \frac{\lambda(e^{(\lambda-\lambda_r)\Delta t}-1)}{\lambda-\lambda_r}\left(\frac{(1-\lambda_r\Delta t)e^{-(\lambda-\lambda_r)\Delta t}}{1-(1-\lambda_r\Delta t)e^{-(\lambda-\lambda_r)\Delta t}}\right)\right| \tag{3.17}$$

This relative error can be seen in Figure 3.8, as the depletion time step to cycle time ratio is exactly 2, and the steady state value is twice is low as the continuous steady state value. Although these equations show only the simplest case of a constant growth, they demonstrate how the important using a small depletion step size is for batchwise methods to generate an accurate result.

Although this example does not seem to apply to the more complicated interactions which are present in the molten salt reactor model, it actually fits closely. As previosuly discussed, by assuming there is a constant amount of fissile mass in the system, in the case of the MSBR this is primarily uranium-233, m_{u233}; the fission yield does not change due to spectral changes, γ; and a constant neutron flux, ϕ; the equations can be expanded using the variables shown in Equation (3.18) and (3.19). The terms in these equations come from the Bateman equation.

$$C_i = \phi\sigma_{u233}\gamma N_{u233} + \sum_j N_j\left(\lambda_{j\to i} + \sigma_{j\to i}\phi\right) \tag{3.18}$$

$$\lambda_i = N_i\left(\lambda_{i,d} + \lambda_{i,r} + \sigma_i\phi\right) \tag{3.19}$$

In these equations, the reprocessing constant, λ_r, is rewritten from its previous form of λ because the decay constant, λ_d, is now present. Due to the simple assumptions made, which are valid over a short net depletion time or at steady state, these equations is of the same form as the simple equation presented in the example above. Thus, by replacing the C and λ terms, the previous example provides a reasonable estimate of many isotopes in the reactor simulation. This means that the error equation generated in Equation (3.17) remains reasonable for many fission product isotopes generated in the reactor at steady state.

3.1.3.1.2 Computational Cost One of the weaknesses of batchwise methods is the necessarily higher computational cost compared to continuous methods. The first reason for this is because batchwise methods which approximate a continuous process have reprocessing based error which scales with depletion step size, as shown in Equation (3.17). This means that smaller depletion step sizes are required for the same error as a continuous method.

The second reason is because of the "double-running" of simulations which is required when using a batchwise method in Serpent2. When running a batchwise depletion program, the first time step is run and the new material compositions are generated, providing 2 complete simulation results at times 0 and Δt. However, these generated materials do not have reprocessing factored in, so the external reprocessing script, such as SaltProc, then performs

the reprocessing on the materials. The updated materials are fed back in, and the simulation is run again. This results in two new simulation results at Δt and $2\Delta t$.

Because of the necessity of regenerating data based on the updated material properties after the external depletion is performed, each time step requires two simulations to be performed, essentially doubling the computational cost. With a continuous method, the materials are updated during the depletion step, which means the generated data already uses the data of reprocessing materials. This eliminates any need for double running, allowing continuous methods to be run for much less computational cost.

3.2 Continuous Reprocessing

The continuous reprocessing functionality of Serpent2 was undocumented for several years aside from on the Serpent2 forums, but more recently work has gone into providing documentation. However, there has not been comparisons of Serpent2 continuous reprocessing with Serpent2 batchwise reprocessing with no differences between the models aside from the mathematical reprocessing approach used.

There are three different options which can be used for the Serpent2 built-in continuous reprocessing. The different options are defined in Serpent2 as 0, 1, and 2; which are henceforth referenced as Constant, Decay, and Step reprocessing. These names are selected because they closely describe the behaviour of each of the methods.

3.2.1 Constant

The Constant reprocessing method is the only method of the three options which does not conserve mass. This option instead generates mass the same way as the Decay reprocessing method for the initial step, then continues using that mass rate from that point onwards.

This method is useful for handling a constant fresh fuel feed rate, or any other situation where a constant addition of mass is desired. Because this method does not remove mass, it is not useful for any sort of constant removal.

3.2.2 Decay

The Decay reprocessing method is the most useful of the three options, and that is due to its flexibility in its usefulness. This method adds a decay term to whatever material it is attached to, and that decay term becomes a feed term to whatever material is set to receive the flow. For fission product removal, this is likely some material which is not modeled in the geometry of the problem but only exists to receive the fission product waste. However, this term can instead be leveraged to turn it into a feed rate.

This can be performed by attaching the decay term to a material which contains a volume of the desired feed, and then having that material feed into the core. The feed then "decays" from the feed tank into the core, essentially operating as a feed. Using this same method but altering the reprocessing constant or volume of the feed tank allows for an essentially constant feed rate, where the removed mass is negligible compared to the net mass.

Additionally, for this method, there are multiple approaches which can be used to calculate the reprocessing constants that should be implemented. These approaches are named "Cycle Time Decay", "Cycle Rate", and "SaltProc Cycle Rate" accordingly. The Cycle Time Decay approach treats the cycle time as if it is twice a half-life, and the cycle time is an exponential process. The Cycle Rate approach treats the cycle time as the time where 100% of removal occurs, and then linearly extrapolates by assuming an equal percentage removal occurs up to that point.

The reason these approaches are implemented is because molten salt reactor reprocessing schemes provide reprocessing data in terms of cycle times. The cycle time is the amount of time it takes for a given element to be removed from the system. This cannot be directly solved using the Bateman equation, as shown in Equations (3.20) through (3.22) and (3.23) through (3.26).

$$\frac{dN}{dt} = -\lambda_r N \tag{3.20}$$

$$N(\Delta t) = 0 = N_0 e^{-\lambda_r \Delta t} \tag{3.21}$$

$$-ln(0) = \lambda_r \Delta t \tag{3.22}$$

Equations (3.20) through (3.22) show how the solution results in a reprocessing constant of infinity when only continuous removal is considered.

$$\frac{dN}{dt} = C - \lambda_r N \tag{3.23}$$

$$N(\Delta t) = 0 = N_0 e^{-\lambda_r \Delta t} + \frac{C}{\lambda_r}\left(1 - e^{-\lambda_r \Delta t}\right) \tag{3.24}$$

$$0 = N_0 e^{-\lambda_r \Delta t} + \frac{C}{\lambda_r} - \frac{C}{\lambda_r} e^{-\lambda_r \Delta t} \tag{3.25}$$

$$0 = \lambda_r N_0 e^{-\lambda_r \Delta t} + C - C e^{-\lambda_r \Delta t} \tag{3.26}$$

Equations (3.23) through (3.26) show that the only real solution to a steady accumulation and continuous removal is with a reprocessing constant of 0.

3.2.2.1 Cycle Time Decay

The Cycle Time Decay approach is a simple method which makes use of the mathematical form of the Decay reprocessing method. A similar method of treating removal periods for reprocessing can be seen in Brovchenko et al [11]. Because it adds a "decay-like" term to the Bateman equation, this approach takes the cycle time for the target, cuts it in half, and then treats that value as the reprocessing half-life for that target. This process is shown in Equations (3.27) and (3.28).

$$\Delta t_{1/2} = \frac{T_{cyc}}{2} \tag{3.27}$$

$$\lambda_r = \frac{ln(2)}{\Delta t_{1/2}} \tag{3.28}$$

An example of implementing this approach for a 30 second cycle time would then have a 15 second reprocessing half-life. This is then converted to a reprocessing constant in the same way that a decay half-life is converted to a decay constant, which can be seen in Equation (3.29), resulting in a value of 0.0462 s^{-1}.

$$\lambda_r = \frac{ln(2)}{15} = 0.0462 s^{-1} \tag{3.29}$$

3.2.2.2 SaltProc Cycle Time Decay

This approach is the same as Cycle Time Decay, but alters for any target which has a cycle time less than the batchwise reprocessing step incorporated by SaltProc. For example, a 3 day cycle time for some target would be treated the same as the standard Cycle Time Decay. However, a cycle time shorter than the 3 day batchwise reprocessing step used by SaltProc for the MSBR has its half-life extended to the SaltProc minimum value of 1.5 days. For example, a 30 second cycle time would instead be treated as a 3 day cycle time, which would result in a half-life of 1.5 days. Plugging in, this would result in a reprocessing constant of 0.462 $days^{-1}$, or 5.348E-6 s^{-1}. This is the reprocessing constant for any cycle time which is less than or equal to three days. Additionally, the decay constant is multiplied by the fractional efficiency of the removal used by SaltProc. In most cases the value is 1, but it is roughly 0.91 for xenon and krypton.

3.2.2.3 Cycle Rate

The Cycle Rate approach uses a linear approximation such that the inverse of the cycle time is the rate at which material is removed. This is represented by, for example, 10% removal per second would neglect the efficiency decrease over time and give 100% removal after 10 seconds. This is calculated by investigating a unit time progression, i.e. 1 second or 1 day. Over this time period, the removal of atoms should be the fractional rate value, which means the final atom count at a time of 1 should be 1 - f, where f is the fractional removal rate. This can be seen in Equations (3.30) and (3.36).

$$f = \frac{1}{T_{cyc}} \tag{3.30}$$

$$\frac{dN}{dt} = -\lambda_r N \tag{3.31}$$

$$N(t) = N_0 e^{-\lambda_r t} \tag{3.32}$$

$$N(t=1) = (1-f)N_0 \tag{3.33}$$

$$(1-f)N_0 = N_0 e^{-\lambda_r (1)} \tag{3.34}$$

$$-ln(1-f) = \lambda_r \tag{3.35}$$

$$\lambda_r = ln\left(\frac{1}{1-f}\right) \tag{3.36}$$

An example cycle time of 30 seconds would be modeled by taking the inverse, which gives 0.033 s^{-1}. This value is then converted to a reprocessing constant by plugging it into the solved differential equation form shown in Equations (3.37) and (3.38), giving a value of 0.0339 s^{-1}.

$$f = \frac{1}{30} = 3.33E\text{-}2s^{-1} \tag{3.37}$$

$$\lambda_r = ln\left(\frac{1}{1-3.33E\text{-}2}\right) = 0.0339s^{-1} \tag{3.38}$$

3.2.2.4 SaltProc Cycle Rate

The SaltProc Cycle Rate approach is the same as the Cycle Rate approach, but takes into account the limiting nature of the 3 day batchwise reprocessing step used by SaltProc. For example, a 6 day cycle time target would be modeled the same using the SaltProc Cycle Rate approach as the standard Cycle Rate approach. However, anything shorter than 3 days would be modeled differently, since that is the batchwise reprocessing step inocorporated by SaltProc for modeling the MSBR. For example, a 30 second cycle time would be analyzed instead as a 3 day cycle time, since SaltProc can only remove 100% of material after a minimum of 3 days. This means the inverse would be 0.333 $days^{-1}$, or 3.858E-4 s^{-1}. Converting to a reprocessing constant gives a value of 3.858E-6 s^{-1}. This is the reprocessing constant for any cycle time which is less than or equal to three days. Additionally, the decay constant is multiplied by the fractional efficiency of the removal used by SaltProc. In most cases the value is 1, but it is roughly 0.91 for xenon and krypton.

3.2.2.5 Direct Linear Approach

Another approach is to directly apply the Cycle Rate removal rate, which is the inverse of the cycle time, as the reprocessing constant [22]. This is referred to as the Direct Linear approach. This approach is similar to the Cycle Rate approach, though the derivation for it is slightly different. This can be seen in Equations (3.39) through (3.50), where Equations (3.48) and (3.49) uses L'Hôpital's rule.

$$f = \frac{1}{T_{cyc}} \tag{3.39}$$

$$\frac{dN}{dt} = -\lambda_r N \tag{3.40}$$

$$N(t) = N_0 e^{-\lambda_r t} \tag{3.41}$$

$$N_{cur} = N_{prev} e^{-\lambda_r \Delta t} \tag{3.42}$$

$$N_{cur} = (1 - \Delta t f) N_{prev} \tag{3.43}$$

$$(1 - \Delta t f) N_{prev} = N_{prev} e^{-\lambda_r \Delta t} \tag{3.44}$$

$$-ln\big((1-\Delta tf)\big) = \lambda_r \Delta t \tag{3.45}$$

$$\lambda_r = \frac{-ln\big((1-\Delta tf)\big)}{\Delta t} \tag{3.46}$$

$$\lambda_r = lim_{\Delta t \to 0} \frac{-ln\big((1-\Delta tf)\big)}{\Delta t} \tag{3.47}$$

$$\lambda_r = \frac{0}{0} \tag{3.48}$$

$$\lambda_r = lim_{\Delta t \to 0} \frac{f}{1-f\Delta t} \tag{3.49}$$

$$\lambda_r = f = \frac{1}{T_{cyc}} \tag{3.50}$$

These equations follow a similar path to the Cycle Rate approach, but instead of using the linear approximation value after a unit time step, this approach generates the reprocessing constant while implementing the linear approximation as the time step goes to zero.

The differences between Cycle Rate and Direct Linear approaches can be seen in Table 3.2, where for longer cycle times, the difference is negligible, but for shorter cycle times, the difference becomes larger. However, since extremely short cycle times are not realistically practical, the two approaches are approximately equivalent. Overall, the Direct Linear approach is more numerically stable, however, since it does not have asymptotic until reaching a cycle time of 0 seconds, whereas the Cycle Rate method has asymptotic behaviour at 1 second cycle time. This can be seen in Figure 3.9. The asymptotic behaviour for a 0 second cycle time is not an issue because there are no negative cycle times, meaning the value is only approached from the positive side and behaves as physically expected.

