Development of Neutronics Analysis Capabilities for Application to Flowing Fuel Molten Salt Reactors

by

Mustafa K. Jaradat

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Nuclear Engineering and Radiological Sciences)
in the University of Michigan
2021

Doctoral Committee:

Professor Won Sik Yang, Chair
Professor Thomas Downar
Professor Krzysztof Fidkowski
Dr. Changho Lee, Argonne National Laboratory

Mustafa K. Jaradat

mjaradat@umich.edu

ORCID ID: 0000-0003-2750-6055

In memory of my late father, I dedicate this thesis. I wish he could be here to see me through this, yet somehow, I know he is. May Allah bless his soul with mercy and forgiveness.

Acknowledgements

First of all, I would like to thank everyone who helped and supported me professionally, academically, and personally during my Ph.D. study. I would like to express my sincere gratitude to my advisor Professor Won Sik Yang, for allowing me to pursue my Ph.D. under his guidance and to thank him for his help, patience, support, and directing me on the right path. I learned a lot from him during the past few years, and I am grateful for every moment he spent teaching and advising me.

I would like to thank the committee members for their invaluable support, helping me in growing my scientific research skills, the wise advice, and all that I have learned from your classes. In particular, I would like to thank Dr. Changho Lee for securing funding resources to accomplish this work and his valuable comments and discussions over the stages of this work. I would also like to thank the Department of Energy (DOE) and Argonne National Lab (ANL) for funding almost all my thesis research under Contract number DE-AC02-06CH11357 and DOE-NE Nuclear Energy Advanced Modeling and Simulation (NEAMS) program.

I would also like to thank my colleagues in Nuclear Reactor Design and Simulation Lab (NRDSL) especially my friend Gang Yang for the great times we had together, valuable discussions, exchange of ideas, and late-night work. I would like to thank the Nuclear Engineering and Radiological Sciences (NERS) department for all the help and support I had received.

Most importantly, I would like to thank my family, my beloved mother, sisters, brothers, nieces, and nephews for their unconditional love and support. To my late father, who inspired and encouraged me to set and fulfill ambitious goals all over my life. Fifteen years ago, I gave him a promise of getting my Ph.D., and now I honor my promise to him, I wish him here. To my little brother “Fares”, whom I love the most, I wish him all the success in shaping his future.

Table of Contents

List of Tables

List of Figures

List of Algorithms

List of Appendices

Abstract

As one of generation IV reactor designs, molten salt reactors (MSRs) with fuel dissolved in liquid salt are gaining increasing interest from the industry due to their excellent characteristics in sustainability, passive safety, and resource utilization. Most of the currently proposed MSR concepts with liquid fuel utilize flowing fuel. The design and safety analyses of flowing fuel MSRs require new modeling and simulation capabilities that can address the unique characteristics of flowing liquid fuel, which are different from conventional solid fuel reactors. The fuel salt circulates throughout the primary system, acting as fuel and coolant at the same time. Thus, part of the delayed neutron precursors generated in the core decay in positions of low importance or even out of the core. This changes the effective delayed neutron fraction and makes the dynamic behavior of MSRs different from solid fuel reactors. Furthermore, the neutronics and thermal hydraulics are strongly coupled due to the large thermal expansion of liquid fuel.

Most of the available neutronics tools for the design and safety analysis of MSRs are based on various approximations tailored to a targeted system. Motivated by the need for a general neutronics analysis tool for application to fast and thermal MSRs, this thesis develops and implements new modeling capabilities for flowing fuel to the nodal transport code PROTEUS-NODAL. The steady state solver has been extended to model the drift of the delayed neutron precursors, and new transient solvers have been developed for flowing fuel applications. A thermal hydraulics solver has also been developed and implemented to address the strong coupling of the neutronics and thermal hydraulics by considering the velocity, temperature, and density fields in the thermal feedback calculations. In order to perform efficient and practical transient analyses of MSRs, an adaptive time-step selection scheme has been developed based on control theory. In this scheme, the time-step size is varied based on the estimated local solution errors during the transient. This helps in reducing the computational inefficiency due to unnecessarily small time-step size and avoiding the loss of accuracy that might be introduced due to a large time-step size.

Additionally, MSR fuel cycle analysis capability has been implemented by considering online refueling of fuel salt and reprocessing to extract fission products. Utilizing the fuel cycle model,

fission products are categorized into a few decay heat precursor groups based on their chemical characteristics to include the decay heat in the transient and fuel cycle analyses, and a nuclide drift model has been added to consider the distribution of the fuel salt nuclides in the feedback model to simulate a hypothetical over-fueling of the fuel salt during normal operation.

The developed capabilities have been verified using the Molten Salt Fast Reactor (MSFR) numerical benchmarks and validated against the available measured data of the Molten Salt Reactor Experiment (MSRE). The verification and validation test results demonstrate that these capabilities can model the relevant physics of flowing fuel MSRs accurately and efficiently. The developed code can be used for steady state, transient, and fuel cycle analyses of thermal and fast spectrums MSRs designs.

Chapter 1. Introduction

The concept of the molten salt reactors (MSRs) with liquid fuel was introduced in 1945, by dissolving the fuel into a liquid salt such as molten fluoride and by circulating the salt throughout the primary loop in the reactor, utilizing the salt as fuel and coolant at the same time. The advantages of the molten salts over other coolants lie in the elimination of the high-pressure operation as in water and gaseous coolants, and chemically less reactive to water or air as in liquid metal coolants [1, 2]. However, the motion of the liquid fuel salt results in a partial release of the delayed fission neutrons outside the reactor core due to the decay of the fission products. The fuel motion and the decay process are reflected directly on the neutronics parameters, which will change the behavior of the reactor compared to the conventional solid fuel reactors. Also, refueling and reprocessing of the liquid fuel can be performed while the reactor is in operation. These phenomena of the liquid fuel must be addressed in performing neutronics analyses, which add more complications to the modeling and simulation tools compared to the solid fuel.

The available tools for modeling and simulation of liquid fuel MSRs are focused on certain type of MSRs with several approximations. In order to provide accurate and efficient analysis capabilities for both fast and thermal spectrum MSRs, this thesis develops, verifies, and validates modeling and simulation capabilities for steady state and time-dependent neutronics analyses of MSRs with flowing fuel. The developed capabilities include steady state solvers for forward and adjoint equations, transient solvers with adaptive time-stepping scheme, nuclide depletion and fuel cycle analysis solver, and models for decay heat, thermal hydraulics, and thermal feedback to perform coupled neutronics and thermal hydraulics calculations. The developed capabilities are verified using numerical benchmarks of MSRs and validated against the available experimental data of liquid fuel MSRs.

This chapter highlights the concept and development of liquid fuel MSRs, and briefly reviews the progress and development of the existing modeling tools for safety analysis of MSRs. Then,

the motivation and goals of the work of this thesis are presented. At the end of this chapter, the outline of this thesis is provided.

1.1. MSRs Design and Development

MSRs have several advantages over other reactors as homogeneous fuel composition, lower fissile inventories, low radiation damage, and no fuel fabrication or handling. Besides low-pressure operation, MSRs have large negative temperature and void coefficients of reactivity, and they passively reduce reactor power increase in the case of reactivity insertion accident. Also, if excessive temperatures are reached, the fuel salt in the primary system will be drained by gravity to a decay tank. Furthermore, the MSR can be configured in fast and thermal neutron spectrums depending on the use of a solid moderator with liquid fuel salt. In thermal spectrum MSR, the core is configured of un-cladded graphite channels allowing the fuel to flow through it for neutron moderation. However, there is a limitation on the operation of the graphite moderated reactors due to exposure of the graphite to the high neutron flux. In fast spectrum MSR, the fuel flows in the reactor region without passing through flow channels. In both designs, fuel reprocessing is required to remove gaseous fission products continuously from the fuel salt. Also, MSRs can be designed with single-fluid as in homogenous MSR or two-fluid as in heterogeneous MSR. In homogenous MSR, thorium can be dissolved with the fissile isotopes in a single fluid mixture. However, heterogeneous MSR would have fertile salt containing thorium that is separated from the fuel salt to be operated as a breeder reactor to produce ^{233}U in the blanket region which is continuously extracted and transferred to the fuel salt in the primary system to compensate for the fuel burnup [1, 2, 3, 4, 5].

1.1.1. History and Current Designs

The history of liquid fuel reactors dates back to the late 1940s when it was originally proposed at Oak Ridge National Laboratory (ORNL). The construction and operation of the Aircraft Reactor Experiment (ARE) in 1954 successfully demonstrated the MSR concept after 100 hours of operation until shutdown. It was realized that the crash of a nuclear-powered plane would have serious consequences [6]. In the 1960s, the Molten Salt Reactor Experiment (MSRE) was constructed and successfully operated [7, 8]. The main purpose of the MSRE was to demonstrate the practicality of the liquid fuel operation at high temperatures and to ensure the safety and reliability for developing a full-scale molten salt breeder reactor (MSBR), which was suspended

during the design stage. Figure 1.1 shows the reactor assemblies of the ARE and MSRE. In 1969, the MSRE was operated with ^{233}U fuel which could be bred from thorium fuel salt demonstrating the visibility of MSBR, which was designed to operate with two salts mainly to breed ^{233}U isotope from fertile thorium blanket. The MSRE program was terminated by the US government in favour of the development of the liquid metal fast breeder reactor concept to breed ^{239}Pu from ^{238}U instead of breeding ^{233}U from ^{232}Th as proposed in MSBR design [9, 10].

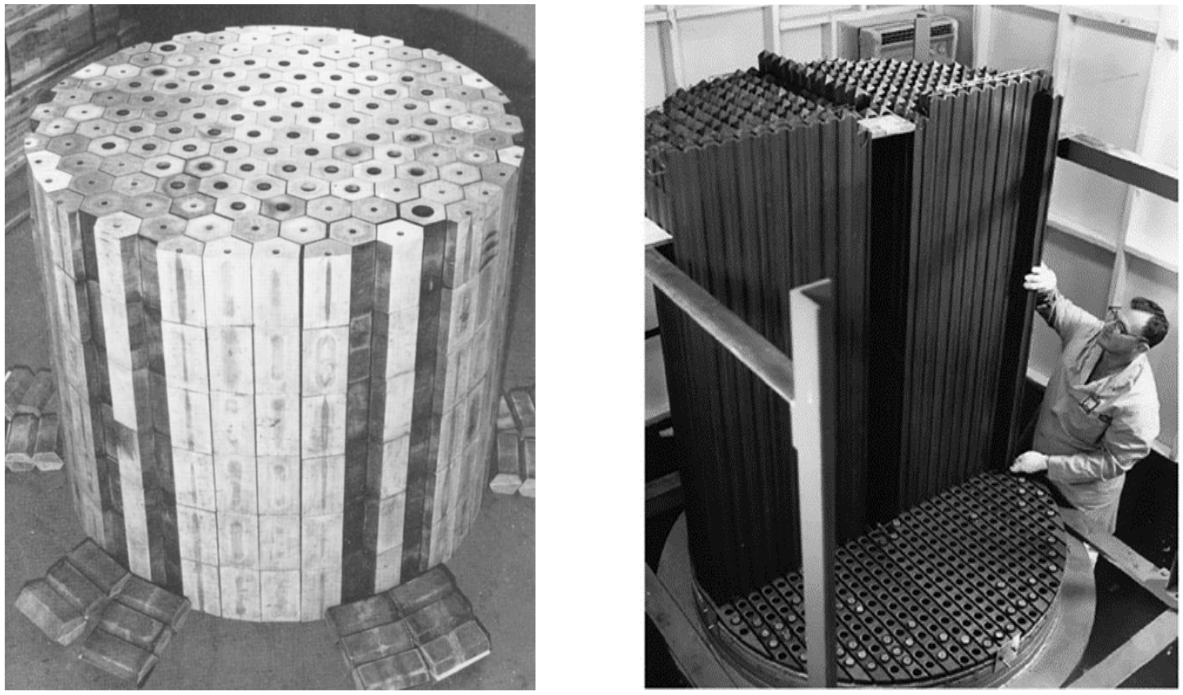


Figure 1.1. Reactor assemblies of ARE (left) [6] and MSRE (right) [10].

In the early 2000s, there was a renewed interest in the MSR concept, and it was selected as one of the Generation IV reactors [11]. However, different designs of MSRs are being developed considering flowing fuel in fast and thermal spectrums with closed fuel cycle, breeding capabilities, and multi-recycling of actinides. Other designs of MSRs utilize a once-through fuel cycle without online refueling. Also, some designs utilize static liquid fuel with online refueling to leverage the inherent safety aspects of the liquid fuel salt.

In the Generation IV International Forum, liquid fueled MSR research and development have focused on fast spectrum MSR options for fissile breeding and transuranic (TRU) isotope burning, including the 1000 MWe MOlten-Salt Actinide Recycler and Transmuter (MOSART) in Russia [12] and the 1400 MWe thorium Molten Salt Fast Reactor (MSFR) in France [13, 14]. The MSFR

was designed to utilize the thorium fuel cycle with mixed lithium fluoride and actinide fluoride salt. A schematic diagram of the MSFR is provided in Fig. 1.2. In the U.S., Molten Chloride Faster Reactor (MCFR) design [15] was proposed by TerraPower which is similar to the MSFR concept. MCFR utilizes a fast spectrum with a continuous fueling process and fuel in form of molten chloride salts.

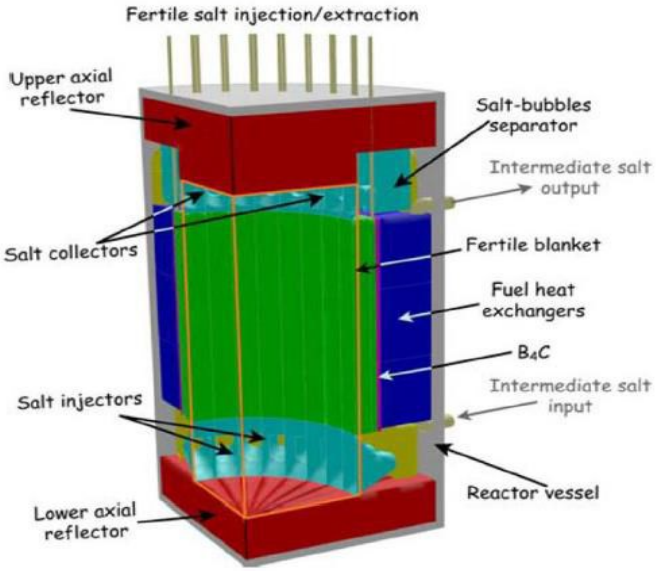


Figure 1.2. Schematic diagram of MSFR [14].

A small modular Integral Molten Salt Reactor (IMSR) with 190 MWe is being designed by Terrestrial Energy [16] in Canada. The IMSR is based on the MSRE experiment with a reconfiguration and scale-up. The IMSR utilizes a thermal neutron spectrum with a hexagonal arrangement of solid graphite moderator. It utilizes a once-through uranium fuel cycle with the fuel dissolved in fluoride salt. It integrates the primary system components in a sealed, compact, and replaceable core vessel with a projected life of seven years. Also, the concept of using static or stable salt reactor (SSR) with fuel salt is filled in fuel rods and assemblies similar to conventional solid fuel designs was suggested by Moltex to utilize the inherent safety aspects of the liquid fuel salt while maintaining the design features of the traditional solid-fuel reactors [17]. The suggested design has a power of 320 MWe. The individual fuel tubes are vented to allow gaseous fission products to escape into the coolant salt. It requires online refueling of fuel assemblies with liquid salt to maintain continuous operation.

1.1.2. Analysis Tools Development

As discussed in the previous subsection, several designs of MSRs are being developed and new modeling and simulation tools are required to perform design and safety analyses of the proposed MSR concepts to address their unique features and characteristics, which are different from conventional solid fuel reactors. The fuel salt circulates throughout the primary system, acting as fuel and coolant at the same time, and thus part of the delayed neutron precursors generated in the core decay in positions of low importance or even out of the core. This is reflected as reactivity losses due to the reduction of the effective delayed neutron fraction, which makes the dynamic behavior of MSRs different from solid fuel reactors. The heat generated by fission is deposited directly into the fuel salt and extracted from the fuel salt in the heat exchanger region. In thermal MSRs, a fraction of the heat is generated in the moderator due to photon interactions and neutron scattering. Also, part of the decay heat is generated in the outer loop mainly in the heat exchanger due to the flow of fuel salt. Figure 1.3 shows a representation of the flow of the fuel salt in the reactor core and the outer loop. Furthermore, the neutronics and thermal hydraulics are strongly coupled due to the liquid fuel and its motion in MSRs. Also, the online reprocessing and refueling of the fuel salt are required to perform fuel cycle analysis of MSRs. This increases the need for advanced design analysis capabilities for MSR applications.

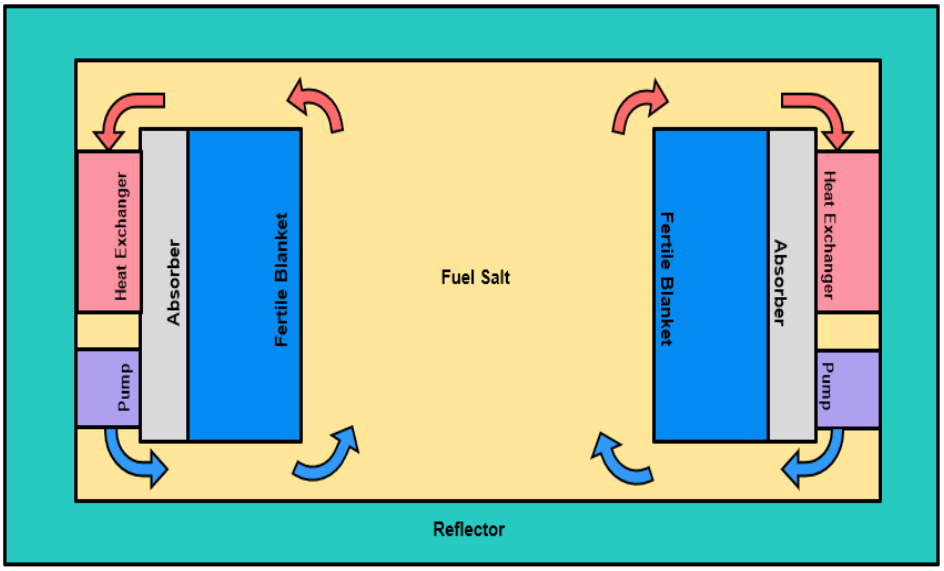


Figure 1.3. Flow of fuel salt in reactor core and outer loop of MSR.

Over the years, various efforts were made to develop simulation tools for the coupled neutronics and thermal hydraulics analysis of flowing fuel MSRs, focused on certain types of

MSRs with various approximations. To assess the available dynamic simulation tools for MSRs, benchmark exercises were developed as well. A thermal-spectrum MSR benchmark was developed based on some of the MSRE experiments under the European MOST project [18]. Also, a fast-spectrum MSR benchmark was also defined under the Evaluation and Viability of Liquid fuel fast reactor system (EVOL) project for the pre-conceptual design and steady state analysis of the MSFR design [12].

For the application to thermal spectrum MSRs, ORNL developed in the 1960s a modified point kinetics model for circulating fuel with a heat transfer model using a multi-node approach to perform dynamic analysis of the MSRE [19]. In the past 20 years, several efforts were dedicated to code development of diffusion and Monte Carlo codes, like the modified version of the MCNP4C code [20] which was used for steady state analysis of the MSRE [21]. Also, the Cinsf1D code [22] was developed for MSR analysis based on the two-group diffusion theory. For transient analysis, the quasi-static method was generalized for modeling the flowing fuel system in two-dimensional (2D) cylindrical geometry [23]. For dynamic analysis, the DYN1D-MSR [24] and DYN3D-MSR [25] codes were developed based on a two-group nodal expansion method in hexagonal geometry for safety analysis of graphite-moderated MSRs. Similarly, the MOREL code [26, 27] is a recent code that was dedicated to the channel-type MSR analysis. Recently, nonlinear dynamic models were developed to perform dynamic analysis of the MSRE experiments [28] following the same approach of ORNL in the 1960s. Also, an open-source simulation tool Moltres [29] was developed for the simulation of MSRs by coupling neutronics and thermal hydraulics modules in the MOOSE framework [30]. More recently, a set of MSRE benchmarks were developed based on the initial zero-power criticality experiment [31] and the measured reactivity data [32] have been included in the International Reactor Physics Experiments Evaluation Project (IRPhEP) handbook. Also, the System Analysis Module (SAM) [33] was modified to model MSRs by including a delayed neutron precursor drift model and a modified point kinetics model for flowing fuel [34].

For fast-spectrum MSR applications, the SIMMER-III code [35] was modified for the analysis of the MOSART design, focused on the thermal hydraulics behavior of MSRs [36]. For the MSFR benchmark transient analysis, the Politecnico di Milano and the Delft University of Technology (TUDelft) performed a safety analysis through coupled neutronics and computational fluid dynamics (CFD) calculations in R-Z geometry [37]. Also, the Paul Scherrer Institute (PSI) coupled

the system analysis code TRACE [38] with the nodal neutronics code PARCS [39] to perform a similar analysis [40] utilizing a hexagonal geometry. To access multiphysics analysis codes for fast spectrum MSR applications, a numerical benchmark was developed within the SAMOFAR project [41]. Recently, the open-source CFD code OpenFOAM [42] was adopted in several multiphysics analysis codes for transient analysis of fast spectrum MSRs, like coupling the OpenFOAM toolkit with a one-group diffusion solver to perform MSFR transient analysis [43]. Also, a multiphysics code TANSY was developed by coupling the OpenFOAM toolkit and a multigroup diffusion solver and applied for transient analysis of MSFR [44].

In addition, many fuel cycle analysis codes have been developed to simulate the process of online reprocessing and refueling in MSRs. In most of these developed codes, the effects of the fuel flow and the thermal feedback were not considered in the performed analysis.

1.2. Motivation

Most of the available neutronics tools for design and safety analysis are developed for a certain type of MSRs and have limitations on applicability and require several approximations. Also, the availability of simulation tools that can be used to perform neutronics analyses, including steady state, transient, and fuel cycle analysis, using a single developed model is very limited. Motivated by the needs for accurate and efficient neutronics analyses tool for both fast and thermal spectrums MSRs and due to limitations of the current existing tools, new modeling and simulation capabilities for MSRs with liquid flowing fuel have been developed and implemented into the PROTEUS-NODAL code, which is a three-dimensional (3D) variational nodal transport code based on a homogeneous assembly model for various nuclear reactor applications developed at Argonne National Laboratory (ANL) under the U.S. DOE's Nuclear Energy Advanced Modeling and Simulation (NEAMS) program [45, 46].

The steady state solvers have been extended to model flowing fuel the delayed neutron precursors drift. Also, transient solvers have been developed for flowing fuel applications with acceleration scheme and thermal feedback capability which includes thermal hydraulics models of the reactor and the outer loop, and models to functionalize cross sections for fast and thermal spectrum reactors [47, 48, 49, 50]. To improve the efficiency of the transient capabilities of MSRs and to reduce the total computational time without loss of accuracy, an adaptive time-stepping algorithm has been developed based on the control theory approach. The developed steady state

and transient analyses capabilities have been verified using the MSFR numerical benchmark [7] and validated against the measured data of the MSRE Experiment [13] which are the only source of experimental data for MSRs with flowing fuel [51, 52, 53].

The main objective of this thesis is to develop neutronics analysis capabilities for steady state and time-dependent analyses of MSRs within a single tool. To fulfil this objective, a fuel cycle analysis capability has been developed to perform depletion calculations and to obtain fission products build-up in the fuel salt. A two-point depletion model considering the core and the outer loop regions has been developed with the capability of online refueling and reprocessing. In order to perform a more accurate and realistic analysis, the heat generated from the decay of the fission products and its contribution to steady state, transient, and fuel cycle analyses need to be considered. A decay heat model has been added considering the drift of the fuel to obtain the distribution of the decay heat precursors in the core and the fractional release of decay heat in the outer loop. Utilizing the added fuel cycle capability, a procedure to categorize fission products into a few decay heat precursor groups with unique decay constant and fraction is developed and applied considering fission products chemical characteristics and removal time during the fuel reprocessing.

In real operation of MSRs with online refueling, fissile and fertile components are injected into the fuel salt in the outer loop while the reactor is operational. This will result in reactivity perturbations due to redistribution of the fuel salts nuclides in the core. An over-fueling of either of the components will induce reactivity insertion transient, and the reactor power and fuel salt temperature will change accordingly. In order to study this unique transient scenario, a nuclide drift model has been developed to consider the redistribution of the fuel salt nuclides in the core region to simulate over-fueling accident scenario. This will add an extra component to the feedback model by considering the distribution of the salt nuclides in the core in updating the cross sections along with the density and temperature distributions. In this way, a more realistic thermal feedback model of MSRs can be utilized for multiphysics analysis.

1.3. Dissertation Layout

This thesis consists of eight chapters and can be divided into three parts. The first part focuses on the steady state and transient analyses of MSRs with verification and validation tests. The second part is dedicated to improving the transient solution by providing an algorithm to select the

size of the time-step. The last part discusses the fuel cycle analysis of the MSRs and further improvements for more realistic analysis of MSRs.

The mathematical foundation for steady state and transient analyses of the flowing fuel and the solution methodology are discussed in Chapter 2. First, the PROTEUS-NODAL methodology is discussed briefly. Then, the steady state solution scheme of the flowing fuel is discussed, along with the adjoint solution. This is followed by a discussion of the developed transient solvers based on the transient fixed source problem (TFSP) and the solution of the delayed neutron precursor concentration equations for stationary and flowing fuels with a discussion on the coarse mesh finite difference (CMFD) acceleration scheme. Then, a description of the implemented thermal feedback capabilities is provided, including thermal hydraulics solver, outer loop and heat exchanger models. Then, the cross sections functionalization for fast and thermal spectrum reactors are provided, along with the coupling scheme of the neutronics and thermal hydraulics solvers.

Chapter 3 provides numerical verification results of the steady state and transient analysis capabilities discussed in Chapter 2 for fast and thermal spectrum reactors. Two problems are analyzed, the first problem is the MSFR benchmark problem, and the second problem is the MOST benchmark exercise. The MSFR benchmark problem contains steady state tests for the stationary fuel eigenvalue, temperature coefficients, kinetics parameters, and reactivity losses due to fuel salt drift. The transient problems have several unprotected transients categorized into three groups based on the driving force as reactivity-, flow-, and temperature- driven transients. The reactivity-driven transients focused on the unprotected transient over power (UTOP) where sub-prompt and super-prompt reactivities are inserted. The flow-driven transients are related directly to the velocity of the fuel salt and its direct impact on the solution of the delayed neutron precursor equation and the thermal feedback, as in the unprotected pump over speed (UPOS) and unprotected loss of flow (ULOF) accidents. The temperature-driven transients are related to the inlet temperature of the fuel salt and the cooling rate at the heat exchanger and the secondary side, as in the unprotected loss of heat sink (ULOHS) and unprotected fuel salt over cooling (UFSOC) accidents. Also, the simplified problem of the MSRE experiment that was defined under MOST benchmark exercise is simulated. The MOST benchmark contains steady state tests for temperature coefficients and reactivity losses of the effective delayed neutron fraction. The transient problems include protected pump transients at low power and the natural circulation test. For both benchmark problems,

PROTEUS-NODAL results are presented in comparison with other reference solutions obtained from the open literature.

Validation test results using the experimental data of the MSRE experiment are provided in Chapter 4. First, the geometry modeling and the multigroup cross sections generation procedure using Monte Carlo code OpenMC are discussed. Then the PROTEUS-NODAL model is verified against OpenMC results. The outer regions of the MSRE configuration beyond the reactor vessel are excluded from the model, instead, Albedo boundary conditions generated with OpenMC are employed at the outer surface of the reactor vessel in order to preserve the reaction rates and the total neutron leakage. Also, the OpenMC model is used to calculate the fractional heat generated in the graphite moderator due to neutron slowing down and gamma reactions. The experimental data are available for steady state experiments for control rod worth, temperature coefficients, and reactivity losses of the effective delayed neutron fraction. The transient experiments contain the unprotected pump start up and coast down and the natural circulation test. The simulation results in comparison to the measured values are presented using the developed model of the exact MSRE core.

In Chapter 5, an adaptive time-stepping algorithm is developed to enhance the transient analyses of MSR based on the control theory approach. The developed approach and the implementation scheme are discussed first. Then, test results of the developed controller algorithm are presented using the MSFR benchmark and the MSRE experiment to ensure and verify its robustness and applicability for various types of transients including short-prompt and slow-long transients. This is followed by a parametric study with suggested ranges of the preselected controller parameters are provided. The same approach can be applied to the depletion analysis to determine the time-step size while maintaining an accurate solution.

Fuel cycle analysis capabilities of MSRs with online reprocessing and refueling are discussed in Chapter 6. The modified nuclide depletion equation for flowing fuel with refueling and reprocessing is provided first. Then, the solution algorithms with fissile feed rate and criticality search option are discussed, followed by verification test results using the MSFR fuel cycle benchmark. Furthermore, the effects of considering fuel flow, coupling to thermal hydraulics, and decay heat on the calculated critical fissile feed rate and breeding ratio are investigated. In Chapter 7, further improvements to the feedback model are added by considering the decay heat and the nuclide distribution in the transient analysis. A decay heat model with a description of the decay

heat calculation procedure is provided. Also, categorizing the decay heat into several precursor groups to be used in the decay heat model for neutronics analysis with simulation results for steady state and transient analysis are discussed. Finally, the results and conclusions of the work are summarized in Chapter 8, and potential topics for future research are mentioned.

Chapter 2. Theory and Methodology

In a liquid fuel MSR, the fuel is dissolved into the liquid salt that is circulated throughout the whole primary system. Some of the delayed neutron precursors generated in the core may decay in a position of low importance or even out of the core, which will reduce the effective delayed neutron fraction of the reactor. As a result, the multiplication factor of MSR will be less than a reactor with conventional solid fuel when all the other conditions are the same. The movement of the liquid fuel and the subsequent decay of the delayed neutron precursors during the fuel motion need to be considered explicitly in modeling a liquid fuel MSR. This makes the existing neutronics tools for conventional stationary fuel reactors are not valid for the application of liquid fuel MSR with flowing fuel, which requires modification and development of the neutronics tools to consider the motion of the fuel salt in solving the delayed neutron precursor equations to account for their decay outside the core and their distribution in the core region for steady state and transient analyses, which is different from the stationary fuel.

For flowing fuel, the time-dependent Boltzmann transport equation for the neutron flux and the delayed neutron precursor concentration equation in the multigroup form are given by

$$\frac{1}{v_g}\frac{\partial}{\partial t}\psi_g(\vec{r},\Omega,t)+\Omega\cdot\nabla\psi_g(\vec{r},\Omega,t)+\Sigma_{tg}(\vec{r},t)\psi_g(\vec{r},\Omega,t)=\sum_{g'=1}^{G}\Sigma_{sg'\to g}(\vec{r},\Omega'\cdot\Omega,t)\psi_{g'}(\vec{r},\Omega',t) + \lambda\frac{\chi_{pg}(\vec{r},t)}{4\pi}\sum_{g'=1}^{G}\nu_p\Sigma_{fg'}(\vec{r},t)\phi_{g'}(\vec{r},t)+\frac{1}{4\pi}\sum_{k=1}^{K}\chi_{dkg}(\vec{r},t)\lambda_k C_k(\vec{r},t),\quad g=1,2,\cdots,G, \tag{2.1}$$

$$\frac{\partial}{\partial t}C_k(\vec{r},t)+\nabla\cdot[\vec{u}(\vec{r},t)C_k(\vec{r},t)]+\lambda_k C_k(\vec{r},t)=\lambda\sum_{g'=1}^{G}\nu_{dk}\Sigma_{fg'}(\vec{r},t)\phi_{g'}(\vec{r},t),\quad k=1,2,\cdots,K\,, \tag{2.2}$$

where

$\vec{r}$ = position vector.
Ω = angular direction.
t = time variable.
g = energy group index, $g=1,2,\cdots,G$.
v_g = neutron velocity.
ψ_g = neutron angular scalar flux.

ϕ_g = neutron scalar flux.

Σ_{tg} = macroscopic total cross section.

Σ_{fg} = macroscopic fission cross section.

$\Sigma_{sg'\to g}$ = macroscopic scattering cross section from group g' to group g.

C_k = concentration of delayed neutron precursor of family k.

k = delayed neutron precursor family index, $k = 1, 2, \cdots, K$.

λ_k = decay constant of precursor family k.

$\vec{u}$ = velocity of fuel salt.

χ_{pg} = fraction of prompt neutrons emitted into the energy group g.

χ_{dkg} = fraction of delayed neutrons emitted into the energy group g from the precursor family k.

ν_p = number of prompt neutrons per fission.

ν_{dk} = number of delayed neutrons of precursor group k per fission.

$\lambda = 1/k_{eff}$ = eigenvalue of the problem.

The term $\nabla \cdot [\vec{u}(\vec{r},t) C_k(\vec{r},t)]$ in Eq. (2.2) accounts for the drift of the fuel. In order to solve Eq. (2.2), it must be completed with a boundary condition along with the initial condition. Also, the eigenvalue is introduced in the above equations to make the time derivatives identically zero at the initial steady state condition in case the system is off critical.

In order to model and simulate MSRs with flowing fuel, new modeling capabilities of flowing will be implemented into the PROTEUS-NODAL code [45, 46] which is a 3D variational neutron transport code based on a homogeneous assembly model of conventional solid fuel for various nuclear reactor applications that is being developed at ANL and it is applicable for stationary fuel only. The code has two solution options: the P_1 solver based on the DIF3D-VARIANT methodology [55] and the simplified P_3 (SP_3) solver based on the triangle-based polynomial expansion nodal (TPEN) method for the radial direction and the one-dimensional nodal expansion method (NEM) for the axial direction [45, 56]. The P_1 solver is applicable to Cartesian, triangular, and hexagonal geometry problems, while the SP_3 solver is applicable to hexagonal geometry problems only. Recently, a new P_1 solver for R-Z and R-θ-Z geometries has been developed based on the variational nodal method to apply the P_1 solver to cylindrical geometry problems. In this study, the steady state solvers of the PROTEUS-NODAL code have been extended to model MSRs with flowing fuels by considering the delayed neutron precursors drift [47, 48]. Transient analysis capabilities for stationary and liquid fuels have been developed, which solve transient fixed source

problems (TFSPs) in all available reactor geometries of PROTEUS-NODAL. A Coarse Mesh Finite Difference Acceleration (CMFD) scheme has been implemented for steady state and transient solvers to enhance the calculation speed and reduce the computational time. In addition, a thermal hydraulics solver is implemented in the PROTEUS-NODAL code based on a parallel one-dimensional (1D) multi-channel model for the core and simple lumped parameter models for the components outside of the core region [49, 53].

In this chapter, the methodology of the PROTEUS-NODAL code for steady state analysis is first discussed. This is followed by a discussion on the steady state solution of precursor concentration equation for flowing fuel and its solution scheme for forward and adjoint analyses with the kinetics parameters evaluation method. Then the method to solve the TFSP for stationary fuel is discussed. This is followed by the discussion on the modified precursor concentration equation for flowing fuel and its solution scheme. The CMFD acceleration method for transient analyses, the thermal feedback models, including thermal hydraulics, primary loop, cross sections functionalization, and the coupling scheme of the neutronics and thermal hydraulics calculations are described in the subsequent subsections.

2.1. PROTEUS-NODAL Methodologies

In the PROTEUS-NODAL code, two solution methodologies are implemented: P_N and simplified P_N (SP_N) approaches and both methods are applicable for stationary fuel only. The P_N approach is identical to the methodology used in DIF3D-VARIANT [55, 57, 58] utilizing P_1 approximation and variational nodal methods. The P_1 approximation can be derived from Eq. (2.1) since it is applicable for both stationary and flowing fuels, while Eq. (2.2) is applicable for flowing fuel and can be applied for stationary fuel if the precursors drift term is set to zero as discussed in the following derivation. Then, Eq. (2.1) can be transformed into an even parity form by splitting up the angular flux into even and odd parity components as

$$\psi_g^+(\vec{r},\Omega,t)=\frac{1}{2}\left[\psi_g(\vec{r},\Omega,t)+\psi_g(\vec{r},-\Omega,t)\right], \tag{2.3}$$

$$\psi_g^-(\vec{r},\Omega,t)=\frac{1}{2}\left[\psi_g(\vec{r},\Omega,t)-\psi_g(\vec{r},-\Omega,t)\right], \tag{2.4}$$

$$\psi_g(\vec{r},\Omega,t)=\psi_g^+(\vec{r},\Omega,t)+\psi_g^-(\vec{r},\Omega,t), \tag{2.5}$$

where the superscript + denotes even parity and – denotes odd parity. Introducing Eq. (2.5) into

Eq. (2.1) and evaluating it at $+\Omega$ and $-\Omega$ with the isotropic scattering assumption, a system of coupled time-dependent even- and odd-parity equations can be written as

$$\frac{1}{v}\frac{\partial}{\partial t}\psi_g^+(\vec{r},\Omega,t)+\Omega\cdot\nabla\psi_g^-(\vec{r},\Omega,t)+\Sigma_{tg}(\vec{r},t)\psi_g^+(\vec{r},\Omega,t)=q_g^+(\vec{r},t)\,, \tag{2.6}$$

$$\frac{1}{v}\frac{\partial}{\partial t}\psi_g^-(\vec{r},\Omega,t)+\Omega\cdot\nabla\psi_g^+(\vec{r},\Omega,t)+\Sigma_{tg}(\vec{r},t)\psi_g^-(\vec{r},\Omega,t)=0\,, \tag{2.7}$$

$$\begin{aligned} q_g^+(\vec{r},t)=&\frac{1}{4\pi}\sum_{g'=1}^{G}\Sigma_{sg'g}(\vec{r},t)\phi_{g'}(\vec{r},t)+\lambda\frac{\chi_{pg}(\vec{r},t)}{4\pi}\sum_{g'=1}^{G}\nu_p\Sigma_{fg'}(\vec{r},t)\phi_{g'}(\vec{r},t)\\ &+\frac{1}{4\pi}\sum_{k=1}^{K}\chi_{dkg}(\vec{r},t)\lambda_k C_k(\vec{r},t) \end{aligned}\,. \tag{2.8}$$

In order to simplify the solution scheme of the neutron flux equations, the time derivative of odd-parity flux is neglected, and the odd-parity flux component can be written in terms of the even-parity flux component as

$$\psi_g^-(\vec{r},\Omega,t)=-\frac{1}{\Sigma_{tg}(\vec{r},t)}\Omega\cdot\nabla\psi_g^+(\vec{r},\Omega,t)\,, \tag{2.9}$$

The basis of this approximation is derived from the approach used to solve the transient diffusion equation in which Fick's Law is assumed to be valid under a transient condition with small perturbations and neglecting the time derivative of the current. This is a valid approximation since the current formulation focuses mainly on perturbations due to material temperature changes specifically the cross sections and density, or due to velocity perturbations. By substituting Eq. (2.9) into Eq. (2.6), a second-order equation for the time-dependent even-parity flux can be obtained as

$$\frac{1}{v}\frac{\partial}{\partial t}\psi_g^+(\vec{r},\Omega,t)+\Omega\cdot\nabla\left[\frac{-1}{\Sigma_{tg}(\vec{r},t)}\Omega\cdot\nabla\right]\psi_g^+(\vec{r},\Omega,t)\ +\Sigma_{tg}(\vec{r},t)\psi_g^+(\vec{r},\Omega,t)=q_g^+(\vec{r},t)\,, \tag{2.10}$$

This differential equation is accompanied with boundary conditions that constrain the angular flux along the boundary of the problem domain. The scalar flux can be written in terms of the even-parity flux and the net current in terms of the odd-parity flux as

$$\phi_g(\vec{r},t)=\int_{4\pi}\psi_g^+(\vec{r},\Omega,t)d\Omega\,, \tag{2.11}$$

$$j_g(\vec{r},t)=\int_{4\pi}\Omega\psi_g^-(\vec{r},\Omega,t)d\Omega=\frac{-1}{\Sigma_{tg}(\vec{r},t)}\int_{4\pi}\Omega\Omega\cdot\nabla\psi_g^+(\vec{r},\Omega,t)d\Omega\,. \tag{2.12}$$

Integrating Eq. (2.10) over the angle and using the diffusion approximation, then the multigroup time-dependent diffusion equation derived from the even parity transport equation can be written as

$$\frac{1}{v_g}\frac{\partial}{\partial t}\phi_g(\vec{r},t)-\nabla\cdot D_g(\vec{r},t)\nabla\phi_g(\vec{r},t)+\Sigma_{tg}(\vec{r},t)\phi_g(\vec{r},t)=\ q_g(\vec{r},t)\,, \tag{2.13}$$

where

$$q_g(\vec{r},t)=\sum_{g'=1}^{G}\Sigma_{sg'g}(\vec{r},t)\phi_{g'}(\vec{r},t)+\lambda\chi_{pg}(\vec{r},t)\sum_{g'=1}^{G}\nu_p\Sigma_{fg'}(\vec{r},t)\phi_{g'}(\vec{r},t)+\sum_{k=1}^{K}\chi_{dkg}(\vec{r},t)\lambda_k C_k(\vec{r},t)\,. \tag{2.14}$$

The time-dependent delayed neutron precursor equation for stationary fuel can be obtained from Eq. (2.2) by setting the fuel velocity term to zero as

$$\frac{\partial}{\partial t}C_k(\vec{r},t)+\lambda_k C_k(\vec{r},t)=\lambda\psi_k(\vec{r},t),\quad k=1,2,\cdots,K\,, \tag{2.15}$$

where ψ_k the delayed neutron precursor generation rate of the delayed neutron precursor group k is given by

$$\psi_k(\vec{r},t)=\sum_{g'=1}^{G}\nu_{dk}\Sigma_{fg'}(\vec{r},t)\phi_{g'}(\vec{r},t)\,. \tag{2.16}$$

In the steady state analysis of stationary fuel, both time derivatives in Eq. (2.13) and Eq. (2.15) are set to zero. Then, Eq. (2.15) is reduced to

$$\lambda_k C_k(\vec{r})=\sum_{g'=1}^{G}\nu_{dk}\Sigma_{fg'}(\vec{r})\phi_{g'}(\vec{r})\,. \tag{2.17}$$

Since the delayed neutron precursors for stationary fuel decay at the same position as they are produced from fission, the prompt and delayed neutrons sources can be considered as a single source representing the total neutrons produced from fission. Thus, the fraction of the total fission neutrons χ_g emitted in group g can defined in terms of the fractions of the prompt χ_{pg} and delayed χ_{dkg} neutrons emitted in group g as

$$\chi_g(\vec{r})=\frac{\chi_{pg}(\vec{r})\sum_{g'=1}^{G}\nu_p\Sigma_{fg'}(\vec{r})\phi_{g'}(\vec{r})+\sum_{k=1}^{K}\chi_{dkg}(\vec{r})\sum_{g'=1}^{G}\nu_{dk}\Sigma_{fg'}(\vec{r})\phi_{g'}(\vec{r})}{\sum_{g'=1}^{G}\nu\Sigma_{fg'}(\vec{r})\phi_{g'}(\vec{r})}\,, \tag{2.18}$$

The steady state or time-independent multigroup diffusion equation for stationary fuel can be derived by setting the time derivatives in Eq. (2.13) to zero and using Eq. (2.17) for the delayed

neutron precursor source and the definition of the fraction of the total fission neutrons in Eq. (2.18) as

$$-\nabla\cdot D_g(\vec{r})\nabla\phi_g(\vec{r})+\Sigma_{tg}(\vec{r})\phi_g(\vec{r})=\sum_{g'=1}^{G}\Sigma_{sg'g}(\vec{r})\phi_{g'}(\vec{r})+\lambda\chi_g(\vec{r})\sum_{g'=1}^{G}\nu\Sigma_{fg'}(\vec{r})\phi_{g'}(\vec{r})\,. \tag{2.19}$$

2.1.1. Variational Nodal P_1 Solver

For the P_1 approximation, the variational functional for an internal node v can be written as

$$\begin{aligned}F_v[\phi_g,\chi_g^-]=&\int_v dV\left\{D_g(\vec{r})\nabla\phi_g(\vec{r})\cdot\nabla\phi_g(\vec{r})+\Sigma_{r,g}(\vec{r})\left(\phi_g(\vec{r})\right)^2-2\phi_g(\vec{r})S_g(\vec{r})\right\}\\&+2\sum_{\gamma=1}^{N_\Gamma}\int_\gamma d\Gamma\int d\Omega\left(\Omega\cdot\hat{n}_\gamma\right)\phi_g(\vec{r})\chi_{\gamma,g}(\vec{r},\Omega)\end{aligned}\,, \tag{2.20}$$

where g is the energy group index, γ is the node surface index, N_Γ is the number of surfaces, $\phi_g(\vec{r})$ is the zeroth moment of the even parity flux (i.e., the scalar flux), $\chi_{\gamma,g}(\vec{r},\Omega)$ is the first moment of the odd parity flux on surface γ, and $S_g(\vec{r})$ is the isotropic source. The functional in Eq. (2.20) is defined for each node such that it is dependent only upon each node's even parity zeroth moment flux ϕ_g and its odd parity boundary or interface flux χ_g^-. The change from ψ_g^- to χ_g^- was made to indicate that the odd parity flux along each node interface is unknown and that only the odd parity flux along the boundary of the problem domain is known via the boundary conditions [55, 57, 58]. Carrying out the angular integration in Eq. (2.20) yields

$$\begin{aligned}F_v[\phi_g,\chi_g]=&\int_v dV\left\{D_g(\vec{r})\nabla\phi_g(\vec{r})\cdot\nabla\phi_g(\vec{r})+\Sigma_{r,g}(\vec{r})\left(\phi_g(\vec{r})\right)^2-2\phi_g(\vec{r})S_g(\vec{r})\right\}\\&+2\sum_{\gamma=1}^{N_\Gamma}\sum_{\gamma}\int_\gamma d\Gamma\,\phi_g(\vec{r})j_{\gamma,g}(\vec{r})\end{aligned}\,, \tag{2.21}$$

where $j_{\gamma,g}(\vec{r})$ is the net surface current normal to surface γ. By determining orthogonal basis functions $f_i(\vec{r})$ for each node and $h_{\gamma,j}(\vec{r})$ for each surface γ, the scalar flux and the surface current can be expanded in terms of these basis functions as

$$\phi_g(\vec{r})=\sum_{i=1}^{N_V}f_i(\vec{r})\xi_{g,i}\,, \tag{2.22}$$

$$j_{\gamma,g}(\vec{r})=\sum_{j=1}^{N_s}h_{\gamma,j}(\vec{r})\chi_{\gamma,g,j}\,. \tag{2.23}$$

The source term can also be expanded in terms of $f_i(\vec{r})$ as

$$S_g(\vec{r}) = \sum_i f_i(\vec{r}) s_{g,i} \,. \tag{2.24}$$

The expansion coefficients can be written in a vector form as $\boldsymbol{\xi}_g$, $\boldsymbol{\chi}_{\gamma,g}$, and $\mathbf{s}_g$, respectively. The first elements of $\boldsymbol{\xi}_g$ and $\boldsymbol{\chi}_{\gamma,g}$ respectively represent the node-averaged scalar flux and the surface-averaged current as

$$\xi_{g,1} = \frac{1}{V_v}\int_v dV \phi_g(\vec{r}) f_1(\vec{r}) = \frac{1}{V_v}\int_v dV \phi_g(\vec{r}) = \bar{\phi}_g \,, \tag{2.25}$$

$$\chi_{\gamma,g,1} = \frac{1}{\Gamma_\gamma}\int_\gamma d\Gamma j_{\gamma,g}(\vec{r}) h_1(\vec{r}) = \frac{1}{\Gamma_\gamma}\int_\gamma d\Gamma j_{\gamma,g}(\vec{r}) = \bar{j}_{\gamma,g} \,, \tag{2.26}$$

where V_v is the volume of node v and Γ_γ is the area of surface γ. Assuming constant cross sections in each node and using the expansions in Eq. (2.22) and Eq. (2.23), Eq. (2.21) can be rewritten with the summation convention rule as

$$\begin{aligned} F_v[\phi_g, \chi_g] = {} & D_g \int_v dV \nabla_\omega f_i(\vec{r}) \xi_{g,i} \nabla_\omega f_j(\vec{r}) \xi_{g,j} + \Sigma_{r,g} \int_v dV f_i(\vec{r}) \xi_{g,i} f_j(\vec{r}) \xi_{g,j} \\ & -2\int_v dV f_i(\vec{r}) \xi_{g,i} f_j(\vec{r}) \xi_{g,j} s_{g,j} + 2\sum_\gamma \int_\gamma d\Gamma f_i(\vec{r}) \xi_{g,i} h_{\gamma,j}(\vec{r}) \chi_{\gamma,g,j} \end{aligned} \,, \tag{2.27}$$

where ω sums over *x*, *y*, and *z* for Cartesian, hexagonal, and triangular geometries and over *r*, θ, and *z* for cylindrical geometries. By requiring the reduced functional in Eq. (2.27) be stationary with respect to the variation of scalar flux, $\boldsymbol{\xi}_g$ can be obtained in terms of $\boldsymbol{\chi}_{\gamma,g}$, and $\mathbf{s}_g$. Projection of the node flux on a surface γ yields the surface flux expansion coefficient vector $\boldsymbol{\varphi}_{\gamma,g}$ in terms of $\boldsymbol{\xi}_g$. To obtain a response matrix in conventional form, the node surface flux and total current on surface γ can be obtained using the diffusion approximation as

$$\boldsymbol{\varphi}_{\gamma,g} = 2(\boldsymbol{\chi}^+_{\gamma,g} + \boldsymbol{\chi}^-_{\gamma,g}) \,, \tag{2.28}$$

$$\boldsymbol{\chi}_{\gamma,g} = \boldsymbol{\chi}^+_{\gamma,g} - \boldsymbol{\chi}^-_{\gamma,g} \,, \tag{2.29}$$

where $\boldsymbol{\chi}^+_{\gamma,g} = (\boldsymbol{\varphi}_{\gamma,g} + 2\boldsymbol{\chi}_{\gamma,g})/4$ and $\boldsymbol{\chi}^-_{\gamma,g} = (\boldsymbol{\varphi}_{\gamma,g} - 2\boldsymbol{\chi}_{\gamma,g})/4$ are the outgoing and incoming partial current moment vectors of surface γ, respectively. By further defining the partial current vectors $\mathbf{j}^\pm_g = \mathrm{col}[\boldsymbol{\chi}^\pm_{1,g}, \cdots, \boldsymbol{\chi}^\pm_{N_\Gamma,g}]$ for each node, where "col" denotes a column vector, and by requiring the

surface current to be continuous across each interface, a system of equations for $\boldsymbol{\xi}_g$ and $\mathbf{j}_g^{\pm}$ are obtained as

$$\boldsymbol{\xi}_g = \mathbf{H}_g \mathbf{s}_g - \mathbf{C}_g (\mathbf{j}_g^+ - \mathbf{j}_g^-), \tag{2.30}$$

$$\mathbf{j}_g^+ = \mathbf{R}_g \mathbf{j}_g^- + \mathbf{B}_g \mathbf{s}_g, \tag{2.31}$$

where $\mathbf{H}_g, \mathbf{C}_g$, $\mathbf{B}_g$, and $\mathbf{R}_g$ are the response matrices. Eq. (2.30) and Eq. (2.31) are solved iteratively with the power iteration method. At each power iteration, the total fission neutron source is calculated using the scalar flux solution from the previous power iteration. Then, the response matrix equations for partial current vectors in Eq. (2.31) are solved iteratively. Once the partial current vectors are determined, the scalar flux vectors are updated using Eq. (2.30). This iteration continues until the solution converged within the specified error criteria.

The basis functions for the within-node and surface flux distributions are obtained by orthogonalizing a set of monomials of local coordinate variables of which order is less than or equal to a user-specified value. The orthogonalization is performed using the Gram-Schmidt procedure. The resulting orthogonal basis functions $f_i(\vec{r})$ for a node and $h_{\gamma,j}(\vec{r})$ for a surface γ can be written as

$$f_i(\vec{r}) = \sum_p \sum_q \sum_w C_i^{pqw} r^p \theta^q z^w , \tag{2.32a}$$

$$h_{\gamma,j}(\vec{r}) = \sum_p \sum_q C_{\gamma,j}^{pq} \alpha^p \beta^q , \tag{2.32b}$$

where α and β are the local coordinate variables on surface γ. In Cartesian, hexagonal, and triangular geometries, the coefficients in Eq. (2.32) are the same for all the nodes because they have translational symmetry. In cylindrical geometries, however, there is no translational symmetry in the radial direction, and thus the nodal basis functions and the surface basis functions are uniquely generated for each radial node [53].

2.1.2. Extension to Flowing Fuel

For steady state analysis of stationary fuel, the neutron diffusion equation and the delayed neutron precursor equation were reduced into one equation with total fission neutron source utilizing the definition of the fraction of the total fission neutrons in Eq. (2.18), as shown in the previous section. However, the solution procedure is not valid for MSRs with flowing fuels. Instead, the delayed neutron precursor equations need to be solved explicitly to account for the

drift of the fuel salt in the core and the decay of the delayed neutron precursors in a different location than they were produced. By setting the time derivatives in Eq. (2.13) and Eq. (2.2) to zero and including a convection term to represent the drift of delayed neutron precursors for a given velocity field, the steady state neutron diffusion equation for flowing fuel can be written as

$$\begin{aligned} -\nabla \cdot D_g(\vec{r})\nabla\phi_g(\vec{r}) + \Sigma_{tg}(\vec{r})\phi_g(\vec{r}) &= \sum_{g'=1}^{G}\Sigma_{sg'g}(\vec{r})\phi_{g'}(\vec{r}) \\ &+ \lambda\chi_{pg}(\vec{r})\sum_{g'=1}^{G}\nu_p\Sigma_{fg'}(\vec{r})\phi_{g'}(\vec{r}) + \sum_{k=1}^{K}\chi_{dkg}(\vec{r})\lambda_k C_k(\vec{r}), \quad g=1,2,\cdots,G, \end{aligned} \tag{2.33}$$

$$\nabla\cdot[\vec{u}(\vec{r})C_k(\vec{r})] + \lambda_k C_k(\vec{r}) = \lambda\sum_{g'=1}^{G}\nu_{dk}\Sigma_{fg'}(\vec{r})\phi_{g'}(\vec{r}), \quad k=1,2,\cdots,K\,, \tag{2.34}$$

where the term $\nabla\cdot[\vec{u}(\vec{r})C_k(\vec{r})]$ accounts for the drift of the delayed neutron precursors, and $\vec{u}(\vec{r})$ is the fuel salt velocity at position $\vec{r}$. In order to solve Eq. (2.34) simultaneously with Eq. (2.33), it should be accompanied with a boundary condition to preserve the total amount of the delayed neutron precursors and to account for the decay outside the core region. The inlet precursor concentrations are calculated by assuming that precursors leaving the core are uniformly mixed and flow back to the core region except for those that decay outside the core region. Thus, the boundary condition at the core inlet is evaluated as

$$C_k(r,0) = \frac{\int_{A_{out}} dAu(r,H)C_k(r,H)e^{-\lambda_k\tau}}{\int_{A_{in}} dAu(r,0)}\,. \tag{2.35}$$

where the position vector $\vec{r}$ is separated into the radial r and axial z variables, H is the core height, A_{in} is the core inlet flow area, A_{out} is the core outlet flow area, and τ is the time spent outside of the core by the precursors. For an axial velocity field, Eq. (2.34) can be solved analytically or numerically within each node. However, considering a general velocity field, Eq. (2.34) cannot be solved analytically, instead, it can be solved numerically using the upwind finite difference method (FDM) for spatial discretization as

$$\frac{u_{r,ij}C_{k,ij} - u_{r,i-1,j}C_{k,i-1,j}}{\Delta r_i} + \frac{u_{z,ij}C_{k,ij} - u_{z,i,j-1}C_{k,i,j-1}}{\Delta z_j} + \lambda_k C_{k,ij} = \lambda\psi_{k,ij}\,, \tag{2.36}$$

$$\psi_{k,ij} = \sum_{g'=1}^{G}\nu_{dk}\Sigma_{fg',ij}\phi_{g',ij}\,, \tag{2.37}$$

where u_r and u_z, are the radial and axial components of the fuel salt velocity, respectively. $\Delta z_j = z_j - z_{j-1}$ and $\Delta r_i = r_i - r_{i-1}$ are the axial and radial widths of node ij, respectively. i and j,

are the radial and axial indexes of node ij, respectively. At each power iteration, Eq. (2.36) is solved iteratively with the Gauss-Seidel method considering the axial and radial velocity fields. The main advantage of this scheme is it accounts for the radial flow between nodes by considering the radial and axial component of the velocity field. In this upwind FDM approach, the average delayed neutron precursor concentrations in a node are assumed to be equal to the outgoing delayed neutron precursor concentrations for that node. Considering an axial velocity field $u(\vec{r}) = u(r,z)\vec{e}_z$, Eq. (2.34) is reduced to

$$\frac{\partial}{\partial z}[u(r,z)C_k(r,z)] + \frac{\lambda_k}{u(r,z)}[u(r,z)C_k(r,z)] = \lambda\psi_k(r,z), \quad k = 1,2,\cdots,K\,, \tag{2.38}$$

$$\psi_k(r,z) = \sum_{g'=1}^{G} \nu_{dk}\Sigma_{fg'}(r,z)\phi_{g'}(r,z)\,. \tag{2.39}$$

In this form, Eq. (2.38) can be solved analytically for $C_k(r,z)$ in terms of the inlet precursor concentration and considering the intranodal flux distribution as

$$u(r,z)C_k(r,z) = u(r,z_0)C_k(r,z_0)e^{-\lambda_k\int_{z_0}^{z}\frac{dz'}{u(r,z')}} + \lambda\int_{z_0}^{z} dz'\psi_k(r,z')e^{\lambda_k\int_{z_0}^{z'}\frac{dz''}{u(r,z'')}}\,, \tag{2.40}$$

Therefore, the outgoing and average delayed neutron precursor concentrations of the j^{th} axial node can be obtained as

$$C_{k,ij} = \frac{u_{i,j-1}}{u_{ij}}e^{-\lambda_k\Delta t_{ij}}C_{k,i,j-1} + \frac{\lambda}{u_{ij}}\int_{z_{j-1}}^{z_j} dz'\psi_k(\overline{r}_i,z')e^{\lambda_k(z'-z_j)/\overline{u}_{ij}}\,, \tag{2.41}$$

$$\overline{C}_{k,ij} = \frac{u_{i,j-1}}{\overline{u}_{ij}}\frac{1}{\lambda_k\Delta t_{ij}}(1-e^{-\lambda_k\Delta t_{ij}})C_{k,i,j-1} + \frac{\lambda}{u_{ij}\Delta z_j}\int_{z_{j-1}}^{z_j} dz\int_{z_{j-1}}^{z} dz'\psi_k(\overline{r}_i,z')e^{\lambda_k(z'-z)/\overline{u}_{ij}}\,, \tag{2.42}$$

where

$$\Delta t_{ij} = \frac{\Delta z_j}{\overline{u}_{ij}} = \int_{z_{j-1}}^{z_i}\frac{dz}{u(r,z)}\,. \tag{2.43}$$

Also, Eq. (2.38) can be solved with FDM approach as

$$C_{k,ij} = \frac{1}{\overline{u}_{ij}/\Delta z_j + \lambda_k}\left(\overline{u}_{ij-1}/\Delta z_j\, C_{k,ij-1} + \lambda\psi_{k,ij}\right). \tag{2.44}$$

The delayed neutron precursor distributions can be calculated for a given fission source distribution using Eq. (2.36) for a general velocity field or using Eq. (2.42) or Eq. (2.44) for an axial velocity field along with the boundary condition Eq. (2.35). Thus, the delayed neutron source is calculated, and the total fission neutron source can be obtained as a summation of prompt and

delayed fission neutron sources as

$$S_{g,ij} = \lambda \chi_{pg,ij} \sum_{g'=1}^{G} \nu_p \Sigma_{fg',ij} \phi_{g',ij} + \sum_{k=1}^{K} \chi_{dkg} \lambda_k C_{k,ij} \ . \tag{2.45}$$

At each power iteration, delayed neutron precursor concentrations are determined using the scalar flux solution from the previous iteration. Thus, the total fission neutron source is calculated from prompt and delayed fission neutron sources as in Eq. (2.45). Then, the response matrix equations for partial current vectors in Eq. (2.31) are solved iteratively using the updated fission source. The last step is to calculate the scalar flux vectors using Eq. (2.30) with updated partial current vectors and the updated fission source. This iteration scheme continues until the solution converged within the specified error criteria.

2.1.3. Adjoint Flux Solution for Flowing Fuel

In order to calculate kinetics parameters and delayed neutron losses of MSRs with flowing fuel accurately, a steady state adjoint flux calculation capability is added to PROTEUS-NODAL. As discussed in the previous section, the distribution of the delayed neutron precursor concentrations is very important in calculating the total neutron fission source of flowing fuel. The shifted delayed neutron precursor distribution and their decay outside the reactor core will result in significant losses of the delayed neutrons and as a result a smaller effective delayed neutron fraction. However, the definitions of the kinetic parameters for flowing fuel are defined by including the delayed neutron source from the decay of the precursors explicitly and using the adjoint solution will be used as an importance weighting function.

For flowing fuel, the adjoint system of equations to the coupled steady state neutron diffusion equation and the delayed neutron precursor concentration equation as in Eq. (2.33) and Eq. (2.34) can be obtained as [59]

$$\begin{aligned} -\nabla \cdot D_g(\vec{r}) \nabla \phi_g^*(\vec{r}) + \Sigma_{tg}(\vec{r}) \phi_g^*(\vec{r}) &= \sum_{g'=1}^{G} \Sigma_{sgg'}(\vec{r}) \phi_{g'}^*(\vec{r}) \\ + \lambda \nu_p \Sigma_{fg}(\vec{r}) \sum_{g'=1}^{G} \chi_{pg'}(\vec{r}) \phi_{g'}^*(\vec{r}) &+ \sum_{k=1}^{K} \nu_{dk} \Sigma_{fg}(\vec{r}) C_k^*(\vec{r}), \quad g = 1,2,\cdots,G, \end{aligned} \tag{2.46}$$

$$-\nabla \cdot [\vec{u}(\vec{r}) C_k^*(\vec{r})] + \lambda_k C_k^*(\vec{r}) = \lambda \psi_k^*(\vec{r}), \quad k = 1,2,\cdots,K, \tag{2.47}$$

where

$$\psi_k^*(\vec{r}) = \sum_{g'=1}^{G} \chi_{dkg'}(\vec{r}) \phi_{g'}^*(\vec{r}) \, . \tag{2.48}$$

The negative sign in the adjoint precursor convection term $\nabla\cdot[\vec{u}(\vec{r})C_k^*(\vec{r})]$ in Eq. (2.47) is to make the bilinear concomitant term zero. Thus, an initial value problem becomes a final value problem, and Eq. (2.47) needs to be solved in the opposite direction (from the outlet to the inlet) with a boundary condition at the core outlet that can be represented as

$$C_k^*(r,H)=\frac{\int_{A_{in}} dAu(r,0)C_k^*(r,0)e^{-\lambda_k\tau}}{\int_{A_{out}} dAu(r,H)}, \tag{2.49}$$

where the position vector $\vec{r}$ is separated into the radial and axial variables. Similar to the forward delayed precursor concentration equation in Eq. (2.34), the adjoint delayed precursor concentration equation in Eq. (2.47) can be solved numerically with the FDM approach for general or axial velocity fields or analytical for an axial velocity field. Considering a general velocity field with the FDM approach, Eq. (2.47) can be solved as

$$\frac{u_{r,ij}C_{k,ij}^*-u_{r,i+1,j}C_{k,i+1,j}^*}{\Delta r_i}+\frac{u_{z,ij}C_{k,ij}^*-u_{z,i,j+1}C_{k,i,j+1}^*}{\Delta z_j}+\lambda_k C_{k,ij}^*=\lambda\psi_{k,ij}^*, \tag{2.50}$$

For an axial velocity field, the adjoint precursor concentration equation can be solved analytically in terms of the adjoint precursor concentration at the outlet as

$$u(r,z)C_k^*(r,z)=u(r,H)C_k^*(r,H)e^{-\lambda_k\int_z^H\frac{dz'}{u(r,z')}}+\lambda\int_z^H dz'\psi_k^*(r,z')e^{-\lambda_k\int_z^{z'}\frac{dz''}{u(r,z'')}}, \tag{2.51}$$

or with FDM approach as

$$C_{k,ij}^*=\frac{1}{\bar{u}_{ij}/\Delta z_j+\lambda_k}\left(\bar{u}_{ij+1}/\Delta z_j\, C_{k,ij+1}^*+\lambda\psi_{k,ij}^*\right). \tag{2.52}$$

The solution procedure applied to determine the forward solution of the neutron scalar flux and the delayed neutron precursor concentration can be used to obtain the adjoint solution as discussed in the previous section.

2.1.4. Evaluation of Kinetics Parameters

Using the forward and adjoint flux solutions, the effective delayed neutron fraction and the prompt neutron generation time for circulating fuel can be calculated as [59, 60]

$$\beta_{circ}=\frac{\sum_i V_i\sum_{g=1}^{G}\phi_{g,i}^*\sum_{k=1}^{K}\chi_{dkg}\lambda_k C_{k,i}}{\sum_i V_i\sum_{g=1}^{G}\phi_{g,i}^*\chi_{pg,i}\sum_{g'=1}^{G}\nu_p\Sigma_{fg',i}\phi_{g',i}+\sum_i V_i\sum_{g=1}^{G}\phi_{g,i}^*\sum_{k=1}^{K}\chi_{dkg,i}\lambda_k C_{k,i}}, \tag{2.53}$$

$$\Lambda_{circ} = \frac{\sum_i V_i \sum_{g=1}^{G} \phi_{g,i}^{*} v_g^{-1} \phi_{g,i}}{\sum_i V_i \sum_{g=1}^{G} \phi_{g,i}^{*} \chi_{pg,i} \sum_{g'=1}^{G} \nu_p \Sigma_{fg',i} \phi_{g',i} + \sum_i V_i \sum_{g=1}^{G} \phi_{g,i}^{*} \sum_{k=1}^{K} \chi_{dkg,i} \lambda_k C_{k,i}}, \tag{2.54}$$

where the subscript i denotes spatial nodes, and V_i is the volume of node i. The derivation of the kinetics parameters and point kinetics equations of flowing fuel is provided in Appendix B. In the denominator of Eq (2.53) and Eq. (2.54), the first term represents the total importance of prompt neutrons, and the second term represents the total importance of delayed neutrons. In MSRs, the delayed neutron production by precursor decay and the precursor production from fission should be differentiated since delayed neutron precursors move to other positions before they decay.

For a stationary fuel, delayed neutron precursors decay at the same position where they are produced, and thus the number of delayed neutrons produced at a position is equal to the number of precursors produced by fission. Therefore, the total number of neutrons can be determined from the quasi-stationary total fission neutrons. As a result, the effective delayed neutron fraction and the prompt neutron generation time can be determined as

$$\beta_{eff} = \frac{\sum_i V_i \sum_{g=1}^{G} \phi_{g,i}^{*} \sum_{k=1}^{K} \chi_{dkg,i} \sum_{g'=1}^{G} \nu_{dk} \Sigma_{fg',i} \phi_{g',i}}{\sum_i V_i \sum_{g=1}^{G} \phi_{g,i}^{*} \chi_{g,i} \sum_{g'=1}^{G} \nu \Sigma_{fg',i} \phi_{g',i}}, \tag{2.55}$$

$$\Lambda = \frac{\sum_i V_i \sum_{g=1}^{G} \phi_{g,i}^{*} v_g^{-1} \phi_{g,i}}{\sum_i V_i \sum_{g=1}^{G} \phi_{g,i}^{*} \chi_{g,i} \sum_{g'=1}^{G} \nu \Sigma_{fg',i} \phi_{g',i}}. \tag{2.56}$$

The difference between the effective delayed neutron fraction values in Eq. (2.55) and Eq. (2.53) represents the reactivity loss due to the fuel drift as $\beta_{loss} = \beta_{eff} - \beta_{circ}$.

2.2. Transient Fixed Source Problem

In this section, the formulation of the transient fixed source problem (TFSP) for stationary and flowing fuels is discussed, and the solution procedure with verification tests are provided for each case.

2.2.1. Stationary Fuel

For stationary fuel, the time-dependent multigroup neutron diffusion equation and the delayed neutron precursor concentration equations are defined in Eq. (2.13) and Eq. (2.15), respectively. Using the implicit Euler method, Eq. (2.13) can be temporarily discretized as

$$\frac{\phi_g^n(\vec{r})-\phi_g^{n-1}(\vec{r})}{v_g\Delta t_n}-\nabla\cdot D_g^n(\vec{r})\nabla\phi_g^n(\vec{r})+\Sigma_{rg}^n(\vec{r})\phi_g^n(\vec{r})=q_g^n(\vec{r}),\qquad n=1,2,\cdots,N\,, \tag{2.57}$$

where $\Delta t_n = t_n - t_{n-1}$ is the time-step size of the n_{th} time point. By moving the time derivative term to the right-hand side, Eq. (2.57) can be written as a TFSP for given delayed neutron precursor concentrations as

$$-\nabla\cdot D_g^n(\vec{r})\nabla\phi_g^n(\vec{r})+\Sigma_{rg}^n(\vec{r})\phi_g^n(\vec{r})=q_g'^n(\vec{r})\,, \tag{2.58}$$

where

$$\begin{aligned} q_g'^n(\vec{r}) = {} & \frac{\phi_g^{n-1}(\vec{r})-\phi_g^n(\vec{r})}{v_g\Delta t_n}+\sum_{g'\neq g}^{G}\Sigma_{sg'g}^n(\vec{r})\phi_{g'}^n(\vec{r}) \\ & +\lambda\chi_{pg}(\vec{r})\sum_{g'=1}^{G}\nu_p\Sigma_{fg'}^n(\vec{r})\phi_{g'}^n(\vec{r})+\sum_{k=1}^{K}\chi_{dkg}(\vec{r})\lambda_k C_k^n(\vec{r}) \end{aligned}\,. \tag{2.59}$$

The time dependency of the delayed neutron precursor concentration equations can be treated as an explicit unknown to be solved simultaneously with the flux equation. Applying temporal discretization to the delayed neutron precursor concentration will results into a linear system with an additional six unknowns in each node. Instead, the delayed neutron precursor concentration equations are typically solved by applying the second-order analytic integration method. Based on this approach, the delayed fission source distribution is assumed to have a quadratic shape on time using the delayed fission sources at the current and the two previous time points. Then, Eq. (2.15) can be integrated analytically as [39]

$$\lambda_k C_k^n(\vec{r})=\lambda_k e^{-\lambda_k\Delta t_n}C_k^{n-1}(\vec{r})+\lambda\left[\Omega_k^{n-2}\psi_k^{n-2}(\vec{r})+\Omega_k^{n-1}\psi_k^{n-1}(\vec{r})+\Omega_k^n\psi_k^n(\vec{r})\right], \tag{2.60}$$

with $r_n = \Delta t_{n-1}/\Delta t_n$, the three parameters in Eq. (2.60) are given by

$$\Omega_k^{n-2}=\frac{1}{r_n(1+r_n)\lambda_k\Delta t_n}\left[\frac{2(1-e^{-\lambda_k\Delta t_n})}{\lambda_k\Delta t_n}-(1+e^{-\lambda_k\Delta t_n})\right], \tag{2.61a}$$

$$\Omega_k^{n-1}=\frac{1}{r_n\lambda_k\Delta t_n}\left[1+e^{-\lambda_k\Delta t_n}+(1-e^{-\lambda_k\Delta t_n})\left(r_n-\frac{2}{\lambda_k\Delta t_n}\right)\right]-e^{-\lambda_k\Delta t_n}\,, \tag{2.61b}$$

$$\Omega_k^n=1-\frac{2}{(1+r_n)\lambda_k\Delta t_n}+\frac{1-e^{-\lambda_k\Delta t_n}}{(1+r_n)\lambda_k\Delta t_n}\left(\frac{2}{\lambda_k\Delta t_n}-r_n\right). \tag{2.61c}$$

At each time point n, once the group source $q_g'^n(\vec{r})$ in Eq. (2.59) is calculated, the solution for the group flux in Eq. (2.58) is obtained by solving the response matrix equations in Eq. (2.30) and Eq. (2.31) using the P_1 steady state solver. Since the within-group flux is included in the group source in Eq. (2.58), a few inner iterations are required between Eq. (2.30) and Eq. (2.31).

The developed P_1 transient solver in PROTEUS-NODAL for stationary fuel is first verified using the modified TWIGL benchmark problem with hexagonal geometry [46]. The Doppler feedback effects were taken into account using the provided sets of cross sections without invoking the thermal hydraulics calculations. Figure 2.1 compares the calculation results of the P_1 solution with reference solution obtained from existing SP_3 solver of PROTEUS-NODAL. The P_1 solution agrees very well with the reference solution within 1% error which is related to the transport effect. As the diffusion approximation overestimates leakage in general, the P_1 transient solver yields a 1% smaller peak power than the SP_3 solver.

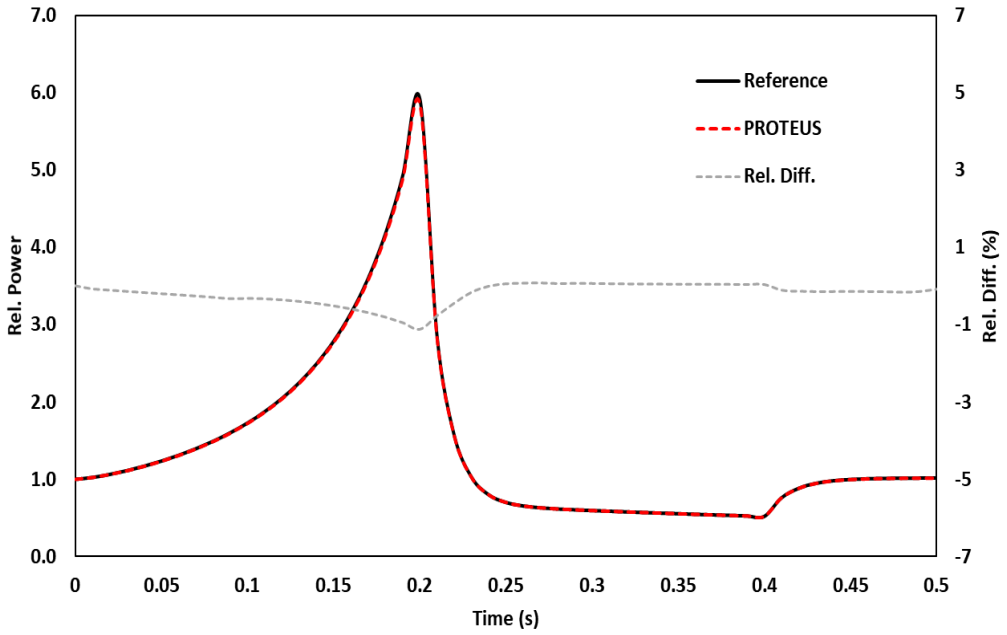


Figure 2.1. Power change with time for the modified TWIGL benchmark problem.

Another transient test was performed for a fast spectrum problem with +100 pcm step reactivity insertion without thermal feedback. Figure 2.2 shows the power changes during 1.0 s for the positive reactivity insertion. The P_1 transient solution agrees well with the reference transient result with about 1% difference is observed during the prompt jump period and the error remains within 1.5% for the same reasoning as in the TWIGL benchmark.

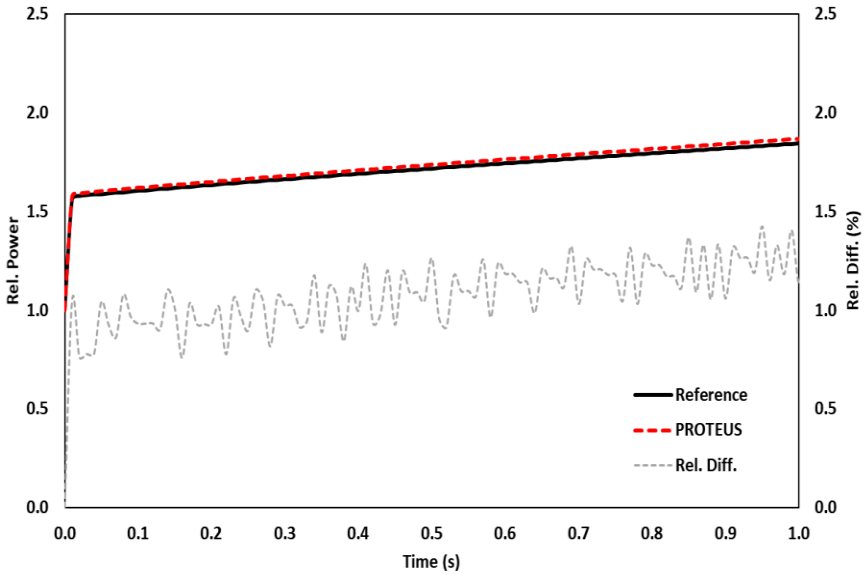


Figure 2.2. Power comparison after the positive step reactivity insertion.

2.2.2. Flowing Fuel

For flowing fuel, the drift of delayed neutron precursors is included in the precursor balance equation in Eq. (2.2) as a convection term as

$$\frac{\partial}{\partial t}C_k(\vec{r},t)+\nabla\cdot[\vec{u}(\vec{r},t)C_k(\vec{r},t)]+\lambda_k C_k(\vec{r},t)=\lambda\psi_k(\vec{r},t)\,, \tag{2.62}$$

where the delayed neutron source $\psi_k(\vec{r},t)$ is provided in Eq. (2.16). The delayed neutron precursor concentrations equations are discretized using the backward Euler scheme for temporal discretization and the upwind FDM for spatial discretization as

$$\frac{C^n_{k,ij}-C^{n-1}_{k,ij}}{\Delta t_n}+\frac{u^n_{r,ij}C^n_{k,ij}-u^n_{r,i-1,j}C^n_{k,i-1,j}}{\Delta r_i}+\frac{u^n_{z,ij}C^n_{k,ij}-u^n_{z,i,j-1}C^n_{k,i,j-1}}{\Delta z_j}+\lambda_k C^n_{k,ij}=\lambda\psi^n_{k,ij}\,, \tag{2.63}$$

For each time step, Eq. (2.63) is solved iteratively with the Gauss-Seidel method considering the axial and radial velocity fields. This scheme provides an unconditionally stable solution with no restrictions on the time step size, and it preserves the initial delayed neutron precursor concentrations since the equation is solved in the same way for both steady state and transient analyses. The inlet precursor concentrations are calculated by assuming that precursors leaving the core are uniformly mixed and flow back to the core except for those that decay outside the core. Thus, the boundary condition at the core inlet is evaluated as

$$C_k\left(r,0,t\right)=\frac{\int_{A_{out}}dAu(r,H,t-\tau)C_k(r,H,t-\tau)e^{-\lambda_k\tau}}{\int_{A_{in}}dAu(r,0,t)}\,. \tag{2.64}$$

The precursor concentration at the core outlet is assumed to be equal to the average precursor concentration of the last node in the core (upwind scheme) or it can be evaluated using the linear extrapolation of the last two nodes in the core. For a fast spectrum MSR with flowing fuel, it would be important to consider the radial velocity field as in Eq. (2.63) for transients in which the fuel velocity is decreasing as in the loss of flow accident scenario. However, in moderated MSRs, the fuel is flowing axially in channels, so Eq. (2.62) can be rewritten considering an axial velocity field $\vec{u}(\vec{r}) = u(r,z)\vec{e}_z$ as

$$\frac{\partial}{\partial t} C_k(r,z,t) + \frac{\partial}{\partial z}[u(r,z,t)C_k(r,z,t)] + \lambda_k C_k(r,z,t) = \lambda \psi_k(r,z,t), \tag{2.65}$$

In order to solve Eq. (2.62), three different ways are investigated. The first approach is the FDM in which the delayed neutron precursor equation is discretized in both time and space in a similar way as in Eq. (2.63). The second approach is the analytical scheme which solves the partial differential equation (PDE) using the method of characteristics (MOC). The third approach is the semi-analytical scheme or the so-called method of lines (MoL). The basis of this method relies on converting the partial differential equation into an ordinary differential equation (ODE) and solving it using a proper ODE solver. Test results showed that only the FDM produces very stable and correct results of the null transient. The MOC and MoL solutions diverged from the steady state solution, although the power deviations were only fractions of a percent. Thus, only the FDM is discussed in this section. The MOC and MoL are summarized in Appendix A, along with some comparison of results. Applying the FDM approach with the backward Euler scheme for temporal discretization and the upwind finite difference scheme for spatial discretization, Eq. (2.65) becomes

$$\frac{C^n_{k,ij} - C^{n-1}_{k,ij}}{\Delta t_n} + \frac{u^n_{ij} C^n_{k,ij} - u^n_{ij-1} C^n_{k,ij-1}}{\Delta z_j} + \lambda_k C^n_{k,ij} = \lambda \psi^n_{k,ij}, \tag{2.66}$$

Solving for $C^n_{k,ij}$ yields

$$C^n_{k,ij} = \left(1 + \lambda_k \Delta t_n + \frac{u^n_{ij} \Delta t_n}{\Delta z_j}\right)^{-1} \left[\frac{u^n_{ij-1} \Delta t_n}{\Delta z_j} C^n_{k,ij-1} + \lambda \Delta t_n \psi^n_{k,ij}\right]. \tag{2.67}$$

At each time step *n*, once the delayed neutron precursor concentrations are calculated using Eq. (2.63) for general velocity field or using Eq. (2.66) for an axial velocity field, the group source $q'^n_g(\vec{r})$ in Eq. (2.59) is obtained and the TFSP for flowing fuel can be formulated in similar way

as in stationary fuel. Then, the solution for the group flux in Eq. (2.58) is obtained by solving the response matrix equations in Eq. (2.30) and Eq. (2.31) using the P_1 solver.

In order to verify the transient solver of flowing fuel and the solution of the delayed neutron precursors concentration equations, transient tests were performed by inserting positive and negative reactivities through adjusting the fuel salt velocity in the core region. An increased fuel velocity as in pump startup transient increases the loss of delayed neutrons and thus decreases the reactivity. On the contrary, a decreased fuel velocity as in the pump coast down transient increases the reactivity due to a decreased loss of delayed neutrons. Figure 2.3 shows the power evolutions of the pump startup transients to different final velocities, and power evolutions of the pump coast down transients from different initial velocities. In the pump startup transient, the power decreases more rapidly with increasing the fuel salt velocity because of the increased delayed neutron losses with the increased fuel salt velocity. The oscillatory behavior of the power is due to the undecayed delayed neutron precursors decay in the core region. In the pump coast down transient, the power increases monotonically, and the rate of increase depends on the initial fuel salt velocity.

