INL/EXT-21-64243

Optimization Methodology of Pebble Bed HTGR Start-Up and Running-in Strategy

September 2021

Status Report

Travis L. Lange
Idaho National Laboratory, Department C120

Paolo Balestra
Idaho National Laboratory, Department C130

INL is a U.S. Department of Energy National Laboratory
operated by Battelle Energy Alliance, LLC

INL/EXT-21-64243

Optimization Methodology of Pebble Bed HTGR Start-Up and Running-in Strategy

Status Report

Travis L. Lange
Idaho National Laboratory, Department C120
Paolo Balestra
Idaho National Laboratory, Department C130

September 2021

Idaho National Laboratory
Advanced Reactor Technologies
Idaho Falls, Idaho 83415

http://www.art.inl.gov

Prepared for the
U.S. Department of Energy
Office of Nuclear Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

Page intentionally left blank

ART Program

Optimization Methodology of Pebble Bed HTGR Start-Up and Running-in Strategy

INL/EXT-21-64243

September 2021

Technical Reviewer: (Confirmation of mathematical accuracy, correctness of data, and appropriateness of assumptions).

Ryan Stewart — 9/1/2021

Ryan H. Stewart
Nuclear Engineer — Date

Approved by:

09/01/2021

Michael E. Davenport
ART Project Manager — Date

Gerhard Strydom — 8/31/2021

Gerhard Strydom
ART National Technical Director — Date

Michelle Sharp — 8/31/2021

Michelle T. Sharp
INL Quality Assurance — Date

Page intentionally left blank

SUMMARY

In recent years, interest in advanced reactor technologies has increased significantly. However, the methods used for the analysis of traditional nuclear reactors are insufficient to consider all the different and varied advanced reactor designs without further development. One promising advanced reactor design is the pebble bed reactor (PBR), which possesses unique operational and fuel cycle features requiring the development of specific analysis methodologies to adequately design and analyze the systems. There is a need in PBR research for a capability to analyze and optimize the process of transitioning from the start-up reactor core to the equilibrium core (known as the "running-in" of the reactor). The start-up of a PBR and the transition to the equilibrium core is a complex, multi-physics challenge that has not yet been well researched and has many opportunities for design, analysis, and process optimization. This work defines a methodology to consider the potential strategies in PBR start-up and run-in to the equilibrium core. Multiple candidate software are considered with their pros and cons discussed for the PBR-specific application in the methodology. A preliminary software selection for the physics engine is made and initial verification of key modules is performed. Software to support the optimization of the reactor run-in strategies through reduced order modeling (ROM) and machine learning are considered. Challenges for a full implementation of the methodology are also discussed. Additional code selections and verification of models relevant to the application are needed before full demonstration of the methodology can be achieved and an optimal strategy determined.

ACKNOWLEDGEMENTS

This research made use of the resources of the High-Performance Computing Center at Idaho National Laboratory, which is supported by the Office of Nuclear Energy of the U.S. Department of Energy and the Nuclear Science User Facilities under Contract No. DE-AC07-05ID14517.

The authors would like to thank and acknowledge Dr. Gerhard Strydom for sharing his knowledge in all matters related to high temperature gas cooled reactors (HTGRs) and his guidance on the direction and execution of the research.

CONTENTS

FIGURES

TABLES

ACRONYMS

BAPL	Bettis Atomic Power Laboratory
HALEU	high-assay low-enriched uranium
HEU	high-enriched uranium
HTGR	high temperature gas cooled reactor
HTR	high-temperature reactor
IAEA	International Atomic Energy Agency
INL	Idaho National Laboratory
LANL	Los Alamos National Laboratory
LEU	low-enriched uranium
MCNP	Monte-Carlo N-Particle code
MEDUL	Mehfachdurchlauf (“multi-pass” in German)
OTTO	once-through-then-out
PBR	pebble bed reactor
ROM	reduced order model
TRISO	tri-structural isotropic
VTT	Technical Research Centre of Finland

Page intentionally left blank

ART Pebble Bed Start-Up Methodology

1. INTRODUCTION

Given the recent interest on advanced reactor technologies and in particular on pebble bed reactors (PBRs), development of new dedicated tools for design and safety analysis is needed.

PBRs utilize thousands of tri-structural isotropic (TRISO) fuel particles that are dispersed in a sphere of graphite matrix. These fuel elements are known as pebbles. Hundreds of thousands of these pebbles are randomly packed into the core region of a reactor where they create and sustain a nuclear chain reaction to generate heat. High pressure helium is forced through the interstitial spaces between the fuel pebbles for cooling. One of the unique operational and fuel cycle features of PBRs is that the fuel pebbles slowly flow through the reactor core from the top to the bottom. This allows refueling of the PBR while reactor is operating. The fuel pebbles that exit the reactor could be immediately disposed in a once-through-then-out (OTTO) fuel cycle, or they could also be passed through the reactor again in a multi-pass fuel scheme called MEDUL, which is derived from "mehfachdurchlauf" meaning "multi-pass" in German. These unique reactor characteristics offer benefits such as no reactor downtime for re-fueling and the ability to modify the isotopic composition of the reactor during operation. However, these unique features also bring multiple technical- and process-oriented challenges. These challenges require the development of specific analysis methodologies to adequately design and analyze the systems.

PBR research is needed to analyze to analyze and optimize the process of transitioning from the start-up reactor core to the equilibrium core (known as the "running-in" of the reactor). Research has been performed to design, analyze, and optimize the fuel parameters for the equilibrium core [1] [2] [3] [4]. However, there has been limited work on the start-up core and transition to the equilibrium core at the beginning of the reactor life cycle [5]. The start-up of a PBR and the transition to the equilibrium core is a complex, multi-physics challenge that has many opportunities for design, analysis, and optimization of the process. This research begins to address those many challenges by developing a methodology that is specific to the potential strategies in PBR start-up and run-in to the equilibrium core.

2. PURPOSE AND STATUS OF RESEARCH

The purpose of this research is to develop a methodology that will allow full core simulation and analysis of a pebble bed high temperature gas cooled reactor (HTGR) start-up sequence and run-in with the following modeling capabilities:

- Perform a mass flow through the reactor system
- Perform mixing of user-defined fuel/moderator/absorber pebble ratios during the start-up and run-in sequence
- Perform material burnup and depletion
- Perform an analysis looking at key reactor parameters including power density, power peaking, and fuel pebble burnup at discharge
- Include an optimization software wrapper to drive multiple simulations that can modify select input parameters in search of optimal parameters.

Current development of the methodology will focus on an OTTO fuel cycle for initial demonstration of the methodology. Once the OTTO fuel cycle is demonstrated, the methodology will be expanded to include the MEDUL fuel cycle with decay time between passes. The final methodology will be applied to a HTGR pebble bed start-up sequence to optimize the fuel enrichment, fuel loading, number of pebble types, power density, and power peaking in the reactor.

The research has developed a high-level methodology process that captures requirements for the successful design, analysis, and optimization of a PBR start-up and run-in process. Multiple candidate software to fulfill the methodology requirements have been identified and analyzed. A selection has been made for the physics engine software and initial verification of key physics modules has been performed. The developed methodology has great opportunity for improvement through the application of machine learning and the use of a reduced order model (ROM). The ability to incorporate future work on machine learning and ROM has been considered as well.

There have been significant advancements in the development of a methodology for the design, analysis, and optimization of PBR start-up and run-in strategies. Additional code selections and verification of models relevant to the application are needed before full demonstration of the methodology can be achieved and an optimal strategy is determined.

3. START-UP AND RUN-IN METHODOLOGY

The start-up and run-in of a PBR is a complex transient process taking many months or even years, with many parameters that could be optimized to achieve specific goals. Some of these goals are safety-related, such as remaining below fuel temperature limits, while others are efficiency- and cost-related, such as increase pebble discharge burnup to efficiently use fuel during the run-in and reduce costs.

3.1 Overview of Methodology

The developed methodology has two main considerations: (1) the accurate physical modeling and analysis of a PBR; and (2) the optimization of user-defined parameters in the model to achieve desired reactor characteristics.

The first consideration is related to the selection of a physics engine for the geometry modeling and simulation of physical phenomena in the reactor core. This includes neutron transport and criticality calculations, the flow of material through the core, material depletion, and radioactive decay. The physics engine must also export the important resulting data from the calculations for post-processing and optimization software analysis.

The optimization of the reactor parameters is the second consideration of the methodology and will be automated using the optimization software. This software must have the capability to initiate simulations of the physics engine, interact with the simulation and results, post-process the simulation results, store the results of interest, modify the physics engine input file, and repeat the process through the parameter optimization space.