Table 3.2: Subset of Decay Reprocessing Approaches

Cycle Time	Removal Rate $[s^{-1}]$	CR λ_r $[s^{-1}]$	DL λ_r $[s^{-1}]$	$\Delta\lambda_r$
3 d	3.86E-6	3.86E-6	3.86E-6	7.44E-12
20 s	0.05	5.13E-2	5.00E-2	1.29E-3
5 s	0.2	2.23E-1	2.00E-1	2.31E-2
2 s	0.5	6.93E-1	5.00E-1	1.93E-1
1 s	1	-	1	-

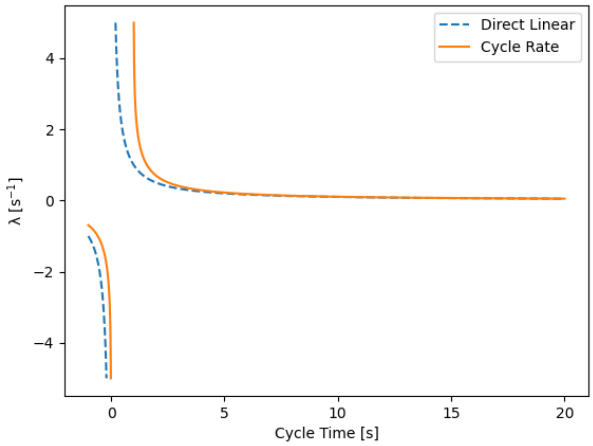

Figure 3.9: A comparison of the Direct Linear and Cycle Rate reprocessing constants for different cycle times.

Because the shortest cycle time for the MSBR is 20 seconds, this shows that the Cycle Rate and Directly Linear approaches are roughly equivalent for this reprocessing scheme.

3.2.2.6 SaltProc Direct Linear Approach

The SaltProc Direct Linear approach follows the same trend as the other SaltProc-based modified approaches. This includes changing the smallest time scale to 3 days and altering the fractional removal values of xenon and krypton.

3.2.2.7 Decay Approaches Summary

Figure 3.10 shows how the reprocessing constants for the different approaches vary as a function of cycle time.

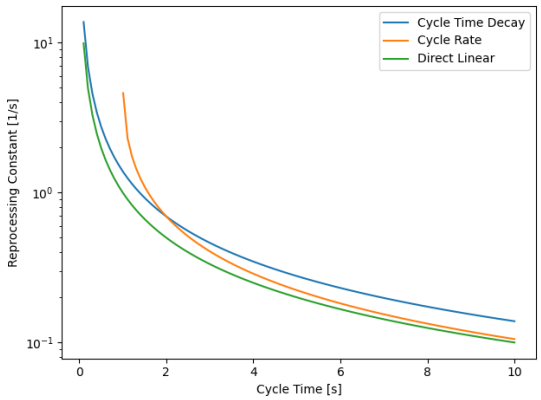

Figure 3.10: Plot of how reprocessing constants for different approaches vary with cycle time.

Figure 3.10 shows the three different continuous decay based methods for reprocessing as a function of various cycle times. From the figure, it can be seen that the Cycle Rate method does not have results for cycle times less than or equal to 1 second, which is a significant weakness of the method since physical processes could exist with a cycle time in that range. The other two methods provide results down to a 0 second cycle time, which allows full coverage of possible cycle time values.

The Direct Linear method behaves similarly to the Cycle Time Decay method during small cycle times and the Cycle Rate method for longer cycle times. This is due to the general form of the equations for each, where at small cycle times the inverse cycle time relationship with the Cycle Time Decay method dominates, while at larger cycle times the Cycle Rate log of the inverse term matches more closely.

Table 3.3 shows all the different decay based continuous reprocessing approaches for various cycle times. From this table, it can be seen that the direct linear and cycle rate methods are very close, while the cycle time decay method is off by a small amount at all cycle times. In terms of the SaltProc variants, the main difference is the fact that the cycle times below three days have been assigned the reprocessing constant of the three day cycle time.

Table 3.3: Full Set of Decay Reprocessing Approaches for Various Cycle Times

Method	$T_{cyc} = 1s$	$T_{cyc} = 20s$	$T_{cyc} = 3d$	$T_{cyc} = 50d$
DL	1	5.00E-2	3.86E-6	2.31E-7
CR	-	5.13E-2	3.86E-6	2.31E-7
CTD	1.39	6.93E-2	5.35E-6	3.21E-7
SPDL	3.86E-6	3.86E-6	3.86E-6	2.31E-7
SPCR	3.86E-6	3.86E-6	3.86E-6	2.31E-7
SPCTD	5.35E-6	5.35E-6	5.35E-6	3.21E-7

The values shown in Table 3.3 for the SaltProc-type continuous methods are based on the cycle times given from the Robertson report, which matches what is used in the bulk SaltProc, version 0.1, results. The steady SaltProc results, verson 0.3, do not use these exact values [37, 39]. The differences in steady SaltProc are as follows: xenon has 91.2% removal over three days instead of 100%, krypton has 91.5% removal over three days instead of 100%, protactinium has 9.5% removal over three days instead of 100%, and the discard has a removal of 0.9% over three days instead of 0.09%.

3.2.3 Step

The Step reprocessing method implemented in Serpent2 is mathematically very similar to the Decay model, but instead of updating continuously in time, it instead is updated during new depletion steps. In this manner, it is a sort of mix between the Constant method and the Decay method. This is because over a single depletion step, it is a constant added or subtracted from the Bateman equation, while over many depletion steps, it follows the same exponential decay form of the Decay method.

This particular method is not useful for extracting fission products, and is not needed for constant feed rates since that can be modeled using the Decay method. This method could be useful for inducing a step drop in feed rate, though this would require running only a single depletion step until the drop occurred. Additionally, this drop could be simulated by running the Decay method and reducing the reprocessing constant. This would allow for flexibility in the distribution of depletion steps as well without having to worry about changing the behaviour of the reprocessing functionality.

Another potential issue with the Step method is that the depletion step can last long enough that the constant

mass removal causes the mass to go negative. However, Serpent2 will cease running if this occurs. The Decay method does not mathematically allow for negative mass, the Constant method does not conserve mass, and the Step method stops running once there is negative mass.

The main potential use of the Step reprocessing may be for movement of material at a set rate. This could be implemented by writing a script to check the current number of atoms of the target and adjusting the reprocessing rate accordingly. However, for the MSBR, this form of reprocessing is not necessary, and is thus not implemented.

3.3 Mass Balancing

One of the useful features of batchwise reprocessing is that the net mass of the core can be balanced by adjusting the feed rates to provide the same amount into the core that is removed through reprocessing. Alternative methods also exist, such as removing excess mass or increasing the volume [36]. SaltProc version 0.1 does not account for mass balancing but maintains a constant thorium mass, whereas SaltProc version 0.3 has the feed rates set to maintain mass. Mass balancing is particularly important in Serpent2 due to the way masses in Serpent2 are handled.

In Serpent2, an increase in mass does not affect volume, but instead increases the density of that isotope in the material accordingly. This affects macroscopic cross section calculations, and can lead to variation in results if not accounted for, since in reality volumetric expansion could be assumed. Though there are several methods to handle mass balancing with a batch method, it not currently continuously possible in Serpent2.

In order to balance the mass in Serpent2 continuously, one approach could be to iteratively perform depletion calculations while updating feed rates until a balanced mass is found while also minimizing some other parameter, such as net mass of material added. However, the current reprocessing options available in Serpent2 only allows for pseudo-constant and decaying feed rates over the depletion step. With those two feed types available, it is not possible to have a constant mass balance.

It is possible to have the masses at the end of each depletion step remain constant, but this does handle mass fluctuations during depletion. It does, however, solve the issue of the density variation causing a difference in cross sections.

Overall, if the net mass difference is sufficiently small, it does not have to be considered since the results would not be significantly impacted. To check this for the MSBR, a simple back-of-the-envelope calculation can be used. To determine the maximum possible increase in density, the mass loss due to fission and reprocessing is neglected, and only mass addition from the thorium feed rate is considered. The uranium feed is not added because it is equivalent to the protactinium removal, so it would have a negligible impact on net mass.

$$\Delta m = \dot{m}_{feed} \Delta t \approx (2.5)(6000) = 15,000 kg \tag{3.51}$$

It can be seen in Equation (3.51) that the net mass gain for an average feed rate of 2.5 kilograms per day over 6000 days is 15,000 kg [40, 8]. This does seem to be a very large value, but the importance of the mass in the depletion calculation is primarily in how it affects cross sections, which means that the impact on the overall thorium density is important. The thorium density is 1.46 grams per cubic centimeter, and the net volume is approximately 48.71 cubic meters. The density with the added mass is calculated using Equation (3.52).

$$\rho_f = \frac{m_0 + \Delta m}{V} = \frac{(4.871E7)(1.45919E\text{-}3) + (15,000)}{4.871E7} \tag{3.52}$$

The final density comes out to be 1.767 grams per cubic centimeter, which is a percent difference of 21%. Although this is an upper bound on the mass difference, a 21% difference in the expected cross section would result in significant error in the results. Therefore, since it is possible for the mass balancing to have an impact, it should be investigated to ensure the mass balancing is not impacting the results in any unexpected manner.

3.4 Effects of Delayed Neutrons on Depletion

Delayed neutrons have a softer energy spectrum and drift along with the movement of the fuel salt in a fluid fueled molten salt reactor. This means that the delayed neutron precursors, of which some fraction leave the core and some move to less neutron important regions, have the potential to alter the depletion results of the MSBR model.

However, it has been shown by Zhou et al that for the MSBR, the delayed neutron precursor drift has a negligible impact on depletion results [49]. Although Zhou et al has shown this, it is still worth considering the maximum possible effect delayed neutron precursors could have on depletion results. In order to determine this, the Serpent2 functionality of disabling delayed neutrons is employed [28]. Using this method, two different models are generated of the MSBR. The first is one in which the delayed neutron precursors are evenly distributed within the core, which is the model implemented by SaltProc and this work [41]. The second model is one in which delayed neutrons no longer exist. This is beyond the greatest effects the delayed neutron precursor drift could have on depletion, so the absolute maximum effect of delayed neutron precursors on depletion results can be determined.

Because the delayed neutron precursors are fission products or come from the decay of fission products, depletion must be performed. For reprocessing, the average feed rate from SaltProc is used with continuous Direct Linear reprocessing, while Direct Linear reprocessing is used for fission product removal.

The impact on k_{eff} was investigated, and it was determined that the delayed neutrons from fission products

add roughly 20 pcm to k_{eff} after 6,000 days of operation, which is roughly steady state operation. The masses of various isotopes, such as uranium-235 and thorium-232, were also compared, though no statistically significant difference was found. Therefore, the net impact of delayed neutrons from fission product precursors is negligible on depletion results, which agrees with the results from Zhou et al.

3.5 Serpent MSBR Model

The results generated in this work come from the MSBR, particularly the model developed by Rykhlevskii [39]. The geometry of the model has remained unchanged, while the reprocessing has been overhauled using the continuous reprocessing functionality in Serpent2 which was previously undocumented. Previously, SaltProc's batchwise reprocessing was used, but the continuous reprocessing in Serpent2 is incorporated in this work.

3.5.1 Reprocessing Structure

There are two different reprocessing schemes developed in this work. Both schemes follow the MSBR reprocessing scheme, and vary how refueling functions. The first method matches the method of Rykhlevskii, where the uranium-233 feed is treated as equivalent to the protactinium-233 removal rate. In this case, the average uranium-233 feed rate from Rykhlevskii is used over the entire simulation time [39]. A simplified overview of this scheme can be seen in Figure 3.11.

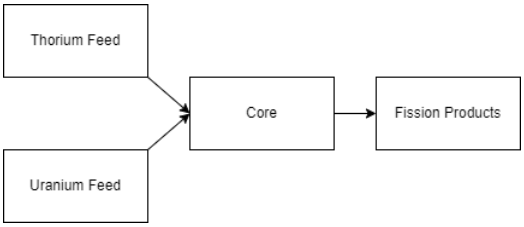

Figure 3.11: Simplified reprocessing scheme based on mimicking SaltProc.

The second reprocessing scheme is physically based, and instead creates a protactinium decay tank. From this tank, any generated uranium is continuously removed and sent back to the core, which is the intended design scheme for the MSBR [37]. A simplified version of the phyiscally realistic uranium feed can be seen in Figure 3.12.

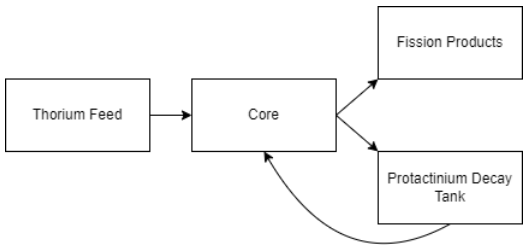

Figure 3.12: Simplified reprocessing scheme based on the phyiscal MSBR processes.

Overall, it can be anticipated that the two methods will be approximately equal at steady-state, since at that point the assumption by Rykhlevskii that the uranium input is the same as the protactinium output should be valid. However, there is expected to be a fairly large difference for the several months of operation, as the half life of protactinium-233 is on the order of one month, and the initial decrease in fissile material will impact reactor performance. This means that the uranium feed will take some time before it is providing a steady source of fissile material to the core.

Chapter 4

Results

4.1 Depletion Step Mesh Refinement

As previously shown, there are many different methods available for modeling the online reprocessing of molten salt reactors. Previously, Rykhlevskii has generated results for the molten salt breeder reactor using the bulk batchwise method. This method is primarily useful due to the straightforward nature of its implementation. However, it is less accurate and more costly than other methods. The other batchwise method, steady batch, has approximately the same computational cost, but provides more accuracy by making better use of each depletion step available.

The continuous methods are all the same in terms of computational cost. The main difference is in the interpretation used when provided with the cycle time data. Because the cycle times given are the amount of time to remove 100% of a target, and the continuous methods all use exponential terms, different strategies are used. The most straightforward method is Direct Linear, while the methods of Cycle Rate and Cycle Time Decay are based on assumptions.