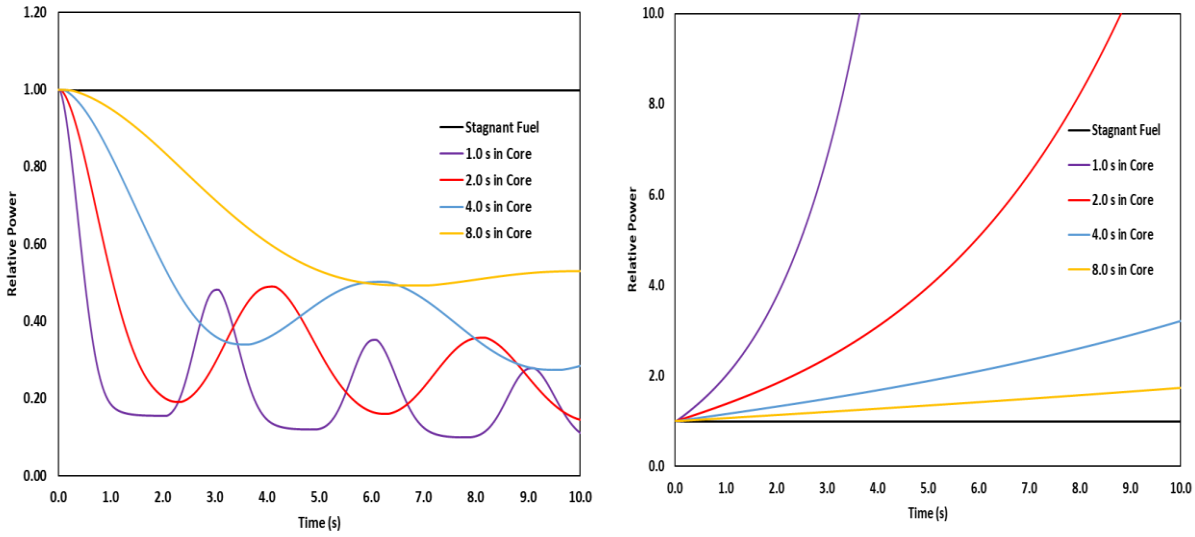


Figure 2.3. Power evolution after the step increase (left) and step decrease (right) of the fuel velocity.

Figure 2.4 shows the delayed neutron distribution in the core region for a pump startup transient during the first 10.0 s. The velocity was increased from zero to its final value exponentially with a time constant of 1.0 s and the total fuel salt transit time of 4.0 s. The power decreases initially due to the decay of the delayed neutron precursors outside the core region. Then the power increases at 3.0 s as the undecayed precursors come back and decay at the core region in oscillatory behavior.

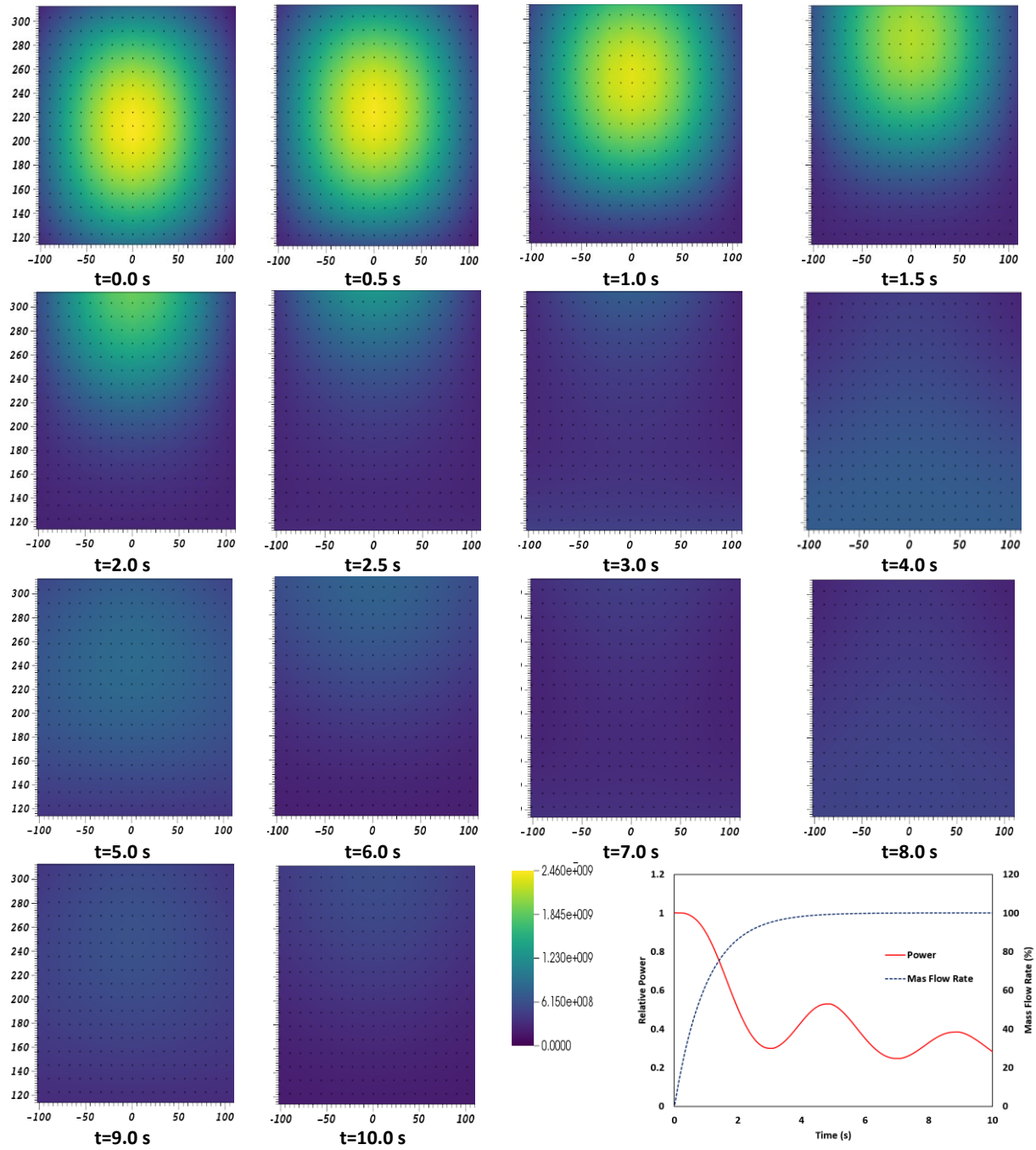


Figure 2.4 Delayed neutrons distribution in the core during pump-start-up transient.

2.3. CMFD Acceleration

The coarse mesh finite difference (CMFD) acceleration scheme has been implemented in the PROTEUS-NODAL code to reduce the computational time of the steady state and the transient analyses. The CMFD solver is implemented consistently with the steady state and transient solvers, and it is called at every power or outer iteration during the calculations.

2.3.1. Acceleration Scheme

The CMFD acceleration method can be applied to the TFSP in a similar way as in the eigenvalue problem [54, 61]. In the CMFD acceleration method, the neutron flux preserving interfacial net currents determined from a higher-order calculation is sought for a given source from the previous time step. To preserve the net current, the current correction coefficients are obtained for all surfaces and groups as

$$\hat{D}^n_{g,i\to u} = \frac{-\left(J^n_{g,i\to u}\right)^{nodal} + \tilde{D}^n_{g,i\to u}\left(\phi^{i,n}_g - \phi^{u,n}_g\right)^{nodal}}{\left(\phi^{i,n}_g + \phi^{u,n}_g\right)^{nodal}}, \tag{2.68}$$

where $\tilde{D}^n_{g,i\to u}$ is the coupling coefficient between nodes *i* and *u* of the conventional finite difference diffusion formulation. In the steady state analysis, the convergence rate is governed by the dominance ratio of the CMFD matrix. On the other hand, the convergence rate of the transient fixed source problem is governed by the spectral radius of the matrix, which is very close to unity for most of the transient cases and the solution will converge very slowly. In the implementation of CMFD acceleration for transient analysis, the CMFD linear system is formed by applying the TFSP with the scattering, the prompt fission, and the delayed neutron sources are moved to the left-hand side of the CMFD equation in each time step. The right-hand side of the CMFD equation has the fixed source terms, which consist of the previous time step contributions. Then, the following CMFD equations for the TFSP are constructed by moving all the terms containing quantities at the current time step to the left-hand side as

$$\begin{aligned} &-\sum_u A_{iu}(\tilde{D}^n_{g,i\to u} + \hat{D}^n_{g,i\to u})\phi^{u,n}_g + \left[\sum_u A_{iu}(\tilde{D}^n_{g,i\to u} - \hat{D}^n_{g,i\to u}) + \left(\Sigma^{n,i}_{r,g} + \frac{1}{v_g \Delta t_n}\right)V_i\right]\phi^{i,n}_g \\ &-q^{i,n}_g V_i = q^{i,n-1}_g V_i \end{aligned}, \tag{2.69}$$

where A_{iu} is the area of the interface between nodes *i* and *u*, V_i is the volume of node i, and

$$q^{i,n}_g = \sum_{g'\neq g}^{G} \Sigma^{i,n}_{sg'g}\phi^{i,n}_{g'} + \lambda\chi^i_{pg}\sum_{g'=1}^{G}\nu_p\Sigma^{i,n}_{fg'}\phi^{i,n}_{g'} + \sum_k \chi^i_{dkg}\Omega^n_k\psi^{i,n}_k, \tag{2.70}$$

$$q^{i,n-1}_g = \sum_k \chi^i_{dkg}\left\{\lambda_k e^{-\lambda_k\Delta t_n}C^{i,n-1}_k + \lambda\left(\Omega^{n-2}_k\psi^{i,n-2}_k + \Omega^{n-1}_k\psi^{i,n-1}_k\right)\right\} + \frac{1}{v_g\Delta t_n}\phi^{i,n-1}_g. \tag{2.71}$$

For flowing fuel, the delayed neutron precursor concentration equation is solved with the FDM approach and not with the analytical integration method as for stationary fuel. Thus, the group sources can be written as

$$q_g^{i,n} = \sum_{g' \neq g}^{G} \Sigma_{sg'g}^{i,n} \phi_{g'}^{i,n} + \lambda \chi_{pg}^{i} \sum_{g'=1}^{G} \nu_p \Sigma_{fg'}^{i,n} \phi_{g'}^{i,n} + \sum_k \chi_{dkg}^{i} \lambda_k C_k^{i,n}, \qquad q_g^{i,n-1} = \frac{1}{v_g \Delta t_n} \phi_g^{i,n-1}. \tag{2.72}$$

The entire linear system of equations for all energy groups is formed and the resulting fixed source problem is solved using the Krylov subspace (GMRES) method with no source iteration. After the execution of the CMFD acceleration, the scalar neutron flux is updated using the CMFD flux solution. In order to avoid any instabilities due to the CMFD acceleration, the under-relaxation scheme was added to the solution process of the CMFD acceleration. Then, the scalar flux of the nodal flux solution is updated as

$$\phi_g^{m,n}\Big|_{nodal} = \omega \phi_g^{m,n}\Big|_{CMFD} + (1-\omega)\phi_g^{m-1,n}\Big|_{nodal}, \tag{2.73}$$

where ω is the relaxation parameter, and m is the index for the outer iteration. For steady state calculations, the CMFD acceleration equation can be obtained directly from Eq. (2.69) by setting the discretized time derivatives to zero and moving the scattering and fission sources to the right-hand side of the equation as

$$\begin{aligned} &-\sum_u A_{iu}(\tilde{D}_{g,i\to u} + \hat{D}_{g,i\to u})\phi_g^u + \left[\sum_u A_{iu}(\tilde{D}_{g,i\to u} - \hat{D}_{g,i\to u}) + \Sigma_{r,g}^i V_i\right]\phi_g^i \\ &= V_i \sum_{g' \neq g}^{G} \Sigma_{sg'g}^i \phi_{g'}^i + \lambda \chi_g^i V_i \sum_{g'=1}^{G} \nu\Sigma_{fg'}^i \phi_{g'}^i \end{aligned}, \tag{2.74}$$

Equation (2.74) is solved is similar way as Eq. (2.69), and once the solution converged the nodal solution is updated accordingly.

2.3.2. Performance Testing

The performance of the CMFD acceleration was examined for steady state and transient calculations. For the steady state calculations, small and large fast spectrum problems were solved with and without the CMFD acceleration. The results are summarized in Table 2.1 for the eigenvalue, the number of outer iterations, and the computational time. Also, Fig. 2.5 compares the fission source error with number of the outer iterations for small and large core problems. The results show that the CMFD acceleration significantly reduces the number of outer iterations for both problems. However, the total computational time is almost similar to that of the calculation without the CMFD acceleration since the mesh size of the CMFD and NODAL calculations are the same and the CMFD implementation requires a longer computational time than that of the NODAL calculations.

Table 2.1. Performance of CMFD for Eigenvalue Problems

Problem	Small Core		Large Core	
No. Hexagonal Assemblies	19		625	
No. Energy Groups	9		33	
	w/o CMFD	w/ CMFD	w/o CMFD	w/ CMFD
Eigenvalue	1.000903	1.000903	1.022056	1.022056
Total Time (s)	0.972	0.815	287.15	304.55
Number of Outer Iterations	56	11	53	17

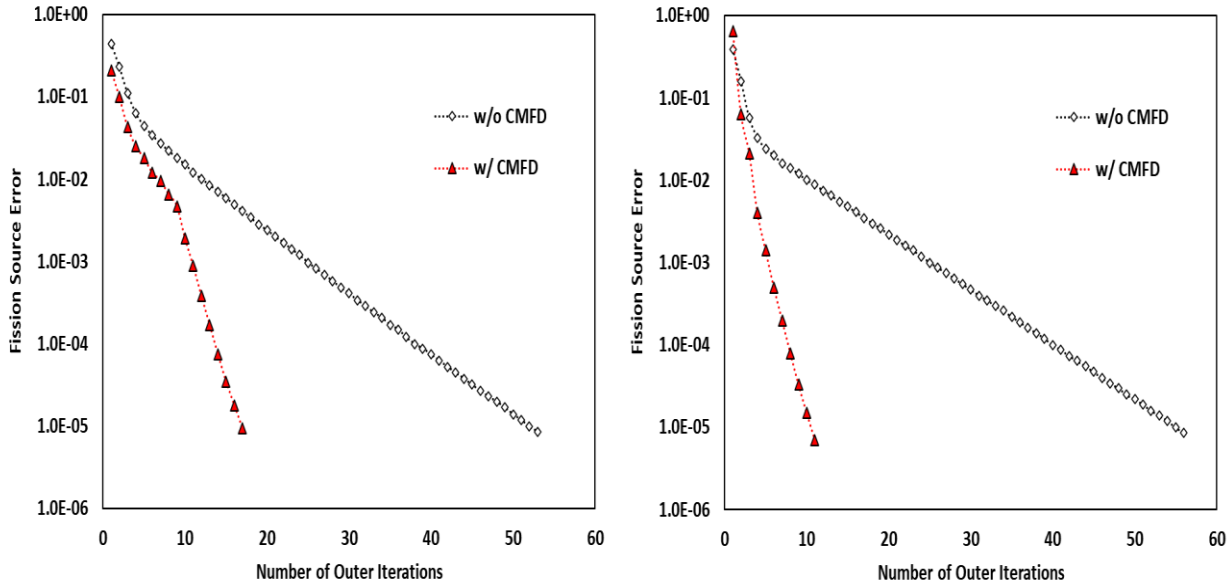


Figure 2.5. Comparison of error reduction in fission source with and without CMFD for eigenvalue problems of small (left) and large (right) cores.

For transient calculations, the modified TWIGL benchmark problem in hexagonal geometry was used to test the performance of the CMFD acceleration for transient analysis. The results of the test problem are presented in Fig. 2.6 by comparing the total computational time and number of outer iterations required for convergence at each time point. A significant reduction is achieved in both the number of outer iterations and the computational time, and the number of outer iterations is reduced by a factor of 20 to 60, and the computational time is reduced by a factor of 40. The reason for this large computational gain for transient calculations is due to the fact that the entire CMFD linear system is solved only once without source iterations.

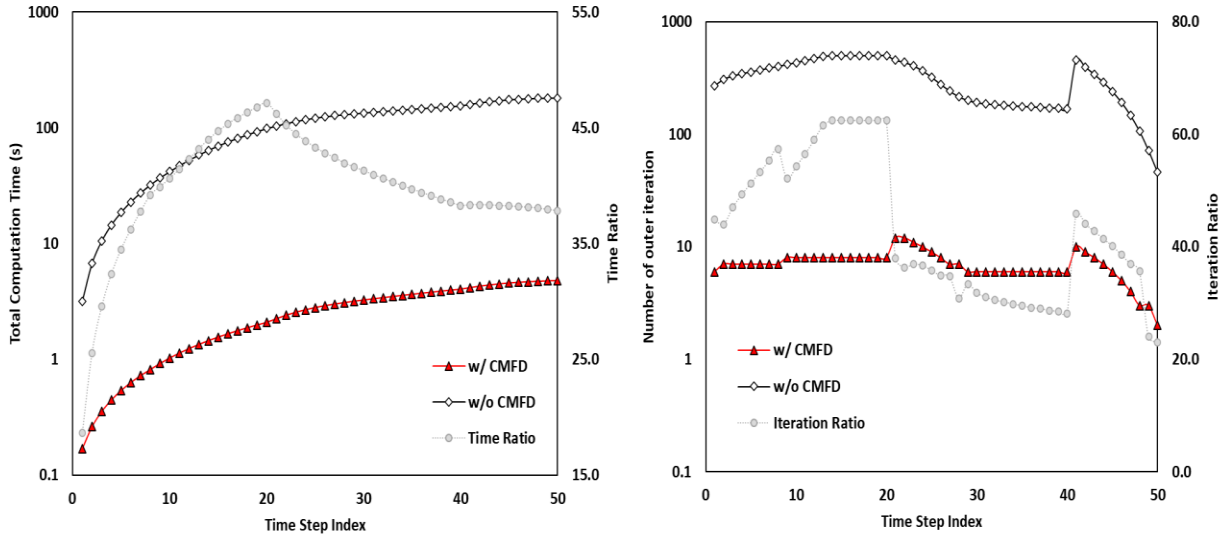


Figure 2.6. Comparison of the computational time (left) and the number of outer iterations (right) at each time step with and without CMFD for the modified TWIGL benchmark problem.

Also, the quasi-static methods were implemented into the PROTEUS-NODAL code in a consistent way to the TFSP solver for stationary and flowing fuels to reduce the computational time without a significant loss of accuracy. The quasi-static methods could speed up the solution by 3 to 10 times for most of the studied cases, but it failed to reproduce the reference solution for transients with fuel salt velocity changes significantly as in pump startup or pump overspeed transients. For this reason, the quasi-static methods were not used for transient analysis in the following chapters. However, the derivation and the solution algorithms of the quasi-static methods are included in Appendix B for stationary and flowing fuels with some test results.

2.4. Thermal Feedback Calculation

In order to perform coupled neutronics and thermal hydraulics calculations for steady state and transient analyses, the reactivity feedbacks due to the fuel salt temperature and density changes, and due to the moderator temperature change in reactors with solid moderator need to be considered in updating the multigroup cross sections. To account for thermal feedback, a thermal hydraulics solver dedicated to MSRs has been implemented in PROTEUS-NODAL. In this solver, the core is represented with parallel 1D flow channels, and the components outside the core are represented by simple lumped parameter models. Considering the drift of the fuel salt, the decay heat is calculated by solving the decay heat equations similar to the delayed neutron precursor equations to account for the decay heat generated in the outer loop.

The calculation scheme for thermal feedback has been developed for fast and thermal spectrum

reactor applications, where the multigroup cross sections are functionalized in terms of fuel salt and moderator temperatures. In the coupled neutronics and thermal hydraulics calculations, the calculated power from neutronics calculations is used to determine velocity, temperature, and density distributions of the core using the thermal hydraulics solver. Then the temperature and density distributions are used to update the cross sections in order to perform neutronics calculations and determine the new power distribution in the core. This section explains the coupled neutronics and thermal hydraulics calculation scheme to account for thermal feedback. The thermal hydraulics, primary loop, and decay heat models are discussed first. Then the cross section functionalization scheme for thermal feedback is discussed. In the end, the overall computational procedure of the coupled calculation for thermal feedback is presented.

2.4.1. Thermal Hydraulics Model

In order to account for thermal feedback effects, a single-phase, parallel-channel thermal hydraulics model has been implemented in PROTEUS-NODAL. In this model, the thermal hydraulic behaviour of the core is described by representing the core with 1D parallel channels. Ignoring the axial heat conduction and the shear forces due to velocity gradients in the fuel salt, the mass, momentum, and energy balance equations for 1D flow in a vertical channel can be written as [62]

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial z} = 0, \tag{2.75}$$

$$\frac{\partial (\rho u)}{\partial t} + \frac{\partial}{\partial z}(\rho u^2) + \frac{\partial P}{\partial z} + \rho g + \left(\frac{\partial P}{\partial z} \right)_{fric} = 0, \tag{2.76}$$

$$\frac{\partial (\rho h)}{\partial t} + \frac{\partial (\rho u h)}{\partial z} = q'''_{salt}. \tag{2.77}$$

where ρ, u, h, and P are the density, velocity, enthalpy, and pressure of the fuel salt, respectively, q'''_{salt} is the volumetric heat source produced in the fuel salt, and g is the gravitational acceleration. A control volume approach was adopted for the spatial discretization of the flow equations. The spatially discretized equations provide a system of nonlinear ordinary differential equations. For the time discretization, a semi-implicit temporal difference scheme was employed, as in the RELAP-5 code [63]. This results in a system of linear equations for the time advanced pressures with a tri-diagonal coefficient matrix. This system of linear equations is solved by the forward

elimination and backward substitution algorithm. For reactors of closed flow channels, individual channel flows are determined to satisfy the equal pressure drop condition. For reactors with a solid moderator, the moderator temperature is calculated by solving the following heat conduction equation

$$\frac{\partial}{\partial t}[\rho_m c_{p,m} T_m(r,z,t)] = \frac{1}{r}\frac{\partial}{\partial r}\left[k_m r \frac{\partial}{\partial r} T_m(r,z,t) \right] + q'''_m(r,z,t) \ , \tag{2.78}$$

where ρ_m, $c_{p,m}$, T_m are the solid moderator density, specific heat, and temperature, respectively, q'''_m is the volumetric heat source generated in the solid moderator due to photon interactions and neutron scattering, and k_m is the heat conduction coefficient of the moderator. More details about the thermal hydraulics solver implemented in PROTEUS-NODAL can be found in Ref. [48, 50, 64].

2.4.2. Primary Loop Thermal Fluidic Model

In the reactor system, the fuel salt exiting the core flows through the primary system before returning to the core inlet. To describe the core behavior during transients more accurately, the heat exchanger in the primary system is also included in the thermal hydraulics model of MSR. The heat exchanger has been modeled with a lumped parameter approach to simulate the heat removal by the intermediate loop, considering it as a single node. The fuel salt temperatures at the core inlet and outlet are directly coupled with the fuel salt temperatures at the heat exchanger outlet and inlet, respectively. The other components of the primary loop, such as primary pumps, inlet and outlet plenums, are not considered in this model.

Assuming the material temperature variations in the heat exchanger during a transient are proportional to the steady state temperatures, the heat exchanger can be characterized by the average fuel salt temperature $\bar{T}_{HX}$, of which variation is determined by

$$C_{HX}\frac{d\bar{T}_{HX}}{dt} = \dot{m}_{HX} c_p \left(T_{in}^{HX} - T_{out}^{HX} \right) + \dot{Q}_{decay} - \dot{Q}_{HX} , \tag{2.79}$$

where $C_{HX} = m_{HX} c_p$ is the heat capacity of the heat exchanger, c_p is the specific heat of the fuel salt at the corresponding temperature, $\dot{m}_{HX}$ is the fuel salt mass flow rate, and m_{HX} is the mass of the heat exchanger. T_{in}^{HX} and T_{out}^{HX} are the fuel salt temperatures at the inlet and outlet of the heat exchanger, respectively, $\dot{Q}_{decay}$ is the decay heat generated by the fuel salt in the heat exchanger,

and $\dot{Q}_{HX}$ is the heat removal rate by the intermediate fluid in the heat exchanger, and it can be written as

$$\dot{Q}_{HX} = UA(\overline{T}_{HX} - \overline{T}_{IF}), \tag{2.80}$$

where $\overline{T}_{IF}$ is the average intermediate fluid temperature, U is the overall heat transfer coefficient from the fuel salt to the intermediate fluid, and A is total heat transfer area in the heat exchanger. Using the upwind scheme for spatial differencing with $T_{in}^{HX} = T_{out}^{c}$, $T_{out}^{HX} = \overline{T}_{HX}$, and $T_{in}^{c} = T_{out}^{HX}$, and the explicit scheme for temporal differencing, the discretized equation for heat exchanger can be derived as

$$\overline{T}_{HX}^{n} = \overline{T}_{HX}^{n-1} + \frac{\dot{m}_{HX}^{n-1} c_p^{n-1}}{C_{HX}} \left(T_{in}^{n-1} - T_{out}^{n-1} \right) \Delta t_n - \frac{\Delta t_n}{C_{HX}} \left(\dot{Q}_{HX}^{n-1} - \dot{Q}_{decay}^{n-1} \right). \tag{2.81}$$

Equation (2.83) can be rearranged and combined with an initial condition as

$$\overline{T}_{HX}^{n} = \left[1 - \frac{\dot{m}_{HX}^{n-1} c_p^{n-1} \Delta t_n}{C_{HX}} \right] \overline{T}_{HX}^{n-1} + \frac{\dot{m}_{HX}^{n-1} c_p^{n-1} \Delta t_n}{C_{HX}} T_{in}^{n-1} - \frac{\Delta t_n}{C_{HX}} \left(\dot{Q}_{HX}^{n-1} - \dot{Q}_{decay}^{n-1} \right), \tag{2.82}$$

$$\overline{T}_{HX}^{0} = T_{out}^{c} - \frac{\dot{Q}_{HX}^{0}}{\dot{m}_{HX}^{0} c_p^{0}}. \tag{2.83}$$

2.4.3. Cross Section Functionalization

For thermal feedback calculation, multigroup cross sections are functionalized in terms of fuel salt and moderator temperatures. Multigroup cross sections are prepared using the multigroup cross section generation code MC^2-3 [65], or the Monte Carlo codes OpenMC [66] and Serpent [67]. MC^2-3 code is dedicated to fast spectrum reactors and it cannot handle thermal spectrum reactors. Thus, it is used to generate cross sections for fast spectrum MSRs. On the other hand, the Monte Carlo codes are used for thermal spectrum MSRs since they can handle heterogeneous geometries easily, and the anisotropic scattering effect is not significant in thermal spectrum reactors since the higher order scattering cross section is weighted with scalar flux which can introduce a significant error in the eigenvalue. In PROTEUS-NODAL code the cross sections need to be provided in the format of the isotopic cross section dataset ISOTXS and the delayed neutron cross section dataset DLAYXS. The ISOTXS and DLAYXS datasets can be generated directly with MC^2-3 code, while the utility code GenISOTXS [68] is used to convert the output tallies into the ISOTXS and DLAYXS datasets format when Monte Carlo codes are used to generate cross

sections.

Cross sections are functionalized differently for fast and thermal spectrum MSRs. In a fast spectrum MSR, the reactivity feedback is due to the Doppler effects and the change of fuel density. In large fast reactors, the Doppler effects are approximately proportional to the inverse of the fuel temperature [69], and thus microscopic cross sections of an isotope i can be represented as a linear function of the logarithm of temperature T_f as

$$\sigma_{g,i}(T_f) = \sigma_{g,i}(T_{f0}) + \alpha_{g,i} \ln(T_f / T_{f0}), \tag{2.84}$$

where T_{f0} is a reference fuel temperature. After each thermal hydraulics calculation, the macroscopic cross sections for a fuel salt temperature T_f are updated as

$$\Sigma(T_f) = \frac{\rho(T_f)}{\rho(T_{f0})} \sum_i N_i(T_{f0})[\varepsilon_l \sigma_i(T_{fl}) + \varepsilon_h \sigma_i(T_{fh})], \tag{2.85}$$

where T_{f0} is a temperature for reference fuel salt density ρ, T_{fl} and T_{fh} are two fuel salt temperatures for which cross sections are prepared, and

$$\varepsilon_l = \frac{\ln(T_{fh} / T_f)}{\ln(T_{fh} / T_{fl})}, \quad \varepsilon_h = 1 - \varepsilon_l. \tag{2.86}$$

In a thermal spectrum MSR, cross sections depend on the moderator temperature T_m as well as the fuel salt temperature T_f. Based on the observation that the resonance integral in thermal reactors is approximately proportional to the square root of temperature [70], the microscopic cross sections of fuel isotopes are represented as a linear function of $T_f^{1/2}$. On the other hand, the moderator temperature dependency is represented as a linear function of T_m. After each thermal hydraulics calculation, the macroscopic cross sections are updated as

$$\begin{aligned} \Sigma(T_f, T_m) = \frac{\rho(T_f)}{\rho(T_{f0})} &\sum_{i \in f} N_i(T_{f0})[\varepsilon_{fl} \sigma_i(T_{fl}) + \varepsilon_{fh} \sigma_i(T_{fh})] \\ &+ \sum_{j \in m} N_j[\varepsilon_{ml} \sigma_j(T_{ml}) + \varepsilon_{mh} \sigma_j(T_{mh})] \end{aligned}, \tag{2.87}$$

where

$$\varepsilon_{fl} = (T_{fh}^{1/2} - T_f^{1/2}) / (T_{fh}^{1/2} - T_{fl}^{1/2}), \quad \varepsilon_{fh} = 1 - \varepsilon_{fl}, \tag{2.88a}$$

$$\varepsilon_{ml} = (T_{mh} - T_m) / (T_{mh} - T_{ml}), \qquad \varepsilon_{mh} = 1 - \varepsilon_{ml}. \tag{2.88b}$$

2.4.4. Coupled Neutronics and Thermal Hydraulics Calculation

The overall computational scheme of coupled neutronics and thermal hydraulics calculations for steady state and transient analyses is illustrated in Fig. 2.7. In the steady state calculations, the neutronics and thermal hydraulics equations are tightly coupled and solved iteratively until the power and temperature solutions converge. The initial power distribution in the core is determined with uniform temperature and velocity fields. At the end of each power iteration of the nodal calculation of PROTEUS-NODAL, thermal hydraulics calculations are performed to determine the fuel salt temperature, density, and velocity fields and the moderator temperature profile, if exists, using the updated power distribution. Then, the cross sections are updated using the calculated fuel salt and moderator temperature and density distributions. Using the updated cross sections and the salt velocity field, the nodal calculation is performed again. This process continues until the power and temperature distributions converge.

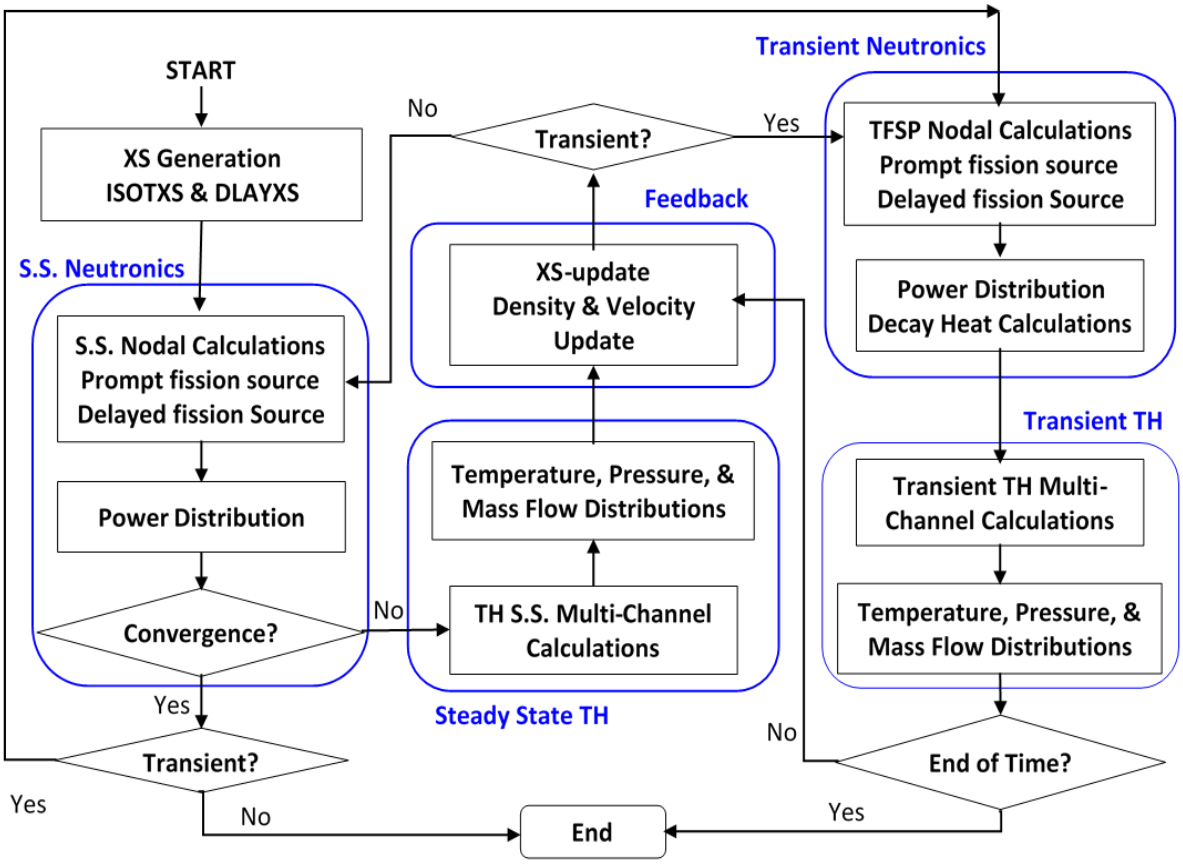


Figure 2.7. Overall computational flow of coupled neutronics and thermal hydraulics calculations.

In the transient calculations, the neutronics and thermal hydraulics calculations are performed once for each time step. At the beginning of a time step, the temperature and density distributions

at the previous time step are used to update cross sections. Using the updated cross sections and the velocity field, neutronics calculations are performed to calculate the power distribution. Using the updated power distribution, the thermal hydraulics calculations are performed to determine the new fuel salt temperature, density, and velocity fields, and the moderator temperature profile. Then, these updated temperature, density, and velocity distributions are used for the feedback calculations in the next time step until the end of the time steps is reached.

2.5. Summary

This chapter provided the mathematical foundation for steady state and transient analysis capabilities implemented in PROTEUS-NODAL code for MSRs applications with flowing fuel. New steady state and transient solvers were implemented into the PROTEUS-NODAL code to handle MSRs with liquid fuel. Several features were added to simulate flowing fuel including adjoint solution capability, kinetics parameters calculations, CMFD acceleration scheme, thermal hydraulics solver, heat exchanger model, and thermal feedback model to update cross sections and perform coupled calculations of the neutronics and thermal hydraulics calculations. The main focus was on solving the delayed neutron precursor equation with precursor drift term and incorporating its solution in the steady state and transient calculations.

The solution of the time-dependent neutron diffusion equation was provided for stationary and flowing fuels using the TFSP formulation. Several methods were investigated for solving the delayed neutron precursor equations and the FDM was selected as it provides a very stable solution, and it can be extended to consider a general velocity field easily. Simple verification tests were performed for the developed neutronics steady state and transient solvers without thermal feedback to test the robustness of the solvers for various cases. The effect of changing the fuel salt velocity and accompanied delayed neutron losses on the reactor power was tested for different velocities and different decay times. Further verification tests are performed in the following chapter with widely used benchmarks to verify implemented steady state and transient analysis capabilities using fast and thermal spectrum MSRs. Also, validation tests are provided in Chapter 4 using the available experimental results of the MSRE experiments.

Chapter 3. Verification Test Results

The most important part of newly developed capabilities into software is to verify the implemented capabilities using existing benchmark problems or other code solutions and performing validation tests with available experimental data or analytical solutions if possible. This consists of several tasks starting with collecting data of the problem, developing computational models, and perform verification tests. To verify the newly implemented capabilities in the PROTEUS-NODAL code for MSRs with flowing fuel, two benchmark problems were analyzed considering fast and thermal spectrum reactors. The first benchmark is the MSFR, which is a fast spectrum reactor where the fuel flows in a tank or one large flow channel since there is no solid moderator. The second benchmark is the MOST exercise, which was developed based on the MSRE experiment with several design simplifications in order to develop a benchmark for the verification of neutronics codes for flowing fuel without involving the design complications of the MSRE. The MSRE is a thermal spectrum reactor where the fuel salt is flowing into solid moderator channels of graphite. While Chapter 4 presents validation results of the PROTEUS-NODAL code for the MSRE experiment.

This section discusses the verification results of the steady state and transient solvers of PROTEUS-NODAL. The first part of this section discusses verification results of the MSFR benchmark problem, and the other part discusses the MOST benchmark exercise verification results. In each part, the problem specifications, the computational models, multigroup cross section generation, and the steady state and transient solutions in comparison with the other code solutions found in the open literature are presented.

3.1. MSFR Benchmark

In the Generation IV International Forum [11], liquid-fueled MSR research and development (R&D) has focused on fast-spectrum MSR options for fissile breeding and transuranic (TRU) isotope burning like the 1400 MWe MSFR. The MSFR design has a thermal power of 3000 MW_{th} and utilizes a fast neutron spectrum based on the thorium fuel cycle.

3.1.1. Specifications of MSFR Benchmark

The MSFR design utilizes a binary fluoride salt composed of LiF (77.5 molar %) and a heavy nuclei mixture (22.5 molar %) initially composed of fertile thorium and fissile material. The fuel salt is circulated in the core region and in the external loop, which consists of 16 branches each with the heat exchanger, pump, and associated instruments and pipes. The MSFR is operated between 650 °C and 750 °C with a circulation time of 4.0 s, which can be controlled based on the power level and the temperature rise in the core. The fuel salt temperature rise across the core region during normal operating conditions is fixed at 100 °C. Table 3.1 provides the main characteristics of the MSFR core and the thermo-physical properties for the fuel salt obtained from Ref. [12, 13].

Table 3.1. Characteristics of Molten Salt Fast Reactor.

Thermal/Electric Power	3,000 MWth/1,400 MWe
Fuel Composition	LiF-ThF_4-$^{233}UF_4$, LiF-ThF_4-$(TRU)F_3$
Fuel Inlet/Outlet Temperature	650 ºC/750 ºC
Core Height/Core Diameter	2.255 m/2.255 m
Fuel Salt Volume	18 m^3
Total Fuel Salt Transit Time	4.0 s
Flow Rate	4.5 m^3/s
Density [kg/m^3]	$\rho = 4094 - 0.882 \times (T[\mathrm{K}] - 1008)$
Kinematic Viscosity [m^2/s]	$\nu = 5.54 \times 10^{-8} e^{3689/T[\mathrm{K}]}$
Dynamic Viscosity [Pa·s]	$\mu = \left[2.268 - 4.886 \times 10^{-4} (T[\mathrm{K}] - 1008)\right] \times 10^{-4} e^{3689/T[\mathrm{K}]}$
Thermal Conductivity [W/m·K]	$k = 0.928 + 8.397 \times 10^{-5} T[\mathrm{K}]$
Specific Heat [J/kg·K]	$c_p = -1111 + 2.78T[\mathrm{K}]$

As shown in Fig. 3.1, the active core has a cylindrical shape with the same height and diameter of 2.255 m to improve the breeding ratio and reduce neutron leakage. The total fuel salt volume is 18 m^3, half of which is in the active core. The radial reflector includes a fertile blanket of 50 cm thick to increase the breeding ratio and shield the external components, and it is filled with a fertile salt of LiF-ThF_4. The walls of the blanket are surrounded by a 20 cm thick layer of B_4C to enhance the shielding of the external components. The external core structures and the heat exchangers are protected by thick reflectors made of nickel-based alloys for corrosion resistance. Specifications of the MSFR benchmark are provided in Appendix C.

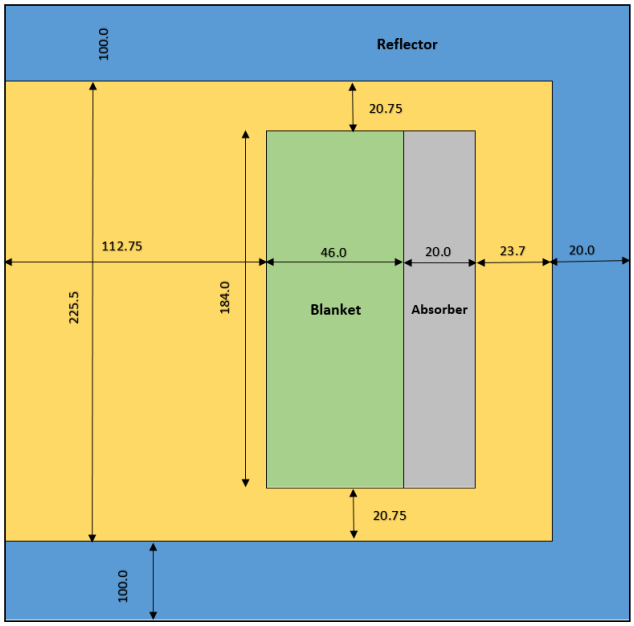


Figure 3.1. Vertical view of the MSFR core.

3.1.2. Computational Models

To perform steady state and transient analyses of the MSFR with PROTEUS-NODAL, multigroup cross sections need to be prepared using Monte Carlo codes or MC^2-3 code and provided as input. Multigroup cross sections for MSFR were prepared as a function of the fuel salt temperature using the MC^2-3 code with the ENDF/B-VII.0 data [71]. The fuel temperature considered for the feedback calculations ranges from 800 K to 1200 K. Microscopic cross sections for specified compositions were generated in 9 and 33 energy group structures and written in ISOTXS and DLAYXS datasets. The 33-group cross section set was used in steady state calculations. On the other hand, for transient calculations, the 9-group cross section set was used to reduce the computational time without a significant loss of solution accuracy. The energy group structures are provided in Table 3.2 and Table 3.3 for the 9-group and 33-group structures, respectively.

Table 3.2. Upper Energy Boundaries of 9-Group Structure.

Group	Energy (eV)	Group	Energy (eV)	Group	Energy (eV)	Group	Energy (eV)
1	1.419E+07	4	1.832E+05	6	9.119E+03	8	4.540E+02
2	2.231E+06	5	4.087E+04	7	2.035E+03	9	5.044E+00
3	8.209E+05						

The MSFR model in PROTEUS-NODAL was developed in R-Z geometry as shown in Fig. 3.2 for radial and axial views. In the core region, the radial node size of ~15 cm and axial node size of

11.25 cm were used. The PROTEUS-NODAL model for MSFR was tested by comparing the solution against the R-Z geometry with the FDM solution of the DIF3D code [72] for stationary fuel using the same 33-group cross sections set generated for ^{233}U fuel salt composition. The reference solution was obtained by DIF3D FDM solutions of 1 cm mesh size. The eigenvalue and power distribution of PROTEUS-NODAL agreed very well with the reference solution with an eigenvalue error of 1 pcm and the relative mean square error of nodal power was 0.051%.

Table 3.3. Upper Energy Boundaries of 33-Group Structure.

Group	Energy (eV)	Group	Energy (eV)	Group	Energy (eV)	Group	Energy (eV)
1	1.419E+07	10	1.832E+05	19	2.035E+03	28	2.260E+01
2	1.000E+07	11	1.111E+05	20	1.234E+03	29	1.371E+01
3	6.065E+06	12	6.738E+04	21	7.485E+02	30	8.315E+00
4	3.679E+06	13	4.087E+04	22	4.540E+02	31	3.928E+00
5	2.231E+06	14	2.479E+04	23	2.754E+02	32	5.316E-01
6	1.353E+06	15	1.503E+04	24	1.670E+02	33	4.175E-01
7	8.209E+05	16	9.119E+03	25	1.013E+02		
8	4.979E+05	17	5.531E+03	26	6.144E+01		
9	3.020E+05	18	3.355E+03	27	3.727E+01		

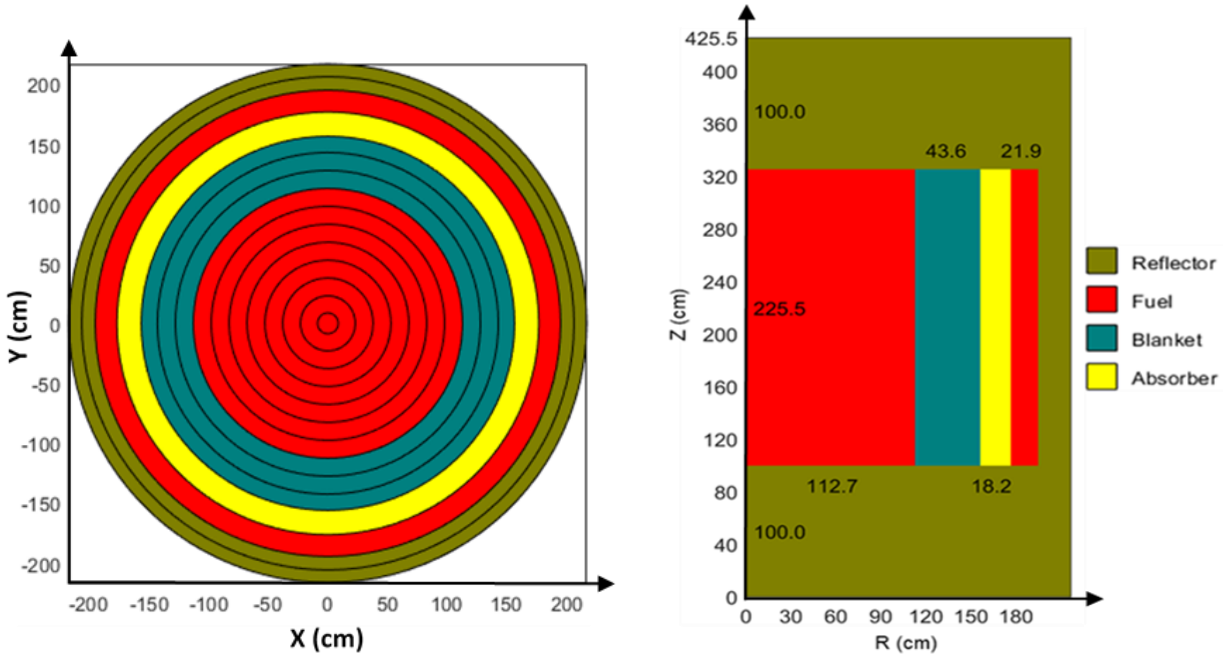


Figure 3.2. Planar (left) and axial (right) views of MSFR model in R-Z geometry.

Thermal hydraulics calculations were performed using the thermal hydraulics solver of PROTEUS-NODAL by dividing the core into eight parallel annular channels. The radial cross

flow was neglected due to the limited capabilities of the thermal hydraulics solver, and a uniform inlet mass flow rate was used. The fuel temperature, density, and velocity distributions in the core region were determined from single-phase thermal hydraulics calculations. The heat removal rate by the intermediate loop was considered using a lumped parameter model for the heat exchanger.

3.1.3. Static Analyses

The steady state calculations of the MSFR benchmark problem were performed for a stationary fuel to determine the effective multiplication factor of the initial core composition, and the Doppler and fuel density coefficients. Then, the flowing fuel modeling capabilities of PROTEUS-NODAL were verified by calculating the kinetics parameters, the reactivity losses due to delayed neutron precursor drift, and its distribution in the core region.

3.1.3.1. Reactivity Feedback Coefficients

For a stationary fuel, to calculate the eigenvalue for a given initial core composition of ^{233}U or TRU fuel salt, steady state calculations were performed for a uniform temperature distribution using the cross sections generated at an average core temperature of 973 K. Doppler and fuel density coefficients were calculated using temperature dependent cross sections. Table 3.4 compares the multiplication factors of the ^{233}U and TRU fuel cores from PROTEUS-NODAL calculations with the values reported in Ref. [13, 59]. The effective multiplication factor of MSFR is less than 1.0 for the ^{233}U fuel core but larger than 1.0 for the TRU fuel core. For both the ^{233}U and TRU fuel salt cores, the effective multiplication factor of PROTEUS-NODAL agrees well with the Serpent Monte Carlo solution of POLIMI obtained with the same ENDF/B-VII.0 data. It can be seen that different cross section libraries yield relatively large variations of the multiplication factor.

Table 3.5 presents the reactivity feedback coefficients of ^{233}U and TRU fuel cores. The Doppler coefficient was calculated for a fixed fuel salt density by changing the fuel salt temperature from 923 K to 1023 K, which are the core inlet and outlet temperatures, respectively. The salt density feedback coefficient was calculated by the reactivity difference between the core inlet and outlet salt densities for a fixed temperature of 973 K. The Doppler coefficient of the ^{233}U fuel core is about three times larger than that of the TRU fuel core. This is due to the higher ^{232}Th concentration in the ^{233}U fuel core. The salt density coefficient is slightly larger in the ^{233}U fuel core than in the TRU fuel core, and thus the total feedback coefficient is about 80% higher in the ^{233}U fuel core

than in the TRU fuel core. For both the ^{233}U and TRU fuel cores, the Doppler and density coefficients calculated with PROTEUS-NODAL agree reasonably well with the POLIMI results obtained with the same ENDF/B-VII.0 data.

Table 3.4. Comparison of Effective Multiplication Factors of MSFR.

Institute		POLITO	POLIMI		Purdue		UM
Code		Serpent	Serpent	Serpent	OpenMC	DIF3D	PROTEUS
XS Generation						OpenMC	MC^2-3
XS Library		JEFF-3.1.1	JEFF-3.1	ENDF/B-VII.0	ENDF/B-VII.1	ENDF/B-VII.1	ENDF/B-VII.0
No. of Groups						33	33
k_{eff}	^{233}U Core	0.99211 ±0.00011	0.99406 ±0.00040	0.98301 ±0.00041	0.97749	0.97936	0.98366
	TRU Core	1.02873 ±0.00012	1.01651 ±0.00044	1.01955 ±0.00045	1.02334	1.02506	1.02238

* POLITO (Politecnico di Torino), POLIMI (Politecnico di Milano), Purdue (Purdue University), UM (University of Michigan)

Table 3.5. Reactivity Feedback Coefficients (pcm/K) of MSFR.

Institute		KI	LPSC	POLITO	POLIMI		Purdue	UM
Code		MCNP-4B	MCNP	Serpent	Serpent	Serpent	DIF3D	PROTEUS
XS Generation							OpenMC	MC^2-3
XS Library		ENDF/B-VI	ENDF/B-VI	JEFF-3.1.1	JEFF-3.1	ENDF/B-VII.0	ENDF/B-VII.1	ENDF/B-VII.0
^{233}U Core	Doppler	-4.7±0.2	-2.6±0.1	-3.15±0.05	-3.84±0.07	-3.73±0.07	-3.66	-4.36
	Density	-2.8±0.2	-3.6±0.1	-3.42±0.02	-3.45±0.07	-3.55±0.07	-2.86	-3.32
	Total	-7.5±0.2	-6.2±0.1	-6.57±0.06	-7.29±0.07	-7.28±0.07	-6.52	-7.68
TRU Core	Doppler	-1.6±0.2	-1.5±0.1	-1.29±0.04	-1.64±0.06	-1.63±0.06	-1.73	-1.63
	Density	-2.7±0.2	-2.2±0.1	-2.85±0.04	-2.92±0.06	-2.75±0.06	-2.73	-2.62
	Total	-4.3±0.2	-3.7±0.1	-4.14±0.07	-4.56±0.06	-4.38±0.06	-4.46	-4.25

* KI (The Kurchatov Institute), LPSC (Laboratory of Subatomic Physics and Cosmology), POLITO (Politecnico di Torino), POLIMI (Politecnico di Milano), Purdue (Purdue University), UM (University of Michigan)

3.1.3.2. Reactivity Loss and Kinetics Parameters

The capability of PROTEUS-NODAL to model delayed neutron precursor drift was tested using different combinations of the transit time in the core (i.e., fuel salt speed) and that in the external loop (i.e., decay time). It is expected that when all the other conditions are the same, the effective multiplication factor of the flowing fuel is smaller than that of the stationary fuel due to the decay of the delayed neutron precursors in the regions of less importance or outside the core.

This reduces the effective delayed neutron fraction of the core, which in turn decreases the effective multiplication factor. Figure 3.3 presents the reactivity loss of the TRU fuel core due to the fuel flow as a function of the transit time in the core for six different values of transit time in the external loop. The MSFR design has a primary loop circulation time of 4.0 s with 2.0 s in the core and 2.0 s in the external loop. For this reference transit time, the reactivity difference is about 159 pcm, which is almost half of the effective delayed neutron fraction of the TRU fuel core (about 300 pcm). As the transit time in the core increases with decreasing fuel salt speed, the effective multiplication factor increases, so the reactivity loss decreases, and the effective multiplication factor converges to the stationary fuel value as the speed of the fuel salt approaches zero.

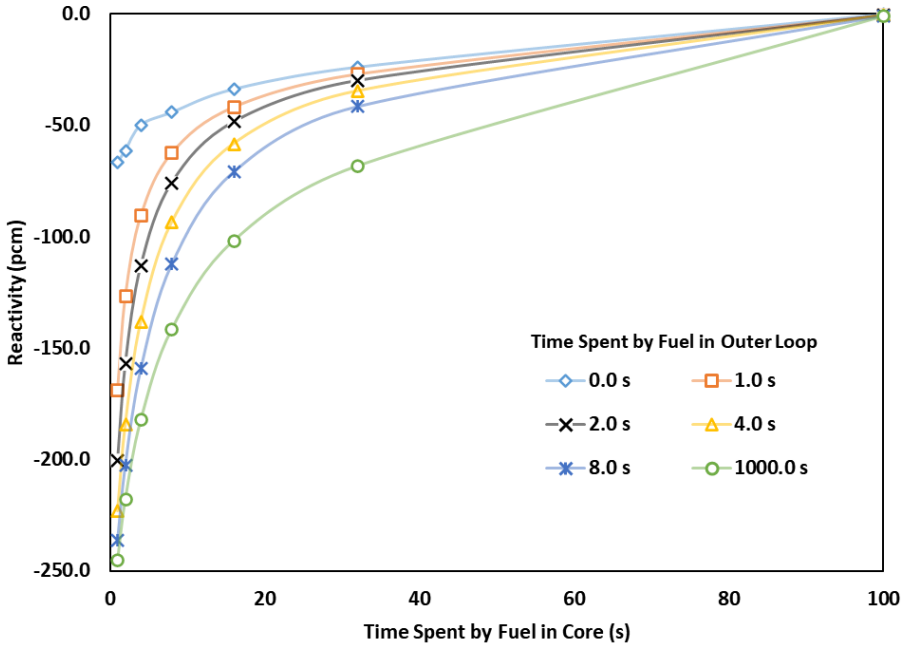


Figure 3.3. Reactivity losses of TRU fuel MSFR due to fuel flow.

The adjoint flux solution capability implemented in PROTEUS-NODAL was verified by calculating the kinetics parameters of the MSFR benchmark problem and comparing them with the other code results. Table 3.6 compares the effective delayed neutron fractions of stationary and flowing fuel salts, and the prompt neutron generation time for the ^{233}U and TRU fuel cores. The effective delayed neutron fraction of the flowing fuel salt is almost half of that of the stationary fuel salt. This is because the residence time of the fuel salt in the core equals the transit time of the fuel salt in the external loop. Both the effective delayed neutron fraction and the prompt neutron generation time calculated with PROTEUS-NODAL are consistent with other reported values.

Table 3.6. Calculated Kinetics Parameters of MSFR.

Core	Institute/Code	XS library	βeff (pcm)	βcirc (pcm)	$\frac{\beta_{circ}}{\beta_{eff}}$	βloss (pcm)	$\frac{\beta_{loss}}{\beta_{eff}}$	Λ (μs)
^{233}U	LPSC/MCNP	ENDF/B-VI	320.0	169.5	0.529	150.5	0.470	1.200
	POLITO/Serpent	JEFF-3.1.1	305.0	117.3	0.385	187.7	0.615	0.971
	POLIMI/Serpent	JEFF-3.1	305.0	146.0	0.479	159.0	0.521	1.090
	TUDelft	ENDF/B-VII	290.0	124.6	0.430	165.4	0.570	1.150
	Purdue/DIF3D	ENDF/B-VII.1	318.8	142.1	0.446	176.7	0.554	–
	UM/PROTEUS	ENDF/B-VII.0	317.7	141.2	0.444	176.6	0.556	1.103
TRU	LPSC/MCNP	ENDF/B-VI	312.8	165.5	0.529	147.3	0.471	0.900
	POLITO/Serpent	JEFF-3.1.1	301.0	–	–	–	–	0.783
	POLIMI/Serpent	JEFF-3.1	302.0	147.0	0.487	155.0	0.513	0.650
	Purdue/DIF3D	ENDF/B-VII.1	298.8	138.5	0.464	160.3	0.536	–
	UM/PROTEUS	ENDF/B-VII.0	294.6	135.0	1.464	159.6	0.542	0.623

* LPSC (Laboratory of Subatomic Physics and Cosmology), POLITO (Politecnico di Torino), POLIMI (Politecnico di Milano), TUDelft (Technical University of Delft), Purdue (Purdue University), UM (University of Michigan)

Figures 3.4 and 3.5 show the spatial distributions of six delayed neutron precursor concentrations in the TRU started MSFR for stationary and flowing fuels, respectively. For the flowing fuel, a uniform axial velocity field and complete mixing of fuel salt in the external loop were assumed with a fuel transit time of 4 seconds (2 s in the core and 2 s outside of the core). For the stationary fuel case, all the six distributions of precursor concentrations are symmetrical around the core mid-plane as the power distribution is. It is clearly seen that in the flowing fuel case, the high concentration region is shifted upward because of the upward fuel motion. The delayed neutron precursor groups of a small decay constant (large half-life) such as groups 1 and 2 show a larger shift since it takes a longer time to decay. On the other hand, the delayed neutron precursor groups of a large decay constant (small half-life) such as group 6 are less affected by the fuel flow and show a similar distribution to that in the stationary fuel. The motion of the delayed neutron precursors directly affects the delayed neutron distribution. The peak flux region of delayed neutrons is shifted upward, and a fraction of delayed neutrons is lost in the external loop. As a result, the effective delayed neutron fraction is reduced by about half in the flowing fuel case relative to the stationary fuel case.

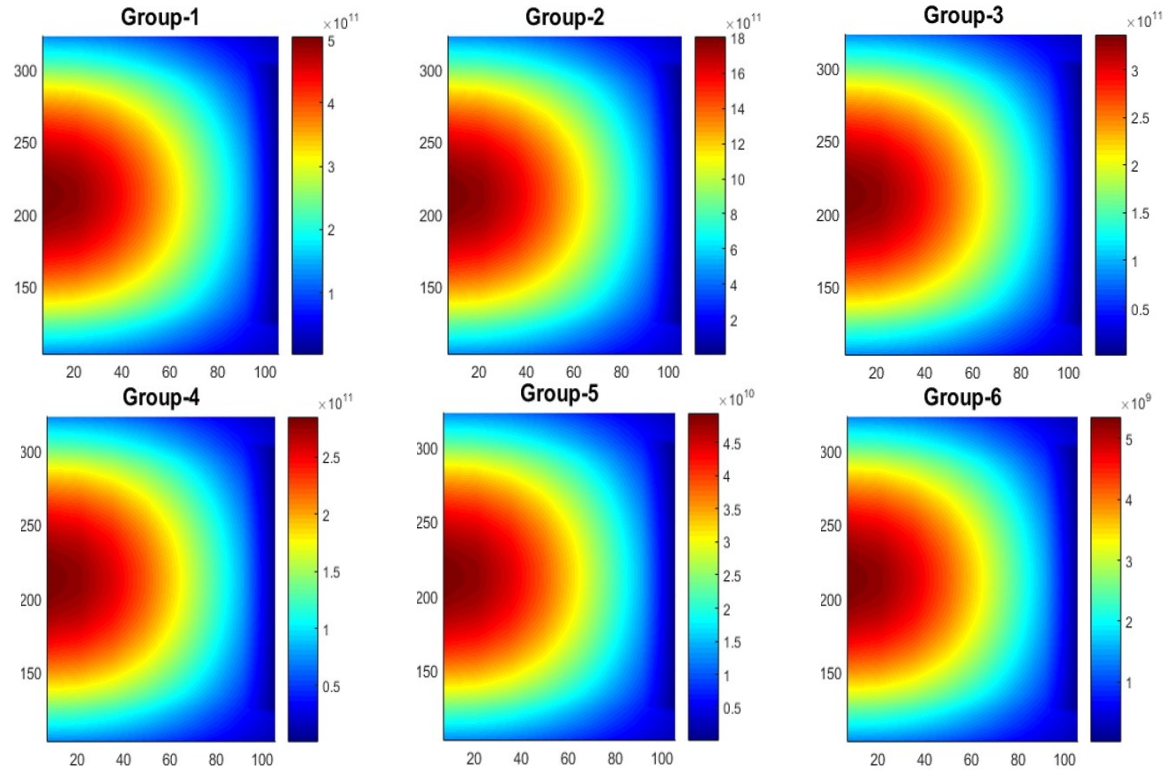


Figure 3.4. Delayed neutron precursor concentration distributions in stationary TRU fuel (#/cm^3).

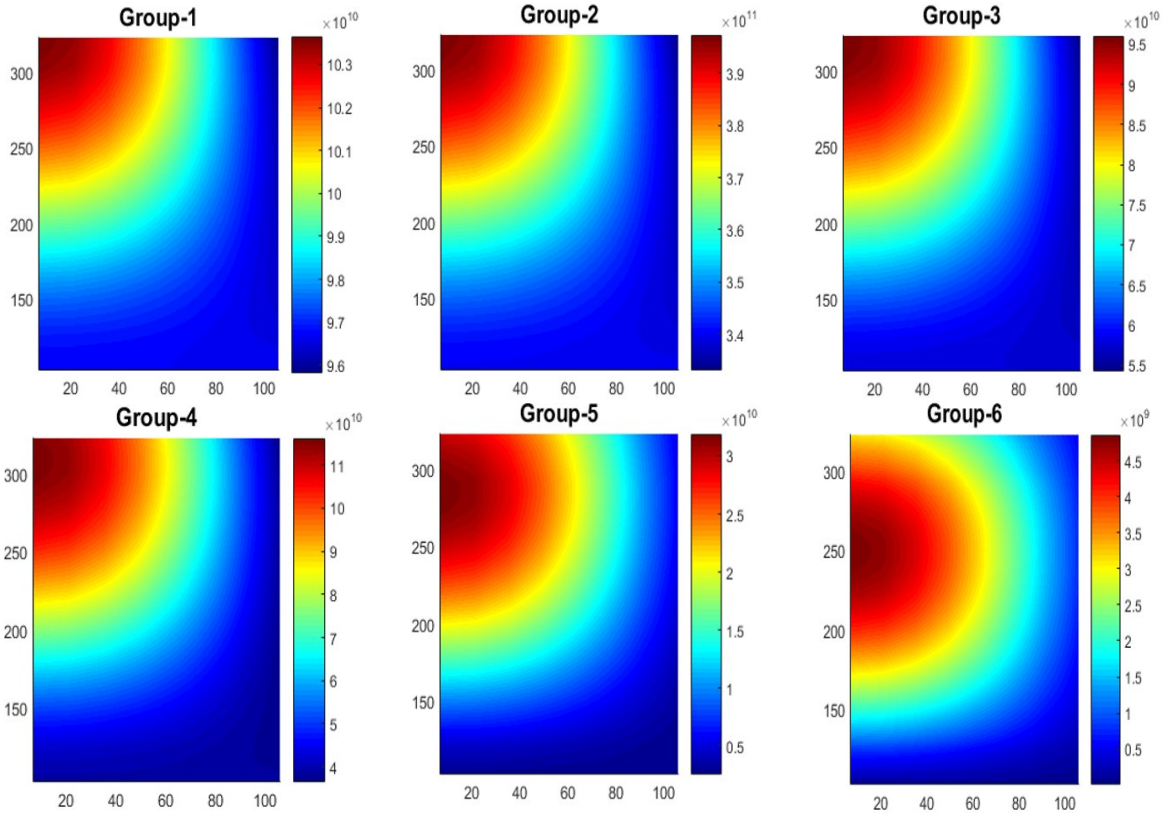


Figure 3.5. Delayed neutron precursor concentration distributions in flowing TRU fuel (#/cm^3).

3.1.4. Transient Analyses

The transient analyses of the MSFR benchmark problem were performed for of ^{233}U fuel salt. Since no numerical reference solution was available, the transient solutions of PROTEUS-NODAL were compared with two plot solutions presented in the open literature as plots. One is the solution of the Paul Scherrer Institute (PSI), which was obtained by coupling the system analysis code TRACE with the nodal neutronics code PARCS [40]. The other is the solution of the Delft University of Technology (TUDelft) through coupled neutronics and computational fluid dynamics (CFD) calculations [37]. The analysis was focused on the unprotected transients where the reactor protection system is assumed to fail to respond.

3.1.4.1. Reactivity Driven Transients

In the unprotected transient over power (UTOP) scenario, a step insertion of positive reactivity and subsequent failure of the reactor protection system to shut down the reactor were assumed. A step reactivity was inserted by increasing fuel salt concentration while maintaining the nominal fuel salt flow in the core. Two UTOP transients were simulated for a sub-prompt critical reactivity insertion of 50 pcm and a super-prompt critical reactivity insertion of 200 pcm at full power. Figures 3.6 and 3.7 compare the PROTEUS-NODAL and TUDelft solutions for the time-dependent core power and average temperature rise across the core. The sub-prompt critical transient results in Fig. 3.6 show that the power increases initially and attains its maximum (~1.5 times of the nominal power) around 8.0 ms into the transient. The increased power raises the fuel salt temperature, which introduces a negative reactivity due to the Doppler effects and the reduced fuel salt density. Because of the negative reactivity feedback, the power starts to decrease at 8.0 ms and approaches an asymptotic power at around 1.0 s. Around 3.0 s into the transient, the power experiences a step decrease again because of the heated fuel salt flowing back into the core.

It can be seen that the power and temperature solutions of PROTEUS-NODAL agree well with the TUDelft solutions. However, the peak power predicted by PROTEUS-NODAL is slightly lower than the TUDelft solution. This can be attributed to the different feedback coefficients. As shown in Table 3.5, the total reactivity feedback coefficient of PROTEUS-NODAL is -7.68 pcm/K, while the TUDelft's value is -6.79 pcm/K. The more negative reactivity feedback coefficient of PROTEUS-NODAL reduces the asymptotic core temperature rise by about 2.0 K. The super-prompt critical transient results in Fig. 3.7 show qualitatively similar power and temperature

behaviors as in the sub-prompt critical transient, but the initial power burst is 54 times higher and ~20 times narrower than that of the sub-prompt critical transient. The PROTEUS-NODAL results agree well with the reference TUDelft solutions, but the peak power is lower because of the more negative reactivity feedback coefficient. As discussed in Ref. [73], under the adiabatic approximation, the peak value and width of the power burst for a super-prompt critical transient can be estimated as

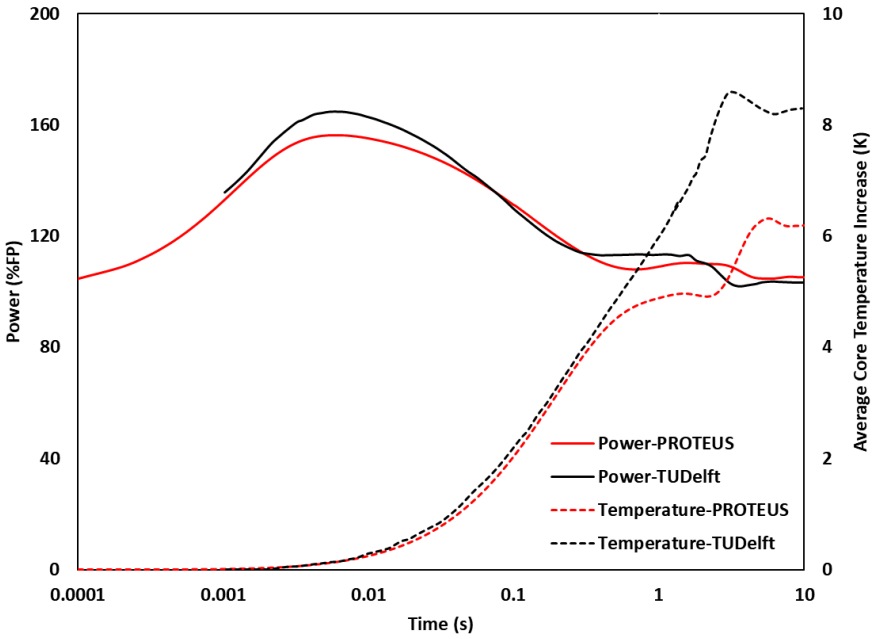


Figure 3.6. Power and average core temperature rise for 50-pcm reactivity insertion at full power.

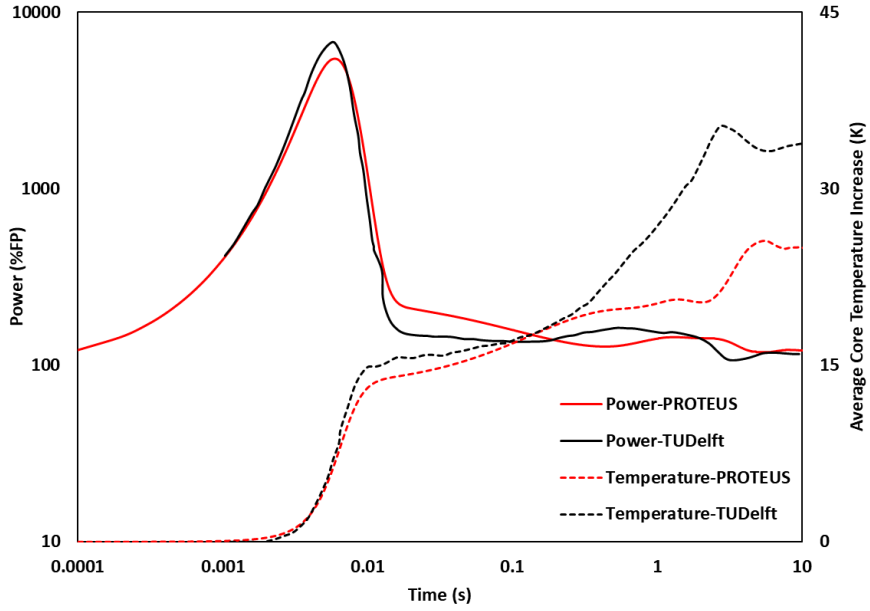


Figure 3.7. Power and average core temperature rise for 200-pcm reactivity insertion at full power.

$$P_m = P^0 - \frac{(\rho - \beta_{eff})^2}{2\Lambda\gamma} \quad , \tag{3.1}$$

$$\overline{\Delta t} = \frac{4\Lambda}{\rho - \beta_{eff}} \left(1.0 - \frac{P^0}{P_m} \right), \tag{3.2}$$

where P_m is the maximum power, $\overline{\Delta t}$ is the power burst width, ρ is the inserted reactivity, and γ is the energy feedback coefficient. Using the total feedback coefficient and fuel properties, the energy feedback coefficient was estimated to be 1.28×10^{-6} MJ^{-1} in PROTEUS-NODAL calculations and 1.15×10^{-6} MJ^{-1} in the TUDelft calculations. The estimated values of peak power and power burst width of PROTEUS-NODAL and TUDelft results are compared in Table 3.7. The calculated results are consistent with the analytically estimated values.

Table 3.7 Maximum Power Estimation for Super Prompt Critical Case.

		TUDelft	PROTEUS-NODAL
Energy feedback coefficient (MJ^{-1})		1.15×10^{-6}	1.28×10^{-6}
P_m / P_0	Code value	66	54
	Eq (3.1)	74	65
$\overline{\Delta t}$ (ms)	Code value	8.0	8.0
	Eq (3.2)	6.0	6.0

The UTOP transients at zero power conditions were analyzed considering an initial power of 0.01% of the nominal full power. Figures 3.8 and 3.9 show the power and temperature evolutions for a sub-prompt critical reactivity insertion of 50 pcm and a super-prompt critical reactivity insertion of 200 pcm, respectively. At low initial power, it takes a significant time to increase the power to heat the fuel salt, and thus thermal feedback is delayed significantly. The sub-prompt critical transient at zero power shows a somewhat different power evolution than at full power case. The power attains its maximum and then decreases to the asymptotic value after 400 s in the transient time. The peak power is about 58 times the initial value and occurs around 70.0 s. In the super-prompt critical transient, the power attains the peak value of about 2.6×10^5 times the initial value at 0.04 s. The fuel transit time affects the core temperature change, being different from the super-prompt critical at the full power.

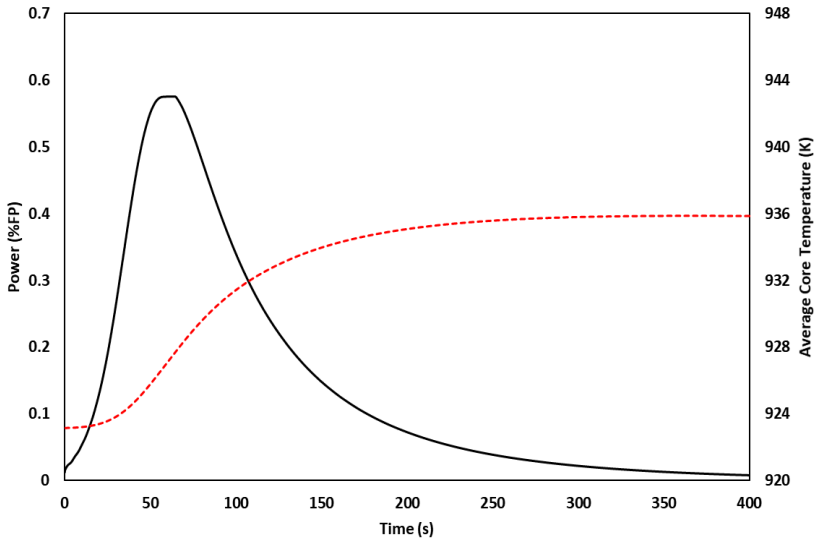


Figure 3.8. Power and average core temperature for 50 pcm insertions at zero power.

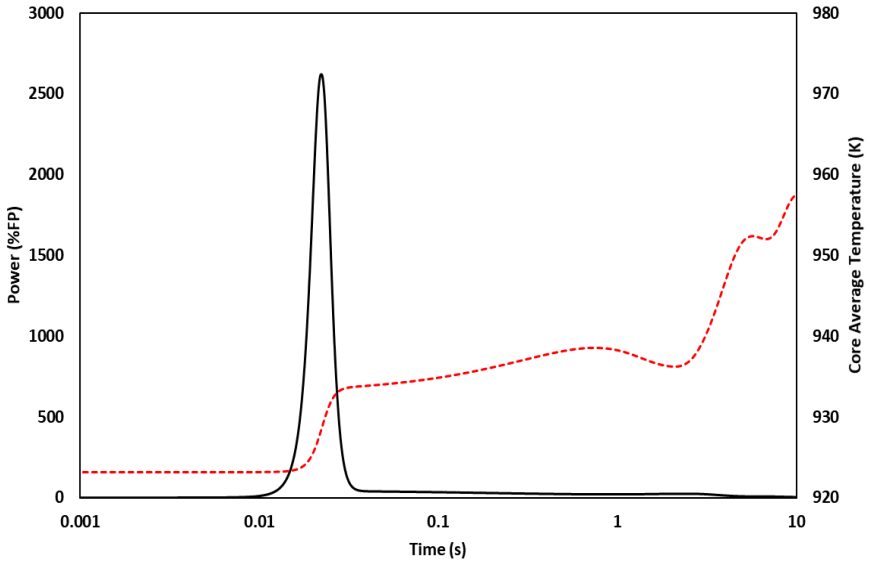


Figure 3.9. Power and average core temperature for 200 pcm insertions at zero power.

3.1.4.2. Flow Driven Transients

The primary pump controls the mass flow rate of the fuel salt in the core, which can be considered as a reactivity control mechanism for the current design of the MSFR. It is important to analyze the fuel flow driven transients related to pump operation, such as pump over-speed and loss of flow transients that could happen due to malfunction of the pump control system. Increasing the fuel flow rate will result in more delayed neutron loss while decreasing the fuel salt flow rate will reduce the delayed neutron loss. At the full power operation, the flow rate change affects the reactivity more through the fuel temperature change than through the change in delayed neutron loss.

In the unprotected pump over speed (UPOS) transient scenario, the mass flow rate was increased by 50% of the nominal flow rate with a time constant of 5.0 s. Figure 3.10 compares the time-dependent power solution of PROTEUS-NODAL with the PARCS-TRACE and TUDelft solutions. The evolutions of fuel flow rate and average temperature rise across the core are shown in Fig. 3.11. As the flow rate increases, the loss of delayed neutrons increases, but the average temperature rise decreases. The positive reactivity introduced by the cold fuel salt flowing into the core is larger than the negative reactivity due to increased delayed neutron loss. The net positive effect increases the power level by ~22% of the full power.

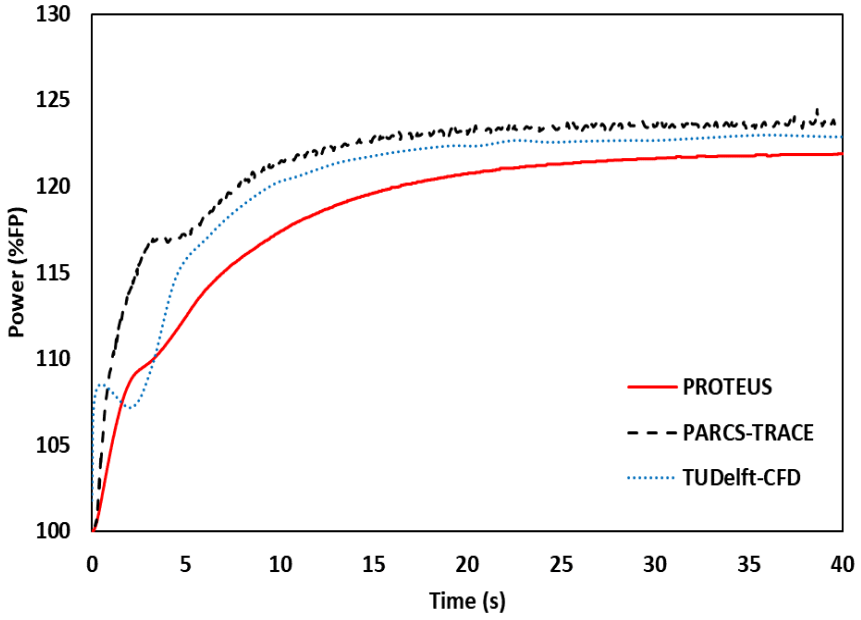


Figure 3.10. Power evolution during UPOS transient of MSFR at full power.

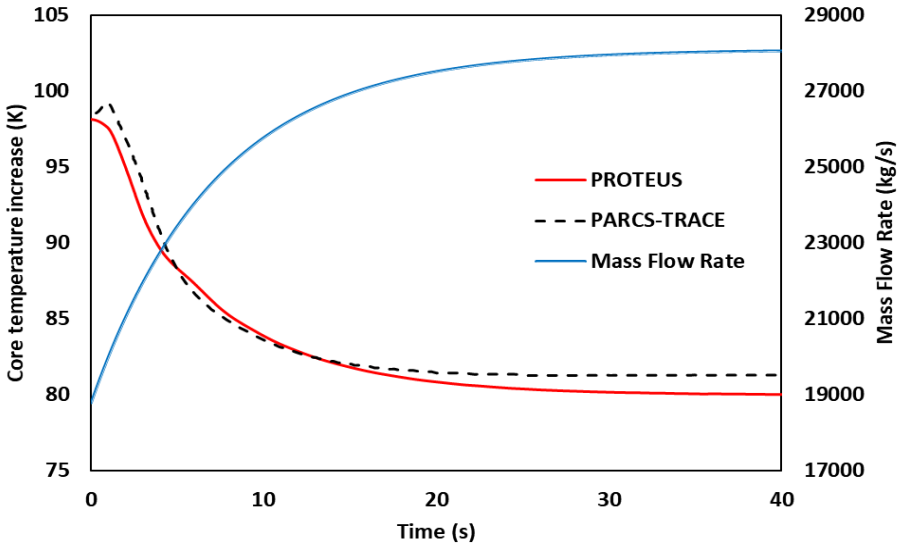


Figure 3.11. Average core temperature rise during UPOS transient at full power.

At low power, the core temperature rise is small, and thus the effect of increased delayed neutron loss is larger than the effect of decreased average core temperature rise. As a result, the net reactivity introduced by the increased salt flow rate is negative. Figure 3.12 shows the power evolution for a UPOS transient at zero power. As the flow rate increases in the core, the power starts to decrease due to the increased delayed neutron loss while the average core temperature almost remains constant. Thermal feedback has no effect on the UPOS at zero power since the fuel temperature is unchanged.

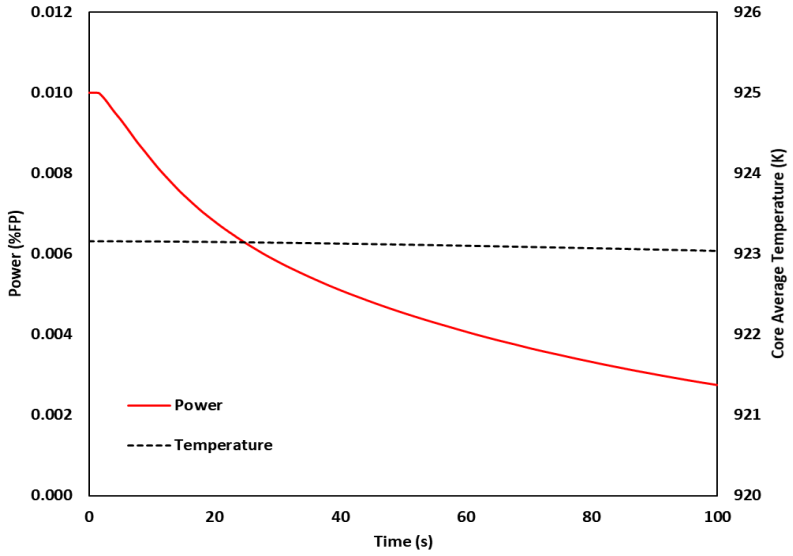


Figure 3.12. Power and average core temperature increase during UPOS transient at zero power.