These considerations and their interactions are depicted in the process flow diagram shown in Figure 1. Here is a breakdown of the flow process and its interactions:

1. The initial PBR model is generated, which includes geometry, materials, and enabled physics models. This model is then captured in an input file that is specific to the physics engine.
2. The initial model input file is sent to the optimization software, which is given information regarding the parameters in the input file to modify and over what range those modifications need to be completed. The optimization software then can be used to drive many tens, hundreds, or even thousands of physics simulations, which obviously can be very computationally expensive and represents a potential challenge in the methodology.
3. The physics engine performs a reactor simulation calculating the power density, mass flow through the reactor, and material depletion for the entire start-up and run-in process.
4. Once the equilibrium core has been reached, the physics simulation exports the data to the output files and the physics engine simulation terminates.
5. The output files from the physics simulation are post-processed by the optimization software to collate the results for the final output from the optimization software.
6. If the last simulation in the optimization space has been performed, the optimization software terminates. If not, the optimization software modifies the input and initiates the cycle again. This is the point in the process that a machine learning algorithm may be incorporated to process the results from the physics simulation and calculate the next set of reactor parameters to be analyzed in the optimization.

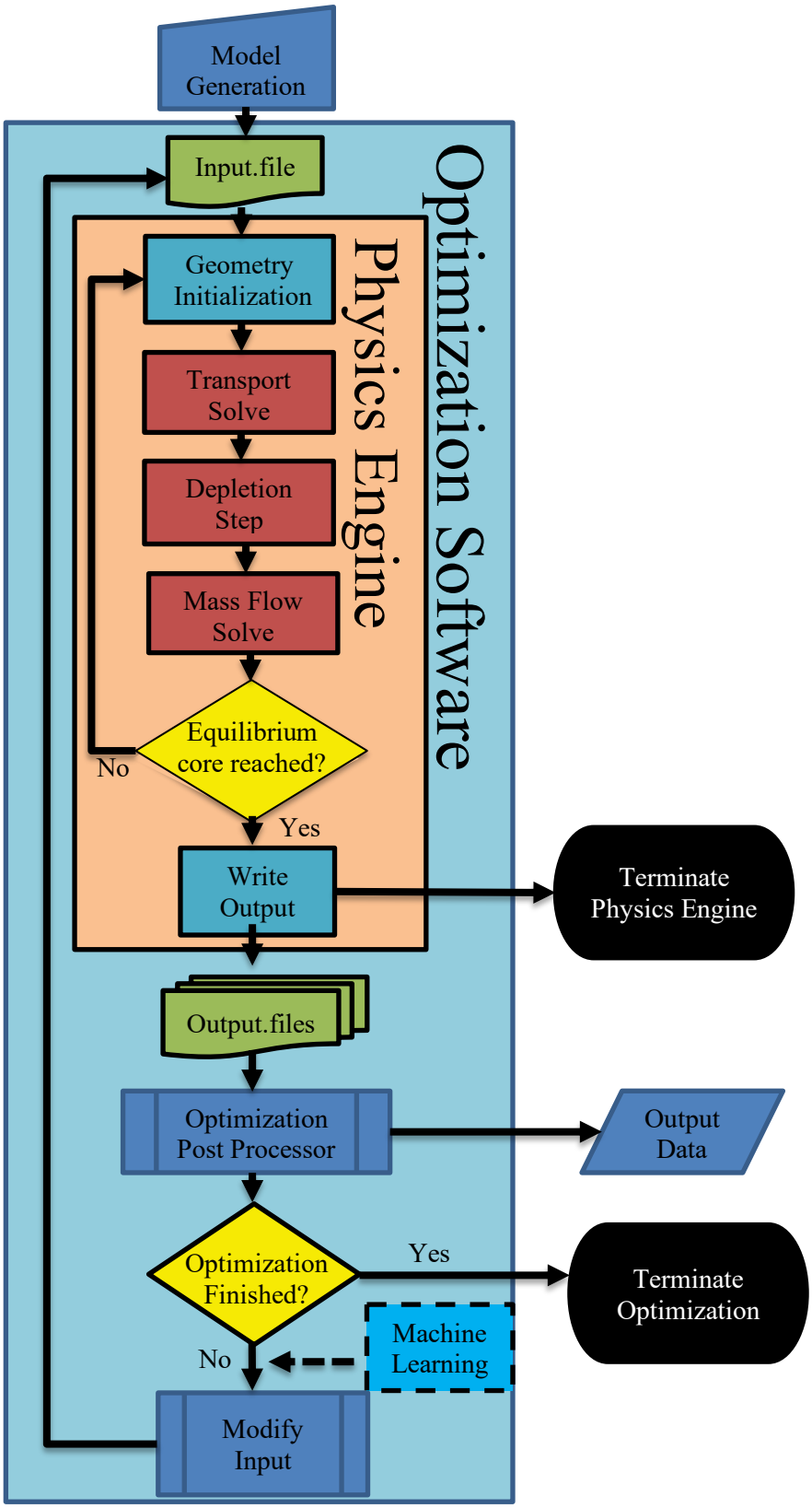

Figure 1. High-level optimization methodology process flow.

These desired characteristics of the methodology will require the use of at least two different sets of software. The physics engine preferably will model all key physical phenomena for a PBR internally without the development of new physics models. The optimization software will ideally have multiple optimization methods from which to choose and the potential for including feedback from machine learning and the use of ROMs in the method.

3.1.1 Parameter Optimization in the Methodology

Several optimization parameters are available for this complex multiphysics scenario, and they all have to be selected taking into account the different constrains (e.g. fuel temperature, peaking factors) and optimization goals (e.g. fuel utilization, equivalent full power day per year). Some of the most common choices are listed below:

- Power Density – Power density will ultimately determine the temperature of the fuel kernel and the fuel burnup in the TRISO particle and the pebble. Fuel temperature limits must be maintained, preferably with a maximum margin.
- Initial Uranium Mass Loading – The uranium in the fuel is critical to maintaining reactor power. It is also one of the most expensive parts of the fuel elements. Minimizing the initial mass of uranium loading for reactor operation increases fuel efficiency and reduces costs.
- Fuel U-235 Enrichment – Higher U-235 enrichment in the fuel increases reactivity in the core and can reduce the total amount of fuel required for reactor operation. However, it will also increase localized power density and has a greater potential for high-power peaking factors. There are also regulatory factors in using low-enriched uranium (LEU), high-assay low-enriched uranium (HALEU), or high-enriched uranium (HEU). HEU will not be considered in this methodology due to regulatory concerns with its use and proliferation.
- Number and Type of Pebbles – Due to the unique fuel loading of discrete pebbles, PBR fuel can take advantage of multiple pebble types with different fuel loadings and enrichments, absorber material loadings, or a simple carbon sphere for the initial core configuaration. All these pebble types can be mixed in different proportions to optimize the specific characteristics of their core behavior. This provides a great degree of flexibility in PBR start-up and run-in strategies, but also increases the complexity of the reactor operation and optimization process.
- Pebble Flow Rate – This parameter determines the resident time of each pebble in the reactor. This affects fuel burnup at discharge (and therefore fuel efficiency), fission product buildup, and core reactivity.

These parameters are either user-defined or calculated using the physics engine and the PBR model. There are trade-offs to consider with the selection of each of these parameters depending on the importance of each category and which category is prioritized. These parameters will be used for optimization of the reactor loading, start-up, and run-in strategy using the developed methodology.

4. PHYSICS ENGINE SOFTWARE FOR METHODOLOGY

The physics engine capabilities are central to the design and analysis of the start-up and run-in strategies of the PBR model. The design and analysis of the start-up and run-in strategies will be limited in scope and application according to the abilities of the physics models in the physics engine. Therefore, significant effort has been put into researching the capabilities of the physics engine software available to us. This section details the candidate software and their capabilities relevant to this application.

The primary physics capability in the development of a pebble bed start-up and run-in strategy is the neutron transport solver. Two Monte-Carlo type codes were considered for this application: Monte-Carlo N-Particle (MCNP) and Serpent. For this research, the following capabilities of each software will be analyzed for applicability to PBR start-up and run-in simulations:

- Particle/pebble bed geometry modeling
- Simulating mass flow through geometry
- Depletion/burnup capability
- Multi-physics coupling capabilities
- Simulation restart capabilities
- Data export for analysis.

The depletion, mass flow, and data export capabilities are integral to a meaningful PBR start-up and run-in simulation and analysis. The full capabilities of particle/pebble bed geometry modeling, multi-physics coupling, and restart capabilities are not strictly necessary as approximations or workarounds in the model could be made to manage their absence. However, a full set of the capabilities mentioned above will make the application of the developed methodology more accurate, efficient, and robust to unforeseen challenges.

A detailed description and discussion of the relevant physics models and capabilities for PBR modeling for both MCNP and Serpent is given below.

4.1 MCNP for HTGR PBR Modeling

The MCNP code is a general-purpose transport code for neutrons, photons, electrons, or heavy ions. It is developed and maintained by Los Alamos National Laboratory (LANL) [6]. MCNP is widely used in the nuclear industry for nuclear criticality and safety, radiation protection and dosimetry, and reactor design and analysis, among many other applications. The version used for this research is the latest release, MCNP 6.2.