Though the different methods have been discussed in theory, it is more enlightening to view a practical application of each in order to compare them. In order to compare the different reprocessing methods more clearly, each is first taken through a mesh refinement study in order to determine the effects of depletion step. Intuitively, it is expected that the batchwise methods will perform exceedingly poorly with increasing depletion step size because of two main reasons. Firstly, less frequent depletion steps means the cross section and flux data is not updated as the fuel is depleted, leading to a less accurate results. However, this reason is shared by continuous reprocessing as well. The second reason is due to the less frequent application of reprocessing on a process which is intended to be occurring continuously.

This second reason can be more intuitively understood with an example. Consider a simple thermal-spectrum system where uranium is fissioning and reprocessing is taking place. If xenon is physically continuously reprocessed, then the total absorption by xenon is reduced. However, using a longer batchwise step size results in xenon forming and capturing neutrons. Only after the depletion step size time elapses is batchwise reprocessing performed and the xenon is removed. By this point, many neutrons have already been captured, affecting reactor performance.

4.1.1 Batchwise Reprocessing

4.1.1.1 Bulk Batchwise

A depletion step mesh refinement study using the bulk batchwise reprocessing version of SaltProc was performed by Rykhlevskii comparing 3, 6, 12, and 24 day depletion steps over 25 years for the Transatomic Power Molten Salt Reactor [40]. Because the bulk batchwise method has already been analyzed and discussed, results for that method will not be separately generated here. The bulk batchwise model of the MSBR by Rykhlevskii will be used as an item of comparison to the continuous and steady batchwise methods implemented in this work [39].

4.1.1.2 Steady Batchwise

Based on this previous work, this work instead checks 1, 1.5, 3, and 6 day depletion steps for the steady batchwise reprocessing method which is now used in SaltProc. This is to check if a smaller step size than 3 days should be implemented, or if the accuracy gained is not worth the increased computational cost. Previously, Rykhlevskii used a 3 day depletion step in the bulk batchwise analysis of the MSBR over 20 years [39]. Considering the computational cost increases associated with using a smaller depletion step size, this depletion step size selection was well made. Therefore, for the longer net depletion time results, the 3 day depletion step will also be used. However, it is still beneficial to check how the 3 day depletion step results would compare to longer and shorter depletion times.

Although the cycle time was previously implemented as 100% removal, the krypton and xenon removal is instead around 91%. This is because of the different liquid phase mass transfer coefficient, and was discussed by Rykhlevskii previously [40]. For the other elements, a fixed value is assumed. One difference this has is that instead of having a mass of zero in the xenon-135 and xenon-136 figures, there will be plot showing how the xenon concentrations vary over time. Additionally, the protactinium removal is 9.5% instead of 95% every three days, which results in a shift in the average feed rates which were determined in the bulk batchwise SaltProc solution.

4.1.1.2.1 Effects of Reprocessing Figures 4.1 through 4.5 show how the different depletion step sizes impact the effective multiplication factor and concentration of different isotopes in the system after reprocessing has occurred.

It can be seen by the evolution of the effective multiplication factor that different steady state values are being approached. The difference after 24 days between the 1 day depletion step and 6 day depletion step is roughly 656 pcm, 490 pcm for the 1 and 3 day steps, and 260 pcm for the 1 and 1.5 day steps. Based on the trend shown, it is expected for these values to only diverge further apart over time.

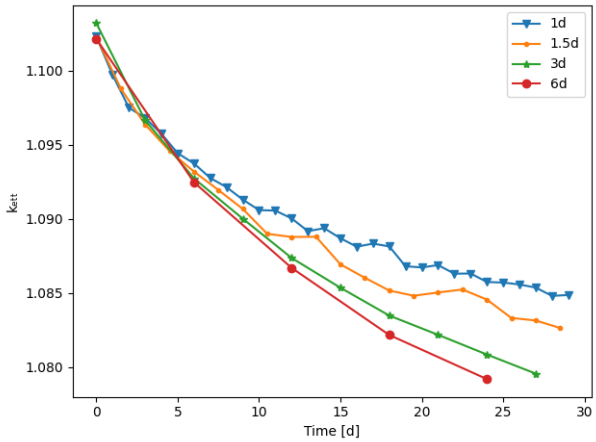

Figure 4.1: k_{eff} over time using various depletion step sizes with steady batchwise reprocessing.

A similar trend of approaching a different steady state value can be seen from the thorium-232 and uranium-233 masses over time. After 30 days, the thorium-232 maximum difference is roughly 22 kilograms between the 1 day and 6 day depletion steps. The uranium-233 difference is roughly 13 kg, which is a slightly smaller difference than the thorium-232. The reason the longer time step has a reduced thorium-232 and uranium-233 mass is because they are both treated as feed rates which maintain a constant net mass for the system. Because the reprocessing is less frequent, there are more parasitic absorbers built up in the core, reducing the total amount of material which is burned. Therefore, there is a smaller feed rate provided to the system.

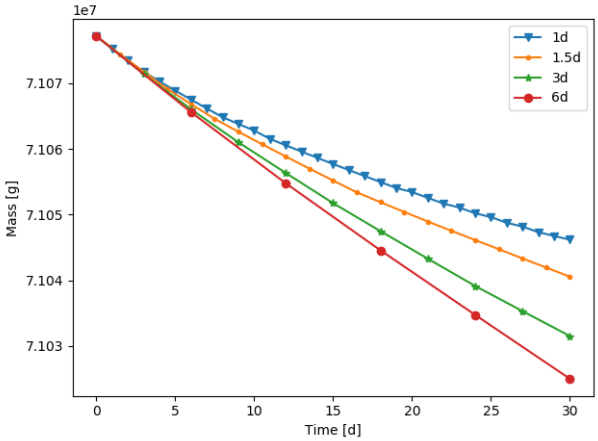

Figure 4.2: Thorium-232 mass over time using various depletion step sizes with steady batchwise reprocessing.

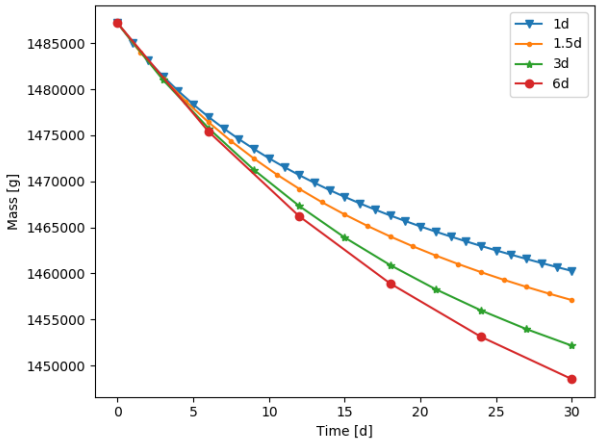

Figure 4.3: Uranium-233 mass over time using various depletion step sizes with steady batchwise reprocessing.

Because xenon-135 has a very high thermal neutron absorption cross section, it is useful to see how its mass varies. It can be seen that it is at a steady state value which is the same for each of the different depletion step sizes. Previously, it was shown that the relative error should scale based on the depletion step size, yet these values are the same. The reason for this is because the cycle time of xenon in the system is 20 seconds, while the smallest depletion step size is significantly larger than that. This results in a situation where the mass of xenon-135 builds up, stabilizes, and is reprocessed. In order to view the actual steady state of xenon-135 for the cycle time of 20 seconds, a depletion step size would have to be on the order of seconds instead of days.

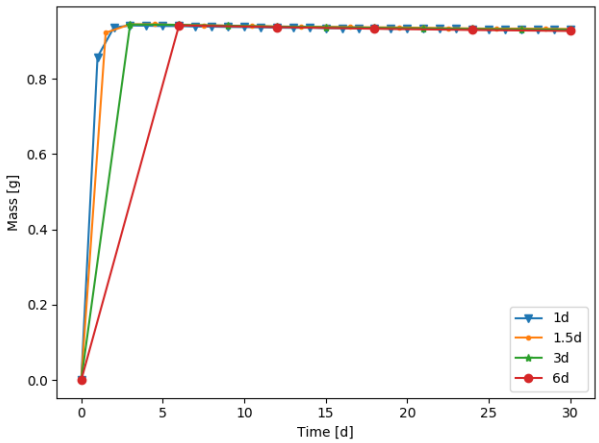

Figure 4.4: Xenon-135 mass over time using various depletion step sizes with steady batchwise reprocessing.

After xenon-135 absorbs a neutron, it can form xenon-136. By monitoring the concentration of xenon-136 in the system, the amount of xenon-135 which has parasitic absorptions can be understood more clearly, as there is a directly proportional relationship. Figure 4.5 shows how this concentration varies over time, which can clearly

show the difference between the depletion step sizes. The steady state concentration of xenon-136, which directly correlates to the amount of parasitic absorption by xenon-135, directly correlates to the depletion step size. The ratio of depletion step sizes is somewhat close to the ratio of the steady state concentrations, with the ratio of 1.5, 3, and 6 day depletion steps to the 1 day depletion steps being 1.53, 3.1, and 6.3, respectively.

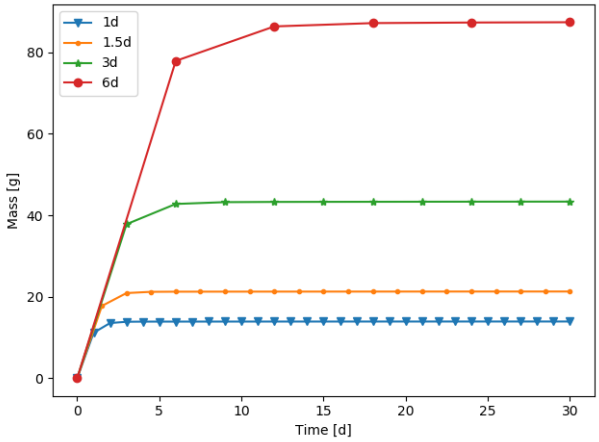

Figure 4.5: Xenon-136 mass over time using various depletion step sizes with steady batchwise reprocessing.

4.1.1.2.2 Feed and Removal Rates Figures 4.6 and 4.7 show the feed rates of thorium-232 and uranium-233, respectively. As previously discussed, the overall feed rate has the objective of maintaining constant mass. This explains why the feed rate for longer depletion steps is smaller than the feed rates associated with shorter depletion steps, as the reprocessing is less frequent, leaving more parasitic absorbers in the core.

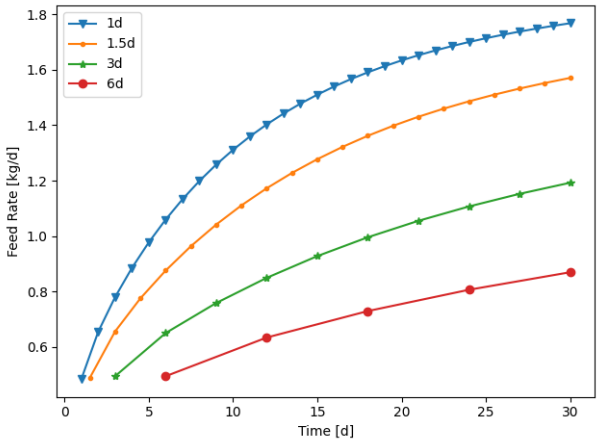

Figure 4.6: Thorium-232 feed rate over time using various depletion step sizes with steady batchwise reprocessing.

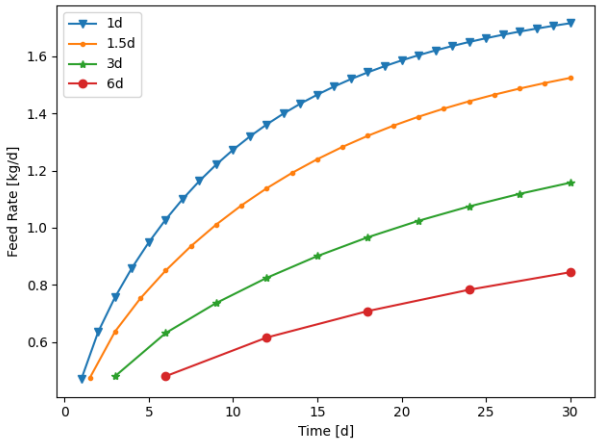

Figure 4.7: Uranium-233 feed rate over time using various depletion step sizes with steady batchwise reprocessing.

Figures 4.8 and 4.9 show the removal rates of xenon-135 and xenon-136, respectively. The xenon-136 removal rate is fairly straightforward to understand, as it was previously shown that the steady state concentration of xenon-136 is higher for the longer depletion step sizes. This means that the removal rate would also need to be higher, since the amount removed is proportional to the amount present in the system. For the xenon-135, the removal rate relationship may not immediately be clear since the steady state mass of xenon-135 after reprocessing is the same for the different depletion step sizes. However, the removal rate is higher for the shorter depletion step size because the xenon-135 builds up to a steady state value before reprocessing occurs. This means that every time reprocessing occurs, roughly the same amount of xenon-135 is removed. Thus, more frequent reprocessing yields an increased removal rate.

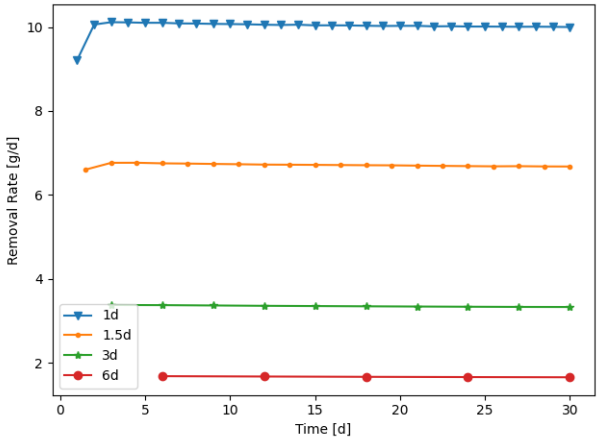

Figure 4.8: Xenon-135 removal rate over time using various depletion step sizes with steady batchwise reprocessing.