An unprotected loss of flow (ULOF) accident is initiated by a pump failure that reduces the mass flow rate of the fuel salt in the core. The ULOF transient was simulated by decreasing the mass flow rate exponentially from its nominal value with a time constant of 5.0 s. The reduction of the flow rate introduces positive reactivity through the decreased loss of delayed neutrons out of the core. At the same time, the decreased fuel flow rate increases the fuel temperature in the core region and hence introduces negative reactivity feedback. This negative temperature feedback is larger than the positive reactivity introduced by the reduced delayed neutron loss out of the core. As a result, the net reactivity effect of loss of flow is negative. Figure 3.13 compares the power evolution of PROTEUS-NODAL to those of PARCS-TRACE and TUDelft. The TUDelft power approaches an asymptotic value at ~18% of the nominal full power due to the natural circulation buildup. On the other hand, the PROTEUS-NODAL and PARCS-TRACE solutions do not reach an asymptotic level since the inlet fuel salt velocity instead of the pump head was set to decrease exponentially with a time constant of 5 seconds, which leads to a more rapid decrease of the

average inlet velocity and no natural circulation buildup. Figure 3.14 shows the mass flow rate and compares the average core temperature rises of PROTEUS-NODAL and PARCS-TRACE. A large difference in the core temperature rise between PROTEUS-NODAL and PARCS-TRACE, which can be attributed to the simple lumped parameter thermal hydraulics model used in PROTEUS-NODAL to analyze the primary loop.

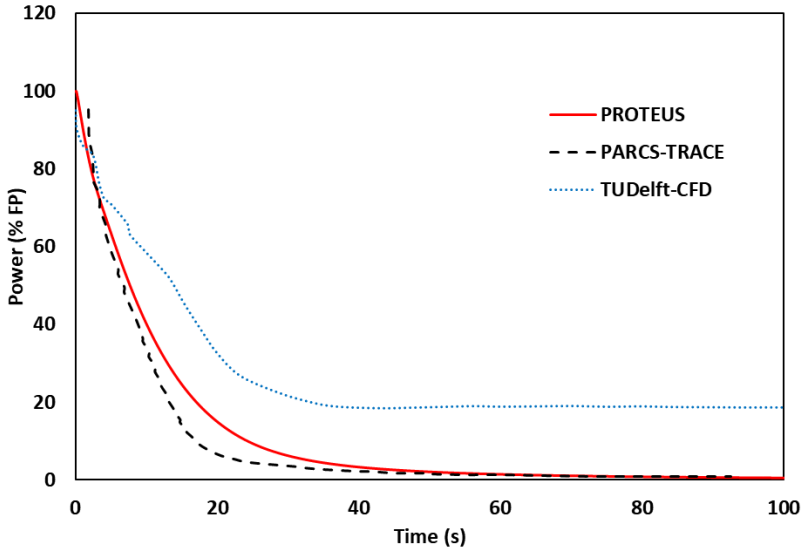


Figure 3.13. Power evolution during ULOF transient of MSFR at full power.

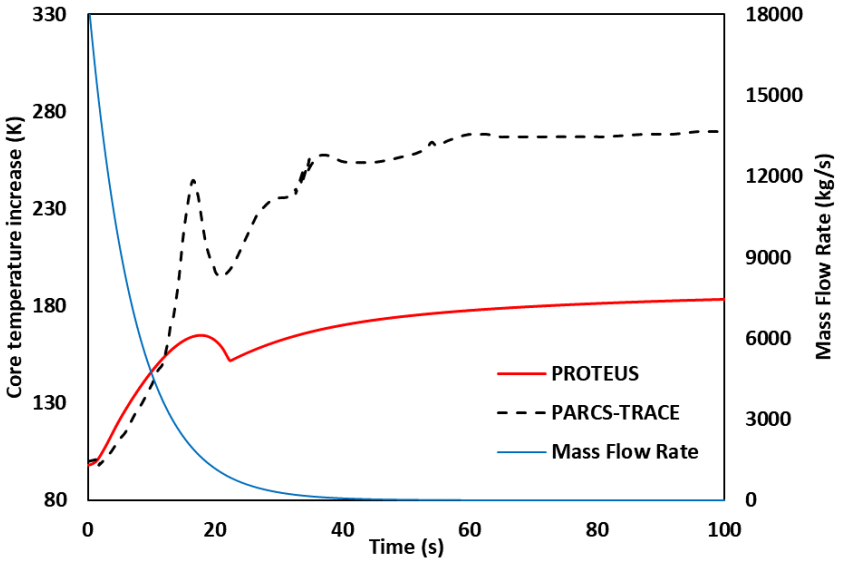


Figure 3.14. Average core temperature rise during ULOF transient at full power.

The ULOF scenario at zero power condition was also investigated. As shown in Fig. 3.15, the ULOF at zero power shows a different power evolution from the ULOF transient at full power. In this case, the positive reactivity introduced by the reduced loss of delayed neutrons is larger than

the negative reactivity introduced by increased salt temperature. As a result, the power increases and reaches ~1.1% of the nominal full power around 40 s. The fuel salt temperature starts to increase at around 30 s and introduces negative thermal feedback. Around 40 s in the transient time, the negative temperature feedback overwhelms the positive reactivity introduced by the decreased loss of delayed neutrons out of the core, and hence the power starts to decrease. The power approaches an asymptotic power that is slightly higher than the initial power, and the positive reactivity inserted by the reduced delayed neutron loss is balanced by the negative feedback due to increased salt temperature.

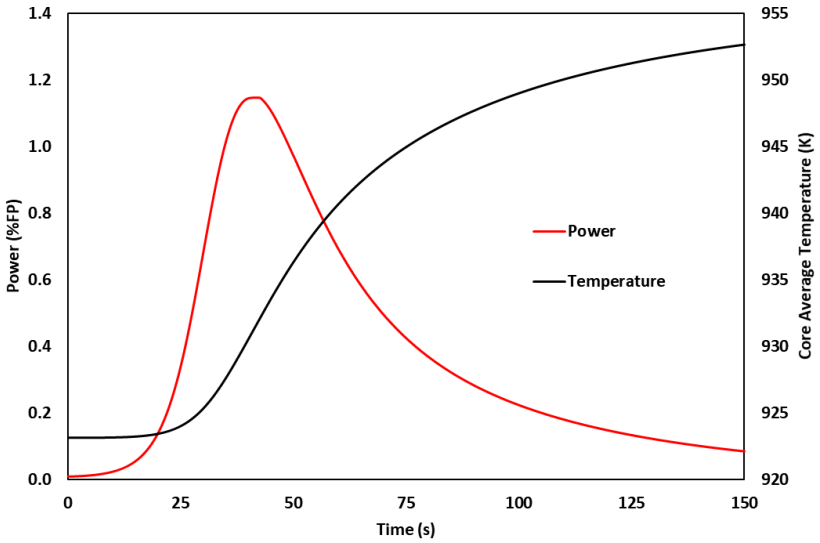


Figure 3.15. Power and average core temperature increase during ULOF transient at zero power.

3.1.4.3. Temperature Driven Transients

The temperature-driven transients resulted from changes in core inlet temperature are caused by a change in heat extraction rate in the heat exchanger. Decreasing the heat extraction rate or loss of heat sink will lead to an increase in core inlet temperature, and the increasing heat extraction rate or chilled inlet will lead to a decrease in core inlet temperature. Changes in inlet temperature have an immediate impact on reactivity. The unprotected loss of heat sink (ULOHS) or loss of cooling capabilities of the heat exchangers may happen due to the loss of flow or pump failure in the intermediate loop. The loss of cooling capabilities of the intermediate heat exchangers was assumed to follow an exponential decay with a time constant of 1.0 s, while the fuel flow in the core was maintained. As the heat removal rate decreases, the fuel salt temperature at the core inlet increases, and thus the core-averaged fuel salt temperature increases. The negative reactivity introduced by the increased fuel salt temperature makes the power decrease. The power results in

Fig. 3.16 show that the PROTEUS-NODAL solution agrees well in the shape with the TUDelft and PARCS-TRACE solutions but shows about 1.0 s lag. This is due to the assumption used in the PROTEUS-NODAL calculation that the uncooled fuel salt reaches the core inlet 1.0 s after the transient started. Figure 3.17 compares the average core temperature rise calculated with PROTEUS-NODAL to the PARCS-TRACE solution. The average fuel salt temperature rise across the core region of PROTEUS-NODAL shows a good agreement in shape with the TRACE result. However, it shows an oscillatory behavior with a period corresponding to the fuel circulation time in the primary system. This is because the heat removal by the intermediate loop was not considered during this ULOHS transient analysis.

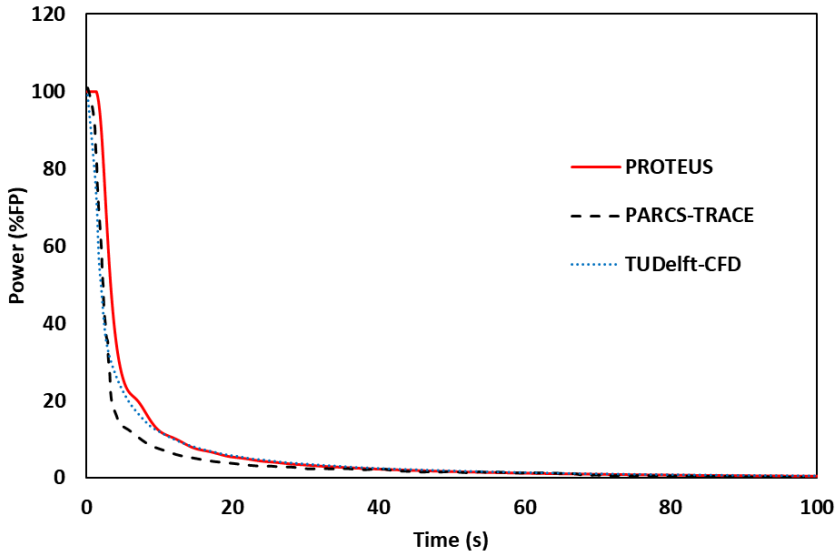


Figure 3.16. Power evolution during ULOHS transient at full power.

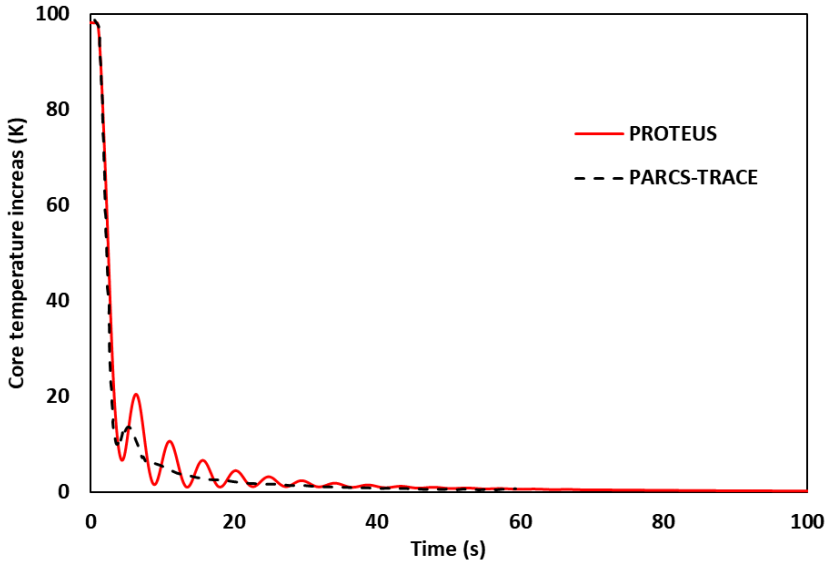


Figure 3.17. Average core temperature rise during ULOHS transient at full power.

An unprotected fuel salt over-cooling (UFSOC) or chilled inlet transient may happen due to the enhanced heat extraction capabilities in the intermediate heat exchanger, following depressurization of the steam generator or pump over-speed in the intermediate loop. The UFSOC transient was simulated by reducing the inlet temperature of the secondary coolant in the intermediate heat exchanger by 100 K exponentially with a time constant of 1.0 s while the flow rate was maintained at its nominal value. Such a decrease in the secondary coolant temperature will enhance the heat transfer rate between the fuel salt in the primary system and the coolant in the intermediate loop. This will lead to a decrease in the core inlet temperature, so the average fuel salt temperature in the core will decrease. As a result, the power will increase due to the positive reactivity inserted by the over-cooled fuel salt. Then the power will stabilize as the fuel salt is heated.

Figure 3.18 compares the power evolution of PROTEUS-NODAL with the PARCS-TRACE and TUDelft solutions. The power solution of PROTEUS-NODAL is in between the two solutions. The power increases during the first 5 s, and then reaches an asymptotic value of about 170% of the nominal full power. The power evolution of this UFSOC transient is highly affected by the heat exchanger and intermediate loop models, in particular the overall heat transfer coefficient of the heat exchanger. Figure 3.19 shows the average temperature rise of fuel salt across the core, which shows similar behavior to the power. The asymptotic fuel temperature rise across the core is about 70 K higher than the steady state core temperature rise of 100 K.

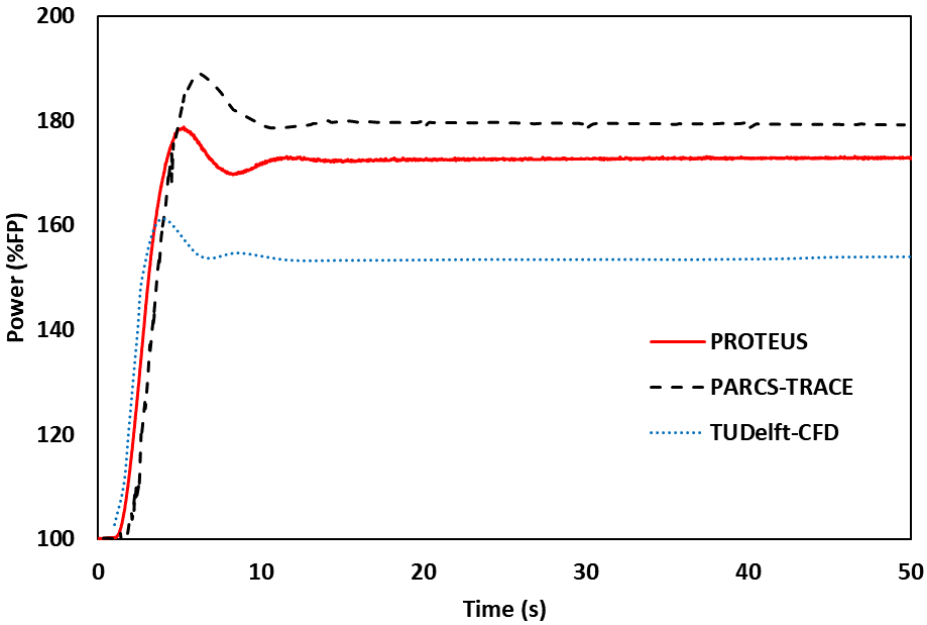


Figure 3.18. Power evolution during UFSOC transient at full power.

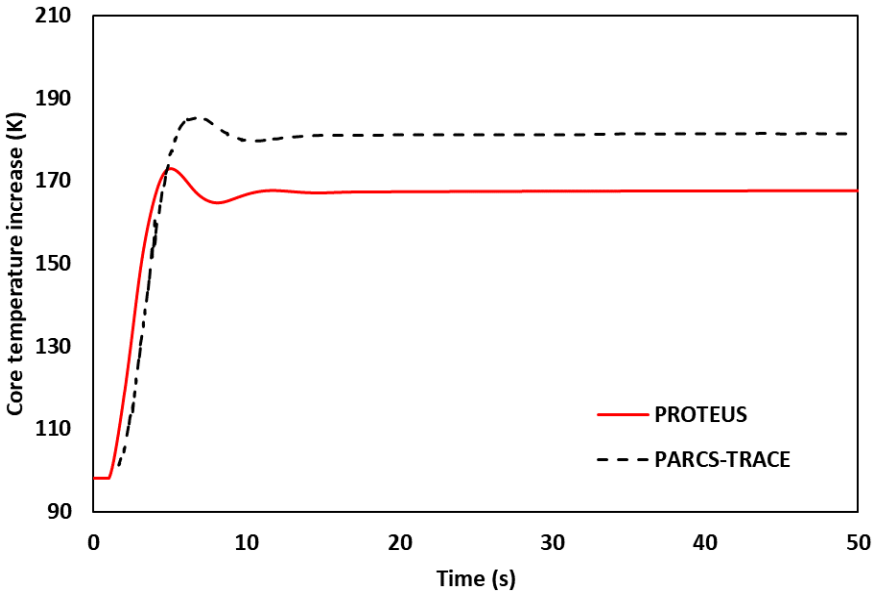


Figure 3.19. Average core temperature rise during UFSOC transient at full power.

3.2. MOST Benchmark Exercise

The MOST benchmark exercise was defined to assess the available computational tools for simulating flowing fuel systems under the MOST project with the participation of several European institutes [18]. Static and transient problems were developed based on some of the MSRE experiments. However, several simplifications were made to the MSRE geometry to simplify the simulation and analysis. Specifically, the core configuration was developed by neglecting the control rods and the outside of the active core, including the core container, the downcomer, the reactor vessel, the gap between reactor vessel and insulator, the insulator, and the thermal shield. The participants analyzed the benchmark problems using the point-kinetics to 3-D dynamics models. In this section, the MOST benchmark is analyzed to develop appropriate MSRE core models and multigroup cross section generation procedure and to verify the PROTEUS-NODAL code. The description of the MOST benchmark problem is first provided. Then, the PROTEUS-NODAL core models and the multigroup cross section generation with the Monte Carlo code OpenMC are discussed. The verification results for various steady state and transient problems are compared with the reported results in the open literature.

3.2.1. Specification of MOST Benchmark

The MOST project was started within the 5th European Framework Program to address the physical modeling issues of the molten salt reactors and to develop a benchmark exercise to assess

the adequateness of the computational tools available for flowing fuel system with reference to some of the experimental data from MSRE including steady state experiments to calculate the reactivity loss due to delayed neutron precursor drift and the fuel salt, graphite, and isothermal temperature coefficients of the MSRE. The transient experiments include the protected pump startup and coast down, and the natural circulation experiment [18].

The main difference of the MOST benchmark from the MSRE experiment is the simplifications that were adopted into the geometry of the MSRE core in order to simplify the analysis. The control rods and the outside of the active core, including the core container, the downcomer, the reactor vessel, the gap between reactor vessel and insulator, the insulator, and the thermal shield, were not considered in the defined model of the benchmark problem. The defined core model is limited up to the outer fuel regions of the MSRE core with a core diameter of 139.0 cm and a height of 163.0 cm. The inner core region was specified by a regular lattice made of the square graphite stringers and the rectangular fuel salt channels with round corners in the sides of the stringers. Figure 3.20 shows the radial configuration of the core and a graphite stringer cell with four half channels of fuel salt of the defined core under the MOST project. Vacuum boundary conditions were specified on the outer surfaces of the cylindrical core. Furthermore, the core characteristics were defined based on the design data and reports, which are slightly different from the chosen values or measured data during the MSRE experiments. The main characteristics of the MOST benchmark are provided in Table 3.8. Specifications of the MSRE Experiment are provided in Appendix D [74, 75, 76, 77].

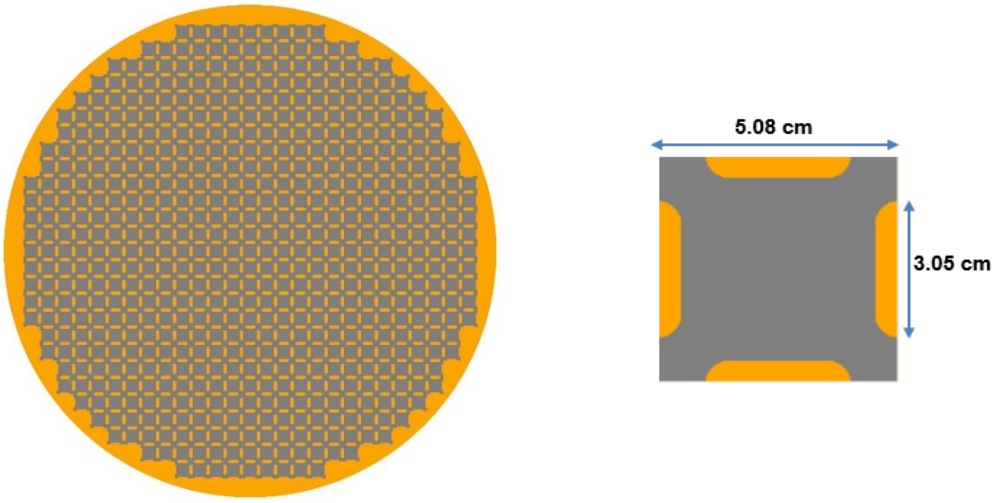


Figure 3.20. Radial core configuration of MOST benchmark (left) and a unit graphite stringer cell with four half channels of fuel salt (right).

Table 3.8. Main Characteristics of MOST Benchmark Problem.

Thermal power	10 MWth
Fuel composition	LiF-BeF_2-ZrF_4-UF_4
Molar composition	^{235}U (65.0%-29.1%-5.0%-0.9%)
Fuel inlet/ Outlet temperature	908 K / 936 K
Core height / Core diameter	1.63 m / 1.39 m
Total fuel salt transit time	25.2 s
Fuel salt density [kg/m^3]	$\rho = 2263 - 0.4798 \times (T(K) - 923.0)$
Fuel salt dynamic viscosity [Pa·s]	$\mu = 0.008268$
Fuel salt thermal conductivity [W/m·K]	$k = 1.4$
Fuel salt specific heat [J/kg·K]	$c_p = 1868.0$
Graphite density [kg/m^3]	$\rho_g = 1860.0$
Graphite thermal conductivity [W/m·K]	$k_g = 40.1$
Graphite specific heat [J/kg·K]	$c_{p,g} = 1757.3$

3.2.2. Core Model and Energy Group Structure

To model heterogeneous geometry of the MSRE core and to account for the spectral changes in generating cross sections, the Monte Carlo code OpenMC was used to generate multigroup cross sections of the MOST benchmark and the MSRE experiment. Also, the developed models with OpenMC are used to verify the developed models with PROTEUS-NODAL for stationary fuel case. For the MOST benchmark problem, two PROTEUS-NODAL core models were developed in Cartesian and R-Z geometries and tested along with five energy group structures against the OpenMC Monte Carlo solution. Also, the OpenMC code was used to generate multigroup cross sections with ENDF/B-VII.1 data [82] at several energy group structures to determine adequate group structure to perform the analysis. The selected core model for the MOST benchmark and the selected energy group structure for multigroup cross section condensation for the analyses of the MSRE experiment are provided in Appendix E.

In summary, the R-Z geometry model was selected to perform the analysis of the MOST exercise because it represents the outer regions of the core accurately so that the reaction rates are preserved. Also, an energy structure with a 16-group structure was selected to perform the analysis because it provided the smallest difference in eigenvalue and leakage fraction of the tested cases a detailed analysis is provided in Appendix E.

3.2.3. Static Analyses

The static problems were solved with PROTEUS-NODAL, and the results were compared with the reported solutions in open literature by the participants in the MOST benchmark exercise. Specifically, the eigenvalue, the isothermal, fuel, and moderator temperature coefficients, and the reactivity loss due to fuel circulation were calculated without thermal feedback. The effective multiplication factor and the fuel salt, graphite moderator, and isothermal temperature coefficients for stationary fuel were calculated without thermal feedback. The multiplication factor was calculated at the core average temperature of 922 K since the core inlet temperature was 908 K and the core outlet temperature was 936 K during full power operation. The temperature coefficients were calculated by the least-square fitting of the reactivity calculated at the temperatures of 20 K interval from 860 K to 1000 K, as shown in Fig. 3.21.

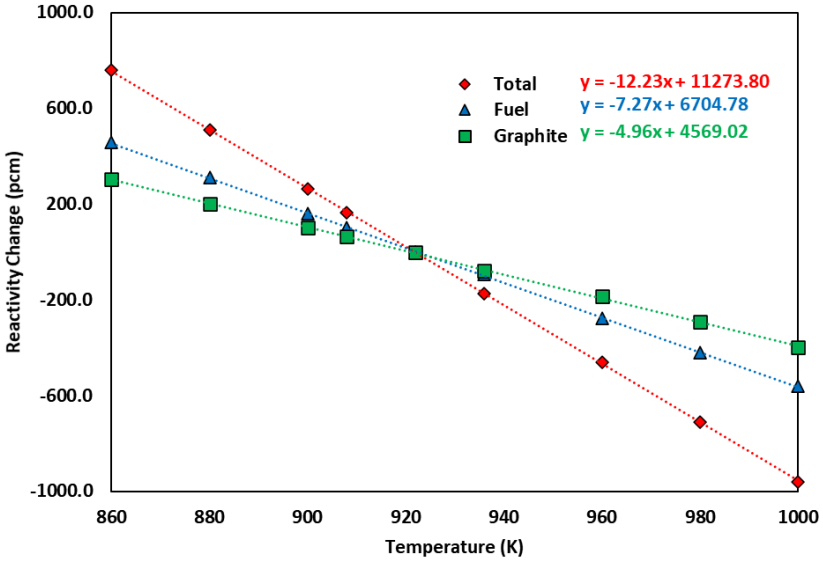


Figure 3.21. Reactivity feedback coefficients of MOST benchmark problem.

Table 3.9 compares the multiplication factor and reactivity feedback coefficients results of PROTEUS-NODAL with those of the MOST benchmark participants [18] and the DYN3D [25] and MOREL [27] codes. The multiplication factor of PROTEUS-NODAL is in good agreement with other institute results. The fuel and isothermal temperature coefficients of PROTEUS-NODAL are slightly smaller than the other institute values while the graphite temperature coefficient is slightly higher. The observed differences are mainly due to the different cross section libraries used in these analyses. This can be confirmed by the consistently good agreement of the PROTEUS-NODAL multiplication factors with the OpenMC results within 22 pcm at all temperatures, as shown in Table 3.10.

Table 3.9. Reactivity Feedback Coefficients (pcm/K) of Stationary Fuel.

Institute or Code	k-eff	Salt Temp. Coefficient	Graphite Temp. Coefficient	Isothermal Temp. Coefficient
BUTE	1.05980	-6.2	-4.8	-11.0
ENEA	1.07513	-6.8	-4.3	-11.1
EDF	1.06752	-7.8	-4.6	-12.4
FZR	1.06966	-6.9	-4.0	-10.9
POLITO	1.07060	-6.3	-3.8	-10.1
DYN3D	1.06161	-7.1	-4.5	-11.1
MOREL	1.06643	-7.4	-4.5	-11.9
PROTEUS	1.06232	-7.3	-4.9	-12.2

Table 3.10. Comparison of Multiplication Factors at Different Temperatures.

Temperature (K)	OpenMC	PROTEUS	Diff. (pcm)
860	1.07106 ± 0.00011	1.07095	-11.3
880	1.06819 ± 0.00011	1.06810	-9.2
900	1.06549 ± 0.00010	1.06528	-21.0
908	1.06423 ± 0.00011	1.06418	-4.7
922	1.06252 ± 0.00011	1.06232	-20.4
936	1.06042 ± 0.00011	1.06037	-5.0
960	1.05720 ± 0.00010	1.05712	-8.1
980	1.05444 ± 0.00010	1.05437	-7.5
1000	1.05180 ± 0.00011	1.05158	-22.0

Table 3.11 compares the reactivity loss by fuel circulation of PROTEUS-NODAL with the results reported in the open literature. The reactivity loss of PROTEUS-NODAL is consistent with the MOREL code result, of which cross sections are based on the same ENDF/B-VII data.

Table 3.11. Loss of Delayed Neutron Fractions of MOST Benchmark Problem.

XS Library	Institute/Code	Total	Delayed Neutron Precursor Family					
			1	2	3	4	5	6
JEF	BUTE	224.0						
	EDF	207.6	12.8	65.6	55.3	71.4	2.5	0.0
	ENEA	234.5	14.9	76.2	62.9	77.9	2.9	0.0
	FZK	258.3	16.4	83.0	68.3	85.8	4.5	0.3
	FZR	223.0	10.2	74.6	60.5	75.1	2.6	0.0
	POLITO	251.7	17.0	84.6	65.5	80.4	4.0	0.2
ENDF	MOREL	246.3	13.6	88.9	67.7	73.6	2.5	0.0
	PROTEUS	245.6	16.5	81.2	61.2	77.0	9.3	0.5

3.2.4. Transient Analysis

The transient problems of the MOST benchmark were solved with PROTEUS-NODAL, and the results were compared with the reported solutions in the open literature. The protected pump transient and natural circulation tests were analyzed. The pump transient tests were performed at constant low power to evaluate the effects of flow rate changes on the reactivity during pump startup and coast down transients. The natural circulation test was performed to determine the characteristics of the heat removal from the MSRE fuel system by natural convection of the primary fuel salt.

3.2.4.1. Protected Pump Transient Tests

In the protected pump transients of the ^{235}U fueled MSRE, the reactor was operated at a low power level (~10 W), and the fuel flow rate in the primary loop was increased or decreased by adjusting the speed of the fuel pump. The control rod position was adjusted in order to maintain a constant power level by compensating for the reactivity loss or gain due to the change in the fuel velocity in the core and consequently due to the redistribution of delayed neutron precursors in the core and their decay outside the core. The recorded control rod position is provided in Ref. [77] as shown in Fig. 3.22. The recorded control rod positions were converted into reactivity using the integral control rod worth curves. Since this test was performed at very low power and thus there is no thermal feedback effect during this transient, the reactivity inserted by the control rod movement is equal to the reactivity change due to the flow perturbations.

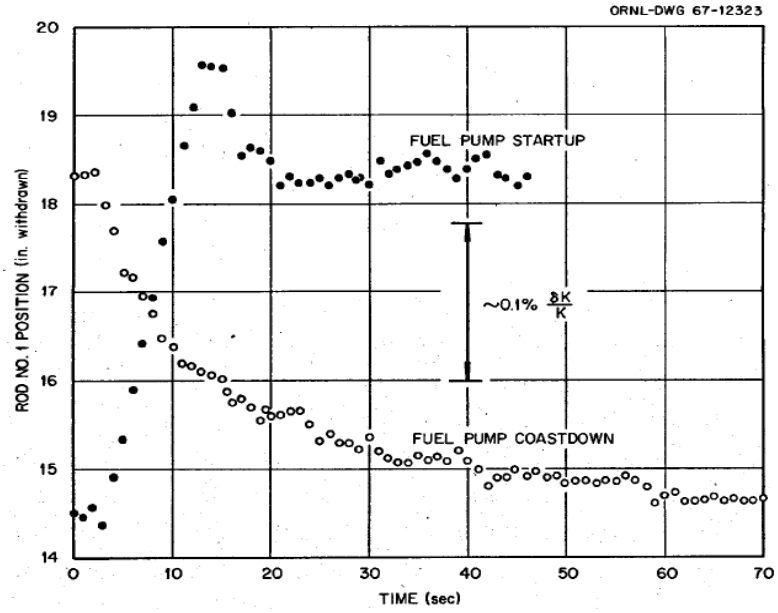


Figure 3.22. Control rod response to fuel pump start-up and coast-down [77].

PROTEUS-NODAL calculations were performed by providing the fuel flow rate and using uniform fuel temperature and velocity. The fuel flow rate was increased from zero to its nominal value in 10 seconds during the pump startup, and it was decreased from its nominal value to zero in 20 seconds during the pump coast down, as shown in Fig. 3.23. Constant power was maintained in PROTEUS-NODAL calculations by adjusting the neutron production source according to the calculated reactivity loss. The reactivity loss was calculated by taking the difference between the effective delayed neutron fractions of stationary and flowing fuel at each time step.

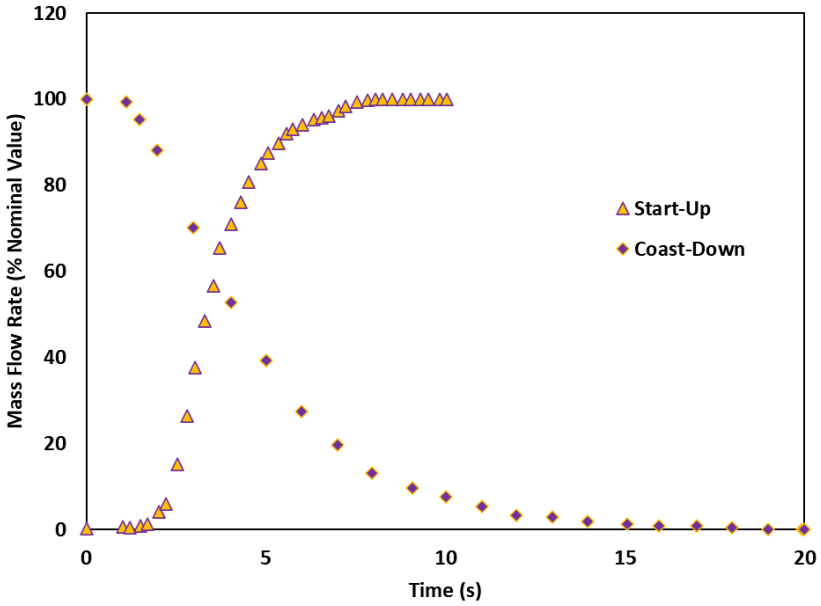


Figure 3.23. MSRE fuel flow rates during protected pump startup and coast down transients.

Figure 3.24 presents the reactivity compensations during the protected pump startup test of the MOST benchmark participants [18] and the comparison of the PROTEUS-NODAL reactivity compensation with the measured data [77] and the calculated values with the DYN3D [25] and MOREL [27] codes. During the protected pump startup test, the fuel starts to flow outside the core, which leads to reactivity loss because of the decay of delayed neutron precursors outside the core or at regions of lower importance. The reactivity loss increases with the increasing flow velocity. Oscillatory behavior is observed in the compensated reactivity because of recirculated precursors. The positive reactivity effect of the recirculated precursors entering the core is clearly seen 13 seconds after pump start-up. The difference between the calculated reactivity loss of PROTEUS-NODAL and the measurement can be attributed to the delay in the response of the reactor control system and the difference in the delayed neutron data.

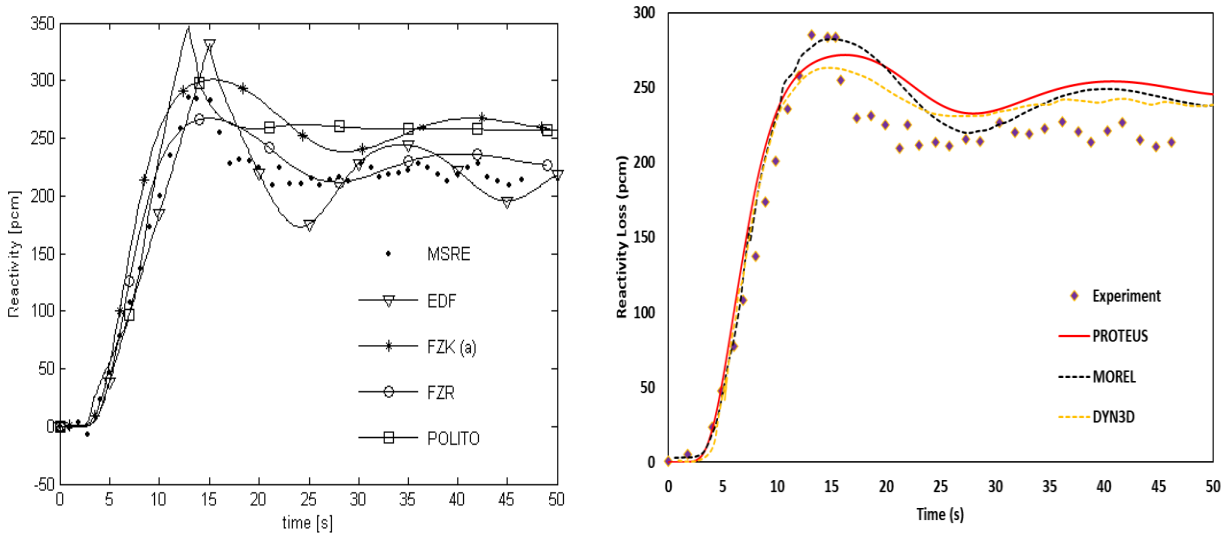


Figure 3.24. Reactivity compensations during protected pump startup transient of MOST participants (left) [18] and PROTEUS-NODAL (right).

Figure 3.25 shows the compensated reactivity plots for the protected pump coast down test. During the protected pump cost down transient, the reactivity increases because more precursors decay in the core region with decreasing fuel velocity. Therefore, the compensated reactivity continues to decrease and reaches zero when all the precursors decay in the core. It can be seen that PROTEUS-NODAL follows the measured data very well.

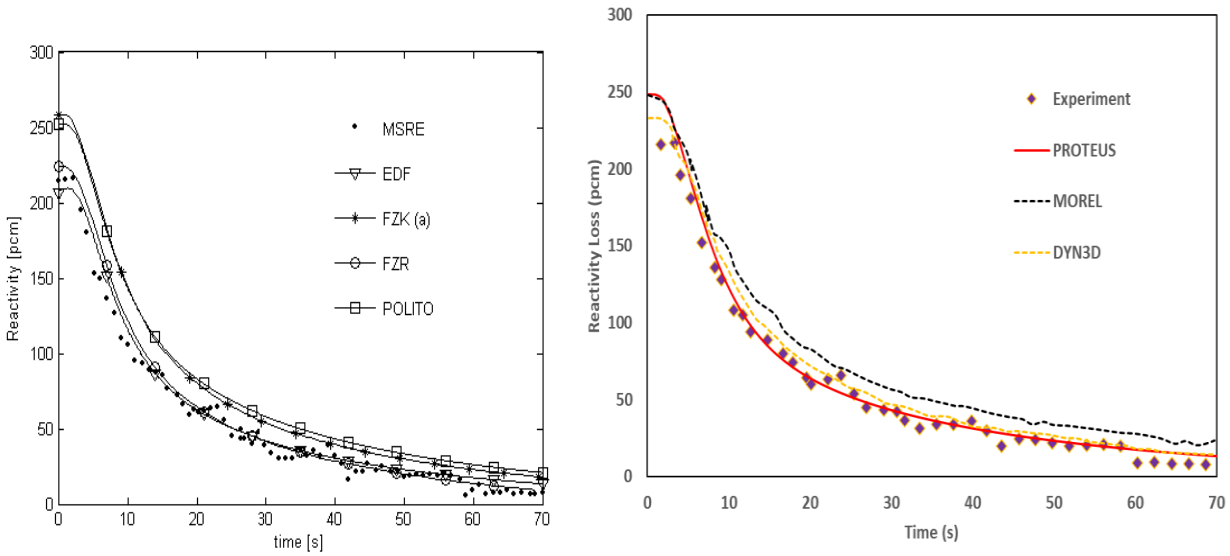


Figure 3.25. Reactivity compensation during protected pump coast down transient of MOST participants (left) [18] and PROTEUS-NODAL (right).

3.2.4.2. Natural Circulation Test

The natural circulation test of the MSRE with ^{233}U fuel salt was also analyzed. The purpose of the natural circulation test was to determine the characteristics of heat removal from the MSRE fuel system by natural convection. At the beginning of the transient, the reactor was operated at low power of about 4.1 kW with a limited fuel flow rate. After that, the core inlet temperature was decreased by increasing the heat removal rate in the air radiator in the secondary loop for 6 hours. The reactor was controlled entirely by the inherent thermal feedback of the system during this transient and no adjustments were made to the control rods. In this test, there is no data recorded for the fuel salt inlet mass flow rate or the heat removal rate by the secondary loop. Therefore, the inlet mass flow rate of the fuel salt was deduced using the recorded data of power and core inlet and outlet temperatures and the transit time of the fuel salt in the core. The recorded data over the transient period is provided in Fig. 3.26 [83].

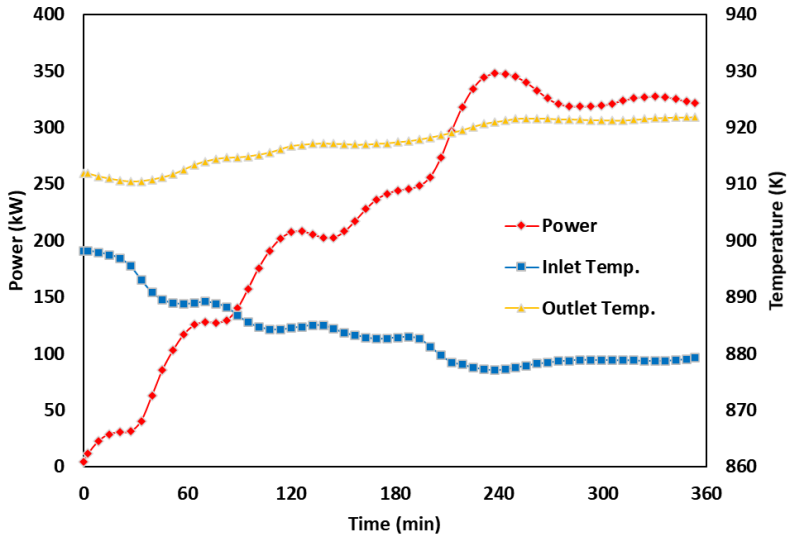


Figure 3.26. Recorded power and average inlet and outlet temperatures during natural circulation test of MSRE [83].

The natural circulation test was simulated with the PROTEUS-NODAL code. The core inlet temperature and the deduced inlet mass flow rate of the fuel salt were used as the boundary conditions for thermal hydraulics calculations during the transient period without considering the heat removal in the external loop. Figure 3.27 compares the calculated power history of PROTEUS-NODAL with the measured data [83] and the reported solutions in the open literature [18, 25, 27]. The PROTEUS-NODAL solution follows the measured power evolution very well.

The observed deviation at the beginning of the transient could be due to the assumed initial steady state conditions, which might not be true. In general, PROTEUS-NODAL is able to reproduce the measured power level during this long transient for a given inlet temperature and flow rate.

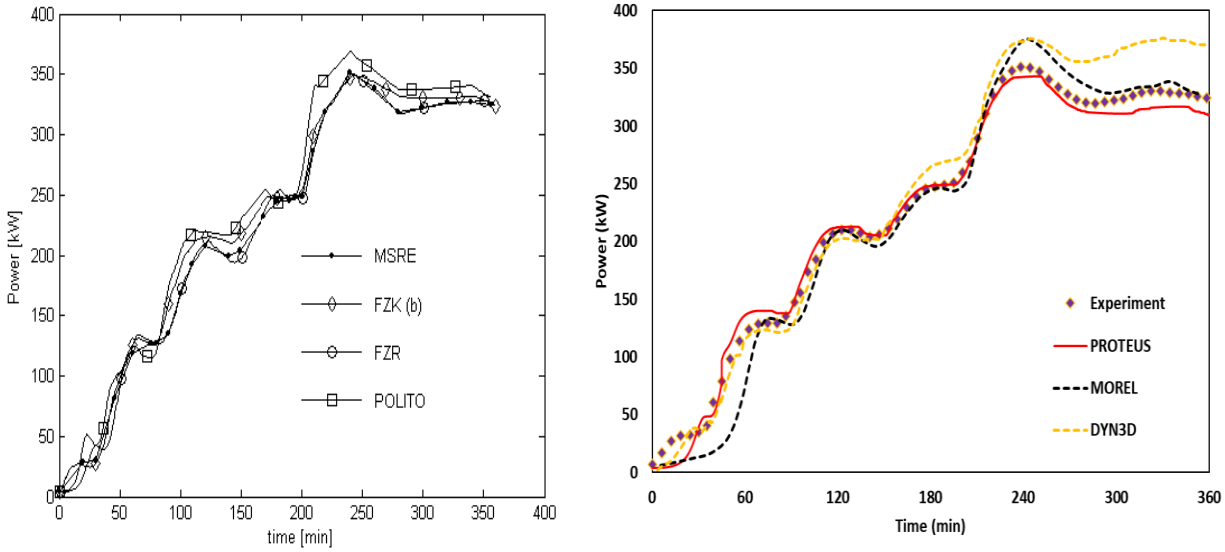


Figure 3.27. Power evolutions during natural circulation test of MOST benchmark participants (left) [18] and PROTEUS-NODAL (right).

3.3. Summary

The implemented capabilities in the PROTEUS-NODAL code for flowing fuel analysis were verified by conducting code-to-code comparisons using the MSFR benchmark, which is a fast spectrum reactor, and using the MOST benchmark exercise, which is a thermal spectrum reactor. For the MSFR benchmark, steady state analysis was conducted based on the available information in the preliminary design report of MSFR and compared with other code solutions. The PROTEUS-NODAL results for the multiplication factor, critical composition, and reactivity coefficients agreed well with other code solutions obtained with the same evaluated nuclear data file. Also, various unprotected transient tests with thermal feedback were performed for the MSFR benchmark problem including UTOP, UPOS, ULOF, ULOHS, and UFSOC. The results were compared with the other code solutions found in the open literature. For all the transients considered, the time evolution of power and average core temperature generally agreed well in shape and magnitude with the reference results reported in other studies. However, some noticeable deviations were observed mainly because of the simple hydraulics model used in the calculations.

In the MOST benchmark, the defined core for MSRE was utilized to determine the required models for generating the multigroup cross sections using the Monte Carlo code OpenMC. The energy group structure and the number of energy groups were determined by achieving the best match for eigenvalue and leakage fraction compared to the OpenMC results. Also, verifications tests were performed for steady state and transient analysis including, reactivity feedback coefficients, reactivity losses due to delayed neutron precursor drift, protected pump startup and coast down transients, and natural circulation test. The PROTEUS-NODAL results show reasonable agreement with reference solutions that solved the MOST benchmark problem.

Chapter 4. Analyses of MSRE Experiments and Validation Results

The Molten Salt Reactor Experiment (MSRE) provides the only source of experimental data that can be used to validate modeling and simulation tools for flowing fuel. The PROTEUS-NODAL code was used to analyze the steady state and transient experiments of the MSRE. The computational models of the MSRE were developed for the exact core without the simplifications introduced in the MOST benchmark. After describing the main characteristics of the MSRE, the developed computational models using OpenMC and PROTEUS-NODAL and multigroup cross section generation procedure are discussed. Since the internal void region cannot be modeled by PROTEUS-NODAL, the outside of the reactor vessel that includes a large gap between the reactor vessel and the insulation liner was excluded by developing an albedo boundary condition to compensate for neutron reflection from the outer regions. Static and dynamic experiments of the MSRE were analyzed which includes experimental data for the control rod worth, the temperature feedback coefficients, and the reactivity loss due to fuel circulation. The transient experiments include the protected pump startup and coast down tests at zero power, and the natural convection test, which was inherently controlled by the thermal feedback of the system.

4.1. Description of MSRE Configuration and Systems

In the 1960s, the MSRE was constructed and successfully operated at the ORNL to be the first liquid fuel salt and graphite moderator reactor. Furthermore, it was the first reactor ever to be operated using ^{233}U fuel. The main purpose of the MSRE was to demonstrate the practicality of the liquid fuel operation at high temperatures and ensure the safety and reliability for developing the full-scale MSBR project [7, 8]. Although the MSRE was operated for about 10 years, it provides the only source of experimental data for validating the developed simulating tools for flowing fuel reactors.

The MSRE was built in 1964 with an 8 MWth power and utilizes a thermal neutron spectrum with liquid fuel salt flowing into graphite moderator channels. Although the MSRE was designed to be a 10 MWth reactor, it was discovered in the final stages that the heat extraction in the secondary system was smaller than expected, and the maximum power level was restricted to 8.0

MW as measured from heat balances. The operation of the MSRE was started in 1965 and continued until 1968, and it was fueled with ^{235}U fuel (33% enriched) with the salt composition of LiF-BeF_2-ZrF_4-UF_4 (65.0%-29.17%-5.0%-0.83%). It was designed to be operated between 908 K and 936 K, with a temperature rise of the salt by 28 K and the core average temperature of 922 K. In 1968, the fuel was replaced with ^{233}U fuel (91% enriched) with a slightly different molar composition (64.5%-30.18%-5.19%-0.13%). Table 4.1 provides the main characteristics of the MSRE core and the thermos-physical properties for the fuel salt as obtained from Ref. [74, 75]. Specifications of the MSRE Experiment are provided in Appendix D.

Table 4.1. Characteristics of Molten Salt Reactor Experiment.

Thermal power	8 MWth
Fuel composition	LiF-BeF_2-ZrF_4-UF_4
Fuel inlet/ Outlet temperature	908 K / 936 K
Core height / Core radius	1.66 m / 0.762 m
Total fuel salt transit time	25.2 s
Fuel salt density [kg/m^3]	$\rho = 2322.7 - 0.502 \times \left(T[K] - 922\right)$
Fuel salt dynamic viscosity [Pa·s]	$\mu = 0.00785$
Fuel salt thermal conductivity [W/m·K]	$k = 1.44$
Fuel salt specific heat [J/kg·K]	$c_p = 1967.8$
Graphite density [kg/m3]	$\rho_g = 1874.16$
Graphite thermal conductivity [W/m·K]	$k_g = 39.8$
Graphite specific heat [J/kg·K]	$c_{p,g} = 1758.5$

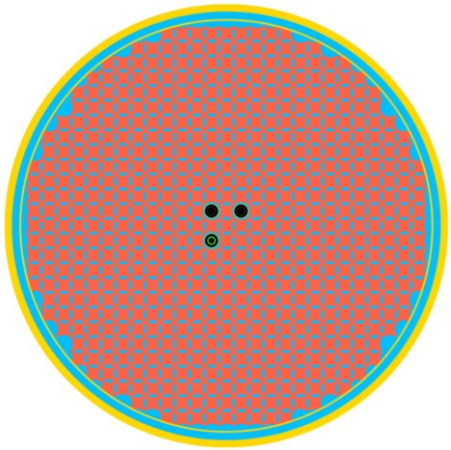
4.2. Computational Models

This section presents the developed computational models with OpenMC and PROTEUS-NODAL to analyze the MSRE experiments. OpenMC was used to generate the multigroup cross sections and albedo boundary conditions at the reactor vessel's outer surface. PROTEUS-NODAL was used for static and transient neutronics analyses of the MSRE.

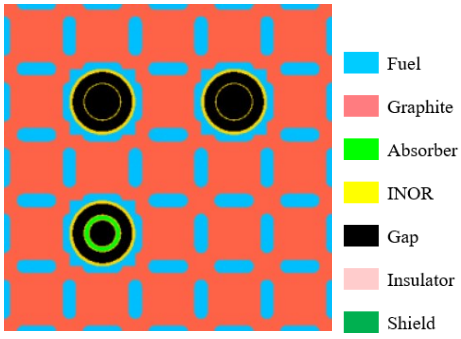
4.2.1. OpenMC Model

A detailed 3-D OpenMC model was developed to generate region-dependent multigroup cross sections by considering the neutron spectral changes in the core and to verify the PROTEUS-NODAL model for stationary fuel experiments. The simplifications introduced in the MOST benchmark were removed, and the control rods and the outside regions of the reactor core were

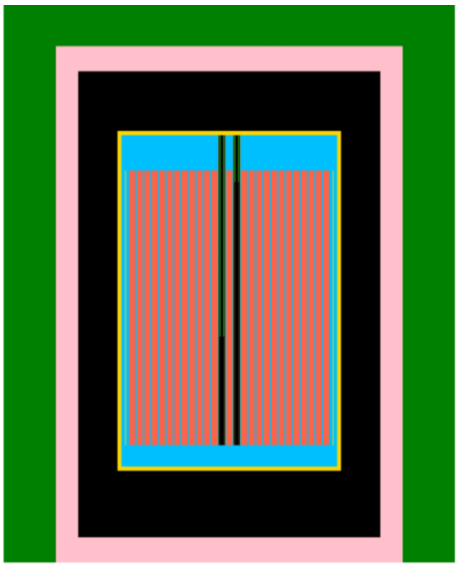
modeled explicitly. The outside of the core includes the core container, the downcomer, the reactor vessel, the air gap, the insulator, and the thermal shield, which reflect a significant fraction of leaking neutrons back to the core and thus affect the neutron flux distribution in the outer core region. Figure 4.1 shows the radial and axial representations of the core and the control region in the OpenMC model of MSRE. The OpenMC code was also employed to develop the albedo boundary condition at the outer surface of the reactor vessel, which is used in PROTEUS-NODAL calculations. Also, the OpenMC model is used to determine the fraction of the heat generated in the graphite moderator due to gamma heating and neutron slowing down by performing coupled neutron gamma calculations, which is used in the calculations of the thermal hydraulics solver of PROTEUS-NODAL to determine the temperature distribution in the graphite moderator.



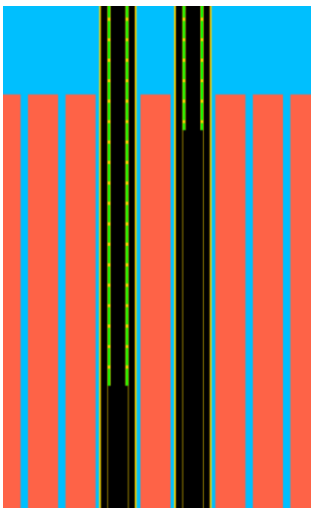
Radial Layout of Core



Radial Layout of Control Region



Axial Layout of Core



Axial Layout of Control Region

Figure 4.1. Developed OpenMC model for MSRE Analysis.

The multigroup cross sections of the MSRE were prepared as a function of the fuel salt and moderator temperatures using the OpenMC code with the ENDF/B-VII.1 data. The multigroup cross sections were generated at a specified temperature and material compositions and converted into the isotopic cross section dataset ISOTXS and the delayed neutron cross section dataset DLAYXS using the utility code GenISOTXS. The multigroup cross sections were generated with the 3-D full core model with the 16-group structure as discussed in Appendix E. The outer core regions (i.e., gap, insulator, and thermal shield) were included in this OpenMC model because they affect the eigenvalue and the neutron spectrum of the outer regions significantly due to neutron reflection. Region-dependent cross sections were generated by dividing the inside the reactor vessel into multiple regions in the radial and axial directions. Figure 4.2 shows the cross section tally regions.

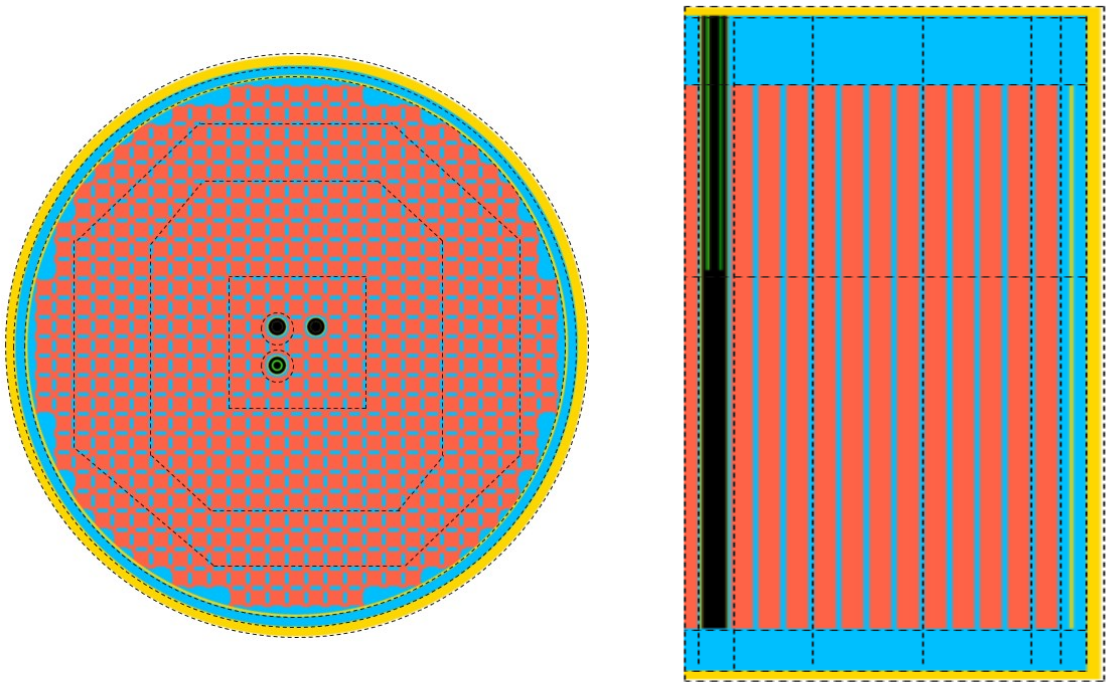


Figure 4.2. Radial (left) and axial (right) configurations of cross section tally regions.

4.2.2. PROTEUS-NODAL Model

The PROTEUS-NODAL models for the MSRE were developed in Cartesian and R-θ-Z geometries. The Cartesian model was included for comparison purposes, although it was found that the boundary perturbation affects the neutron leakage significantly and thus the eigenvalue. PROTEUS-NODAL is based on the variational nodal diffusion method, and it cannot model the large air gap between the reactor vessel and the thermal shield. The PROTEUS-NODAL models

are limited to the reactor vessel region. However, the active core region is next to the reactor vessel in the MSRE, and more than 30% of neutrons leak out from the reactor vessel. Since the insulator and thermal shield are made of approximately 50% steel and 50% water, a large fraction of those leaking neutrons is thermalized in the thermal shield and insulator and reflected into the reactor vessel. In order to account for these reflected neutrons properly without including the regions beyond the reactor vessel, a multigroup albedo boundary condition at the outer surface of the reactor vessel was developed from a separate OpenMC calculation for the outer region of the reactor vessel.

The thermal hydraulics calculations were performed using the thermal hydraulics solver of PROTEUS-NODAL. The moderator temperature and the fuel salt temperature, density, and velocity are calculated and used in the thermal feedback calculations. The core is represented by multiple parallel channels in the axial direction which are connected at the inlet and outlet. The mass flow rates of individual fuel channels are calculated using the equal pressure drop boundary conditions over all channels. The moderator temperature is calculated by considering the heat generation and radial heat conduction in graphite moderator and the heat transfer from graphite moderator to fuel salt. The fractional heat generation rate in graphite due to photon interactions and neutron scattering was determined from a coupled neutron and gamma transport calculation with OpenMC.

Figure 4.3 shows the developed PROTEUS-NODAL models in Cartesian and R-θ-Z geometries for the MSRE core. The Cartesian geometry model can represent the fuel-graphite stringer accurately. However, the outer cylindrical regions (i.e., the unmoderated outer core, core container, down-comer, and reactor vessel) need to be homogenized and approximately represented in Cartesian nodes, which affects the neutron leakage significantly. On the other hand, the R-θ-Z geometry model needs additional homogenization in the inner core region and approximate representation of control rods although it can represent the outer core regions accurately without homogenization.

The albedo boundary condition at the outer surface of the reactor vessel was developed as a multigroup response matrix to account for the reflection of the neutron slowed down in the thermal shield and insulator regions to the core region. The albedo boundary condition with the defined albedo response matrix can be written as

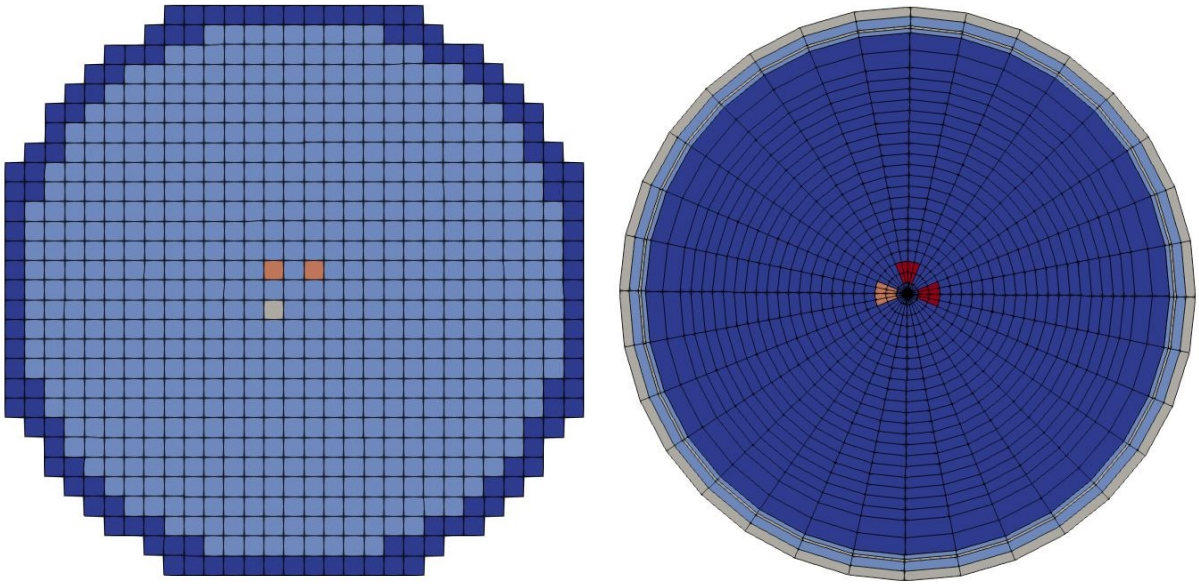


Figure 4.3. PROTEUS-NODAL models of MSRE core in Cartesian (left) and R-θ-Z (right) geometries.

$$\begin{bmatrix} J_1^- \\ \vdots \\ J_g^- \\ \vdots \\ J_G^- \end{bmatrix} = \begin{bmatrix} \beta_{11} & \cdots & \beta_{g1} & \cdots & \beta_{G1} \\ \vdots & \ddots & \vdots & & \vdots \\ \beta_{1g} & & \beta_{gg} & & \beta_{Gg} \\ \vdots & & \vdots & \ddots & \vdots \\ \beta_{1G} & \cdots & \beta_{gG} & \cdots & \beta_{GG} \end{bmatrix} \begin{bmatrix} J_1^+ \\ \vdots \\ J_g^+ \\ \vdots \\ J_G^+ \end{bmatrix}. \tag{4.1}$$

With this boundary condition, the incoming partial currents (J_g^-) at the outer boundary are determined by multiplying the albedo response matrix to the multigroup vector of calculated outgoing partial currents (J_g^+). The elements $\beta_{gg'}$ are constants depending on the composition and the geometry of the outer region to be represented by the albedo response matrix.

The albedo response matrix was obtained from OpenMC Monte Carlo simulations for a 2-D model representing the outer core regions from the reactor vessel to the thermal shield of the MSRE as shown in Fig. 4.4. The columns of the albedo response matrix were determined by solving a fixed source problem for each energy group with a uniform mono-directional neutron source was specified on the inner surface of the reactor vessel and the incoming partial currents at the outer surface of the reactor vessel were tallied. The calculated 16-group albedo response matrix on the outer surface of the MSRE reactor vessel is given in Appendix F.

For comparison purposes, a simplified albedo boundary condition was also examined by approximating the response matrix by a diagonal matrix as

$$\begin{bmatrix} J_1^- \\ \vdots \\ J_g^- \\ \vdots \\ J_G^- \end{bmatrix} = \begin{bmatrix} \tilde{\beta}_1 & & & & \\ & \ddots & & & \\ & & \tilde{\beta}_g & & \\ & & & \ddots & \\ & & & & \tilde{\beta}_G \end{bmatrix} \begin{bmatrix} J_1^+ \\ \vdots \\ J_g^+ \\ \vdots \\ J_G^+ \end{bmatrix}. \tag{4.2}$$

With this approximation, the incoming partial current for each group is simply determined from the outgoing partial current of the same group only. By comparing Eq. (4.1) and (4.2), the following relation can be obtained

$$\tilde{\beta}_g = \frac{J_g^-}{J_g^+} = \beta_{gg} + \sum_{g' \neq g}^{G} \beta_{gg'} \frac{J_{g'}^+}{J_g^+}. \tag{4.3}$$

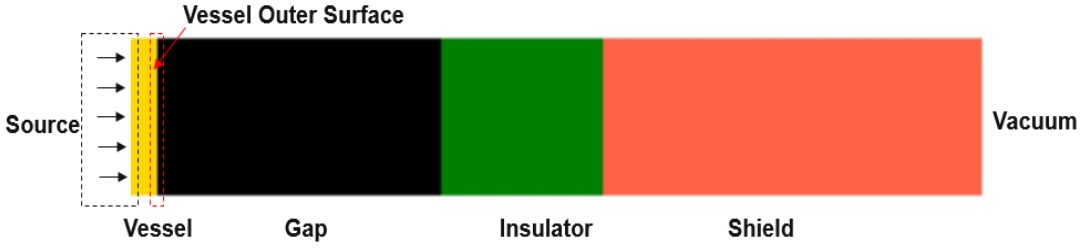


Figure 4.4. OpenMC model for driving albedo response matrix for MSRE ex-vessel regions.

If the reflection is dominant to slowing down in the ex-vessel regions, $\tilde{\beta}_g$ can be approximated by the diagonal elements β_{gg} of the albedo response matrix. However, since the neutron slowing down in the thermal shield and insulator is significant, the group-wise albedo values $\tilde{\beta}_g$ depend on the outgoing partial currents and thus the problem to be solved, whereas the albedo response matrix in Eq. (4.1) depends on the ex-vessel regions only. The diagonal albedo matrix was calculated from OpenMC calculations with the full core model including the ex-vessel regions, as shown in Fig. 4.1, by tallying the outgoing and incoming partial currents on the outer surface of the reactor vessel. A set of 16-group albedo values were calculated for each of the radial, bottom, and top boundary surfaces. Appendix F provides the calculated diagonal albedo matrices of the ^{235}U and ^{233}U fuel cores with control rods at the fully withdrawn position. It is noted that a few albedo values for the three lowest energy groups are greater than one. These unphysical values are due to the slowing down of leaked neutrons in the ex-vessel region.

The PROTEUS-NODAL core models in Cartesian and R-θ-Z geometries and the albedo boundary conditions were assessed using steady state core configurations of the ^{235}U and ^{233}U fuel

cores of MSRE. The effects of the ex-core regions were also estimated by performing an additional OpenMC calculation without the ex-vessel regions. The exclusion of the ex-vessel regions increased the net leakage from the reactor vessel from 18.4% to 21.9% and thus reduced the multiplication factor from 1.04826 to 1.04255. The resulting effective multiplication factor and leakage fraction of PROTEUS-NODAL were compared with the reference results from the OpenMC full core model including the ex-vessel regions and the results are summarized in Table 4.2. The Cartesian geometry model introduces a significant error in the leakage fraction because of the approximate representation of the radial boundary and the associate homogenization. For both albedo boundary conditions, the Cartesian geometry model underestimates the eigenvalue by ~750 pcm for the ^{235}U core and by ~1100 pcm for the ^{233}U core. The leakage fraction error depends on the employed boundary condition. With the approximate diagonal albedo matrices generated separately for each of the ^{235}U and ^{233}U cores, the leakage fraction error is 8.9% for the ^{235}U core and 8.5% for the ^{233}U core. The full albedo response matrix reduces the leakage fraction error from 8.9% to 4.4% for the ^{235}U core and from 8.5% to 5.1% for the ^{233}U core.

Table 4.2. Comparison of Effective Multiplication Factor and Leakage Fraction of MSRE.

Core	Code	Core Model	Albedo Matrix	k-eff	Diff. (pcm)	Leakage Fraction	Diff. (%)
^{235}U	OpenMC	-	-	1.04826	-	0.184	-
	PROTEUS -NODAL	Cartesian	Diagonal	1.04076	-750.4	0.200	8.89
			Full	1.04107	-719.2	0.192	4.35
		R-θ-Z	Diagonal	1.04801	-25.0	0.191	3.78
			Full	1.04818	-8.3	0.184	0.05
^{233}U	OpenMC	-	-	1.07103	-	0.216	-
	PROTEUS -NODAL	Cartesian	Diagonal	1.06020	-1082.5	0.234	8.54
			Full	1.06014	-1089.4	0.227	5.13
		R-θ-Z	Diagonal	1.07200	97.1	0.221	2.62
			Full	1.07208	104.6	0.214	-0.55

Even with the additional homogenization in the inner core region, the R-θ-Z geometry model yields very accurate eigenvalue results. The eigenvalue is underestimated by less than 25 pcm for the ^{235}U core and overestimated by ~100 pcm for the ^{233}U core. Both the full albedo response matrix and the approximate albedo matrix yield comparable eigenvalue results. However, for the leakage fraction, the approximate diagonal albedo matrices yield 3.8% error for the ^{235}U core and

2.6% for the ^{233}U core while the full albedo response matrix yields very accurate results within 1% error (0.05% for the ^{235}U core and 0.55% for the ^{233}U core). Based on these observations, the MSRE experiments were analyzed using the R-θ-Z geometry core model and the full albedo response matrix.

4.3. Analysis of Static Experiments

Validation tests of the PROTEUS-NODAL code were performed using the static experiments performed at the ^{235}U and ^{233}U cores of the MSRE. The experimental data were obtained from Ref. [8, 77, 83]. These data include the control rod worth, the isothermal temperature coefficient, and the reactivity loss due to fuel circulation in the ^{235}U and ^{233}U cores. The measured reactivity values have an uncertainty of 2% that is attributed mainly to the reactor period measurement, which has the most contribution to the experimental uncertainty [77].

4.3.1. Control Rod Worth

The MSRE was designed to be operated with three control rods made from gadolinium in the form of Gd_2O_3-Al_2O_3 (70-30 wt.%) ceramic. The control rod worth was determined from the reactivity change caused by the control rod movement. In the experiment, the reactivity was calculated from the in-hour equation using the precalculated kinetics parameters and the inverse of the measured stable period of the reactor. The period measurements were generally made in pairs. The rod being calibrated was first adjusted to make the reactor critical at about 10.0 W. Then it was pulled a prescribed distance and held there until the power had increased by about two decades. Then, the rod was then inserted to bring the power back to 10.0 W, and the measurement was repeated. The stable period was determined by averaging the slopes of the two curves, which usually agreed within about 2%. The observed periods were generally in the range of 30 to 150 seconds. The differential worth measurements were made with the fuel pump off, and theoretical corrections were applied to these measurements to account for the rod shadowing effect and put all the measurements relative to the initial critical concentration. In addition, the measurements were repeated while the fuel was circulating in the system, and the inverse of the measured period was used to calculate the reactivity using the modified in-hour equation for flowing fuel. Similar values of the control rod worth were obtained. However, the uncertainty in the calculated reactivity was higher because of the flow effect on the reactivity and the existence of the circulating gas [77, 83].

The control rod worth for stationary fuel was calculated at a temperature of 922 K using the OpenMC and PROTEUS-NODAL codes. The worth of one, two, and three (bank) control rods were calculated at different uranium loadings by taking the reactivity difference of the desired control rods at fully withdrawn and fully inserted positions. Table 4.3 compares the calculated values of OpenMC and PROTEUS-NODAL with the measured values. The OpenMC results agree well with the measured values within one measurement uncertainty (1σ) except for the cases of control rod No. 1 (Cases 1 and 8), for which the deviation is between 1σ and 2σ. The PROTEUS-NODAL results also show good agreement with the measurements within 2σ except for Cases 1 and 8. The results of Cases 1 and 8 deviate from the measurements by 3.6σ and 6.6σ, respectively. Figure 4.5 compares the differential and integral worth of the control rod No. 1 (regulating rod) calculated with OpenMC and PROTEUS-NODAL with the measured data, respectively. The OpenMC results agree well with the measurements. However, the PROTEUS-NODAL results show larger deviations due to the homogenization error in the control rod region and the diffusion theory limitations.

Table 4.3. Measured and Calculated Values (%∂k/k) of Control Rod Worth.

Case	Core	Control Rods	^{235}U (kg)	Measurement	Calculation	
					OpenMC	PROTEUS
1	^{235}U	1	65.25	2.260 ± 0.064	2.291 ± 0.014	2.491
2		1 and 2	67.94	4.099 ± 0.116	4.045 ± 0.014	3.928
3			69.94	3.975 ± 0.112	3.946 ± 0.014	3.854
4			71.71	4.075 ± 0.115	3.892 ± 0.013	4.071
5		Bank (1, 2, and 3)	67.94	5.596 ± 0.158	5.744 ± 0.014	5.497
6			69.94	5.611 ± 0.159	5.607 ± 0.015	5.394
7			71.71	5.570 ± 0.158	5.552 ± 0.014	5.674
8	^{233}U	1		2.580 ± 0.073	2.713 ± 0.015	3.064
9		Bank		6.900 ± 0.195	6.776 ± 0.015	7.183

4.3.2. Temperature Coefficients

For the ^{235}U fuel core, the effect of temperature on reactivity was measured in three separate experiments in which the electric heaters were adjusted to change the system temperature slowly while the critical position of the regulating rod was recorded. The experiment gave the isothermal temperature coefficient (ITC), which represents the sum of the fuel salt and the graphite temperature coefficients. The change in the critical position of the regulating control rod was

converted into reactivity using the control rod worth curves. The ITC was measured for three ^{235}U loadings with a temperature range from 895 K to 936 K, and the ITC values were ranging from -13.032 pcm/K to -13.41 pcm/K as shown in Fig. 4.6. The experiment at 71.7 kg loading shows a lower slope below about 895 K because of the increasing amount of helium bubbles in the circulating salt as the temperature was lowered, which reduces the amount of fuel salt in the core. Thus, the slope of the lower part of the curve was not considered in calculating the temperature feedback coefficient [77].

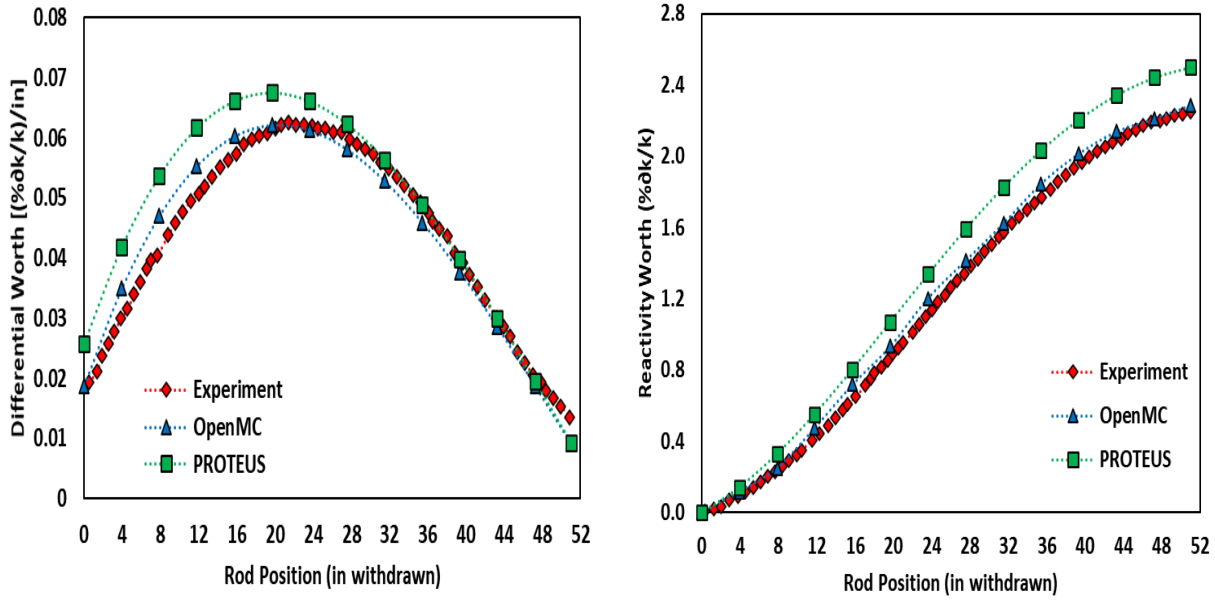


Figure 4.5. Differential (left) and integral (right) worth of control rod No. 1 of MSRE.

The fuel salt temperature coefficient was measured by keeping the fuel salt in circulation and heating the stagnant coolant salt in the secondary loop. Then, the coolant pump was started, introducing a hot slug into the heat exchanger and subsequently into the core. The change in reactivity was attributed entirely to the change in temperature of the fuel and the fuel temperature coefficient was -8.46 ± 1.26 pcm/K. The graphite temperature coefficient was assumed to be equal to the difference between the ITC and the fuel salt temperature coefficient [77].

For the ^{233}U fuel core, the effect of temperature on reactivity in the MSRE was measured three times by slowly changing the temperature from 908 K to 936 K. There was considerable variation found among the measured values in these experiments due to the presence of helium bubbles in the circulating fuel. In the first measurements, the fuel pump was operating to circulate the heated fuel salt throughout the core. Then, the pump was turned off to allow gas bubbles that were

circulating with the fuel salt to float up out of the core. The measured ITC with the pump off was -13.95 pcm/K and -12.42 pcm/K with the circulating fuel. The difference is due to a large amount of gas bubbles circulating with the salt, and the fact that the void fraction is increasing with decreasing temperature. The fuel salt density change due to the dependence of gas tends to offset the salt density change with temperature by reducing the magnitude of the reactivity effect.

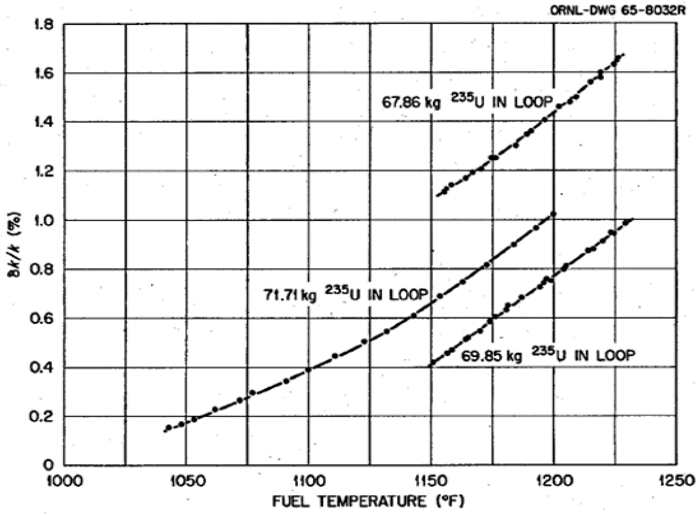


Figure 4.6. Effect of core temperature on reactivity [77].

In the second measurement, the ITC measurement was repeated. The results with the pump off were the same as the first measurement, but -13.32 pcm/K with the circulating fuel. In the third measurement, the ITC measurement was performed at a reduced fuel circulation rate without turning the pump turned off to reduce the void fraction in the fuel salt by allowing the gas bubbles to float up out of the core. The measured ITC value with the pump running at reduced speed was -15.3 pcm/K. This indicates that the effects of gas had not been eliminated in the first and second measurements by just stopping the pump. The ITC value of the third experiment was considered as the final measured value of the ITC. The individual contribution of the fuel salt to the overall temperature coefficient was not measured experimentally [83].

Using the OpenMC and PROTEUS-NODAL codes, the temperature coefficients of the ^{235}U and ^{233}U fuel cores were calculated. The temperature was varied between 850 K and 1000 K, and the temperature coefficients were determined by the linear regression of the reactivity changes with temperature, as shown in Fig. 4.7 for the ^{235}U and ^{233}U cores, respectively. The graphite temperature coefficient was determined from the difference between ITC and fuel temperature coefficient.

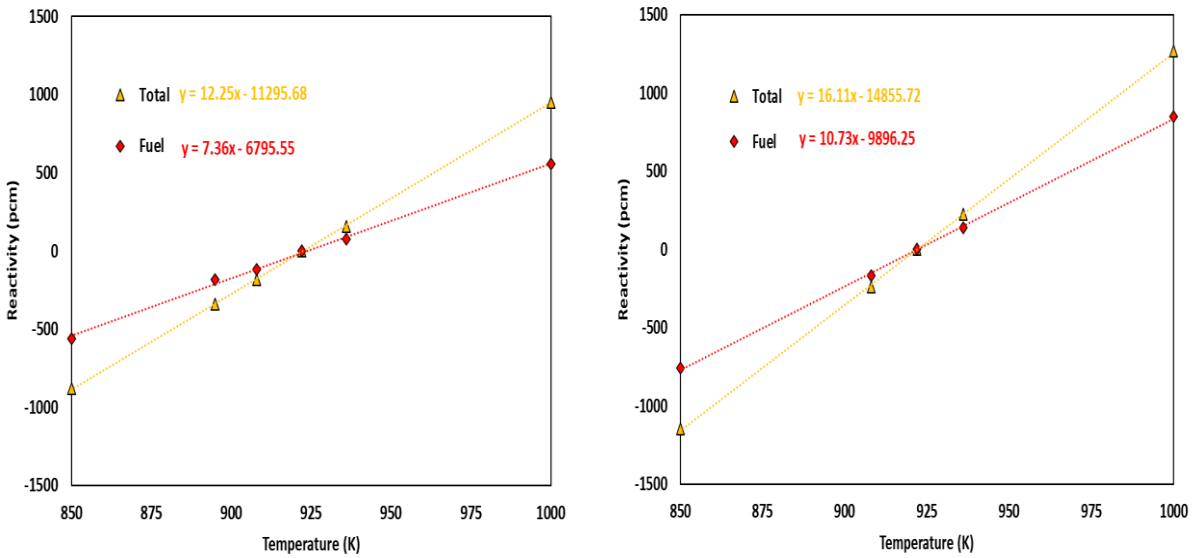


Figure 4.7. Reactivity feedback coefficients of ^{235}U core (left) and ^{233}U core (right) calculated with OpenMC.

Table 4.4 compares the calculated temperature coefficients with the measured values. The PROTEUS-NODAL results agree very well with the OpenMC results for all temperature coefficients as expected. For the ^{235}U fuel core, all the calculated temperature coefficients of PROTEUS-NODAL show good agreements with the measurements within 2.5 measurement uncertainties (2.5σ). The calculated ITC for the ^{233}U fuel core with PROTEUS-NODAL also agrees with the measured value within 1.7σ.

Table 4.4. Reactivity Feedback Coefficients (pcm/K) of MSRE.