4.1.1 Particle/Pebble Bed Geometry Modeling

MCNP has been used in previous research for approximating pebble bed geometries [7]. It uses multiple instances of the lattice feature to create a repeated TRISO particle lattice inside a fuel pebble, as well as a lattice of individual fuel pebbles in a core region. These lattices can be defined as infinite or finite. Infinite lattices are less complex to model, which leads to fewer geometry errors in the model. The infinite lattice is inserted into a bounding geometry that it fills. One side-effect of filling a general geometry with an infinite lattice is that the boundary of that geometry “clips” the lattice at the boundary. This lattice clipping creates partial fuel pebbles and TRISO particles at the core boundary, which is not an accurate physical model of a PBR core. A geometry filled with an infinite lattice is shown on the left side of Figure 2.

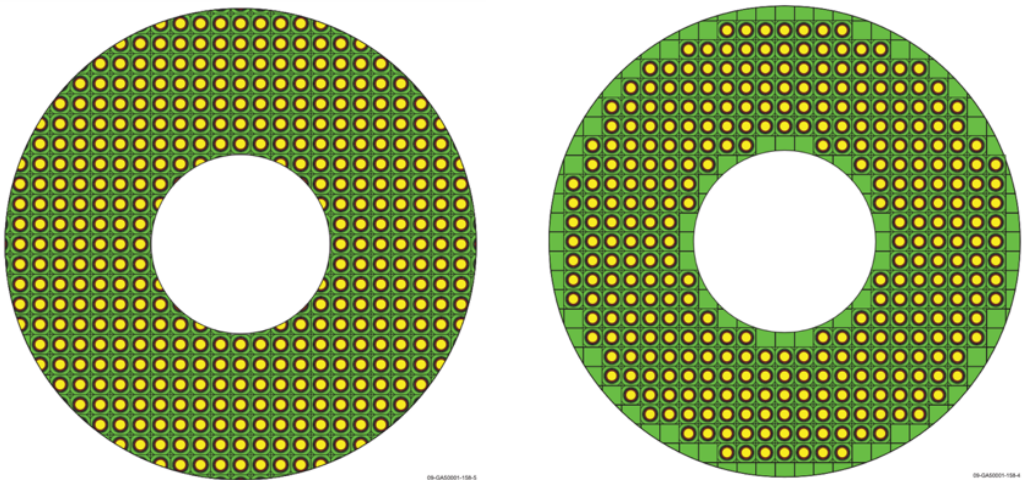

Figure 2. Different modeling approaches using MCNP to model a pebble bed: the infinite lattice inside a bounding universe that clips the fuel pebbles (left); and the finite lattice of pebbles that selectively avoids clipped pebbles at the core boundary (right) [7].

Using a finite lattice, the pebble and particle clipping can be avoided. However, the set-up of large finite lattices is more complicated and can lead to geometry errors. There is also concern that the empty lattice spaces created at the edge of the reactor core reduces the effective packing fraction of fuel in the core and can lead to inaccurate results for reactivity and power profiles. A depiction of a finite lattice in a core geometry is shown on the right side of Figure 2.

The same challenges of using an infinite or finite lattice also exists for TRISO particle distribution modeling inside a fuel pebble.

4.1.2 Simulating Mass Flow Through Geometry

Currently, MCNP does not have any inherent capability to model the flow of mass through the geometry in the model. The only two reactor concepts that need this capability are the molten salt-fueled reactor and PBRs. Most all other reactor types do not require the movement of fuel material and fission products through the system. The flow of material through the geometry could be approximated by using a restart file for each depletion step that reads in the isotropics of the material in the volume upstream of the material flow. However, computationally, this would be very inefficient as a lot of time would be spent unnecessarily initializing the same geometry for every depletion step.

4.1.3 Depletion/Burnup Capability

MCNP 6.2 does have a “burn” function that allows it to calculate material burnup and depletion. The depletion capability uses CINDER90 [8], which is a nuclear inventory code that originated at Bettis Atomic Power Laboratory (BAPL). Using CINDER90, MCNP6 has been used in a published depletion study for the high-temperature reactor (HTR)-10 PBR design [9].

4.1.4 Multi-Physics Coupling Capabilities

MCNP does not currently have a native way to couple it to other external physics modules in the standard release. However, MCNP has been used as a coupled code in multi-physics simulations in multiple studies. MCNP has been used in multi-physics simulations through internal and external coupling. The external coupling of MCNP to a thermal-hydraulic code is the most common type of coupling performed with MCNP [10] [11] [12] [13]. External coupling to MCNP uses input and output files from both codes to facilitate data transfer during simulation iterations. Additionally, MCNP has been internally coupled to a subchannel thermal-hydraulic code called “CTF” [14]. Internal coupling is

preferred because the data transfer for solution convergence is interior to both codes, which allows for faster simulation convergence because there is no need to re-initialize the simulation in every iteration.

Multi-physics coupling for a PBR applications would be possible, but it would require significant effort through the development of an external driver program to perform data extraction, processing, and transfer for every iteration between the coupled codes.

4.1.5 Restart Capability

MCNP has a native ability to restart a simulation from a special restart file written by MCNP during a previous simulation. MCNP's restart capability is referred to as a "continue-run." MCNP uses a binary restart file called the "RUNTPE" file, which contains all the geometry, cross-sections, and other parameters necessary to restart a simulation. The continue-run can also take an optional input file that can be used to modify certain parameters for the restarted simulation.

4.1.6 Data Export for Analysis

The extraction and export of data to check key parameters such as power density and power peaking for analyzing the PBR simulation is required. MCNP is fully capable of this type of data extraction and export for all the parameters necessary for PBR analysis. MCNP has cell tallies, surface tallies, and mesh tallies to calculate and export this data.

4.2 Serpent for HTGR PBR Modeling

Serpent is a continuous energy MCNP transport code. Serpent has been in development since 2004. Serpent is used for reactor design and analysis, fuel cycle analysis, criticality studies, and multi-physics simulations, among other applications [15].

4.2.1 Particle/Pebble Bed Geometry Modeling

Serpent has previously been used for pebble bed/TRISO fuel modeling and reactor design and analysis [16]. Of particular interest to pebble bed HTGR reactors, Serpent includes a native explicit particle/pebble bed fuel model developed for fuel pebbles with TRISO fuel particles, which allows the user to employ scripts native to Serpent to generate three-dimensional (3D) geometries of fuel pebbles within an HTGR core and the locations of TRISO particles within a fuel pebble. These capabilities allow for randomly distributed spheres of a given packing fraction to be modeled explicitly in Serpent with no particle clipping at the cell boundaries. A plot of a geometry created in Serpent using this method is shown in Figure 3. This is an improvement over MCNP and reduces the need for additional validation of the effects of the geometry on the physics.

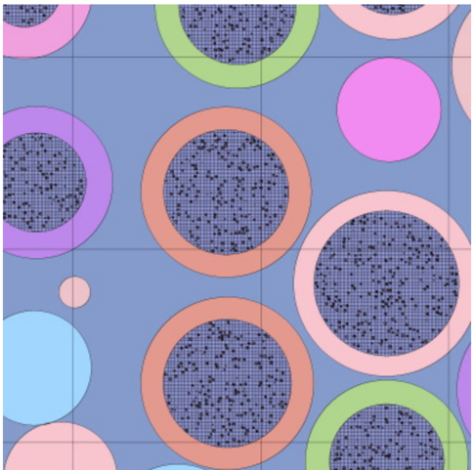

Figure 3. Cross-section of randomly packed fuel pebbles with randomly packed TRISO particles on the pebble interior [16].

4.2.2 Simulating Mass Flow During Simulation

Serpent has been developing a native capability for mass flow in a reactor model. It should be noted that it is included in the standard Serpent release version, which is 2.1.32, but this capability is technically still under development, verification, and validation. There is some information on this capability in the wiki for the standard Serpent input syntax [17], but it is limited. The capability uses multiple input cards in conjunction with each other to define the rate of flow, the direction of flow between the materials, and other required information for flow simulation. The primary input card initiating mass flow in the model is called "Mflow." This Mflow capability has been used in a few research studies for molten salt fuel reactors [18], as well as been presented in multiple International Serpent Users Group Meetings [19] [20]. A significant amount of effort was spent in this research understanding the capability, its uses, and its limitations. Section 0 is dedicated to the verification test suite developed for the Mflow capability in this research. Additionally, the inputs for the Mflow capability, their use, and their inter-relationships have been documented in Appendix A.

The Mflow capability in Serpent appears to be working as expected and will be an effective way to model a key physical phenomenon in a PBR start-up and run-in process.

4.2.3 Multi-Physics Coupling Capabilities

Serpent has a native ability to interact with the simulation after neutron transport and depletion have been solved before the next step. This was developed for multi-physics coupling with thermal-hydraulic codes using subchannel or computation fluid dynamics calculations. The multi-physics coupling capability is called by using the "ifc" card in the standard Serpent input [21] [22]. The "ifc" card points to a multi-physics interface file that can be modified by a thermal-hydraulic solver for every solution iteration. The multi-physics interface file can send and receive data from Serpent to control the material densities and temperatures during the multi-physics simulation. The multi-physics capability in Serpent also can be used in a PBR simulation to provide temperature and density feedback to the start-up and run-in process if coupled to a heat transfer solver.