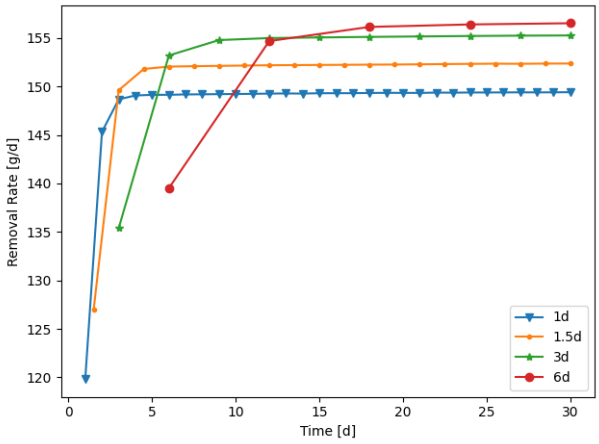

Figure 4.9: Xenon-136 removal rate over time using various depletion step sizes with steady batchwise reprocessing.

4.1.2 Continuous Reprocessing

For the batchwise methods, depletion time steps beyond 6 days are not considered. This is because a larger time step causes more error, and it can be seen that the difference between the 6 day depletion step and the smaller depletion steps is non-negligible.

For the continuous methods, the error buildup from using larger depletion steps is purely due to not updating various simulation data rather than from lack of updating the material compositions based on reprocessing. Due to this, larger time steps can be considered. In order to determine the impact of the updated simulation data, initially chosen depletion time steps of 1, 3, 6, 15, and 30 days are implemented. Additionally, although different continuous methods have been introduced, the depletion step mesh refinement only needs to consider one of the methods. This is because the continuous methods used all have the same general form and only alter the specific reprocessing constants implemented. Thus, the direct linear method is selected to determine the optimal depletion step size for the continuous methods. Additionally, it has been used in several other works and is very similar to the Cycle Rate method [48, 33, 49].

4.1.2.1 Smaller Depletion Steps

The results from using depletion steps of 1, 3, 6, 15, and 30 days can be seen in Figures 4.10 through 4.14. The results regarding the effective multiplication factor show that, after 30 days, all of the effective multiplication factors are within stochastic error of each other.

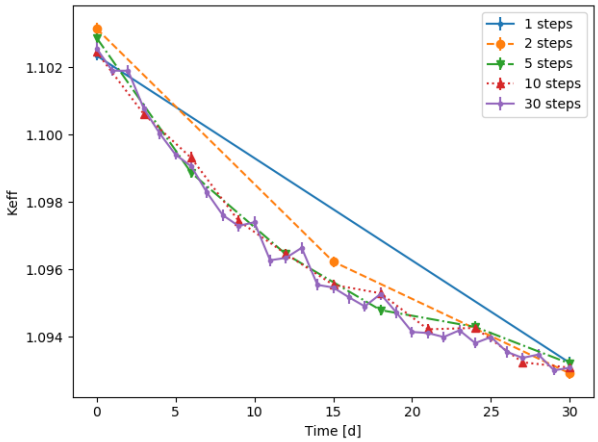

Figure 4.10: k_{eff} over time using various depletion step sizes with direct linear continuous reprocessing.

For the thorium-232 and uranium-233 masses, the values align within fractions of a percent of each other after 30 days of depletion. Additionally, it can be seen that there are sudden jumps in the thorium-232 mass. This is because of rounding which occurs due to a limit to the precision of the data which is available from Serpent2.

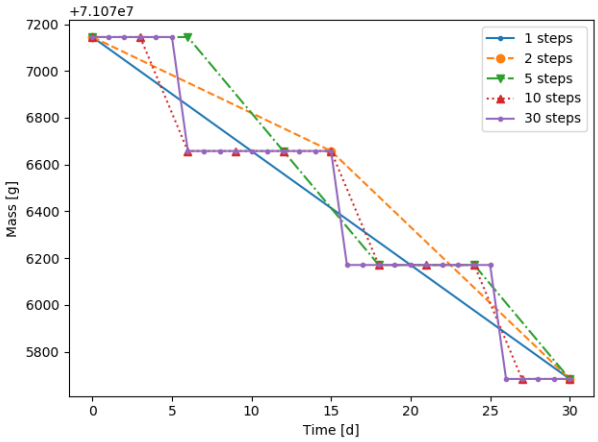

Figure 4.11: Thorium-232 mass over time using various depletion step sizes with direct linear continuous reprocessing.

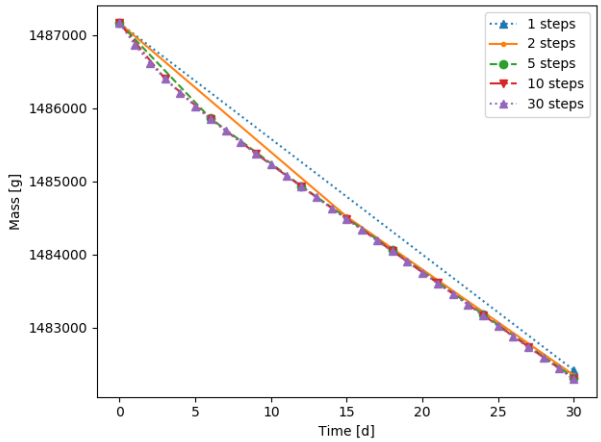

Figure 4.12: Uranium-233 mass over time using various depletion step sizes with direct linear continuous reprocessing.

The xenon-135 steady state mass is the same for each of the different step sizes, which is expected. Additionally, the xenon-136 steady state mass is also the same for the different depletion step sizes, showing that the capture rate of the xenon-135 does not vary significantly even when adjusting the depletion step size from 1 day to 30 days.

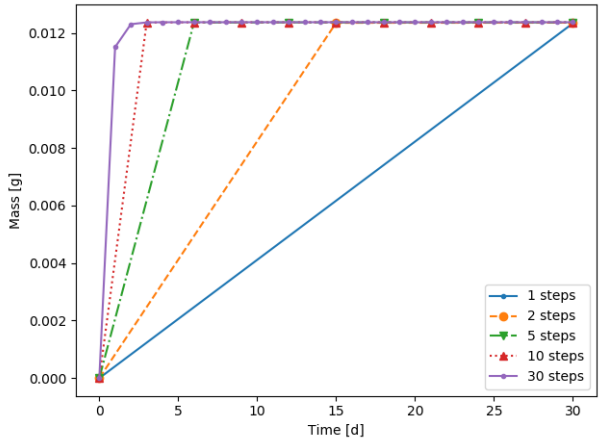

Figure 4.13: Xenon-135 mass over time using various depletion step sizes with direct linear continuous reprocessing.

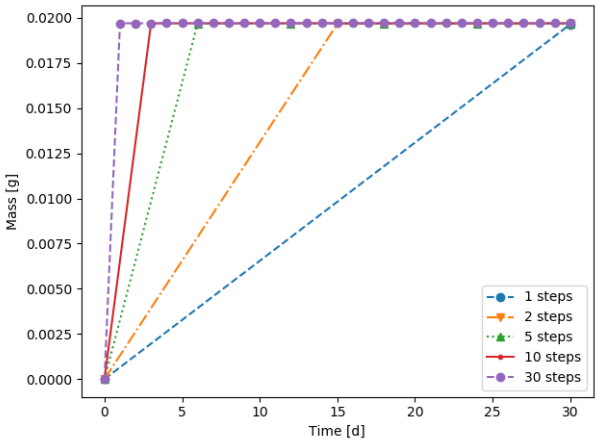

Figure 4.14: Xenon-136 mass over time using various depletion step sizes with direct linear continuous reprocessing.

Because these results are all very close to each other, it is prudent to test for larger depletion step sizes in order to determine how much computational cost reduction can be afforded while maintaining high accuracy in the results.

4.1.2.2 Larger Depletion Steps

The next set of depletion step sizes selected are 30, 60, 120, and 360 days. The smallest step size, 30 days, is selected in order to properly compare the larger depletion step sizes with a step size which has been shown to give results which are very close to those of a 1 day depletion step size. The results can be seen in Figures 4.15 through 4.19.

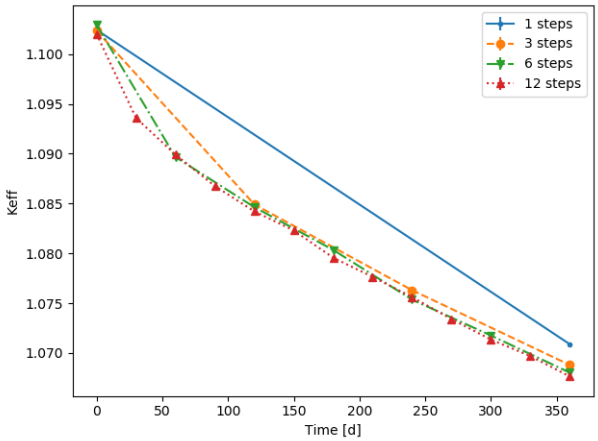

Figure 4.15: k_{eff} over time using various depletion step sizes with direct linear continuous reprocessing.

The 60 day depletion time effective multiplication factor at 360 days is not within stochastic error of the value for the 30 day depletion time. The error bounds are only 5 pcm apart, but since the computational cost decrease is only a factor of two, the 30 day depletion step size is sufficiently large for the cases implemented in this work while

maintaining a high level of precision and are used as the default when not specified.

The thorium-232 and uranium-233 masses follow a similar trend where the larger depletion step size has a higher mass of both isotopes. The difference in thorium can be seen by visualizing how the larger step sizes operate similarly to a tangent line from the shorter depletion step sizes. This is because the cross section and spectral data is not updated, and so the same rate of decrease is used over the entire depletion step. Because the system starts at beginning of cycle, the thorium is not yet in steady state and more is burned than added, so the decrease is expected.

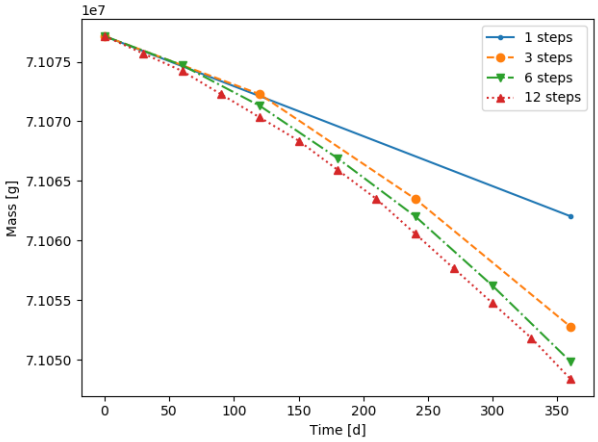

Figure 4.16: Thorium-232 mass over time using various depletion step sizes with direct linear continuous reprocessing.

For the uranium-233, a similar trend can be seen, though it is less pronounced. The main difference with thorium-232 is the fact that the thorium mass changes due to feed and neutron absorption, whereas the uranium-233 has feed, neutron absorption, and accumulation from the decay chain of the thorium-232 absorption. With this in mind, the difference of the uranium-232 and thorium-233 figures becomes more clear. For the shorter depletion steps, a similar trend as in the thorium-232 occurs where the rate is matches for the depletion step. However, the thorium-232 is also a factor, and since the thorium-232 has a greater mass for larger depletion step sizes, this results in more uranium-232 being bred as well.

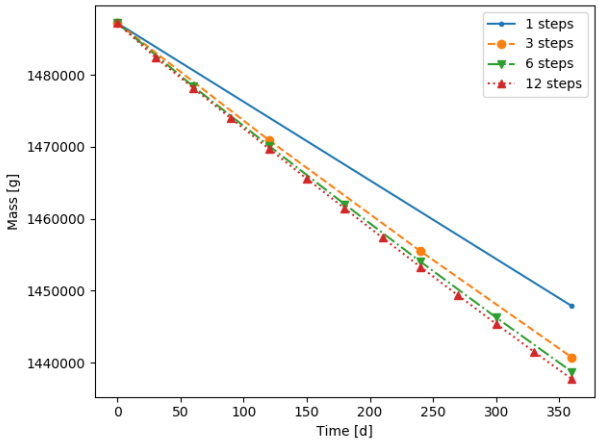

Figure 4.17: Uranium-233 mass over time using various depletion step sizes with direct linear continuous reprocessing.

The xenon-135 and xenon-136 masses match closely, with differences on the order of milligrams. This agrees well with the previous results generated as well. The reason the results agree well is because the xenon-135 concentration is expected to approach a constant result with increasing flux. This is because the xenon-135 comes from the decay of iodine-135, which scales proportionally to the flux. However, the xenon-135 has a large capture cross section, and is altered into xenon-136 proportionally to the flux as well. This shows why there is not a large difference even though there is a difference of 324 pcm between the effective multiplication factors of the 360 and 30 day depletion step sizes.

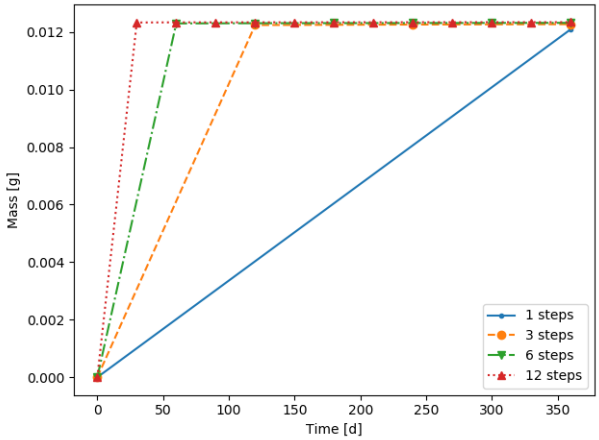

Figure 4.18: Xenon-135 mass over time using various depletion step sizes with direct linear continuous reprocessing.