Core	Temperature Coefficient	^{235}U (kg)	Measurement	Calculation	
				OpenMC	PROTEUS
^{235}U	ITC	67.86	-13.410 ± 0.379	-13.003 ± 0.500	-12.710
		69.85	-13.032 ± 0.369	-12.186 ± 0.525	-12.117
		71.71	-13.140 ± 0.372	-12.245 ± 0.488	-12.229
	Fuel	71.71	-8.460 ± 1.260	-7.358 ± 0.508	-7.251
	Graphite	71.71	-4.680 ± 1.314	-4.887 ± 0.733	-4.978
^{233}U	ITC	-	-15.300 ± 0.433	-16.115 ± 0.505	-16.031
	Fuel	-		-10.733 ± 0. 531	-10.506
	Graphite	-		-5.382 ± 0.733	-5.525

4.3.3. Reactivity Loss by Fuel Circulation

In PROTEUS-NODAL, the reactivity loss by fuel circulation for MSRE was calculated by the difference of the effective delayed neutron fractions between stationary and flowing fuels. During

the zero-power experiment of the MSRE with ^{235}U fuel salt, the critical rod position and period measurements were made after each addition of the fuel salt, with the fuel pump off and with the circulating fuel. The reactivity change between stationary and circulating fuels was measured to be 212.0 ± 4.0 pcm, which is caused by the loss in the delayed neutron fraction due to the circulation of the fuel salt. For the ^{233}U fuel core, the effect of the fuel circulation on the reactivity was not measured [77, 83]. Table 4.5 presents the measured and calculated reactivity losses due to fuel circulation for the ^{235}U and ^{233}U cores. The PROTEUS-NODAL result for the ^{235}U fuel core agrees reasonably well with the measured value with an error of 5.1%.

Table 4.5. Loss of Delayed Neutron Fraction in Steady State Operation of MSRE.

Core	Measurement/Code	Total	1	2	3	4	5	6
^{235}U	Experiment	212.0						
	PROTEUS	222.9	15.2	74.4	56.0	68.8	8.0	0.4
^{233}U	Experiment	-						
	PROTEUS	110.4	16.1	40.4	24.6	26.8	2.5	0.1

Table 4.6 compares the effective delayed neutron fraction of stationary and circulating fuels and the prompt neutron generation time calculated with PROTEUS-NODAL. The effective delayed neutron fraction of the ^{233}U core is much lower than the ^{235}U core since the delayed neutron fraction of the thermal fission of ^{233}U is 270 pcm, while it is 650 pcm for ^{235}U. However, the fractional loss of delayed neutrons by fuel circulation is about 35% for both ^{235}U and ^{233}U cores since the flow rates are similar in both cores.

Table 4.6. Calculated Kinetics Parameters of PROTEUS-NODAL.

Core	β_{eff} (pcm)	β_{circ} (pcm)	β_{circ}/β_{eff}	β_{loss} (pcm)	β_{loss}/β_{eff}	Λ (ms)
^{235}U	653.7	430.8	0.66	222.9	0.34	0.245
^{233}U	293.8	183.3	0.62	110.4	0.38	0.379

The forward and adjoint delayed neutron precursor concentrations were calculated for the stationary and flowing ^{235}U fuels. Figure 4.8 compares the axial distributions of precursor concentrations of the ^{235}U core for stationary and flowing fuels. The precursor concentration of stationary fuel is symmetric about the core mid-plane for all the six precursor groups and shows a peak in the bottom and the top region because of the slowing down of fast neutrons leaking out of the core. On the other hand, for the flowing fuel, the distribution of the precursor concentration is almost uniform for the precursor group 1 because of its relatively long half-life, is top skewed for

precursor groups 2 to 5 due to the upward flow and is similar to the stationary fuel concentration for group 6 due to the short half-life. Because of the upward flow, flowing fuel shows a smaller peak in the bottom region but a larger peak in the top region than stationary fuel. Figure 4.9 compares the axial distributions of adjoint precursor concentrations (i.e., the importance of precursor concentration to reactivity). The adjoint precursor concentrations of stationary fuel are also symmetric about the core mid-plane for all the six precursor groups, and they are similar to each other since the six delayed neutron spectra are similar. On the other hand, in flowing fuel, the axial distributions of adjoint precursor concentrations are bottom skewed because of their downward flow. The magnitude of adjoint precursor concentration increases with increasing precursor group number as the half-life decreases. It also shows a peak in the bottom and the top region due to the massive change in the fuel salt volume and flow rate.

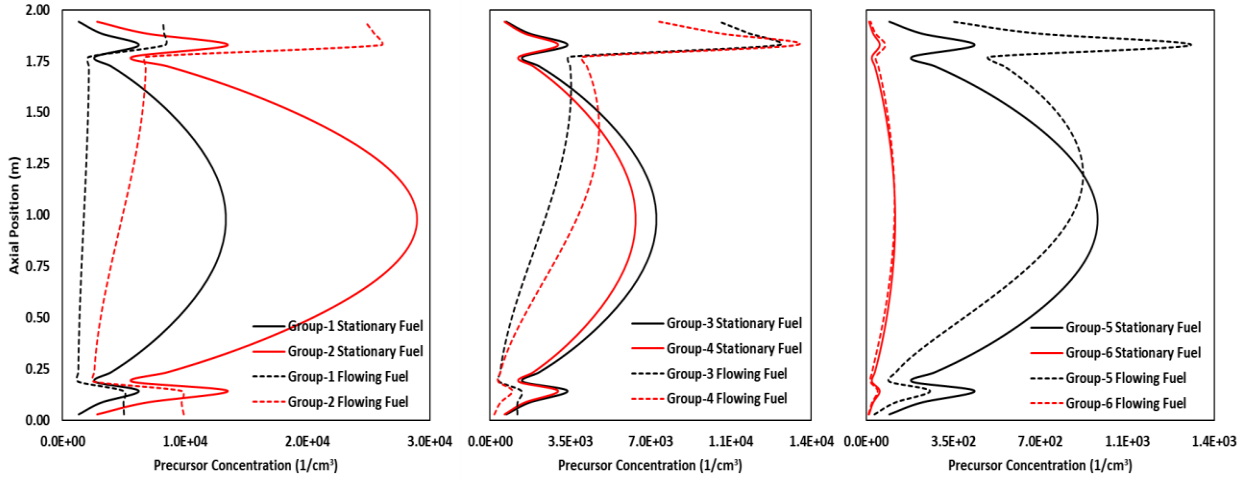


Figure 4.8. Axial distributions of delayed neutron precursor concentrations.

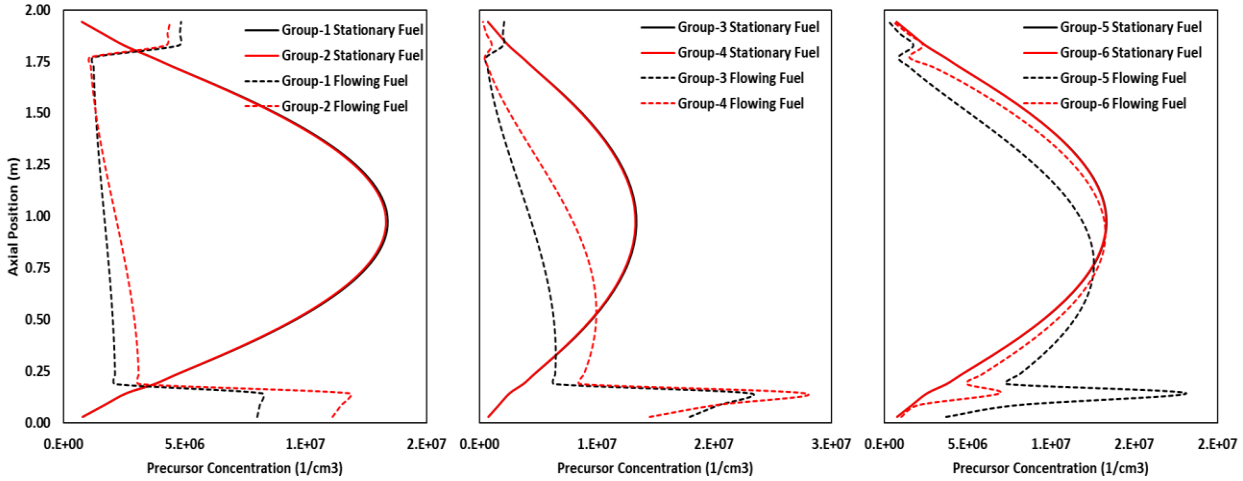


Figure 4.9. Axial distributions of adjoint delayed neutron precursor concentrations.

4.3.4. Temperature Distribution

The fuel salt temperature, density, and velocity distributions, and the graphite temperature distribution within the core were determined for the thermal feedback calculation. The thermal hydraulics calculations for the MSRE were performed using the thermal hydraulics solver of PROTEUS-NODAL. The core was divided into multiple parallel axial channels, which are connected at the inlet and outlet. The mass flow rates of individual fuel channels were calculated using the equal pressure drop boundary conditions over all channels.

The moderator temperature was calculated by considering the heat generation and radial heat conduction in graphite moderator and the heat transfer from graphite moderator to fuel salt. The fractional heat generation rate in graphite due to photon interactions and neutron scattering was determined from a coupled neutron and gamma transport calculation with OpenMC. Unit cell and full core calculations were performed with and without gamma ray transport. Table 4.7 summarizes the resulting fractional heat deposits in the moderator. The full core model yields a slightly higher heating fraction in the moderator than the unit cell model since the photons and neutrons produced in the unmoderated top and bottom regions contribute to the heating in the moderator. The full core calculation with gamma transport yields a fractional heat deposit in graphite of 5.5%.

Table 4.7. Fractional Heat Deposits in Moderator of MSRE Core.

Computational Model	Fractional Heat Deposit in Graphite (%)	
	No Gamma Transport	Gamma Transport
Unit Cell	1.182 ± 0.001	4.553 ± 0.001
Core	1.836 ± 0.001	5.467 ± 0.001

Figure 4.10 shows the distribution of heat deposit at the core mid-plane obtained from the coupled neutron and gamma transport calculation with OpenMC for the full core model. The heat deposit is the highest in the fuel salt channels in the central core region except for a few central channels around the control rods. Also, Fig. 4.10 shows the distributions of fractional heat deposits in fuel and moderator over energy groups. The fuel salt heating is mainly due to thermal fission and the graphite heating is mainly by the neutrons and photons around 1 MeV.

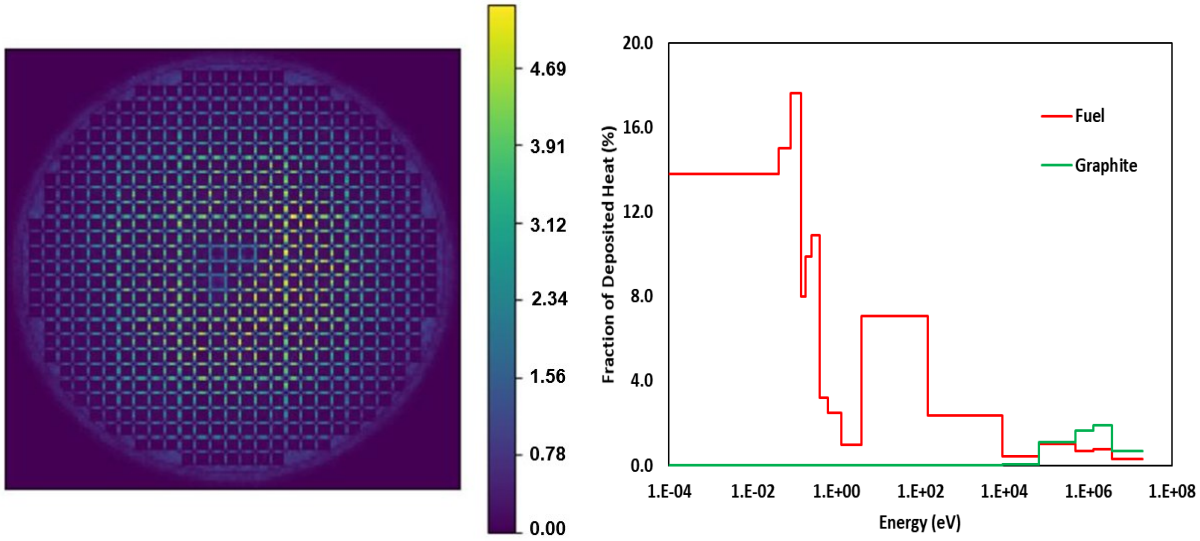


Figure 4.10. Distribution of heat deposit rate (kW) at core mid-plane (left) and fractional heat deposits in fuel and moderator over energy groups (right).

Figure 4.11 compares the axial temperature distributions of fuel and moderator in the hottest channel calculated using the thermal hydraulics solver of PROTEUS-NODAL and compared with the ORNL design calculation results [84]. It also presents the temperature field in the core obtained from PROTEUS-NODAL calculation. The temperature distributions of PROTEUS-NODAL agree well with the ORNL results for both fuel and graphite temperatures. The axial temperature distribution in graphite shows a slightly higher peak than the ORNL distribution. This might be due to the different graphite heating fractions used in the calculations.

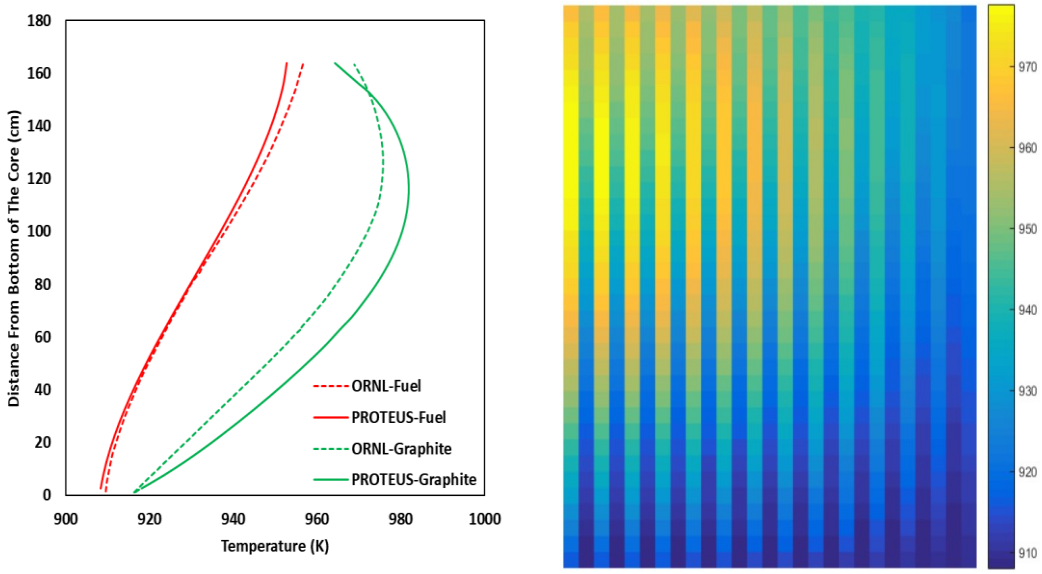


Figure 4.11. Comparison of axial temperature distributions in hottest channel (left) and PROTEUS-NODAL temperature field in core (right).

4.4. Protected Pump Transients

In the protected pump transients of the ^{235}U fueled MSRE, the reactor was operated at a low power level, and the fuel flow rate in the primary loop was increased or decreased by adjusting the speed of the fuel pump. The control rod position was adjusted to maintain a constant power level by compensating for the reactivity changes due to the redistribution of delayed neutron precursors. The recorded control rod positions were converted into reactivity using the integral control rod worth curves. The PROTEUS-NODAL calculation was performed for the specified fuel flow rate as a function of time under the assumption of uniform fuel temperature and velocity. The fuel flow rates in Fig. 3.23 were used as the PROTEUS-NODAL input data for pump startup and coast down, which were deduced from the coolant flow rate and the fuel and coolant pump speeds. The reactivity loss was calculated at each time step by the difference in the effective delayed neutron fraction between stationary and flowing fuels. Figure 4.12 compares the compensated reactivity transient calculated by PROTEUS-NODAL with the measured data for the protected pump startup and coast down.

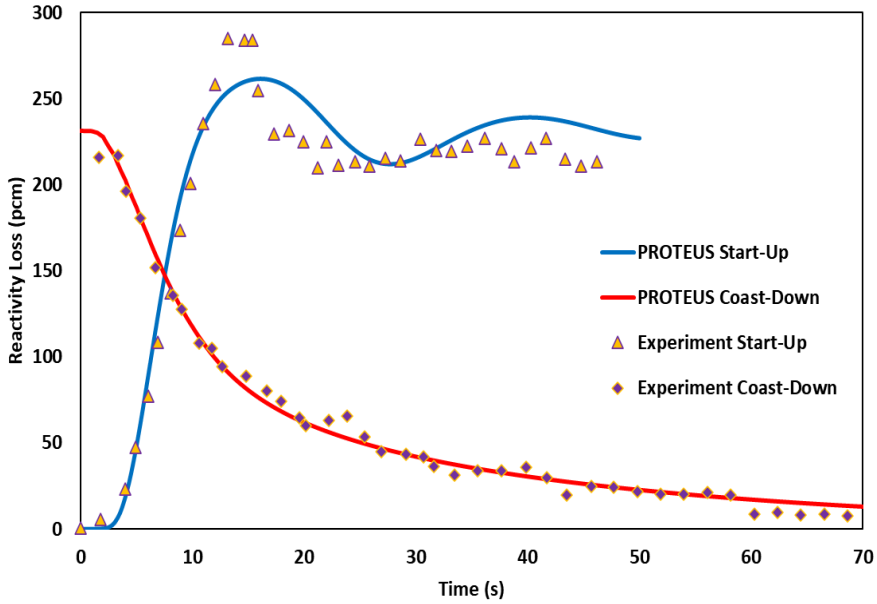


Figure 4.12. Compensated reactivity during protected pump startup and coast down transients.

During the protected pump startup transient, the fuel starts to flow outside the core, which leads to reactivity loss because of the decay of delayed neutron precursors outside the core or at regions of lower importance. The reactivity loss increases with increasing flow velocity. Oscillatory behavior is observed in the compensated reactivity because of recirculated precursors. The positive

reactivity effect of the recirculated precursors entering the core is clearly seen 13 s after pump startup. The difference between the calculated reactivity loss of PROTEUS-NODAL and the measurement can be attributed to the delay in the response of the reactor control system and the difference in the delayed neutron data. During the protected pump cost down transient, the reactivity increases because more precursors decay in the core region with decreasing the fuel velocity. Therefore, the compensated reactivity continues to decrease and reaches almost zero when all the precursors decay in the core. It can be seen that the PROTEUS-NODAL solution follows the measured data very well.

4.5. Natural Circulation Test

The natural circulation test in the ^{233}U fueled MSRE core was analyzed to validate the coupled neutronics and thermal hydraulics solvers of the PROTEUS-NODAL code. The purpose of the natural circulation test was to determine the characteristics of heat removal from the MSRE fuel system by natural convection. At the beginning of the transient, the reactor was operated at low power of about 4.1 kW with a limited fuel flow rate. After that, the core inlet temperature was decreased by increasing the heat removal rate in the air radiator of the secondary loop for six hours. The reactor was controlled entirely by the inherent thermal feedback of the system during this transient and no adjustments were made to the control rods. In this test, there is no data recorded for the fuel salt inlet mass flow rate or the heat removal rate by the secondary loop. Therefore, the inlet mass flow rate of the fuel salt was deduced using the recorded data of power and core inlet and outlet temperatures and the transit time of the fuel salt in the core. The recorded data over the transient period is provided in Fig. 4.13 [83].

The natural circulation test was simulated with the PROTEUSNODAL code by assuming that the reactor was in steady state conditions at 4.1 kW power. The core inlet temperature and the deduced inlet mass flow rate of the fuel salt were used as the boundary conditions for thermal hydraulics calculations during the transient period without considering the heat removal in the external loop. Figure 4.14 compares the calculated power evolution with the measured data. The PROTEUS-NODAL solution follows the measured power evolution very well. The observed deviation at the beginning of the transient could be due to the assumed initial steady state. This result indicates that PROTEUS-NODAL is able to reproduce the measured power level during this long transient for a given inlet temperature and flow rate.

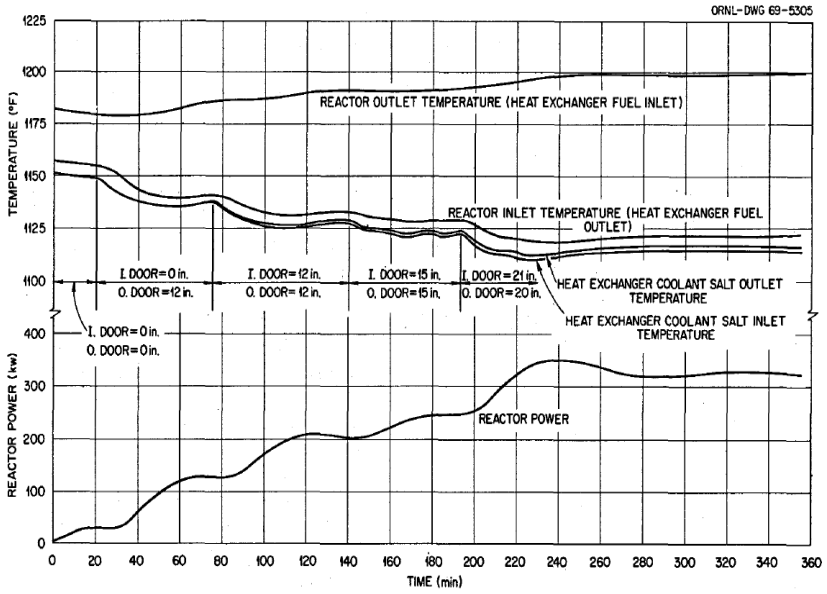


Figure 4.13. Recorded power and temperatures during natural circulation test of MSRE [83].

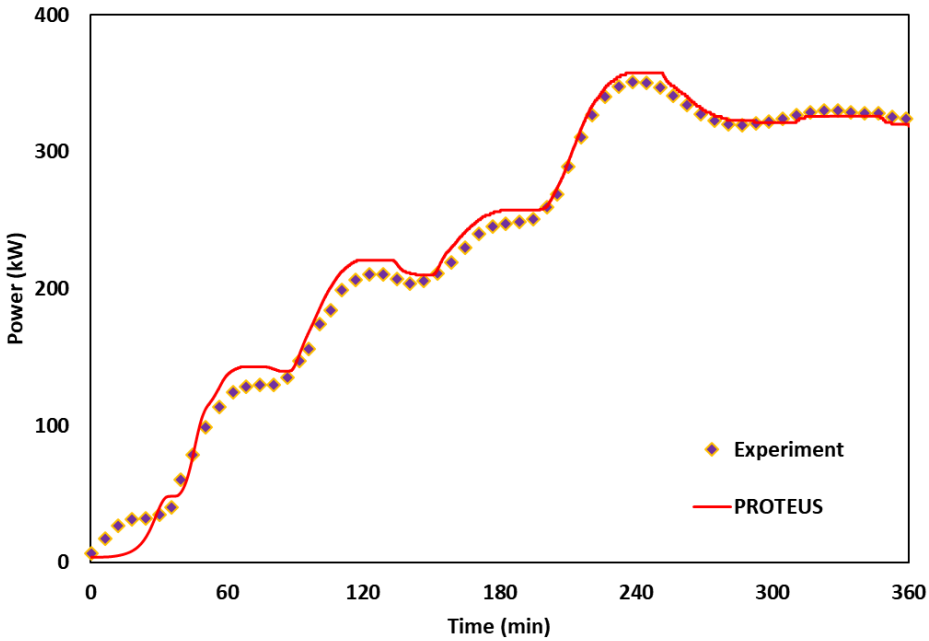


Figure 4.14. Comparison of power evolution during natural circulation test of MSRE.

4.6. Summary

Validation tests of PROTEUS-NODAL were performed using the measurements in the ^{235}U and ^{233}U cores of the MSRE. The temperature coefficients, the reactivity loss by fuel circulation, the protected pump startup and coast down transients at zero power, and the natural circulation test of the MSRE were analyzed. The calculated temperature coefficients of PROTEUS-NODAL show good agreements with the measurements within 2.5σ (of measurement uncertainties) for the ^{235}U

fuel core and 1.7σ for the ^{233}U fuel core. The calculated reactivity loss due to fuel circulation for the ^{235}U core agreed well with the measurement with an error of 5.1%. The analysis of the protected pump startup and coast down transients at zero power showed that the PROTEUS-NODAL calculation with a given inlet flow rate could follow the time-dependent reactivity change due to delayed neutron precursor drift. The analysis of the natural circulation test for six hours showed that the PROTEUS-NODAL code could reproduce the measured power evolution due to thermal feedback for given inlet temperature and flow rate.

Chapter 5. Adaptive Time Stepping for Transient Analyses

In many transients, the power change rate varies significantly throughout a transient duration. For example, in an UTOP transient of the MSFR benchmark, the power increases rapidly to its peak value within 1.0 ms, decreases slowly and then remains almost constant. A small time-step size is necessary to represent the rapid initial transient accurately, but it becomes inefficient in modeling the slow transient in the later stage. Thus, the use of a fixed time-step size throughout the whole duration will yield a solution that is not sufficiently accurate or make the computation inefficient, especially for the multiphysics simulation with a coupled neutronics and thermal-fluidic code system. In addition, for slow transients that last many hours such as the natural circulation experiment of the MSRE, an optimum time-step size is necessary to avoid an excessively large computational time while maintaining the desired accuracy, but it is not obvious to select an optimum time-step size. Therefore, adaptive time stepping is necessary to achieve computational accuracy and efficiency at the same time.

Motivated by the needs for practical transient analyses of fast and thermal spectrum MSRs, an adaptive time-step selection scheme is developed based on the control theory. In this scheme, the time-step size is selected based on the estimated local solution errors during the transient. This will help in avoiding the loss of accuracy that might be introduced due to a large time-step size and in eliminating the computational inefficiency due to unnecessarily small time-step size. In addition, variation of time-step size might be necessary to obtain an accurate solution for nonlinearly coupled neutronics and thermal-fluidic simulation. Adaptive time-stepping has received much attention in various engineering fields in order to select an optimum time-step size within a specified solution error tolerance. The theory of automatic control is widely used for adaptive numerical time-stepping in solving ordinary differential equations (ODE's) by numerical integration methods to minimize the computational efforts while maintaining the accuracy of the solution. The efficiency of an adaptive time-stepping method depends on the discretization scheme and the problem properties. The basic idea of the control theory for adaptive time-stepping is to select the time-step size while monitoring the local error of the problem solution. The local error

is estimated and maintained within the specified accuracy requirement by selecting the next time-step size according to a certain control algorithm. Adaptive time-stepping using control theory was studied and applied thoroughly in several engineering fields. Also, other adaptive schemes were applied in the field of nuclear reactor physics. These schemes are discussed very briefly in this section.

The work of Soderlind and Gustafsson was focused on control theory applications to automate time-step size control techniques for numerical integration methods of ODE's with explicit and implicit Runge-Kutta methods [85, 86, 87, 88, 89]. Additional applications of more advanced controllers and digital filters can be found in [90, 91 92]. Also, Ref. [93, 94] investigated the use of control strategies for time-step selection and convergence rate improvement of non-linear iterative processes in 2D coupled viscous flow and heat transfer by utilizing proportional integral derivative controllers to select the time-step size. More applications of the control theory for time-step size control in other fields can be found in Ref. [95, 96]. Recently, the control theory was used in the field of reactor physics to determine the time-step size for nuclide depletion calculations and the quasi-static method. In Ref. [97], the control theory approach was adopted to determine the time-step size of nuclide depletion simulations for 2D lattices with strong absorbers, which require a small depletion step size. Another application was introduced in Ref. [98], where an adaptive selection of the time-step size was employed for the quasi-static method for the neutron transient analysis. A different approach was introduced in Ref. [99, 100, 101], where adaptive time-stepping schemes were suggested for the neutron diffusion equation as well as neutron transport equation by estimating the local truncation error from the second derivative of the neutron flux with a finite difference approach. A similar approach was adopted in Ref [102], where the alpha eigenvalue problem was used to estimate the second derivative of the neutron flux. Also, an adaptive time-step approach was used in Ref. [103] for the quasi-static method and the neutron diffusion equation by doubling or halving the time-step size based on the estimated solution error.

In this Chapter, the theory and the derivation of a time-step selection scheme based on the control theory are discussed first. Then, the adaptive time-step selection algorithm introduced in the PROTEUS-NODAL code for transient analysis of MSRs is presented. This is followed by application and numerical examples to test the robustness of the implemented algorithm using the previously analyzed transient scenarios of the MSFR benchmark and the MSRE experiment with

comments and suggestions on selecting the controller parameters implemented in PROTEUS-NODAL.

5.1. Truncation Error

The implicit Euler method (backward differencing) is typically employed in discretizing the TFSP as discussed in Chapter 2. The truncation error can be derived by applying Taylor series expansion in time with backward differencing for the neutron scalar flux as

$$\phi(t_{n-1}) = \sum_{n=0}^{\infty} \frac{\left(t_{n-1} - t_n\right)^n}{n!} \left.\frac{\partial^n \phi}{\partial t^n}\right|_{t_n}, \tag{5.1}$$

$$\phi(t_{n-1}) = \phi(t_n) + \left(t_{n-1} - t_n\right) \left.\frac{\partial \phi}{\partial t}\right|_{t_n} + \frac{\left(t_{n-1} - t_n\right)^2}{2} \left.\frac{\partial^2 \phi}{\partial t^2}\right|_{t_n} + o\left(\left(t_{n-1} - t_n\right)^3\right). \tag{5.2}$$

In Eq. (5.1) and Eq. (5.2), the position and energy dependency of the scalar flux are omitted for simplicity. By rearranging the terms of Eq. (5.2), the first derivative can be written with a backward difference formula as

$$\left.\frac{\partial \phi}{\partial t}\right|_{t_n} = \frac{\phi(t_n) - \phi(t_{n-1})}{t_n - t_{n-1}} + \tau_n, \tag{5.3}$$

$$\tau_n = \frac{h_n}{2} \left.\frac{\partial^2 \phi}{\partial t^2}\right|_{t_n} + o\left(h_n{}^2\right), \tag{5.4}$$

where $h_n = \Delta t_n = t_n - t_{n-1}$ is the time-step size, and τ_n is the local truncation error. In Eq. (5.4), τ_n is proportional to h_n and the second derivative of the scalar flux. Thus, as h_n goes to zero, the truncation error approaches zero. This indicates that using a very small time-step size will reduce the truncation error and the discretized solution approaches the true solution. Introducing the backward difference formula in Eq. (5.3) into the neutron diffusion equation in Eq. (2.13) and the delayed neutron precursors equations in Eq. (2.64) will lead to the following

$$\frac{1}{v_g} \frac{\phi_g^n(\vec{r}) - \phi_g^{n-1}(\vec{r})}{h_n} - \nabla \cdot D_g^n(\vec{r}) \nabla \phi_g^n(\vec{r}) + \Sigma_{rg}^n(\vec{r}) \phi_g^n(\vec{r}) = q_g^n(\vec{r}) + \frac{1}{v_g} \tau_n^{\phi_g^n}(\vec{r}), \tag{5.5}$$

$$\frac{C_k^n(\vec{r}) - C_k^{n-1}(\vec{r})}{h_n} + \nabla \cdot [\vec{u}^n(\vec{r}) C_k^n(\vec{r})] + \lambda_k C_k^n(\vec{r}) = \lambda \psi_k^n(\vec{r}) + \tau_n^{C_k^n}(\vec{r}). \tag{5.6}$$

The truncation errors introduced by the discretization of the time derivatives of the neutron flux and the delayed neutron precursor equations in Eq. (5.5) and Eq. (5.6) act as an artificial sink or source depending on the solution of the problem and the time-step size. This term needs to be

controlled and maintained as a constant small value for the whole simulation period of the transient in order to maintain an accurate solution and to avoid the buildup of the error and deviation from the true solution. The local error of the problem can be controlled by selecting a time-step size that ensures the local error is within a specified limit. From Eq. (5.4), the time-step size can be predicted if the second derivative of the scalar flux is known. An adaptive time-stepping approach can be derived to predict the time-step size by limiting the truncation error to be smaller than a prespecified error tolerance ε as

$$\tau_n \leq \varepsilon . \tag{5.7}$$

Then, with the estimated second derivative of the scalar flux, the time-step size can be calculated by fulfilling the above criterion as

$$h_n = 2\varepsilon \left[\left. \frac{\partial^2 \phi}{\partial t^2} \right|_{t_n} \right]^{-1} . \tag{5.8}$$

Eq. (5.8) raises two fundamental issues for the prediction of the time-step size regarding the stability and accuracy of the approach. The first one is how to estimate the second derivative accurately. The second one arises from the dependency of the second derivative on the space and energy, so constraint on the truncation error must be selected accordingly. References [99, 100, 101] investigated the estimation of the second derivative and the selection of the constraint on the truncation error for neutron diffusion equation and point kinetics equation. Later in this Chapter, the selection of the tolerance level and estimating the second derivative will be discussed.

5.2. Control Theory Approach

The main objective of the control theory is to develop an algorithm governing the application of system inputs to drive the system to an optimal state while ensuring control stability level. In the following, the elementary local error control algorithm will be revisited. Then, the control theory terms are defined from a feedback control point of view for the discrete-time integral (I), proportional-integral (PI), and proportional-integral-derivative (PID) controllers [90, 91]. The step size selection can be considered as a control problem as in Fig. 5.1. An estimate of the local error is compared to a user-specified accuracy requirement, and the result is fed back and used to determine a new time-step size. In this way, the controller should keep the local error close to a specified tolerance level [87].

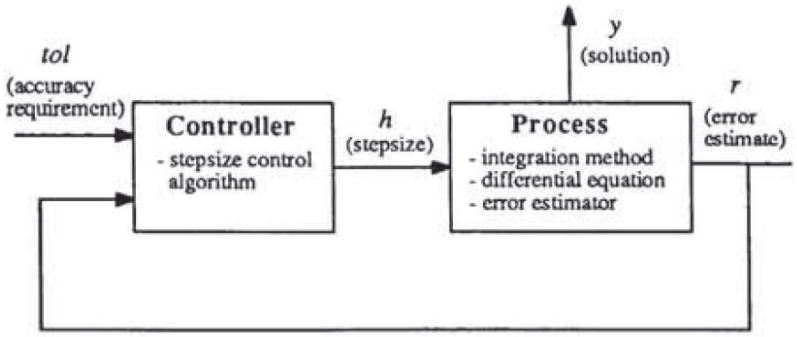


Figure 5.1. View of feedback control system for time-step size selection [87].

5.2.1. Basic Controllers

For an approximate solution, there is a possibility of deviation of that solution from the true or the exact solution. This deviation is called the error which represents how the approximate solution fails to predict the true solution. As an example, consider an initial value problem as in Eq. (5.9) has a true solution y^{True} and an approximate or numerical solution y^{Num}

$$\frac{dy}{dt} = f(y), \qquad y(0) = y_0, \qquad t > 0. \tag{5.9}$$

The local error can be defined as the deviation of the numerical solution from the exact solution at a single time point n as $e^n = y_n^{Num} - y_n^{True}$. The numerical solution converges to the true solution as the local error approaches zero. The global error is usually defined as the difference between true and numerical solutions after a certain time. In other words, the global error e is the accumulation of the local error up to time *t* and can be written as

$$e = \sum_{n=1}^{N} e^n = \sum_{n=1}^{N} y_n^{Num} - y_n^{True}. \tag{5.10}$$

If the numerical method used to obtain the approximate solution has a global order of accuracy *p*, then it is expected that the global error will equal the truncation error as $e = o\left(h^p\right)$. Suppose that the local error is $e^n = o\left(h^q\right)$, where *q* is the order of the error estimate for all time points *n*, then for time $T = Nh$ one can write

$$e = o\left(h^p\right) = \sum_{n=1}^{N} e^n = \sum_{n=1}^{N} o\left(h^q\right) = No\left(h^q\right) = \frac{T}{h} o\left(h^q\right) = o\left(h^{q-1}\right). \tag{5.11}$$

Eq. (5.11) gives the relationship between p and q as $q = p+1$. This means that if the global order of accuracy is p, then the local error behaves as $e^n = o\left(h^{p+1}\right)$. In solving ODE, the local error introduced from discretizing the time component is accumulated into the global error as marching with time. It is hard to control the global error of the problem, but it is much easier to control the local error at each time step considering the appropriate time-step size. The local error is proportional to the time-step size as $e^{n+1} \sim h^{p+1}$, and by expanding the local error in an asymptotic series [104], then one can write

$$e^{n+1} = \Phi^n h^{p+1} + o\left(h^{p+2}\right), \tag{5.12}$$

where Φ^n is the principal error function at state point n and it depends on the solution. Similarly, an estimate for the local error $\hat{e}^{n+1}$, which is the difference between the approximated solution and the reference discrete solution $\hat{y}^{n+1}$ as $\hat{e}^{n+1} = y^{n+1} - \hat{y}^{n+1}$, can be derived as

$$\hat{e}^{n+1} = \hat{\Phi}^n h^{\hat{p}+1} + o\left(h^{\hat{p}+2}\right), \tag{5.13}$$

where the order $\hat{p} < p$ is depending on the design objectives of the method. The accuracy of the approximated solution can be maintained by forcing the local error estimate to be always below a prespecified value or tolerance level as $\hat{r}^{n+1} = \left\|\hat{e}^{n+1}\right\| < tol$. This can be achieved by selecting a time-step size that fulfills the above requirement. The elementary local error control algorithm predicts the time-step size can be derived as presented in Ref. [104, 105, 106] as

$$h_{n+1} = \left(\frac{\varepsilon}{\hat{r}^{n+1}}\right)^{1/k} h_n, \tag{5.14}$$

where ε is the user-specified tolerance, and $k = p+1$ is the order of the integration method. The derivation of this control algorithm is discussed in Ref. [91]. This scheme ensures that the estimated local error is close to the user-specified tolerance. The derivation of elementary controller assumes the local error varies asymptotically with the time-step size as $\hat{r}^{n+1} = \left\|\hat{\Phi}^n\right\| h^k$, and the norm of the principal error function varies slowly as $\left\|\hat{\Phi}^n\right\| \approx \left\|\hat{\Phi}^{n-1}\right\|$. In order to derive the discrete-time integral controller (I) with control theory, the elementary local error controller in Eq. (5.14) is rewritten as a linear difference equation as

$$\log\left(h_{n+1}\right) = \log\left(h_n\right) + \frac{1}{k}\left(\log\left(\varepsilon\right) - \log\left(\hat{r}^{n+1}\right)\right), \tag{5.15}$$

where the term $\log(\varepsilon)-\log(\hat{r}^{n+1})$ is called the control error, the factor $k_I = 1/k$ is the integral gain. The controller acts to make the local error estimate $\log(\hat{r}^{n+1})$ equals the setpoint $\log(\varepsilon)$, so that the control error vanishes, and only the control $\log(h_n)$ will change. Solving Eq. (5.15) by taking the sum of all past control errors as

$$\log(h_n) = \log(h_0) + k_I \sum_{m=1}^{n} \left(\log(\varepsilon) - \log(\hat{r}^m) \right). \tag{5.16}$$

In this integral control, the step size is obtained as a weighted sum of all past control errors. Under the assumption that $\hat{r}^{n+1} = \left\| \hat{\Phi}^n \right\| h^k$ and the local error varies asymptotically with time step, then

$$\log(\hat{r}^{n+1}) = \log\left(\left\| \hat{\Phi}^n \right\| \right) + k \log(h_n). \tag{5.17}$$

Inserting Eq. (5.17) into Eq. (5.15) then

$$\log(h_{n+1}) = (1 - kk_I)\log(h_n) + k_I \left(\log(\varepsilon) - \log\left(\left\| \hat{\Phi}^n \right\| \right) \right). \tag{5.18}$$

Thus, the root of the characteristic equation in Eq. (5.18) is $z = 1 - kk_I$. The integral gain k_I is not determined by the asymptotic model of the process. It is a free design parameter used to achieve a good overall closed-loop dynamic behavior for process and controller together. In Ref. [90, 91], it is mentioned that the choice of kk_I is a tradeoff between response time and sensitivity; for an *I* controller to work well one must choose $kk_I \in (0,1)$. The controller behavior based on the choice of kk_I can be summarized as

$$\begin{array}{cc} kk_I \in [0,2] & \text{Stable} \\ kk_I \in (0,1) & \text{Slow, smooth control} \\ kk_I = 1 & \text{Deadbeat control} \\ kk_I \in (1,2) & \text{Fast, oscillatory control} \end{array}. \tag{5.19}$$

Therefore, a low gain controller $kk_I \in (0,1)$ yields smoother step size sequences than the deadbeat control $kk_I = 1$. The general I controller [90, 91], can be written as

$$h_{n+1} = \left(\frac{\varepsilon}{\hat{r}^{n+1}} \right)^{k_I} h_n. \tag{5.20}$$

Comparing the elementary controller in Eq. (5.14) to the I controller in Eq. (5.20), the only change is the integral gain k_I without violating the assumption of the asymptotic behavior of the

error model in order to achieve smoother step size sequences. By dividing the local error estimate by the prespecified tolerance level, Eq. (5.20) can be rewritten as

$$h_{n+1} = \left(\hat{R}^{n+1}\right)^{-k_I} h_n, \qquad \hat{R}^{n+1} = \frac{\hat{r}^{n+1}}{\varepsilon}. \tag{5.21}$$

The calculated time-step size with the I controller depends on the initial time-step size, the integral gain, and the tolerance level. It is very important to select these parameters carefully based on the expected outcome. Later in this Chapter, further analysis, and recommendation on the optimum values of these parameters will be provided depending on the transient that is being analyzed.

5.2.2. Higher Order Controllers

In control theory, it is common to use more advanced control structures to increase robustness and to improve control performance, namely the proportional-integral (PI), and proportional-integral-derivative (PID). The PI controller adds an extra term proportional to the control error of the previous integral term called the proportional gain k_P. A relation for the PI controller can be derived in a similar way to the I controller as

$$h_{n+1} = \left(\hat{R}^{n+1}\right)^{-(k_I+k_P)} \left(\hat{R}^{n}\right)^{k_P} h_n. \tag{5.22}$$

The proportional gain will have an important role in increasing or decreasing the time-step size based on the error behavior. The proportional gain accounts for error trends, if $\hat{R}^n$ is increasing, then, the ratio of the control error is less than 1, and this will result in a quicker step size reduction than the I controller. Also, a decreasing error leads to a faster step size increase. The integral and proportional gains need to be determined based on the required behavior.

The PID controller adds an extra term that accounts for the derivative action of the previous errors rates of change to achieve faster response and allowing larger integral and proportional gains. The PID controller can be written like PI controller with the extra term accounts for the error rate of change as

$$h_{n+1} = \left(\hat{R}^{n+1}\right)^{-(k_I+k_P+k_D)} \left(\hat{R}^{n}\right)^{k_P+2k_D} \left(\hat{R}^{n-1}\right)^{-k_D} h_n, \tag{5.23}$$

where k_D is the derivative gain. The PID controller combines the I and PI controllers and introduces an extra term to achieve a faster response to the system changes. In the following analysis, the PID controller is selected for the adaptive time-stepping algorithm.

5.3. Estimation of Local Error

In order to predict the new time-step size using I, PI, or PID controllers as in Eq. (5.21), Eq. (5.22), or Eq. (5.23), the local error must be calculated. The local error estimate normalized by the tolerance level can be generalized from Eq. (5.13) as

$$\hat{R}^{n+1} = \frac{1}{\varepsilon}\left\|\hat{\Phi}^n h^{\hat{p}+1}\right\|_q . \tag{5.24}$$

The tolerance in Eq. (5.20) is a user-specified value, and it is a function of the parameters of the problem as the energy and space. To provide more freedom for the user, Eq. (5.24) can be normalized to the solution so that the tolerance value can be specified as a dimensionless parameter. Then, Eq. (5.24) can be written in a more generalized way as

$$\hat{R}^{n+1} = \frac{\left\|\hat{\Phi}^n h_n^{\hat{p}+1}\right\|_q}{\varepsilon\left\|y^n\right\|_q} . \tag{5.25}$$

In order to calculate the local error estimate, the principal error function needs to be estimated. In solving the time-dependent neutron diffusion equation, the neutron flux is discretized in time, and it is the most sensitive parameter to changes compared to the delayed neutron precursor concentration, the fuel salt temperature, and velocity. So, it should be used to estimate the principal error function. The local error is proportional to the time-step size and the principal error function as shown in Eq. (5.13). Also, Eq. (5.4) shows that the local error is proportional to the time-step size and the second derivative of the neutron flux. By comparing Eq. (5.4) and Eq. (5.13) and considering the second order of accuracy ($\hat{p}=1$), then the following relation can be derived as

$$\hat{e}^{n+1} = h_n \tau_n = \hat{\Phi}^n h_n^2 + o\left(h^3\right) = \frac{h_n^2}{2}\left.\frac{\partial^2 \phi}{\partial t^2}\right|_{t_n} + o\left(h_n^3\right). \tag{5.26}$$

Then, a relation of the principal error function in terms of the second derivative can be written as

$$\hat{\Phi}^n \simeq \frac{1}{2}\left.\frac{\partial^2 \phi}{\partial t^2}\right|_{t_n} . \tag{5.27}$$

The second derivative in Eq. (5.27) can be calculated using the finite difference approach as

$$\left.\frac{\partial^2 \phi}{\partial t^2}\right|_{t_n} = \frac{h_{n-1}\phi(t_n) - \left(h_{n-1}+h_n\right)\phi(t_{n-1}) + h_n \phi(t_{n-2})}{h_n h_n h_{n-1}} . \tag{5.28}$$

Then, the local error estimate can be calculated as

$$\hat{R}_{2nd}^{n+1} = \frac{1}{\varepsilon\left\|\phi(t_n)\right\|_2}\left\|\frac{1}{2}\phi(t_n) - \frac{1}{2}\left(1+\frac{h_n}{h_{n-1}}\right)\phi(t_{n-1}) + \frac{h_n}{2h_{n-1}}\phi(t_{n-2})\right\|_2 . \tag{5.29}$$

An alternative way to calculate the second derivative is to assume the flux is varying exponentially in the time interval $[t_{n-1}, t_n]$ as $\phi(t) = \phi\left(t_{n-1}\right) e^{\omega_n (t - t_{n-1})}$, then the second derivative can be obtained analytically as

$$\left.\frac{\partial^2 \phi}{\partial t^2}\right|_{t_n} = \omega_n^2 \phi(t_n), \tag{5.30}$$

where the inverse period ω_n is approximately determined using the flux solutions at t_{n-2} and t_{n-1} as

$$\omega_n = \frac{1}{h_{n-1}} \ln\left(\frac{\phi(t_{n-1})}{\phi(t_{n-2})}\right). \tag{5.31}$$

Eq. (5.30) will always lead to a value greater or equal to zero. To avoid this situation, the second derivative is evaluated with the implicit Euler scheme as in Eq. (5.3), then calculating the first derivative analytically as

$$\left.\frac{\partial^2 \phi}{\partial t^2}\right|_{t_n} = \frac{1}{h_n}\left(\left.\frac{\partial \phi}{\partial t}\right|_{t_n} - \left.\frac{\partial \phi}{\partial t}\right|_{t_{n-1}}\right) = \frac{1}{h_n}\left(\omega_n \phi(t_n) - \omega_{n-1}\phi(t_{n-1})\right). \tag{5.32}$$

Then, the local error estimate can be calculated as

$$\hat{R}_{\omega}^{n+1} = \frac{\left\| \left(\omega_n \phi(t_n) - \omega_{n-1}\phi(t_{n-1})\right) h_n / 2 \right\|_2}{\varepsilon \left\| \phi(t_n) \right\|_2}. \tag{5.33}$$

If the principal error function is slowly varying in time as assumed in Eq. (5.14), it can be approximated from the first derivative of the neutron flux in Eq. (5.3) with the first-order accuracy ($\hat{p} = 0$), and the local error estimate can be calculated as

$$\hat{R}_{1st}^{n+1} = \frac{\left\| \phi(t_n) - \phi(t_{n-1}) \right\|_2}{\varepsilon \left\| \phi(t_n) \right\|_2}. \tag{5.34}$$

At the beginning of the transient calculations, the initial time step must be specified, and after that, the I controller can be used to predict the step size of the next transient step. Once the size of the first step is determined, the procedure will continue until the end of the transient time using I, PI, or PID controllers.

5.4. Time-Step Control Algorithm

An adaptive time-step selection algorithm is implemented in the PROTEUS-NODAL code for transient analysis of the MSRs based on the control theory approach. The implemented scheme will help to avoid any instability issues that might be introduced due to a large time-step size and

to avoid any unnecessary calculations for a small time-step size, especially for coupled systems. Also, for multiphysics simulations with as loosely coupled physics, the coupled systems are nonlinear in time, and step size changes may be necessary to obtain an accurate coupled solution. This will help the user by eliminating the burden of prespecifying the time-step size without knowing many details of the nature of the transient.

The implemented time-step selection algorithm can be summarized as in the following. The first step is to calculate the neutron flux and the delayed neutron precursor concentration at time step *n* using the updated cross sections and velocity field of the previous time step as in the time-lagged approach or using the previous Picard iteration values. Then, the fuel salt temperature, density, and velocity fields are calculated with the thermal hydraulics solver using the latest power distribution. After that, the cross sections of the next time step or the next Picard iteration are updated using the fuel salt temperature and density distributions. Once the solution is converged, the local error estimates are calculated, and the next time-step size is calculated using I, PI, or PID controllers utilizing the values of the previous time-step size, prespecified tolerance, and controller gains values. In the final step, the newly calculated time-step size should be maintained within certain limits specified by the user or due to limitations on the coupled codes system. To avoid solution instabilities, step size filters are applied, and the new time step will be checked if it satisfies these limits. Otherwise, the new time-step size will be adjusted accordingly and proceed to the next time step. Algorithm 5.1 provides the steps to select the time-step size with the PID controller.

If the transient that is being analyzed has a massive change in the power at a certain time point due to quick insertion of reactivity due to a step movement of control rods or flow back of heated fuel to the core with higher temperature, Algorithm 5.1 might miscalculate the next time-step size because it is estimated from the previous time step solutions, which didn't embrace the new change yet. A more effective algorithm for time-step size selection can be used that allows rejecting the predicted time-step size if the local error estimate is larger than the specified tolerance level. In this way, the predicted time-step size will be adjusted to fulfill the prespecified tolerance level or minimum time-step size. Algorithm 5.2 provides the steps to select the time-step size with the PID controller while allowing the rejection of time-step size to monitor the level of the local error. Although this algorithm is computationally more expensive than Algorithm 5.1, it ensures the accuracy level of the solution. The steps in Algorithm 5.2 are quite similar to the steps in Algorithm

5.1 except for step 5, which will check the local error estimate, adjust the time-step size, and repeat the calculations if necessary.

Algorithm 5.1. Time-step size selection algorithm with PID controller.

1. Calculate $(\phi, C, P)^n$ with the previous time step $(\Sigma, u)^{n-1}$
2. Calculate $(T, \rho, u)^n$ with the current time step power distribution $(P)^n$
3. Update cross sections $(\Sigma)^n$ using $(T, \rho)^n$
4. Calculate $\hat{R}^{n+1}$
5. Calculate the next time step

$$h_{n+1} = \left(\hat{R}^{n+1}\right)^{-(k_I+k_P+k_D)} \left(\hat{R}^{n}\right)^{k_P+2k_D} \left(\hat{R}^{n-1}\right)^{-k_D} h_n$$

$$\Delta t_l = 0.95 \min\left(\Delta z / u_n\right)$$

$$h_{n+1} = \max\left(\min\left(h_{n+1}, \Delta t_l, \Delta t_{\max}\right), \Delta t_{\min}\right)$$

$$(\phi, C, T, \rho, u)^{n-1} = (\phi, C, T, \rho, u)^n$$

$$\hat{R}_{n-1} = \hat{R}_n, \quad \hat{R}_n = \hat{R}_{n+1}$$

$$h_n = h_{n+1}, \quad t = t + h_{n+1}, \quad n = n+1$$

6. If $(t = t_{end})$ terminate calculations Else Go back to 1

To explain the difference between the two adaptive time-stepping (ATS) algorithms, a reactivity insertion (RI) transient without thermal feedback was simulated for 0.2 s. In this transient, two step reactivities were inserted, one at time 0.0 s with +100 pcm, and the second at time 0.1 s with -100 pcm reactivity insertion. The reference solution was obtained with a pre-defined or scheduled constant time-step size of 0.1 ms. Figure 5.2 compares the power change of the ATS solution algorithms and the reference solution. Figure 5.3 shows the time-step size change and local error variation during the transient. After 0.01 s, the power starts increasing monotonically due to the positive reactivity insertion. Both solution algorithms provided the same solution as the reference, and the predicted time-step size is the same. At time 0.1 s, negative reactivity was inserted, and a difference in the ATS solutions is observed.

Algorithm 5.1 predicts the time-step size based on the behavior of the previous solution and it does not allow monitoring of the local error. As a result, the time-step size is miss predicted after insertion of the negative reactivity, and it leads to miscalculating the power as well. Algorithm 5.2 allows monitoring of the local error by rejecting the predicted time-step size if it leads to an increase in the local error beyond the prespecified limit, and the time-step size will be reduced and

repeat the solution until the criterion is met. It is clearly seen that the solution of Algorithm 5.2 matches the reference solution all the time. The time-step size tends to increase as the rate of change in the power level is low, and it decreases as the rate of change in the power change is high. Algorithm 5.2 ensures the accuracy of the solution all the time, despite that it requires more calculations which will increase the computational time and reduce the speedup compared to Algorithm 5.1. In the remaining analysis in this chapter, Algorithm 5.2 will be used to perform the calculations, and its performance will be investigated for various transits of MSRs.

Algorithm 5.2. Time-step size selection algorithm with PID controller and error monitoring.

1. Calculate $(\phi, C, P)^n$ with the previous time step $(\Sigma, u)^{n-1}$
2. Calculate $(T, \rho, u)^n$ with the current time step power distribution $(P)^n$
3. Update cross sections $(\Sigma)^n$ using $(T, \rho)^n$
4. Calculate $\hat{R}^{n+1}$
5. If $\left(\left(\hat{R}^{n+1} > 1.0\right)\ and\ \left(h_n > \Delta t_{\min}\right)\right)$ then (Reject time step)

   $t = t - h_n,\quad n = n-1,\quad Nrej = Nrej + 1$

   $h_n = \max\left(\min\left(0.5h_n, \left[\varepsilon / \hat{r}_{n+1}\right]h_n\right), \Delta t_{\min}\right)$

   $(\phi, C, T, \rho, u)^n = (\phi, C, T, \rho, u)^{n-1}$
6. Else (Calculate the next time step)

   $h_{n+1} = \left(\hat{R}^{n+1}\right)^{-(k_I + k_P + k_D)} \left(\hat{R}^n\right)^{k_P + 2k_D} \left(\hat{R}^{n-1}\right)^{-k_D} h_n$

   $\Delta t_l = 0.95 \min\left(\Delta z / u_n\right)$

   $h_{n+1} = \max\left(\min\left(h_{n+1}, \Delta t_l, \Delta t_{\max}\right), \Delta t_{\min}\right)$

   $(\phi, C, T, \rho, u)^{n-1} = (\phi, C, T, \rho, u)^n$

   $\hat{R}_{n-1} = \hat{R}_n,\quad \hat{R}_n = \hat{R}_{n+1}$

   $h_n = h_{n+1},\quad t = t + h_{n+1},\quad n = n+1$

   End If
7. If $\left(t = t_{end}\right)$ terminate calculations Else Go back to 1

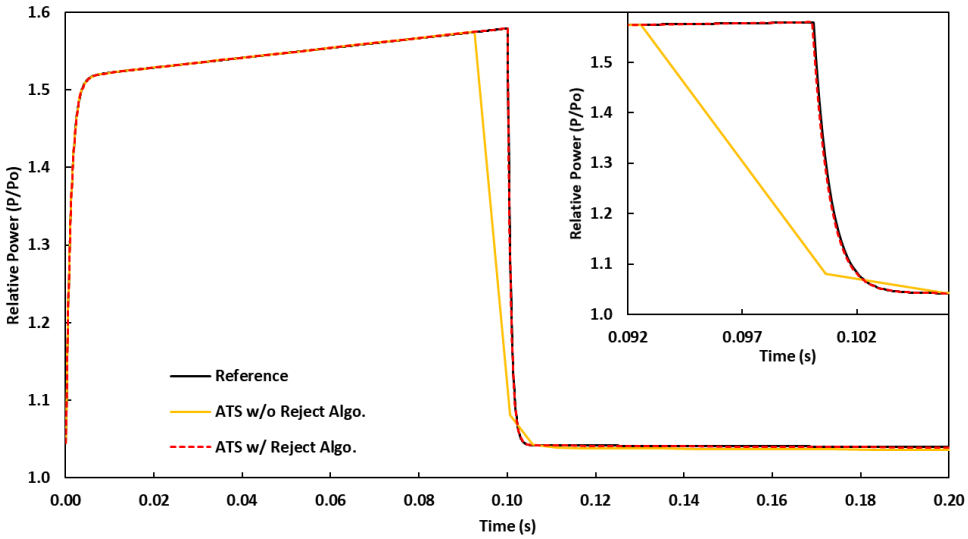


Figure 5.2. Comparison of ATS solution algorithm after step reactivity insertion transient without thermal feedback.

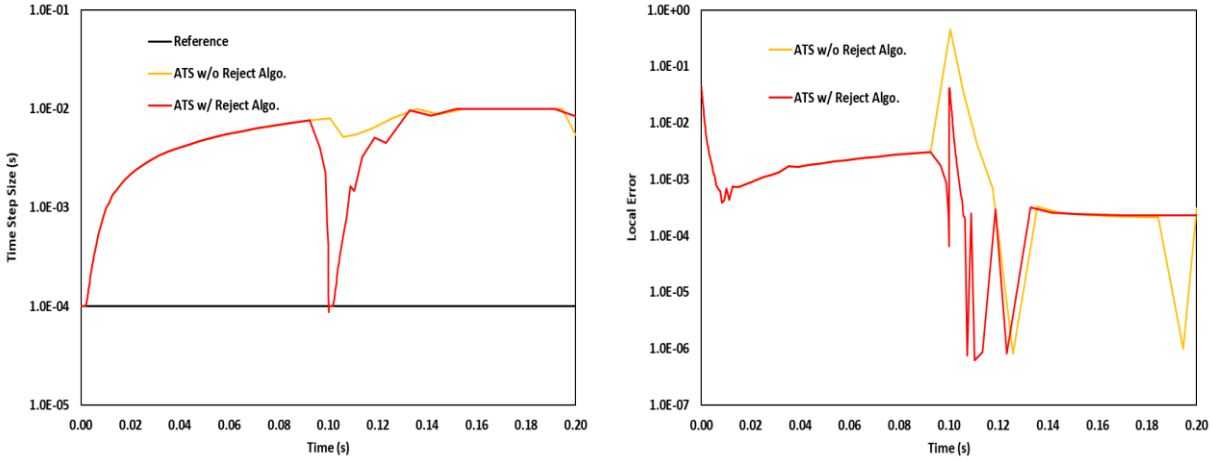


Figure 5.3. Variation of time-step size (left) and local error (right) with ATS solution algorithm after step reactivity insertion without thermal feedback.

5.4.1. Step Size Filters

For coupled transient calculations of the MSR, digital filters were employed on the time-step size prediction to avoid any instabilities in the solution of the coupled solvers. The first filter is controlled by the user, and it allows to contain the time-step size within upper and lower limits to avoid undesired very large or very small time-step sizes as

$$\Delta t_{\min} \leq h^{n+1} \leq \Delta t_{\max} . \tag{5.35}$$

Another filter is applied to avoid a significant change in the time-step size of two successive time points in order to avoid a vast change in the time-step size at a single time point as

$$0.5h_n \leq h_{n+1} \leq 2h_n . \tag{5.36}$$

The last filter is applied for coupled calculations to ensure the stability of the solution by limiting the time-step size according to the change of fuel salt velocity within the axial distance of the node to satisfy the CFL (Courant Friedrichs Lewy) condition as

$$h_{n+1} < \frac{\Delta z}{u^n}, \qquad h_{n+1} = 0.95\frac{\Delta z}{u^n}. \tag{5.37}$$

An increase of the time-step size beyond this limit will introduce some discrepancies in the thermal hydraulics solver. The 0.95 is a safety factor, and it is introduced to make sure that the time-step size is less than the limit. If any of these conditions are met, the predicted time step will be adjusted to the minimum allowed limit on the time-step size.

5.4.2. Coupling Scheme

The transient solvers of PROTEUS-NODAL for neutronics and thermal hydraulics consist nonlinear system, and they were coupled by employing the operator splitting approach and utilizing the implicit Euler (backward differencing) scheme. Two coupling schemes were implemented in PROTEUS-NODAL, the time-lagged (TL) scheme and the Picard iteration (PI) scheme. In the time-lagged approach, the neutronics and thermal hydraulics calculations are performed once at each time step and marching forward until the end of the simulation time. An illustration of the time-lagged coupling scheme is shown in Fig. 5.4. First, the neutronics calculations are performed with updated the cross sections and the velocity field from the previous time-step to determine the power distribution in the core. Then, the thermal hydraulics calculations are performed to obtain the temperature, density, and velocity fields. Finally, the multigroup cross sections are updated using the temperature and density fields and move to the next time step.

The Picard iteration coupling scheme was added to obtain a more accurate solution of the coupled physics, and it might allow a larger time step because of the reduced local error. In this scheme, the neutronics and thermal hydraulics are performed iteratively within a time step until the residual of the solution meets the convergence criterion. An illustration of the Picard iteration coupling scheme is shown in Fig. 5.5. At each time step, the neutronics and thermal hydraulics are performed at least twice, so that the latest updated cross sections are used for the power calculation, which is used in thermal hydraulics parameters calculations.

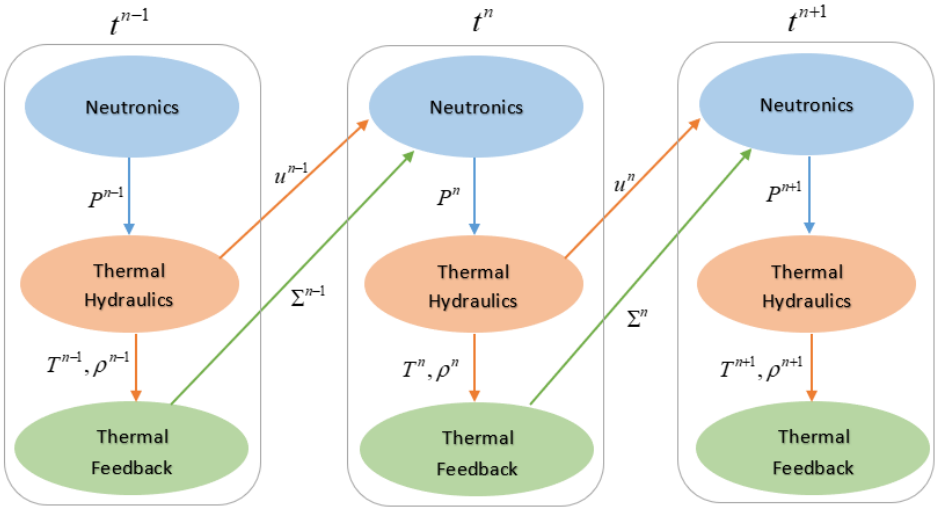


Figure 5.4. Time lagged coupling scheme implemented in PROTUES-NODAL.

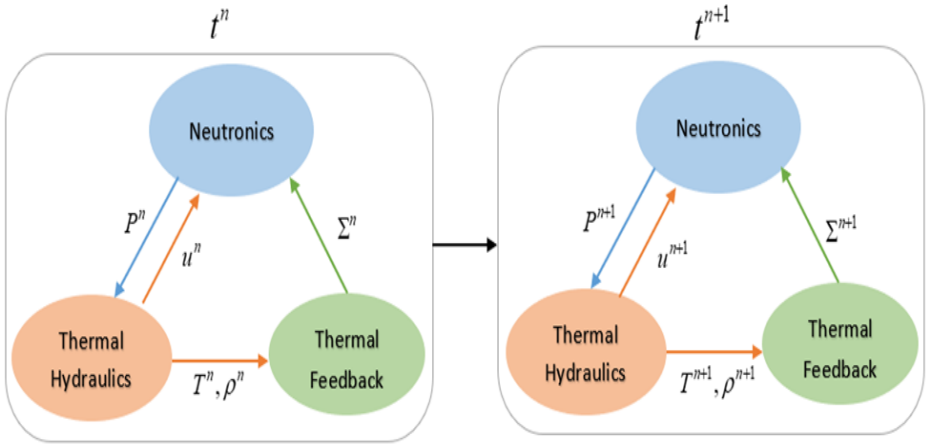


Figure 5.5. Picard iteration coupling scheme implemented in PROTUES-NODAL.

The Picard iteration coupling scheme will ensure the accuracy of the solution of the coupled system if a large time-step size is used. However, the time-lagged approach provides a very quick solution, and it gives accurate results for most of the transients that were analyzed. In order to demonstrate this, two UTOP transients of the MSFR benchmark were simulated with 50 pcm and 200 pcm reactivity insertions with both coupling schemes and using a constant time-step size of 0.05 ms. Figure 5.6 compares the solution of the time-lagged scheme to the Picard iteration scheme for the 50 pcm reactivity insertion. For this case, the two solutions are matching very well with a difference of 0.05% for the whole transient duration.

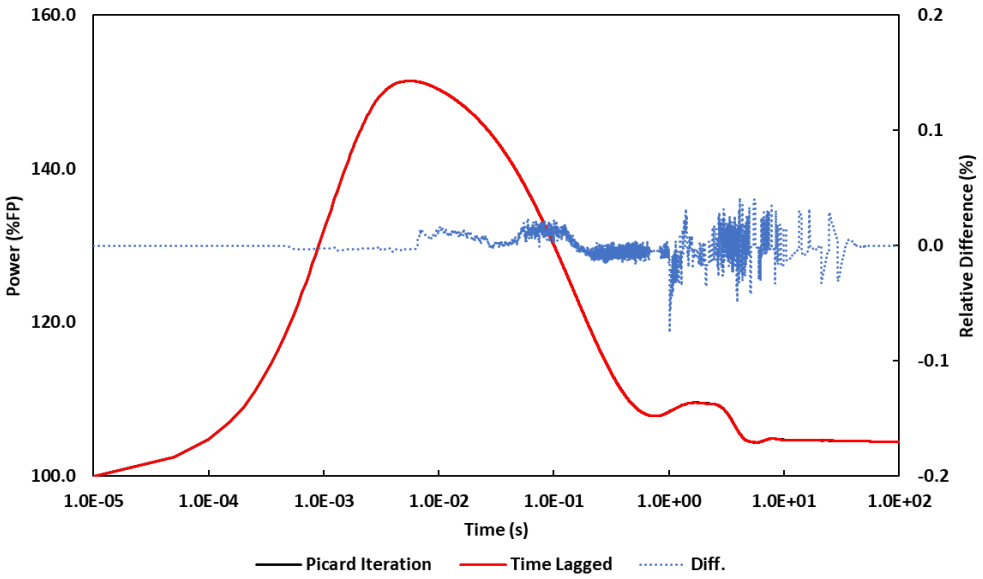


Figure 5.6. Comparison of coupling schemes power evolution after 50 pcm reactivity insertion.

On the contrary, for the 200 pcm reactivity insertion, a larger error can be seen at the peak region with about 2.0% as shown in Fig. 5.7. In this case, the time-lagged approach overestimates the peak power because of the lagged thermal feedback. However, the difference looks small, but the magnitude of the power is very large at the peak region. In the following analysis, the time-lagged approach will be used since it provides accurate results if a small time-step size is used, and it reduces the computational time by at least a factor of 2 for the same time-step size.

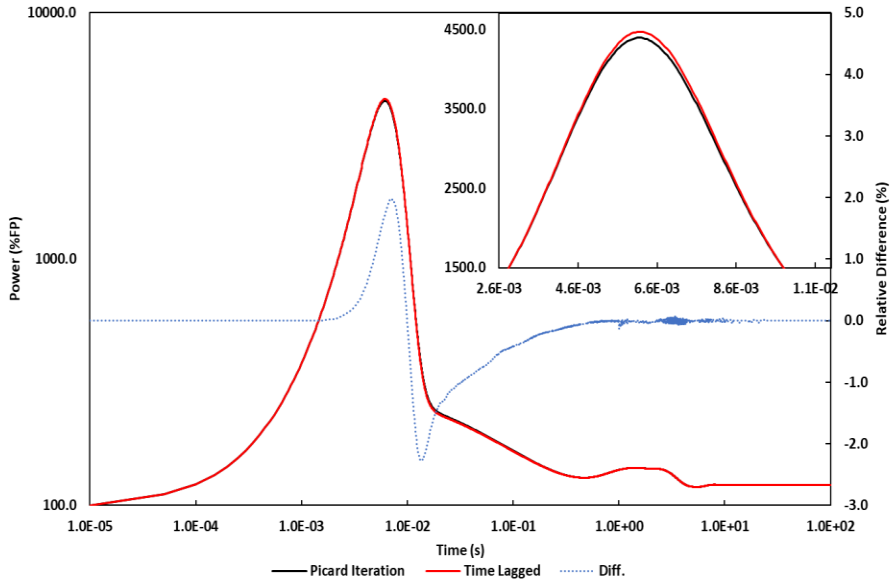


Figure 5.7. Comparison of coupling schemes power evolution after 200 pcm reactivity insertion.

5.4.3. Solution Assessment

In order to assess the ATS solution of the simulated transients, the power was compared to the reference solution considering the maximum relative difference, the average relative difference or the root mean square value, and the integral of the difference. The numerical reference solution was obtained using a very small time-step size. For reactivity insertion transients like UTOP transient, the peak power relative difference and the time to the peak are more important quantities to assess the solution, so they were used in the comparison. These quantities are calculated as

$$\varepsilon_{peak} = \left| \frac{P^m - P^m_{ref}}{P^m_{ref}} \right|, \tag{5.38}$$

$$\varepsilon_{\max} = MAX_t \left| \frac{P(t) - P_{ref}(t)}{P_{ref}(t)} \right|, \tag{5.39}$$

$$\varepsilon_{rms} = \sqrt{\sum_t \left(\frac{P(t) - P_{ref}(t)}{P_{ref}(t)} \right)^2}, \tag{5.40}$$

$$\varepsilon_{\text{int}} = \frac{1}{T} \int_0^T \left| \frac{P(t) - P_{ref}(t)}{P_{ref}(t)} \right| dt . \tag{5.41}$$

5.5. Numerical Results

The adaptive time-step size selection algorithm implemented in the PROTEUS-NODAL code has been tested and verified considering a wide range of transient scenarios with thermal feedback. The transient scenarios are classified based on the rate at which the solution varies with time, and it includes rapid or prompt, moderate, and slow varying transients. The rapid or prompt transients focus on the reactivity insertion with and without thermal feedback, as in the UTOP transient. The moderate varying transients mainly focus on the transient scenarios with thermal feedback where the power varies at a mild rate, as in the pump-driven and temperature-driven transients that were analyzed for the MSFR benchmark, including UPOS, ULOF, UFSOC, and ULOHS. The last class is for very slow and long transient, as in the natural circulation test of the MSRE experiment, where varying the time-step size is very necessary to reduce the computational cost.

Also, another category was analyzed considering localized transient scenarios as in the control rod withdrawal or fuel salt channel blockage in thermal spectrum MSRs such as the MSRE experiment. In this section, the verification test results are presented and compared to the reference solution obtained from PROTEUS-NODAL calculations with a very small time-step size. Also,

the savings in the computational cost and the speedup of calculations will be presented and recommended values for the controller initial time-step size, tolerance level, and integral, proportional, and derivative gains will be provided.

5.5.1. Rapid Transients

To test the robustness of the ATS solution algorithm implemented in PROTEUS-NODAL, the UTOP transients of the MSFR benchmark reactor were simulated. In Chapter 3, the UTOP transients with two step reactivity insertions were simulated for a sub-prompt critical with 50 pcm step reactivity insertion and super-prompt critical with 200 pcm reactivity insertion. In the following analysis, a parametric study was performed to test the ATS algorithm with different coupling schemes, the value of the integral, proportional, and derivative gains, and the level of the solution tolerance. Since these parameters are selected by the user, a recommended range for these parameters will be suggested to avoid any undesired instabilities in the solution. To assess the ATS algorithm solution, the reference solution of the simulated transients must be available. To provide an accurate assessment, the reference solution of the UTOP transients was obtained with a very small and constant time-step size. Figures 5.8 and 5.9 show the MSFR UTOP transient with positive 50 pcm and 200 pcm reactivity insertions with different constant time-step sizes. Also, Table 5.1 provides a comparison of the peak power value, time to peak power, and the solution error relative to the solution with the smallest time-step size considered as 0.05 ms.

Table 5.1. Comparison of Time-step Size and Peak Power of MSFR UTOP Transients.

Case	**Δt (ms)**	Peak Power (%FP)	Time to Peak (ms)	Error (%)		
				Peak	RMS	Integral
UTOP-50 pcm	0.05	1.514	5.55	-	-	-
	0.10	1.514	5.50	0.02	1.76	0.39
	0.50	1.514	6.00	0.00	5.31	0.44
	1.00	1.511	7.00	0.21	6.96	0.45
	5.00	1.504	10.00	0.65	6.12	0.50
UTOP-200 pcm	0.05	44.681	6.15	-	-	-
	0.10	45.019	6.10	0.76	23.43	0.38
	0.50	48.261	5.50	8.01	109.04	1.21
	1.00	54.368	5.00	21.68	210.35	2.58

For the 50 pcm reactivity insertion transient, as the time-step size increases, the peak power is slightly underestimated. As a result, the time to peak is delayed because of the delay in the

feedback response, also the relative error is increasing. However, the solution is almost the same for all time-step sizes for the remaining of the transient after the peak region (after 1.0 s). For the 200 pcm reactivity insertion transient an opposite behavior was observed as the time-step size increases, the peak power is overestimated because of the discretization error, and the time to peak is brought forward due to the advance on the thermal feedback response. Also, the relative error is increasing. However, the solution is almost the same for all time-step sizes for the remaining of the transient after the peak region (after 1.0 s).

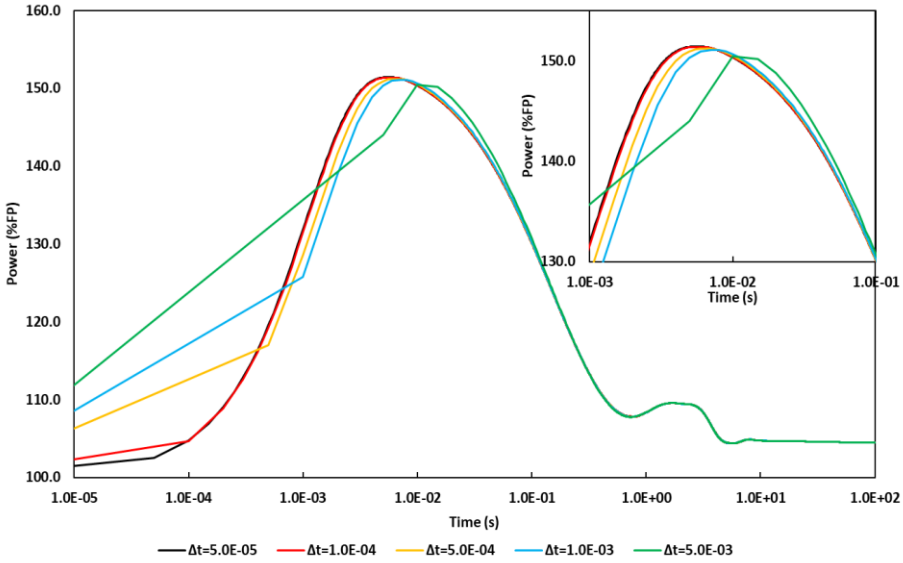


Figure 5.8. Power evolution after 50 pcm reactivity insertion with different time-step size.

For both UTOP cases, a time-step size of 0.1 ms provides a reasonable solution compared to the 0.05 ms. Based on the above observations, it is recommended to use a time step in the range of 0.05 to 0.1 ms up to 1.0 s, and the step size can be increased after that up to 5.0 ms for a pre-defined or scheduled time-step size. In the remaining analysis, the UTOP transient reference solution was obtained with a time-step size of 0.05 ms up to 1.0 s of the transient. Then, it was increased to 10.0 ms for the remaining of the transient since the power is varying slowly after the peak. The use of the ATS algorithm will eliminate the need for such calculations to determine the optimum time-step size. The performance of the ATS with different local error estimates, gain value, and tolerance level is investigated for the UTOP transients of the MSFR, as discussed in the following subsections.

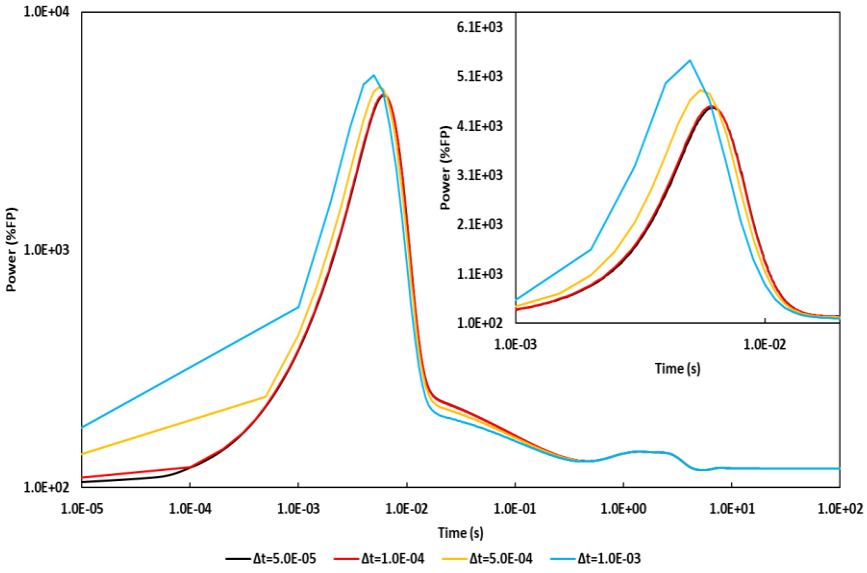


Figure 5.9. Power evolution after 200 pcm reactivity insertion with different time-step size.

5.5.1.1. Local Error Estimate

The performance of the ATS solution algorithm with different local error estimates is investigated using the UTOP transients of the MSFR benchmark. The solution of the ATS for all the local error estimates was performed with an initial time-step size of 0.05 ms, gain value of 0.05, and tolerance level of 1.0×10^{-3}. Table 5.2 summarizes the UTOP transient results of the MSFR for the peak power, relative errors, computational time, and speedup ratio for the three local error estimates. Figure 5.10 shows the power evolution for 50 pcm reactivity insertions, while Fig. 5.11 shows the time-step size change and the local error change during the transient, respectively. For both UTOP transients, smaller relative errors are observed with local error estimates obtained by the second derivative, while the first derivative shows larger relative errors. On the other hand, the computational time is reduced significantly for all local error estimates with more speedup was achieved with the second derivative for the local error estimates.

For the 50 pcm reactivity insertion, the speedup is about 16 times for the first derivative, and about 24 times for the second derivative local error estimates. In Fig. 5.10, a small deviation in the power can be seen after the peak for the second derivative approach due to the larger predicted time-step size. Both the finite difference and exponential approximations show consistent results. In Fig. 5.11, the predicted time-step size with the second derivative approach is larger than the first derivative approach. Also, the total number of the time steps and the rejected steps is much smaller for the same controller parameters. This can be explained by the smaller value of the estimated

local error with the second derivative approach. This will result in a larger time-step size since the predicted time step is inversely proportional to the local error, as shown in Fig. 5.11. The oscillatory behavior of the estimated local error is due to thermal feedback changes and the fuel velocity change during the transient. However, the size of the predicted time-step size can be controlled by adjusting the tolerance level and the gain value. By reducing the values of the tolerance and the gain, a smaller time-step size is guaranteed.

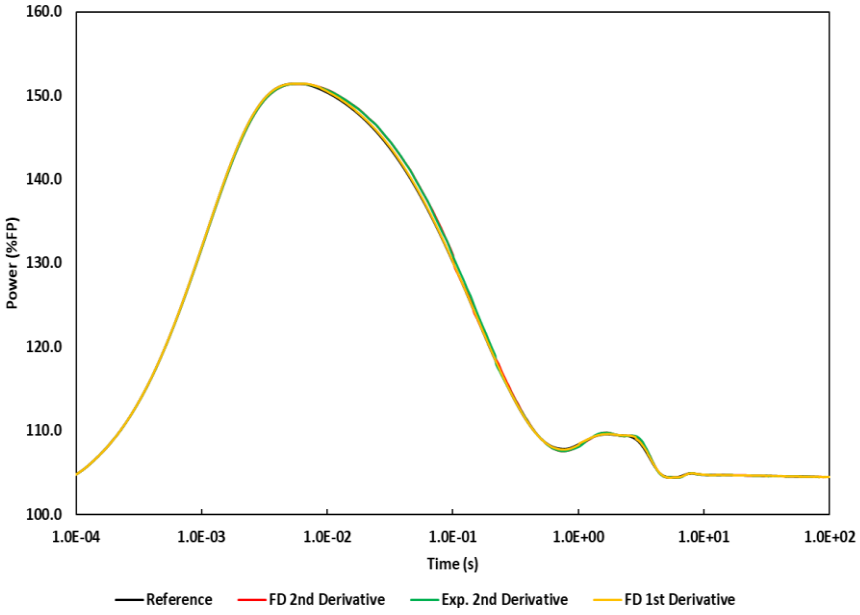


Figure 5.10. Power evolution in MSFR after 50 pcm RI with different error estimates.

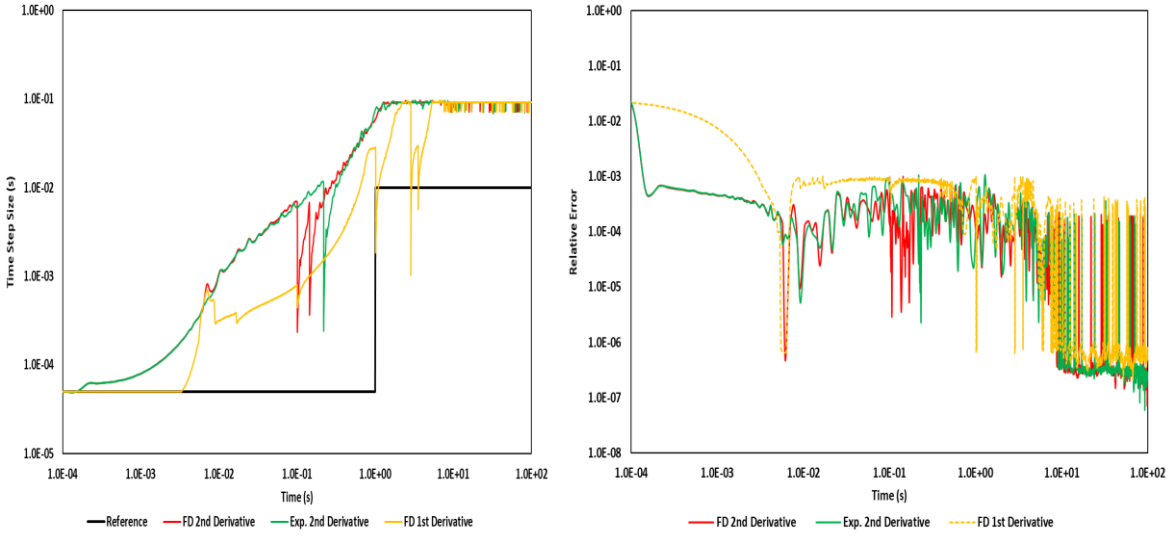


Figure 5.11. Step size change (left) and estimated local error (right) in MSFR after 50 pcm RI with different error estimates.