4.2.4 Depletion/Burnup Capability

The depletion capability in Serpent was developed early in the software's life and is fully internal without any coupling to external solvers. Serpent provides for user-defined depletion zones throughout the model that do not have to align with the physical geometry. The depletion capability in Serpent has been compared to multiple codes in other studies [23] [24].

4.2.5 Restart Capability

The restart feature in Serpent is similar to the one in MCNP. A dedicated binary restart file is written by Serpent at a user's request. This restart file can then be used in subsequent simulations to restart the simulation from a chosen depletion step that is read from the restart file. The restart feature in Serpent can be used for the movement of materials in processes like fuel shuffling. This capability is useful in the simulation of a PBR start-up and run-in when used in combination with the Mflow capability by simulating multiple changes in feed materials for the core. This is a capability that MCNP 6.2 does not have available to the analyst. Serpent's capability is demonstrated in the verification tests and results presented in the sections that follow.

4.2.6 Data Extraction for Analysis

Serpent has been used extensively for reactor analysis. To support the extraction of data from the simulation for analysis, Serpent provides the common data of interest in analysis in the standard output files [25]. Serpent also has an ability equivalent to MCNP's tallies available to the analyst using "detectors" [17]. These allow the analyst to select specific data from specific regions of the core to be tallied and printed at the end of the simulation. This is useful for calculating spatially dependent power densities for power peaking calculations.

4.3 Discussion and Decision

Both MCNP or Serpent possess depletion, restart, and data export capabilities that would be sufficient for simulating and analyzing the PBR start-up and run-in process. However, only Serpent provides a native simulation of mass flow through the geometry, explicit modeling of TRISO particles and pebble bed geometry, and the ability for multi-physics interfacing. Due to these advantages, Serpent is being used for this research.

Initial verification of Serpent models for these PBR-specific phenomena has been performed and is documented in Section 0.

5. PROCESS OPTIMIZATION SOFTWARE FOR METHODOLOGY

The selection of Serpent for the physics engine in the methodology allows for a simulation of a PBR model that accurately captures many of the key physical phenomenon in a start-up and run-in process. However, the optimization itself is still a challenge, since the number of degree of freedom is quite large and the correlation between the goal functions and the optimization parameters is strongly non-linear.

Fortunately, software exists that is designed to manage the optimization efforts and that can be used in conjunction with the selected physics engine, which act as drivers for the physics engine by running multiple simulations with perturbations on key parameters and collecting the output data for analysis of the options. The three existing optimization software tools that are being considered for use in the developed methodology for finding an optimized PBR start-up and run-in strategy are: (1) MOOSE Stochastic Tools; (2) RAVEN; and (3) Dakota.

5.1.1 MOOSE Stochastic Tools

The MOOSE Stochastic Tools [26] module is a toolbox designed for parameter studies, statistics, and uncertainty analysis. MOOSE Stochastic Tools was developed at Idaho National Laboratory (INL) and comes as a standard part of the MOOSE framework. The MOOSE Stochastic Tools module also allows for the use of "surrogate models," also known as ROMs.

5.1.2 RAVEN

RAVEN is a software that is also developed at INL for uncertainty quantification, regression analysis, data analysis, and as an optimization framework [27]. RAVEN can be used in conjunction with a ROM in the optimization framework as well.

5.1.3 Dakota

Dakota is a general purpose toolkit developed at LANL that is used for uncertainty quantification, sensitivity analysis, and design optimization [28]. It is developed with a focus on utilizing high-performance computing in its process management. Dakota has been applied to the research done by the Consortium for the Advanced Simulation of Light-Water Reactors [29]. In addition, Dakota has been used in conjunction with ROMs in previous research [30].

Further investigation into the capabilities offered by each optimization software tool is needed prior to their possible selection for use in this PBR start-up and run-in strategies optimization methodology.

6. VERIFICATION TEST SUITE: DEFINITIONS AND RESULTS

A suite of tests was developed to perform verification on the physics models available in Serpent that will be used in the simulation of the PBR start-up and run-in process. These verification tests focus on the use of the models through the standard Serpent input and calculated results in each test. These tests are part of the initial verification of key physics modules in Serpent. However, more verification is needed before the demonstration of a PBR start-up and run-in process.

6.1 Common Model Description

6.1.1 Overview

The material parameters for the equilibrium fuel pebble element are taken from the International Atomic Energy Agency (IAEA) technical document describing the Chinese PBR HTR-10 [31]. The start-up fuel element parameters are modified to distinguish the flow of materials through the model and are not based on any historical use of these fuel element compositions. The material parameters used in the common model are given in Table 1.

Table 1. Summary of material parameters used in the common model.

Materials in the Common Model	
Start-Up Fuel Element	
Diameter of ball	6.0 cm
Diameter of fuel zone	5.0 cm
Density of graphite in matrix and outer shell	1.73 g/cm^3
Heavy metal (uranium) loading (weight) per ball	5.0 g
Enrichment of U-235 (weight)	10%
Volumetric filling fraction of balls in the core	0.61
Equilibrium Fuel Element	
Diameter of ball	6.0 cm
Diameter of fuel zone	5.0 cm
Density of graphite in matrix and outer shell	1.73 g/cm^3
Heavy metal (uranium) loading (weight) per ball	5.0 g
Enrichment of U-235 (weight)	17%
Volumetric filling fraction of balls in the core	0.61
TRISO Fuel Kernel	
Radius of the kernel	250 μm
UO2 density	10.4 g/cm^3
Absorber Element	
Diameter of ball	6.0 cm
Diameter of fuel zone	5.0 cm
Density of graphite in matrix and outer shell	1.73 g/cm^3
B4C loading per ball	2.9 g
Enrichment of B-10	19.9%
Volumetric fraction of B4C in pebble	0.01

The common model was designed for a 1-zone representation of a PBR core and a 3-zone representation. The model consists of a feed material volume that contains the material properties of the material that is flowing into the first material zone. Both common models also have a disposal material volume in which the material from the last zone in the reactor flows into the disposal volume. The 1-zone representation of the common model and the 3-zone representation are shown in Figure 4 and Figure 5, respectively. The volume for all materials is the same at 1e7 cm^3. This eliminates any changes in material densities due to the discrepancies in volumes.

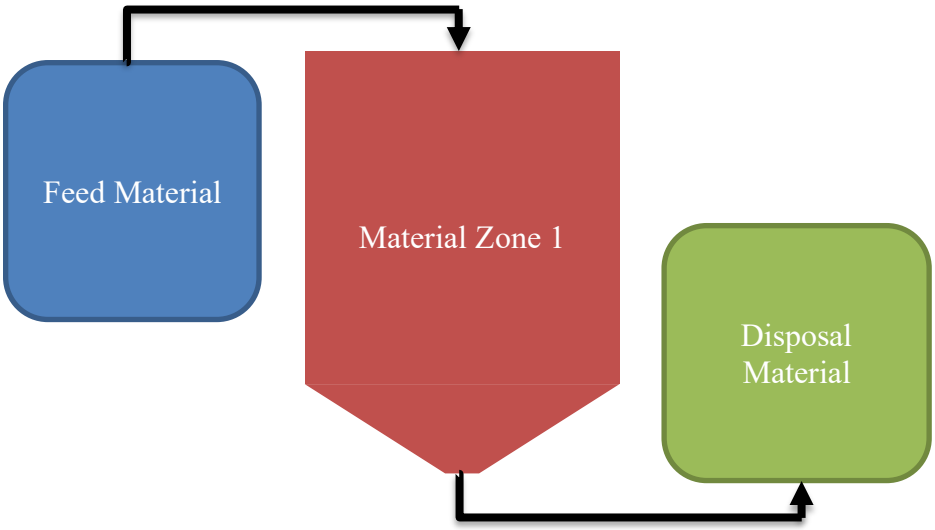

Figure 4. Geometry of basic common model with single axial core material zones.