The xenon-136 is also captured with a rate proportional to the flux, though the capture cross section is significantly lower than that of xenon-135 for the thermal spectrum used in the MSBR. This can be seen in the mass of xenon-136, which has a higher steady state concentration that that of xenon-135. The main limiting factor on both

of these xenon isotopes is the reprocessing term, which keeps the steady state mass of both much smaller than they would be without the reprocessing.

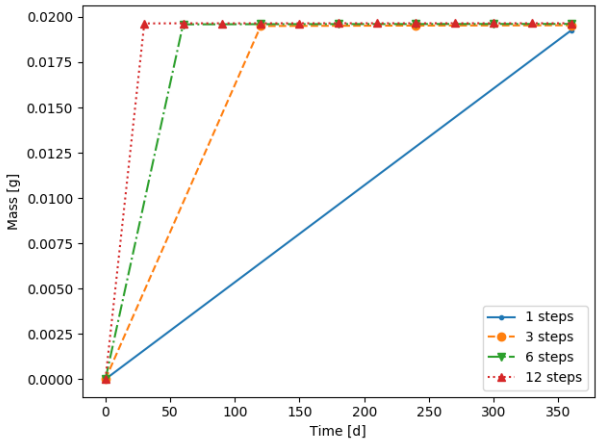

Figure 4.19: Xenon-136 mass over time using various depletion step sizes with direct linear continuous reprocessing.

4.1.3 Variable Depletion Step Size

Previously, the depletion step size needed for a reasonably converged result was investigated using a constant depletion step size. However, it is understood that the reactor will experience some rapid initial changes, after which the changes slow and eventually the reactor reaches steady state. Therefore, it is reasonable to expect that in the early times, small depletion step sizes should be used, while at later times, larger depletion step sizes should be used.

Because it was previously shown for continuous reprocessing that the depletion step sizes smaller than 30 days give results within stochastic error for the effective multiplication factor, the 30 day depletion step size will be the smallest considered step size. Several cases are considered over a net depletion time of 360 days in order to determine the effects of varying the depletion step size at various points in the simulation. Case A uses 12 30 day steps; Case B uses 6 30 day steps and 1 180 day step; Case C uses 3 30 day steps, 1 90 day step, and 1 180 day step; Case D uses 1 30 day step and 1 330 day step; Case E uses 1 180 day step and 6 30 day steps; and Case F uses 1 360 day step. The results for each case at the time value of 360 days are given in Table 4.1.

Table 4.1: Variable Depletion Step Size Results

Case	k_{eff}	Δk_{eff} [pcm]	Simulations
A	1.06801 ± 0.00017	-	12
B	1.06844 ± 0.00015	43 ± 32	7
C	1.06857 ± 0.00016	56 ± 33	5
D	1.07034 ± 0.00016	233 ± 33	2
E	1.06846 ± 0.00016	45 ± 33	7
F	1.07071 ± 0.00017	270 ± 34	1

For the result of the effective multiplication factor, the difference from Case A was taken for the rest of the cases. These results show that Cases B, C, and E are all very close to Case A. Cases D and E are the outliers, which is also reflected in the number of simulations run. Case D uses a short depletion time step initially followed by a single long step, which should have allowed for the simulation data to update based on the presence of fission products which were not initially present. However, this impact is likely offset by the continuous reprocessing, resulting in more impact purely due to the general changes in the system. It seems that multiple steps are needed during the first 360 days of operation in order to properly account for the changes within the reactor, such as Case E, which leads with a depletion step size of 180 days but then proceeds to follow it with six 30 day depletion steps.

The variable depletion step size could be considered to have an impact if the reactor were at steady-state, but after only 360 days of operation, there are still changes occurring [39]. This is the reason why having more simulations yielded better results than the cases which had fewer simulations.

4.1.4 Computational Cost Analysis

In order to provide a useful measure of computational cost, Table 4.2 is generated. This table represents the compuational cost of forming the depletion step mesh refinement study in the units of "number of simulations required," as this is the largest impact on run time. The depletion step size used by the batchwise methods is 3 days, while the continuous method uses 30 day depletion steps.

The cost for SaltProc based batchwise methods can be calculated by first accounting for all internal depletion times. These all require 2 simulations when run using Serpent2, as the pre-reprocessed simulation is also run. An optimized batchwise method could remove the double running, which would significantly reduce computational

cost. Additionally, as can be seen in the forumla for calculating the cost, reduction can be achieved by maximizing the depletion step size or reducing the net depletion time, or T_{net}.

Table 4.2: Computation Cost Using Constant Depletion Steps

Method	Cost Formula	30 days	1 year	10 years
SaltProc Batchwise	$2T_{net}\Delta t^{-1}$	20	244	2435
Optimized Batchwise	$T_{net}\Delta t^{-1}+1$	11	123	1219
Continuous (All)	$T_{net}\Delta t^{-1}+1$	2	14	123

Because continuous reprocessing enables the use of depletion time steps on the order of 10x larger while retaining a similar level of precision, the computational cost is significiantly lowerer than the batchwise computational cost.

4.2 Advanced Protactinium Decay Model

The original method used by Rykhlevskii to model the decay of protactinium-233 into uranium-233 was to feed a mass of uranium-233 into the core equivalent to the mass of protactinium-233 pumped out every 3 days [39]. With the implementation of continuous reprocessing methods, the breeding can be more accurately modeled. In the new method, protactinium-233 is continuously removed and allowed to decay. Once it decays, the uranium-233 is then continuously added back into the core.

It is expected for the largest impact to be at beginning of cycle, while at steady state this improved method will likely have only a minor effect on the results. The results of this analysis can be seen in Figures 4.20 through 4.24, where the continuous method implemented for both was the Direct Linear method, and 305 day depletion steps were used over a period of 3660 days, or roughly ten years. Although this depletion step size is larger than those previously used, it is expected to still give reasonable accurate results. Additionally, the discard and salt discard processes are not considered here, as only the differences between the different feed rate methods are being investigated.

The effective multiplication factor for the realistic uranium-233 feed approach can be seen to drop rapidly, and then begin leveling off after around 3 years. The reason for this difference in behaviour is because the feed which uses the SaltProc averaged value makes a steady state assumption, which allows for a full supply of uranium-233 available for feeding the reactor at all times. In can be seen that the slopes of the two different methods are beginning

to converge slowly after a few year, and it is expected that they would eventually match. However, if the batch discard processes are included, a true steady state equilibrium would not be reached, resulting in at least some minor differences between the methods.

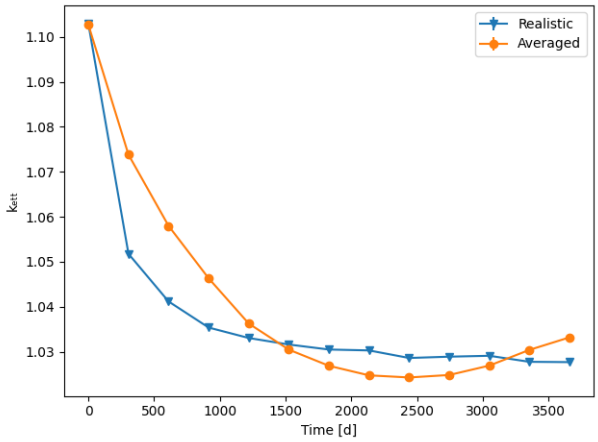

Figure 4.20: k_{eff} over time using various depletion step sizes with direct linear continuous reprocessing.

The thorium-232 masses have a fairly constant difference from each other over the entire run. The reason for this is likely due to the fact that there is less uranium-233 being added to the system with the realistic approach, meaning more interactions with thorium-232 occur instead. This results in a lower effective multiplication factor, at least initially, which provides increased uranium-233 production, and lower thorium-232 mass.

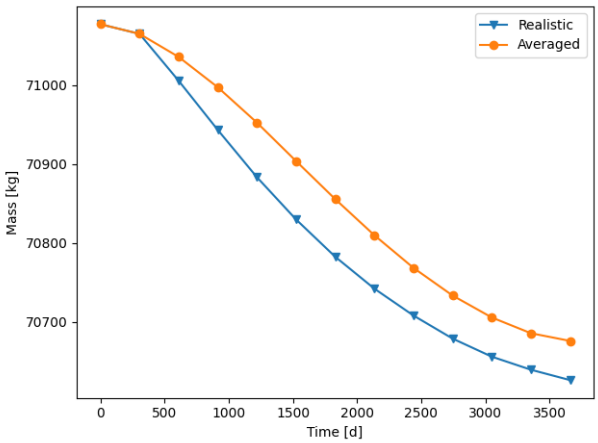

Figure 4.21: Thorium-232 mass over time using various depletion step sizes with direct linear continuous reprocessing.

The uranium-233 for the realistic method follows a trend which is expected, which is a sharp initial drop followed by a decrease in the rate of change. After the sharp drop, the protactinium decay tank fills, and fresh uranium-233 is supplied to the system in an increasing quantity. For the averaged method, because the uranium-233 was burned at

a higher rate, this leads to a decrease in the total amount. It is expected that there will be a few oscillations in the uranium-233 mass for the averaged system until it finally achieves a steady state value, though this is only true if the system only has continuous reprocessing processes. This oscillation is expected because initially, there is more uranium-233 which fissions, rapidly decreasing the quantity while maintaining a higher effective multiplication factor. This causes the uranium-233 to burn quickly, dropping the net mass until it reaches a point that the effective multiplication factor decreases while the uranium-233 mass is also lowered, reducing probability of fission as well as the number of neutrons. This allows the feed rate to build up more uranium-233 in the system, repeating the process.

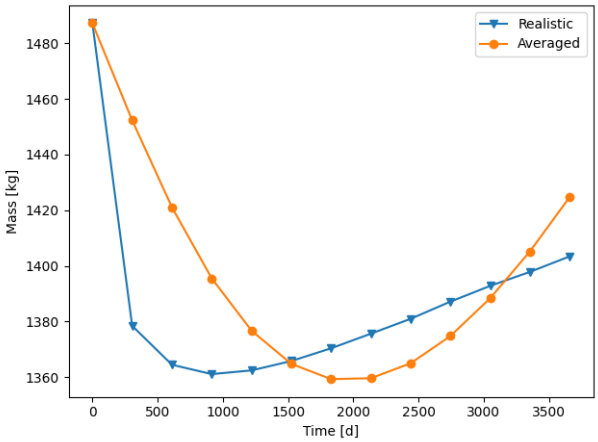

Figure 4.22: Uranium-233 mass over time using various depletion step sizes with direct linear continuous reprocessing.

For the xenon-135 and xenon-136, the masses are slightly smaller than the averaged method initially, which is due to the decreased uranium-233, resulting in fewer fission products being generated. However, after more time passes, the values match. This is due to the steady state value of xenon-135 not significantly changing when the flux is already large.

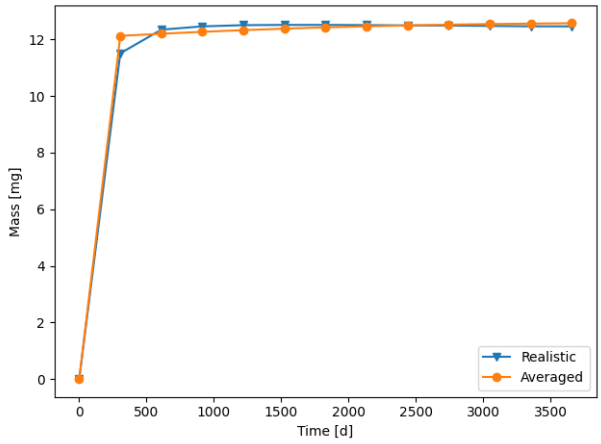

Figure 4.23: Xenon-135 mass over time using various depletion step sizes with direct linear continuous reprocessing.

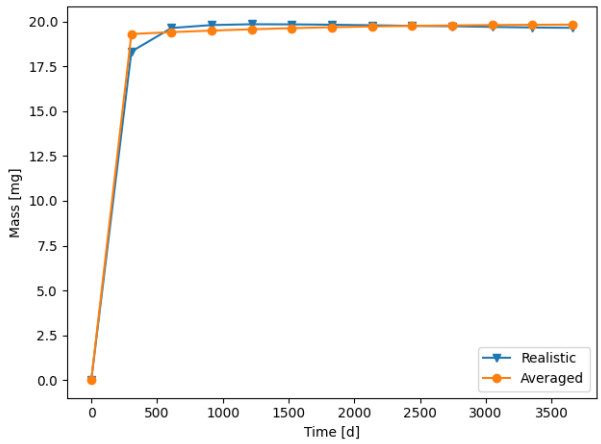

Figure 4.24: Xenon-136 mass over time using various depletion step sizes with direct linear continuous reprocessing.

4.3 Comparisons of Methods

There are many different works which have implemented batchwise reprocessing computational methods to simulate a continuous, online reprocessing scheme; many of which are discussed in the literature review. Thus, the question of how accurate these models are should be resolved. From previous work by Rykhlevskii as well as through the depletion step mesh refinement provided in this work, it has been shown that increasing the depletion step size causes batchwise methods to become less effective at simulating the physical, continuous process of online reprocessing [40].

4.3.1 Continuous Methods

Although only the Direct Linear continuous reprocessing method has been used so far due to similarities between the continuous methods, it is still useful to determine how the different continuous methods compare in the MSBR. For this comparison, the steady SaltProc-based averaged feed rates and modified cycle times are used as well as a 30 day depletion step size.