Figure 5.12 shows the power evolution for 200 pcm reactivity insertions, while Fig. 5.13 shows the time-step size change and the local error change during the transient, respectively. The speedup is about 8 times for the first derivative and about 16 times for the second derivative local error estimates, with a large reduction in the total number of time steps and the rejected steps due to the same reasoning of the smaller estimated local error by the second derivative approach. The peak power is slightly overestimated with an error of less than 0.8% from the second derivative approach. All three approaches have provided consistent results under the same conditions with a large reduction in the computational cost, and they can be used for predicting the time-step size accurately.

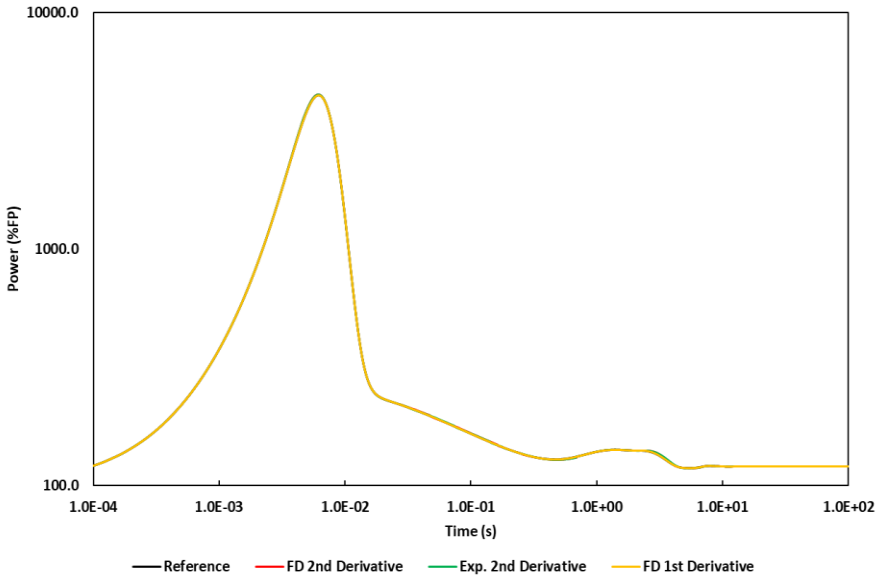


Figure 5.12. Power evolution in MSFR after 200 pcm RI with different error estimates.

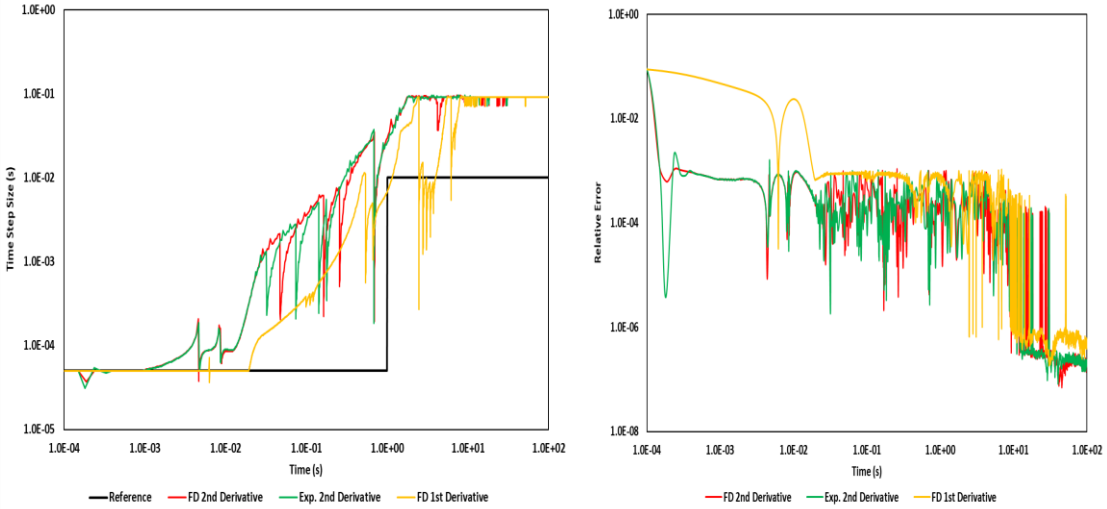


Figure 5.13. Step size change(left) and estimated local error (right) in MSFR after 200 pcm RI with different error estimates.

Table 5.2. ATS Performance with Different Local Error Estimates of MSFR UTOP Transients.

Case	Local Error Estimate	Peak Power (P/Po)	Time to Peak (ms)	Relative Error (%)			No. Steps		Comp. Time	Speedup
				Peak	RMS	Integral	Total	Rejected		
50-pcm	Reference	1.514	5.55	-	-	-	29900	-	1.000	-
	2nd FD	1.514	5.40	0.02	3.82	1.98	1252	14	0.042	24.09
	Exponential	1.514	5.73	0.01	4.17	2.11	1237	9	0.040	24.76
	1st FD	1.514	5.32	0.00	1.87	1.48	1726	38	0.060	16.65
200-pcm	Reference	44.682	6.15	-	-	-	29900	-	1.000	-
	2nd FD	45.015	6.11	0.75	9.52	3.31	1496	46	0.052	16.56
	Exponential	44.987	6.18	0.68	9.52	3.71	1504	44	0.053	16.18
	1st FD	44.682	6.15	0.00	2.64	1.18	2670	85	0.098	8.73

5.5.1.2. Gain Value

The gain value in the ATS controller plays an important role in increasing or decreasing the time-step size based on the ratio of the tolerance and the local error. Selecting the gain value influences how fast the time-step size will increase or decrease. Too small gain values will result in slow and steady control, while large values will result in fast and oscillatory control. It is important to provide a reasonable value of this parameter to reach an optimum control. The performance of the ATS solution algorithm considering different gain values is investigated using the UTOP transients of the MSFR benchmark. The solution of the ATS was performed with an initial time-step size of 0.05 ms, a tolerance level of 1.0×10^{-3}, and the local error was estimated with the first derivative. Also, both the Picard iteration and the time-lagged coupling schemes were tested. Table 5.3 summarizes the UTOP transient results of the MSFR with 50 pcm reactivity insertion for the peak power, the relative errors, the computational time, and the speedup ratio for a gain value ranging between 0.01 and 0.5.

Figure 5.14 shows the power evolution obtained with the time-lagged coupling scheme for 50 pcm reactivity insertion, while Fig. 5.15 shows the power evolution obtained with the Picard iteration coupling scheme. For each coupling scheme, the ATS solution was compared with the reference solution obtained using the same coupling scheme. For both coupling approaches, the results are not so sensitive to the gain value, however, a large gain value, like 0.5, leads to an undesired solution at the peak region due to overestimated time-step size, as shown in Fig. 5.14

and Fig. 5.15. It is recommended to use a gain value in the range of 0.05 to 0.1 for such transient with both coupling schemes.

Table 5.3. ATS Performance with Different Gain Level of MSFR UTOP-50 pcm Transient.

Case	Gain	Peak Power (P/Po)	Time to Peak (ms)	Relative Error (%)			No. Steps		Comp. Time	Speedup
				Peak	RMS	Integral	Total	Rejected		
Time-Lagged	**Reference**	1.514	5.55	-	-	-	29900	-	1.000	-
	0.01	1.514	5.50	0.00	1.30	1.10	2100	7	0.077	13.02
	0.05	1.514	5.32	0.00	1.92	1.45	1735	17	0.065	15.42
	0.10	1.514	5.30	0.01	2.05	1.57	1747	39	0.067	15.00
	0.50	1.517	19.56	0.17	6.56	2.01	1237	3	0.051	19.65
Picard Iteration	**Reference**	1.514	5.50	-	-	-	29900	-	1.000	-
	0.01	1.514	5.49	0.00	0.99	1.30	2227	28	0.107	9.37
	0.05	1.514	5.54	0.00	1.45	1.47	2086	172	0.102	9.78
	0.10	1.514	5.58	0.01	1.36	1.28	2133	415	0.114	8.79
	0.50	1.514	7.82	0.01	10.28	2.53	1475	289	0.094	10.60

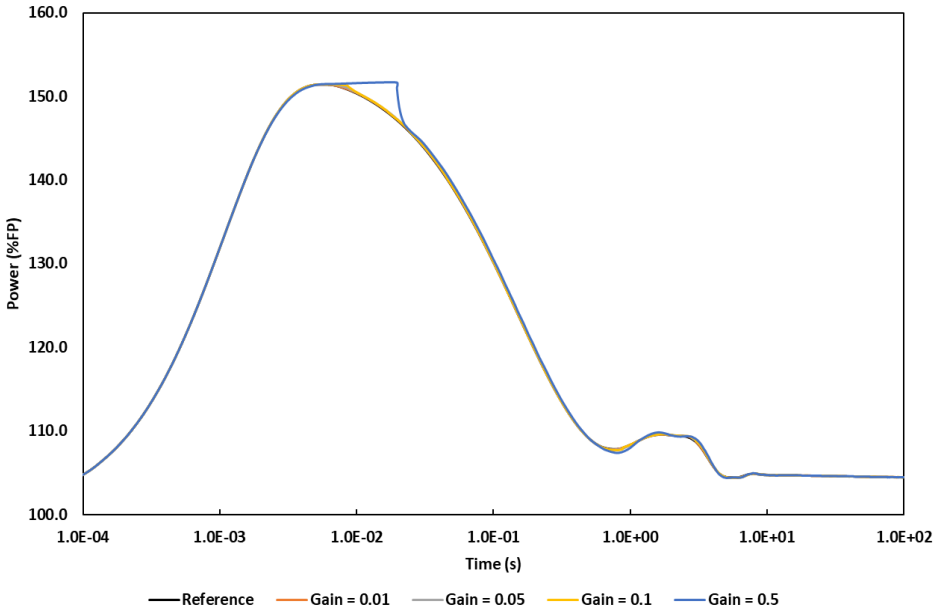


Figure 5.14. Power evolution after 50 pcm RI with TL scheme and different gain levels.

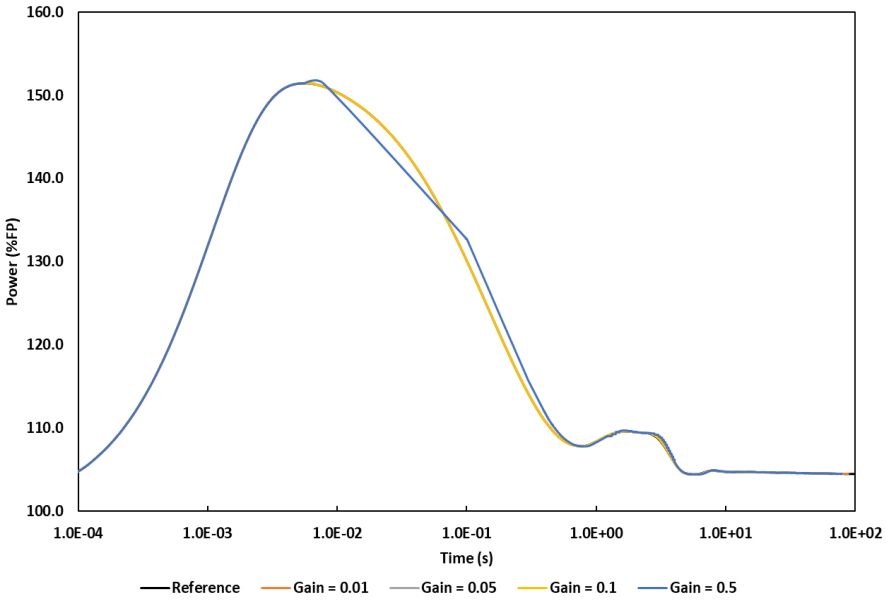


Figure 5.15. Power evolution after 50 pcm RI with PI scheme and different gain levels.

5.5.1.3. Tolerance Level

The tolerance is a user-specified value, so it should be provided in advance. This value depends on the transient being simulated, for rapid varying transients such as UTOP, it is recommended to use a small tolerance to maintain the solution accuracy. But very small values will result in very slow performance while using a large value will result in an inaccurate solution specifically at the peak power region. The performance of the ATS solution algorithm considering different tolerance levels is investigated using the UTOP transients of the MSFR benchmark. The solution of the ATS was performed with an initial time-step size of 0.05 ms, a gain value of 0.05, and the local error was estimated with the first derivative. Also, both the Picard iteration and the time-lagged coupling schemes were tested. Table 5.4 summarizes the UTOP transient results of the MSFR with 200 pcm reactivity insertion for the peak power, relative errors, computational time, and speedup ratio for tolerance levels ranging between 1.0×10^{-3} and 1.0×10^{-1}.

Figure 5.16 shows the power evolution obtained with the time-lagged coupling scheme for 200 pcm reactivity insertion, while Fig. 5.17 shows the power evolution obtained with the Picard iteration coupling scheme. For each coupling scheme, the ATS solution was compared with the reference solution obtained using the same coupling scheme. In the time-lagged coupling approach, as the tolerance level increases, the peak power is slightly overestimated, and the relative errors are increased due to the large predicted time-step size, which causes a lag in the thermal feedback. However, the Picard iteration coupling scheme leads to underestimation of the peak

power as the tolerance level is increased. This is because there is no delay of thermal feedback as in the time-lagged approach. A large tolerance level will result in more speedup of the solution but at the expanse of accuracy. It is recommended to use a tolerance level in the range of 1.0×10^{-3} and 5.0×10^{-2} for such transient with both coupling schemes.

Table 5.4. ATS Performance with Different Tolerance of MSFR UTOP-200 pcm Transient.

Case	Tolerance	Peak Power (P/Po)	Time to Peak (ms)	Relative Error (%)			No. Steps		Comp. Time	Speedup
				Peak	RMS	Integral	Total	Rejected		
Time-Lagged	**Reference**	44.682	6.15	-	-	-	29900	0	1.000	-
	1.00E-03	44.682	6.15	0.00	2.69	1.21	2721	40	0.122	8.22
	1.00E-02	44.708	6.23	0.06	7.78	3.71	1593	7	0.072	13.84
	1.00E-01	45.714	6.26	2.31	20.99	6.12	1253	1	0.046	21.92
Picard Iteration	**Reference**	43.927	6.15	-	-	-	29900	0	1.000	-
	1.00E-03	43.927	6.15	0.00	2.04	0.90	3323	386	0.259	3.87
	1.00E-02	43.897	6.15	0.07	5.69	2.64	1585	5	0.114	8.75
	1.00E-01	42.780	6.05	2.61	43.24	4.12	1258	1	0.071	14.16

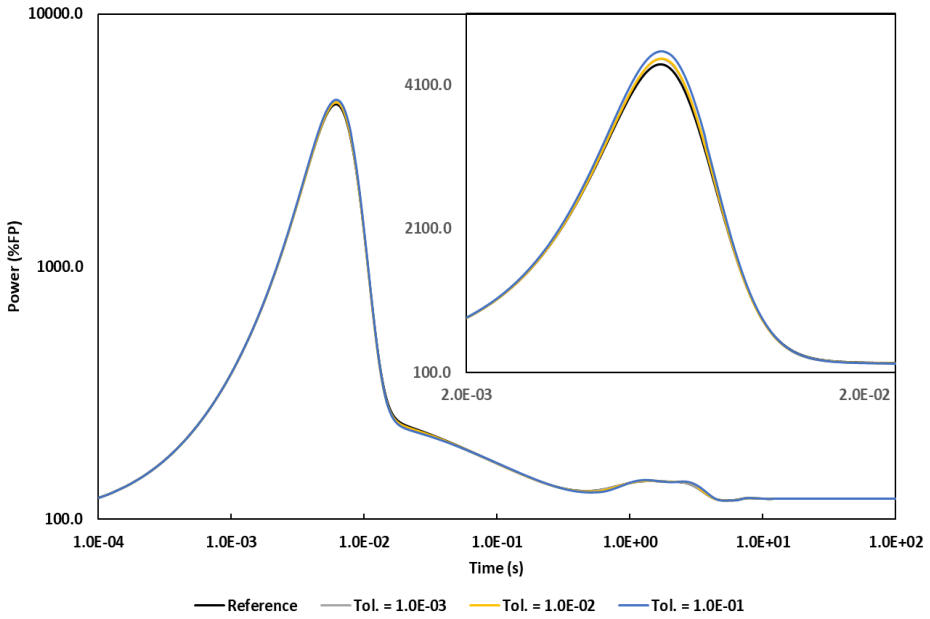


Figure 5.16. Power evolution after 200 pcm RI with TL scheme and different tolerance levels.

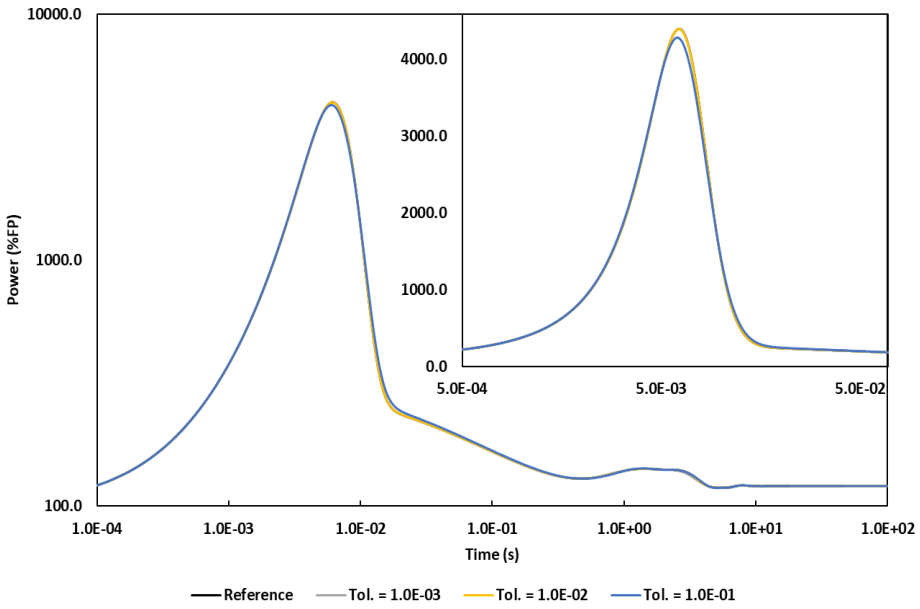


Figure 5.17. Power evolution after 200 pcm RI with PI scheme and different tolerance levels.

5.5.2. Moderate Transients

In this category, the transient is driven by the change in the fuel salt temperature due to the changes in heat removal rate at the heat exchanger, or the change in the fuel mass flow rate in the core due to changes or malfunction of the primary fuel salt pump. As discussed earlier in Chapter 3 for the MSFR benchmark, the analyzed flow-driven transients include UPOS and ULOF, and the temperature-driven transients include UFSOC and ULOHS. These transients are less severe than the reactivity-driven transients, and they occur over a longer period of about 10.0 s to 100.0 s compared to the UTOP transient where the power peaks within 5.0 ms. The selection of the optimum time step for such transients is also challenging, but since the rate of change of the power is much less than in the reactivity-driven transients, it is expected for the time-step size to be larger. Table 5.5 summarizes various transients results of the MSFR for the relative errors, the computational time, and the speedup ratio for the three local error estimates. The reference results were obtained with a constant time-step size of 5.0 ms. The solution of the ATS for all the local error estimates was performed with an initial time-step size of 5.0 ms, a gain value of 0.1, and a tolerance level of 5.0×10^{-3}. All the three error estimates provided a consistent result, and the ATS solutions show a huge reduction of the total time steps number and the computational time about 7 to 17 times compared to the reference solution. With selected gain and tolerance values, almost no steps were rejected for the second derivative approach, while the first derivative approach

requires a slightly longer time for the calculations, and more steps were rejected because the estimated error was slightly higher.

Table 5.5. ATS Performance During Moderate Varying Transients of MSFR Benchmark.

Case	Local Error Estimate	Relative Error (%)			No. Steps		Comp. Time	Speedup
		Max.	RMS	Integral	Total	Rejected		
UPOS	**Reference**	-	-	-	20000	-	1.000	-
	2nd FD	0.55	2.14	2.30	2103	0	0.104	9.58
	Exponential	0.52	2.15	2.51	2105	0	0.107	9.37
	1st FD	0.56	2.48	2.19	2136	23	0.110	9.06
ULOF	**Reference**	-	-	-	20000	-	1.000	-
	2nd FD	0.62	9.41	24.49	1015	0	0.069	14.58
	Exponential	0.61	8.94	22.50	1051	0	0.074	13.54
	1st FD	0.66	7.77	15.07	1365	620	0.137	7.30
UFSOC	**Reference**	-	-	-	20000	-	1.000	-
	2nd FD	2.70	8.18	4.95	1127	0	0.059	17.05
	Exponential	1.60	4.75	3.44	1127	0	0.059	17.00
	1st FD	0.94	5.35	2.34	1223	25	0.070	14.19
ULOHS	**Reference**	-	-	-	20000	-	1.000	-
	2nd FD	6.80	144.55	407.48	1102	17	0.079	12.67
	Exponential	6.74	142.19	398.33	1088	0	0.078	12.85
	1st FD	5.77	123.29	318.51	1692	528	0.147	6.80

The errors of the ULOF and ULOHS are slightly high because the power is decreased to a very low level and the difference from the reference tends to be magnified. However, this does not mean the solution of the ATS is wrong, but the relative difference is large compared to the power magnitude. The power evolution during ULOHS is shown in Fig. 5.18. It is clearly seen that the power agrees very well from all solutions all over the transient period. Figure 5.19 shows the variation of the time-step size over the transient period. At the beginning of the transient, small time-step size is required, but as the power becomes steadier, a larger time step is used. Also, Fig. 5.19 shows the local error estimated by the three schemes. The estimated local error by the second derivative approach is much smaller than the estimated local error by the first derivative approach, which means a larger predicted time-step size. All these approaches can provide accurate results for this type of transient.

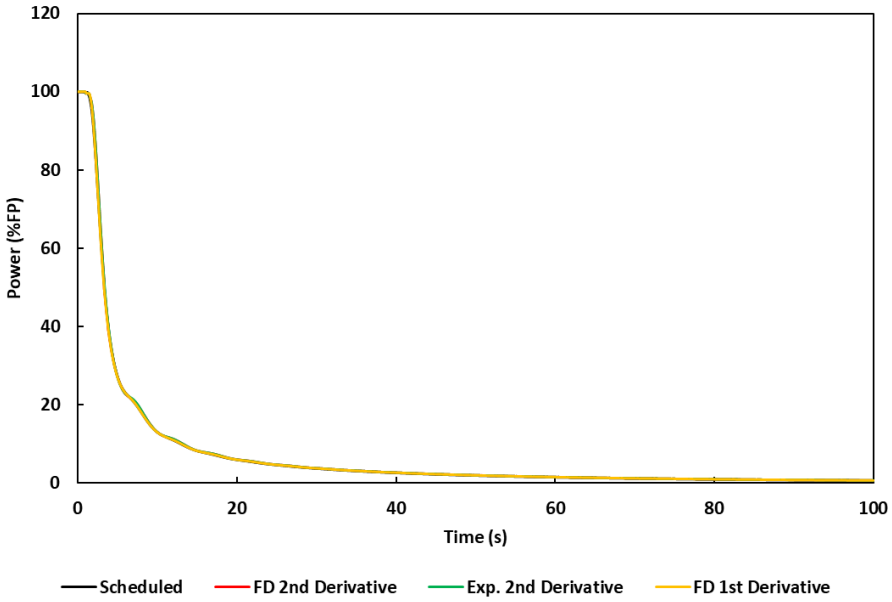


Figure 5.18. Power change during ULOHS transient with different error estimates.

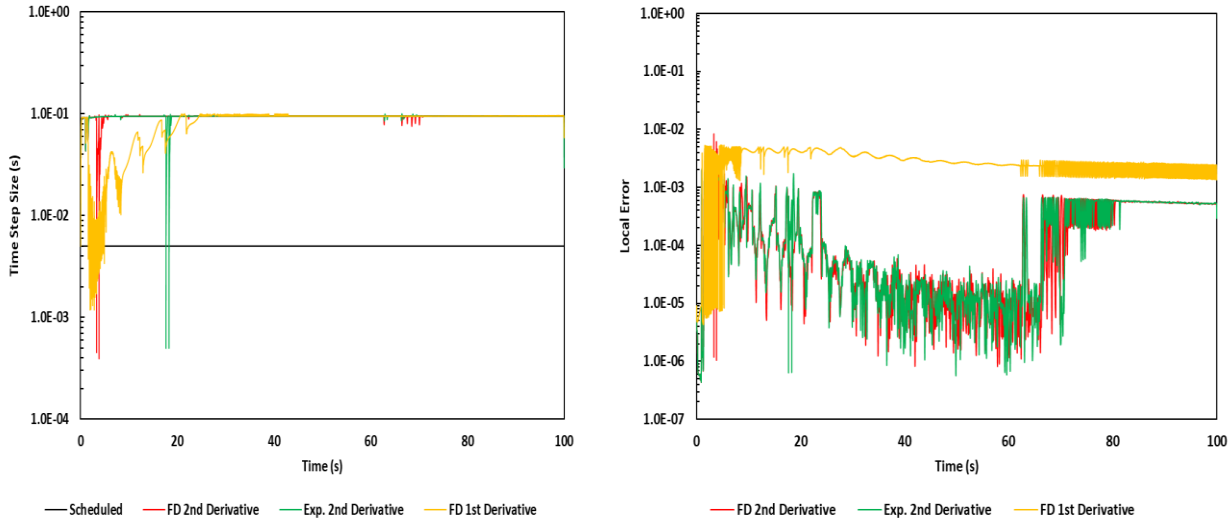


Figure 5.19. Time-step size change (left) and estimated local error (right) during ULOHS transient with different error estimates.

5.5.3. Slow Transients

For transients that happen over a long period of time, such as the natural circulation experiment of the MSRE, as discussed in Chapters 3 and 4, the power keeps varying slowly during the transient period. So, providing an optimum predefined time-step size to perform the calculations is very challenging. The ATS algorithm is very useful in such kind of transient, and it can reduce the computational time significantly by varying the time-step size as needed. To examine the performance of the ATS algorithm for slow varying transients, the natural circulation test of the MSRE experiment was analyzed.

Table 5.6 summarizes the natural circulation results of the MSRE for the peak power, the relative errors, the computational time, and the speedup ratio for the three local error estimates. The reference results were obtained with a constant time-step size of 0.1 s. The solution of the ATS for all the local error estimates was performed with an initial time-step size of 20.0 ms, a gain value of 0.2, and a tolerance level of 1.0×10^{-2}. The peak power is 348.5 kW, and it happens at 245.5 min of the transients. The relative error in the peak power is about 0.16% from all the three ATS approaches. A huge reduction in the computational time is gained with the ATS algorithm for such can kind of transients where the computational time is reduced from about 14 hours to less than an hour with speedup ratio of 20~25 times. The power change of the natural circulation test is compared with experimental data and the reference solution in Fig. 5.20. All three solutions provided consistent results, and they are validated against experimental data of the MSRE. Figure 5.21 shows the variation of the time-step size over the transient period and the local error estimated by the three schemes.

Table 5.6. ATS Performance During Natural Circulation Test of MSRE Experiment.

Local Error Estimate	Peak Power (kW)	Time to Peak (min)	Relative Error (%)			No. Steps		Comp. Time	Speedup
			Peak	RMS	Integral	Total	Rejected		
Reference	348.5	245.5	-	-	-	108000	-	1.000	-
2nd FD	349.0	244.3	0.16	29.45	0.40	3231	0	0.040	25.00
Exponential	349.0	244.4	0.16	29.57	0.41	3233	0	0.041	24.51
1st FD	349.0	244.3	0.15	48.46	0.39	3512	172	0.050	20.13

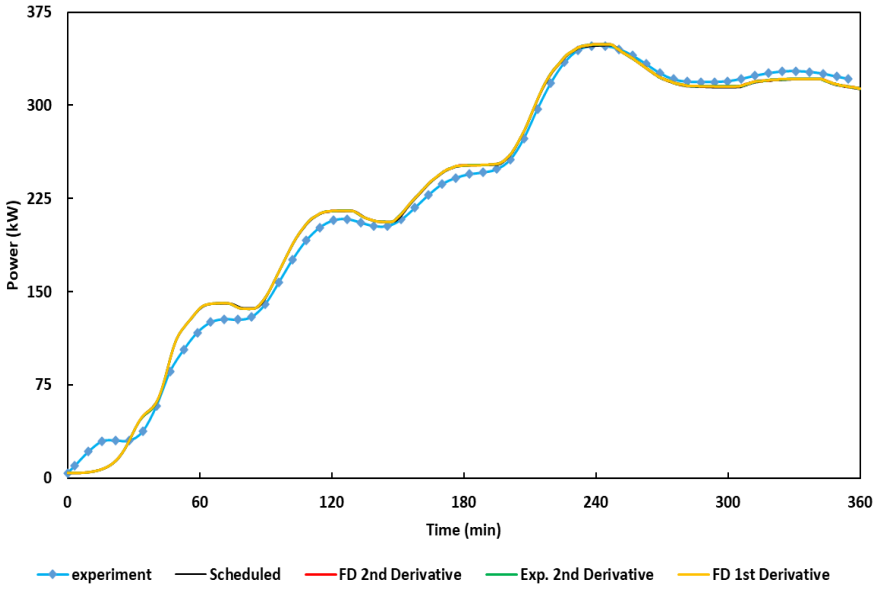


Figure 5.20. Power change during natural circulation test of MSRE with different error estimates.

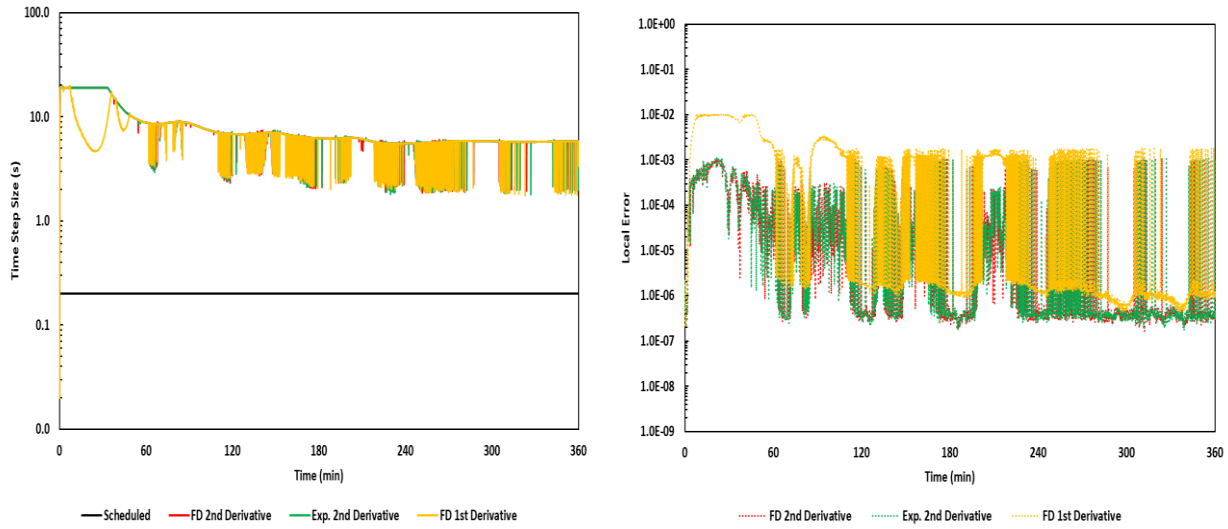


Figure 5.21. Time-step size change (left) and estimated local error (right) during natural circulation test of MSRE.

5.5.4. Localized Transients

For further testing and verification of the ATS solution algorithm, the performance of the ATS is tested with transients that are initiated due to a local event, such as a step movement of a control rod, or a channel blockage in channel-type MSR like MSRE. In these transients, the main contribution to the solution error is related to the region where the transient initial event happened because of the significant change in the reaction rates or the temperature in that region. In the analysis presented in the previous sections of this Chapter, the local error was estimated with the L^2 norm. However, for localized transient, it is more appropriate to use infinite norm which can represent the maximum solution error, which is expected to be at the initial event position. The infinite norm $\left(L^{\infty}\right)$ can be calculated as

$$\hat{R}_{\infty}^{n+1} = MAX\left\|\frac{\phi(t_n)-\phi(t_{n-1})}{\phi(t_n)}\right\|. \tag{5.42}$$

In this section, two localized transients will be simulated for the MSRE experiment, which is a channel-type reactor. The first transient considered is the step withdrawal of the control rod, which will result in massive power and temperature changes at positions near that control rod. The second transient is the channel blockage transient, which might lead to a large increase in the temperature in that region.

The control rod partial ejection or partial withdrawal was simulated for the MSRE experiment, where the reactor is operated at full power. Two cases were considered for this transient with step

withdrawal of the regulating rod of 10 cm and 20 cm. The withdrawal resulted in an insertion of a step reactivity of 240 pcm and 430 pcm, respectively. To examine the performance of the ATS algorithm for these localized transients, the control rod withdrawal of the MSRE experiment was analyzed for the two cases. Table 5.7 summarizes the results for the peak power, the relative errors, the computational time, and the speedup ratio for the L^2 and $L\infty$ norms. The reference results were obtained with a constant time-step size of 5.0 ms. The solution of the ATS was performed with an initial time-step size of 5.0 ms, a gain value of 0.1, and a tolerance level of 1.0×10^{-3}.

Table 5.7. ATS Performance for Localized Transients of MSRE Experiment.

Case	**Local Error Estimate**	Peak Power (P/P_0)	Time to Peak (s)	Relative Error (%)			No. Steps		Comp. Time	Speedup
				Peak	RMS	Integral	Total	Rejected		
CR-Withdraw (10 cm)	Reference	3.57	2.67	-	-	-	50000	-	1.000	-
	L^2	3.60	2.97	0.72	30.52	0.23	3454	743	0.246	4.06
	$L\infty$	3.60	2.85	0.57	28.79	0.22	3567	870	0.253	3.95
CR-Withdraw (20 cm)	Reference	12.52	0.74	-	-	-	50000	-	1.000	-
	L^2	12.53	0.76	0.05	55.26	0.34	5742	1500	0.353	2.83
	$L\infty$	12.53	0.74	0.04	49.10	0.30	6003	1674	0.368	2.72
Channel Blockage	Reference	1.21	78.95	-	-	-	25000	-	1.000	-
	L^2	1.21	78.92	0.01	39.57	0.22	3043	1107	0.271	3.68
	$L\infty$	1.21	78.72	0.08	26.08	0.18	3283	1277	0.280	3.58

Figures 5.22 and 5.24 show the power evolution for the 10 cm and 20 cm withdrawal cases, respectively. Both L^2 and $L\infty$ norms provide consistent results with the reference solution, and the peak relative error is less than 0.8%. The solution speedup is about 3 to 4 times, and more speedup was achieved with L^2 norm than $L\infty$. This is because the predicted time-step size by the L^2 is slightly larger than the predicted time-step size by $L\infty$ norm as shown in Fig. 5.23 and Fig. 5.25 for the time-step size of the 10 cm and 20 cm withdrawal cases, respectively. The reason for that is because the local error calculated with the L^2 norm is smaller than the $L\infty$ local error. Also, Figures 5.23 and 5.25 show the estimated local error by the L^2 and the $L\infty$ norms for the 10 cm and 20 cm withdrawal cases, respectively. Both error estimates used in the controller gave very accurate results with higher accuracy was obtained with the $L\infty$ norm. In case of full ejection of the control rod, it will be necessary to use the $L\infty$ norm for the local error estimation to maintain the accuracy of the solution.

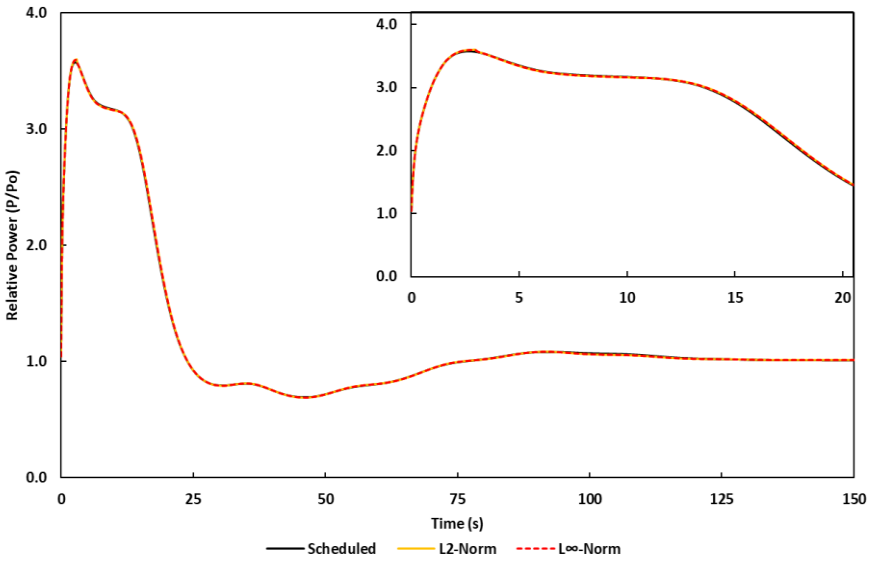


Figure 5.22. Power evolution after 10 cm withdrawal of control rod of MSRE.

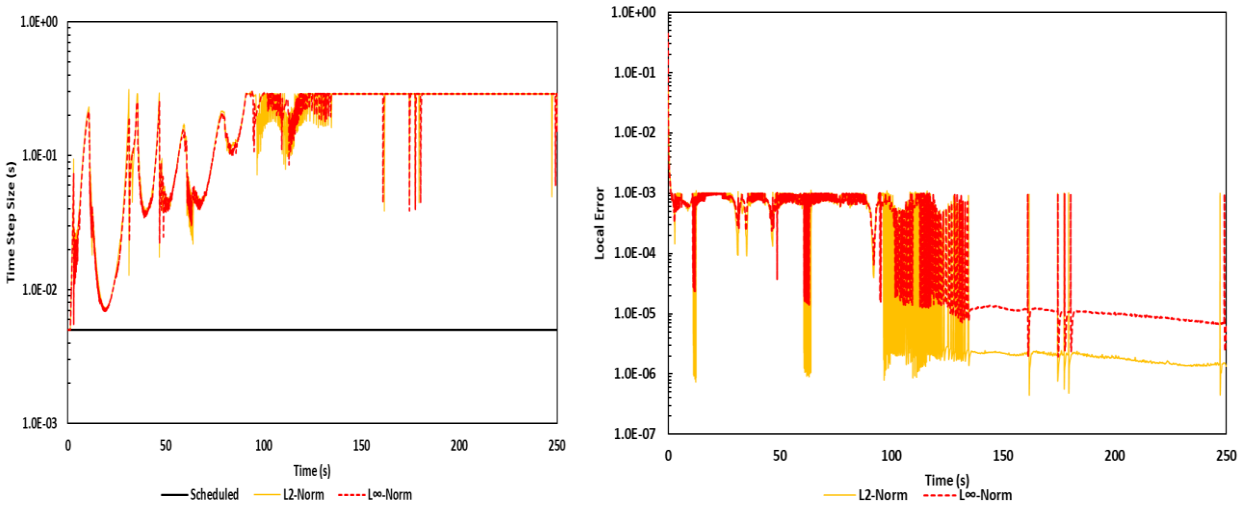


Figure 5.23. Time-step size variation (left) and estimated local error (right) after 10 cm withdrawal of control rod of MSRE.

The channel blockage and unblockage transients were simulated for the MSRE experiment at full power. This type of transient is very important in channel type MSR because the fuel salt flow rate at the blocked channel will be reduced significantly, and the fuel salt will become almost stagnant. As the fuel flow rate is reduced in the channel, the power will slightly increase because the losses in the delayed neutrons are reduced. Then, the power will decrease because of the negative thermal feedback due to the increased fuel salt and moderator temperatures. If the blocked channel is unblocked, the power will start increasing again until it returns to the initial power level

due to the positive reactivity introduced by the circulating fuel salt at lower temperature in the outer loop.

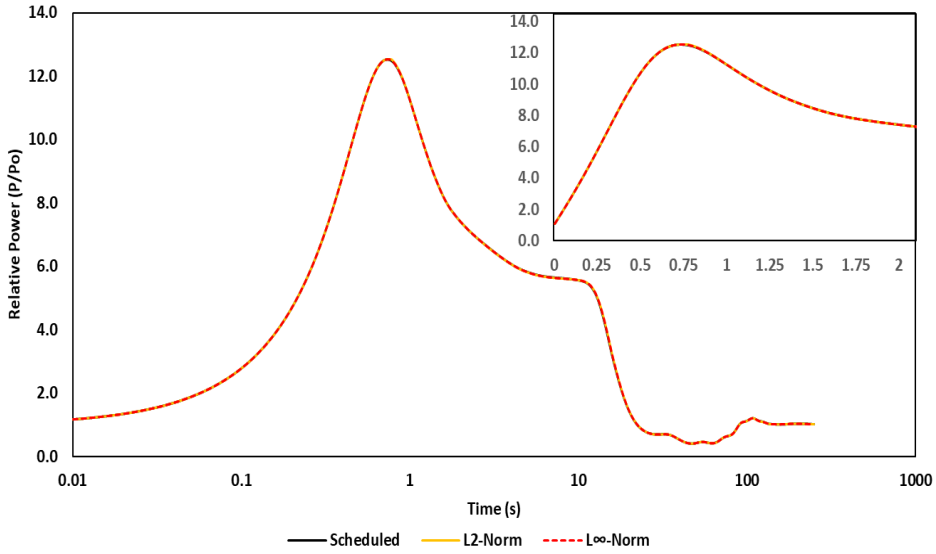


Figure 5.24. Power evolution after 20 cm withdrawal of control rod of MSRE.

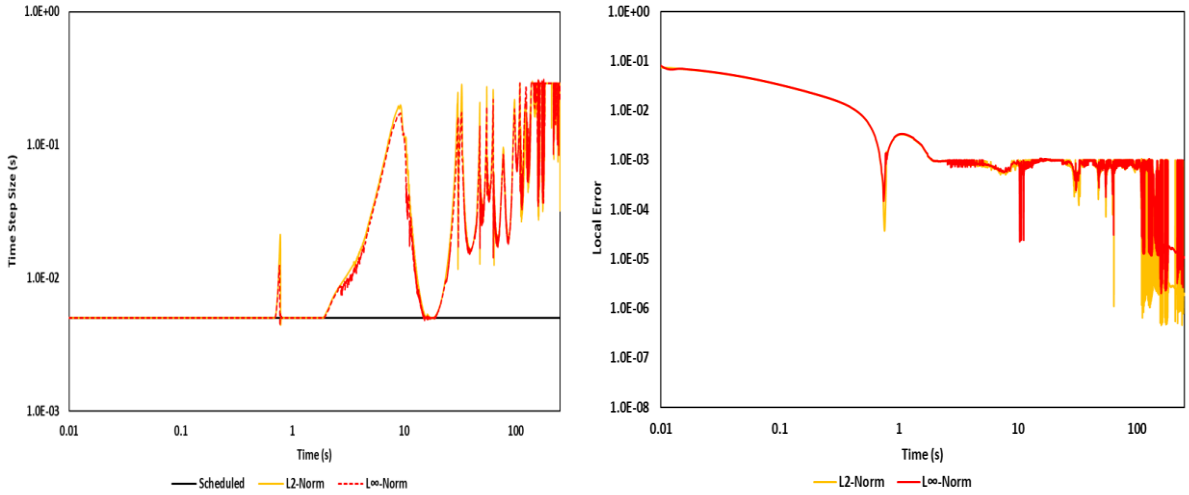


Figure 5.25. Time-step size variation (left) and estimated local error (right) after 20 cm withdrawal of control rod of MSRE.

This transient was simulated with PROTEUS-NODAL for the MSRE core assuming the central channels were completely blocked. After 50.0 s, it was assumed the channel was unblocked, and the transient was simulated up to 250.0 s. Table 5.7 summarizes the results of the channel blockage transient for the L^2 and $L\infty$ norms. The reference solution was obtained with a constant time step of 10.0 ms. The solution of the ATS was performed with an initial time-step size of 10.0 ms, a gain value of 0.1, and a tolerance level of 1.0×10^{-3}.

Figure 5.26 shows the power change after the central channel is completely blocked at the beginning of the transient and assuming the reactor was operated steadily at full power before that. The power is decreasing until 50.0 s, where the channel was assumed to be unblocked, and the power starts increasing and peaks at 1.21 times of its steady state power. The oscillatory behavior of the power after 50.0 s is due to the heated fuel salt from the blocked channel is flowing back into the core. Both L^2 and $L\infty$ norms provide consistent results with the reference solution, and the peak relative error is less than 0.08%, and the peak magnitude and location are matched very well with the reference solution. The solution speedup is about 3.5 times, and more speedup was achieved with L^2 norm than $L\infty$ for the same reason as discussed in the control rod withdrawal transient. The predicted time-step size and the estimated local error by the L^2 and $L\infty$ norms are shown in Fig. 5.27. Both error estimates used in the controller gave very accurate results with higher accuracy was obtained with the $L\infty$ norm.

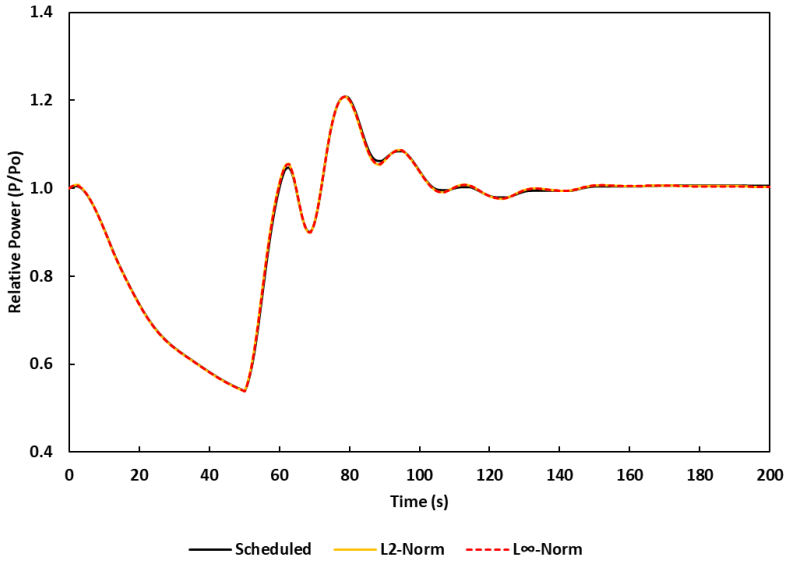


Figure 5.26. Power change after channel blockage and unblockage of MSRE.

5.6. Summary

An adaptive time-stepping solution algorithm was implemented in the PROTEUS-NODAL code for MSR transient analysis based on the control theory approach. The implemented algorithm was verified using the transient scenarios of the MSFR benchmark and validated with the natural circulation test of the MSRE experiment. The simulated transients were classified into three categories based on the rate of change of power with time. Another category was added for localized transients, where the transient is initiated due to a certain change at a certain position of

the reactor core. To test the performance of the ATS solution, it was compared with the reference solution that was obtained with a very fine time-step size. In all the simulated transients, the ATS solution shows a huge reduction in the computational time while ensuring the accuracy of the solution.

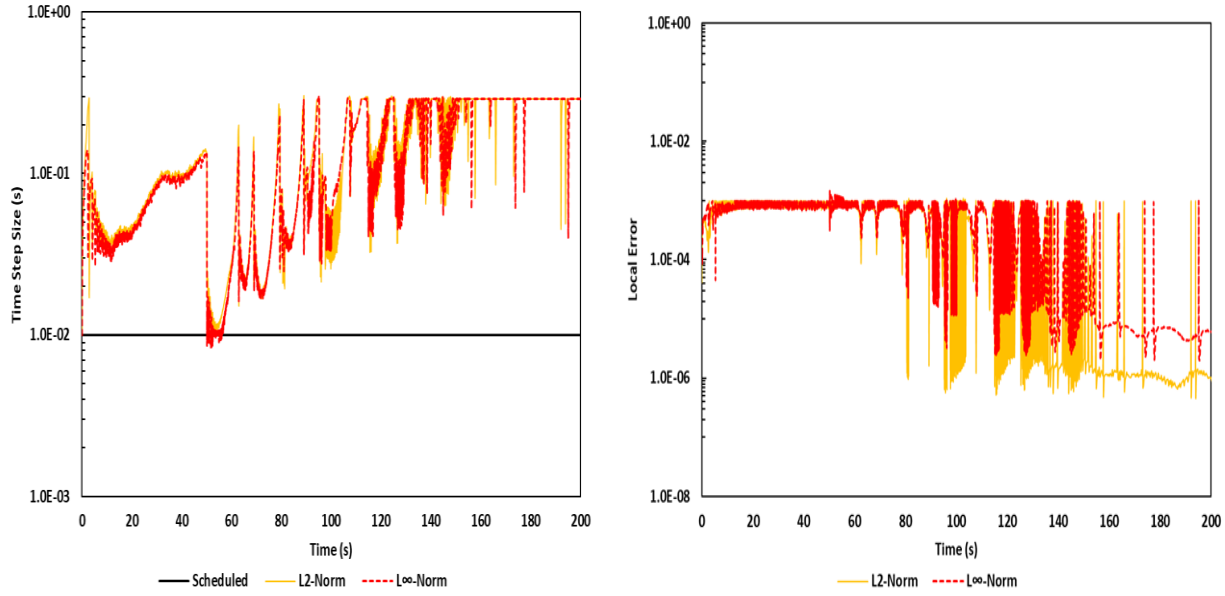


Figure 5.27. Time-step size variation (left) and estimated local error (right) during channel blockage and unblockage of MSRE.

A parametric study was performed on the controller parameters using the UTOP transient of the MSFR to determine the valid range of the gain and the solution tolerance, which are specified by the user. Very large gain or tolerance values might lead to a misprediction of the time-step size, and as a result inaccurate solution. Table 5.8 summarizes the recommended values of the controller parameters for each transient category. Also, various ways were used to estimate the local error of the solution, which is the most important parameter in predicting the time-step size. All the local error estimates provide a very accurate solution, and more speed up can be achieved with higher-order local error estimates. For the slow varying transients like the natural circulation test of the MSRE, a massive reduction in the computational time was achieved with the ATS solutions with a speedup ratio of about 24. For localized transient, it is recommended to use the infinite norm, because it will represent the maximum change of the system instead of the average change, so more accurate solution can be guaranteed as the predicted time-step size is slightly smaller. The ATS solution algorithm can be used in other applications in the nuclear field, as in the depletion calculation.

Table 5.8. Recommended Values of ATS Controller Parameters.

Parameter	Transient Type			
	Rapid	Moderate	Slow	Localized
Initial Step (ms)	0.05	5.0	20.0	5.0
Tolerance	1.00E-03	5.00E-03	1.00E-02	1.00E-03
Gain	0.05	0.1	0.2	0.1

Chapter 6. Fuel Cycle Analysis of MSRs

During the operation of the liquid fuel molten salt reactors, the buildup of the fission products in the fuel salt as the fuel is burned out will reduce the efficiency of the reactor, and it may require a large amount of the fissile component of the fuel salt to compensate the negative reactivity introduced by the fission products. Also, the fuel salt volume will increase significantly. The fission products can be soluble in the salt or non-soluble as in the gaseous fission products and some elements could precipitate. These phenomena make online fuel reprocessing is essential for the operation of MSRs to extract the gaseous and non-soluble fission products from the fuel salt. Also, some MSR designs have no control rods as in the fast spectrum MSRs, and the reactor is being operated with almost zero excess reactivity. To maintain criticality during the reactor operation, online feeding of the fissile and fertile components of the fuel salt is required. Online chemical reprocessing is a unique feature of the liquid fuel MSRs that allows continuous monitoring and control of the salt molar composition.

The fuel cycle analysis of the liquid fuel MSRs is slightly different from conventional reactors for many reasons. The drift of the fuel salt and its decay outside the core causes significant loss in the delayed neutron fraction, which should be considered in the depletion calculations of MSRs. Also, the continuous movement of the fuel salt in the reactor core and the outer loop makes the whole fuel salt almost exposed to the same average neutron flux. Since the fuel spent a significant amount of time outside the core without contributing to the power generation, so that amount of time should be considered in the fuel cycle analysis. The online chemical reprocessing of the fuel salt considering the refueling and fission products removal adds extra terms to the nuclide depletion equation of MSRs. Also, the thermal expansion and temperature feedback of the fuel salt have a non-negligible effect and coupled neutronics and thermal hydraulics are required for depletion calculations. Finally, the decay heat of the fuel salt has a large contribution to the total power generated by the fuel salt (about 4.0%), so it cannot be ignored since it will have a significant effect on the neutron flux normalization.

The main objective of this thesis is to develop reliable, verified, and validated neutronics capabilities of liquid fuel MSRs to the PROTEUS-NODAL code to be used for the design and safety analysis of new MSRs designs. These capabilities include steady state, transient, and fuel cycle analysis capabilities. To fulfill this objective, the fuel cycle analysis capability was implemented in PROTEUS-NODAL, so that the same reactor model can be used to perform the required calculations within a single code. In this Chapter, the implemented fuel cycle analysis capabilities in PROTEUS-NODAL for liquid fuel MSRs are discussed first, including the drift of the fuel salt, thermal feedback, and online reprocessing. This is followed by verification tests for stationary fuel and compared to reference Monte Carlo solution. Then, an analysis of the MSFR benchmark problem is presented and compared to other reference solutions obtained from the open literature. Finally, a summary and future development and applications are provided.

6.1. Introduction

Over the past years, isotopic point depletion tools like ORIGEN [107, 108, 109] and CINDER [110] have been developed for a detailed analysis of the radioactivity properties of depleted fuel. Also, they have been coupled to Monte Carlo codes like MCNP for detailed depletion calculations [111, 112, 113]. Coupled Monte Carlo and depletion code systems have been developed for the analysis of advanced systems due to the ability to model complex geometries, and accurate representation of nuclear data and heterogeneity effect. However, these coupled systems require external scripts to link Monte Carlo codes and depletion solvers which can be challenging. Recently, Monte Carlo codes have incorporated a built-in depletion calculation capability as in MCNP6 [114], McCARD [115], OpenMC [66], and Serpent [67] codes. The REBUS-3 code [116] has been developed at Argonne National Laboratory for 3-D fuel cycle analysis fast spectrum reactors, and it utilizes DIF3D [56, 72], VARIANT [55, 56], or TWODANT [117] for region dependent flux calculations. These codes are dedicated to typical stationery fueled reactors which cannot be used directly for modeling flowing fuel MSRs because of the online refueling and reprocessing during the operation of the reactor.

Several efforts were dedicated to the simulation of the MSR fuel cycle to address the unique characteristics of liquid fuel in the past few years. An MSR fuel cycle analysis code was developed by coupling MCNP code with the in-house depletion code REM, where the continuous removal of fission products was simulated as a fictitious decay [118]. Also, an extended version of the Serpent code [119] was developed for the MSRs fuel cycle analysis considering continuous removal and

external feeding with a reactivity control mechanism. A fuel cycle analysis procedure of MSRs was developed by directly modifying the fuel salt composition between two successive burnup steps based on SCALE/TRITON code [110, 121]. A similar MSR fuel cycle analysis procedure was adopted in other code systems which can be found in Ref. [13, 122, 123, 124]. Also, a modified version of the OpenMC code was developed by coupling OpenMC with an external code for MSR burnup calculations [125]. A code named FAMOS [59] was developed for MSR fuel cycle analysis by considering the reactor core and the external loop as two points instead of one point, and these two points were connected via inlet and outlet so that the transient times of the fuel salt in the core and the outer loop are considered properly. The last approach has been adopted in PROTEUS-NODAL for MSR depletion calculation and fuel cycle analysis since it accounts for the time spent by the fuel salt in the core and the outer loop accurately.

6.2. Depletion Calculation Method

In depletion analysis, the calculation of nuclides concentrations changes is performed by neglecting the delayed neutron precursors and assuming that the rapidly saturating fission products are at equilibrium concentrations. Also, the neutron flux fluctuations arising from the transients of rapidly saturating fission products, such as xenon and samarium, are not considered in the reactor burnup analysis. Instead, the time derivative of the neutron flux is assumed zero, and prompt and delayed neutrons are combined for stationary fuel. So that, the neutron flux is calculated as a steady state. However, for flowing fuel MSRs, the prompt and delayed neutrons should be separated, and the delayed neutron precursor equation is solved along with the neutron flux equation to account for the effect of the fuel drift and decay in the outer loop. Also, online fuel feeding and reprocessing need to be considered in solving the nuclide depletion equation. In this section, the depletion equation for stationary and flowing fuels are presented along with the solution method.

6.2.1. Depletion Equation for Stationary Fuel

For stationary fuel, the change in the atom density of nuclide i at position $\vec{r}$ and time t is determined by the transmutation equation, the so-called Bateman equation as

$$\frac{\partial}{\partial t} N_i(\vec{r},t) + \left(\lambda_i + \int_0^\infty \sigma_a^i(\vec{r},E,t)\phi(\vec{r},E,t)dE \right) N_i(\vec{r},t) = \sum_{j=1}^{I} \left[\gamma_{ij}\lambda_j + \sum_x \gamma_{ij}^x \int_0^\infty \sigma_x^j(\vec{r},E,t)\phi(\vec{r},E,t)dE \right] N_j(\vec{r},t) \quad , \tag{6.1}$$

where

N_i = Nuclide density of nuclide *i*.

σ_x^i = Microscopic cross section of reaction *x*.

σ_a^i = Absorption microscopic cross section.

γ_{ij}^x = Yield of nuclide *i* when nuclide *j* undergoes reaction *x*.

γ_{ij} = Yield fraction of nuclide *i* due to radioactive decay of isotope *j*.

λ_i = Decay constant of nuclide *i*.

The system of depletion equations for all nuclides can be written in operator form as

$$\frac{\partial}{\partial t}\mathbf{N}(\vec{r},t) = \mathbf{A}(\vec{r},t)\mathbf{N}(\vec{r},t), \tag{6.2}$$

where $\mathbf{N}(\vec{r},t)$ and $\mathbf{A}(\vec{r},t)$ are the nuclide density vector and the nuclide transmutation matrix at position $\vec{r}$ at time t, respectively. The nuclide transmutation matrix is defined as

$$\mathbf{A}_{ij}(\vec{r},t) = \begin{cases} \sum_x \gamma_{ij}^x \int_0^\infty \sigma_x^j(\vec{r},E,t)\phi(\vec{r},E,t)dE + \gamma_{ij}\lambda_j & (i \neq j) \\ -\int_0^\infty \sigma_a^i(\vec{r},E,t)\phi(\vec{r},E,t)dE - \lambda_i & (i = j) \end{cases}. \tag{6.3}$$

Integrating Eq. (6.2) over the reactor core or the material region that is being depleted, then, the following equation can be obtained

$$\frac{d\bar{\mathbf{N}}}{dt} = \bar{\mathbf{A}}\bar{\mathbf{N}}, \tag{6.4}$$

where $\bar{\mathbf{N}}(t)$ is the integral of the nuclide density vector over the region, and $\bar{\mathbf{A}}$ is the region averaged transmutation matrix of which elements can be written as

$$\left(\bar{\mathbf{A}}\right)_{ij} = \begin{cases} \sum_x \gamma_{ij}^x \bar{\sigma}_x^j(t)\bar{\phi}(t) + \gamma_{ij}\lambda_j & (i \neq j) \\ -\bar{\sigma}_a^j(t)\bar{\phi}(t) - \lambda_i & (i = j) \end{cases}, \tag{6.5}$$

where $\bar{\sigma}_x^i$ is the region-averaged one-group cross section, $\bar{\phi}$ is the integrated flux in the region, and $\bar{N}_j$ is the integrated nuclide density in the region and can be defined as

$$\bar{\sigma}_x^i(t) = \frac{1}{\bar{\phi}(t)\bar{N}_i(t)} \int_V \int_0^\infty \sigma_x^i(\vec{r},E,t)\phi(\vec{r},E,t)N_i(\vec{r},t)dEdr^3, \tag{6.6}$$

$$\bar{\phi}(t) = \int_V \int_0^\infty \phi(\vec{r},E,t)dEdr^3, \tag{6.7}$$

$$\bar{N}_j(t) = \int_V N_j(\vec{r},t)dr^3. \tag{6.8}$$

6.2.2. Depletion Equation for Flowing Fuel

For flowing fuel, the change in the atom density of nuclide i at position $\vec{r}$ and time t is determined by modifying Eq. (6.1) to account for the nuclide drift in the core and external feeding without considering the fuel salt reprocessing process as

$$\begin{aligned}\frac{\partial}{\partial t}N_i(\vec{r},t)+\nabla\cdot\left[\vec{u}(\vec{r})N_i(\vec{r},t)\right]+\left(\lambda_i+\int_0^\infty\sigma_a^i(\vec{r},E,t)\phi(\vec{r},E,t)dE\right)N_i(\vec{r},t)\\ =\sum_{j=1}^{I}\left[\gamma_{ij}\lambda_j+\sum_x\gamma_{ij}^x\int_0^\infty\sigma_x^j(\vec{r},E,t)\phi(\vec{r},E,t)dE\right]N_j(\vec{r},t)+F(r,t)\end{aligned}, \tag{6.9}$$

where $\vec{u}$ is the fuel salt velocity and F is the external nuclide feeding rate. The system of depletion equations for all nuclides can be written in operator form as

$$\frac{\partial}{\partial t}\mathbf{N}(\vec{r},t)+\nabla\cdot\left[\vec{u}(\vec{r})\mathbf{N}(\vec{r},t)\right]=\mathbf{A}(\vec{r},t)\mathbf{N}(\vec{r},t)+\mathbf{F}, \tag{6.10}$$

where the nuclide transmutation matrix $\mathbf{A}(\vec{r},t)$ in Eq. (6.10) is defined in Eq. (6.3). Usually, in depletion analysis of MSRs, the reactor core and the external loop regions are considered as a one-point since the fuel salt is assumed to be homogenous and uniformly mixed. However, the one-point model doesn't represent the inlet and outlet nuclide flow rates of the core accurately. In other words, the time spent by the fuel in the outer loop and the core region.

In Ref. [59], an alternative approach was suggested by representing the core and the external loop regions as two separate points and connected through the inlet and outlet nuclide flow rates in the core region. This model can accurately represent the time spent by the fuel in the core and the external loop regions and it is adopted in the current implementation in PROTEUS-NODAL. By integrating Eq. (6.10) over the reactor core and the external loop respectively and using the divergence theorem, the following two points model can be obtained

$$\begin{aligned}\frac{d\bar{\mathbf{N}}_{\mathrm{C}}}{dt}&=\bar{\mathbf{A}}_{\mathrm{C}}\bar{\mathbf{N}}_{\mathrm{C}}+\dot{\mathbf{N}}_{in}-\dot{\mathbf{N}}_{out}\\ \frac{d\bar{\mathbf{N}}_{\mathrm{E}}}{dt}&=\bar{\mathbf{A}}_{\mathrm{E}}\bar{\mathbf{N}}_{\mathrm{E}}+\dot{\mathbf{N}}_{out}-\dot{\mathbf{N}}_{in}+\bar{\mathbf{F}}\end{aligned}, \tag{6.11}$$

where $\bar{\mathbf{N}}_{\mathrm{C}}(t)$ and $\bar{\mathbf{N}}_{\mathrm{E}}(t)$ are the integral of the nuclide density vector over the core and the external loop, respectively. $\bar{\mathbf{A}}_C$ is the core averaged transmutation matrix as given in Eq. (6.5), and $\bar{\mathbf{A}}_{\mathrm{E}}$ is the external loop averaged transmutation matrix considering decay process only. Since the fuel reprocessing process is performed out of the core region, it can be simulated with the equivalent effective decay constant as discussed in Ref. [118], and it should be added into the external loop

averaged transmutation matrix $\bar{\mathbf{A}}_{\mathrm{E}}$. If the entire salt in the primary circuit system is reprocessed within the time T_r, then the amount dN_e of element e with total inventory N_e is extracted during dt is proportional to the quantity $N_e \times dt/T_r$ with extraction efficiency of ε_e. Then, the reprocessing process of the element e can be equivalent to an effective decay process with a decay constant $\lambda_e = \varepsilon_e / T_r$ [118], and it can be expressed as a decay relation as

$$\frac{dN_e}{T_r} = \frac{\varepsilon_e}{T_r} N_e = \lambda_e N_e . \tag{6.12}$$

The removal period $T = T_r/\varepsilon_e$ usually represents the amount of time to completely remove the element e from the whole primary system. In Eq. (6.11), $\dot{\mathbf{N}}_{in}$ and $\dot{\mathbf{N}}_{out}$ are the nuclide number vectors flowing from the external loop into the and from the core into the external loop per unit time, respectively. $\dot{\mathbf{N}}_{in}$ and $\dot{\mathbf{N}}_{out}$ are proportional to the inlet and outlet mass flow rate $\dot{m}_{in}$ and $\dot{m}_{out}$ of the core, respectively, because of the homogeneous fuel salt composition assumption. In steady state operation, the inlet and outlet mass flow rates are equal and can be denoted as $\dot{m}$. Furthermore, the incoming and outgoing nuclide flow rates can be represented in terms of the time spent by the fuel salt in the external loop τ_E and the core τ_C as

$$\dot{\mathbf{N}}_{in} = \frac{1}{\tau_E} \bar{\mathbf{N}}_{\mathrm{E}} = \frac{\dot{m}}{m_{\mathrm{E}}} \bar{\mathbf{N}}_{\mathrm{E}} , \tag{6.13}$$

$$\dot{\mathbf{N}}_{out} = \frac{1}{\tau_C} \bar{\mathbf{N}}_{\mathrm{C}} = \frac{\dot{m}}{m_{\mathrm{C}}} \bar{\mathbf{N}}_{\mathrm{C}} , \tag{6.14}$$

where m_{E} and m_{C} is the total fuel salt mass in the external loop and the core regions, respectively. Using Eq. (6.13) and Eq. (6.14) and considering fuel salt reprocessing as an effective decay in the external loop averaged transmutation matrix $\bar{\mathbf{A}}_{\mathrm{E}}$, Eq. (6.11) can be simplified as

$$\frac{d}{dt}\begin{bmatrix} \bar{\mathbf{N}}_{\mathrm{C}} \\ \bar{\mathbf{N}}_{\mathrm{E}} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{A}}_{\mathrm{C}} - (\tau_{\mathrm{C}}^{-1})\mathbf{I}_n & \left(\tau_{\mathrm{E}}^{-1}\right)\mathbf{I}_n \\ (\tau_{\mathrm{C}}^{-1})\mathbf{I}_n & \bar{\mathbf{A}}_{\mathrm{E}} - \left(\tau_{\mathrm{E}}^{-1}\right)\mathbf{I}_n \end{bmatrix} \cdot \begin{bmatrix} \bar{\mathbf{N}}_{\mathrm{C}} \\ \bar{\mathbf{N}}_{\mathrm{E}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \bar{\mathbf{F}} \end{bmatrix}, \tag{6.15}$$

where $\mathbf{I}_n$ is the identity matrix of size N. By denoting the total nuclide number vector, the transmutation matrix, and the feeding source vector as $\mathbf{N}$, $\mathbf{A}$, and $\mathbf{F}$, respectively, Eq. (6.15) can be written in a compact form as

$$\frac{d}{dt}\mathbf{N} = \mathbf{A}\mathbf{N} + \mathbf{F}. \tag{6.16}$$

Considering a constant external feeding, the inhomogeneous system of differential equations can be converted into a homogeneous system as

$$\frac{d}{dt}\tilde{\mathbf{N}} = \tilde{\mathbf{A}}\tilde{\mathbf{N}}, \tag{6.17}$$

where $\tilde{\mathbf{N}}$ vector and $\tilde{\mathbf{A}}$ matrix are defined as

$$\tilde{\mathbf{N}} = \begin{bmatrix} \mathbf{N} \\ 1 \end{bmatrix}, \qquad \tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & \mathbf{F} \\ \mathbf{0} & 0 \end{bmatrix}. \tag{6.18}$$

The two points model for MSR depletion calculations resulted in doubling the size of the nuclide transmutation matrix $\mathbf{A}$ as $2N \times 2N$. In MSR calculations, the number of the nuclide considered in the decay chain is quite large, and this will increase the computational time significantly, especially with criticality search.

If the fuel salt is stationary as in certain designs of MSRs or there is no reprocessing, then the one-point model is a more valid and faster approach. Also, for flowing fuel MSR with the fuel salt spent almost the same time in the core and the external loop, the one-point model can provide accurate results as the two points model. The one-point model can be derived by integrating Eq. (6.10) over the total volume of the reactor core and the external loop, and considering the amount of the fuel salt that is being irradiated in the reactor core, the following nuclide transmutation matrix for the one-point model can be obtained as

$$\left(\bar{\mathbf{A}}\right)_{ij} = \begin{cases} \alpha \sum_x \gamma_{ij}^x \bar{\sigma}_x^j(t)\bar{\phi}(t) + \gamma_{ij}\lambda_j & (i \neq j) \\ -\alpha \bar{\sigma}_a^j(t)\bar{\phi}(t) - \lambda_i & (i = j) \end{cases}, \tag{6.19}$$

The transmutation matrix presented in Eq. (6.19) represents the core and the external loop as one-point. Factor α is the fraction of the fuel salt in the core to the whole system (core and external loop), and it is added to represent the amount of the fuel salt is being irradiated during the depletion calculations period. If the fuel reprocessing is considered, then, the effective decay process with a decay constant $(1-\alpha)\lambda_e$ needs to be added to the nuclide transmutation matrix. The size of the nuclide transmutation matrix is $N \times N$ and considering the number of nuclides in the decay chain, which is more than a thousand nuclides, so solving a system with the above nuclide transmutation matrix is much faster.

6.2.3. Solution Method of Depletion Equation

The solution of the homogeneous system of differential equations as in Eq. (6.4) or Eq. (6.17) can be solved formally using the matrix exponential function with a given initial nuclide vector, $\bar{\mathbf{N}}(0) = [\bar{N}_1(0), \cdots, \bar{N}_n(0), 1]^{\mathrm{T}}$, as

$$\tilde{\mathbf{N}}(t) = \exp(\tilde{\mathbf{A}}t)\tilde{\mathbf{N}}(0)\,. \tag{6.20}$$

For the computation of the matrix exponential in Eq. (6.20), there are many possible approaches in the literature from classical results in analysis, approximation theory, and matrix theory [126, 127]. These methods can be categorized as series methods, ordinary differential equation methods, polynomial methods, and matrix decomposition methods. Among these various methods, the Taylor series or Padé approximation with a scaling and squaring algorithm is known to be one of the most effective methods [116, 128]. Recently, the Chebyshev rational approximation method (CRAM) [129, 130] is widely used to obtain matrix exponential as in Serpent and OpenMC Monte Carlo codes due to the fast and accurate solution of the matrix exponential for a large system of nuclides. Also, the Krylov subspace method is one of the most popular tools for solving large sets of linear and nonlinear equations and computing the exponential of a matrix with a large size because it does not involve a matrix-matrix multiplication or inversion. In PROTEUS-NODAL, the matrix exponential $\exp(\tilde{\mathbf{A}}t)$ is determined using the CRAM option of the open-source software package for computing matrix exponentials EXPOKIT [131], which is used in several studies for nuclide depletion calculations as in Ref. [59, 124].

6.3. Solution Approach for MSRs Depletion

In PROTEUS-NODAL, the solution of the MSR depletion equations is obtained by decoupling the neutron flux equation and nuclide depletion equation at each time point. So, the neutron flux equations and the nuclide depletion equations are solved alternatively. Like coupling the neutronics and thermal hydraulics for transient analysis, several approaches can be adopted here starting from the time-lagged approach where each system of equations is solved once at each time point and marching with time, but this approach requires a small time-step size. Another approach is employing the Picard iteration scheme, where the system of equations is solved iteratively until a certain criterion is satisfied, which can be computationally expensive, especially for depletion calculations even it allows a larger time-step size. A more efficient method for depletion calculations is to use a predictor-corrector method that allows longer time steps than the time-

lagged and shorter than Picard iteration so that it compromises the number of steady state calculations of the neutron flux.

In the predictor-corrector approach, the total depletion period of length T is divided into subintervals Δt_n, $n = 1, \cdots, N_t$. Then, the nuclide depletion equations are solved using the solution of the neutron flux of the previous step ϕ^{n-1} to obtain the predicted nuclide concentrations $\mathbf{N}^P$. Using the predicted nuclide concentrations, the predicted neutron flux ϕ^P is calculated. These two steps are called predictor step, and they are used to determine the correct solution. Algorithm 6.1 describes the predictor-corrector steps implemented in PROTEUS-NODAL for depletion calculations. In the corrector step, the corrected nuclide concentrations $\mathbf{N}^n$ are obtained using the average of the neutron flux $\overline{\phi}$ of the previous step and the predicted solution as

$$\bar{\phi}\left(\vec{r},E\right)=\frac{1}{2}\left(\phi^{n-1}\left(\vec{r},E\right)+\phi^{P}\left(\vec{r},E\right)\right). \tag{6.21}$$

Once the nuclide concentrations $\mathbf{N}^n$ are calculated, the corrected neutron flux ϕ^n is calculated using the steady state solver, and it is stored for the next time step calculations. Figure 6.1 shows an illustration of the predictor-corrector coupling scheme of the nuclide depletion and the steady state solution of the neutron flux implemented in PROTEUS-NODAL. It should be mentioned that the steady state solver can invoke the neutronics solver along with the thermal hydraulics solver if the coupled calculations are required. The neutronics and thermal hydraulics solvers are tightly coupled, as discussed in Chapter 2.

6.3.1. Criticality Search and Salt Control

Some designs of MSRs have no control rods to control the reactor like fast spectrum reactors, instead, the reactor is being controlled by adjusting the flow rate across the core, and in case of an accident, the fuel salt will be drained to a decay tank usually lies below the reactor core. Such a control mechanism can be efficient for small and slow changes and if the reactor has almost no excess reactivity. In conventional nuclear reactors designs, the core has several neutron absorber rods to regulate the reactor and safely shut down the reactor for refueling or during accidents. The reactor has a large excess reactivity to allow operating the reactor for sufficiently enough time before refueling the core with new fuel. As the reactor fuel is burned out, the regulating control rods are withdrawn from the reactor core to compensate for fuel burnup, and to maintain criticality. However, this is not the case for MSRs with online refueling, where the reactor is maintained

critical by adding fissile material while the reactor is in operation. In ideal situations, this means that there is no need to shut down the reactor for refueling so that the reactor can be operated during the lifetime of the reactor continuously.

Algorithm 6.1. Predictor-corrector solution algorithm.

1. Perform nuclide depletion calculations to obtain $\mathbf{N}^P$ with the previous time step ϕ^{n-1}
2. Update macroscopic cross sections Σ^P using $\mathbf{N}^P$
3. Perform steady state calculations to obtain the predicted ϕ^P
   If (Consider Thermal Feedback) then
      Perform neutron flux calculations to obtain ϕ^P
      Perform TH calculations to update $(T,\rho,u)^P$
      Update cross sections $(\Sigma)^P$ using $(T,\rho)^P$
   Else
      Perform neutron flux calculations to obtain ϕ^P
   End If
4. Calculate the averaged neutron flux $\bar{\phi} = \left(\phi^{n-1} + \phi^P\right)/2$
5. Calculate corrected nuclide concentrations $\mathbf{N}^n$ using $\bar{\phi}$
6. Perform steady state calculations to obtain the corrected ϕ^n
   If (Consider Thermal Feedback) then
      Perform neutron flux calculations to obtain ϕ^n
      Perform TH calculations to update $(T,\rho,u)^n$
      Update cross sections $(\Sigma)^n$ using $(T,\rho)^n$
   Else
      Perform neutron flux calculations to obtain ϕ^n
   End If
7. If $(t = t_{end})$ terminate calculations Else Go back to 1

In order to maintain criticality for this type of MSRs, the feeding rate of the fissile isotopes must be determined. In PROTEUS-NODAL, the criticality is maintained through searching for the fissile material feeding rate F by adopting the secant method, so that the targeted eigenvalue $k_{eff}^{target} = 1.0$ is achieved within a user-specified tolerance (i.e., ± 10 pcm) after m criticality search iterations according to the following relation

$$F^{m+1} = \frac{1}{k_{eff}^m - k_{eff}^{m-1}}\left[\left(k_{eff}^{target} - k_{eff}^{m-1}\right)F^m - \left(k_{eff}^{target} - k_{eff}^m\right)F^{m-1}\right]. \tag{6.22}$$

At the beginning of the depletion calculation of MSRs, the criticality searching process requires several iterations, and as the inventory of the fuel salt reaches equilibrium conditions the number of iterations will be reduced significantly since the fissile feeding rate will remain almost constant. Algorithm 6.2 describes the criticality search algorithm implemented in PROTEUS-NODAL for MSRs depletion calculations.

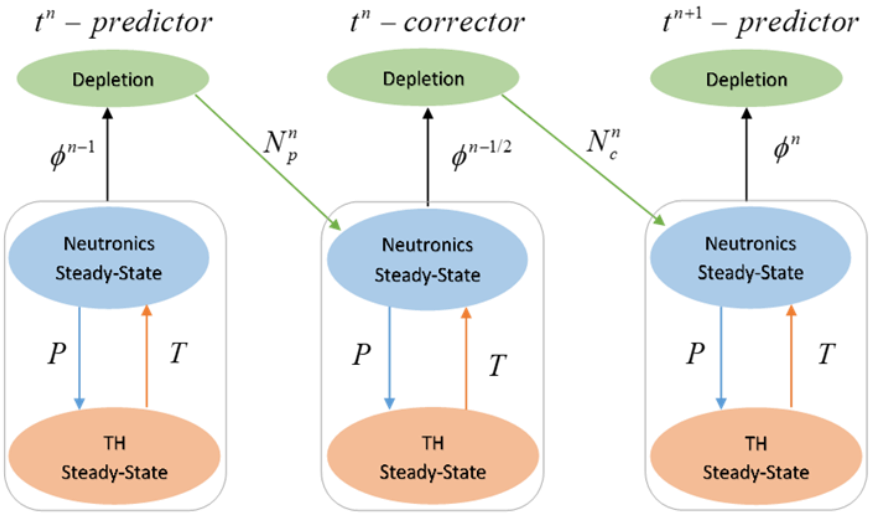


Figure 6.1. Illustration of predictor-corrector coupling scheme in PROTEUS-NODAL.

As the fissile isotopes are being added to the fuel salt, the molar concentration of the fuel salt will be altered significantly, especially with the buildup of the fission products. The depleted fissile material in the fuel salt is being compensated by feeding fissile material externally and through the breeding of the fertile isotopes in the fuel salt. In most designs, the fed material will consist of fissile and fertile isotopes, and the fertile material is fed to ensure the ratio between the amount of the actinides ($90 \le Z \le 99$) and the amount of the other salt components such as Lithium (in LiF or LiCl salts) and fission products are maintained at the eutectic point. So, the fissile feed rate is determined according to Eq. (6.22), while the fertile feed rate is determined by satisfying the following condition

$$\sum_{Z=90}^{99} N^z \Big/ \sum_{Z=1}^{99} N^z = C_{HM}\,, \tag{6.23}$$

where Z is the atomic number of the element, and the constant C_{HM} is the total molar fraction of the heavy metal in the fuel salt. In Eq. (6.23), the salt carriers (like Fluoride or Chloride) are excluded from the denominator.

Algorithm 6.2. Criticality and fissile feeding rate search algorithm for MSR fuel cycle.