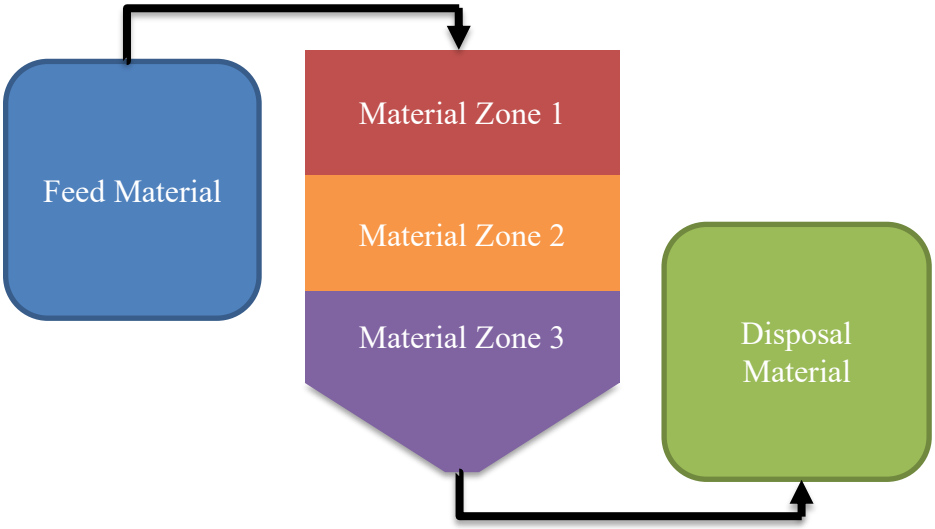

Figure 5. Geometry of common model with three axial core material zones.

The materials in each volume are representative of the fuel elements of different enrichments that have been homogenized. The homogenization decreases the computational requirements of the simulations and does not change the validity of the results and conclusions drawn in the verification tests. Future work will include PBR-specific geometry.

6.1.2 Use of the Mflow Capability in the Common Model

The Mflow capability in Serpent flows mass through the system by connecting material definitions. This implementation of the mass flow capability is not strictly physical since mass does not flow through volumes explicitly. The mass flows through material definitions inside those volumes. Care must be taken to ensure that the mass flow pathway is defined correctly, and that the volumes defined in the material card do not cause changes to mass densities through the user-define flow path. In the material flow path for the 1-zone model, the material flows from the feed material to the material in zone 1 and then to the disposal material. In the material flow path for the 3-zone model, the material flow from the feed material to material zone 1, then to material zone 2, then to material zone 3, and finally to the disposal material.

The Mflow capability in Serpent is directly tied to the depletion module. Both share the same timeline. The depletion time-step is used as the mass flow time-step and cannot be modified separately. This dependence of the Mflow capability in Serpent to the depletion capability requires a few additional inputs related to depletion, but not mass flow. There must be some amount of fissile material defined in the system for Serpent to have a valid neutron transport solution. If there is no fissile material, Serpent will terminate the simulation because a depletion could not successfully be completed. The power of the system must also be defined for Serpent to be able to normalize the reaction rates calculated in the neutron transport and depletion solves. For the verification tests presented in this research, the power of the system was defined at a very low number: 1e-10 Watts. This was to decouple the mass flow simulation from any neutronic effects on the materials. This effort appears to be successful as the verification tests were completed simulations without unexpected effects on materials that are highly sensitive to neutrons, such as Xe-135 and U-235.

6.2 Mass Flow Capability Test 1: Mass Flow for Stable Isotopes

The purpose of this verification test is to examine the different effects of the Mflow options in Serpent on the masses in each material zone. This test uses stable isotopes to isolate the flow of stable materials from those that undergo radioactive decay during the mass flow.

6.2.1 Mflow Option 0 using the 1-zone model

This flow option treats the source material as infinite. It does not reduce the source material over time, which can be used to approximate a very large volume feeding a small volume where mass is effectively unchanged in the large volume. It also can be used to model a source of new fuel material introduced to the reactor over time that has constant material characteristics, such as a single type of fuel pebble being added to the reactor for a given amount of time. This has the effect of adding material to the system. Care must be taken to ensure that non-physical amounts of material build up in any given zone. An example of mass being added to the system is shown using the flow of U-238 through the 1-zone common model. The material in zone 1 and the disposal material definitions did not have any U-238 in them at the beginning of the simulation. The density of U-238 in zone 1 linearly increase in time. This behavior is expected since the feed material U-238 density and the mass flow are constant [19] [20]. On the other hand the U-238 density in the disposal region increase quadratically because of the linear change of the source density in zone 1. This behavior is shown in Figure 6.

Multiple uses in sequence of the Mflow option being 0 should not be used because mass is not preserved in the system.

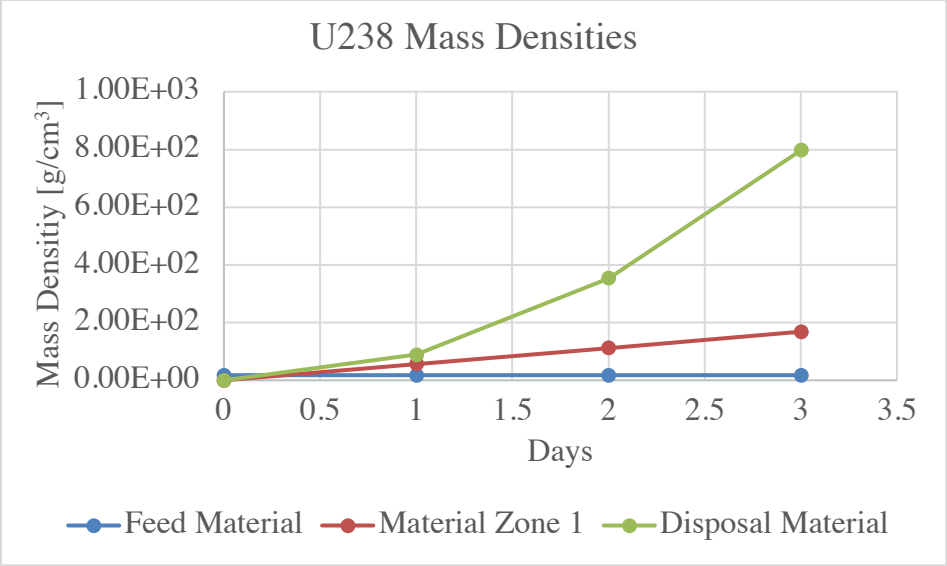

Figure 6. Mass flow characteristics when using Mflow option 0 for all material zones.

6.2.2 Combination of Mflow Options 0 and 1 using the 3-zones model

The use of Mflow option 0 for the feed material and option 1 for all other materials was tested. The resulting flow behavior corresponds well with the physical flow in a reactor system. The resulting mass densities of U-235, U-238, and carbon, are shown in Figure 7, Figure 8, and Figure 9, respectively.

The simulation is mass flow for five years. For both the U-235 and U-238, the mass densities in material zones 1-3 start below the feed material densities at the beginning of the simulation. Then, the U-235 and U-238 mass densities increase slowly over time until they asymptotically approach the feed material mass densities. Material zone 1 mass densities approach the feed material mass density at an accelerated rate than zones 2 or 3, as expected. This shows the mass from the feed material filling material zone 1 first, then zone 2, and finally zone 3.

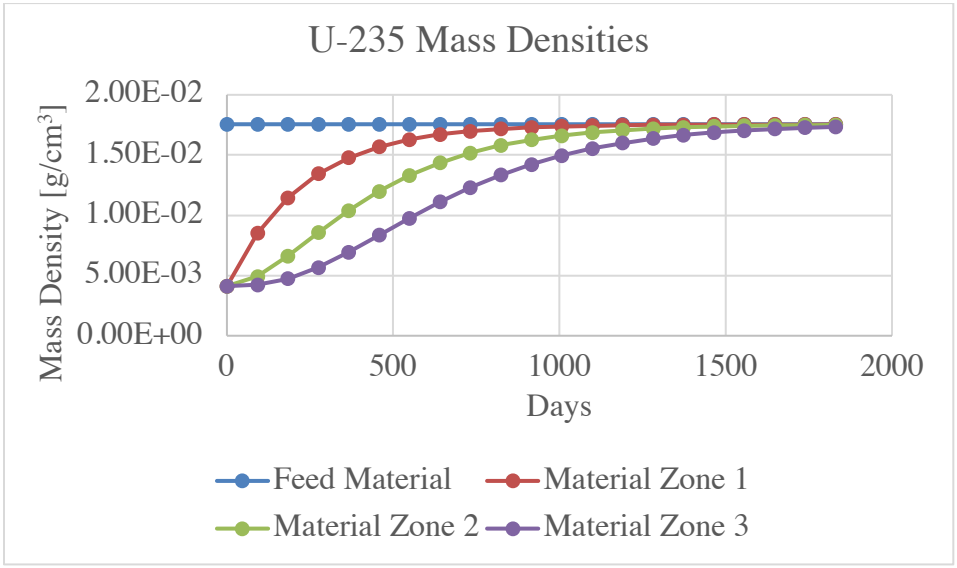

Figure 7. U-235 mass densities for three material zones over a five-year depletion.