The effective multiplication factor, over the course of 360 days, has a difference on the order of 500pcm between the Cycle Time Decay, CTD, method and the Direct Linear and Cycle Rate, DL and CR, methods respectively. The Direct Linear and Cycle Rate methods are very close to each other, which is expected since the reprocessing constants are also very similar. The control method, CTRL, shows the effects of no reprocessing and no feed.

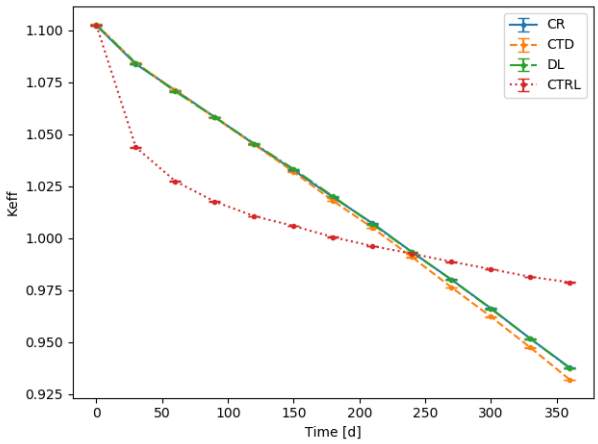

Figure 4.25: k_{eff} over time comparing different continuous reprocessing methods.

The reason the continuous methods shown here continue decreasing and are eventually overtaken by the control reprocessing is due to the protactinium reprocessing. Because the protactinium in these methods is continuously removed and the uranium-233 feed is averaged only over the first 90 days of the steady batchwise method, it is expected that the continuous model will become inaccurate some time after 90 days. This accuracy can be improved by implementing the advanced protactinium decay model which removes the dependency on an average uranium-233 feed rate.

However, the main purpose of this figure is to demonstrate how the different continuous methods compare to each other over a long period of time. Additionally, this figure shows how the effective multiplication factor without reprocessing drops below critical after roughly 300 days, and that having either a good estimate for the uranium-233 feed rate or a continuous method to generate the uranium-233 feed rate is very important for depletion modeling of the MSBR.

4.3.2 Continuous and Batchwise Methods

The continuous method selected for the comparison is the Direct Linear reprocessing method. For the batchwise methods, the bulk and steady batchwise reprocessing schemes from SaltProc early and current versions are used. For the bulk batchwise reprocessing, the cycle times from the Robertson report are followed [37]. Additionally, the uranium-233 and thorium-232 feed rates in the continuous model use the values from the bulk batchwise results [39].

For the steady batchwise comparison, the modified cycle times are used which include an efficiency term on the xenon, krypton, and protactinium reprocessing constants. For the continuous comparison, the average thorium-232 and uranium-233 feed rates are generated from the 90 day runs and are implemented.

For all of these comparisons, the Direct Linear reprocessing method uses the average feed rate of the given batchwise method and the cycle times from the Robertson report [37]. The SaltProc Direct Linear reprocessing method, which is intended to be a continuous approximation of the batchwise reprocessing method, will match the reprocessing constants used in the given batchwise method.

If the results are sufficiently close, then it may be reasonable for a continous method to be implemented in place of a batchwise method when the reprocessing scheme calls for a physically batchwise processes. This would allow for reduced computational cost and potentially large depletion step sizes to be implemented. If the results are not close, then these results will provide information on how the difference in reprocessing methods can affect results.

4.3.2.1 Continuous and Bulk Batchwise Methods

In this comparison, there are three different methods considered. These methods are the continuous Direct Linear, DL, method, the continuous SaltProc Direct Linear, SPDL, method; and the bulk batchwise SaltProc method. Figure 4.26 shows how the effective multiplication factor varies over time for these methods. It can be seen that there is a jump at around 50 days for the bulk SaltProc data, which is caused by removal of the rare earths, as shown in the MSBR reprocessing scheme [37]. However, it is clear to see here that the continuous methods are not able to match the bulk batchwise reprocessing results very closely at all, even using the SaltProc Direct Linear method which attempts to replicate the same reprocessing constants.

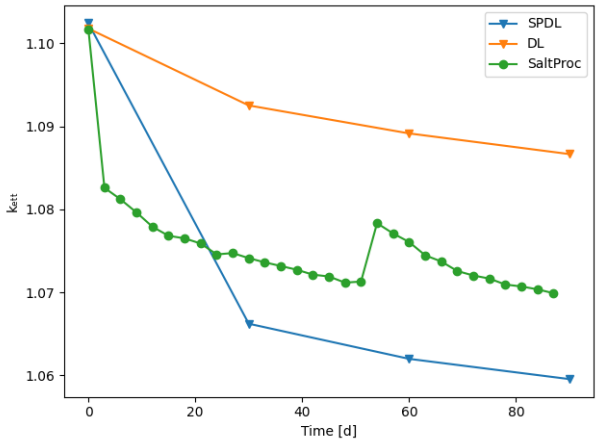

Figure 4.26: k_{eff} over time comparing bulk batchwise results to continuous methods.

Figures 4.27 and 4.28 show the mass of thorium-232 and uranium-233 over time, respectively. Overall, the masses of both isotopes are within a few kilograms, and do not have a large impact on the differences between the methods. For example, the spike in thorium-232 mass at 50 days when the rare earths are removed does not have nearly as much impact as the removal of the rare earths does on reactor behaviour.

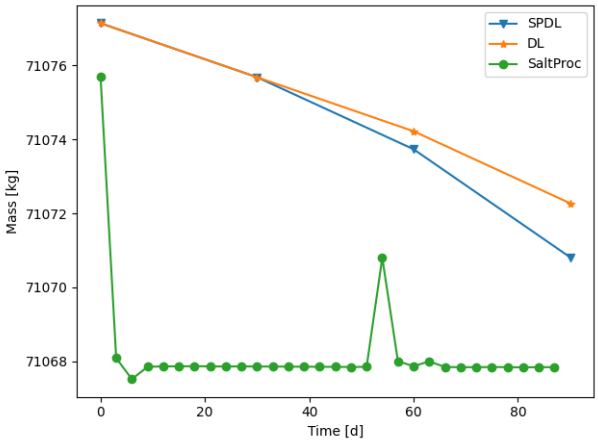

Figure 4.27: Thorium-232 mass over time comparing bulk batchwise results to continuous methods.

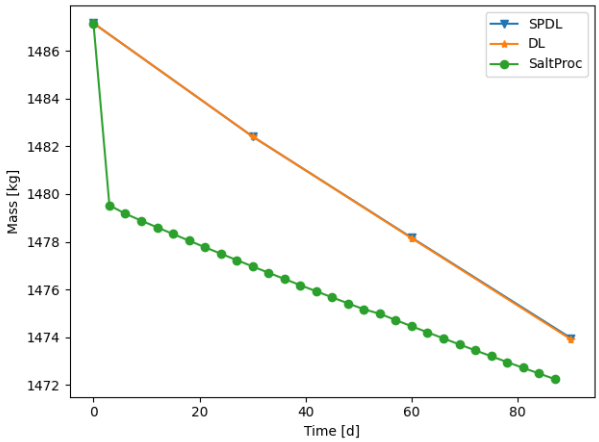

Figure 4.28: Uranium-233 mass over time comparing bulk batchwise results to continuous methods.

Figures 4.29 and 4.30 show how the xenon-135 and xenon-136 masses, respectively, change over time for the different methods. From these figures, it can be seen that the SPDL continuous method does a fairly good job of matching the bulk batchwise reprocessing method's masses. However, the SPDL method allows for slightly more fission product poisons to exist, such as xenon-135, which allows for more parasitic absorption, into isotopes such as xenon-136. This is the reason that the SPDL effective multiplication factor is lower than that of bulk SaltProc.

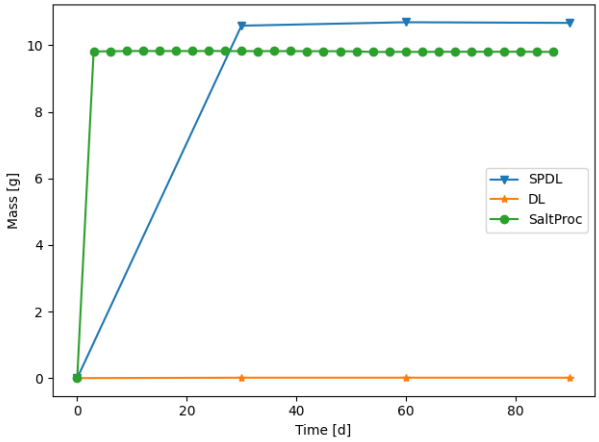

Figure 4.29: Xenon-135 mass over time comparing bulk batchwise results to continuous methods.

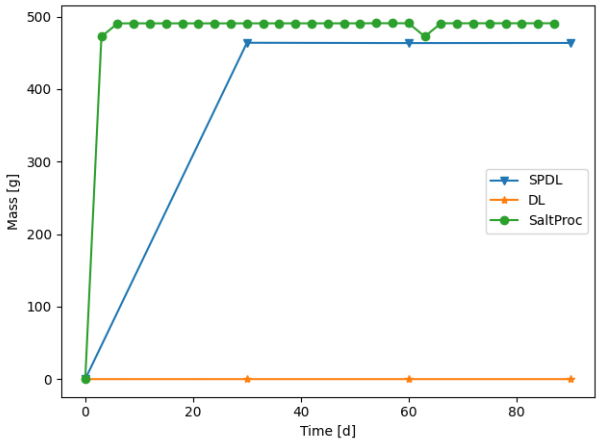

Figure 4.30: Xenon-136 mass over time comparing bulk batchwise results to continuous methods.

From these results, it can be concluded that using continuous reprocessing to perform a bulk reprocessing scheme will not yield same results. This also means that attempting to simulate a continuous reprocessing scheme using a bulk batchwise method will not yield results that are the same as a continuous reprocessing method.

4.3.2.2 Continuous and Steady Batchwise Methods

The next set of comparisons are between the steady batchwise results from SaltProc and two continuous methods, Direct Linear and SaltProc Direct Linear. The effective multiplication factor over time for each of the three methods can be seen in Figure 4.31. It can be seen that the Direct Linear continuous k_{eff} is closer to the steady batchwise k_{eff} than the SaltProc Direct Linear result. Among figures 4.32 through 4.35, the plot which demonstrates the reason behind this is the uranium-233 mass over time.

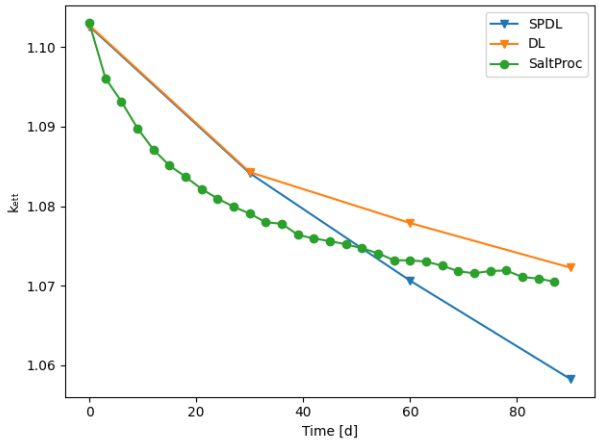

Figure 4.31: k_{eff} over time comparing steady batchwise results to continuous methods.

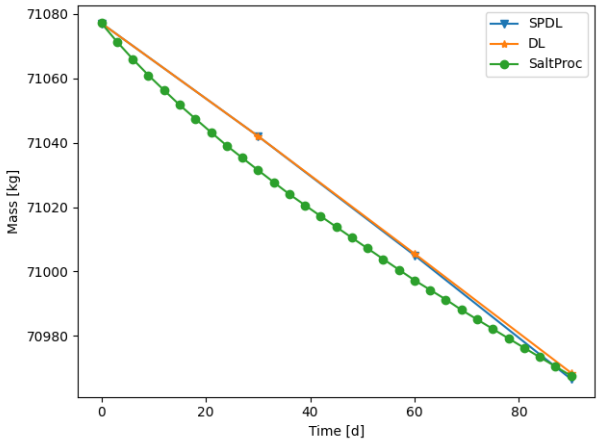

Figure 4.32: Thorium-232 mass over time comparing steady batchwise results to continuous methods.

The uranium-233 mass over time for the different methods reflects very closely with the effective multiplication factors. The uranium-233 comes from beta decay of protactinium-233, where protactinium is one of the elements designated to be removed in the MSBR reprocessing scheme. For Direct Linear, the removal of protactinium is approximated using 100% removal over three days. For SaltProc Direct Linear, the removal is instead approximated to 9.5% over 3 days, which is the value used in steady SaltProc. However, because the SPDL method is continuously removing the protactinium, this means that there is less protactinium-233 which decays into uranium-233 when compared to the steady batchwise method in SaltProc.

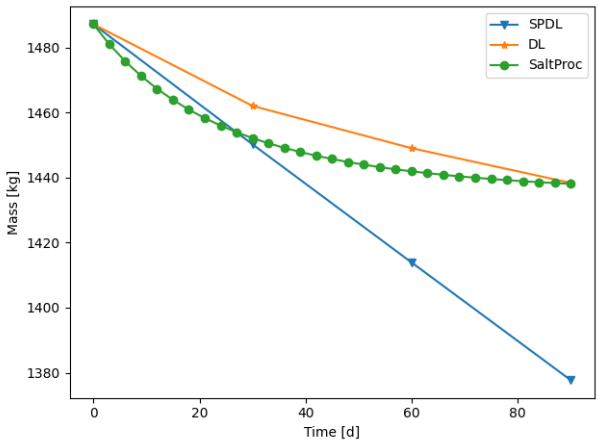

Figure 4.33: Uranium-233 mass over time comparing steady batchwise results to continuous methods.