1. If (Search for Criticality) then
2. Apply algorithm 6.1 to calculate nuclide concentrations $\mathbf{N}^n$ and k_{eff}^m at the end of step
3. Do while $(k_{eff}^m \neq k_{eff}^{\text{target}} \ \& \ m \leq \text{Max.Iter})$

   Adjust the feed rate of the fissile material F^m

   Apply salt control to maintain fuel salt molar composition

   Apply algorithm 6.1 to calculate nuclide concentrations $\mathbf{N}^{n,m}$ and k_{eff}^m

   $m = m + 1$

   End Do
4. Else

   Apply algorithm 6.1 to calculate nuclide concentrations $\mathbf{N}^n$ and k_{eff} at the end of step

   End If
5. If $\left(t = t_{end}\right)$ terminate calculations Else Go back to 1

As the fission products are building up in the fuel salt, they can be soluble and insoluble in the salt. The insoluble fission products are continuously removed from the salt, while the soluble fission products should replace other components of the salt. In the MSFR design, the initial fuel salt composition is LiF-UF_4-ThF_4. After irradiating the fuel salt, fission products and other heavy metals will be added to the initial composition. To maintain the fuel salt composition at the eutectic point, the Lithium concentration is continuously adjusted. In the salt control calculations in PROTEUS-NODAL, the Lithium concentration is modified according to the following relation

$$\frac{dN_l}{dt} + \left(\lambda_l + \int_0^\infty \sigma_a^l \phi dE\right) N_l = \sum_{j=1}^{I} \left[\gamma_{lj}\lambda_j + \sum_x \gamma_{lj}^x \int_0^\infty \sigma_x^j \phi dE\right] N_j - N_{FP}, \tag{6.24}$$

where l represents the nuclides of the element being modified (i.e., ^{6}Li and ^{7}Li), and N_{FP} is the concentration of the produced fission products, which can be expressed as

$$N_{FP} = \sum_{k=90}^{99} \left(\sum_{l=FP} \gamma_{lk}^f\right) N_k \int_0^\infty \sigma_f^k \phi dE\,. \tag{6.25}$$

6.3.2. Neutron Flux Normalization

One of the most significant points in depletion calculations is the neutron flux normalization, which affects the consumption of the fissile material in the core in order to maintain the desired power level. Different normalization will lead to a different consumption rate of the fissile nuclides or burnup. Usually, only the energy deposited from the fission reaction is considered in

normalizing the neutron flux, which will lead to over-depleting of burnable materials. The normalization factor f of the neutron flux considering the energy deposited from fission reaction only can be given as

$$f_{fission}(t) = \frac{P(t)}{\sum_{k=1}^{K} V_k \sum_{i=1}^{Niso} \kappa_{f,i} N_{i,k}(t) \sum_{g=1}^{G} \sigma_{f,g}^{i,k} \phi_{g,k}(t)}, \tag{6.26}$$

where $P(t)$ is the reactor power output at time t, V_k is the volume of region k, and $\kappa_{f,i}$ is the energy released per fission of isotope i. A more accurate expression can be obtained by considering the indirect components that contribute to the generated heat, such as the energy released due to the capture reaction $\kappa_{c,i}$. In this way, the energy released from fission and capture reactions contributes to total generated power, so the normalization factor of the neutron flux can be written as

$$f_{total}(t) = \frac{P(t)}{\sum_{k=1}^{K} V_k \left(\sum_{i=1}^{Niso} N_{i,k}(t) \sum_{g=1}^{G} \kappa_{f,i} \sigma_{f,g}^{i,k} \phi_{g,k}(t) + \sum_{i=1}^{Niso} N_{i,k}(t) \sum_{g=1}^{G} \kappa_{c,i} \sigma_{c,g}^{i,k} \phi_{g,k}(t) \right)}. \tag{6.27}$$

Another important component is the released energy from the decay of the fission products, which contributes to about 5% to 7% of the rated power. At the beginning of the reactor operation or the depletion calculations with fresh fuel, the amount of the fission products is very small, and they start building up as the fuel is burning out. As a result, heat generated from the decay of the fission products starts building up until it reaches saturation level after ~100 days. In order to normalize the neutron flux accurately, the decay heat of the fission products must be subtracted from the total power due to fission deposited energy and capture released energy. Considering the decay heat Q_d, the normalization factor of the neutron flux can be written as

$$f_{total}(t) = \frac{P(t) - Q_d(t)}{\sum_{k=1}^{K} V_k \left(\sum_{i=1}^{Niso} N_{i,k}(t) \sum_{g=1}^{G} \kappa_{f,i} \sigma_{f,g}^{i,k} \phi_{g,k}(t) + \sum_{i=1}^{Niso} N_{i,k}(t) \sum_{g=1}^{G} \kappa_{c,i} \sigma_{c,g}^{i,k} \phi_{g,k}(t) \right)}. \tag{6.28}$$

The decay heat rate Q_d at time t can be calculated from the energy released per decay as

$$Q_d(t) = \sum_{k=1}^{K} V_k \sum_{i=1}^{Niso} q_{d,i} \lambda_i N_{i,k}(t), \tag{6.29}$$

where q_d, λ, and N are the energy released per decay, decay constant, and concentration of nuclide i, respectively. In PROTEUS-NODAL, the flux can be normalized using any of the normalization formulas in Eqs. (6.26), (6.27), or (6.28) based on the user specifications. Also, all the nuclides in the depletion chain are being tracked, which allows the calculation of the decay heat explicitly, as provided in Eq. (6.29). In MSRs with fuel reprocessing, some of the fission products are being extracted from the fuel salt, so that they don't contribute to the generated decay heat by the fuel salt in the system, and they must be excluded from the total generated decay heat, as discussed later in this chapter.

6.3.3. Nuclide Decay Chains

Detailed depletion chain that contains 1356 nuclides of which 93 actinides and 1157 fission products are adopted from ENDF/B-VII.1 cross section data, which includes fundamental data for incident neutron, decay, and fission product yields. The current depletion chain used in the depletion model is adopted from the fast spectrum reactors depletion chain distributed with OpenMC code [66]. For each nuclide, the depletion chain file includes the possible transmutation reactions, the reaction Q value, and the daughter nuclides. Also, the decay modes, branching ratios, products, decay constants, and energy released per decay of each nuclide. For fissionable nuclides, the yield of the fission products based on the incident neutron energy is provided.

In the implemented model for depletion calculations, all the nuclides in the depletion chain are tracked, which allows calculation of the decay heat explicitly based on the information provided in the depletion chain. In order to perform the depletion calculations, the multigroup cross sections are prepared using either OpenMC or MC^2-3 in ISOTXS format so that the one group reaction rates required to construct the transmutation matrix (*Capture, Fission, (n,p), (n,α),* and *(n,2n)*) can be calculated along with multigroup neutron flux obtained from the steady state calculations.

6.4. Numerical Results

In this section, the developed depletion capabilities of PROTEUS-NODAL are verified and tested with stationary and flowing fuel problems and compared with the reference results. The first test is performed for stationary fuel case without considering online fuel reprocessing to verify the developed capabilities in comparison with depletion calculation results of the Monte Carlo code OpenMC. Then, the fuel cycle analysis of the MSFR benchmark is performed considering online

fuel reprocessing and refueling, and the solution of PROTEUS-NODAL is compared with the reference results obtained from the MSFR benchmark report [13].

6.4.1. Verification Test

To verify the implemented depletion capabilities of PROTEUS-NODAL, a small artificial problem was developed with a fast spectrum and stationary fuel without considering online reprocessing or refueling in order to verify the results of the PROTEUS-NODAL against the OpenMC results. The fuel salt is composed of LiF-$^{233}UF_4$-ThF_4 in a cylindrical core of 1.0 m^3 volume with reflective boundary conditions and 800 MW power. For this problem, burnup dependent cross sections were generated with OpenMC code for the same core configuration.

In this problem, the fuel salt was depleted for 350 days, and the eigenvalue and important nuclide concentrations were compared at each depletion point. Figure 6.2 shows the eigenvalue results of the PROTEUS-NODAL and OpenMC codes. The difference in eigenvalue results is within 40 pcm for this simple test problem, and the eigenvalue decreased significantly during the first 50 days, and it decreases at a slower rate after that. This can be explained by breeding the ^{233}U isotope from ^{232}Th isotope according to the following process

$$ {}^{232}_{90}Th + {}^{1}_{0}n \longrightarrow {}^{233}_{90}Th \xrightarrow[21.83\ \text{min}]{\beta^-} {}^{233}_{91}Pa \xrightarrow[26.97\,day]{\beta^-} {}^{233}_{92}U \ . \tag{6.30} $$

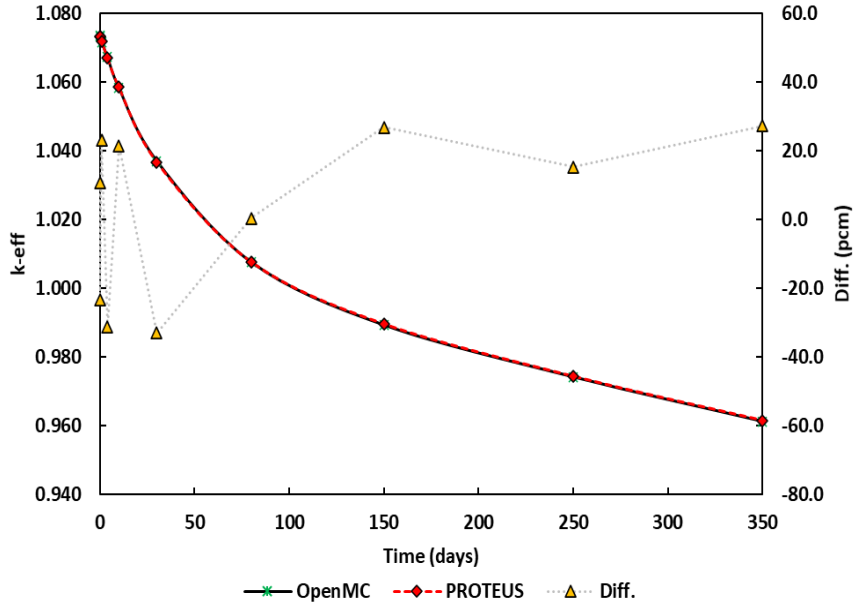


Figure 6.2. Eigenvalue change with time for stationary fuel test problem.

The atomic concentrations calculated by PROTEUS-NODAL and OpenMC codes of the ^{232}Th, ^{233}Pa, and ^{233}U isotopes are shown in Fig. 6.3. The atomic concentration of the ^{232}Th is decreased

linearly due to the large capture reaction of this isotope in the resonance region with a maximum relative difference from OpenMC of 0.012%. The atomic concentration of the ^{233}U isotope is reduced significantly at the first 50 days and starts to increase again due to the decay of the accumulated ^{233}Pa isotope. As shown in Eq. (6.30), the ^{232}Th produces the ^{233}Th isotope from neutron capture reaction, which decays into ^{233}Pa isotope with a half-life of 21.83 minutes. The ^{233}Pa decays into ^{233}U with a half-life of 26.97 days. The atomic concentration of the ^{233}Pa isotope reaches saturation level within 100 days. While the concentration of the ^{233}U continuous to build up. Compared to the OpenMC results, the maximum relative differences in the atomic concentrations calculated by PROTEUS-NODAL for the ^{233}Pa and ^{233}U isotopes are 0.17% and 0.07%, respectively. The results of this test problem show the validity of the PROTEUS-NODAL depletion solver for performing depletion analysis and the selection of the fuel type was to test the developed solver for a problem composition similar to the MSFR benchmark, as discussed in the following subsection.

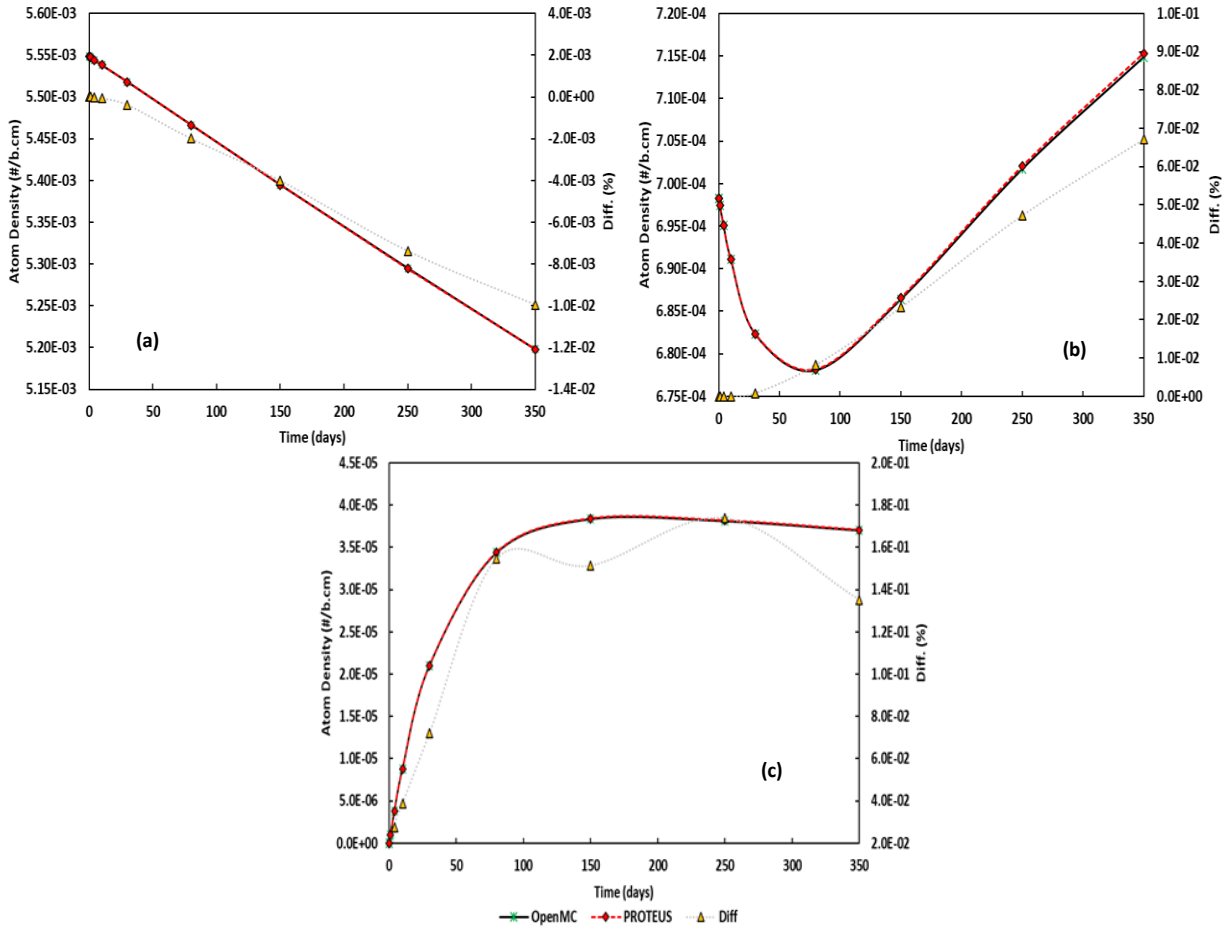


Figure 6.3. Atom densities change of (a) ^{232}Th (b) ^{233}U and (c) ^{233}Pa with time for stationary fuel test problem.

6.4.2. MSFR Fuel Cycle Analysis

The MSFR benchmark report [13] includes fuel cycle analysis of the ^{233}U started core and the TRU started core along with the steady state analysis. The fuel cycle analysis focuses on the evolution of the heavy metal and fission products in the fuel salt over 200 years of operation of the fuel salt. The description of the MSFR benchmark is provided in Chapter 3 and the design specifications are provided in Appendix C. The specifications for the fuel salt reprocessing are provided in this subsection.

The salt management combines a salt control unit, an online gaseous extraction system, and an offline lanthanide extraction component. Figure 6.4 shows a schematic diagram of the online fuel salt reprocessing system in MSR. In the online gaseous extraction system, Helium bubbles are injected into the core to remove all non-soluble fission products (noble metals and gaseous fission products) with a removal period of 30 s. In the offline salt reprocessing component fraction of salt is periodically withdrawn and reprocessed to extract the lanthanides and soluble fission products. The reprocessing rate is 40 *l*/day which corresponds to reprocessing period of 450 days of the whole core considering 100% offline extraction efficiency. Also, Pa is extracted from the core and blanket and stored until the ^{233}Pa decays to ^{233}U that will be injected back into the core with the fresh fuel continuously to maintain criticality. Table 6.1 summarizes the elements that are being removed from the fuel salt and their removal frequencies from the core and the blanket regions.

Table 6.1. Reprocessing Frequencies of Heavy Metal and Fission Products in MSFR.

Removal Group	Elements	Region	Removal Time
Non-Soluble Fission Products	H, He, N, O, Ne, Ar, Kr, Nb, Mo, Tc, Ru, Rh, Pd, Ag, Sb, Te, Xe, Rn	Core	30 sec
		Blanket	30 sec
Soluble Fission Products	Zn, Ga, Ge, As, Se, Br, Rb, Sr, Y, Zr, Cd, In, Sn, I, Cs, Ba, La, Ce, Pr, Nd, Pm, Sm, Eu, Gd, Tb, Dy, Ho, Er, Tm, Yb	Core	450 day
		Blanket	52.7 year
Heavy Metal	U, Pu, Np	Blanket	192.5 day

In PROTEUS-NODAL calculations for the MSFR fuel cycle analysis, the cross sections were prepared for 282 nuclides (58 actinides and 222 fission products) using OpenMC code and ENDF/B-VII.1 cross section data. For nuclides with no cross sections data, decay reactions were considered. A detailed depletion chain contains 1356 nuclides of which 93 actinides and 1157 fission products were considered in the MSFR depletion calculations. The calculations were

performed for ^{233}U and TRU cores to obtain the evolution of U, Pu, minor actinides, and fission products inventories as a function of the operation time. The feeding rate, the breeding ratio, and the breeding gain were calculated over 200 years of operation for the fuel salt in the core and the blanket regions with online reprocessing and feeding. At each time point, the feeding rate of the fissile material was searched in order to maintain criticality considering the flow of the fuel salt without thermal feedback.

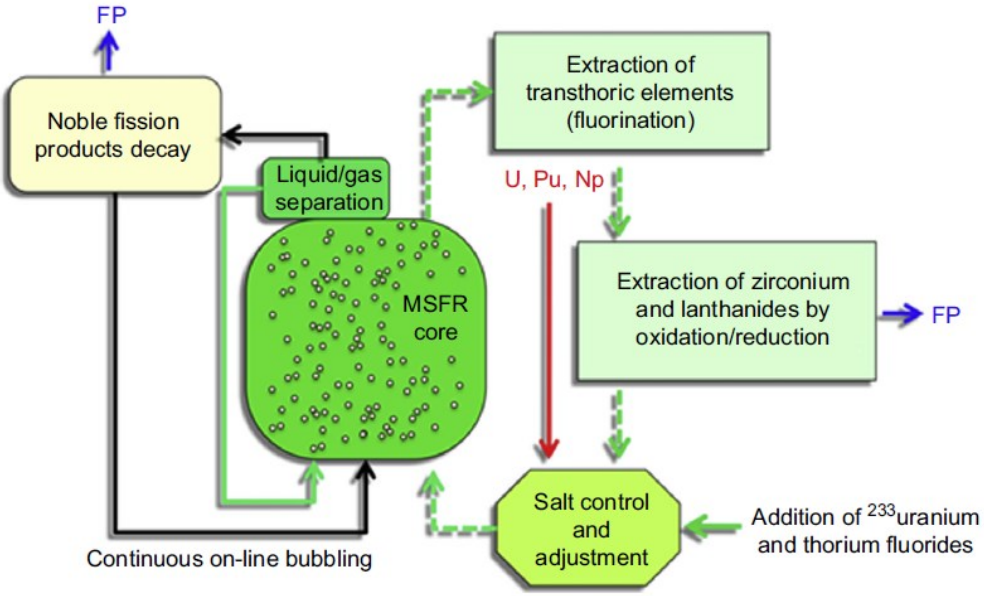


Figure 6.4. Reprocessing diagram of MSFR [3].

Figure 6.5 compares the evolution of the major heavy metal inventories in the MSFR for the ^{233}U and TRU started cores calculated by PROTEUS-NODAL with reference solution obtained from the MSFR design report. The results of the PROTEUS-NODAL of both MSFR cores are similar to the results obtained from the reference solution. Also, Fig 6.6 and Fig. 6.7 show the heavy meatal and fission products inventories for 200 years of operation as calculated by PROTEUS-NODAL and compared to the reference solution for ^{233}U and TRU started cores, respectively. The differences in the PROTEUS-NODAL and the reference results are due to differences in the depletion chains and neutron cross section libraries used in the calculations. After 100 of operation, the heavy metal of both MSFR options reaches its equilibrium concentration except for Cm and Am concentrations, while the fission products reach equilibrium concentration after 5 years of operation with continuous removal of the soluble and non-soluble fission products. Although the initial fuel salt compositions of the ^{233}U and TRU started MSFR are

different, they achieve similar fuel compositions after almost 150 years of operation with the same reprocessing scheme and the same feeding material.

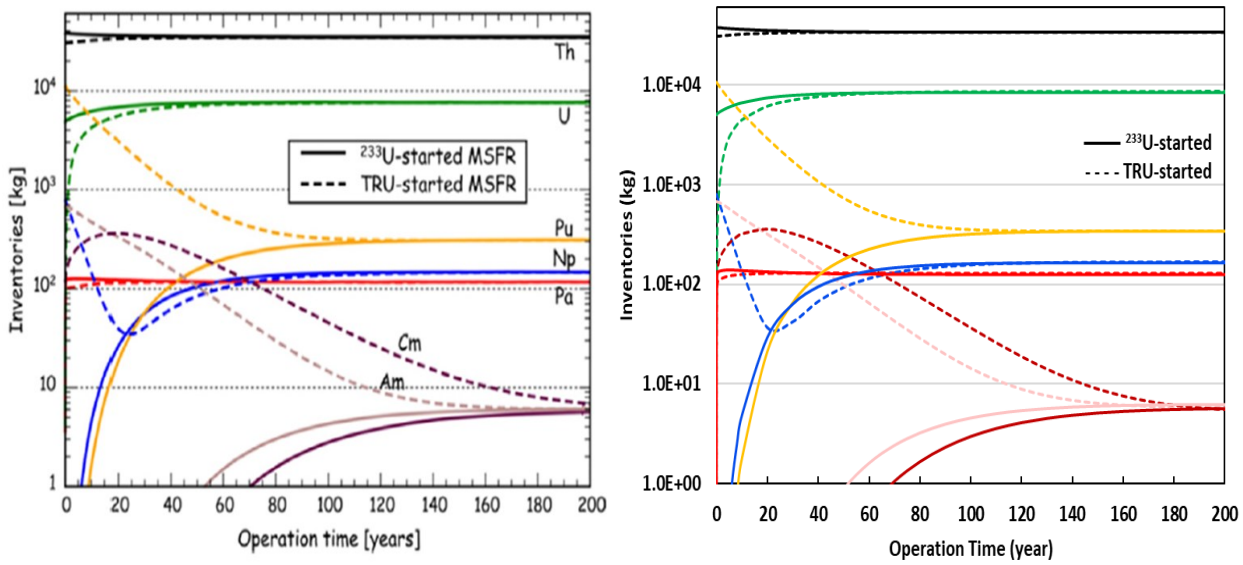


Figure 6.5. Time evolution of heavy element inventory for MSFR calculated by PROITEUS-NODAL (right) and reference [13] (left).

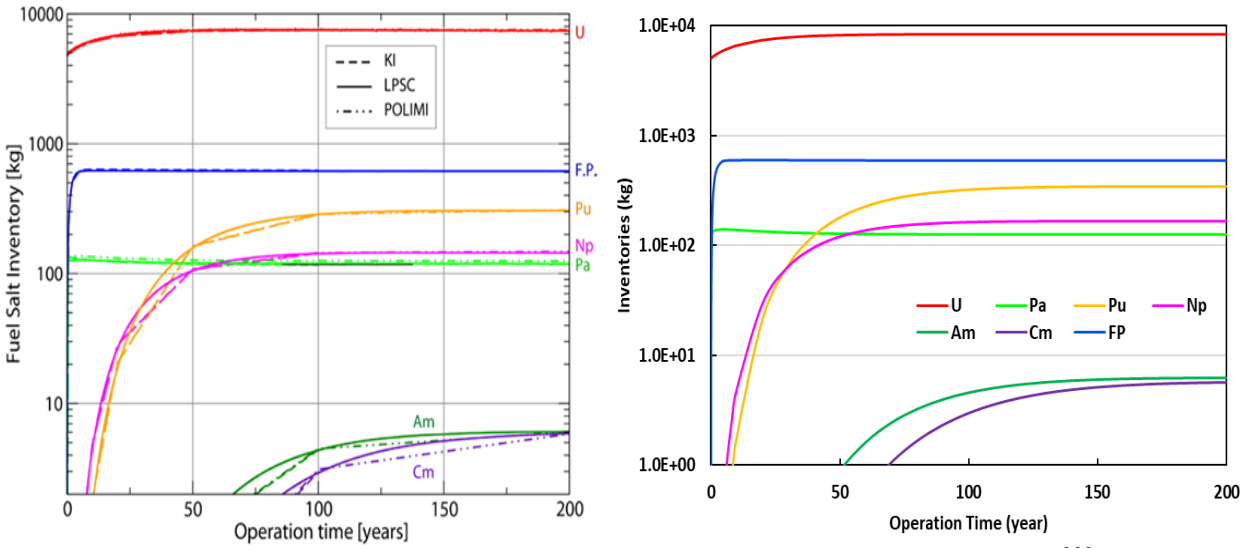


Figure 6.6. Time evolution of heavy element and fission products inventory for ^{233}U started MSFR calculated by PROITEUS-NODAL (right) and reference [13] (left).

The time evolution of the fission products in the ^{233}U started MSFR calculated by PROTEUS-NODAL and compared to other reference solutions is shown in Fig. 6.8. The PROTEUS-NODAL results agree very well with reference results and reach an equilibrium concentration of 600 kg after 5 years of operation. The discrepancies appear in the fission products concentrations because

of the differences in the employed simulation models, cross section data library, and depletion chain library used in the reference solution.

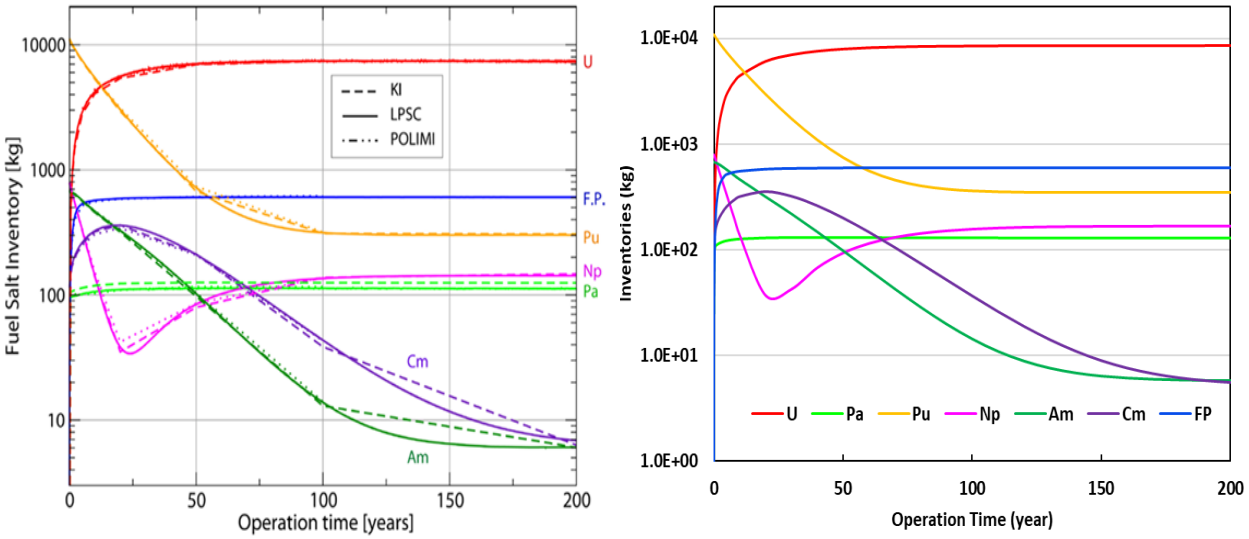


Figure 6.7. Time evolution of heavy element and fission products inventory for TRU started MSFR calculated by PROITEUS-NODAL (right) and reference [13] (left).

Figure 6.9 shows the feeding rate of the fissile and fertile components of the fuel salt for ^{233}U and TRU started fuels along with eigenvalue at each time point. For both fuel salt options, the feeding rate of the fissile component is around 70 kg/year in order to maintain criticality. The high feeding rate at the beginning of the operation is due to the low breeding of the ^{233}U of fresh fuel and the buildup of the fission products.

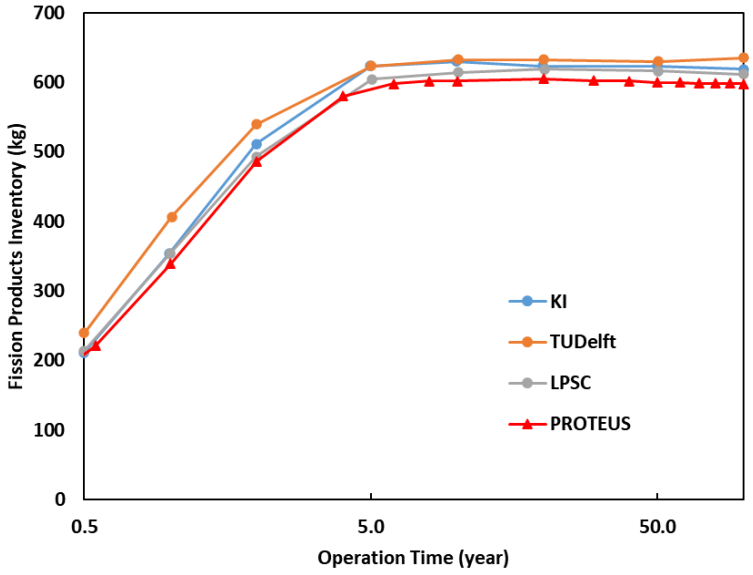


Figure 6.8. Mass inventory of fission products in ^{233}U started MSFR.

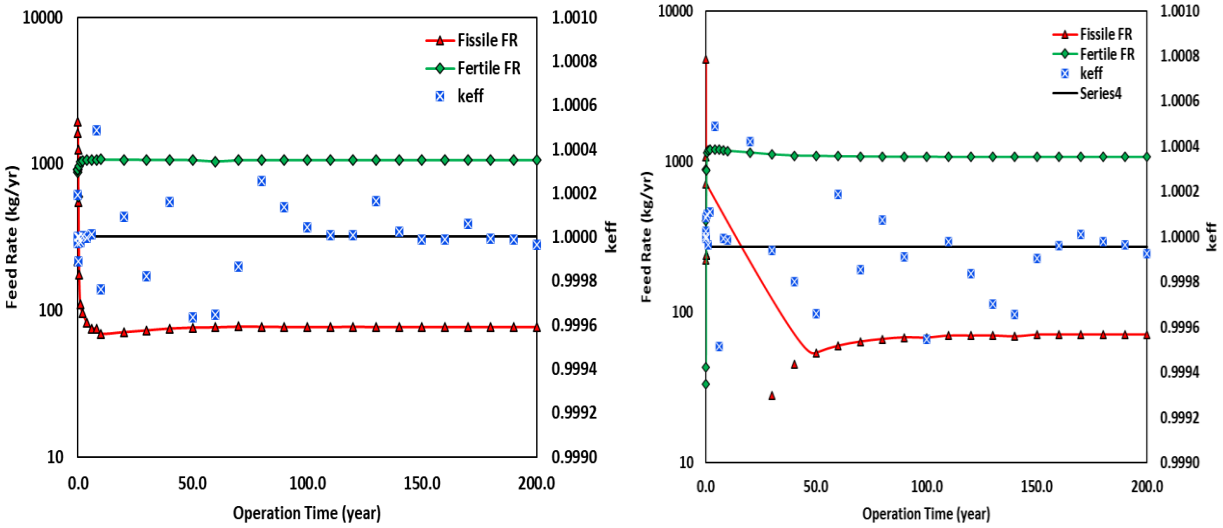


Figure 6.9. Fissile and fertile materials feeding rates in ^{233}U (left) and TRU (right) started MSFR calculated by PROTEUS-NODAL.

For further verification of the developed MSR depletion capabilities of the PROTEUS-NODAL code, the breeding ratio and the breeding gain of both MSFR core options are calculated and compared to the values reported in the MSFR design report. The breeding gain is defined in the MSFR design report as the ratio of the decay rate of the ^{233}Pa to the total neutron fission and capture (absorption) rates of the ^{233}U. Table 6.2 provides the calculated breeding ratio in comparison with other reference solutions. The calculated values with PROTEUS-NODAL are in good agreement with the values of other institutes using the same cross sections library.

Table 6.2. Calculated Breeding Ratio of Both MSFR Options.

	KI	LPSC	POLIMI	POLIMI	FAMOS	PROTEUS
XS	ENDF/B-VI	ENDF/B-VI	JEFF-3.1	ENDF/B-VII	ENDF/B-VII.1	ENDF/B-VII.1
233U	1.07	1.13	1.112	1.039	1.044	1.052
TRU	1.07	1.11	1.112	1.039	1.048	1.056

The breeding gain is defined as the total balance of the ^{233}U in the core and the blanket considering the amount extracted from the blanket and supplied into the core. Figure 6.10 shows the time evolution of the breeding gain for ^{233}U started and TRU started MSFR. The breeding gain of the ^{233}U started MSFR core reached equilibrium level after about 20 years of operation with a value close to 125 kg/year. While for the TRU started MSFR the breeding gain reaches the equilibrium level after 100 years of operations with a value of 140 kg/year.

In the TRU started MSFR, the initial breeding gain is significantly high because the produced ^{233}U is not used due to existed TRU fuel. As the TRU fuel is burned, the breeding gain is reduced significantly because the produced ^{233}U is used to replace the burned TRU fuel. The main sources of the discrepancies among the results of the PROTEUS-NODAL and other participating institutes are due to the differences in the employed simulation models, cross section data library, and depletion chain library used in the reference solution.

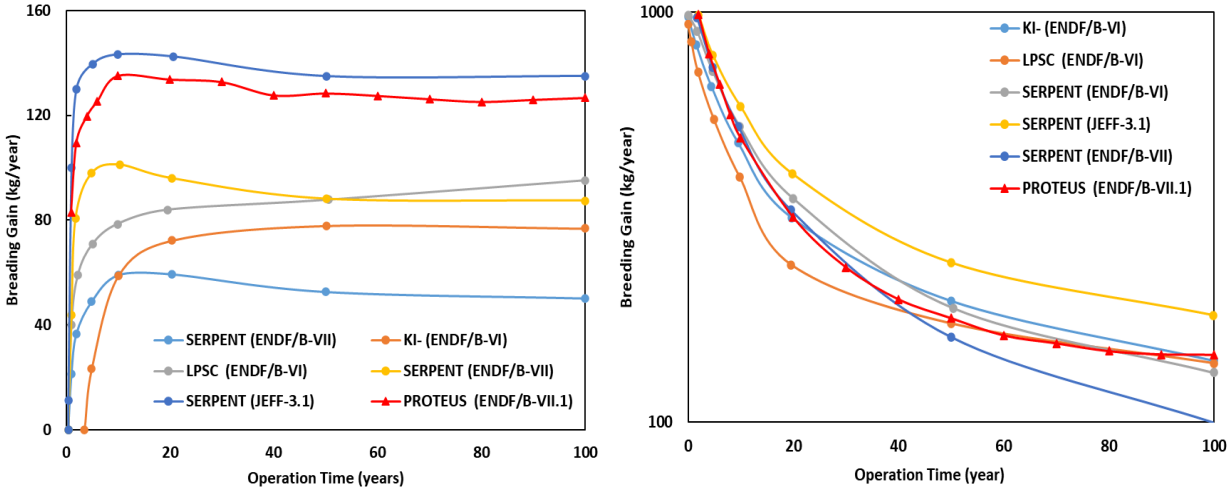


Figure 6.10. Time evolution of breeding gain of ^{233}U (left) and TRU (right) started MSFR.

In most fuel cycle analyses performed for the MSFR in the literature, the flow of the fuel in the core is not considered, instead, the fuel is assumed to be stationary, and the one-point depletion model is utilized as in several studies performed with Monte Carlo codes like Serpent and OpenMC. However, in order to achieve the same heavy metal time evolution, the fuel feeding rate and the breading ratio will be affected. As presented in Table 6.2, the differences in the breeding ratio are also related to several factors including the effects of the fuel flow, decay heat, and thermal feedback. It is worth mentioning that for the MSFR case, the time spent by the fuel in and out the core is 2.0 s each, so the use of the two points or one-point model makes almost no difference. However, for other reactor designs with different fuel residence times in the core and outer loop, both models will provide different results.

To study the effect of the fuel flow, thermal feedback, and the decay heat on the fissile feeding rate and the breeding ratio, several calculations were performed with and without considering fuel flow, thermal feedback, and the decay heat for both MSFR cores. Table 6.3 summarizes the fissile feeding rate and the breeding ratio for several cases. For both MSFR cores, the fissile feeding rate

is slightly underestimated, and the breeding ratio is overestimated because of ignoring the effect of the fuel flow and delayed neutron losses.

Table 6.3. Effect of TH Feedback and Decay Heat on Fissile Feeding Rate and Breeding Ratio.

Case	MSFR Core		Flowing Fuel		Decay Heat		TH Feedback		Fissile Feed Rate (kg/year)	Breeding Ratio
	^{233}U	TRU	w/o	w/	w/o	w/	w/o	w/		
1	x		x		x		x		71.16	1.055
2	x			x	x		x		75.46	1.052
3	x			x		x	x		70.93	1.053
4	x			x	x			x	81.87	1.047
5	x			x		x		x	76.99	1.049
6		x	x		x		x		63.34	1.061
7		x		x	x		x		68.57	1.056
8		x		x		x	x		65.57	1.058
9		x		x	x			x	71.26	1.055
10		x		x		x		x	65.33	1.058

For ^{233}U started core, considering the decay heat (~ 4.0% of the total power) in the neutron flux normalization results in reducing the fissile feeding rate and increasing the breading ratio due to the reduction of the consumption rate of the fissile material since the decay power contributes to a non-negligible amount of the total produced power. More details about the decay heat calculations are provided in the following Chapter. Also, considering the thermal feedback resulted in a reduction of the breeding ratio from 1.052 to 1.047 and increasing of the fissile material feeding rate by about 5.0 kg/year. This is due to the effect of the fuel temperature and density distributions on the feedback model and updating the cross sections. Similar behavior was observed for the TRU started core. Figure 6.11 shows the change of the breeding ratio with time over for several cases for the ^{233}U started and TRU started cores.

6.5. Summary

Fuel cycle analysis capabilities of MSRs have been implemented in the PROTEUS-NODAL code for depletion and fuel cycle analysis of the liquid fuel MSRs. The new depletion capability utilizes the CRAM solver of the open-source toolkit EXPOKIT to solve the matrix exponential, and the predictor-corrector coupling scheme was utilized to couple the steady state solver and the depletion solver. The developed depletion capability has online reprocessing and salt control capabilities to maintain the heavy metal fraction in the fuel salt at a fixed level, and it allows online fuel salt refueling by searching for the feed rates of the fissile and fertile components of the fuel

salt in order to maintain criticality. The implemented capabilities were first verified against Monte Carlo code OpenMC for a stationary fuel problem, then, the fuel cycle analysis of the MSFR benchmark problem was performed with PROTEUS-NODAL and verified against the reference solutions obtained from the design report of the MSFR benchmark. Also, the effects of the drift of the fuel salt, the decay heat, and coupling with the thermal hydraulics on the fissile feeding rate and the breeding ratio were investigated. The results show that these parameters have a significant impact on the feeding rate of the fissile isotopes. The implemented depletion model is used to calculate the decay heat as a function of time after reactor shutdown is discussed in the following Chapter. Also, an adaptive time-stepping solution algorithm was implanted to predict the depletion time step size following a similar procedure utilized for the transient solver as provided by Appendix G. The solution algorithm helps to perform the depletion calculations based on a specified accuracy level without providing prescheduled time steps.

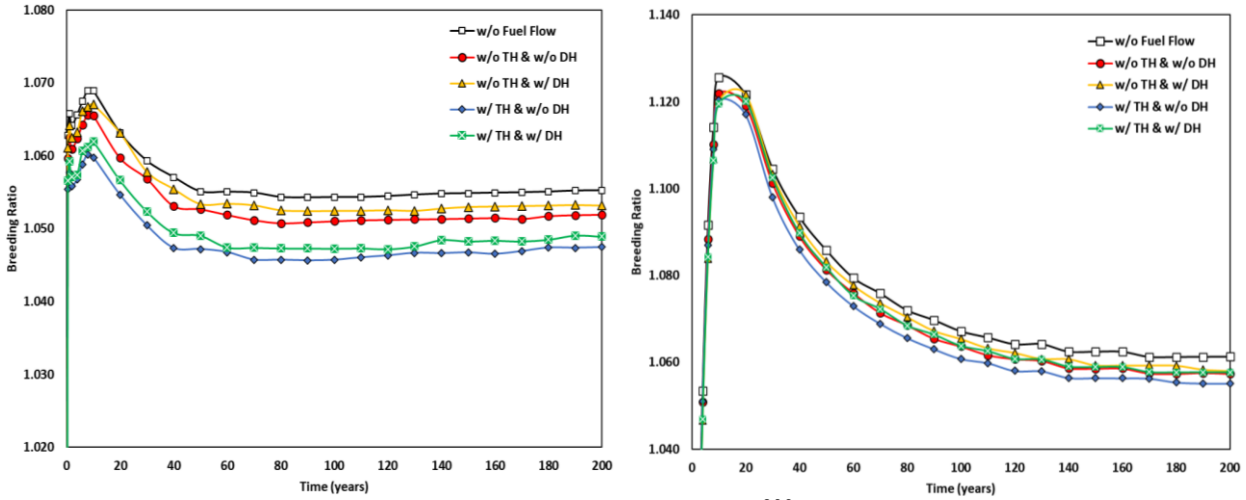


Figure 6.11. Change of the breeding ratio with time for ^{233}U (left) and TRU (right) started MSFR.

Chapter 7. Improved Feedback Model for MSRs

In real operation of MSRs, it is important to consider the heat generated by the decay process of the fission products of liquid fuel or the decay heat in neutronics analyses since it is partially released in the reactor core region and the primary loop, since it contributes to a non-negligible amount of the total generated power. Because of the online reprocessing of the fuel salt, certain fission products are removed from the fuel salt, so they will not contribute to the total decay heat released in the system. After a normal reactor shutdown, most of the heat generated will be dispersed by continued circulation of the fuel and coolant salts. In the case of draining the fuel salt to a drain tank, the heat generated in the salt would be dissipated through the primary drain tank cooling system. However, it is important to calculate the decay heat released by the fission products that exist in the fuel salt for accurate calculations of the fuel burnup and the fissile feeding rate and a more realistic transient analysis.

Also, during online refueling in MSRs, fissile and fertile components are injected into the fuel salt in the outer loop while the reactor is operational. This will result in reactivity perturbations due to redistribution of the fuel salts nuclides in the core. An over-fueling of either of the components will induce reactivity insertion transient or an over-fueling accident, and the reactor power and fuel salt temperature will change accordingly. In order to simulate this unique transient scenario, the redistribution of the fuel salt nuclides in the core region should be considered in the thermal feedback for updating the cross sections. In this way, a more realistic thermal feedback model of MSRs can be utilized for multiphysics analysis.

The feedback model of MSRs implemented in PROTEUS-NODAL has two components, one is related to thermal changes of the fuel salt on the cross sections due to fuel temperature change and expansion of the fuel salt or density change. The other component is the fuel salt velocity, which affects the delayed neutron and decay heat precursors distributions in the core. Also, during online refueling, the distribution of the nuclides in the core region is governed by the fuel velocity, and the redistribution of the fuel salt nuclides in the core region should be included in the thermal feedback model. In this Chapter, further improvements are made for the feedback model of MSRs

considering the nuclide distribution and the decay heat as discussed in the following sections. Figure 7.1 shows the coupling scheme and the exchanged parameters between neutronics and thermal hydraulics solvers, and the thermal feedback model.

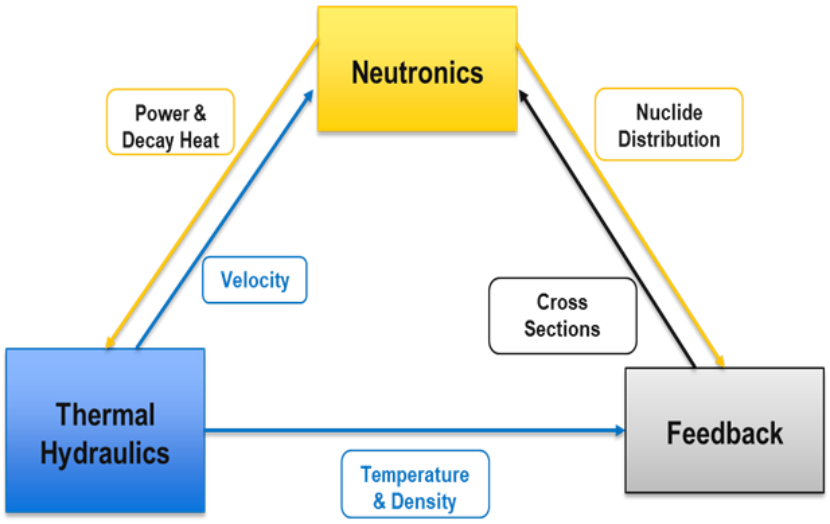


Figure 7.1. Coupling scheme and feedback model for MSRs.

7.1. Decay Heat Model

In MSRs, part of decay heat is released outside the core, especially in the heat exchanger, and it should be considered in the power calculation. A decay heat model has been implemented into the PROTEUS-NODAL code, where the fission products are grouped into a few decay heat precursor groups K_D, and for each group, the following decay heat equation is solved

$$\frac{\partial}{\partial t} h_k(\vec{r},t) + \nabla \cdot \left[\vec{u}(\vec{r},t) h_k(\vec{r},t)\right] + \lambda_k h_k(\vec{r},t) = f_k \sum_{g=1}^{G} \kappa_f \Sigma_{fg}(\vec{r},t) \phi_g(\vec{r},t), \quad k = 1,2,....,K_D, \tag{7.1}$$

where f_k, λ_k, and h_k denote the decay heat fraction, the decay constant, and the product of the precursor concentration and energy release of decay heat group k, respectively, and κ_f is the recoverable energy per fission. Eq. (7.1) is solved using the FDM method in a similar way as the delayed neutron precursor equation. In the outer loop, there is no heat generated from fission, thus the decay heat in the outer loop and heat exchanger is calculated by solving Eq. (7.1) with zero volumetric heat source of the right-hand side of the equation. The total, fission, and decay heat sources in the core are calculated as

$$Q_{tot}(\vec{r},t) = Q_{fiss}(\vec{r},t) + Q_{decay}(\vec{r},t), \tag{7.2a}$$

$$Q_{fiss}(\vec{r},t)=\left[1-\sum_{k=1}^{K_D} f_k\right]\sum_{g=1}^{G}\kappa_f\Sigma_{fg}(\vec{r},t)\phi_g(\vec{r},t)\,, \tag{7.2b}$$

$$Q_{decay}(\vec{r},t)=\sum_{k=1}^{K_D}\lambda_k h_k(\vec{r},t)\,. \tag{7.2c}$$

In order to calculate the distribution and the amount of the heat generated by the decay of the fission products, the decay heat precursors need to be grouped with unique decay constants and fractions in a similar way as the delayed neutron precursors concentration. The delayed heat groups can be obtained by fitting the delayed energy release as a function of emission time. For LWRs, the 2014 ANS decay heat standard includes the time-dependent decay energy as exponential fit from the direct fission of four important fissile isotopes ^{235}U, ^{238}U, ^{239}Pu, and ^{241}Pu. The decay heat power is represented as a summation of 23 exponential terms [132]. In a similar way, the decay heat for MSRs can be represented with exponential fitting as a summation of K exponential terms as

$$Q_{decay}\left(t\right)=Q_{d,0}\sum_{k=1}^{K} f_k\exp\left(-\lambda_k t\right), \tag{7.3}$$

where $Q_{decay}\left(t\right)$ is the decay heat generated at time t, $Q_{d,0}$ is the decay heat generated during a steady state operation before shutdown the reactor, f_k and λ_k are the decay heat fraction and decay constant for group k, respectively. The decay heat fraction and decay constant for each group can be determined from exponential fitting of the decay heat curve after performing depletion calculations at full power for a sufficient amount of time, so that the decay heat reaches a saturation or equilibrium level. Then, shutdown conditions are imposed, and the decay heat is calculated. In PROTEUS-NODAL depletion calculations, all nuclides in the depletion chain are being tracked which allows calculation of the decay heat explicitly as expressed in Eq. (6.29).

As an example, the decay heat calculation capability of the PROTEUS-NODAL code was tested using the MSFR benchmark for stationary fuel case and the results were compared to Serpent code results considering the same problem. Figure. 7.2 shows the decay heat as a fraction of the total power for 350.0 days of full-power operation, and 350.0 after the reactor was shut down. The results of PROTEUS-NODAL agree with Serpent results, and the decay heat reaches about 5.58% of the total generated power in the reactor core.

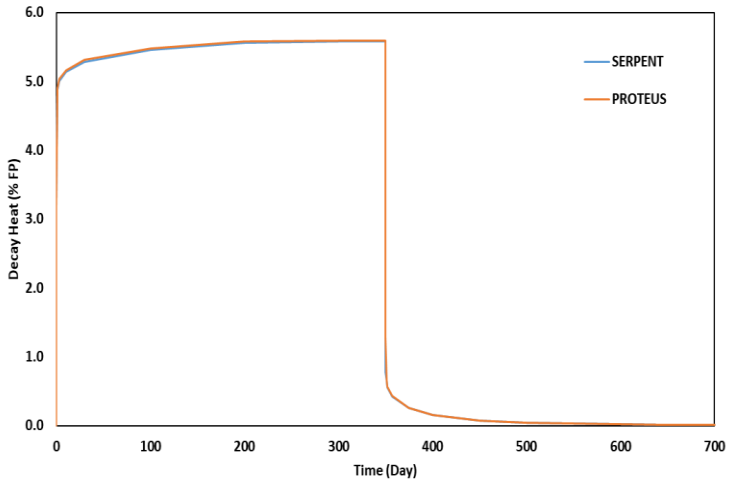


Figure 7.2. Decay heat power calculated by Serpent and PROTEUS-NODAL for MSFR.

In the real operation of MSR, the removal of fission products during reprocessing of the fuel salt must be considered. In Table 6.1, the fission products for the MSFR case were classified soluble as non-soluble fission products with removal frequency of 450 days and 30 s, respectively. Figure. 7.3 shows the decay heat generated by the soluble and non-soluble fission products as a function of time. The soluble fission products account for 4.38% of the total power, while the soluble fission products account for 1.20% of the total power. The major element contribution to the decay heat is for Rb (0.597%), Y (0.550%), La (0.463%), Sr (0.384%), Cs (0.359%), Nb (0.342%), and Kr (0.315%). For the MSFR case, the thermal power of the reactor is 3000 MW, and the non-soluble fission products will contribute to 36 MW which is a huge amount of power and must be considered carefully in the handling of the fission products.

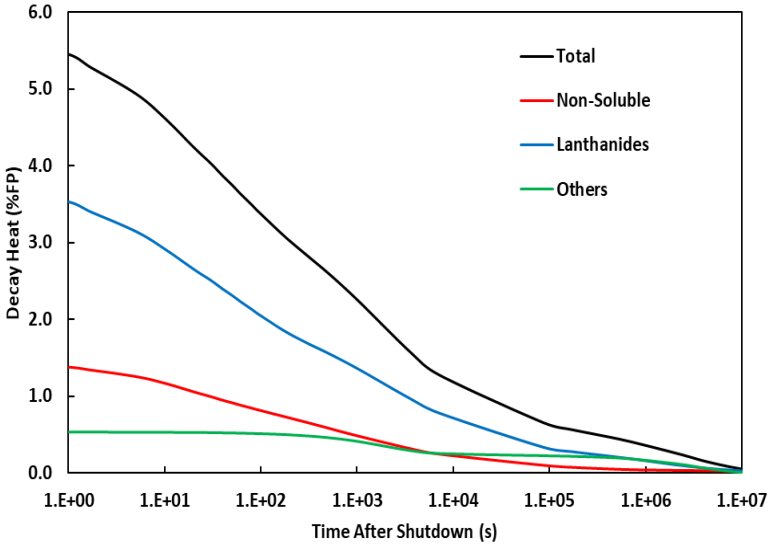


Figure 7.3. Decay heat generated by soluble as non-soluble fission products after shutdown.

If the non-soluble fission products are completely removed through reprocessing, only the soluble fission products will account for the total decay heat. So, they must be eliminated from the decay heat calculations as they are continuously removed from the fuel salt. The total decay generated by the fission products considering different removal frequencies of the non-soluble fission products is provided in Fig. 7.4. As the non-soluble fission products are continuously removed from the fuel salt, their contribution to the total generated decay heat decreases significantly. In the design of the MSFR, the non-soluble fission products are removed from the fuel salt every 30.0 s, and the decay heat will contribute to about 4.05% of the total reactor power compared to 5.58% for the case without removal of the fission products.

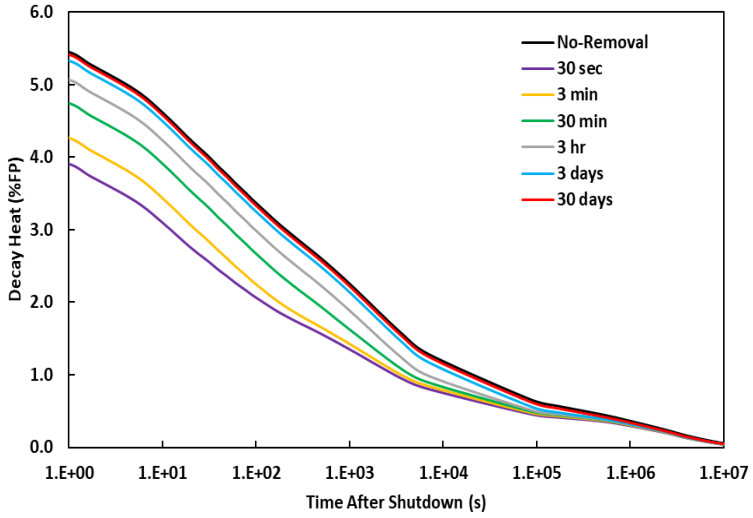


Figure 7.4. Total decay heat generated after shutdown considering different removal times of non-soluble fission products.

To obtain the decay heat distribution in the core, the decay heat precursor equation must be solved as in Eq. (7.1). However, the solution of the decay heat precursor equation can be obtained in a similar way to the delayed neutron precursor equation using the FDM approach, yet the decay constants and the decay heat fractions for each group must be determined. For the MSFR case, the decay constants and the decay heat fractions were determined by exponential fitting of the decay heat curve after shutdown, as in shown Fig. 7.5, to represent the decay heat as a summation of 4 exponential terms, as described in Eq. (7.3). Table 7.1 provides the decay heat parameters of the ^{233}U started MSFR.

Table 7.1. Decay Heat Parameters of ^{233}U started MSFR.

Group	1	2	3	4
f (%)	1.521	0.675	0.980	0.876
λ (s^{-1})	1.312E-04	3.320E-03	2.546E-02	1.740E-01

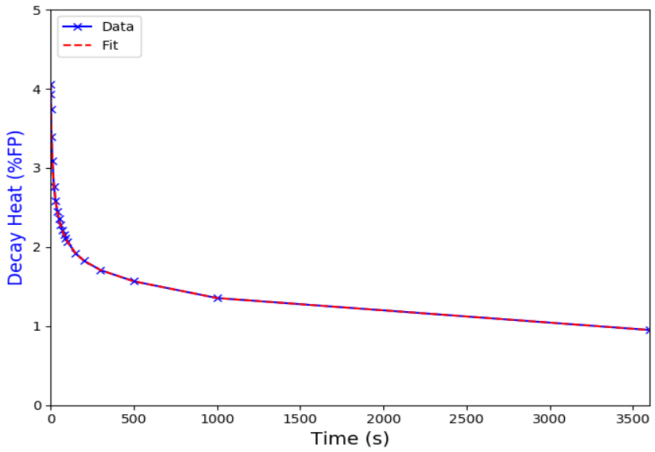


Figure 7.5. Fitted decay heat curve after shutdown of MSFR case.

7.1.1. Steady State

The decay heat model of PROTEUS-NODAL was tested for steady state calculations without thermal feedback to determine the contribution of the decay heat to the total power. Figure 7.6 shows the fraction of decay heat released as a function of the transit time in the core for different out-of-core transit times. The decay heat released in the core region increases as the fuel spends more time in the core region. It approaches the stationary fuel value as the in-core transit time increases. For the reference transit times of 2.0 s in the core region and 2.0 s out of the core region, the decay heat fraction in the core region is 2.1% of the total power, which is about half of the stationary fuel value of 4.05%. If the fuel exiting the core region re-enters the core region immediately, the decay heat fraction would be constant and equal to the stationary fuel value since all the fuel would decay in the core region. Figure 7.7 shows the total and decay power distributions in the core for the reference operating condition. As expected, the total power distribution is symmetrical around the core mid-plane while the decay heat is shifted upward.

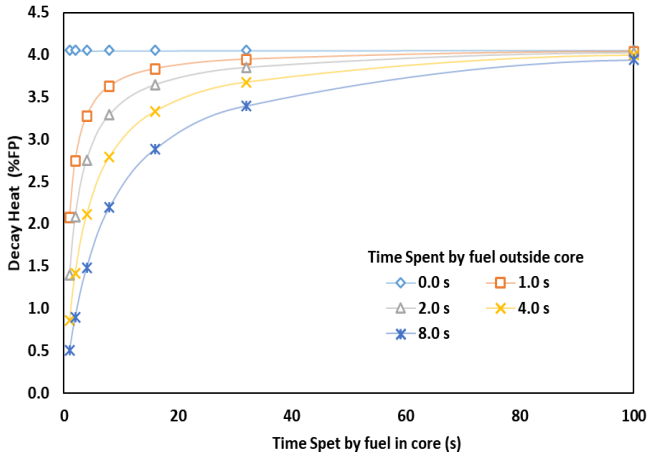


Figure 7.6. Fraction of decay heat release in core region of MSFR.

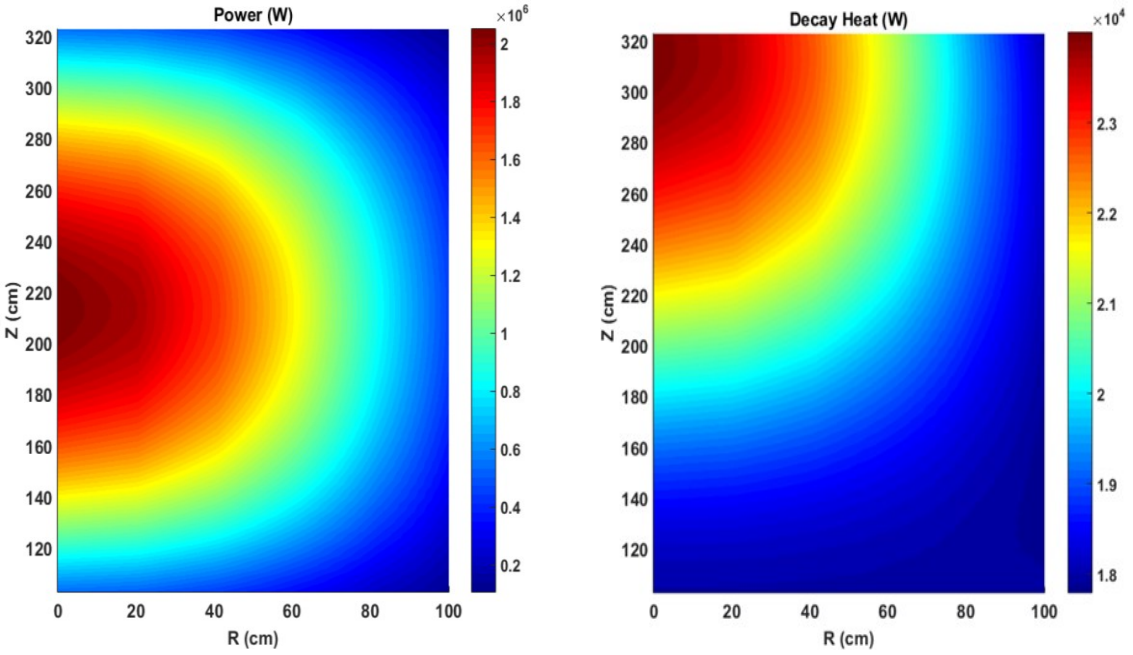


Figure 7.7. Total (left) and decay (right) power distributions in core region of MSFR.

7.1.2. Transient

From a safety analysis point of view, it is important to calculate the fuel salt temperature accurately to make sure that the fuel salt temperature will not exceed the design limits during a transient scenario. The increased fuel salt temperature will result in reducing the reactor power due to the negative feedback, as in transients with loss of cooling capability or loss of fuel salt flow like ULOHS and ULOF scenarios. The decay heat of the fission products in the fuel salt will lead to a further increase of the fuel salt temperature since it contributes to about 4% of the total power, and it is produced over a longer time period. In this section, the ULOHS and ULOF transients of MSFR are analyzed considering the decay heat in a similar way as discussed in Chapter 3 for MSFR benchmark analysis. Figure 7.8 shows the power evolution and the core average temperature during ULOHS transient with and without considering the decay heat in the MSFR benchmark. As the fuel temperature starts increasing due to cooling capability loss, the reactor power starts decreasing significantly. At the beginning of the transient, the case of the power with decay heat decreases at a higher rate due to the higher fuel average temperature. After 10 s, the power became higher than the case without decay heat because the undecayed fission products produce more heat, and the decay heat exceeds the prompt power after 40 s from the start of the transient, and it reaches 80% of the generated power after 100 s. Also, the average core temperature increases significantly compared to the case without decay heat, and it didn't reach saturation in

100 s of the transient with about 100 K higher, and it reaches 1170 K at the end of the transient. This transient shows the importance of the decay heat during ULOHS and how it affects the power evolution and the average core temperature increase.

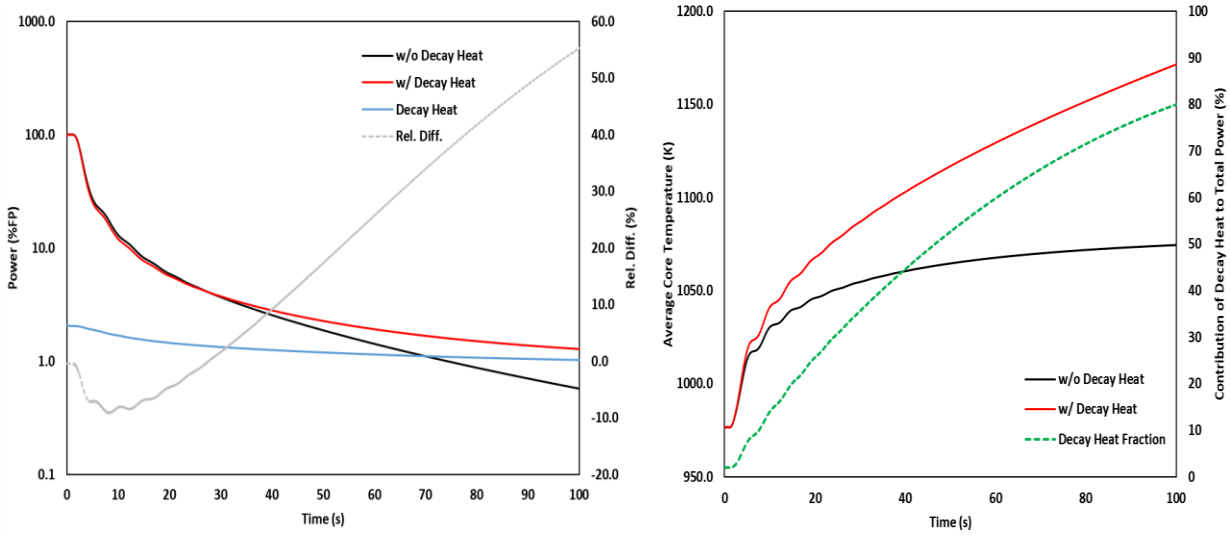


Figure 7.8. Power (left) and average core temperature (right) evolutions during ULOHS considering decay heat in MSFR.

The power evolution and core average temperature during the ULOF transient of the MSFR benchmark are shown in Fig. 7.9. As the fuel flow rate is reduced, positive reactivity is introduced due to the decreased loss of delayed neutrons out of the core. At the same time, the decreased fuel flow rate increases the fuel temperature in the core region, which introduces negative reactivity feedback. This negative temperature feedback is larger, and the reactor power starts decreasing. For the case with decay heat, the power decreases at a higher rate at the beginning due to higher fuel temperature. After 30 s, the generated power with decay heat becomes higher than the case without decay heat as the contribution of the decay heat to the total power increase and reaches 35% of the total generated power after 100 s of the transient. Also, the core average temperature with decay heat case is 50 K higher than the case without decay heat, and it reaches about 1250 K after 100 s of the transient. This transient shows that the decay heat is less significant in ULOF than in ULOHS transient. However, the core average temperature reaches a higher value, and the contribution of the decay heat is still important.

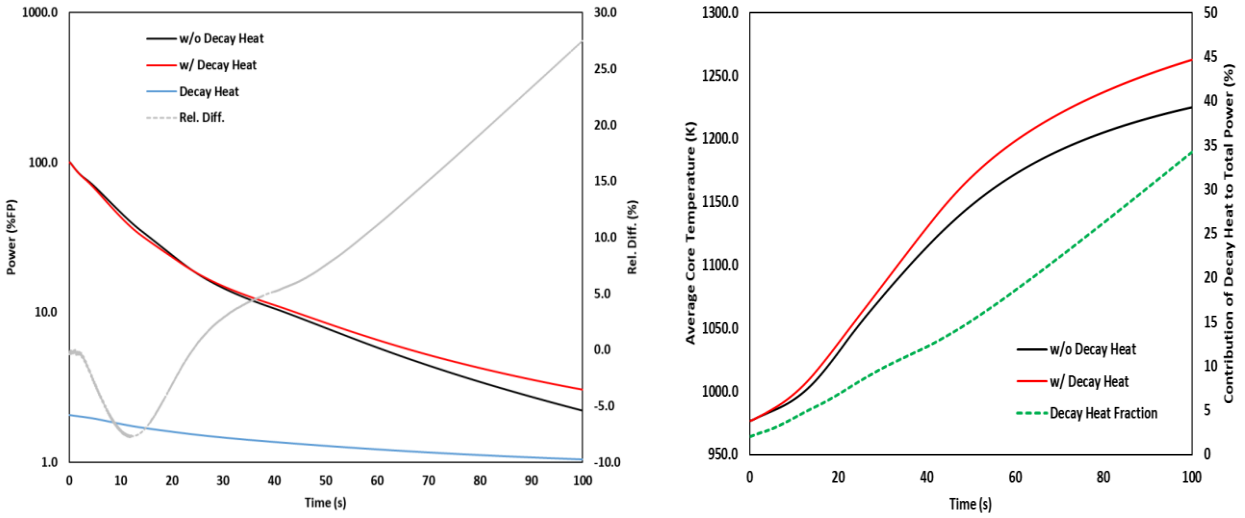


Figure 7.9. Power (left) and average core temperature (right) evolutions during ULOF considering decay heat in MSFR.

Also, the UTOP transient of the MSFR benchmark with 50 pcm reactivity insertion was analyzed considering the decay heat, as shown in Fig. 7.10. For the case with decay heat, the peak power is smaller than the peak power without considering the decay heat with a relative difference of 1.5% because of the lagged decay heat. The lower peak power resulted in a slightly smaller core average temperature. After 1.0 s in the transient, the power agrees for both cases with a relative difference within 0.5%.

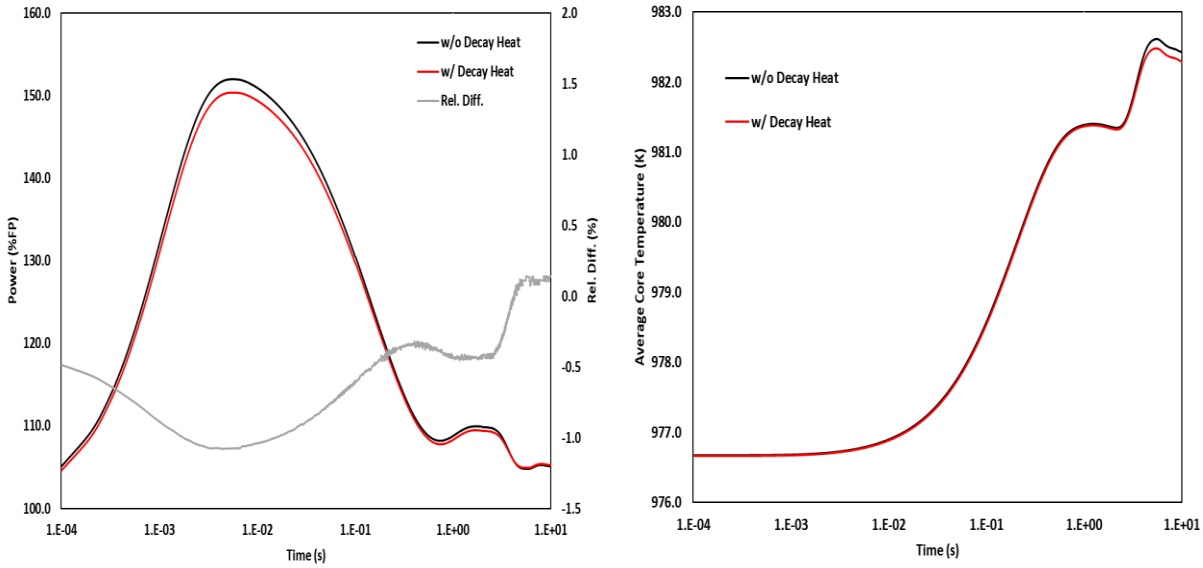


Figure 7.10. Power (left) and average core temperature (right) evolutions during UTOP with 50 pcm reactivity insertion considering decay heat in MSFR.

7.2. Nuclide Drift Model

Online fuel reprocessing is a unique feature of MSRs with flowing fuel, where the fissile and fertile components of the fuel salt are being added, and the fission products are continuously being removed from the fuel salt to maintain criticality and monitor the heavy metal concentration. The addition of the fissile or fertile nuclides to the fuel salt in an operating reactor will results in reactivity perturbation due to the non-uniform distribution of the salt nuclides. These reactivity perturbations depend on the amount and the rate at which these nuclides are being added. Such kind of perturbations is very important for the safety analysis and operation of the reactor. Also, over-fueling accident scenario or unprotected fuel salt over-fueling (UFSOF) need to be analyzed to study the change in the reactor power during the transient.

In order to simulate the over-fueling accident scenario, the distributions of the fuel nuclides in the core need to be considered in the feedback model to account for their drift and redistribution in the core. The following nuclide drift model has been added to the feedback model as

$$\frac{\partial}{\partial t} N_i(\vec{r},t) + \nabla \cdot [\vec{u}(\vec{r},t) N_i(\vec{r},t)] = S_i(\vec{r},t), \quad i = 1,2,....,N_{iso}, \tag{7.4}$$

where N_i is the atomic concentration of nuclide i, S_i is the source term that accounts for the addition of nuclide i. Eq. (7.4) is derived from the nuclide depletion equation of flowing fuel Eq. (6.9), however, the nuclides removals due to decay and burnup were omitted since they occur at a longer time span than the transient duration.

The feedback model of MSRs implemented in PROTEUS-NODAL has two components, one is related to thermal changes of the fuel salt and affects the cross sections due to fuel temperature change and expansion of the fuel salt or density change. The other component is the fuel salt velocity field, which affects the delayed neutron and decay heat precursors distribution in the core. Eq. (7.4) will add an extra component to the feedback model of PROTEUS-NODAL that is related to the fuel salt velocity field, which will lead to changes in the fuel salt nuclide concentration, as a result, a change in the macroscopic cross sections of the fuel salt.

7.2.1. Fissile Over-Fueling Transient

An over-fueling accident is analyzed for ^{233}U started MSFR by injecting fissile nuclide (^{233}U) or fertile nuclide (^{232}Th) into the core while the reactor is at normal operating conditions. Figure 7.11 shows the power and core average temperature increase following the addition of 5.0 kg of ^{233}U to the fuel salt in a period of 1.0 s. The reactor power starts increasing due to the increased

fission induced by the added ^{233}U as the fuel is flowing into the core. The reactor power increases to about 140.0% of the full power in about 1.0 s, and it starts decreasing as the fuel salt temperature increases in the core region. The core average temperature increases by about 12 °C, which will lead to a decrease in the reactor power beyond the steady state power level. As the unheated fuel salt and the portion of the fuel salt with added ^{233}U starts flowing back to the core, the reactor power starts increasing again and reaches a lower peak of about 120.0% of the reactor full power. The fluctuations in the power and temperature continue to decrease as the fuel salt is uniformly mixed in the system and stabilizes at a higher power level than the steady state power level within 50 s. The distribution of the ^{233}U nuclide in the reactor core region during the first 10.0 s is provided in Fig. 7.12. At the beginning of the transient, the distribution follows the fuel salt density distribution with a higher concentration near the core inlet due to a lower temperature and higher salt density, while at the top of the core, lower concentration of the ^{233}U is due to the higher temperature and lower density of the fuel salt. As the added ^{233}U starts flowing in the core, the higher nuclide concentration region is shifted upward until the fuel salt is uniformly mixed, then the ^{233}U nuclide distribution returns to a distribution similar to the initial state case as the power stabilizes at a new asymptotic level.

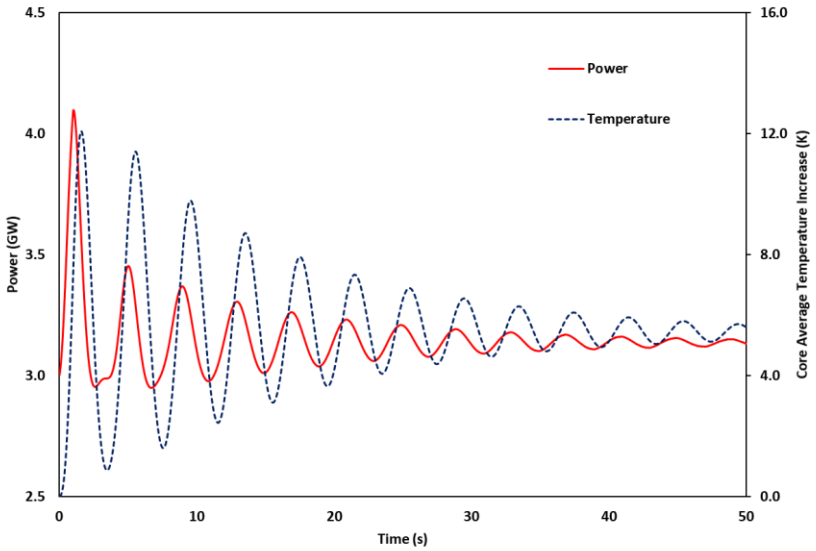


Figure 7.11. Time evolution of power and temperature after refueling 5.0 kg of ^{233}U in 1.0 s.

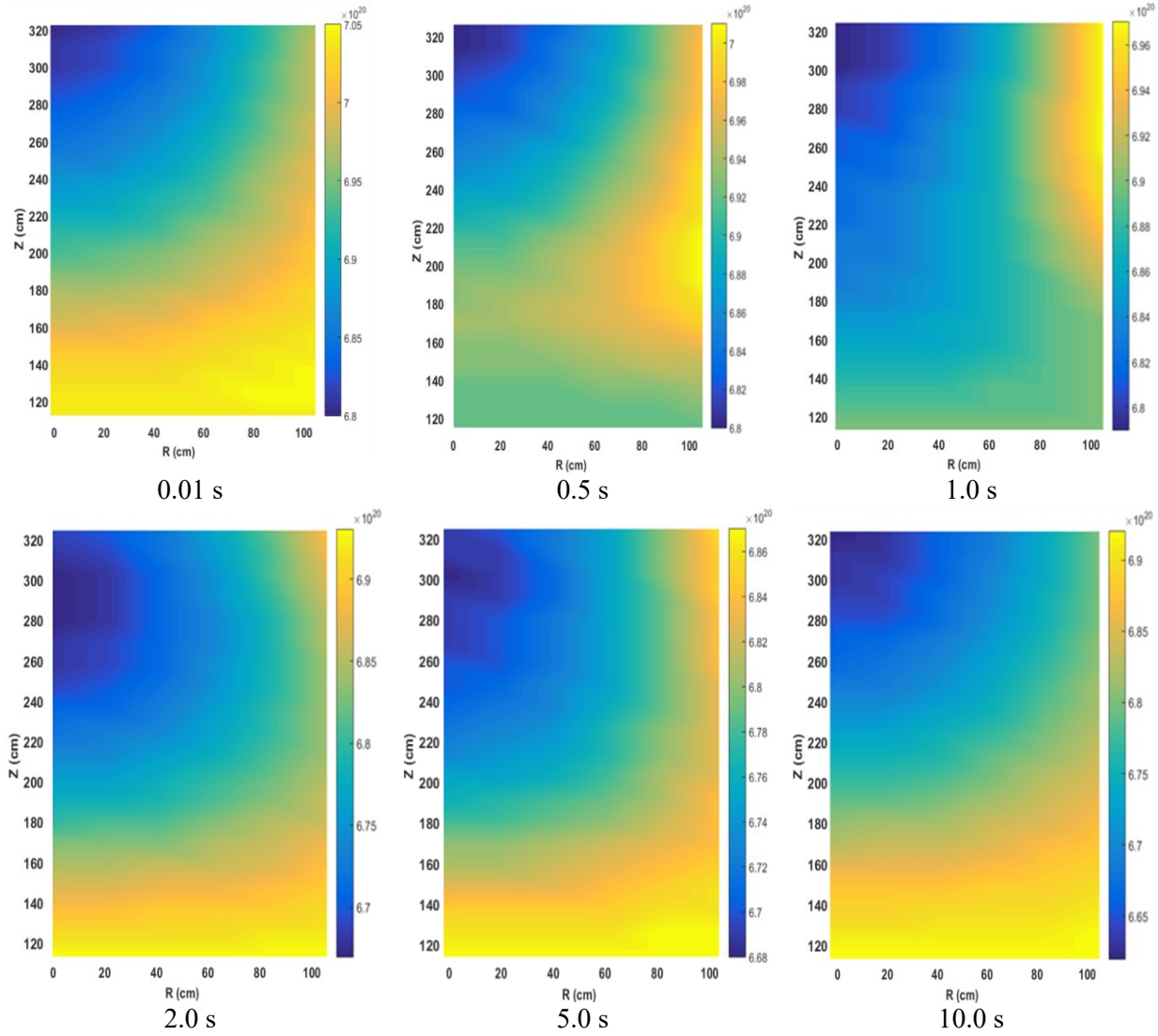


Figure 7.12. Distribution of ^{233}U nuclide in MSFR core after over-fueling transient.

The peak power magnitude and the frequency of the fluctuations depend on the amount of the added fuel salt and the rate at which the fuel salt is being added. To examine the effect of the added amount of the fuel salt on the magnitude of the peak power, three different amounts of the added ^{233}U during 1.0 s were simulated. Figure. 7.13 shows the power evolution after injecting 5.0, 10.0, and 15.0 kg of ^{233}U nuclide into the reactor core during 1.0 s. The frequency of the power fluctuations is similar since it depends on the circulation time of the fuel salt in the whole system. However, the magnitude of the peak power reaches 140.0%, 180.0%, and 220.0% of the reactor full power for the added amount of 5.0, 10.0, 15.0 kg of ^{233}U nuclide, respectively.

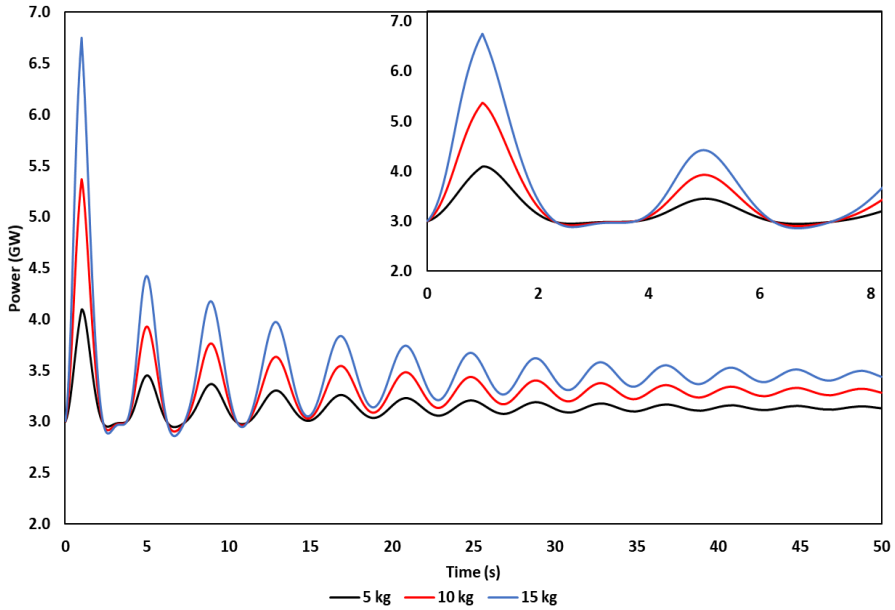


Figure 7.13. Power evolution after refueling different amounts of ^{233}U in 1.0 s.

The effect of the feeding rate on the peak reactor power is shown in Fig. 7.14 with the same amount of the ^{233}U nuclide of 5.0 kg is added during different time periods of 0.1, 0.5, 1.0, and 2.0 s. For the same amount of the added ^{233}U, as the feeding rate increases, the peak power increases, and it occurs earlier with a higher frequency. The peak power ranges from 115.0% to 145.0% of the full power depending on the feeding rate. Eventually, the power will reach the same level at the end of the transient regardless of the feeding rate.

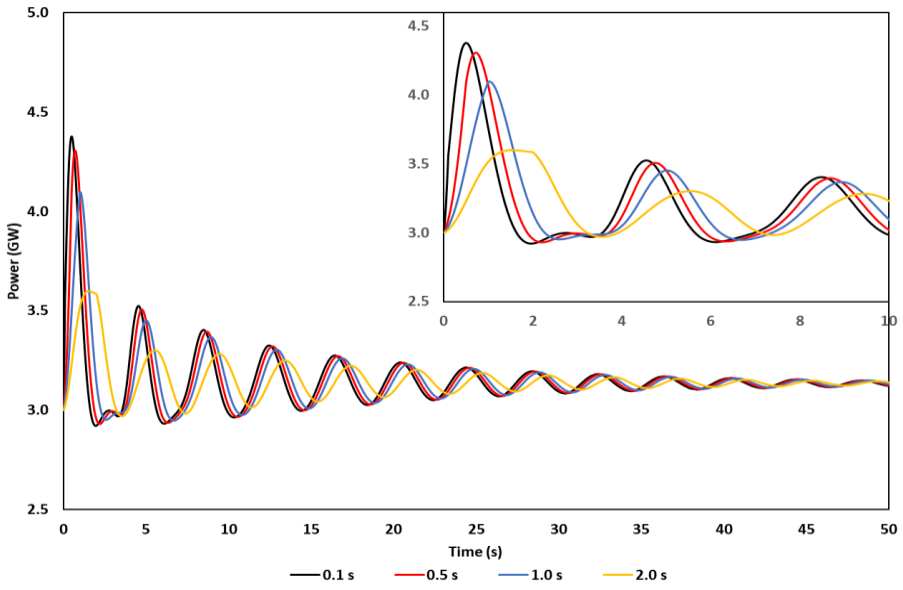


Figure 7.14. Power evolution after refueling 5.0 kg of ^{233}U nuclide at different rates.

7.2.2. Fertile Over-Fueling Transient

As mentioned in the previous sections, the fuel addition will contain a fissile and fertile component of the fuel salt to maintain criticality and breeding fissile nuclide at the same time. Adding fertile nuclide such as ^{232}Th into the fuel salts will results in an opposite scenario to that after adding fissile nuclide. Since the ^{232}Th has a high neutron capture cross section and the breeding of the ^{233}U (as described in Eq. 6.30) depends on the ^{232}Th capture rate, adding ^{232}Th to the fuel salt will increase the neutron capture rate, which will introduce a negative reactivity into the reactor core. As a result, the power will start decreasing which will lead to a decrease in the core average temperature of the fuel salt. As the ^{232}Th nuclide leaves the reactor core, the decreased fuel salt temperature will result in positive feedback, and the power starts increasing again and heating the fuel salt. The power and temperature fluctuate as the ^{232}Th, and the heated fuel slat reenters the core until it reaches a new level, as shown in Fig. 7.15, where a 100 kg of ^{232}Th nuclide is injected into the core region in 1.0 s. The power decreases below 60.0% of the initial power, and the magnitude of the peak power and frequency of the fluctuations depends on the amount and the rate at which the fertile material is being fed to the fuel salt.