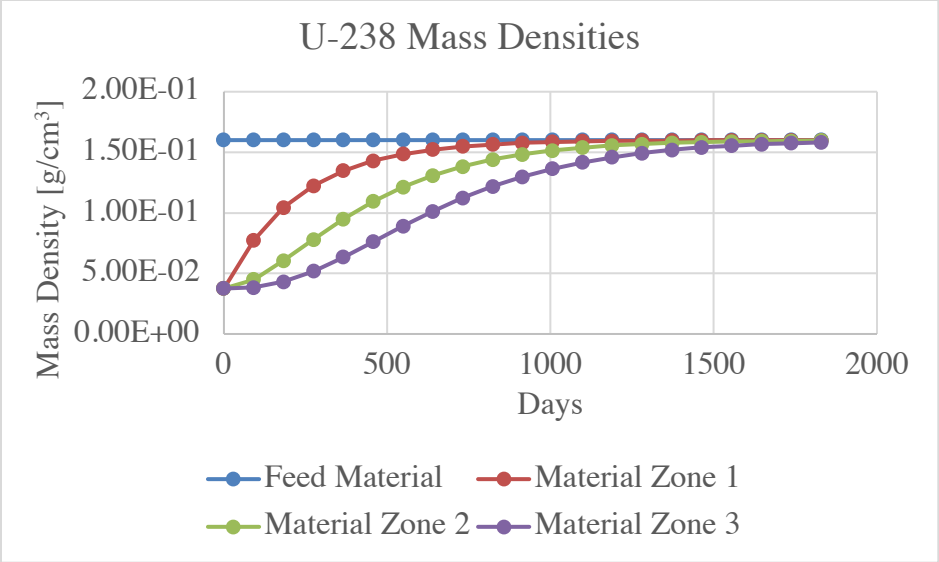

Figure 8. U-238 mass densities for three material zones over a five-year depletion.

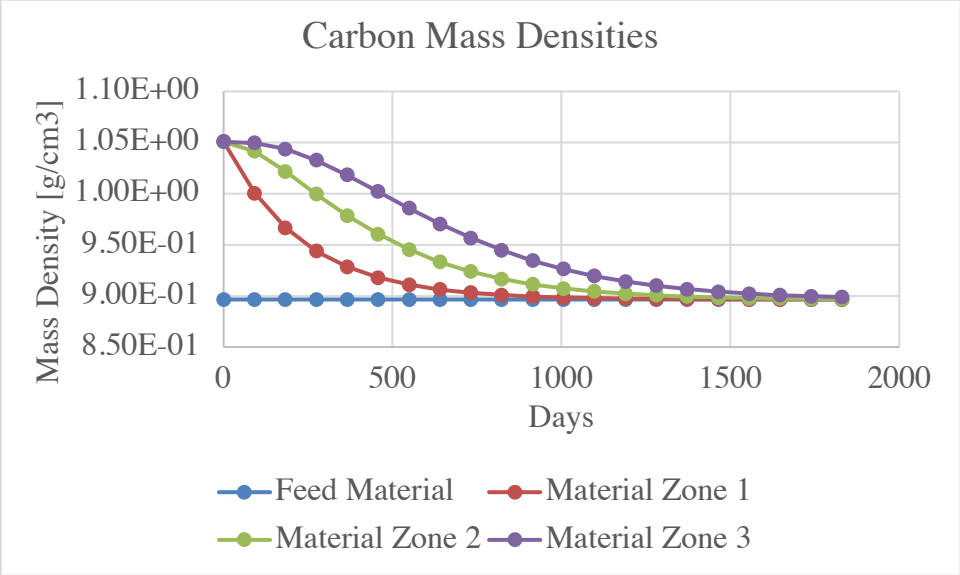

Figure 9. Carbon mass densities for three material zones over a five-year depletion.

The flow of carbon in all material zones over the five-year simulation is shown in Figure 9. The behavior of the carbon mass density is inverted compared to the uranium mass densities. This is because the carbon density in the feed material has a lower mass density of carbon than the other material zones at the beginning of the simulation. The carbon mass decreases in every material zone beginning with zone 1 at the fastest rate, followed by the material in zone 2, and then zone 3. This behavior is expected and consistent with the uranium mass flow results.

6.3 Mass Flow Capability Test 2: Mass Flow in Simplified Geometry with Radioactive Decay using the 3-zones model

The burnup and radioactive decay models in Serpent have been well tested, and has also been used in other research [32] [33]. There is little concern of these standalone capabilities. However, there are not many examples of the burnup and radioactive decay models being utilized and analyzed in conjunction with the mass flow capabilities in Serpent. This verification test is intended to test the behavior of radioisotopes while using Mflow but setting the mass flow rate to 0.

This simulation tracks Xe-135 as the radioisotope. To simplify analysis, the simulation time-step is 9.14 hours, which is the half-life of Xe-135. Each material zone is initialized with a different mass density of Xe-135, and the xenon is tracked over time in the simulation. The results are shown in Figure 10.

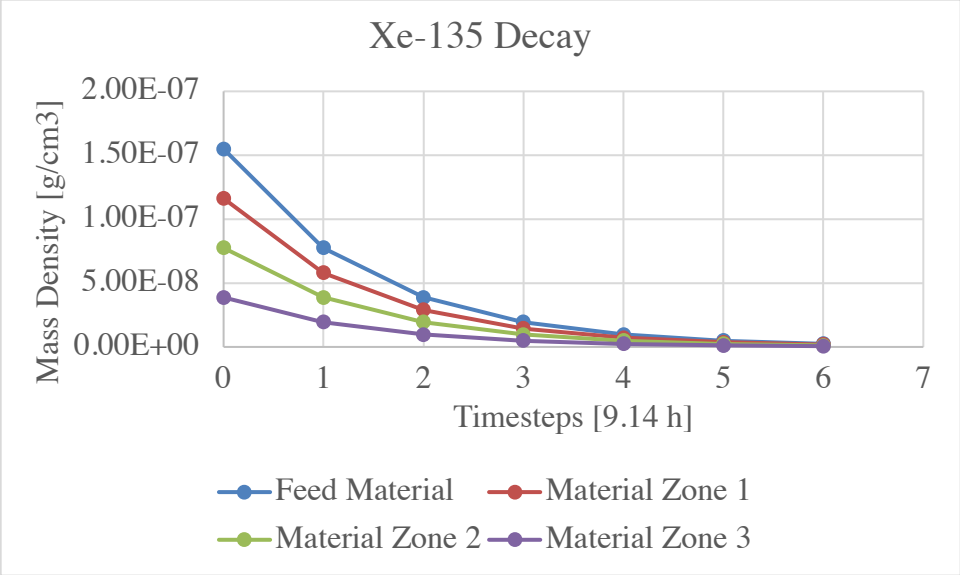

Figure 10. Decay of radioactive isotope Xe-135 in simplified geometry using Mflow option with mass flow rate equal to zero.

The Xe-135 in each material zone decays by half each time-step, which is expected. It is important to note that the Xe-135 in the feed material is also decreasing with time. The 0 option for Mflow treats the feed material isotropics as a constant for stable isotopes. However, the radioactive decay of radioisotopes is still applied to the feed material. This must be understood and applied properly to MEDUL fuel cycles to achieve the desired modeling.

6.4 Mass Flow Capability Test 3: Mass Flow with Restart using the 3-zones model

This simulation is intended to test the combined use of the Mflow capability with the restart capability in Serpent. The analyst could use these two capabilities to change flow rates or feed material properties in the middle of a start-up and run-in process and still accurately model these changes.

In this simulation, the feed material begins the simulation with material properties for the start-up fuel element shown in Table 1. After a five-year simulation, the simulation is terminated and a restart file is written. The model input then is modified so that the feed material properties are now those of the equilibrium fuel elements, as shown in Table 1. The mass densities for U-235, U-238, and carbon, are captured over time and are shown in Figure 11, Figure 12, and Figure 13, respectively.

The results for the initial simulation and restarted simulation show behaviors that are similar to the results in Section 6.2.2. The material properties for each material zone asymptotically approach the material properties of the feed material.

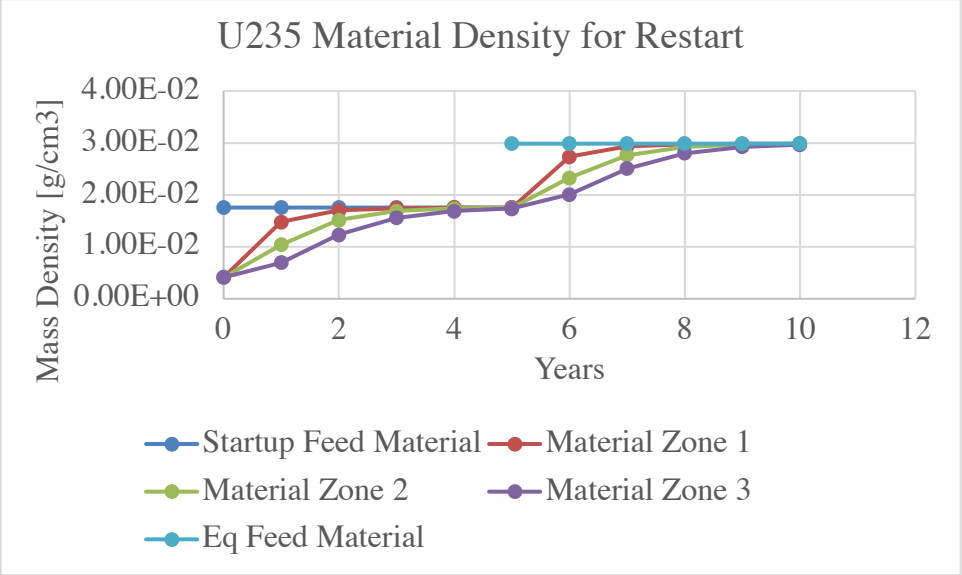

Figure 11. U-235 mass density in three material zones with feed material change and simulation restart at six years.