A similar effect with the protactinium can be seen in Figures 4.34 and 4.35. Although the SPDL method is using a reprocessing constant which should allow for matching SaltProc, the continuous removal results in fairly large discrepancies. The xenon-135 difference is only on the order of tenths of a gram, but because the quantity is

constantly suppressed in the continuous SPDL method, the overall capture is reduced drastically, resulting in a difference of roughly 40 grams in the xenon-136 mass.

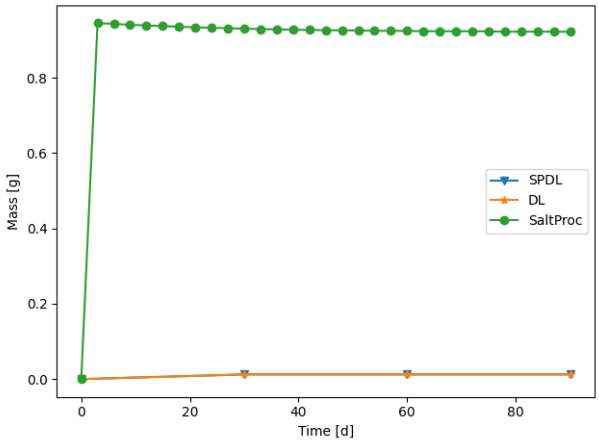

Figure 4.34: Xenon-135 mass over time comparing steady batchwise results to continuous methods.

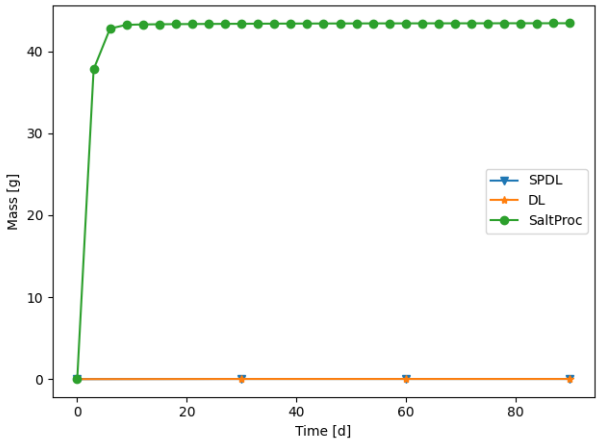

Figure 4.35: Xenon-136 mass over time comparing steady batchwise results to continuous methods.

Overall, using a continuous method to approximate a steady batchwise method does not seem generally viable. This is because isotopes which are continuously removed have a very different behaviour from isotopes which are removed after interactions occur over an entire depletion step. If the reprocessing scheme calls for a batchwise process, it may be important to reactor behaviour to allow for those capture or decay processes to occur without influence from reprocessing.

4.3.3 Overall Differences

Figures 4.36 through 4.40 show how the previous work, using a steady batchwise reprocessing method with the average uranium-233 feed, and the current work, using the direct linear continuous method with the realistic protactinium decay, compare with each other. The previous work uses the long term bulk average uranium-233 feed rate, and implements altered cycle rates for protactinium, xenon, and krypton. The continuous method uses only the cycle time data from the Robertson report to generate the reprocessing constants [37].

The effective multiplication factor for the two different approaches initially has the previous work falling faster. This is due to the initial buildup of fission product poisons, while the continuous reprocessing allows for these poisons to be removed more efficiently. However, as time goes on, the lacking uranium-233 feed due to the empty protactinium decay tank causes the current work to dip below the previous work. Over the course of three months, the longer lived behaviour cannot be seen. However, the section covering the advanced protactinium decay model in more depth shows the results over the course of several years and reveals that the reactivity difference due to the different refueling methods eventually decreases.

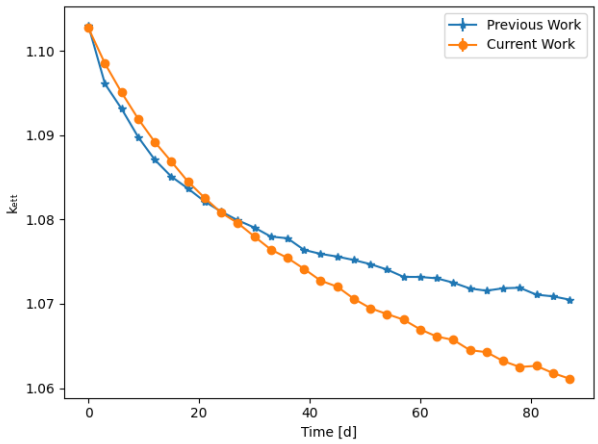

Figure 4.36: k_{eff} over time with three day depletion steps for steady batchwise and Direct Linear continuous reprocessing using different uranium-233 feed schemes.

The thorium-232 feed rate for both approaches is the same, yet the previous work burns it slightly faster than the current work. This effect is likely caused by the presense of more uranium-233 in the previous work, which allows for a faster burn rate of the thorium-233 due to a higher net flux.

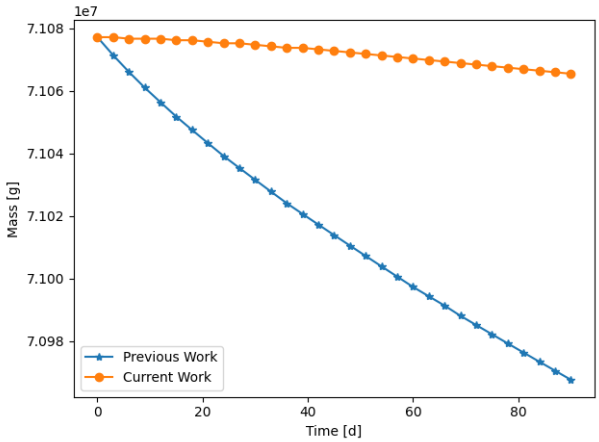

Figure 4.37: ^{232}Th over time with three day depletion steps for steady batchwise and Direct Linear continuous reprocessing using different uranium-233 feed schemes.

This difference in the uranium-233 masses is primarily due to the effects of the different uranium-233 feed rate approaches implemented in each of the different works.

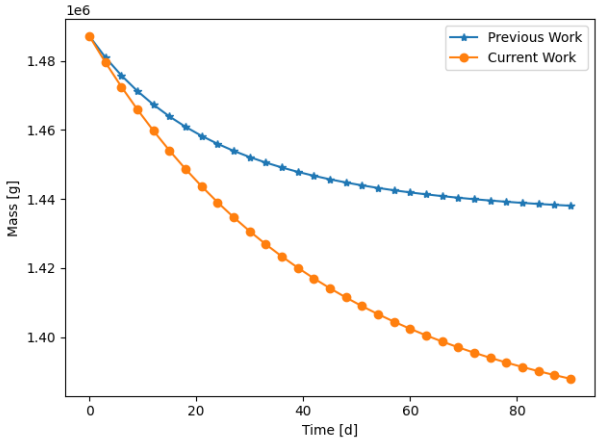

Figure 4.38: ^{233}U over time with three day depletion steps for steady batchwise and Direct Linear continuous reprocessing using different uranium-233 feed schemes.

For the xenon-135 and xenon-136, these results are primarily impacted due to the continuous and reprocessing methods implemented.

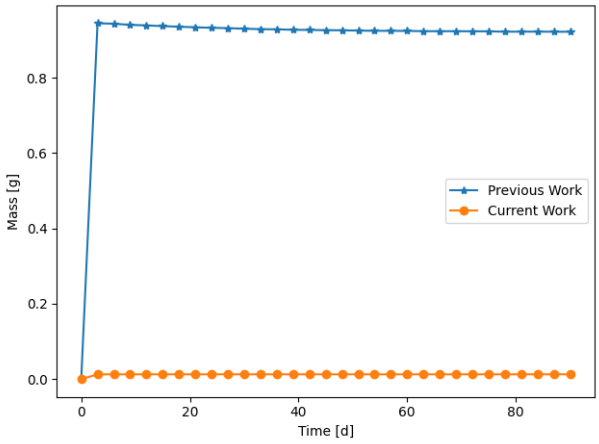

Figure 4.39: ^{135}Xe over time three day depletion steps for steady batchwise and Direct Linear continuous reprocessing using different uranium-233 feed schemes.

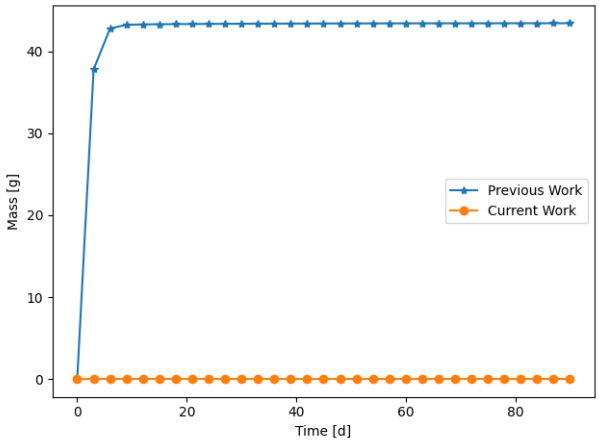

Figure 4.40: ^{136}Xe over time three day depletion steps for steady batchwise and Direct Linear continuous reprocessing using different uranium-233 feed schemes.

Overall, these results show the combined effects of implementing continuous reprocessing methods with a continuously updated uranium-233 refeed tank which is initially empty. This is compared with batchwise reprocessing methods and an averaged uranium-233 feed rate.

4.4 Mass Balancing

The continuous reprocessing method allows for larger depletion steps to be used without losing out on the impact of reprocessing while also allowing for fewer simulations to be run when compared to batchwise methods. However, one potential issue which is possible for continuous methods but not typically an issue for batchwise methods is mass balancing.

The effects of mass balancing are of particular note in the Serpent2 code because an increase in mass causes a corresponding increase in density due to constant volumes. A simple workaround to this is to either remove some fraction of mass to return the net mass to equilibrium or adjust the volumes so the density remains constant, thus maintaining physicality.

However, if the issue of mass balancing is not considered, then the results could be fairly inaccurate, potentially without anyone noticing. This is because changes in isotopic composition and densities are expected. The issue comes about when these changes are caused by adding more material to a space than is physically possible, thus generating non-physical results.

4.4.1 Impact Minimization Using Modified Feeds

One aspect investigated is minimizing the change in the net mass by only altering the average feed rates implemented. It is determined through repeated simulations that increasing the thorium-2323 feed rate drives the second derivative of the net mass negative. The uranium-233 feed increasing drives the second derivative to be positive. The reason for these effects are due to the change in the effective multiplication factor, mass of the added feed, and removal of fission products by reprocessing.

Using these relationships, a process to determine the feed rates to minimize the change in the net mass is determined. The process is as follows: hold one feed rate constant while adjusting the other; if the final net mass is higher than initial net mass, then reduce the feed rates and vice versa; once the final net mass is same as initial net mass, make adjustment based on second derivative; if the second derivative is positive, increase thorium-232 or decrease uranium-233 feed; if the second derivative is negative, increase uranium-233 or decrease thorium-232 feed; iterate until second derivative is zero and net mass remains roughly constant.

After iterating through this processes several times using a depletion time step of 300 days over 10 steps, the modified average feed rates of 1.9 kg/day of thorium-232 and 2.16 kg/day of uranium-233 were determined to result in a very small change to the net mass on the order of 0.02%.

4.4.2 Actual Impact

Previously, the maximum impact of improper mass balancing was calculated and was shown to have a 21% difference impact on the density of thorium-232. However, it is expected for the thorium-232 mass to change over time, so only viewing the mass of thorium is not very useful. Instead, viewing the net mass of the system will allow for insight into how the overall system behaviour could be expected to change due to density variations.

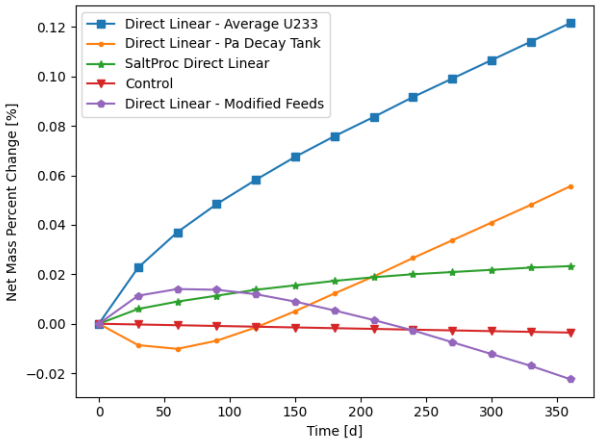

Figure 4.41: Change in net mass of the fuel salt for various methods.

Figure 4.41 shows the evolution of the net mass of the fuel salt over time while using a 30 day depletion time step over 360 days. The method which has the largest change in the net mass is the Direct Linear method with the averaged uranium-233 feed rate. The net mass difference is roughly 198 kilograms, which appears to be fairly substantial. However, the approximate impact upon the simulation accuracy is dependent on the density. Equation (4.1) shows how the final density is impacted by the added mass, where the initial fuel salt density, ρ_0, is 3.35 grams per cubic centimeter.

$$\rho_f = \frac{m_0 + \Delta m}{V} = \rho_0 + \Delta\rho \tag{4.1}$$

Continuing with the Direct Linear case using the averaged uranium-233 feed, the change in the density is on the order of four milligrams per cubic centimeter, yielding a final density percent difference of 0.12%. In terms of the impact on overall results, the change in density due to implementing the continuous method is fairly negligible unless the required simulation is of very high precision.