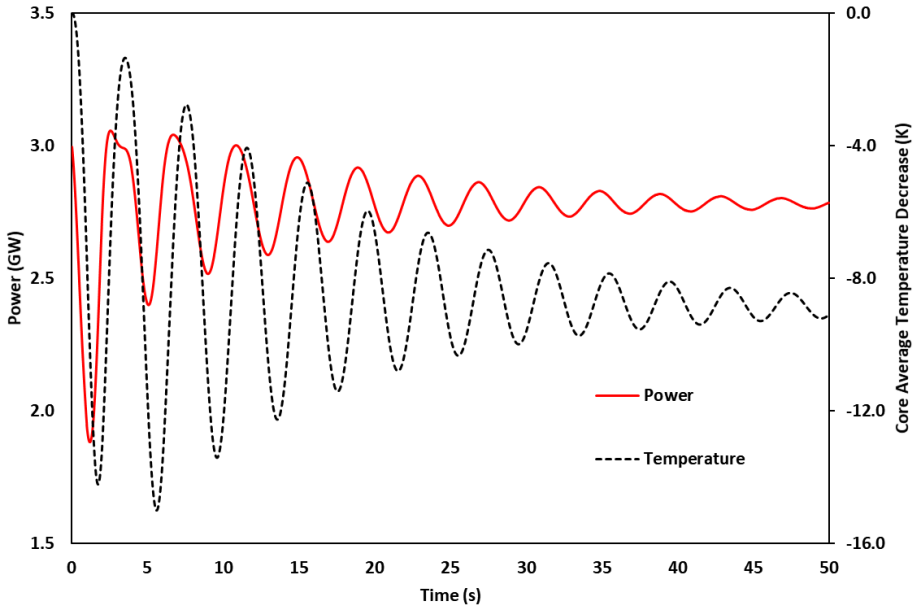


Figure 7.15. Time evolution of power and temperature after refueling 100.0 kg of ^{232}Th in 1.0 s.

These unique types of transients are very important during the normal operation of MSRs with online refueling option. It is important to understand how the power and temperature behave following such type of transient, which will help in determining the amount and the rate at which

the fissile and fertile components of the fuel salt should be added. Knowing the feeding rate of the fissile and fertile material from depletion calculations of the fuel salt is affected by the performed calculations. In the previous Chapter, the calculated feeding rate is sensitive to thermal feedback, decay heat, and considering the drift of the fuel. Also, it is important to know that the fuel salt temperature is within the design limits.

7.3. Summary

In this Chapter, further improvements were made to the feedback model of MSRs by including the decay heat and the distribution of the fuel salt nuclides in the thermal hydraulics calculations and updating the cross sections. The decay heat was calculated as a function of time after reactor shutdown, and it was grouped into several decay heat groups with unique decay constant and decay heat fractions after considering the removal of non-soluble fission products from the fuel salt which contributes to about 1.2% of the total decay heat while the soluble fission products contribute to about 4.1%, and it was used in the depletion and transient analysis of MSFR to demonstrate its significance. Furthermore, a nuclide drift model was added to simulate an over-fueling transient scenario, which is an important phenomenon related to the daily operation of MSRs with flowing fuel. The power excursion depends on the rate and the amount of the added fissile and fertile materials to the reactor core, and the simulation results suggest feeding the fissile and fertile nuclides at a low rate to avoid any fluctuations in the power of the reactor.

Chapter 8. Summary and Future Work

The main objective of this thesis is to develop neutronics analysis capabilities to the PROTEUS-NODAL code for the application of MSRs with flowing fuel. In MSRs, the effect of the fuel salt drift in the core region and the decay of the delayed neutron precursors outside the core region resulted in a smaller eigenvalue than for stationary fuel case and a reduction in the effective delayed neutron fraction, which changes the dynamic behavior of the reactor. Also, the fuel salt is moving in the primary system of the reactor, and it is acting as a fuel and coolant at the same time resulted in strong coupling between the neutronics and thermal hydraulics. In order to account for these effects, the steady state solvers of the PROTEUS-NODAL code were extended to handle the flow of the fuel salt. Also, transient solvers were added to the PROTEUS-NODAL code with a thermal feedback model to consider the strong coupling between the neutronics and thermal hydraulics. To perform fuel cycle analysis of MSRs, a depletion solver was added with online reprocessing and refueling capabilities. Also, further improvements were made to the transient solver by adding an adaptive time stepping algorithm and including more physics in the feedback model by considering the decay heat and the nuclide distributions in the model. Several verification and validation tests of the implemented capabilities of PROTEUS-NODAL were performed to ensure that the steady state, transient, and depletion solvers can be used for the design and safety analysis of MSRs.

8.1. Summary

The theoretical background and mathematical formulation of the solution methods implemented in PROTEUS-NODAL code for steady state and transient analyses of flowing fuel are provided in Chapter 2. The steady state solvers were extended for application for flowing fuel by solving the forward and adjoint delayed neutron precursor equations with the fuel drift term explicitly. The adjoint solution capability was added to calculate the reactor kinetics parameters of flowing fuel. Then, the solution of the time-dependent neutron diffusion equation was provided for stationary and flowing fuels using the TFSP formulation. The time-dependent delayed neutron

source was determined from the solution of the delayed neutron precursors equation, which was solved using the precursor integration method for stationary fuel, while for liquid fuel, several methods were investigated to obtain the solution of the resulting equations with fuel drift. The FDM was selected to solve the delayed neutron precursors equation because it provides a stable solution regardless of the time step size, and it can be applied for a general or an axial velocity field, which is important for fast spectrum reactors since there are no fuel channels. Also, the CMFD acceleration scheme was implemented for steady state and transient calculations, which reduced the computational time significantly, especially for transient problems. The thermal feedback capability was added to consider the strong coupling of the neutronics and thermal hydraulics. This includes steady state and transient thermal hydraulics solvers to calculate the fuel salt temperature, density and velocity fields, and the moderator temperature distribution. An outer loop model was added to consider the heat exchanger. The cross sections were functionalized differently for fast and thermal spectrum reactors considering the temperature and density of the fuel salt and the temperature of the solid moderator. The introduced coupling scheme for transient calculations was performed by operator splitting approach with loose coupling or time-lagged scheme. However, the neutronics and thermal hydraulics were tightly coupled in steady state calculations and solved iteratively until the power and temperature solutions were converged.

Verification results of the developed capabilities were provided in Chapter 3 using two benchmark problems for fast and thermal spectrum MSRs. The first benchmark is the MSFR, which is a fast spectrum reactor based on the thorium fuel cycle with two fuel options utilizing the ^{233}U and TRU element as fissile material. The design report of the MSFR includes several steady state calculations results, which were used to verify the results of the PROTEUS-NODAL including eigenvalue, feedback coefficients, kinetics parameters, and losses in the delayed neutron fraction. However, for transient analysis, the PROTEUS-NODAL results were verified against the solutions of other institutes obtained from the open literature for several unprotected transients related to temperature, flow, or direct reactivity changes. The PROTEUS-NODAL results for steady state analysis were consistent with reference solutions. For time-dependent power and average temperature rise across the core generally agreed well in shape and magnitude with other reported solutions in the open literature. The second benchmark is the MOST exercise, which is a thermal spectrum reactor where the fuel is flowing into solid moderator channels of graphite, and it was developed based on the MSRE experiment with several design simplifications. The

PROTEUS-NODAL results were compared with other participants in the exercise, and the results agree very well with them. The main reason for analyzing the MOST benchmark is to establish a multigroup cross sections generation procedure for the MSRE analysis by determining the energy group structure and number of energy groups required for performing the validation tests of the MSRE experiment presented in Chapter 4 where the exact core of the MSRE was analyzed including the ex-core regions. Validation tests of PROTEUS-NODAL were performed using the measurements of the ^{235}U and ^{233}U cores of the MSRE. The temperature coefficients, the reactivity loss by fuel circulation, the protected pump startup, and coast down transients at zero power, and the natural circulation test of the MSRE were analyzed, and the results agreed very well with measurements. The analysis of the protected pump startup and coast down transients at zero power showed that the PROTEUS-NODAL calculation with a given inlet flow rate could follow the time-dependent reactivity change due to delayed neutron precursor drift. The analysis of the natural circulation test showed that the PROTEUS-NODAL code could reproduce the measured power evolution due to thermal feedback for given inlet temperature and flow rate. The verification and validation results presented in Chapters 3 and 4 indicate that PROTEUS-NODAL can be used reliably for steady state and transient analyses of fast and thermal spectrums MSRs.

Further improvements were incorporated into the transient solvers of the PROTEUS-NODAL code by implementing an adaptive time-stepping solution algorithm based on the control theory approach, as discussed in Chapter 5. Typically, during the transient simulation, prespecified or scheduled sizes of the time steps are provided at the beginning of the simulation. In this way, it is hard to vary the time-step size in advance without knowing the solution behavior over the transient period. Using a small time-step size will ensure the solution's accuracy, but it will be computationally expensive. On the other hand, large time-step size will reduce the computational time at the expense of accuracy. The suggested adaptive time-stepping algorithm varies the time-step size while maintaining the solution accuracy. This was achieved by monitoring the estimated local error of the solution and maintaining it below a prespecified limit. The implemented solution algorithm was tested using various types of transients using the MSFR benchmark and MSRE experiment, and the results were compared to the reference solution obtained with fine time-step size. Different local error estimates were tested, and other parameters of the algorithm like the tolerance level and the gain value were varied to test the robustness of the algorithm. In all the simulated transients, the ATS solution showed a vast reduction in the computational time while

ensuring the accuracy of the solution. At the end of the chapter, suggested ranges of the controller parameters were provided for different transient types.

To perform fuel cycle analysis of MSRs, depletion calculations capability was added to PROTEUS-NODAL with online refueling and reprocessing, as presented in Chapter 6. The modified nuclide depletion equation for MSRs was introduced, and the solution scheme of the resulted system of equations was presented. Verification tests of the implemented capabilities were performed using the MSFR benchmark and compared to other reference solutions provided in the MSFR design report. Further analysis was performed to investigate the effects of the drift of the fuel salt, the decay heat, and coupling with the thermal hydraulics on the fissile feeding rate and the breeding ratio. In Chapter 7, the implemented feedback model was further improved by considering the decay heat and the distribution of the fuel salt nuclides in the model. First, the decay heat produced by the decay of the fission products was grouped into several decay precursor groups with decay constant and decay heat fractions after considering the removal of non-soluble fission products from the fuel salt. Then the decay heat distribution and its partial release in the core region were considered in the steady state and transient analyses, and the test results showed some differences in the power evolution and average core temperature increase. Also, an over-fueling transient scenario in MSR was simulated by injecting fissile and fertile components of the fuel salt into the primary loop by utilizing a nuclide drift model that was developed to determine the nuclide distribution in the core region during the transient. The power excursion depends on the rate and the amount of the added fissile and fertile materials to the reactor core, and the simulation results indicate feeding the fissile and fertile material at a low rate to avoid any fluctuations in the power of the reactor.

8.2. Contributions to MSRs Field

This thesis presented the neutronics analyses capabilities developed for application to flowing fuel MSRs and implemented to PROTEUS-NODAL. These capabilities can be applied to thermal and fast spectrums reactors considering general or axial velocity fields. The verification and validation test results of the MSRs neutronics capabilities of the PROTEUS-NODAL code indicate that they can be used efficiently for steady state, transient, and fuel cycle calculations in order to perform design and safety analyses of flowing fuel MSRs.

The analyses and validation tests that were performed for the MSRE static and transient experiments provide very useful information for developing accurate models and multigroup cross sections generation. The accurate representation of the MSRE geometry by including the control rod and ex-vessel regions in the developed model with appropriate boundary conditions reflect their importance on the performed calculations compared to the results of the simplified benchmark developed for the MSRE core as discussed in Chapter 3. Also, the multigroup cross section generation procedure for the MSRE that was followed to determine the energy group structures, the number of energy groups, and the tally regions to account for spectral transition within the core provides very useful information that can be generalized for graphite moderated reactors with liquid fuel. The improved static and transient analyses results compared to the results of the simplified model of the MSRE show the importance of developing accurate models with appropriate multigroup cross-sections in simulating the experiments.

The implemented adaptive time-stepping algorithm for transient analyses of MSRs with flowing fuel improved the efficiency of transient calculations. The test results show a vast reduction of the computational time while ensuring an accurate solution by maintaining the estimated local error of the solution below a prespecified tolerance level. This helps in avoiding unnecessarily small time-step size to ensure an accurate solution and thus improves the solution efficiency. Also, adjusting the time step size for the simulated transient in advance is very challenging, especially for long and slow transients like the natural circulation test of the MSRE. The suggested ranges of the controller parameters for different transient types provide very valuable information for using the adaptive time stepping algorithm efficiently. Also, the same adaptive time-stepping algorithm can be utilized for performing depletion calculation.

Finally, the developed fuel cycle analysis capability can be utilized for calculating the decay heat generated by the fission products considering their chemical characteristics to account for the decay heat contribution in the performed analysis. Also, the calculation procedure of the decay heat parameters utilized for the MSFR benchmark can be followed for other types of MSRs.

8.3. Future Work

More accurate solution of the steady state and the transient analyses of MSRs can be achieved by coupling PROTEUS-NODAL to a thermal hydraulics analysis code for accurate modeling of the primary loop specifically the heat exchanger region to perform multiphysics simulations for

MSRs. PROTEUS-NODAL provides the basic environments to be coupled with other physics codes by allowing the exchange of coupling variables. Recently, PROTEUS-NODAL was coupled to the system analysis module SAM [42] under Moose framework [39] for steady state and transient analyses of MSRs, and preliminary results are presented in Ref. [50, 51, 52]. SAM was used to replace the thermal hydraulics solver of PROTEUS-NODAL to model the primary system of the simulated reactors accurately. The preliminary test results show that the coupled code system was able to perform transient analyses of MSRs successfully and allowing more realistic modeling of the heat exchanger and primary system and providing a general velocity field required for calculating the delayed neutron source in the MSFR case. Also, the adaptive time-stepping solution algorithm was utilized to predict the time-step size of PROTEUS-NODAL and SAM effectively, and the preliminary test results indicate that the algorithm can be applied to coupled multiphysics codes to determine the time-step size while ensuring the accuracy of the solution.

Furthermore, it is suggested to couple PROTEUS-NODAL and SAM to a thermochemistry code to perform chemical analysis of the fuel salt for fuel cycle analysis. The coupling environment with other codes is already established and successfully tested between PROTEUS-NODAL and SAM and it can be easily extended to include other physics analysis tools. The chemical analysis of the fuel salt is important to determine the fuel salt molar composition required for neutronics calculations, as discussed for the nuclide drift model. Also, to determine the thermophysical properties of the fuel salt during operation, that is affected by gas bubbling and precipitation of certain fission products, which also affects the operation of the heat exchanger. In solid moderated MSR, like graphite as the MSRE experiment, it is important to determine the distribution of the short-lived and highly poisoning gaseous fission products like xenon in the core and their diffusivity in the graphite. The rates of diffusion from the salt to the graphite and to the gas bubbles in the salt need to be calculated. Previous studies [133] show that 10% of the produced noble gases would deposit on the graphite surface in the core, and 40% would deposit on metal surfaces in the circulation system, while 50% would enter the gas bubbles and be transported to the off-gas system. These important phenomena need to be addressed in performing advanced safety and design analysis of MSRs.

Appendix A. Solution Methods for Delayed Neutron Precursor Equation of Flowing Fuel

As mentioned in Chapter. 2, three different numerical methods were examined to solve the delayed neutron precursor equation for 1D axial flow: the method of characteristics (MOC), the method of lines (MoL), and the finite difference method (FDM). FDM was selected and implemented in PROTEUS-NODAL since it produces the null transient result correctly and it can be easily extended to multi-dimensional flows. The FDM method was discussed in Chapter 2, only the MOC and MoL are discussed in this Appendix, and the results for three transient scenarios are compared with the FDM solutions.

A.1. Solution Methods for 1D Axial Flow

A.1.1. Method of Characteristics

Under the assumption that the fuel velocity is constant within a node, the precursor equation for 1D axial flow given in Eq. (2.68) can be solved analytically using the MOC approach. Along a characteristic line that can be represented by a parameter *s*, the terms including the temporal and spatial derivatives can be represented by the total derivative with respect to *s*. Denoting the precursor concentration of family *k* in a radial mesh *i* by $C_{k,i}$, its derivative with respect to *s* becomes

$$\frac{d}{ds}C_{k,i}\left(z(s),t(s)\right)=\frac{\partial C_{k,i}}{\partial t}\frac{dt}{ds}+\frac{\partial C_{k,i}}{\partial z}\frac{dz}{ds}\,, \tag{A.1}$$

where *i* refers to the index for a radial mesh. Therefore, Eq. (2.68) can be written as an ordinary differential equation (ODE) as

$$\frac{d}{ds}C_{k,i}(s)+\lambda_k C_{k,i}(s)=\lambda\psi_{k,i}(s)\,, \tag{A.2}$$

with the following auxiliary ODEs for the characteristic line

$$\frac{dt}{ds}=1,\quad \frac{dz}{ds}=u_i\,. \tag{A.3}$$

The characteristic lines can be determined by solving Eq. (A.3) as

$$\begin{cases} t = s + t_0 \\ z = u_i s + z_0 \end{cases}, \tag{A.4}$$

where t_0 and z_0 are the coordinates for the starting point where the initial or boundary condition is given. Along a characteristic line, the ODE in Eq. (A.2) can be solved as

$$C_{k,i}(s) = C_{k,i}(0)e^{-\lambda_k s} + \lambda \int_0^s \psi_{k,i}(s')e^{-\lambda_k (s-s')} ds' . \tag{A.5}$$

In Eq. (A.5), $C_{k,i}(0) = C_{k,i}(z_0, t_0)$ is the initial or boundary condition given at the starting point of the characteristic line. For example, as illustrated in Fig. A.1, the precursor concentration in an axial node j with an interval $[z_{j-1}, z_j]$ during a time interval $[t_{n-1}, t_n]$ can be obtained starting from the initial condition $C_{k,ij}^{n-1}(z)$, $z_{j-1} \le z \le z_j$ given at $t = t_{n-1}$ or the inlet boundary condition $C_{k,ij}^{in}(t)$, $t_{n-1} \le t \le t_n$ given at $z = z_{j-1}$.

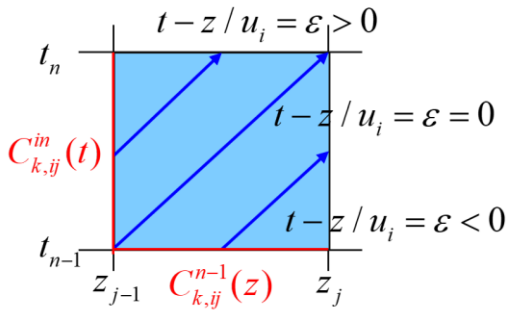


Figure A.1. Illustration of characteristic lines for precursor concentration calculation.

Along a characteristic line starting from the line $t = t_{n-1}$, $z_0 = z - u_i(t - t_{n-1}) > z_{j-1}$ and thus $t - t_{n-1} < (z - z_{j-1})/u_i$. On the other hand, along a characteristic line starting from the line $z = z_{j-1}$, $t_0 = t - (z - z_{j-1})/u_i > t_{n-1}$ and thus $t - t_{n-1} > (z - z_{j-1})/u_i$. Therefore, the precursor concentration in the axial node j during a time interval $[t_{n-1}, t_n]$ can be determined as

$$C_{k,i}(z,t) = \begin{cases} C_{k,ij}^{in}(t)e^{-\lambda_k \frac{z - z_{j-1}}{u_i}} + \dfrac{\lambda}{u_i}\displaystyle\int_{z_{j-1}}^{z} \psi_{k,i}\left[z', t - \frac{z - z'}{u_i}\right] e^{-\lambda_k \frac{z - z'}{u_i}} dz', & t - t_{n-1} > \dfrac{z - z_{j-1}}{u_i} \\ C_{k,ij}^{n-1}(z)e^{-\lambda_k (t - t_{n-1})} + \dfrac{\lambda}{u_i}\displaystyle\int_{z - u_i(t - t_{n-1})}^{z} \psi_{k,i}\left[z', t - \frac{z - z'}{u_i}\right] e^{-\lambda_k \frac{z - z'}{u_i}} dz', & t - t_{n-1} < \dfrac{z - z_{j-1}}{u_i} \end{cases} . \tag{A.6}$$

At each time node, the precursor concentrations at the outgoing surface of each node are

determined using Eq. (A.6). The precursor production and decay during the time interval Δt_n are evaluated along the characteristic line from a starting point z_{start}^n to an ending point at $z = z_{out}$. For an ending point at $z = z_{out}$ at $t = t_n$, the starting point at $t = t_{n-1}$ can be determined as

$$z_{start}^n = z_{out} - u_i \Delta t_n . \tag{A.7}$$

The precursor concentration at the starting position is determined by the linear interpolation of precursor concentrations at the previous time node as

$$C_{k,i}(z_{start}^n) = \gamma C_{k,i}^{n-1}(z_{in}) + (1-\gamma) C_{k,i}^{n-1}(z_{out}), \tag{A.8}$$

where z_{in} is the inlet coordinate of the node of interest and

$$\gamma = (z_{out} - z_{start}^n) / (z_{out} - z_{in}) . \tag{A.9}$$

In order to evaluate the precursor generation rate in Eq. (A.6), the unknown flux at the current time node t_n is approximated by assuming the flux varies exponentially in the time interval $[t_{n-1}, t_n]$ as

$$\phi_{g,ij}(z,t) = \phi_{g,ij}^{n-1}(z) e^{\alpha_n (t - t_{n-1})}, \tag{A.10}$$

The inverse period α_n is approximately determined using the power levels at t_{n-2} and t_{n-1} as

$$\alpha_n = \frac{1}{\Delta t_{n-1}} \ln \frac{P_{n-1}}{P_{n-2}} . \tag{A.11}$$

Figure A.2 illustrates the method to calculate the precursor concentration at $t = t_n$ on the outlet surface of node j (i.e., at $z = z_j$). If the fuel salt moves through L nodes during a time interval $[t_{n-1}, t_n]$, then starting from the node $j-(L-1)$, the outgoing precursor concentrations of the nodes between z_{start}^n to z_j are calculated as

$$C_{k,i,j-l}^n = \begin{cases} C_{k,i}(z_{start}^n) e^{-\lambda_k \frac{z_{out} - z_{start}}{\bar{u}_{i,j-l}^n}} + \dfrac{\lambda}{\bar{u}_{i,j-l}^n} \displaystyle\int_{z_{start}^n}^{z_{j-l}} \psi_{k,i,j-l}^{n-1}(z') e^{-(\lambda_k + \alpha_n) \frac{z_{j-l} - z'}{\bar{u}_{i,j-l}^n}} dz', & z_{j-l-1} < z_{start}^n \\ C_{k,i,j-l-1} e^{-\lambda_k \frac{z_{out} - z_{start}}{\bar{u}_{i,j-l}^n}} + \dfrac{\lambda}{\bar{u}_{i,j-l}^n} \displaystyle\int_{z_{j-l-1}}^{z_{j-l}} \psi_{k,i,j-l}^{n-1}(z') e^{-(\lambda_k + \alpha_n) \frac{z_{j-l} - z'}{\bar{u}_{i,j-l}^n}} dz', & z_{j-l-1} > z_{start}^n \end{cases}, \quad l = 1, \cdots, L-1, \tag{A.12}$$

where $\bar{u}_{i,j}^n$ is the average fuel velocity during the time step $[t_{n-1}, t_n]$ in an axial node j in a radial

mesh i, and the outgoing concentrations from a node are used as the incoming concentrations for the next node. With this method, the precursor concentrations at $t = t_n$ are calculated using the precursor concentrations at $t = t_{n-1}$ and the precursor generation rates extrapolated from the power levels at the previous two time nodes. It was observed that this method could be unstable for low velocities with which the fuel salt can move only a small fraction of a node during a time interval.

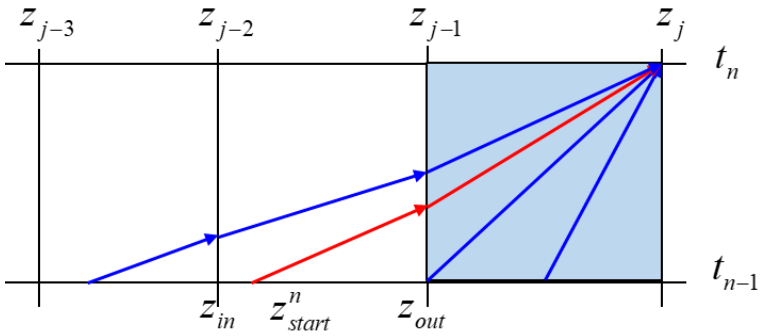


Figure A.2. Fuel salt path at each time step from a starting point to a node outlet.

A.1.2. Method of Lines

In this method, the precursor balance equation in Eq. (2.68) is solved semi-analytically by converting the PDE to a system of ODEs and solving the resulting system of ODEs using a proper ODE solver. This can be done by discretizing one of the derivatives of Eq. (2.68) and solving the system of ODEs for the remaining variable. The most important advantage of this MoL approach comes from the fact that ODE is generally simpler to solve than PDE, and the implicit methods can be used for discretization. In this study, the time derivative is discretized with the backward Euler (implicit) method as

$$\frac{\partial}{\partial t} C_{k,i}(z,t) = \frac{C^n_{k,i}(z) - C^{n-1}_{k,i}(z)}{\Delta t_n}, \qquad \Delta t_n = t_n - t_{n-1}. \tag{A.13}$$

By inserting Eq. (A.13) into Eq. (2.86), the following equation can be obtained for each radial node *i*

$$\frac{d}{dz}[u^n_i(z) C^n_{k,i}(z)] + \left(\lambda_k + \frac{1}{\Delta t_n}\right) C^n_{k,i}(z) = \lambda \psi^n_{k,i}(z) + \frac{1}{\Delta t_n} C^{n-1}_{k,i}(z). \tag{A.14}$$

The total delayed neutron precursor source at the right-hand side is related to both the delayed neutron precursor source at the current time step and the delayed neutron precursor concentration at the previous time step. The time-dependent delayed precursor equation at each time step can be written in a more convenient form to solve as

$$\frac{d}{dz}[u_i^n(z)C_{k,i}^n(z)]+\frac{\lambda_k^n}{u_i^n(z)}[u_i^n(z)C_{k,i}^n(z)]=Q_{k,i}^n(z), \tag{A.15}$$

where

$$\lambda_k^n=\lambda_k+\frac{1}{\Delta t_n},\quad Q_{k,i}^n(z)=\lambda\psi_{k,i}^n(z)+\frac{1}{\Delta t_n}C_{k,i}^{n-1}(z). \tag{A.16}$$

Instead of using an ODE solver, Eq. (A.15) can be solved analytically for the delayed neutron precursor concentration in terms of the inlet precursor concentration and the precursor concentration at the previous time step as

$$u_i^n(z)C_{k,i}^n(z)=e^{-\lambda_k^n\int_{z_0}^{z}\frac{dz'}{u_i^n(z')}}\left[u_i^n(z_0)C_{k,i}^n(z_0)+\int_{z_0}^{z}dz'Q_{k,i}^n(z')e^{\lambda_k^n\int_{z_0}^{z'}\frac{dz''}{u_i^n(z'')}}\right]. \tag{A.17}$$

Then, the outgoing and average precursor concentrations of an axial node j at a time node n can be obtained as

$$C_{k,ij}^n=\frac{u_{i,j-1}^n}{u_{ij}^n}e^{-\lambda_k^n\Delta t_{ij}^n}C_{k,i,j-1}^n+\frac{\bar{u}_{ij}^n}{u_{ij}^n\lambda_k^n}Q_{k,ij}^n\left[1-e^{-\lambda_k^n\left(z-z_{j-1}\right)/\bar{u}_{ij}^n}\right], \tag{A.18}$$

$$\bar{C}_{k,ij}^n=\frac{u_{i,j-1}^n}{u_{ij}^n}\frac{1}{\lambda_k^n\Delta t_{ij}^n}\left[1-e^{-\lambda_k^n\Delta t_{ij}^n}\right]C_{k,i,j-1}^n+\frac{1}{\lambda_k^n}Q_{k,ij}^n\left\{1-\frac{1}{\lambda_k^n\Delta t_{ij}^n}\left[1-e^{-\lambda_k^n\Delta t_{ij}^n}\right]\right\}. \tag{A.19}$$

The node-averaged precursor concentration in Eq. (A.19) depends on the fission rate at the end of a time step. In the MoL method, the fission rates are determined iteratively by coupling Eq. (A.19) with the TFSP at the end of a time step, while in the MOC approach, they were determined approximately with an assumed exponential flux shape with an inverse period determined by extrapolating the power levels of two previous time nodes. However, the time discretization error introduces some errors in the initial precursor concentrations, and this could lead to an instability issue.

A.2. Comparison of Transient Solutions

The accuracy of the three solution approaches, FDM, MOC, and MoL were examined by solving a null transient problem and the pump startup and coast down transient problems without thermal feedback. For the latter two transients, the fuel velocity was increased or decreased exponentially with a time constant of 1 s, and a fuel salt transient time of 4 s (2 s in the core and 2 s outside of the core) was used. Fig. A.3 compares the power evolutions during the first 10 s obtained with the three solution methods for a null transient. It is clearly seen that the FDM is

stable and produces the steady state solution accurately. However, the MOC and MoL power solutions show diverging behaviors due to error accumulation, although the magnitudes are very small. The MoL power solution increases steadily while the MOC power solution decreases. It appears that these unstable behaviors are due to the approximations introduced in MOC and MoL that do not reproduce the initial steady state precursor concentrations.

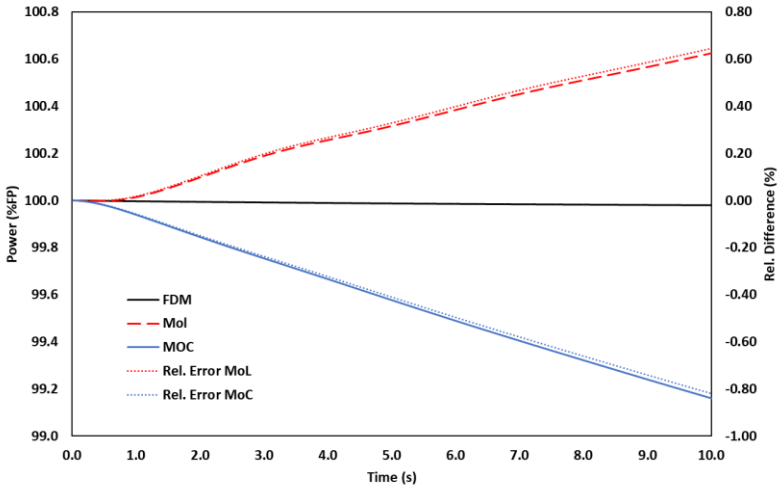


Figure A.3. Power evolution in null transient without thermal feedback.

The power response to the change of the fuel salt flow rate due to the pump startup and pump coast down transients are shown in Fig. A.4 and Fig. A.5, respectively. The pump startup introduces a negative reactivity due to the increased delayed neutron fraction decaying outside of the core, and thus the power decreases with time in an oscillatory way, as shown in Fig. A.4. Relative to the FDM solution, the MOC and MoL solutions show oscillatory errors. The MOC solution error varies from about -6% to ~3%, but it can be seen that the amplitude decreases with time. The MoL solution error varies from about -3% to ~3%, and its amplitude initially increases but decreases after ~20 s. The pump coast down introduces a positive reactivity, and thus the power increases exponentially, as shown in Fig. A.5. It can be seen that both the MOC and MoL solution errors increase monotonically in the opposite direction as in the null transient case shown in Fig. A.3.

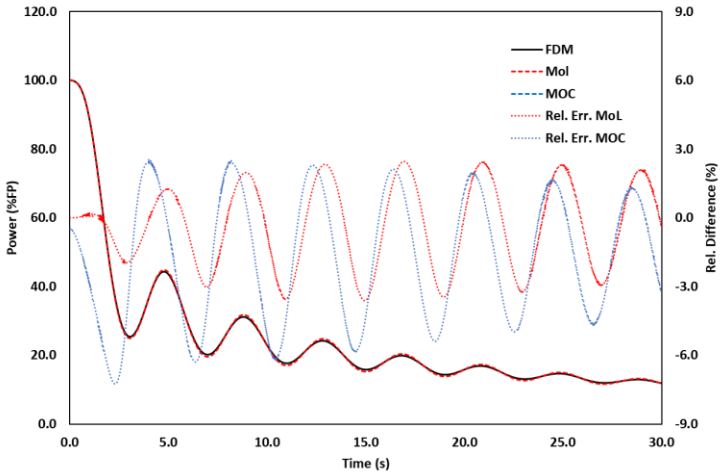


Figure A.4. Power evolution in pump start-up transient without thermal feedback.

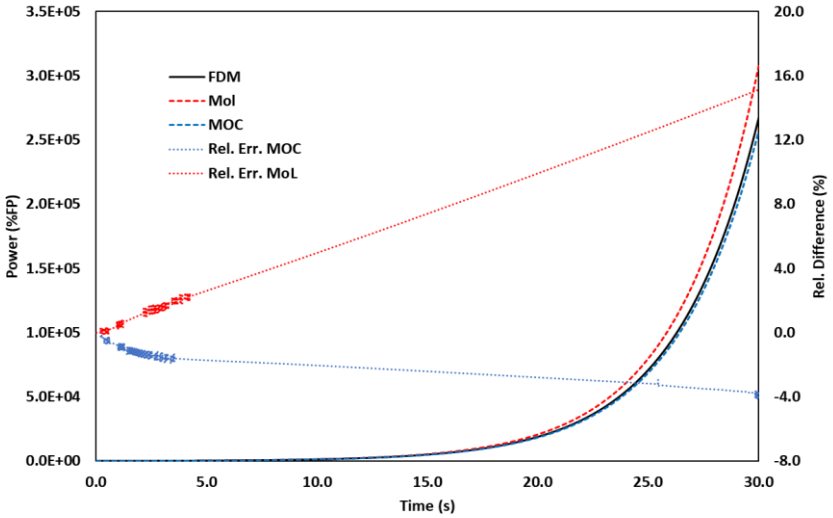


Figure A.5. Power evolution in pump coast-down transient without thermal feedback.

Appendix B. Point Kinetics Equations for Molten Salt Reactors

To reduce the computational time without a significant loss of accuracy, quasi-static solvers have been implemented in a consistent way to the TFSP solver for stationary and flowing fuels into the PROTEUS-NODAL code. This Appendix provides the derivation and the solution algorithms of the quasi-static solvers added for stationary and flowing fuels. However, these methods could speed up the solution by about 3 to 7 times for most of the studied cases, but it fails to reproduce the reference solution for transients with fuel salt velocity changes significantly, as in the pump startup transients and a step change in the fuel salt velocity.

The time-dependent neutron flux and delayed neutron precursor equations for a molten salt reactor can be written in an operator form as

$$\begin{bmatrix} \frac{1}{v}\frac{\partial}{\partial t} - (\lambda_0 F_p - M) & -\lambda_0 D_c \\ -F_c & \frac{\partial}{\partial t} + M_c \end{bmatrix} \begin{bmatrix} \phi \\ \vec{C} \end{bmatrix} = 0, \tag{B.1}$$

where λ_0 is the initial eigenvalue introduced to make the time derivatives identically zero at the initial steady state condition that might be off critical and the other operators are defined as

$$M\phi = -\nabla \cdot D(\vec{r},E,t)\nabla\phi(\vec{r},E,t) + \Sigma_t(\vec{r},E,t)\phi(\vec{r},E,t) - \int dE'\Sigma_s(\vec{r},E'\to E,t)\phi(\vec{r},E',t), \tag{B.2}$$

$$F_p\phi = \chi_p(\vec{r},E,t)\int dE'\nu_p\Sigma_f(\vec{r},E',t)\phi(\vec{r},E',t), \tag{B.3}$$

$$F_c\phi = \left[\int dE'\nu_{d1}\Sigma_f(\vec{r},E',t)\phi(\vec{r},E',t),\cdots,\int dE'\nu_{dK}\Sigma_f(\vec{r},E',t)\phi(\vec{r},E',t)\right]^T, \tag{B.4}$$

$$\vec{C} = [C_1(\vec{r},t),\cdots,C_K(\vec{r},t)]^T, \tag{B.5}$$

$$D_c\vec{C} = \left[\chi_{d1}(\vec{r},E,t)\lambda_1,\cdots,\chi_{dK}(\vec{r},E,t)\lambda_K\right]\vec{C}, \tag{B.6}$$

$$M_c\vec{C} = \mathrm{diag}\{\nabla\cdot[\vec{u}(\vec{r},t)(\cdot)] + \lambda_1,\cdots,\nabla\cdot[\vec{u}(\vec{r},t)(\cdot)] + \lambda_K\}\vec{C}\ . \tag{B.7}$$

B.1. Steady State Equations

B.1.1. Forward Equations

At the initial steady state, the time derivatives become zero and thus Eq. (B.1) is reduced to

$$\begin{bmatrix} (\lambda_0 F_{p0} - M_0) & \lambda_0 D_{c0} \\ F_{c0} & -M_{c0} \end{bmatrix} \begin{bmatrix} \phi_0 \\ \vec{C}_0 \end{bmatrix} = 0, \tag{B.8}$$

where

$$M_0\phi_0 = -\nabla \cdot D(\vec{r},E)\nabla\phi_0(\vec{r},E) + \Sigma_t(\vec{r},E)\phi_0(\vec{r},E) - \int dE'\Sigma_s(\vec{r},E' \to E)\phi_0(\vec{r},E'), \tag{B.9}$$

$$F_{p0}\phi_0 = \chi_p(\vec{r},E)\int dE'\nu_p\Sigma_f(\vec{r},E')\phi_0(\vec{r},E'), \tag{B.10}$$

$$F_{c0}\phi_0 = \left[\int dE'\nu_{d1}\Sigma_f(\vec{r},E')\phi_0(\vec{r},E'), \cdots, \int dE'\nu_{dK}\Sigma_f(\vec{r},E')\phi_0(\vec{r},E')\right]^T, \tag{B.11}$$

$$\vec{C}_0 = [C_{10}(\vec{r}), \cdots, C_{K0}(\vec{r})]^T, \tag{B.12}$$

$$D_{c0}\vec{C}_0 = \left[\chi_{d1}(\vec{r},E)\lambda_1, \cdots, \chi_{dK}(\vec{r},E)\lambda_K\right]\vec{C}_0, \tag{B.13}$$

$$M_{c0}\vec{C}_0 = \mathrm{diag}\{\nabla\cdot[\vec{u}_0(\vec{r},(\cdot)] + \lambda_1, \cdots, \nabla\cdot[\vec{u}_0(\vec{r})(\cdot)] + \lambda_K\}\vec{C}_0 \ . \tag{B.14}$$

The second equation of Eq. (B.8) can be solved formally for the precursor concentration as

$$\vec{C}_0 = M_{c0}^{-1}F_{c0}\phi_0. \tag{B.15}$$

Using Eq. (B.15), the steady state flux equation can be rewritten as

$$M_0\phi_0 = \lambda_0(F_{p0} + D_{c0}M_{c0}^{-1}F_{c0})\phi_0 = \lambda_0 F_0\phi_0, \tag{B.16}$$

where F_0 is a quasi-stationary total fission operator.

B.1.2. Adjoint Equations

The adjoint equations to the steady state flux and precursor concentration equations in Eq. (B.8) can be obtained as

$$\begin{bmatrix} (\lambda_0 F_{p0}^* - M_0^*) & \lambda_0 F_{c0}^* \\ D_{c0}^* & -M_{c0}^* \end{bmatrix} \begin{bmatrix} \phi_0^* \\ \vec{C}_0^* \end{bmatrix} = 0, \tag{B.17}$$

where

$$M_0^*\phi_0^* = -\nabla \cdot D(\vec{r},E)\nabla\phi_0^*(\vec{r},E) + \Sigma_t(\vec{r},E)\phi_0^*(\vec{r},E) - \int dE'\Sigma_s(\vec{r},E \to E')\phi_0^*(\vec{r},E'), \tag{B.18}$$

$$F_{p0}^*\phi_0^* = \nu_p\Sigma_f(\vec{r},E)\int dE'\chi_p(\vec{r},E')\phi_0^*(\vec{r},E'), \tag{B.19}$$

$$D_{c0}^*\phi_0^* = \left[\chi_{d1}(\vec{r},E)\lambda_1, \cdots, \chi_{dK}(\vec{r},E)\lambda_K\right]^T \phi_0^*(\vec{r},E), \tag{B.20}$$

$$\vec{C}_0^* = [C_{10}^*(\vec{r}), \cdots, C_{K0}^*(\vec{r})]^T, \tag{B.21}$$

$$F_{c0}^{*}\vec{C}_{0}^{*}=\left[\nu_{d1}\Sigma_{f}(\vec{r},E),\cdots,\nu_{dK}\Sigma_{f}(\vec{r},E)\right]^{T}\vec{C}_{0}^{*}(\vec{r}),\tag{B.22}$$

$$M_{c0}^{*}\vec{C}_{0}^{*}=\mathrm{diag}\{-\nabla\cdot[\vec{u}(\vec{r})(\cdot)]+\lambda_{1},\cdots,-\nabla\cdot[\vec{u}(\vec{r})(\cdot)]+\lambda_{K}\}\vec{C}_{0}^{*}(\vec{r}).\tag{B.23}$$

As in the flux equation, the steady state adjoint flux equation can also be written as

$$M_{0}^{*}\phi_{0}^{*}=\lambda_{0}[F_{p0}^{*}+F_{c0}^{*}(M_{c0}^{*})^{-1}D_{c0}^{*}]\phi_{0}^{*}=\lambda_{0}F_{0}^{*}\phi_{0}^{*}.\tag{B.24}$$

B.1.3. Stationary Fuel

For a stationary fuel, the operator M_c is reduced to the following simple diagonal operator

$$\tilde{M}_{c}=\mathrm{diag}[\lambda_{1},\cdots,\lambda_{K}].\tag{B.25}$$

Thus, the precursor concentration can be written in terms of the neutron flux as

$$\vec{C}_{0}=\tilde{M}_{c0}^{-1}F_{c0}\phi_{0}=\left[\lambda_{1}^{-1}\int dE'\nu_{d1}\Sigma_{f}(\vec{r},E')\phi_{0}(\vec{r},E'),\cdots,\lambda_{K}^{-1}\int dE'\nu_{dK}\Sigma_{f}(\vec{r},E')\phi_{0}(\vec{r},E')\right]^{T}.\tag{B.26}$$

Similarly, the adjoint precursor concentration can be written in terms of the adjoint flux as

$$\vec{C}_{0}^{*}=\tilde{M}_{c}^{-1}D_{c0}^{*}\phi_{0}^{*}=\left[\chi_{d1}(\vec{r},E),\cdots,\chi_{dK}(\vec{r},E)\right]^{T}\phi_{0}^{*}(\vec{r},E).\tag{B.27}$$

Therefore, the flux and adjoint flux equations can be written as

$$M_{0}\phi_{0}=\lambda_{0}(F_{p0}+D_{c0}\tilde{M}_{c}^{-1}F_{c0})\phi_{0}=\lambda_{0}F_{0}\phi_{0},\tag{B.28}$$

$$M_{0}^{*}\phi_{0}^{*}=\lambda_{0}(F_{p0}^{*}+F_{c0}^{*}\tilde{M}_{c}^{-1}D_{c0}^{*})\phi_{0}^{*}=\lambda_{0}F_{0}^{*}\phi_{0}^{*},\tag{B.29}$$

where the quasi-stationary total fission operator and its adjoint are given by

$$F_{0}\phi_{0}=\int dE'\left[\chi(\vec{r},E)\nu_{p}\Sigma_{f}(\vec{r},E')+\sum_{k=1}^{K}\chi_{dk}(\vec{r},E)\nu_{dk}\Sigma_{f}(\vec{r},E')\right]\phi_{0}(\vec{r},E'),\tag{B.30}$$

$$F_{0}^{*}\phi_{0}^{*}=\int dE'\left[\chi(\vec{r},E')\nu_{p}\Sigma_{f}(\vec{r},E)+\sum_{k=1}^{K}\chi_{dk}(\vec{r},E')\nu_{dk}\Sigma_{f}(\vec{r},E)\right]\phi_{0}^{*}(\vec{r},E').\tag{B.31}$$

B.2. Quasi-Static Methods

To reduce the computational time without a significant loss of accuracy, two improved quasi-static solvers have been implemented in PROTEUS-NODAL in a consistent way to the TFSP solver for stationary and flowing fuels. The basic principle of quasi-static methods is to factorize the neutron flux into a shape and an amplitude function and take advantage of the fact that the flux shape varies more slowly than the amplitude during transient events.

B.2.1. Stationary Fuel

The quasi-static method can be applied to stationary fuel by deriving the exact point kinetics equations using the factorization method. Where the multi-group neutron flux is factored into a purely time-dependent amplitude function $p(t)$ and a space-, energy- and time-dependent shape function $\psi(r,E,t)$ as

$$\phi(r,E,t) = p(t)\psi(r,E,t)\,. \tag{B.32}$$

To make these factorizations unique, the time variation of the shape functions is constrained as

$$\left\langle \phi_0^*, v^{-1}\psi \right\rangle = \left\langle \phi_0^*, v^{-1}\psi_0 \right\rangle = K_0\,, \tag{B.33}$$

where K_0 is a constant. Multiplying the first equation of Eq. (B.1) by the initial adjoint flux and integrating over space and energy yields

$$\left\langle \phi_0^*, v^{-1}\psi \right\rangle \frac{d}{dt}\, p(t) = \left\langle \phi_0^*, (\lambda_0 F_p - M)\psi \right\rangle p(t) + \left\langle \phi_0^*, D_c\vec{C} \right\rangle. \tag{B.34}$$

By dividing Eq. (B.34) by the importance-weighted quasi-stationary source of neutrons per unit amplitude

$$F(t) = \frac{1}{p(t)}\left[\left\langle \phi_0^*, F_p\phi \right\rangle + \left\langle \phi_0^*, F_d\phi \right\rangle\right] = \left\langle \phi_0^*, F_p\psi \right\rangle + \left\langle \phi_0^*, F_d\psi \right\rangle = \left\langle \phi_0^*, F\psi \right\rangle, \tag{B.35}$$

Then, the following equation can be obtained

$$\Lambda(t)\frac{d}{dt}\, p(t) = [\rho(t) - \beta(t)]p(t) + \frac{1}{F(t)}\left\langle \phi_0^*, D_c\vec{C} \right\rangle, \tag{B.36}$$

where

$$\Lambda(t) = \frac{\left\langle \phi_0^*, v^{-1}\psi \right\rangle}{F(t)} = \frac{\left\langle \phi_0^*, v^{-1}\psi \right\rangle}{\left\langle \phi_0^*, F\psi \right\rangle} = \frac{K_0}{F(t)}\,, \tag{B.37}$$

$$\rho(t) = \frac{\left\langle \phi_0^*, (\lambda_0 F - M)\psi \right\rangle}{F(t)} = \frac{\left\langle \phi_0^*, (\lambda_0 F - M)\psi \right\rangle}{\left\langle \phi_0^*, F\psi \right\rangle}, \tag{B.38}$$

$$\beta(t) = \sum_k \beta_k(t) = \sum_k \frac{\left\langle \phi_0^*, \lambda_0 F_{dk}\psi \right\rangle}{F(t)} = \frac{\left\langle \phi_0^*, \lambda_0 F_d\psi \right\rangle}{F(t)} = \frac{\left\langle \phi_0^*, \lambda_0 F_d\psi \right\rangle}{\left\langle \phi_0^*, F\psi \right\rangle}\,. \tag{B.39}$$

Defining importance-weighted reduced precursor concentration, which is defined relative to the initial fission source as

$$\zeta_k(t) = \frac{1}{F(0)}\left\langle \phi_0^*, \chi_{dk} C_k \right\rangle. \tag{B.40}$$

Therefore, Eq. (B.36) can be rewritten as

$$\frac{d}{dt}\, p(t) = \left[\frac{\rho(t) - \beta(t)}{\Lambda(t)}\right] p(t) + \frac{F(0)}{\Lambda(t)F(t)} \sum_{k=1}^{K} \lambda_k \zeta_k(t)\,, \tag{B.41}$$

or

$$\frac{d}{dt}\, p(t) = \left[\frac{\rho(t) - \beta(t)}{\Lambda(t)}\right] p(t) + \frac{1}{\Lambda(0)} \sum_{k=1}^{K} \lambda_k \zeta_k(t)\,. \tag{B.42}$$

Multiplying the second equation of Eq. (B.1) by $\left(\chi_{dk} / F(0)\right)$, taking the inner product with the initial adjoint flux vector yields

$$\frac{d}{dt}\zeta_k(t) + \lambda_k \zeta_k(t) = \frac{F(t)}{F(0)} \beta_k(t)\, p(t). \tag{B.43}$$

Equations (B.42) and (B.43) are the set of point kinetics equations for the flux amplitude function $p(t)$. These equations are solved over a fine time grid by discretizing them using the backward Euler scheme.

B.2.1.1. Improved Quasi-Static Method (IQM)

In the IQM, the amplitude function and its time derivative are obtained from the solution of the point kinetics equations, and the shape function equation can be defined and solved over a coarse time grid as

$$\begin{bmatrix} \frac{1}{v}\left(\frac{\partial}{\partial t} + \frac{\partial p/\partial t}{p}\right) - (\lambda_0 F_p - M) & -\lambda_0 D_c \\ -F_c & \frac{\partial}{\partial t} + M_c \end{bmatrix} \begin{bmatrix} \psi \\ \vec{C}/p \end{bmatrix} = 0. \tag{B.44}$$

Eq. (B.44) with the point kinetics equations Eq. (B.42) and (B.43) form a system of nonlinearly coupled equations and need to be solved iteratively applying the following procedure

1) The initial steady state forward and adjoint flux equations are solved as a first step and the normalization factor is determined.
2) The point kinetic parameters are computed and point kinetics equations are solved over fine time steps to obtain the amplitude function and its time derivative using previous time step shape function.
3) The shape equations Eq. (B.44) are then solved for coarse time step using the updated amplitude function and its time derivative.
4) The error on the normalization condition of the shape is evaluated as

$$\varepsilon_\psi = \frac{\left\langle \phi_0^*, v^{-1}\tilde{\psi}^{n+1} \right\rangle - \left\langle \phi_0^*, v^{-1}\psi^n \right\rangle}{\left\langle \phi_0^*, v^{-1}\psi^n \right\rangle} = \frac{\left\langle \phi_0^*, v^{-1}\tilde{\psi}^{n+1} \right\rangle - K_0}{K_0}. \tag{B.45}$$

5) Steps 2 through 4 are repeated until the shape function converges within the specified error criteria.
6) Once the shape function converges and the shape function is renormalized as below, and the

delayed neutron precursor concentrations are calculated with the updated neutron flux as

$$\psi^{n+1} = \frac{K_0}{\left\langle \phi_0^*, v^{-1}\tilde{\psi}^{n+1} \right\rangle} \tilde{\psi}^{n+1}, \tag{B.46}$$

$$\phi^{n+1} = P^{n+1}\psi^{n+1}. \tag{B.47}$$

7) The above procedure (steps 2 through 6) is repeated until the end of the coarse time steps.

B.2.1.2. Predictor Corrector Quasi-Static Method (PCQM)

The predictor corrector method is easier to implement since it doesn't require an iterative solution scheme which means it reduces the computational time. In this method, the shape function is predicted by solving time dependent neutron flux and precursor equations over coarse time steps. Then, with predicted flux the point kinetics parameters are evaluated, and the point kinetics equations are solved. With calculated amplitude function, the shape function is corrected. The solution procedure can be summarized as in the following steps:

1) The initial steady state forward and adjoint flux equations and the normalization factor are solved as a first step.
2) The neutron flux $\tilde{\phi}^{n+1}$ and delayed precursors concentrations $\tilde{C}_k^{n+1}$ are evaluated over coarse time step as predicted values and to be corrected. The shape function can be obtained as

$$\psi^{n+1} = \frac{K_0}{\left\langle \phi_0^*, v^{-1}\tilde{\phi}^{n+1} \right\rangle} \tilde{\phi}^{n+1}. \tag{B.48}$$

3) The point kinetic parameters are computed and point kinetics equations are solved over fine time steps to obtain the amplitude function.
4) Using the corrected amplitude function, the neutron flux is corrected, and the delayed neutron precursor concentrations are calculated with the corrected neutron flux as

$$\phi^{n+1} = P^{n+1}\psi^{n+1}. \tag{B.49}$$

5) The above procedure (steps 2 through 4) is repeated until the end of the coarse time steps.

The IQM and PCQM solvers have been tested for step and ramp reactivity insertions for stationary fuel. Figure B.1 shows the results for 0.6$ and 1.2$ step reactivity insertions without thermal feedback and Fig. B.2 shows the results for ramp reactivity insertion with the same inserted reactivities over 0.01 s without thermal feedback. The results are compared to the reference solution that was obtained with the direct solution with a fine time-step size of 0.1 ms. The results of the IQM and PCQM were obtained with 1.0 ms and compared to the direct solution with the

same step size. The IQM and PCQM can reproduce the reference solution accurately with a larger time step size, while the solution of the direct method with a large time step deviates from the reference solution. Both methods reduced the computational time by about 3 times for the IQM and about 6 times for the PCQM.

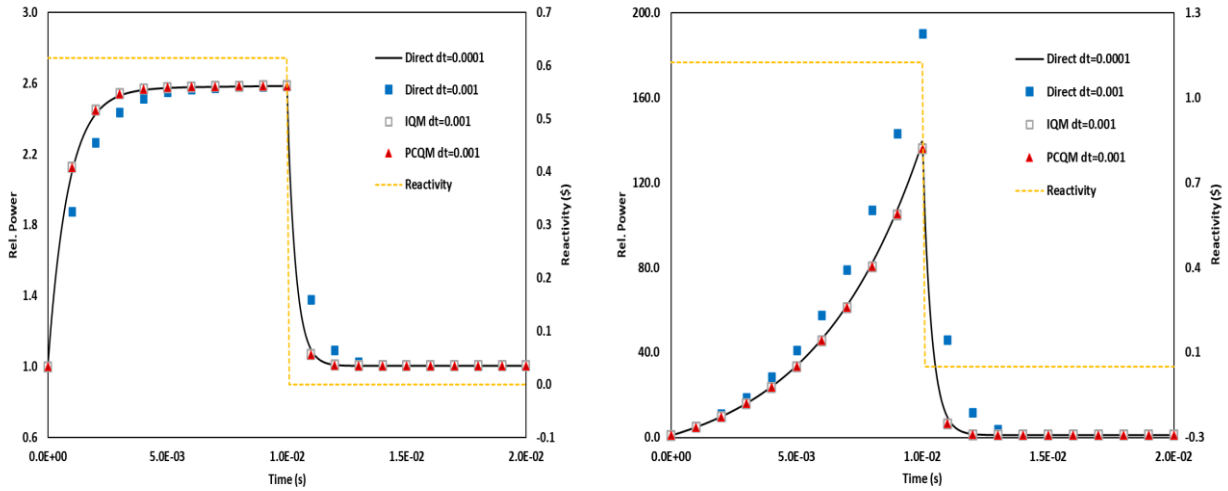


Figure B.1. Power evolution during step reactivity perturbation of 200 pcm (~0.6$) (left) and 350 pcm (~1.2$) (right) for stationary fuel.

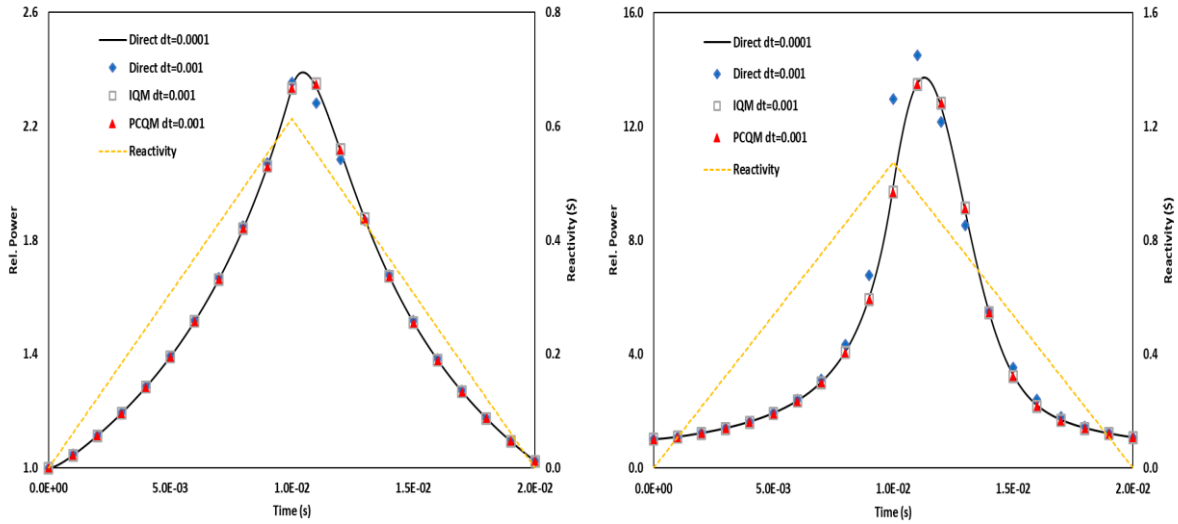


Figure B.2. Power evolution during ramp reactivity perturbation of 200 pcm (~0.6 $) (left) and 350 pcm (~1.2$) (right) for stationary fuel.

B.2.2. Flowing Fuel

As mentioned in the previous section the delayed neutron precursor equations for flowing are different than stationary fuel case, where an extra term appears in the equation to account for the precursor drift in the core. To apply the quasi-static method in solving the time-dependent neutron

diffusion equation of flowing fuel, the exact point kinetics equations must be derived in a different way than the stationary fuel case. The derivation of the exact point kinetics equations for flowing fuel starts by factoring the flux into a purely time-dependent amplitude function $p(t)$ and a space-, energy- and time-dependent shape function $\psi(r,E,t)$, and factoring the delayed neutron precursor concentrations into a purely time-dependent amplitude function $\zeta(t)$ and a space- and time-dependent shape function $G_k(r,t)$ as

$$\phi(r,E,t) = p(t)\psi(r,E,t), \tag{B.50}$$

$$C_k(r,t) = \zeta(t)G_k(r,t). \tag{B.51}$$

To make these factorizations unique, the time variation of the shape functions is constrained as

$$\left\langle \phi_0^*, v^{-1}\psi \right\rangle = K_0, \tag{B.52}$$

$$\left\langle C_0^*, G \right\rangle = G_0, \tag{B.53}$$

where K_0 and G_0 are constants. Multiplying the first equation of Eq. (B.1) by the initial adjoint flux and integrating over space and energy yields

$$\left\langle \phi_0^*, v^{-1}\psi \right\rangle \frac{d}{dt} p(t) = \left\langle \phi_0^*, (\lambda_0 F_p - M)\psi \right\rangle p(t) + \left\langle \phi_0^*, D_c\vec{C} \right\rangle, \tag{B.54}$$

Multiplying the second equation of Eq. (B.1) by the initial adjoint delayed neutron precursor concentrations and integrating over space and energy yields

$$\left\langle C_{0,k}^*, G_k \right\rangle \frac{d}{dt}\zeta_k(t) + \left\langle C_{0,k}^*, G_k \right\rangle \lambda_k \zeta_k(t) + \left\langle C_{0,k}^*, \nabla\cdot[\vec{u}(\vec{r},t)G_k] \right\rangle \zeta_k(t) = \left\langle C_{0,k}^*, F_{c,k}\psi \right\rangle p(t), \tag{B.55}$$

Dividing Eq. (B.55) by $\left\langle C_{0,k}^*, G_k \right\rangle$, then we get

$$\frac{d}{dt}\zeta_k(t) + \tau_k\zeta_k(t) = \frac{\beta_k}{\mu_k} p(t), \tag{B.56}$$

where

$$\tau_k = \lambda_k + \frac{\left\langle C_{0,k}^*, \nabla\cdot[\vec{u}(\vec{r},t)G_k] \right\rangle}{\left\langle C_{0,k}^*, G_k \right\rangle}, \tag{B.57}$$

$$\mu_k = \frac{\left\langle C_{0,k}^*, G_k \right\rangle}{F(t)} = \frac{\left\langle C_{0,k}^*, G_k \right\rangle}{\left\langle \phi_0^*, F_p\psi \right\rangle + \left\langle C_0^*, F_c\psi \right\rangle}, \tag{B.58}$$

$$\beta_k(t) = \frac{\left\langle C_{0,k}^*, F_{c,k}\psi \right\rangle}{F(t)} = \frac{\left\langle C_{0,k}^*, F_{c,k}\psi \right\rangle}{\left\langle \phi_0^*, F_p\psi \right\rangle + \left\langle C_0^*, F_c\psi \right\rangle}, \tag{B.59}$$

$$\beta(t)=\sum_k \beta_k(t)=\frac{1}{p(t)}\frac{\left\langle \phi_0^*, D_c\vec{C}\right\rangle}{F(t)}=\frac{\left\langle C_0^*, F_c\psi\right\rangle}{\left\langle \phi_0^*, F_p\phi\right\rangle+\left\langle C_0^*, F_c\psi\right\rangle}, \tag{B.60}$$

where the importance-weighted quasi-stationary source of neutrons per unit amplitude is defined

$$\begin{aligned} F(t)&=\frac{1}{p(t)}\left[\left\langle \phi_0^*, F_p\phi\right\rangle+\left\langle \phi_0^*, D_c\vec{C}\right\rangle\right]=\frac{1}{p(t)}\left[\left\langle \phi_0^*, F_p\phi\right\rangle+\left\langle C_0^*, F_c\phi\right\rangle\right] \\ &=\left\langle \phi_0^*, F_p\psi\right\rangle+\left\langle C_0^*, F_c\psi\right\rangle \end{aligned}, \tag{B.61}$$

By dividing Eq. (B.54) by the importance-weighted quasi-stationary source of neutrons per unit amplitude we obtain

$$\Lambda(t)\frac{d}{dt}p(t)=[\rho(t)-\beta(t)]p(t)+\sum_k \Lambda_k(t)\zeta_k(t), \tag{B.62}$$

where

$$\Lambda(t)=\frac{\left\langle \phi_0^*, v^{-1}\psi\right\rangle}{F(t)}=\frac{\left\langle \phi_0^*, v^{-1}\psi\right\rangle}{\left\langle \phi_0^*, F_p\psi\right\rangle+\left\langle C_0^*, F_c\psi\right\rangle}, \tag{B.63}$$

$$\rho(t)=\frac{1}{p(t)}\frac{\left\langle \phi_0^*, (\lambda_0 F_p-M)\phi\right\rangle+\left\langle \phi_0^*, D_c\vec{C}\right\rangle}{F(t)}=\frac{\left\langle \phi_0^*, (\lambda_0 F_p-M)\psi\right\rangle+\left\langle C_0^*, F_c\psi\right\rangle}{\left\langle \phi_0^*, F_p\psi\right\rangle+\left\langle C_0^*, F_c\psi\right\rangle}, \tag{B.64}$$

$$\Lambda_k(t)=\frac{1}{F(t)}\left\langle \phi_0^*, D_{c,k}G_k\right\rangle=\frac{\left\langle \phi_0^*, D_{c,k}G_k\right\rangle}{\left\langle \phi_0^*, F_p\psi\right\rangle+\left\langle C_0^*, F_c\psi\right\rangle}. \tag{B.65}$$

Then, Eq. (B.62) can be rewritten as

$$\frac{d}{dt}p(t)=\frac{\rho(t)-\beta(t)}{\Lambda(t)}p(t)+\sum_{k=1}^{K}\frac{\Lambda_k(t)}{\Lambda(t)}\zeta_k(t). \tag{B.66}$$

Eq. (B.62) and Eq. (B.56) are the exact point kinetics equations of flowing fuel.

B.2.2.1. Improved Quasi-Static Method (IQM)

In the IQM for flowing fuel, the amplitude functions of the neutron flux and the precursor concentrations and their time derivative are obtained from the solution of the point kinetics equations. Then, the shape function equations neutron flux and the precursor concentrations can be defined and solved over a coarse time grid as

$$\begin{bmatrix} \frac{1}{v}\left(\frac{\partial}{\partial t}+\frac{\partial p/\partial t}{p}\right)-(\lambda_0 F_p-M) & -\lambda_0 D_c\left(\frac{\zeta}{p}\right) \\ -F_c\left(\frac{p}{\zeta}\right) & \left(\frac{\partial}{\partial t}+\frac{\partial \zeta/\partial t}{\zeta}\right)+M_c \end{bmatrix}\begin{bmatrix}\psi \\ G\end{bmatrix}=0. \tag{B.67}$$

Eq. (B.67) with the point kinetics equations Eq. (B.62) and (B.56) form a system of nonlinearly

coupled equations and need to be solved iteratively applying the following procedure

1) The initial steady state forward and adjoint flux equations are solved for the neutron flux and precursor concentrations as a first step and the normalization factors are determined.
2) The point kinetic parameters are computed for flowing fuel and point kinetics equations for flowing fuel are solved over fine time steps to obtain the amplitude functions of the neutron flux and the precursor concentrations and their time derivative using previous time step shape functions.
3) The shape equations Eq. (B.67) are then solved for coarse time step using the amplitude functions of the neutron flux and the precursor concentrations and their time derivative.
4) The errors on the normalization condition of the shape functions are evaluated as

$$\varepsilon_{\psi} = \frac{\left\langle \phi_0^*, v^{-1}\tilde{\psi}^{n+1} \right\rangle - \left\langle \phi_0^*, v^{-1}\psi^{n} \right\rangle}{\left\langle \phi_0^*, v^{-1}\psi^{n} \right\rangle} = \frac{\left\langle \phi_0^*, v^{-1}\tilde{\psi}^{n+1} \right\rangle - K_0}{K_0}, \tag{B.68}$$

$$\varepsilon_{G} = \frac{\left\langle C_0^*, \tilde{G}^{n+1} \right\rangle - \left\langle C_0^*, G^{n} \right\rangle}{\left\langle C_0^*, G^{n} \right\rangle} = \frac{\left\langle C_0^*, \tilde{G}^{n+1} \right\rangle - Y_0}{Y_0}. \tag{B.69}$$

5) Steps 2 through 4 are repeated until the shape functions of the neutron flux and the precursor concentrations converges within the specified error criteria.
6) Once the shape functions converge, the shape functions are renormalized as

$$\psi^{n+1} = \frac{K_0}{\left\langle \phi_0^*, v^{-1}\tilde{\psi}^{n+1} \right\rangle}\tilde{\psi}^{n+1}, \qquad \phi^{n+1} = P^{n+1}\psi^{n+1}, \tag{B.70}$$

$$G^{n+1} = \frac{Y_0}{\left\langle C_0^*, \tilde{G}^{n+1} \right\rangle}\tilde{G}^{n+1}, \qquad C^{n+1} = \zeta^{n+1}G^{n+1}. \tag{B.71}$$

7) The above procedure (steps 2 through 6) is repeated until the end of the coarse time steps.

B.2.2.2. Predictor Corrector Quasi-Static Method (PCQM)

In this method, the shape functions of the neutron flux and the precursor concentrations are predicted by solving time-dependent neutron flux and precursor equations over coarse time steps. Then, using the predicted neutron flux and precursor concentrations, the point kinetics parameters are evaluated, and the point kinetics equations are solved. With calculated amplitude functions, the shape function of the neutron flux and the precursor concentrations are corrected. The solution procedure can be summarized as in the following steps

1) The initial steady state forward and adjoint flux equations are solved for the neutron flux and

precursor concentrations as a first step and the normalization factors are determined.

2) The neutron flux $\tilde{\phi}^{n+1}$ and delayed precursors concentrations $\tilde{C}_k^{n+1}$ are evaluated over coarse time step as predicted values and to be corrected. The shape functions can be obtained as

$$\psi^{n+1} = \frac{K_0}{\left\langle \phi_0^*, v^{-1}\tilde{\phi}^{n+1} \right\rangle} \tilde{\phi}^{n+1}, \tag{B.72}$$

$$G^{n+1} = \frac{Y_0}{\left\langle C_0^*, \tilde{G}^{n+1} \right\rangle} \tilde{G}^{n+1}. \tag{B.73}$$

3) The point kinetic parameters are computed and point kinetics equations are solved over fine time steps to obtain the amplitude functions of the neutron flux and precursor concentrations.

4) Using the corrected amplitude functions, the neutron flux and precursor concentrations are corrected as

$$\phi^{n+1} = P^{n+1}\psi^{n+1}, \tag{B.74}$$

$$C^{n+1} = \zeta^{n+1}G^{n+1}. \tag{B.75}$$

5) The above procedure (steps 2 through 4) is repeated until the end of the coarse time steps.

The IQM and PCQM solvers have been tested first for step and ramp reactivity insertions for flowing fuel in a similar way as for the stationary fuel. Figure B.3 shows the results for 0.67$ and 1.15$ step reactivity insertions without thermal feedback, and Fig. B.4 shows the results for ramp reactivity insertion with the same inserted reactivities over 0.01 s without thermal feedback. The results are compared to the reference solution that was obtained with the direct solution of the flowing fuel equations with a fine time-step size of 0.1 ms. The results of the IQM and PCQM were obtained with 1.0 ms and compared to the direct solution with the same step size. The IQM and PCQM can reproduce the reference solution accurately with a larger time step size, while the solution of the direct method with a large time-step deviates from the reference solution. Both methods reduced the computational time by about 3.5 times for the IQM and about 6.5 times for the PCQM.

The IQM and PCQM solvers of flowing fuel have been tested for step velocity change without thermal feedback. Figure B.5 shows the results of the power evolution with time after step fuel velocity reduction which acts as positive reactivity insertion due to the reduced losses of the delayed neutrons, and Fig. B.6 shows the results for fuel salt step velocity increase which acts as negative reactivity insertion due to the increased losses of the delayed neutrons.

For step reduction of the fuel salt velocity as shown in Fig. B.5, the IQM and PCQM solvers of flowing fuel are able to reproduce the reference solution accurately which was obtained from the direct solution of the flowing fuel equations with a small time-step size of 10.0 ms. The results of the IQM and PCQM were obtained with a time-step size ranging from 0.05 s to 0.5 s and compared to the direct solution with the same step size. Similar behavior was observed as the step reactivity insertion transients, the IQM and PCQM solutions can reproduce the reference solution accurately with a larger time step size, while the solution of the direct method with a large time-step deviates from the reference solution. Both methods reduced the computational time by about 7 times for the IQM and about 13 times for the PCQM.

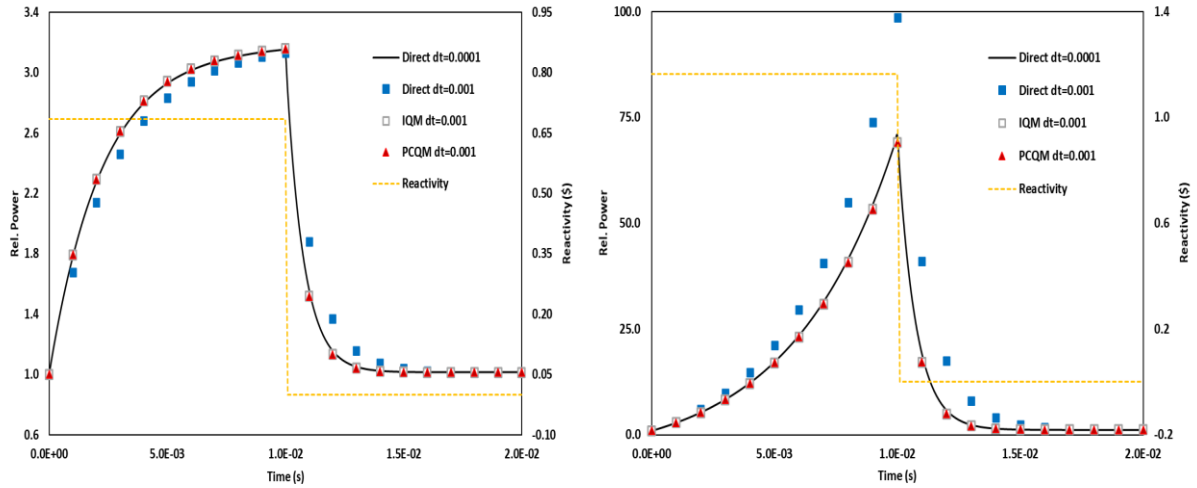


Figure B.3. Power evolution during step reactivity perturbation of 100 pcm (0.66 $) (left) and of 170 pcm (1.15 $) (right) for flowing fuel.

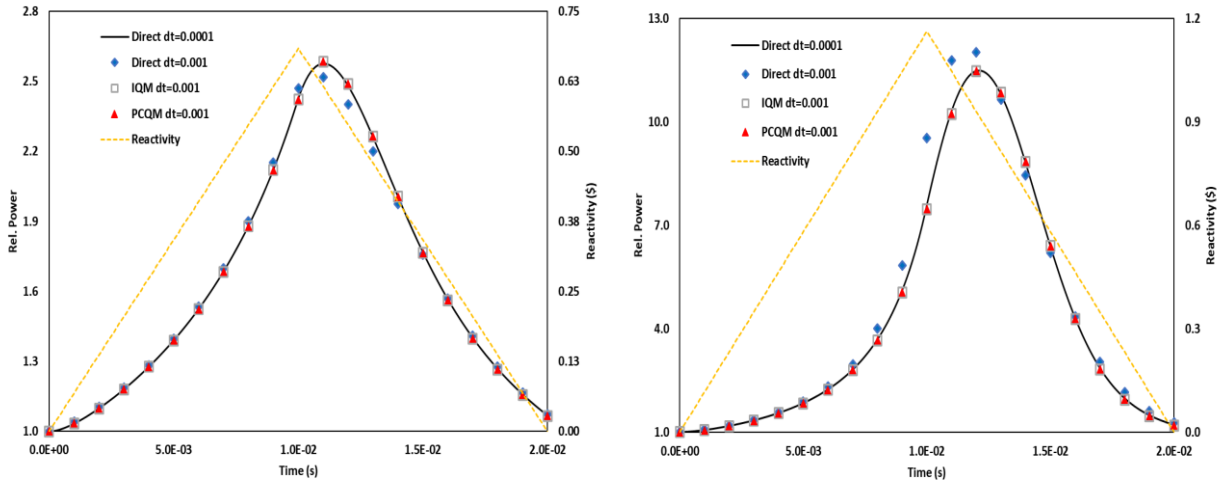


Figure B.4. Power evolution during ramp reactivity perturbation of 100 pcm (0.66 $) (left) and of 170 pcm (1.15 $) (right) for flowing fuel.

For step increment of the fuel salt velocity, as shown in Fig. B.6, the IQM and PCQM solvers of flowing fuel are able to reproduce the reference solution for the first 1.0 s to 2.0 s, however as the undecayed fuel starts flowing back into the reactor core, all the IQM, PCQM, and direct solutions with a large time-step size of 0.1 s deviates from the reference solution which was obtained with a small time-step size of 10.0 ms. This is because of differences in the calculated inlet boundary condition with larger time-step size. Despite the difference being so large, the quasi-static solvers were not used to perform the transient calculations for flowing fuel. The findings and the derivation of the methods are included in this Appendix for documentation.

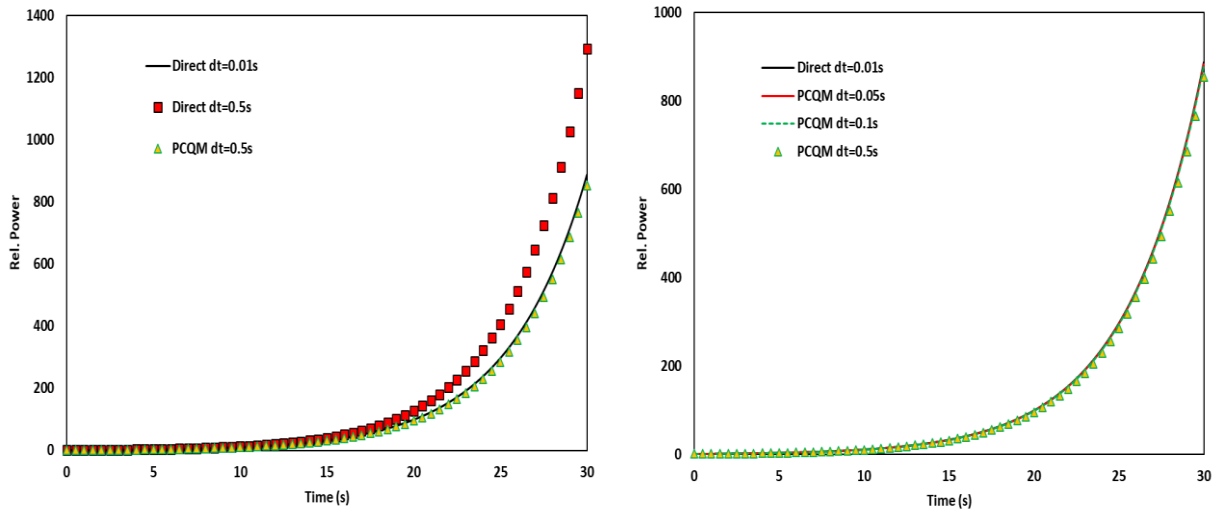


Figure B.5. Power evolution during step velocity perturbation (decreased) for flowing fuel for different time step size with direct solution (left) and PCQM (right).

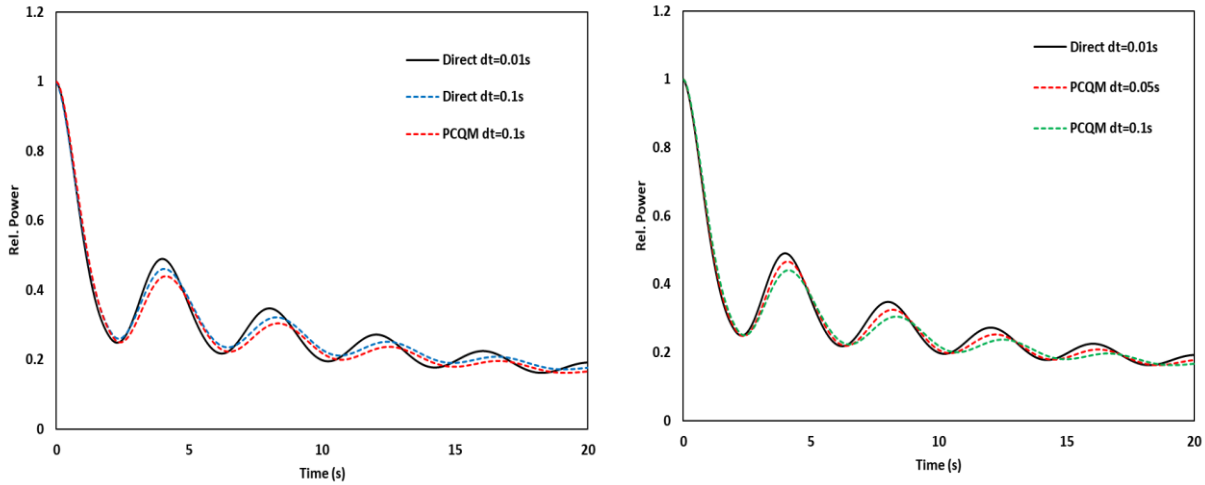


Figure B.6. Power evolution during step velocity perturbation (increased) for flowing fuel for different time step size with direct solution (left) and PCQM (right).

Appendix C. Specifications of MSFR Benchmark

Table C.1 provides the main characteristics of the MSFR core and the thermo-physical properties for the fuel salt obtained from Ref. [12, 13]. The fuel salt in its initial composition has a fissile element of either ^{233}U or the transuranic (TRU) elements produced by PWRs. The ^{233}U fuel salt is composed of LiF-ThF_4-$^{233}UF_4$, and the TRU fuel salt is composed of LiF-ThF_4-$(TRU)F_3$ with a TRU mixture of Pu (87.5%), Np (6.3%), Am (5.3%), and Cm (0.9%). The initial fuel compositions of both options are provided in Table C.2, and the isotopic compositions of the TRU element are listed in Table C.3.

Table C.1. Characteristics of Molten Salt Fast Reactor.

Thermal/Electric Power	3,000 MWth/1,300 MWe
Fuel Composition	LiF-ThF_4-$^{233}UF_4$, LiF-ThF_4-$(TRU)F_3$
Fuel Inlet/Outlet Temperature	650 ºC/750 ºC
Core Height/Core Radius	2.255 m/1.1275 m
Fuel Salt Volume	18 m^3
Total Fuel Salt Transit Time	4.0 s
Flow Rate	4.5 m^3/s
Density [kg/m^3]	$\rho = 4094 - 0.882 \times (T[\mathrm{K}] - 1008)$
Kinematic Viscosity [m^2/s]	$\nu = 5.54 \times 10^{-8} e^{3689/T[\mathrm{K}]}$
Dynamic Viscosity [Pa·s]	$\mu = \left[2.268 - 4.886 \times 10^{-4} (T[\mathrm{K}] - 1008)\right] \times 10^{-4} e^{3689/T[\mathrm{K}]}$
Thermal Conductivity [W/m·K]	$k = 0.928 + 8.397 \times 10^{-5} T[\mathrm{K}]$
Specific Heat [J/kg·K]	$c_p = -1111 + 2.78T[\mathrm{K}]$

Table C.2. Initial Composition of Fuel Salt of MSFR.

^{233}U started		TRU started	
Element	Mole fraction (%)	Element	Mole fraction (%)
Th	19.985	Th	16.068
^{233}U	2.515	Pu	5.628
		Np	0.405
		Am	0.341
		Cm	0.058

The MSFR utilizes 16 heat exchangers and 16 pumps. Each heat exchanger has the capability of extracting 187 MW during normal operation, and each pump provides a flow rate of about 0.28 m^3/s to maintain the temperature rise in the core. The heat exchanger considered in the following simulations consists of about 8000 tubes with a length of 2.0 m and an inner diameter of 0.4 cm, with a fuel salt volume of 0.203 m^3. The inlet and outlet temperatures of the heat exchanger are 750 °C and 650 °C, respectively. The MSFR design includes an intermediate circuit that is used to separate the radioactive fuel salt from the energy conversion system that uses FLiNaK as a coolant with a temperature of 550 °C. The heat exchanger is assumed to be of shell and tube type as the MSRE and MSBR designs developed at the ORNL. The temperature range of the MSFR design is between 838 K (the salt freezing point) and 1600 K (the melting point of the nickel alloy of the core structures).