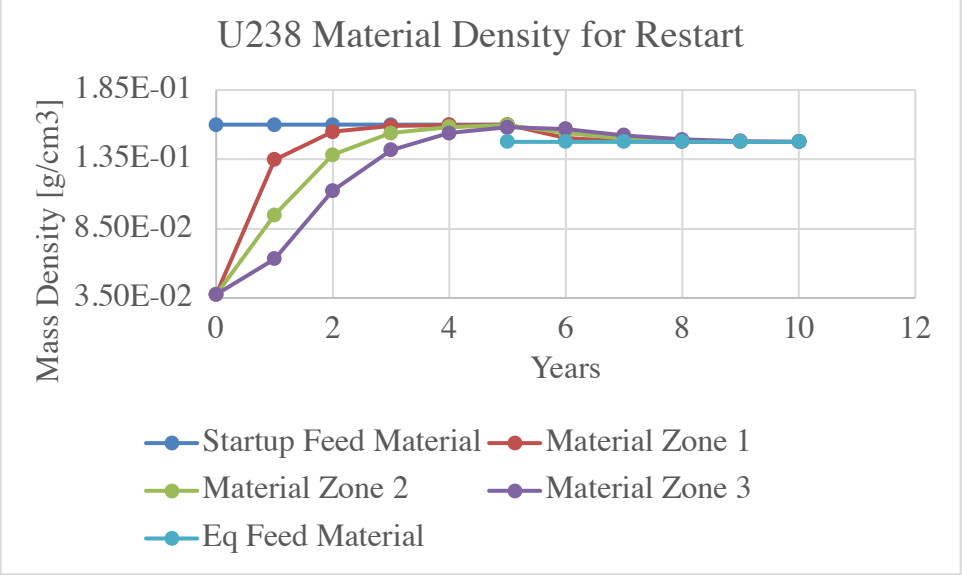

Figure 12. U-238 mass density in three material zones with feed material change and simulation restart at six years.

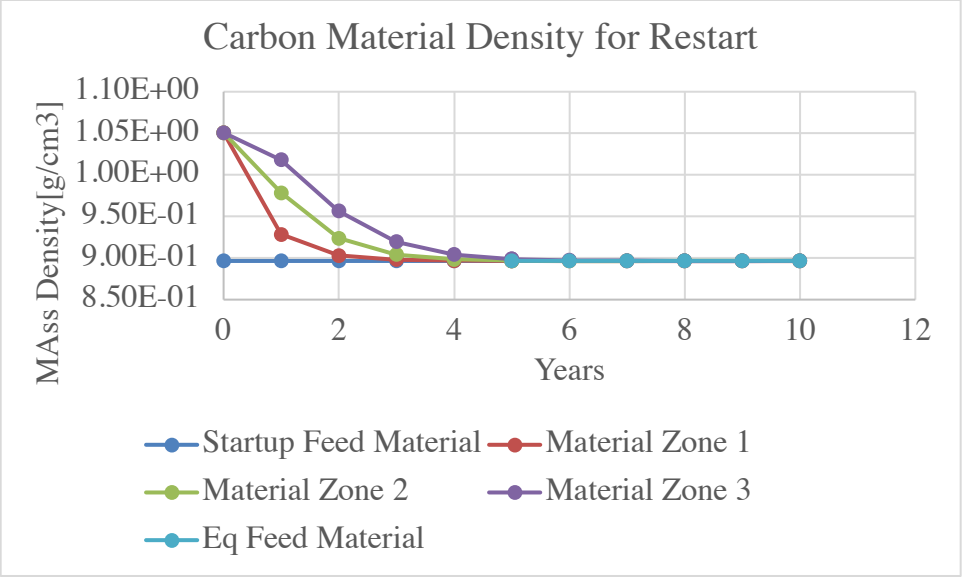

Figure 13. Carbon mass density in three material zones with feed material change and simulation restart at six years.

The simulations to test the restart capability show that Serpent can accurately simulate a change in feed material properties and have those materials propagate through the system after the change. This ability allows the analyst to model and optimize changes to the feed material over the start-up and run-in process.

6.5 Verification Tests Summary

Multiple tests were defined and performed for the initial verification efforts of key capabilities in Serpent for PBR modeling. The successful use of the models, despite some being under development shows promise for their use in the application of optimizing a PBR start-up and run-in strategy. Additional work is required to further verify aspects of the models and to determine the best settings to accurately simulate mass flow through a PBR.

7. FUTURE WORK AND EXPLORATION TASKS

Significant progress has been made in the development of a methodology to design, analyze, and optimize PBR start-up and run-in strategies. However, there are many tasks and additional work that must be performed before a demonstration of the methodology can be achieved. The task list for future work is as follows:

- Additional verification of Serpent capabilities/modules for PBR applications
- Use of Mflow with mixtures
- Use of a multi-physics interface file
- Develop a sub-routine for managing multi-physics interface files between depletion steps
- Verify the effects of a multi-physics interface file on stable, radioactive, and mixture materials
- Select optimization software
- Investigate the application of ROMs in the optimization strategy
- Investigate the use of machine learning compatible with the optimization software
- Develop a representative high-fidelity PBR model with discrete pebbles and TRISO particles
- Test and select the optimization methods available in optimization software for start-up and run-in strategies
- Full demonstration of optimization process of start-up and run-in strategies on a high-fidelity PBR model.

7.1 Challenges in Methodology Implementation

Throughout this research, challenges have manifested in the selection and use of software adequate for the developed methodology.

Key Serpent capabilities are under development and have not been fully verified. There are several software limitations discussed above that must be considered to generate an accurate model and simulation of the start-up and run-in process.

Keeping computational requirements manageable will remain a challenge. There are multiple pathways to address this challenge. Detail of the model, optimization methods, and complexity of start-up and run-in strategies will significantly affect the computational requirements. A promising option to address this challenge is to explore the use of ROM for the bulk of the optimization simulations and then confirming a smaller number of cases with full-fidelity simulations.

8. Bibliography

[1] Fratoni, M. and E. Greenspan, "Equilibrium Core Composition Search Methodologies for Pebble Bed Reactors," *Nuclear Science and Engineering,* vol. 166, no. 1, pp. 1-16, 2010.

[2] Suparlina, L., T. Setiadipura and Suwoto, "Burnup Calculation Study of Pebble Bed Equilibrium Core," *Journal of Physics: Conference Series,* vol. 1198, no. 2, 2019.

[3] Sen, R. S., "Benchmark for Fuel Shuffling and Depletion for Pebble-Bed Reactors," Idaho National Laboratory, 2020.

[4] Gougar, H. D., A. M. Ougouag, W. K. Terry and K. N. Ivanov, "Automated Design and Optimization of PebbleBed Reactor Cores," *Nuclear Science and Engineering,* vol. 165, no. 3, pp. 245-269, 2010.

[5] Setiadipura, T. and T. Obara, "Optimization of Start-up Fuel Management for OTTO Cycle Pebble Bed Reactor," *Energy Procedia,* vol. 71, pp. 42-51, 2015.

[6] Los Alamos National Laboratory, "A General Monte Carlo N-Particle (MCNP) Transport Code," 2021. [Online]. Available: https://mcnp.lanl.gov/. [Accessed August 2021].

[7] Bess, J. D., N. Fujimoto, J. W. Sterbentz, L. Snoj and A. Zukeran, "Evaluation of the Start-Up Core Physics Tests at Japan's High Temperature Engineering Test Reactor Annular Core Loadings," March 2010.

[8] Fensin, M. L., M. R. James, J. S. Hendricks and J. T. Goorley, "The New MCNP6 Depletion Capability," Los Alamos National Laboratory, 2012.

[9] Alzamlya, M. A., M. Aziza, A. A. Badawib, H. Abou, G. Abdel and R. A. Gadallaha, "Burnup analysis for HTR-10 reactor core loaded with uranium and thorium oxide," *Nuclear Engineering and Technology,* vol. 52, no. 4, pp. 674-680, 2020.

[10] Safavia, A., M. H. Esteki, S. M. Mirvakili and M. K. Arani, "Application of a new neutronics/thermal-hydraulics coupled code for steady state analysis of light water reactors," *Nuclear Engineering and Technology,* vol. 52, pp. 1603-1610, 2020.

[11] Vazquez ,M., F. Martin-Fuertes and A. Ivanov, "Experience in neutronic/thermal-hydraulic coupling in Ciemat," in *2nd Serpent International Users Group Meeting*, 2013.