Chapter 5

Conclusions

5.1 Comparison of Continuous and Batchwise Reprocessing Methods

In this work, the differences and similarities between continuous and batchwise reprocessing have been discussed. Overall, it was shown that in terms of step size, computational cost, and steady state relative error, the continuous reprocessing method yields better results when handling continuous online reprocessing of an MSR.

For step sizes, it was shown mathematically how implementing larger depletion step sizes with a batchwise reprocessing method causes the relative steady state error to become larger. It was also shown that this error corresponds to the cycle time of the elemental target, meaning that the maximum depletion time step which will yield valid results is dependent upon the reprocessing scheme of the given reactor. For continuous reprocessing, the depletion time step is instead limited by updating the simulation data and does not depend on the reprocessing scheme, allowing for greater flexibility.

5.2 Continuous Reprocessing Investigations

Because the continuous reprocessing method provides many benifits when compared to the batchwise reprocessing method, investigating weaknesses in the method is important. One of these weaknesses which was investigated in this work was the mass balancing of the fuel salt in the simulation.

DATA

This work also discussed how continuous methods compared to each other and how continuous methods could approximate a batchwise method. DATA

5.3 Future Work

There are a few areas in which this work should be continued. The first area of interest would be analysis of more specific works which have implemented batchwise reprocessing to simulate a continuous reprocessing scheme.

This analysis could include replicating the results using continuous reprocessing, determining what level of error to expect for different isotopes, or analysis of depletion step size implemented in comparison to the reprocessing scheme cycle times provided. Alternatively, works which have implemented continuous reprocessing could have the net mass and feed rates analyzed to determine if the simulation may have had any unexpected results due to density variations which would not have physically occured.

Another area of interest is longer net depletion time analysis. This is more difficult to perform for the batchwise reprocessing methods, as the depletion time steps must be kept shorter and the SaltProc tool currently requires double running each time step. However, it would be useful to generate very long net depletion time results to compare with the steady state error mathematical model predictions. Additionally, the difference between various neutronics parameters could also be compared.

One final area which could be further explored is the error analysis, which in this work only covered steady state relative error of an isotopes concentration in a generic sense. Work could be done to determine the error at each time step to better analyze systems which have not achieved steady state. The error analysis could also include variable depletion time steps instead of requiring a set depletion time step to be valid.

Appendix A: Batchwise Steady State Mass Derivation

5.4 Derivation of Batchwise Reprocessing Steady State Mass

The first step of this derivation is to separate the λ term into the continuous λ_c, which is the continuous losses in the Bateman equation, and the batchwise reprocessing λ_r.

$$\lambda = \lambda_c + \lambda_r \tag{5.1}$$

Next, set up and solve the differential equation for the continuous portions of the process.

$$\frac{dm}{dt} = C - \lambda_c m \tag{5.2}$$

$$m(t) = \left(m_0 - \frac{C}{\lambda_c}\right) e^{-\lambda_c t} + \frac{C}{\lambda_c} \tag{5.3}$$

Change the continuous process to one which updates at discrete time steps.

$$m_{n+1} = \left(m_n - \frac{C}{\lambda_c}\right) e^{-\lambda_c \Delta t} + \frac{C}{\lambda_c} \tag{5.4}$$

Apply the batchwise reprocessing after each new discrete time step, treating each time step evaluated as a depletion time step.

$$m_{n+1} = \left(\left(m_n - \frac{C}{\lambda_c}\right) e^{-\lambda_c \Delta t} + \frac{C}{\lambda_c}\right)(1 - \lambda_r \Delta t) \tag{5.5}$$

Define some arbitrary variables for ease of use.

$$\alpha = \frac{C}{\lambda_c} \tag{5.6}$$

$$\epsilon = e^{-\lambda_c \Delta t} \tag{5.7}$$

$$\gamma = (1 - \lambda_r \Delta t) \tag{5.8}$$

Using these simplified variables, the mass can be calculated for several subsequent steps.

$$m_{n+1} = ((m_n - \alpha)\,\epsilon + \alpha)\,\gamma \tag{5.9}$$

$$m_0 = m_0 \tag{5.10}$$

$$m_1 = m_0\epsilon\gamma - \alpha\epsilon\gamma + \alpha\gamma \tag{5.11}$$

$$m_2 = m_0\epsilon^2\gamma^2 - \alpha\epsilon^2\gamma^2 + \alpha\epsilon\gamma^2 - \alpha\epsilon\gamma + \alpha\gamma \tag{5.12}$$

From these steps, a more general formula can be derived.

$$m_n = m_0\epsilon^n\gamma^n + \alpha \sum_{j=1}^{n} \gamma^j(\epsilon^{j-1} - \epsilon^j) \tag{5.13}$$

Moving towards the goal of the steady state mass, the value of n is taken to be approaching infinity. Because the values of γ and α are both between zero and one, simplifications can be made.

$$m_{ss} = m_0\epsilon^\infty\gamma^\infty + \alpha \sum_{j=1}^{\infty} \gamma^j(\epsilon^{j-1} - \epsilon^j) \tag{5.14}$$

$$m_{ss} = \alpha \sum_{j=1}^{\infty} \gamma^j(\epsilon^{j-1} - \epsilon^j) \tag{5.15}$$

The next step is to simplify the summation term by changing ϵ back to its original form.

$$m_{ss} = \alpha \sum_{j=1}^{\infty} \gamma^j(e^{-\lambda_c \Delta t(j-1)} - e^{-\lambda_c \Delta t j}) \tag{5.16}$$

$$m_{ss} = \alpha \sum_{j=1}^{\infty} \gamma^j e^{-\lambda_c \Delta t j}(e^{\lambda_c \Delta t} - 1) \tag{5.17}$$

$$m_{ss} = \alpha(e^{\lambda_c \Delta t} - 1) \sum_{j=1}^{\infty} \gamma^j e^{-\lambda_c \Delta t j} \tag{5.18}$$

Next, the infinite series can be solved by using the following equation.

$$\sum_{j=1}^{\infty} x^j = \frac{x}{1-x} \ni |x| < 1 \tag{5.19}$$

$$m_{ss} = \alpha(e^{\lambda_c \Delta t} - 1) \frac{\gamma^j e^{-\lambda_c \Delta t j}}{1 - \gamma^j e^{-\lambda_c \Delta t j}} \tag{5.20}$$

Plugging in the variables α and γ yields the full equation for the steady state mass with batchwise reprocessing.

References

[1] (2005). SCALE: A Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluations. vols IIII. ORNL/TM-2005/39, version 5.

[2] 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.

[3] Aufiero, M., Brovchenko, M., Cammi, A., Clifford, I., Geoffroy, O., Heuer, D., Laureau, A., Losa, M., Luzzi, L., Merle-Lucotte, E., Ricotti, M. E., and Rouch, H. (2014a). Calculating the effective delayed neutron fraction in the Molten Salt Fast Reactor: Analytical, deterministic and Monte Carlo approaches. *Annals of Nuclear Energy*, 65:78–90.

[4] 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.

[5] Aufiero, M., Cammi, A., Geoffroy, O., Losa, M., Luzzi, L., Ricotti, M. E., and Rouch, H. (2014b). Development of an OpenFOAM model for the Molten Salt Fast Reactor transient analysis. *Chemical Engineering Science*, 111:390–401.

[6] Betzler, B. (2021). Liquid-fueled Molten Salt Reactor Depletion Modeling. Technical report, Oak Ridge National Lab. (ORNL), Oak Ridge, TN (United States).

[7] Betzler, B., Powers, J., and Brown, N. (2017a). Implementation of Molten Salt Reactor Tools in SCALE.

[8] 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.

[9] Bredimas, A. and Nuttall, W. J. (2008). An international comparison of regulatory organizations and licensing procedures for new nuclear power plants. *Energy Policy*, 36(4):1344–1354.

[10] Brook, B. W., Alonso, A., Meneley, D. A., Misak, J., Blees, T., and van Erp, J. B. (2014). Why nuclear energy is sustainable and has to be part of the energy mix. *Sustainable Materials and Technologies*, 1-2:8–16.

[11] 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.

[12] 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.

[13] Chadwick, M. B., Herman, M., Obložinský, P., Dunn, M. E., Danon, Y., Kahler, A. C., Smith, D. L., Pritychenko, B., Arbanas, G., Arcilla, R., Brewer, R., Brown, D. A., Capote, R., Carlson, A. D., Cho, Y. S., Derrien, H., Guber, K., Hale, G. M., Hoblit, S., Holloway, S., Johnson, T. D., Kawano, T., Kiedrowski, B. C., Kim, H., Kunieda, S., Larson, N. M., Leal, L., Lestone, J. P., Little, R. C., McCutchan, E. A., MacFarlane, R. E., MacInnes, M., Mattoon, C. M., McKnight, R. D., Mughabghab, S. F., Nobre, G. P. A., Palmiotti, G., Palumbo, A., Pigni, M. T., Pronyaev, V. G., Sayer, R. O., Sonzogni, A. A., Summers, N. C., Talou, P., Thompson, I. J., Trkov, A., Vogt, R. L., van der Marck, S. C., Wallner, A., White, M. C., Wiarda, D., and Young, P. G. (2011). 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.

[14] Cui, Y., Chen, J., Wu, J., Zou, C., Cui, L., He, F., and Cai, X. (2022). Development and verification of a three-dimensional spatial dynamics code for molten salt reactors. *Annals of Nuclear Energy*, 171:109040.

[15] Du, Y. and Parsons, J. E. (2009). Update on the Cost of Nuclear Power.

[16] Fei, T., Feng, B., and Heidet, F. (2020). Molten salt reactor core simulation with PROTEUS. *Annals of Nuclear Energy*, 140:107099.

[17] 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.

[18] 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.

[19] Gehin, J. C. and Powers, J. J. (2016). Liquid Fuel Molten Salt Reactors for Thorium Utilization. *Nuclear Technology*, 194(2):152–161.

[20] Goluoglu, S. and McClure, J. (1998). SOFTWARE QUALIFICATION REPORT for MCNP Version 4B2.

[21] 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.

[22] Hombourger, B., Křepel, J., and Pautz, A. (2020). The EQL0D fuel cycle procedure and its application to the transition to equilibrium of selected molten salt reactor designs. *Annals of Nuclear Energy*, 144:107504.

[23] Ignatiev, V., Feynberg, O., Gnidoi, I., Merzlyakov, A., Smirnov, V., Surenkov, A., Tretiakov, I., Zakirov, R., Afonichkin, V., Bovet, A., Subbotin, V., Panov, A., Toropov, A., and Zherebtsov, A. (2007). Progress in Development of Li,Be,Na/F Molten Salt Actinide Recycler & Transmuter Concept. *Proceedings of ICAPP*, page 10.

[24] 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.

[25] Kelly, J. E. (2014). Generation IV International Forum: A decade of progress through international cooperation. *Progress in Nuclear Energy*, 77:240–246.

[26] Kowalski, K. A. (1977). *Scheduling fuel-shuffling operations for a nuclear power reactor*. Thesis, Virginia Polytechnic Institute and State University. Accepted: 2015-07-28T20:42:18Z.

[27] Leppänen, J. (2007). Development of a New Monte Carlo Reactor Physics Code. *VTT Publications*, 640:241.

[28] 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.

[29] Liaoyuan, H., Shaopeng, X., Jingen, C., Guimin, L., Jianhui, W., and Yang, Z. (2021). Th-U Breeding Performances in an Optimized Molten Chloride Salt Fast Reactor. *Nuclear Science and Engineering*, 195(2):185–202. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/00295639.2020.1798728.

[30] 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. page 7, Brussels, Belgium.

[31] Moir, R. W. (2002). Cost of Electricity from Molten Salt Reactors. *Nuclear Technology*, 138(1):93–95. Publisher: Taylor & Francis _eprint: https://doi.org/10.13182/NT02-A3281.

[32] Nagy, K., Kloosterman, J. L., and Lathouwers, D. (2008). Parametric studies on the fuel salt composition in thermal molten salt breeder reactors. *International Conference on the Physics of Reactors*, page 8.

[33] 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.

[34] 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.

[35] 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).

[36] Ridley, G. and Chvala, O. (2017). A method for predicting fuel maintenance in once-through MSRs. *Annals of Nuclear Energy*, 110:265–281.

[37] Robertson, R. (1971). CONCEPTUAL DESIGN STUDY OF A SINGLE-FLUID MOLTEN-SALT BREEDER REACTOR. Technical Report ORNL–4541, 4030941, ORNL.

[38] 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*.

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

[40] Rykhlevskii, A. (2020). *Fuel Processing Simulation Tool for Liquid-fueled Nuclear Reactors*. Doctoral Dissertation, University of Illinois at Urbana-Champaign, Urbana, IL.

[41] 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.

[42] 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.

[43] 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.

[44] Singh, V., Lish, M. R., Chvála, O., and Upadhyaya, B. R. (2017). Dynamics and control of molten-salt breeder reactor. *Nuclear Engineering and Technology*, 49(5):887–895.

[45] 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.

[46] Tyra, B., Cassar, C., Harrison, E., Gorski, A., and Yildiz, O. (2022). Electric Power Monthly with data for August 2022. Technical report, U.S. Electric Information Administration.

[47] Wooten, D. and Powers, J. J. (2018). A Review of Molten Salt Reactor Kinetics Models. *Nuclear Science and Engineering*, 191(3):203–230.

[48] Xia, S., Chen, J., Guo, W., Cui, D., Han, J., Wu, J., and Cai, X. (2019). Development of a Molten Salt Reactor specific depletion code MODEC. *Annals of Nuclear Energy*, 124:88–97.

[49] 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.

[50] 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.

[51] Zou, C., Yu, C., Wu, J., Cai, X., and Chen, J. (2020). Transition to thorium fuel cycle in a small modular molten salt reactor based on a batch reprocessing mode. *Annals of Nuclear Energy*, 138:107163.