Table C.3. TRU-element Fuel Composition.

Isotope	Mole fraction (%)	Isotope	Mole fraction (%)
^{238}Pu	2.7	^{237}Np	6.3
^{239}Pu	45.9	^{244}Cm	0.8
^{240}Pu	21.5	^{245}Cm	0.1
^{241}Pu	10.7	^{241}Am	3.4
^{242}Pu	6.7	^{243}Am	1.9

Appendix D. Specifications of MSRE Experiment

The operation of the MSRE was started in 1965 and continued until 1968, and it was fueled with ^{235}U fuel (33% enriched) with salt composition of LiF-BeF_2-ZrF_4-UF_4 (65.0%-29.17%-5.0%-0.83%). It was designed to be operated between 908 K and 936 K, with a temperature rise of the salt by 28 K and the core average temperature of 922 K. In 1968, the fuel was replaced with ^{233}U fuel (91% enriched) with a slightly different molar composition (64.5%-30.18%-5.19%-0.13%). Table D.1 provides the isotopic concentration of the MSRE fuel salts [75, 76].

Table D.1. Molar Composition of Fuel Salts.

	^{235}U Core	^{233}U Core
Molar composition	LiF-BeF_2-ZrF_4-UF_4 (65.0%-29.17%-5.0%-0.83%)	LiF-BeF_2-ZrF_4-UF_4 (64.5%-30.18%-5.19%-0.13%)
Enrichment	33%	91.5%
Isotope	Atom Fraction (%)	Atom Fraction (%)
Li	2.62E-01	2.62E-01
Be	1.19E-01	1.23E-01
Zr	2.06E-02	2.11E-02
F	5.95E-01	5.94E-01
^{233}U		4.98E-04
^{234}U		3.12E-05
^{235}U	1.09E-03	3.78E-06
^{238}U	2.22E-03	1.07E-06

The MSRE reactor vessel has an inner diameter of 147.32 cm with a wall thickness of 2.54 cm. The core container is inside the vessel and has an inner diameter of 140.97 cm and a thickness of 0.635 cm. The main function of the container is to support the graphite in the reactor. All salt-containing components were made from nickel-alloy INOR-8 including the reactor vessel. The MSRE lattice is made of vertical graphite stringers with a 5.08 cm × 5.08 cm cross section and 166.37 cm long. The fuel salt flows through a rectangular channel (3.048 cm × 1.016 cm with round corners of radius 0.508 cm) in the sides of the stringers with a total of 1140 fuel channels. The dimensions of the channel were chosen to prevent small objects from blocking the channels

and to obtain a nearly optimum fuel-to-graphite ratio in the core [78]. Fig. D.1 shows a schematic representation of the MSRE reactor vessel.

The reactor vessel is installed in a thermal shield that reduces the radiation damage to the reactor containment vessel, serves as a biological shielding, and provides support for the reactor vessel. The shield is a water-cooled, steel- and water-filled container surrounding the reactor vessel. The thermal shield is 40.64 cm thick and contains approximately 50% steel and 50% water. The inside of the thermal shield is lined with high temperature insulation of 15.24 cm thickness. Figure D.2 shows a representation of the outer vessel regions and the effective core region and the control rod movement level, and Table D.2 provides the material composition of the shield, insulator, and INOR [77, 78, 79].

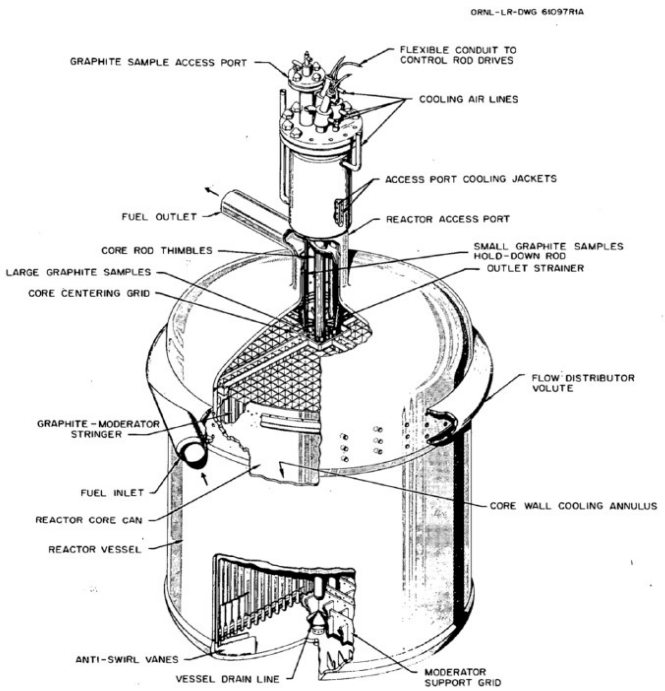


Figure D.1. Schematic diagram of MSRE reactor vessel [78, 79].

The primary and secondary system's layout of the MSRE is shown in Fig. D.3. The heated fuel salt from the core outlet was cooled through a shell-and-tube heat exchanger and recirculated by a centrifugal pump [77]. In the heat exchanger, the primary fuel salt flows through the shell side and the secondary coolant salt flows through the tube side. The coolant salt for the secondary system was molten LiF-BeF_2 salt. The heat from the primary loop was removed to the secondary side through the heat exchanger and the radiator to the atmosphere [78].

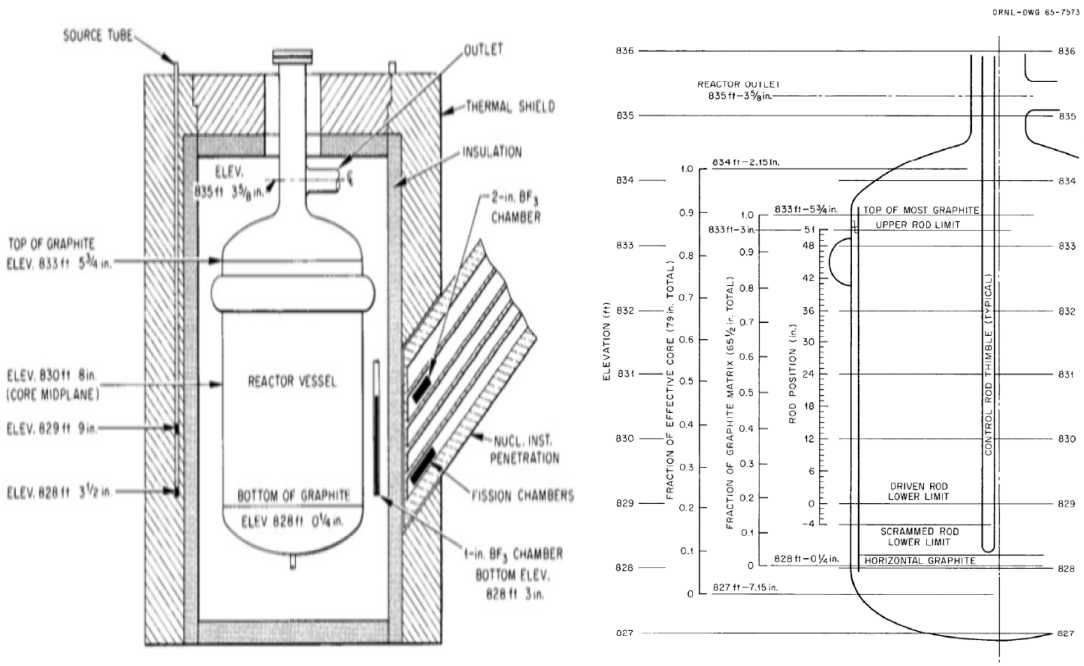


Figure D.2. Representation of outer regions of MSRE core (left) and control rod level and effective core region of MSRE (right) [77].

Table D.2. Material Compositions of MSRE [78].

Thermal Shield (4.42 g/cm^3)		Insulation	
Component	Atom Fraction	Component	Atom Fraction
Fe	4.47E-01	Ca	1.05E-02
K	1.49E-04	Fe	4.50E-02
C	2.10E-02	Si	1.20E-01
B-10	3.61E-06	Al-27	4.50E-02
B-11	1.45E-05	H-1	3.00E-01
N-14	9.48E-05	O-16	4.80E-01
H-1	3.54E-01		
O-16	1.78E-01		
INOR-8 (8.775 g/cm^3)		Poison (5.873 g/cm^3)	
Component	Atom Fraction	Component	Atom Fraction
Ni	7.10E-01	Gd	2.80E-01
Mo	1.70E-01	Al-27	1.20E-01
Cr	7.00E-02	O-16	6.00E-01
Fe	5.00E-02		

The MSRE was designed to be operated with three control rods made from gadolinium in the form of Gd_2O_3-Al_2O_3 (70-30 wt.%) ceramic, with a density of 5.78 g/cm^3. Each control rod is made of 38 elements for a total length of the poison section of 150.876 cm, with each element has

a 2.8956 cm outer diameter, 2.0066 cm inner diameter, and 3.96748 cm height. Figure D.4 shows a representation of the control rod lattice arrangement and element geometry [78].

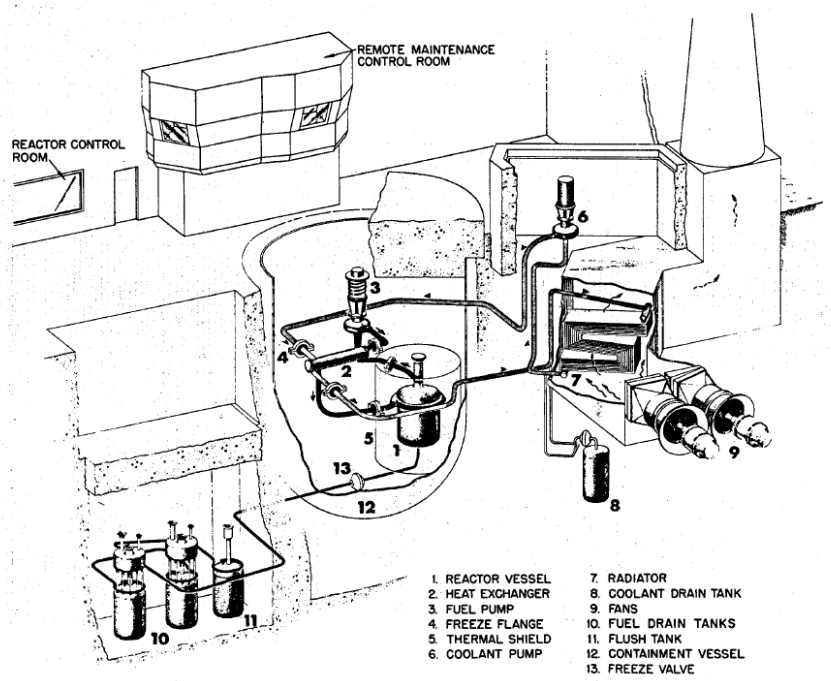


Figure D.3. Layout of MSRE primary and secondary systems [78].

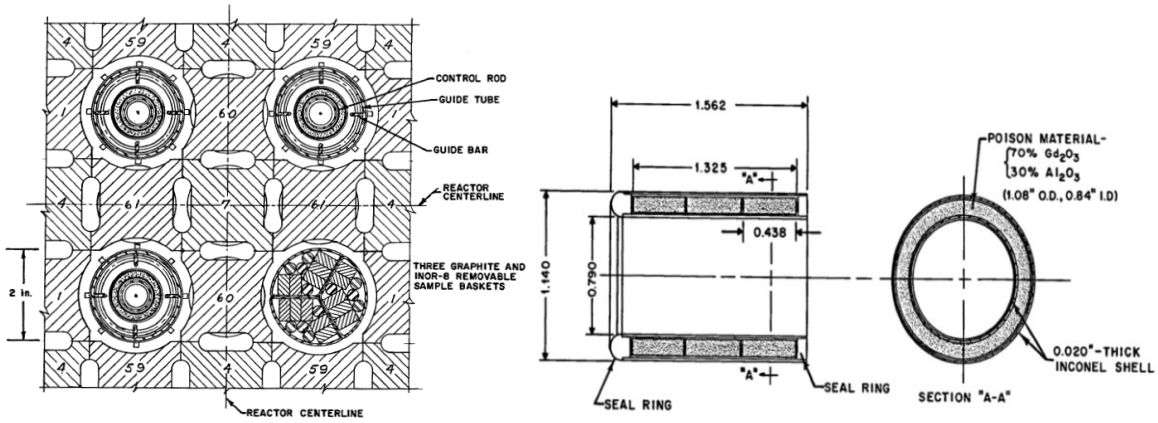


Figure D.4. Lattice arrangement of control rods (left) geometry and composition of the control rod element (right) (units in inch) [78].

Appendix E. Cross Sections Generation Procedure for MSRE Analysis

To analyze the MOST benchmark problem with PROTEUS-NODAL, two core models were developed. One is a Cartesian geometry model, and the other is a cylindrical-z (R-Z) model. In the Cartesian geometry model, each graphite stringer and four half channels of fuel salt are homogenized into a square node with a side length of 5.08 cm. The Cartesian geometry model necessitates an approximate representation of the circular outer fuel region by a jagged shape. This approximate representation of the radial boundary involves artificial relocation of fuel and thus affects the neutron leakage. An R-Z model was developed in order to represent the outer core region of the core accurately, but this involves more homogenization in the inner core region. In the axial direction, the core was divided into 22 nodes. The vacuum (zero incoming current) boundary condition is applied to all the outer boundaries.

In order to evaluate the two PROTEUS-NODAL core models, an OpenMC core model was developed for comparison. Also, the OpenMC core model is used for generating region-dependent multigroup cross sections by considering the spectral variation in the core, and for determining the fraction of heat generated in the graphite moderator due to gamma heating and neutron scattering. Fig. E.1 compares the radial configurations of the OpenMC model and the PROTEUS-NODAL core models in Cartesian and cylindrical geometries.

E.1. Energy Group Structures

To determine an adequate number of energy groups and group structure, the multigroup structures previously used for graphite moderated gas cooled reactors [80, 81] were examined with additional considerations to the neutron spectrum in MSRE and the broad resonances of salt nuclides. Five subsets of the WIMS-D 69 group structure [80], as provided in Table. E.1, were investigated. Fig. E.2 shows the neutron energy spectrum in the fuel salt and the selected five energy group structures. Fig. E.3 shows the total cross sections of the fuel salt nuclides along with the selected energy group structures.

The two core models and five energy group structures were tested by comparing the eigenvalue

and the leakage fraction obtained from PROTEUS-NODAL calculations for the steady state MOST benchmark problem in Cartesian and R-Z geometries with the reference OpenMC results. The multigroup cross sections were prepared from the OpenMC simulations using the core model and the ENDF/B-VII.1 data [82].

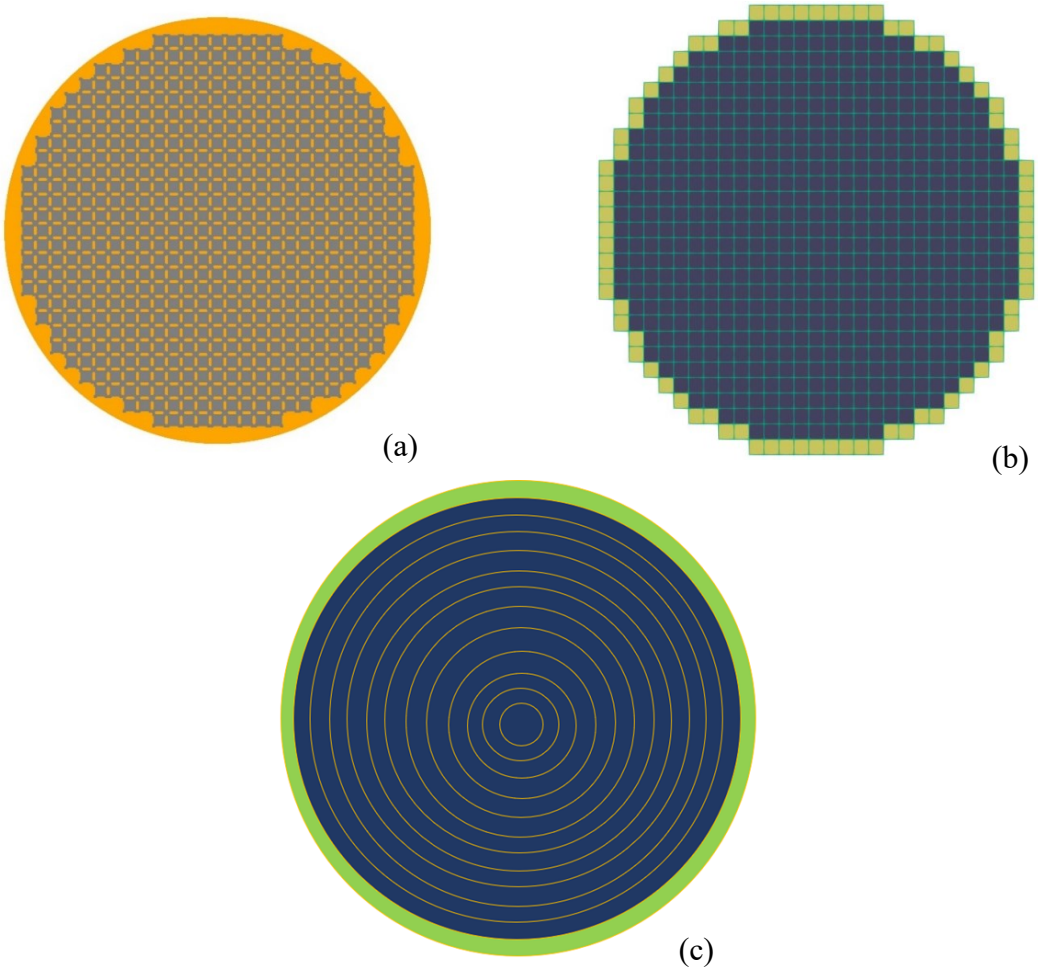


Figure E.1. Radial configurations of MOST benchmark in (a) OpenMC model, (b) Cartesian model, and (c) R-Z model of PROTEUS-NODAL.

Table E.2 compares the eigenvalues and leakage fractions of PROTEUS-NODAL calculations with different group structures. It can be clearly seen that except for the four-group case, the Cartesian geometry model significantly overestimates the neutron leakage and hence leads to a highly underestimated eigenvalue. For the four-energy group case, the leakage is highly underestimated, resulting in a highly overestimated eigenvalue. These results suggest that the artificial change in the surface area and the relocation of fuel salt in the outer fuel region due to the approximate representation of the cylindrical boundary introduces a significant change in the

neutron leakage and subsequently in the eigenvalue. In order to reduce these errors due to the boundary perturbation, a large number of radial nodes need to be used, which increases the computational time significantly.

Table E.1. Selected Energy Group Structures for MSRE Analysis.

16-Groups	14-Groups	12-Groups	8-Groups	4-Groups
2.000E+07	2.000E+07	2.000E+07	2.000E+07	2.000E+07
3.679E+06	3.679E+06	3.679E+06		
1.353E+06	1.353E+06	1.353E+06	1.353E+06	
5.000E+05	5.000E+05	5.000E+05		
6.734E+04	6.734E+04	6.734E+04	6.734E+04	
9.118E+03	9.118E+03	9.118E+03	9.118E+03	9.118E+03
1.487E+02	1.487E+02	1.487E+02	1.487E+02	
4.000E+00	4.000E+00	4.000E+00	4.000E+00	
1.300E+00	1.300E+00			
6.250E-01		6.250E-01	6.250E-01	6.250E-01
4.000E-01	4.000E-01	4.000E-01		
2.500E-01	2.500E-01			
1.800E-01				
1.400E-01	1.400E-01			
8.000E-02	8.000E-02	8.000E-02	8.000E-02	8.000E-02
4.200E-02	4.200E-02	4.200E-02		
1.000E-03	1.000E-03	1.000E-03	1.000E-03	1.000E-03

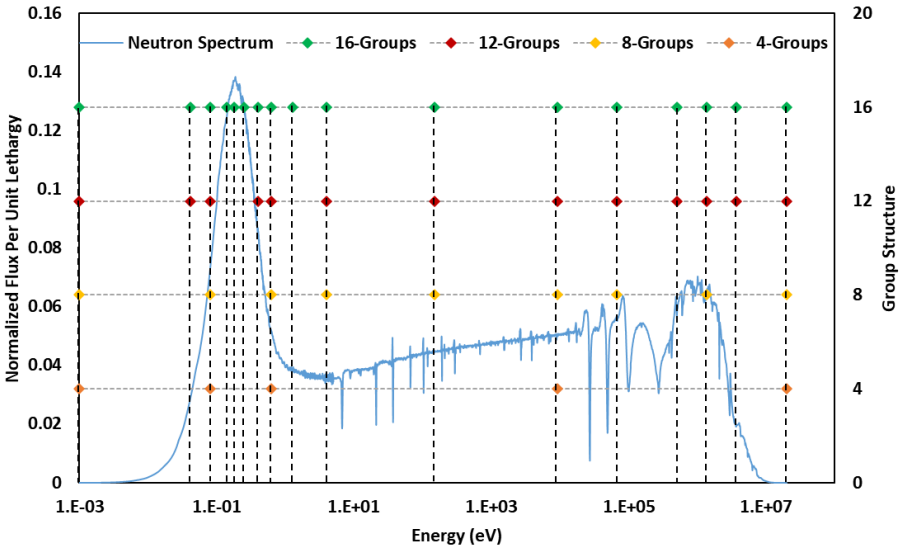


Figure E.2. Neutron spectrum in fuel salt and selected energy group structures.

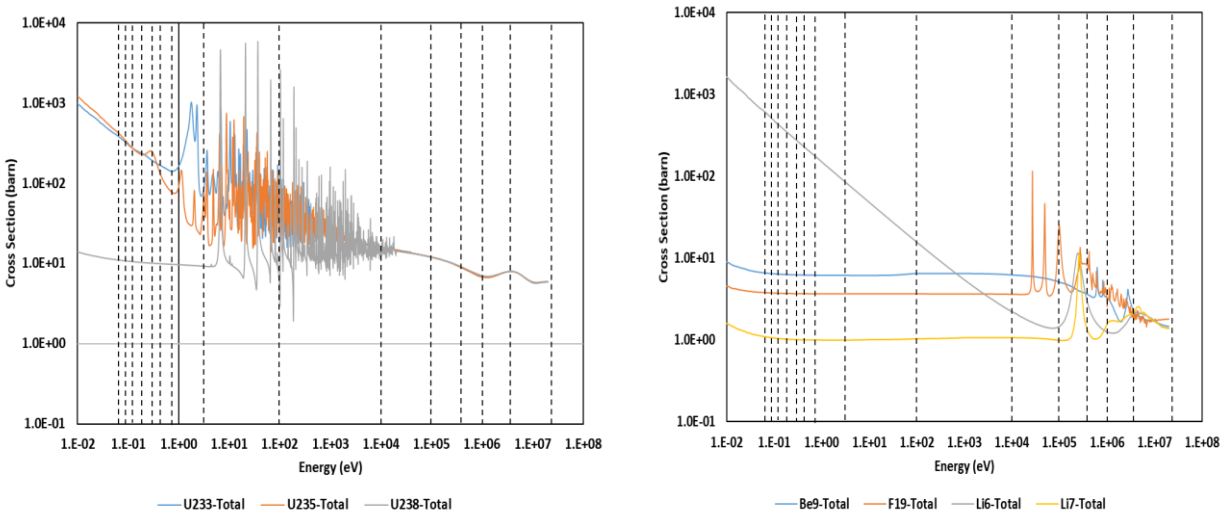


Figure E.3. Total cross sections of nuclides in MSRE fuel salt.

Table E.2. Comparison of Eigenvalues and Leakage Fractions of PROTEUS-NODAL Obtained with Different Core Models and Group Structures.

Code	Core Model	No. of Groups	Multiplication Factor	Diff. (pcm)	Leakage Fraction	Diff. (%)
OpenMC	-		1.06252 ± 0.00011	-	0.3362	-
PROTEUS-NODAL	Cartesian	16	1.05573	-679.2	0.3402	1.198
		14	1.05567	-685.5	0.3402	1.201
		12	1.05562	-689.7	0.3402	1.202
		8	1.05933	-319.1	0.3380	0.533
		4	1.07045	793.4	0.3313	-1.446
	R-Z	16	1.06232	-20.4	0.3362	0.007
		14	1.06226	-26.5	0.3362	0.015
		12	1.06220	-32.0	0.3363	0.024
		8	1.06575	323.1	0.3341	-0.617
		4	1.07603	1350.6	0.3279	-2.483

On the other hand, as shown in Fig. E.4, the eigenvalue and the leakage fraction of PROTEUS-NODAL solutions obtained with the R-Z model converge monotonically to the reference OpenMC results as the number of energy groups increases. It can be clearly seen that the leakage fraction and eigenvalue almost converge to the OpenMC solutions when the number of energy groups is increased to 12. With the 16-group structure, the eigenvalue and leakage fraction errors are only 20.4 pcm and 0.01%, respectively. Based on these results, the R-Z core model and the 16-group structure were selected for subsequent analyses of the MOST benchmark problems and the MSRE experiments.

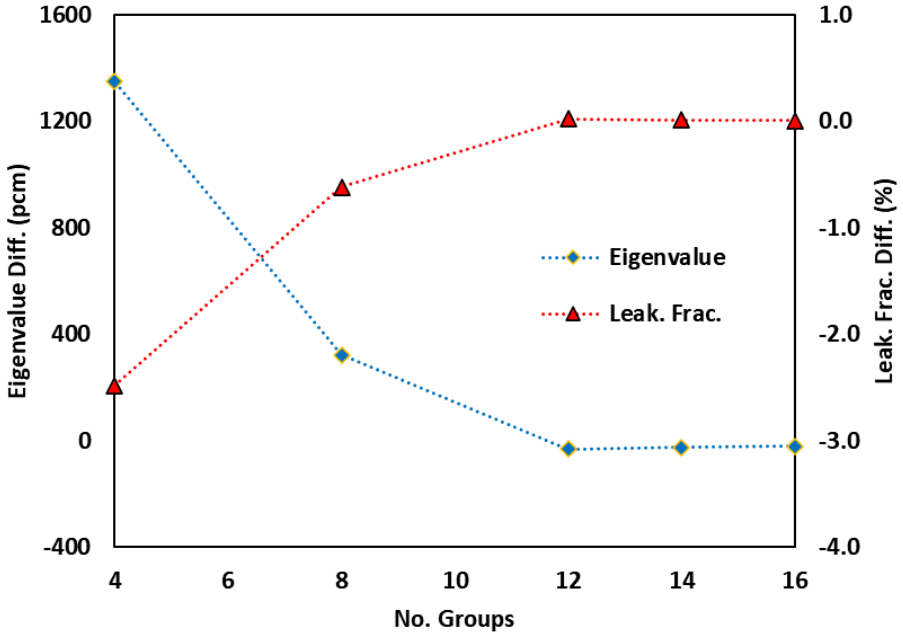


Figure E.4. Trends of eigenvalue and leakage fraction errors of PROTEUS-NODAL obtained with R-Z geometry model.

E.2. Multigroup Cross Section Generation Models

Different OpenMC models for generating multigroup cross sections were investigated, including unit cell, simplified 2-D core, and 3-D full core models. Figure E.5 shows these OpenMC models used for multigroup cross section generation. The left two figures show the unit cell models with reflective boundary conditions. They were defined separately for the graphite-moderated inner fuel region and the unmoderated outer fuel region. The middle figure shows the radial layout of the 2-D and 3-D core models, and the right figure shows the axial layout of the 3-D core model. The 2-D core model is an axially extruded core of the radial layout with the vacuum boundary condition at the radial boundary and the reflective boundary conditions at the top and bottom boundaries. The core was divided into 3 radial regions as marked in the middle figure, and region-dependent cross sections were generated for each region. In the 3-D core model, the vacuum boundary condition was applied to all outer boundaries. The core was divided into 11 regions as marked in the right figure, and region-dependent cross sections were generated for each region.

Using these cross section generation models, different cross section sets were generated for the steady state MOST benchmark problem in the selected 16-group structure. With each cross section set, PROTEUS-NODAL calculations were performed for the steady state MOST benchmark problem in Cartesian and R-Z geometries and the resulting eigenvalue, leakage fraction, main

reaction rates, and axial and radial flux distributions were compared with the reference OpenMC solutions obtained with the as-built full core model.

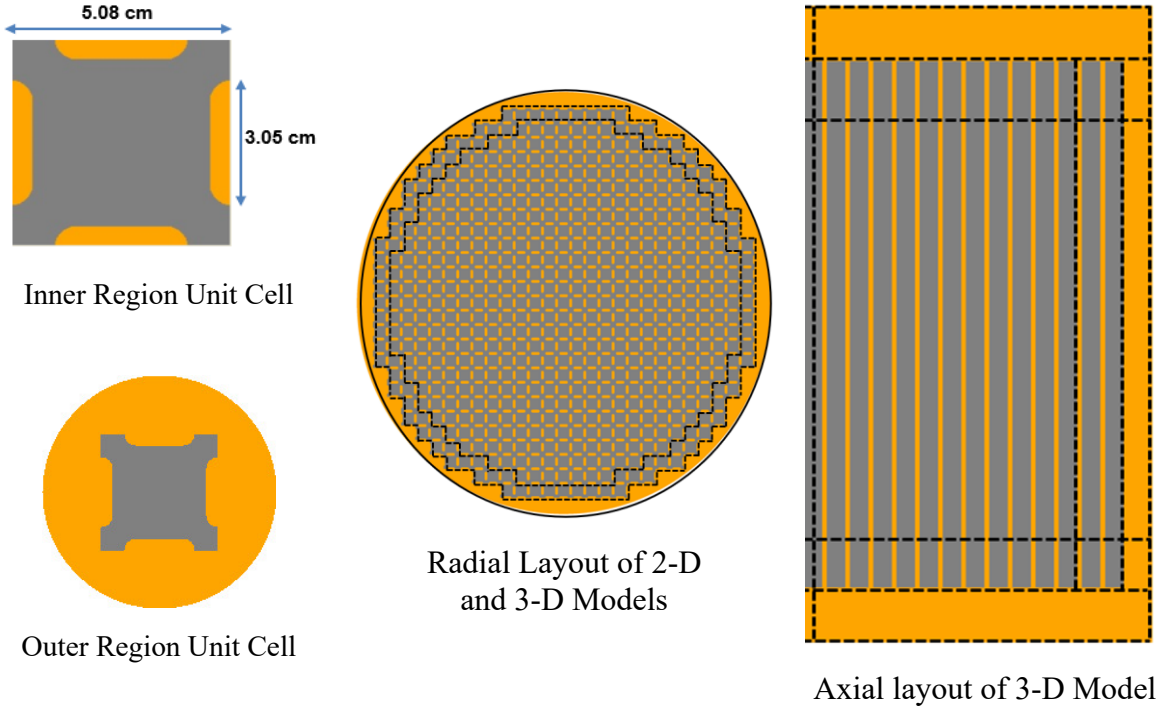


Figure E.5. OpenMC models for multigroup cross section generation.

Table E.3 compares the eigenvalues and leakage fractions obtained with different core and cross section generation models. It is observed that with the Cartesian geometry core model, the cross section set obtained with unit-cell models yields better results than those obtained with the 2-D and 3-D models. To examine these unexpected results, a reactivity error analysis was performed for the Cartesian geometry model. Table E.4 presents the reactivity errors due to reaction and leakage rate errors in three energy ranges. The results show large error cancelations in fission, absorption, and leakage rates among different energy ranges for the unit cell and 2-D models, which cannot capture the neutron spectral changes near the core periphery accurately. It appears that the cross section errors are canceled out by the boundary perturbation errors. On the other hand, in the case of the 3-D model, no significant error cancellation is observed since accurate region-dependent cross sections are prepared and thus the boundary perturbation becomes the dominant error source. As a result, the 3-D model yields larger errors than the unit cell and 2-D models.

Table E.3. Comparison of Eigenvalues and Leakage Fractions Obtained with Different Core and Cross Section Models.

Code	Core Model	XS Model	Multiplication Factor	Diff. (pcm)	Leakage Fraction	Diff. (%)
OpenMC	-	-	1.06252 ± 0.00011	-	0.3362	-
PROTEUS-NODAL	Cartesian	3-D	1.05573	-679.2	0.3402	1.198
		2-D	1.05828	-424.0	0.3392	0.889
		Unit Cell	1.06329	76.9	0.3367	0.149
	R-Z	3-D	1.06232	-20.4	0.3362	0.007
		2-D	1.06487	235.3	0.3352	-0.294
		Unit Cell	1.06993	741.3	0.3327	-1.035

Table E.4. Reactivity Errors (pcm) in Cartesian Geometry Core Due to Reaction and Leakage Rate Errors in Three Energy Ranges.

XS Model	Energy Range	Reaction/Leakage				
		Fission	Absorption	Leakage	Scattering	Sum
3-D	Fast	0.5	0.5	-304.7	4.6	-299.0
	Resonance	-21.0	-37.1	-116.6	0.0	-174.7
	Thermal	20.5	12.6	-238.5	0.0	-205.4
	Total	0.0	-24.0	-659.7	4.6	-679.1
2-D	Fast	1.8	2.5	-358.9	-3.0	-357.6
	Resonance	190.5	138.0	202.2	0.0	530.7
	Thermal	-192.3	-98.2	-306.8	0.0	-597.3
	Total	0.0	42.3	-463.5	-3.0	-424.2
Unit Cell	Fast	7.8	24.9	-179.3	-27.3	-174.0
	Resonance	238.4	234.9	401.2	0.0	874.5
	Thermal	-246.2	-124.8	-252.9	0.0	-623.9
	Total	0.0	135.0	-31.1	-27.3	76.6

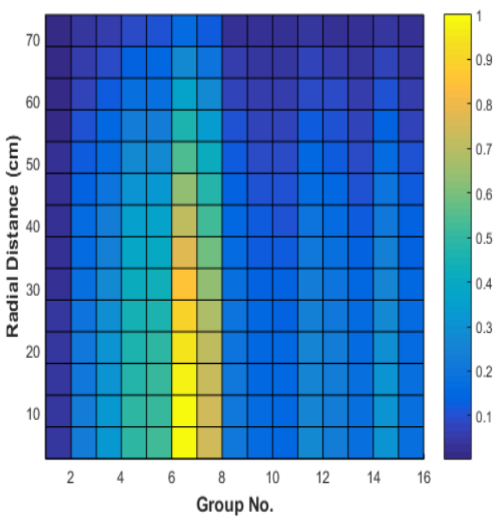
In the R-Z core model, the cylindrical core boundary is represented accurately. Thus, the eigenvalue errors in Table E.3 are mainly due to the cross section error. The eigenvalue and leakage fraction obtained with the cross sections generated with the 3-D model agree well with the reference OpenMC results as expected. On the other hand, the cross sections prepared with the unit cell and 2-D models yield significantly underestimated leakage fraction and hence significantly overestimated eigenvalue. Table E.5 presents the results of reactivity error analysis.

It can be seen that the cross sections generated from the unit cell and 2-D models introduce significant errors in fission, absorption, and leakage terms. When the same cross section set prepared with the 3-D model is used, the Cartesian and the R-Z geometry results show almost the same errors in the fission, absorption, and scattering reaction components, but very different errors in the leakage component. These results also indicate that the boundary perturbation in the Cartesian geometry core model introduces a significant error in neutron leakage.

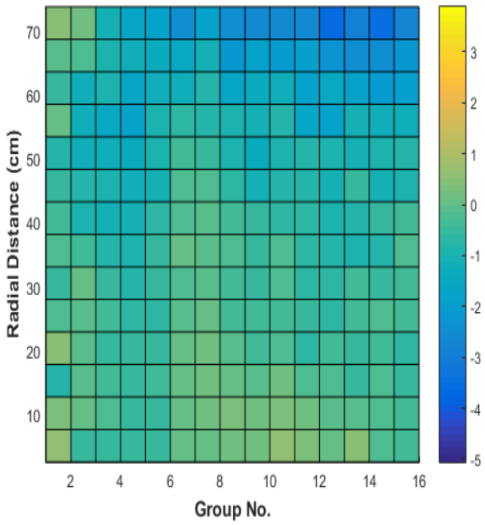
Table E.5. Reactivity Errors (pcm) in R-Z Geometry Core Due to Reaction and Leakage Rate Errors in Three Energy Ranges.

XS Model	Energy Range	Reaction/Leakage				
		Fission	Absorption	Leakage	Scattering	Sum
3-D	Fast	-0.1	0.0	-132.7	6.0	-126.8
	Resonance	-36.3	-34.0	67.3	0.0	-3.0
	Thermal	36.4	24.4	49.4	0.0	110.2
	Total	0.0	-9.6	-16.0	6.0	-19.6
2-D	Fast	1.2	2.0	-187.7	-2.1	-186.6
	Resonance	177.2	142.3	388.6	0.0	708.1
	Thermal	-178.3	-87.3	-19.9	0.0	-285.5
	Total	0.0	57.0	181.0	-2.1	235.9
Unit Cell	Fast	7.1	24.3	-9.3	-26.1	-3.9
	Resonance	233.0	245.7	601.3	0.0	1079.9
	Thermal	-240.1	-117.9	23.5	0.0	-334.6
	Total	0.0	152.0	615.4	-26.1	741.4

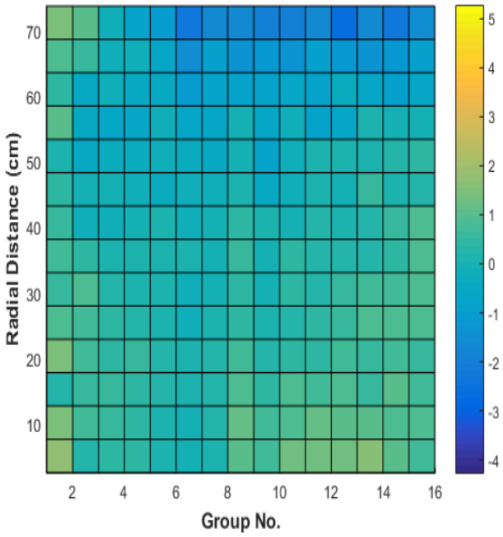
Figures E.6 and E.7 respectively present the radial profiles at the core mid-plane and the axial profiles at the core center of the reference OpenMC flux and the PROTEUS-NODAL flux solutions in R-Z geometry obtained with the three different cross section sets. The PROTEUS-NODAL flux solution obtained with the cross sections prepared with the 3-D model shows good agreement with the reference OpenMC flux distribution except for the radial periphery and top of the core. The cross section sets prepared with the unit cell and 2-D models introduce significantly larger errors in the flux solution than that prepared with the 3-D model. In particular, the axial profile shows large errors near the top of the core since the unit cell and 2-D model cannot capture the spectral transition effect in the unmoderated top region.



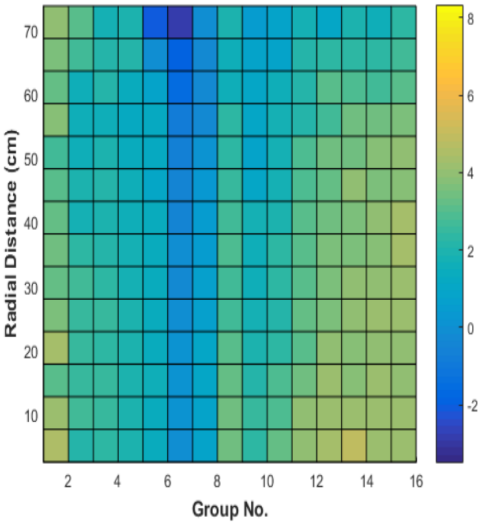
OpenMC Radial Flux (Rel. Unit)



3-D XS Model / Rel. Diff. (%)



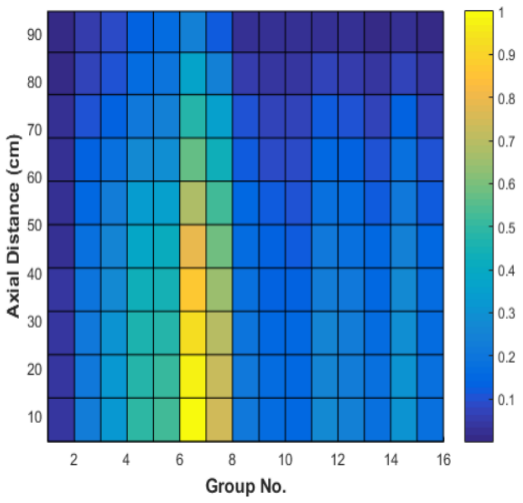
2-D XS Model / Rel. Diff. (%)



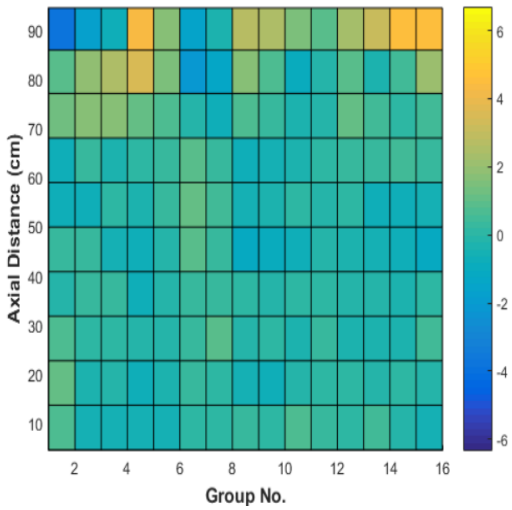
Unit Cell XS Model / Rel. Diff. (%)

Figure E.6. Comparison of radial profile at core mid-plane of OpenMC flux and PROTEUS-NODAL solutions in R-Z geometry obtained with three cross section sets.

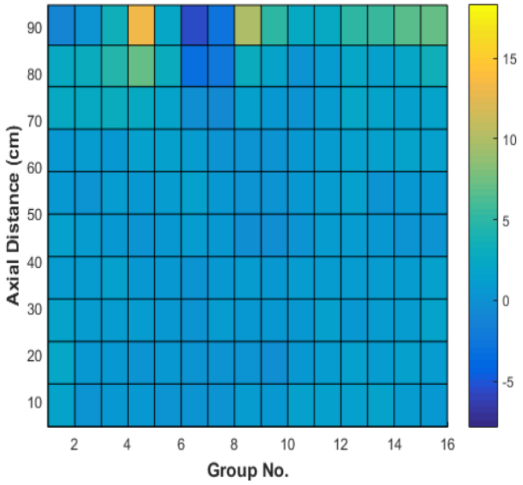
Based on the above results, the 3-D model was selected for multigroup cross section generation. Region-dependent 16-group cross sections were generated as a function of the fuel salt and graphite temperatures. The resulting cross sections were written into the isotopic cross section dataset ISOTXS and the delayed neutron cross section dataset DLAYXS using a modified version of the utility code GenISOTXS [68] that can provide the DLAYXS cross section dataset.



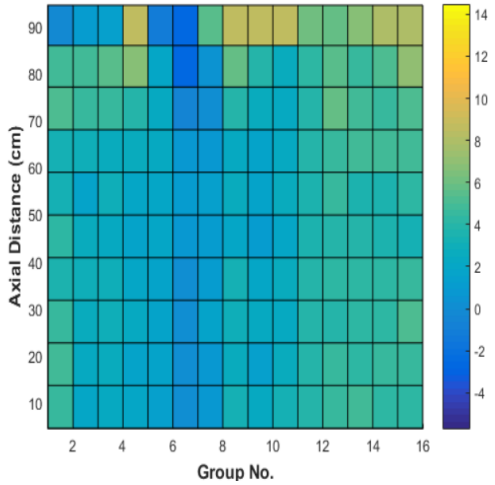
OpenMC Axial Flux (Rel. Unit)



3-D XS Model / Rel. Diff. (%)



2-D XS Model / Rel. Diff. (%)



Unit Cell XS Model / Rel. Diff. (%)

Figure E.7. Comparison of axial profile at core center of OpenMC flux and PROTEUS-NODAL solutions in R-Z geometry obtained with three cross section sets.

Appendix F. Multigroup Albedo Response Matrix for MSRE Analysis

Table F.1. 16-Group Albedo Response Matrix on Outer Surface of MSRE Reactor Vessel.

$\beta_{gg'}$	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16
G1	2.77E-02	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
G2	3.66E-02	5.92E-02	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
G3	3.15E-02	4.25E-02	6.92E-02	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
G4	2.94E-02	3.87E-02	5.51E-02	9.44E-02	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
G5	1.47E-02	2.06E-02	2.99E-02	4.81E-02	7.04E-02	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
G6	2.30E-02	3.31E-02	4.92E-02	8.22E-02	1.29E-01	1.47E-01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
G7	1.72E-02	2.44E-02	3.56E-02	5.68E-02	8.89E-02	1.18E-01	1.40E-01	4.67E-06	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
G8	4.91E-03	6.89E-03	9.91E-03	1.52E-02	2.26E-02	2.98E-02	4.36E-02	5.63E-02	1.33E-04	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
G9	3.09E-03	4.34E-03	6.15E-03	9.47E-03	1.39E-02	1.78E-02	2.65E-02	3.21E-02	4.41E-02	8.40E-04	4.33E-06	6.67E-07	3.33E-07	1.33E-06	1.00E-06	0.00E+00
G10	1.84E-03	2.64E-03	3.68E-03	5.59E-03	8.15E-03	1.04E-02	1.55E-02	1.85E-02	2.30E-02	3.35E-02	1.80E-03	5.90E-05	2.63E-05	2.20E-05	1.83E-05	1.57E-05
G11	2.01E-03	2.79E-03	3.95E-03	5.89E-03	8.60E-03	1.08E-02	1.61E-02	2.05E-02	2.08E-02	2.89E-02	3.56E-02	5.37E-03	1.06E-03	5.68E-04	4.35E-04	3.92E-04
G12	1.59E-03	2.20E-03	3.05E-03	4.58E-03	6.56E-03	8.22E-03	1.20E-02	1.60E-02	1.67E-02	1.72E-02	2.39E-02	3.02E-02	1.07E-02	3.82E-03	2.39E-03	2.20E-03
G13	1.49E-03	2.05E-03	2.92E-03	4.19E-03	5.90E-03	7.27E-03	1.06E-02	1.45E-02	1.56E-02	1.60E-02	1.72E-02	2.46E-02	2.75E-02	1.18E-02	6.31E-03	5.68E-03
G14	4.79E-03	6.61E-03	9.08E-03	1.30E-02	1.79E-02	2.19E-02	3.06E-02	4.23E-02	4.64E-02	4.97E-02	5.19E-02	5.64E-02	6.70E-02	7.18E-02	4.74E-02	4.00E-02
G15	6.53E-03	9.07E-03	1.25E-02	1.75E-02	2.39E-02	2.85E-02	3.97E-02	5.45E-02	6.10E-02	6.61E-02	7.05E-02	7.50E-02	7.89E-02	8.86E-02	1.04E-01	9.30E-02
G16	7.47E-03	1.05E-02	1.43E-02	2.02E-02	2.77E-02	3.30E-02	4.54E-02	6.28E-02	7.14E-02	7.83E-02	8.44E-02	9.07E-02	9.60E-02	1.05E-01	1.24E-01	1.72E-01

Table F.2. Diagonal Albedo Matrices of ^{235}U and ^{233}U Fuel Cores of MSRE.

Group No.	Upper Energy	^{235}U Core			^{233}U Core		
		Radial	bottom	top	Radial	bottom	top
1	2.00E+07	2.11E-02	6.77E-03	2.31E-02	2.09E-02	6.51E-03	2.22E-02
2	3.68E+06	6.32E-02	1.99E-02	7.10E-02	6.42E-02	1.92E-02	7.10E-02
3	1.35E+06	1.13E-01	4.24E-02	1.20E-01	1.14E-01	4.17E-02	1.21E-01
4	5.00E+05	1.81E-01	6.92E-02	1.87E-01	1.83E-01	6.74E-02	1.89E-01
5	6.73E+04	2.13E-01	6.64E-02	1.83E-01	2.15E-01	6.57E-02	1.83E-01
6	9.12E+03	3.16E-01	9.81E-02	2.87E-01	3.17E-01	9.69E-02	2.91E-01
7	1.49E+02	5.17E-01	1.63E-01	5.38E-01	4.78E-01	1.48E-01	4.99E-01
8	4.00E+00	5.74E-01	1.90E-01	6.70E-01	5.38E-01	1.74E-01	6.18E-01
9	1.30E+00	5.89E-01	1.90E-01	6.88E-01	5.90E-01	1.91E-01	7.13E-01
10	6.25E-01	5.53E-01	1.87E-01	6.97E-01	4.91E-01	1.68E-01	6.86E-01
11	4.00E-01	5.15E-01	1.87E-01	7.22E-01	3.91E-01	1.45E-01	6.31E-01
12	2.50E-01	5.77E-01	2.09E-01	7.90E-01	4.09E-01	1.51E-01	6.58E-01
13	1.80E-01	7.32E-01	2.55E-01	8.92E-01	5.26E-01	1.89E-01	7.64E-01
14	1.40E-01	1.17E+00	3.43E-01	1.12E+00	8.92E-01	2.77E-01	1.04E+00
15	8.00E-02	2.00E+00	4.54E-01	1.40E+00	1.68E+00	4.15E-01	1.39E+00
16	4.20E-02	3.17E+00	5.88E-01	1.66E+00	2.91E+00	5.71E-01	1.73E+00

Appendix G. Adaptive Time Stepping for Depletion Calculations

The adaptive time-stepping selection algorithm implemented in PROTEUS-NODAL for transient analysis was also utilized for depletion analysis to predict time step-size for depletion calculations of MSRs based on the control theory approach. The depletion the time-step size is predicted based on the estimated local error of the solution and maintained within a specified accuracy requirement or tolerance level using the control theory. The discrete-time integral (I), proportional-integral (PI), and proportional-integral-derivative (PID) controllers described in Chapter 5 can be utilized to predict the time-step size. The local error is estimated from the L_2 norm of the predicted $\phi(t_{n+1/2})$ and corrected $\phi(t_{n+1})$ solutions of the neutron flux as

$$\hat{R}_{1st}^{n+1} = \frac{\left\|\phi(t_{n+1}) - \phi(t_{n+1/2})\right\|_2}{\varepsilon \left\|\phi(t_{n+1})\right\|_2}. \tag{G.1}$$

At the beginning of the depletion calculations the initial time-step size, the solution tolerance level, and the gain values must be specified as described for transient analysis. After that, the I controller can be used to predict the step size of the next depletion step. Once the size of the first step is determined the procedure will continue until the end of the depletion time. Algorithm G.1 provides the adaptive time-stepping solution scheme of PROTEUS-NODAL for MSR depletion calculations.

The adaptive time-stepping selection algorithm of MSR depletion calculations was tested using the MSFR benchmark problem without considering fuel salt reprocessing, and the depletion calculations were performed for 350 days. The reference solution was obtained by performing depletion calculation with a small time-step size of 1 day, with a total of 350 steps. The I controller was used to predict the time-step size. To test the performance of the solution algorithm, 27 cases were studied by changing the initial step-size between 0.5 to 5.0 days, the tolerance level between 10^{-5} to 10^{-3}, and the integral gain value between 0.02 and 0.2. Table G.1 provides the results of the tests problems sorted based on the number of the time steps and the speedup of the depletion calculations. The results show that the adaptive time-stepping algorithm could speed up the

calculations by about 3 to 30 times while maintaining the solution accuracy with a 4.6 maximum difference in the eigenvalue results at the end of the calculations. Cases 11, 14, and 17 uses the same initial step size of 1.0 days and a gain value of 0.05 while the tolerance level is ranging from 10^{-5} to 10^{-3}. These cases show that results even with a larger tolerance level a 15 times speedup could be achieved while maintaining the same accuracy level.

Algorithm G.1. Adaptive time-stepping for MSR depletion calculations.

1. Apply algorithm 6.1 or 6.2 to calculate neutron flux ϕ^n, nuclide concentrations $\mathbf{N}^n$,and k_{eff} at the end of step.
2. From the predicted and corrected solutions calculate $\hat{R}^{n+1}$
3. If $\left(\left(\hat{R}^{n+1} > 1.0\right) \; and \; \left(h_n > \Delta t_{\min}\right)\right)$ then (Reject time step)

   $t = t - h_n, \quad n = n - 1, \quad Nrej = Nrej + 1$

   $h_n = \max\left(\min\left(0.5h_n, \left[\varepsilon / \hat{r}_{n+1}\right]h_n\right), \Delta t_{\min}\right)$

   $\left(\phi, C, N\right)^n = \left(\phi, C, N\right)^{n-1}$
4. Else (Calculate the next time step)

   $h_{n+1} = \left(\hat{R}^{n+1}\right)^{-(k_I + k_P + k_D)} \left(\hat{R}^n\right)^{k_P + 2k_D} \left(\hat{R}^{n-1}\right)^{-k_D} h_n$

   $h_{n+1} = \max\left(\min\left(h_{n+1}, \Delta t_l, \Delta t_{\max}\right), \Delta t_{\min}\right)$

   $\left(\phi, C, N\right)^{n-1} = \left(\phi, C, N\right)^n$

   $\hat{R}_{n-1} = \hat{R}_n, \quad \hat{R}_n = \hat{R}_{n+1}$

   $h_n = h_{n+1}, \quad t = t + h_{n+1}, \quad n = n + 1$

   End If
5. If $\left(t = t_{end}\right)$ terminate calculations Else Go back to 1

Figure G.1 shows the variation of the time step size for cases 11, 14, and 17 considering different tolerance levels and shows the variation of the time step size for cases 25, 26, and 27 considering different gain values. For variant gain values, cases 25, 26, and 27 uses the same initial step size of 5.0 days and tolerance level of 10^{-5} while the gain value ranging between 0.02 and 0.2. A large gain value could result in a fluctuating behavior of the time step size because the predicted time step is too large, and it will increase the local error as for case 25. For the adaptive time-stepping controller in depletion calculations of MSR, it is recommended to use an initial time-step size of 1.0 days, a tolerance of 10^{-4}, and an integral gain of 0.05 to achieve a reasonable speedup without losing solution accuracy.

Table G.1. Sensitivity of Time-step Size to Change in Controller Parameters.

Case	Initial Step (days)	Tolerance	Integral Gain	No. Time Steps	Speedup	k_{eff} Diff. (pcm)*	Max. Step (days)	Min. Step (days)
Reference	1.0	-	-	350	-	-	-	-
9	0.5	1.00E-05	0.02	104	3.3	0.1	11.4	0.5
18	1.0	1.00E-05	0.02	93	3.7	0.1	11.4	1.0
8	0.5	1.00E-05	0.05	76	4.5	0.1	16.9	0.5
17	1.0	1.00E-05	0.05	71	5.0	0.0	17.0	0.9
6	0.5	1.00E-04	0.02	64	5.6	0.1	19.9	0.5
27	5.0	1.00E-05	0.02	58	5.9	1.2	11.4	3.7
15	1.0	1.00E-04	0.02	57	6.3	0.1	20.3	1.0
7	0.5	1.00E-05	0.2	55	6.3	0.0	31.4	0.4
16	1.0	1.00E-05	0.2	54	6.7	0.2	33.0	0.7
26	5.0	1.00E-05	0.05	54	6.7	1.3	17.2	2.8
25	5.0	1.00E-05	0.2	50	7.1	1.4	28.6	1.8
3	0.5	1.00E-03	0.02	45	7.7	0.1	27.9	0.5
5	0.5	1.00E-04	0.05	41	8.3	0.1	34.2	0.5
12	1.0	1.00E-03	0.02	40	9.1	0.2	28.5	1.0
14	1.0	1.00E-04	0.05	38	9.1	0.1	34.1	1.0
24	5.0	1.00E-04	0.02	37	9.1	1.3	21.0	4.9
23	5.0	1.00E-04	0.05	28	12.5	1.2	33.1	4.7
2	0.5	1.00E-03	0.05	27	12.5	-0.2	52.7	0.5
21	5.0	1.00E-03	0.02	27	12.5	1.3	31.1	5.0
11	1.0	1.00E-03	0.05	25	14.3	0.0	52.2	1.0
4	0.5	1.00E-04	0.2	23	14.3	-0.1	45.7	0.5
13	1.0	1.00E-04	0.2	22	16.7	-0.3	48.8	1.0
22	5.0	1.00E-04	0.2	20	16.7	1.1	45.9	4.1
20	5.0	1.00E-03	0.05	19	20.0	1.5	54.0	5.0
1	0.5	1.00E-03	0.2	13	25.0	-4.6	106.8	0.5
10	1.0	1.00E-03	0.2	12	33.3	-2.2	91.9	1.0
19	5.0	1.00E-03	0.2	11	33.3	-3.2	106.9	5.0

k_{eff} Difference = $(k_{EOD} - k_{Ref\_EOD}) \times 10^5$, EOD: end of depletion time.

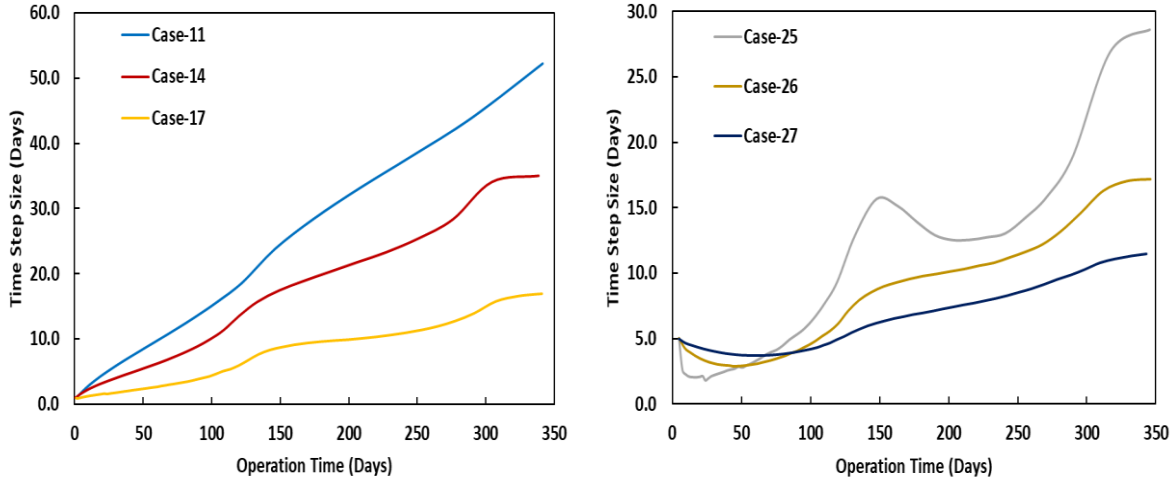


Figure G.1. Variation of depletion time-step size with different tolerance levels (left) and different gain values (right).

Bibliography

[1] L. Pioro, "Handbook of Generation IV Nuclear Reactors: Introduction: Generation IV international forum" pp. 37-54, 2016.

[2] T. Dolan, "Molten Salt Reactors and Thorium Energy: Introduction," pp. 1-12, 2017.

[3] M. Allibert, et al., "Handbook of Generation IV Nuclear Reactors: Molten Salt Fast Reactors," pp. 175-188, 2016.

[4] R. Yoshioka, M. Kinoshita, "Molten Salt Reactors and Thorium Energy: Liquid fuel, thermal neutron spectrum reactors," pp. 281–373, 2017.

[5] A. Lizin, et al., "Molten Salt Reactors and Thorium Energy: Fast-spectrum, liquid-fueled reactors," pp. 375–433, 2017.

[6] E. Bettis, et al., "The Aircraft Reactor Experiment Design and Construction", Nucl. Sci. Eng., 2, 804-825, 1957.

[7] M. Rosenthal et al., "Molten-salt reactors– history, status, and potential," Nucl. App. Technol., 8 (2), pp. 107–117, 1969.

[8] P. Haubenreich, J. Engel, "Experience with the molten-salt reactor experiment," Nucl. Appl. Technol., 8 (2), 118–136, 1970.

[9] R. Robertson, et al., "Conceptual Design Study of a Single-Fluid Molten Salt Breeder Reactor," ORNL-4541, Oak Ridge National Laboratory, 1971.

[10] R. Briggs, "Molten-Salt Reactor Program Semiannual Progress Report," ORNL-3626, Oak Ridge National Laboratory, Oak Ridge, Tennessee, 1964.

[11] GIF, "A Technology Roadmap for Generation IV Nuclear Energy Systems," GIF-002-00, U.S. DOE Nuclear Energy Advisory Committee and the Generation IV International Forum, 2002.

[12] V. Ignatiev, et al., "Progress in Development of MOSART Concept with Th Support," Proc. of ICAPP 2012, Chicago, USA, June 24-28, 2012.

[13] M. Brovchenko, et al., "Optimization of the Pre-conceptual Design of the MSFR: Evaluation and Viability of Liquid Fuel Fast Reactor System," EVOL Project, Deliverable D2.2, 2013.

[14] E. Merle-Lucotte, et al., "Launching the thorium fuel cycle with the Molten Salt Fast Reactor," In: Proc. of ICAPP 2011, Nice, France, May 2–5, 2011.

[15] K. Kramer, "TerraPower's molten chloride fast reactor technology," ETEC Nuclear Suppliers Workshop, 2018.

[16] D. LeBlanc, C. Rodenburg, “Molten Salt Reactors and Thorium Energy: Integral Molten Salt Reactor,” pp. 541-556, 2017.

[17] Moltex Energy. An Introduction to the Moltex Energy Technology Portfolio. Technical report, Moltex Energy, 2018.

[18] M. Delpech, et al., “Benchmark of dynamic simulation tools for molten salt reactors,” Proc. of International Conference GLOBAL 2003, New Orleans, LA, USA, 2003.

[19] T. Kerlin, S. Ball, R. Steffy, “Theoretical Dynamics Analysis of the Molten-Salt Reactor Experiment,” Nucl. Technol., 10, 118–132, 1971.

[20] J. Briesmeister, “MCNPTM - A General Monte Carlo N-Particle Code, Version 4C,” LA-13709-M, Los Alamos National Laboratory, 2000l.

[21] J. Kópházi, et al., “MCNP-based calculation of reactivity loss in circulating fuel reactors,” Proc. of ICNC 2003, October 20–24, Ibaraki, Japan, 2003.

[22] D. Lecarpentier, V. Carpentier, “A neutronics program for critical and non-equilibrium study of mobile fuel reactors: the Cinsf1D code,” Nucl. Sci. Eng., 143, 33–46, 2003.

[23] S. Dulla, P. Ravetto, M. Rostagno, “Neutron kinetics of fluid-fuel systems by the quasi-static method,” Ann. Nucl. Energy, 31, 1709–1733, 2004

[24] J. Krepel, et al., “DYN1D-MSR dynamics code for molten salt reactors,” Ann. Nucl. Energy, 32, 1799–1824, 2005.

[25] J. Krepel, et al., “DYN3D-MSR spatial dynamics code for molten salt reactors,” Ann. Nucl. Energy, 34, 449–462, 2007.

[26] K. Zhuang, et al., “Studies on the molten salt reactor: code development and neutronics analysis of MSRE type,” Nucl. Sci. Technol., 52(2), 251-263, 2015.

[27] K. Zhuang, et al., “Improvements and validation of the transient analysis code MOREL for molten salt reactors,” Nucl. Sci. and Technol., 54, 878-890, 2017.

[28] V. Singh, et al., “Nonlinear dynamic model of Molten-Salt Reactor Experiment Validation and operational analysis,” Ann. Nucl. Energy, 113, 177-193, 2018.

[29] A. Lindsay, et al., “Introduction to Moltres: An application for simulation of Molten Salt Reactors,” Ann. Nucl. Energy, 114, 530-540, 2018.

[30] D. Gaston, et al., “MOOSE: A Parallel Computational Framework for Coupled Systems of Nonlinear Equations,” Nucl. Eng. Des., 239, 1768-1778, 2020.

[31] D. Shen, et al., “Zero-Power Criticality Benchmark Evaluation of the Molten Salt Reactor Experiment,” Proc. of PHYSOR 2018, Cancun, Mexico, April 22-26, 2018.

[32] D. Shen, M. Fratoni, “Benchmark Evaluation of Reactivity Effects and Reactivity Coefficients in the Molten Salt Reactor Experiment,” Proc. of PHYSOR 2020, Cambridge, United Kingdom, March 29-April 2, 2020.

[33] R. Hu, l. Zou, g. Hu, “SAM User’s Guide,” ANL/NSE-19/18, Argonne National Laboratory, 2019.

[34] T. Fei, et al., “MSRE Transient Benchmarks using SAM,” Proc. of PHYSOR 2020, Cambridge, United Kingdom, March 29-April 2, 2020.

[35] S. Kondo, et al., “SIMMER-III, A Computer Program for LMFR Core Disruptive Accident Analysis,” JNC TN9400 2000-002, Japan Nuclear Cycle Development Institute, 2000.

[36] S. Wang, A. Rineiski, W. Maschek, “Molten salt related extensions of the SIMMER-III code and its application for a burner reactor,” Nucl. Eng. Des., 236, 1580–1588, 2006.

[37] C. Fiorina, et al., “Modelling and analysis of the MSFR transient behavior,” Ann. Nucl. Energy, 64, 485-498, 2014.

[38] NRC, “TRACE V5.0 User Manual,” U. S. Nuclear Regulatory Commission, 2007.

[39] T. Downar, Y. Xu, V. Seker, “PARCS v3.0, U.S. NRC Core Neutronics Simulator,” Theory Manual, University of Michigan, 2010.

[40] E. Pettersen, “Coupled multi-physics simulations of the Molten Salt Fast Reactor using coarse-mesh thermal-hydraulics and spatial neutronics,” M.S. Thesis, University Paris-Saclay & Paul Scherrer Institute, 2016.

[41] M. Tiberga, et al., “Results from a multi-physics numerical benchmark for codes dedicated to molten salt fast reactors,” Ann. Nucl. Energy, 142, 107-428, 2020.

[42] OpenFOAM Documentation. https://cfd.direct/openfoam/documentation, 2020.

[43] M. Aufiero, et al., “Development of an OpenFOAM model for the Molten Salt Fast Reactor transient analysis,” Chem. Eng. Sci., 111, 390–401, 2014.

[44] T. Hu, et al., “Coupled neutronics and thermal hydraulics simulation of molten salt reactors based on OpenMC/TANSY,” Ann. Nucl. Energy, 109, 260-276, 2017.

[45] Y. Jung, M. Smith, C. Lee, “PROTEUS-NODAL User Manual,” ANL/NE-18/4, Argonne National Laboratory, 2018.

[46] C. Lee, Y. Jung, “FY18 Progress on Improvements and V&V of MC^2-3 and PROTEUS,” ANL/NSE-18/9, Argonne National Laboratory, 2018.

[47] Y. Jung, C. Lee, M. Smith, M. Jaradat, W. Yang, “Molten Salt Reactor Analysis Capability of PROTEUS-NODAL,” Trans. Am. Nucl. Soc, 119, 1131-1134, 2018.

[48] M. Jaradat, W. Yang, “Development of Molten Salt Reactor Analysis Capabilities of the NEAMS Neutronics Code PROTEUS-NODAL,” UM/NRDSL-19/01, University of Michigan, 2019.

[49] M. Jaradat, W. Yang, Y. Jung, C, Lee, “Molten Salt Reactor Transient Analysis Capability of PROTEUS-NODAL,” Proc. of Global 2019, Seattle, USA, September 22-26, 2019.

[50] M. Jaradat, G. Yang, H. Park, W. Yang, "Multi-physics Coupling of PROTEUS-NODAL and SAM Codes for Molten Salt Reactor Simulation," UM/NRDSL-20/01, University of Michigan, 2020.

[51] G. Yang, M. Jaradat, H. Park, W. Yang, C. Lee, "Multi-physics Coupling of PROTEUS-NODAL and SAM for MSR Simulation under MOOSE Framework," Trans. Am. Nucl. Soc, 122, 647-650, 2020.

[52] G. Yang, M. Jaradat, H. Park, W. Yang, "Verification and Validation Tests of Coupled PROTEUS-NODAL and SAM Code System for Molten Salt Reactor Simulation," UM/NRDSL-20/02, University of Michigan, 2020.

[53] M. Jaradat, H. Park, W. Yang, C. Lee, "Development and Validation of PROTEUS-NODAL Transient Analyses Capabilities for Molten Salt Reactors," Ann. Nucl. Energy, 160, 108402, 2021.

[54] C. Lee, Y. Jung, M. Smith, "FY17 Status Report on NEAMS Neutronics Activities," ANL/NE-17/28, Argonne National Laboratory, 2017.

[55] G. Palmiotti, et al., "VARIANT: VARIational Anisotropic Nodal Transport for Multidimensional Cartesian and Hexagonal Geometry Calculation," ANL-95/40, Argonne National Laboratory, 1995.

[56] M. Bae, "Development of Triangle-based Polynomial Expansion Nodal SP3 Method for Hexagonal Core Transport Calculation," M.S. Thesis, Seoul National University, 2010.

[57] M. Smith, E. Lewis, E. Shemon, "DIF3D-VARIANT 11.0: A Decade of Updates," ANL/NE-14/1, Argonne National Laboratory, 2014.

[58] H. Zhang, E. Lewis, "Generalization of the Variational Nodal Method to Spherical Harmonics Approximations in R-Z Geometry," Nucl. Sci. Eng. 152 (1), 29–36, 2006.

[59] S. Zhou, W. Yang, T. Park, H. Wu, "Fuel cycle analysis of molten salt reactors based on coupled neutronics and thermal-hydraulics calculations," Ann. Nucl. Energy 114, 369–383, 2018.

[60] M. Aufiero, et al., "Calculating the effective delayed neutron fraction in the Molten Salt Fast Reactor: analytical, deterministic and Monte Carlo approaches," Ann. Nucl. Energy 65, 78–90, 2014.

[61] Y. Jung, W. Yang, "A Consistent CMFD Formulation for the Acceleration of Neutron Transport Calculations Based on the Finite Element Method," Nucl. Sci. Eng. 185 (2), 307–324, 2018.

[62] N. Todreas, M. Kazimi, "Nuclear Systems, Vol. I: Thermal Hydraulic Fundamentals," CRC Press, 2012.

[63] "RELAP5/MOD3.3 Code Manual," Nuclear Safety Analysis Division, Information Systems Laboratories, Inc., Rockville, MD, 2002.

[64] W. Yang, T. Jing, Y. Jung, S. Shi, G. Yang, "Stationary Liquid Fuel Fast Reactor," PU/NE-15/08, Purdue University, 2015.

[65] C. Lee, W. Yang, "MC^2-3 Multigroup Cross Section Generation Code for Fast Reactor Analysis," Nucl. Sci. Eng. 187 (3), 268–290, 2017.

[66] P. Romano et al., "OpenMC: A State-of-the-art Monte Carlo Code for Research and Development," Ann. Nucl. Energy, 82, pp. 90-97, 2015.

[67] J. Leppänen, et al., "The Serpent Monte Carlo Code: Status, Development and Applications in 2013," Ann. Nucl. Energy 82, 142–150, 2015.

[68] C. Lee, Y. Jung, "Generation of the Cross Section Library for PROTEUS," ANL/NE- 18/2, Argonne National Laboratory, 2018.

[69] A. Waltar, D. Todd, P. Tsvetkov, "Fast Spectrum Reactors." Springer US, Boston, MA, 2012.

[70] R. Stammler, M. Abbate, "Methods of Steady-State Reactor Physics in Nuclear Design," Academic Press, 1983.

[71] M. Chadwick, et al., "ENDF/B-VII. 0: next generation evaluated nuclear data library for nuclear science and technology," Nucl. Data Sheets 107, 2931–3060, 2006.

[72] K. Derstine, "DIF3D: A Code to Solve One-, Two-, and Three-Dimensional Finite-Difference Diffusion Theory Problems," ANL-82-64, Argonne National Laboratory, 1984.

[73] K. Ott, R. Neuhold, "Introductory Nuclear Reactor Dynamics," American Nuclear Society, La Grange Park, Illinois, 1985.

[74] R. Steffy, "Inherent neutron source in MSRE with clean 233U fuel," ORNL-TM-2685, Oak Ridge National Laboratory, 1696.

[75] R. Thoma, "Chemical aspects of MSRE operations," ORNL-4658, Oak Ridge National Laboratory, 1971.

[76] R. Kedl, "Fluid Dynamic Studies of The Molten-Salt Reactor Experiment (MSRE) Core," ORNL-TM-3229, Oak Ridge National Laboratory, 1970.

[77] B. Prince, et al., "Zero-Power Physics Experiments on the Molten-Salt Reactor Experiment," ORNL-4233, Oak Ridge National Laboratory, 1968.

[78] R. Robertson, "MSRE Design and Operations Report, Part I, Description of Reactor Design," ORNL-TM-728, Oak Ridge National Laboratory, 1965.

[79] P. Haubenreich et al., "MSRE Design and Operations Report, Part III, Nuclear Analysis," ORNL-TM-730, Oak Ridge National Laboratory, 1964.

[80] WIMS 69 energy group structure, https://www-nds.iaea.org/wimsd/energy.htm.

[81] C. Lee, et al., "Status of Reactor Physics Activities on Cross Section Generation and Functionalization for the Prismatic Very High Temperature Reactor, and Development of Spatially-Heterogeneous Codes," ANL-GenIV-075, Argonne National Laboratory, 2006.

[82] M. Chadwick, et al., "ENDF/B-VII. 1: nuclear data for science and technology: cross sections, covariances, fission product yields and decay data," Nucl. Data Sheets 112, 2887–2996, 2011.

[83] M. Rosenthal et al., "Molten-salt reactor program semiannual progress report for period ending February 28, 1969," ORNL-4396, Oak Ridge National Laboratory, 1969.

[84] J. Engel and P. Haubenreich, "Temperatures in the MSRE Core During Steady-State Power Operation," ORNL-TM-378, Oak Ridge National Laboratory, 1962.

[85] K. Gustafsson, M. Lundh, G. Söderlind, "A PI stepsize control for the numerical solution for ordinary differential equations," BIT, 28, 270-287, 1988.

[86] K. Gustafsson, "Using Control Theory to Improve Stepsize Selection in Numerical Integration of ODE," IFAC Proceedings Volumes, 23, 405-410, 1990.

[87] K. Gustafsson, "Control theoretic techniques for stepsize selection in explicit Runge Kutta methods," ACM Trans. Math. Softw., 17, 533-554, 1991.

[88] K. Gustafsson, "Control theoretic techniques for stepsize selection in implicit Runge Kutta methods," ACM Trans. Math. Softw., 20, 496-517, 1994.

[89] K. Gustafsson, G. Söderlind, "Control Strategies for the Iterative Solution of Nonlinear Equations in ODE Solvers," SIAM J. Sci. Comput, 18, 23-40, 1997.

[90] G. Söderlind, "The Automatic Control of Numerical Integration" DWI Quart, 11, 55-74, 1998.

[91] G. Söderlind, 2002. "Automatic control and adaptive time stepping," Numerical Algorithms, 31, 281-310, 2002.

[92] G. Söderlind, "Digital Filters in Adaptive Time Stepping," ACM Trans. Math. Softw., 29, 1-26, 2003

[93] A. Valli, G. Carey, A. Coutinho, "Control strategies for timestep selection in simulation of coupled viscous flow and heat transfer," Comm. in Num. Methods in Eng., 18, 131-139, 2002.

[94] A. Valli, G. Carey, A. Coutinho, "Control strategies for timestep selection in finite element simulation of incompressible flows and coupled reaction convection diffusion processes," International Journal for Numerical Methods in Fluids, 47, 201-231, 2005.

[95] P. Burrage, R. Herdiana, K. Burrage, "Adaptive stepsize based on control theory for stochastic differential equations," Journ. of Comp. and App. Math., 170, 317-336, 2004.

[96] J. Geiser, C. Fleck, "Adaptive Step Size Control in Simulation of Diffusive CVD Processes," Mathematical Problems in Engineering, 1-34, 2009.

[97] D. Walter, A. Manera, "Adaptive burnup stepsize selection using control theory for 2-D lattice depletion simulations," Prog. Nucl. Energy 88, 218-230, 2016.

[98] D. Caron, S. Dulla, P. Ravetto, "Adaptive time step selection in the quasi-static methods of nuclear reactor dynamics," Ann. Nucl. Energy 105, 266–281, 2017

[99] J. Boffie, J. Pounders, "An adaptive time step control scheme for the transient diffusion equation," Ann. Nucl. Energy, 116, 280–289, 2018.

[100] J. Boffie, J. Pounders, S. Aghara, “Validating an adaptive time step scheme for the neutron diffusion equation with the UMass-Lowell Research Reactor kinetics data,” Ann. Nucl. Energy, 132, 404–412, 2019.

[101] J. Boffie, S. Aghara, J. Pounders, “Application of an adaptive time step scheme for solving the time-dependent discrete ordinates transport equation,” Nucl. Eng. Design, 363, 110620. 2020.

[102] A. Gerlach, “Adaptive Time Stepping for the Neutron Transport Solution with the Alpha Eigenvalue,” Ph.D. thesis, University of Michigan, 2020.

[103] A. Carreno, et al., “Adaptive time-step control for modal methods to integrate the neutron diffusion equation,” Nucl. Eng. Tech. 53, 399-413, 2021.

[104] C. Gear, “Numerical Initial Value Problems in Ordinary Differential Equations,” Prentice-Hall, Englewood Cliffs, 1971.

[105] E. Hairer, S. Nørsett, G. Wanner, “Solving Ordinary Differential Equations I: Nonstiff Problems,” Springer, Berlin, 1987.

[106] E. Hairer, G. Wanner, “Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems,” Springer, Berlin, 1991.

[107] “RSICC Computer Code Collection – ORIGEN 2.1, Isotope Generation and Depletion Code Matrix Exponential Method,” CCC-371, Contributed by Oak Ridge National Laboratory, 1999.

[108] C. Gauld, O. Hermann, R. Westfall, “ORIGEN-S: SCALE System Module Calculate Fuel Depletion, Actinide Transmutation, Fission Product Buildup and Decay, and Associated Radiation Source Terms,” ORNL/NUREG/CSD-2/V2/R7, Oak Ridge National Laboratory, 2002.

[109] C. Gauld, et al., “Isotopic Depletion and Decay Methods and Analysis Capabilities in SCALE,” Nucl. Technol., 174, 169, 2011.

[110] W. Wilson, et al., "Status of CINDER'90 Codes and Data," Proc. 4th Workshop on Simulating Accelerator Radiation Environments, Knoxville, Tennessee, September 13-16, 1998.

[111] R. Moore, et al., “MOCUP: MCNP-ORIGEN2 Coupled Utility Program,” INEL-95/0523, Idaho National Engineering Laboratory, 1995.

[112] D. Poston, H. Trellue, “User’s Manual Version 2.0, for MONTEBURNS Version 1.0,” LA-UR-99-4999, Los Alamos National Laboratory, 1999.

[113] Z. Xu, P. Hejzlar, “MCODE, Version 2.2: An MCNP-ORIGEN Depletion Program,” MIT NFC-TR-104, Massachusetts Institute of Technology, 2008.

[114] M. Fensin, et al., “The New MCNP6 Depletion Capability,” Proceedings of ICAPP12, Chicago, USA, June 24-28, 2012.

[115] H. Shim, C. Kim, “McCARD Monte Carlo Code for Advanced Reactor Design and Analysis, User’s Manual Version 1.1,” Seoul National Laboratory, 2011.

[116] W. Yang, M. Smith, 2020. “Theory Manual for the Fuel Cycle Analysis Code REBUS,” ANL/NE-19/21, Argonne National Laboratory, 2020.

[117] R. Alcouffe, et al, “User’s Guide for TWODANT: A Code Package for Two-Dimensional, Diffusion-Accelerated, Neutral- Particle Transport,” LA-10049-M, Los Alamos National Laboratory, 1990.

[118] A. Nuttin, et al., “Potential of thorium molten salt reactors detailed calculations and concept evolution with a view to large scale energy production,” Prog. Nucl. Energy 46, 77–99, 2005.

[119] M. Aufiero, et al., “An extended version of the SERPENT-2 code to investigate fuel burn-up and core material evolution of the Molten Salt Fast Reactor,” J. Nucl. Mater. 441, 473–486, 2013.

[120] J. Powers, T. Harrison, J. Gehin, “A new approach for modeling and analysis of molten salt reactors using SCALE,” Proceeding of M&C 2013, Sun Valley, Idaho, May 5–9, 2013.

[121] R. Sheu, C. Chang, C. Chao, Y. Liu, “Depletion analysis on long-term operation of the conceptual Molten Salt Actinide Recycler & Transmuter (MOSART) by using a special sequence based on SCALE6/TRITON,” Ann. Nucl. Energy 53, 1–8, 2013.

[122] B. Merk, et al., “On the use of a molten salt fast reactor to apply an idealized transmutation scenario for the nuclear phase out,” PLoS ONE 9(4): e92776, 2014.

[123] S. Xia, et al., “Development of a molten salt reactor specific depletion code MODEC,” Ann. Nucl. Energy 124, 88–97, 2019.

[124] A. Rykhlevskii, J. Bae, K. Huff, “Modeling and simulation of online reprocessing in the thorium-fueled molten salt breeder reactor,” Ann. Nucl. Energy 128, 366–379, 2019.

[125] K. Zhuang, et al., “Extended development of a Monte Carlo code OpenMC for fuel cycle simulation of molten salt reactor,” Prog. Nucl. Energy 118, 103115, 2020.

[126] C. Moler, C. Van Loan, “Nineteen Dubious Ways to Compute the Exponential of Matrix,” SIAM Rev., 20, 801, 1978.

[127] G. Golub, C. Van Loan, “Matrix Computations,” Johns Hopkins University Press, Baltimore, Maryland, 1989.

[128] H. Oh, W. Yang, “Comparison of Matrix Exponential Methods for Fuel Depletion Calculations,” Journal of the Korean Nuclear Society, 31, 172-181, 1999.

[129] M. Pusa, J. Leppänen, “Computing the matrix exponential in burnup Calculations,” Nucl. Sci. Eng. 164 (2), 140–150, 2010.

[130] M. Pusa, “Rational Approximations to the Matrix Exponential in Burnup Calculations,” Nucl. Sci. Eng. 169 (2), 155–167, 2011.

[131] R. Sidje, “EXPOKIT: A software package for computing matrix exponentials,” ACM Trans. Math. Softw. 24, 130–156, 1988.

[132] American Nuclear Society, ‘‘Decay Heat Power in Light Water Reactors,” an American National Standard, ANSI/ANS-5.1, 2014.

[133] R. Robertson, “Conceptual Design Study of a Single-Fluid Molten-Salt Breeder Reactor,” Oak Ridge National Lab. Tech. Rep. ORNL-4541, 1971.