[12] Ivanov, A., V. Sanchez, R. Stieglitz and K. Ivanov, "High fidelity simulaiton of conventional and innovative LWR with the Coupled Monte-Carlo thermal-hydraulic system MCNP-SUBCHANFLOW," *Nuclear Engineering Design,* vol. 262, pp. 264-275, 2013.

[13] Kim, S. J. and C. E. Buechler, "Development of Multi-physics (Multiphase CFD + MCNP) simulation for generic solution vessel power calculation," Los Alamos National Laboratory, 2017.

[14] Bennett, A., M. Avramova and K. Ivanov, "Coupled MCNP6/CTF Code: Development, Testing and Application," *Annals of Nuclear Energy,* vol. 96, pp. 1-11, 2016.

[15] Leppanen, J., "PSG2 / Serpent – a Continuous-energy Monte Carlo Reactor Physics Burnup Calculation Code," VTT Techical Research Centre of Finland , 2009.

[16] Rintala, V., H. Suikkanen, J. Leppanen and R. Kyrki-Rajamaki, "Modeling of realistic pebble bed reactor geometries using the Serpent Monte Carlo code," *Annals of Nuclear Energy,* vol. 77, pp. 223-230, 2015.

[17] VTT Technical Research Centre of Finland, Ltd., "Input Syntax Manual," 2021. [Online]. Available: http://serpent.vtt.fi/mediawiki/index.php/Input_syntax_manual#rep. [Accessed August 2021].

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

[19] Seifert, L., A. Wheeler and O. Chvala, "Reprocessing Settings in Serpent," in *9th International Serpent Users Group Meeting*.

[20] Seifert, L., A. Wheeler and O. Chvala, "Material Flows and Online Reprocessing," in *10th International Serpent User Group Meeting*.

[21] VTT Technical Research Centre of Finland, Ltd, "Multi-physics interface," 2021. [Online]. Available: http://serpent.vtt.fi/mediawiki/index.php/Multi-physics_interface#Point-average_interface_.28type_1.29. [Accessed August 2021].

[22] VTT Technical Research Centre of Finland, Ltd., "Coupled multi-physics calculations," 2021. [Online]. Available: https://serpent.vtt.fi/mediawiki/index.php/Coupled_multi-physics_calculations. [Accessed August 2021].

[23] Ebiwonjumi, B., H. Lee, W. Kim and D. Lee, "Validation of nuclide depletion capabilities in Monte Carlo code MCS.pdf," *Nuclear Engineering and Technology,* vol. 52, pp. 1907-1916, 2020.

[24] Lopez-Solis, R. C., J. L. Francois, M. Becker and V. H. Sanchez-Espinosa, "COMPARISON OF KANEXT AND SERPENT FOR FUEL DEPLETION CALCULATIONS OF A SODIUM FAST REACTOR," in *The 19th Pacific Basin Nuclear Conference*, Vancouver, British Columbia, 2014.

[25] VTT Technical Research Centre of Finland, Ltd., "Output Parameters," 2021. [Online]. Available: http://serpent.vtt.fi/mediawiki/index.php/Output_parameters. [Accessed August 2021].

[26] Idaho National Laboratory, "Stochastic Tools," 2021. [Online]. Available: https://mooseframework.inl.gov/modules/stochastic_tools/index.html. [Accessed August 2021].

[27] Idaho National Laboratory, "RAVEN," 2021. [Online]. Available: https://raven.inl.gov/SitePages/Overview.aspx. [Accessed August 2021].

[28] Sandia National Laboratories, "Dakota," 2021. [Online]. Available: https://dakota.sandia.gov/. [Accessed August 2021].

[29] Adams, B. M. and R. W. Hooper, "Dakota Optimization and Uncertainty Quantification Software: CASL Capabilities and Impact," Sandia National Laboratories, 2020.

[30] Khuwaileh, B., B. Williams, P. Turinsky and D. Hartanto, "Verification of Reduced Order Modeling based Uncertainty/SensitivityEstimator (ROMUSE).pdf," *Nuclear Engineering and Technology,* vol. 51, pp. 968-976, 2019.

[31] International Atomic Energy Agency, "Evaluation of High Temperature Gas Cooled Reactor Performance: Benchmark Analysis Related to the PBMR-400, PBMM, GT-MHR, HTR-10 and the ASTRA Critical Facility," IAEA, Vienna, 2013.

[32] Isotalo, A. and P. Aarnio, "Comparison of depletion algorithms for large systems of nuclides," *Annals of Nuclear Energy,* vol. 38, pp. 1987-1995, 2011.

[33] Leppanen, J. and M. Pusa, "Burnup calculation capability in the PSG2 / Serpent Monte Carlo reactor physics code.," in *M&C*, Saratoga Springs, NY, 2009.

Appendix A –
Mflow Model Implementation in Serpent

The Mflow card is used to define the mass flow of certain isotopes (or elements) from one material to another. It is used in conjunction with multiple other input cards to create a reprocessing workflow. Other input cards include Rep, Rc, Rem, Dep, and Pro.

Mflow: The Mflow card defines the isotopes (or elements) that will be moved and their flow rate. The flow parameters of the given isotopes are given a name and applied with this name in the *rc* input card.

Generic input syntax:
Mflow <user_defined_name>
<X1> <f1>
<X2> <f2>
where X1 can be an isotope (i.e., 92235), an element symbol (i.e., U), or the option to move all isotopes in the material with *all* input. The f1 input is a float number that depends on what the <setting> input is on the *rc* card as to how it is interpreted. A 0 for <setting>.

Example:
Mflow feed
U 5e-6

Rep: A user-defined name for the reprocessing scheme. This name is applied in a *dep pro* input card later on in an input file.

Input syntax:
Rep <user_defined_name>

Rc: This input tells the code which material to move the isotopes from, which material to move the isotopes to, which flow scheme to apply, and what option to use for the type of flow. This input must be defined below the *rep* card. This input can be defined multiple times to define paths of material movement from different materials to other materials.

Input syntax:
Rc <mat1> <mat2> <flow_name> <option>

In this case, material flows from <mat1> to <mat2> using the flow scheme defined previously in the *Mflow* card with the name given there. The <option> can be either 0, 1, or 2. The selection of this option defines how the numbers (<f1>, <f2>, etc.) are interpreted from the *Mflow* input. See below for a discussion of details and differences for these options.

Example:
rc fuel_feed_source fuel feed_flow 0
rc fuel scrubber_tank exit_flow 1
rc scrubber_tank fuel_feed_source refill_flow 2

Dep Pro: This card activates depletion and gives the time-steps needed for the flow and depletion processes. It takes the name of the reprocessing scheme defined by the user with the *rep* card earlier in the file. This card additionally needs the same inputs as the standard *dep* card with a selection of the <step_type> and a following array of time-step list.

Input syntax:
Dep pro <user_defined_rep_name> <step_type> <t1 t2 t3 … tn>

Example:

Dep pro reprocessing_scheme_1 daystep 1 4 5 10 10 50 50 50

Notes:

- Units of flow in the Mflow card for isotopes is the fraction of mass in mat1 moved to mat2 every second:
  - This assumes the following input:
    - Mflow <flow_rate>
    - U 1.234e-8
    - *Rep* <reprocess_name>
    - *Rc* mat1 mat2 <flow_rate> 0.
- The *rep* input and the *rc* input need to be together.
- The fourth entry in *rc* card is the setting option. It can be 0, 1, or 2. Here is a summary of the options:
  - 0 – constant mass flow given in g/s. does not deplete the source material. This can result in more mass than initially defined.
  - 1 – depletes the source material. Total flow rate dependent on source amount. Value in Mflow card is fraction of material in source moved to next material. Results in exponential decay removal of source mass/material.
  - 2 – This is the same as option 1, except the step at the initial time value cannot cause discontinuity of mass.
- Using in conjunction with "printm 1" and using the *inventory* card to define isotopes to print in the _dep.m output file allows for the export of isotopic data in each material throughout the simulation.
- Multiple *rc* cards can be used to define multiple flows from different materials at different flow rates.

Limitations:

- Material depletion with reprocessing only works with a shared memory parallelism (i.e., it cannot use Message passing interface (MPI) or depletion does not work [Mar 2016]). There may be a work-around by changing the source code (see Jaakko post Mar 30, 2016 7:34 am)).
- Cannot use ZAID in Mflow card. Must use 'all' setting or element names (e.g., uranium, carbon).
- Using 'continue' option does not work in "*rsr*" card. Depletion days must be chosen explicitly.
- In the source code for Serpent, there is a code developer's note that documents a potential bug. The note is in Finnish and is as follows: "Tässä on se potentiaalinen ongelma että muodoissa 92235 ja U-235 annettuja nuklidinimiä ei tunnisteta samaksi." Google translation provides a translation of: "Here is the potential problem that the nuclide names given in Forms 92235 and U-235 are not recognized as the same." According to the user's experience, this problem exists as stated.