Multi-Physics Investigation of a Natural Circulation Molten Salt Micro-Reactor that Utilizes an Experimental In-Pile Device to Improve Core Physics and System Thermal-Hydraulic Performance

A Dissertation
Presented in Partial Fulfilment of the Requirements for the
Degree of Doctor of Philosophy
with a
Major in Nuclear Engineering
in the
College of Graduate Studies
University of Idaho
by
John P. Carter

Approved by: Major Professor: Richard Christensen, Ph.D.
Committee Members: David J. Arcilesi Jr., Ph.D.; R.A.Borrelli, Ph.D., Mark DeHart, Ph.D.;
Michael McKellar, Ph.D.
Department Administrator: Indrajit Charit, Ph.D.

May 2022

Abstract

The Molten Salt Reactor (MSR) concept is a rapidly evolving Generation IV design that has recently attracted favorable attention due to the potential for reducing waste generation, realizing passive safety features, and seizing on the opportunity for cost effective economics. This thesis investigates the performance benefit of a new device invented by the doctoral candidate. The device is referred to as a Wrapped Helix around an Inclined Plane or WHIP. The WHIP is protected under a provisional patent filed with the USA Patent Office on 28 September 2021 under application number 63/261,776, BEA docket number BA-1254. The WHIP device can be located in-core or near-core to promoted enhanced thermal-hydraulics and neutronics performance. While the WHIP can be employed in a variety of solid or liquid fuel designs, this thesis investigates the device's benefit in the application of a natural-circulation, micro-molten salt nuclear battery concept (MsNB). This thesis will specifically investigate the temperature coefficient of the MsNB fuel (FLiNaK) using novel temperature sensitivity techniques unique to Serpent particle transport code, evaluate the WHIP's thermal-hydraulic and neutronic performance effects using both established (STAR-CCM+) and novel (Python program developed to estimate the circulation effective delayed neutron fraction, β_{eff}) analytical and numerical methods, evaluate the neutron noise behavior of the MsNB and how the WHIP may alter the character of the MsNB's transfer function, and how the WHIP affects the autonomous, load-following performance under transient power conditions using a Python program developed by the candidate. The sum body of this work, in part, has been published in three journal articles as the timing of the provisional patent process has allowed. Results show that creative utilization of WHIP engineering design and function reduces reactor system volume, fuel loading, control/stability in the buoyant regime.

Acknowledgements

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

Dr. David LeBlanc, the Chief Technology Officer at Terrestrial Energy, Inc offered the time and had a big hand in motivating the early stages of this work on the integral molten salt reactor. I'd also like to thank Dr. Doug Crawford, INL TREAT Director, for his support throughout my program. Matthew Anderson, the INL HPC support manager, was consistent in his support and helpfulness working through my lack of knowledge in HPC matters.

In support of the work in chapter two, I would like to especially thank Dr. Ville Valtavirta, Serpent code developer, for his time and consideration answering questions regarding Serpent functionality and interpreting simulation output. In addition, I am is thankful to Matthew Anderson at the Idaho National Laboratory High Performance Computing (HPC) Center for his outstanding and timely support answering questions, and consistently making himself available to resolve technical HPC issues.

In support of the work in chapters three and five, I am grateful to Dr. SuJong Yoon in the INL modeling and simulation group for his mentorship and tutelage on STAR-CCM+ and help developing the java macro scripts that made the transient modeling on STAR possible. Robin Roper and Kristen Geddes work on experimental Ohmic Heated - Natural Circulation (OH-NC) loop. Jan Lambrechtsen, Alberto Cardenas Melgar, Jason Palmer, and Kristen Geddes were instrumental in helping me on various facets of coding, CAD development, and data interpretation.

In support of the work in chapter four, I would like to thank Dr. David Arcilesi and Dr. Imre Pazsit, a leading expert in the field of neutron noise in reactors for their outstanding support and availability to improve the quality of my knowledge in this area.

In support of the work in chapter five, I would like to thank Dr. SuJong Yoon and Dr. Ben Baker for providing of their personal time to mentor, train, and exchange ideas regarding my modeling approach.

Kellen Giraud TREAT experiment safety engineer made his time available to answer questions on MCNP syntax and providing time to proof read select journal manuscripts. Jared P. Varney helped develop python text file reading script which saved hours processed simulation results.

Dedication

Foremost, I am grateful to my late wife Cheryl Lynn Carter and my two beautiful daughters Jessica and Emily for their loving support of my passion for life-long learning; and to my wife, Kathy Lingle Carter, whose enduring patience during the long hours of study, classes, research, and writing can never be repaid. Our incredible adventure continues!

I owe most in my Life to my late Mother, Margareta Elizabeth 'Ilsa' Glunt; and my dearest Aunt and Uncle Anni and Alfred Friedhoff. Their loving involvement early in my Life significantly shaped how I view the World and embrace Life. They exemplified the Absolute Best in Human Nature and I am eternally grateful for their Love. This accomplishment is dedicated to them.

Mountains are climbed one step at a time - Take the next step.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

Molten Salt Reactors

1.1 Introduction

Molten salt reactors (MSRs) were first developed over a half a century ago. With the advent of new materials, fabrication methods, and analysis tools there has been increasing interest in this type of reactor around the globe. MSR concepts have gained renewed interest in Japan, Russia, China, France, and the USA. One of the six Generation IV designs selected for further development by the Generation IV International Forum (GIF) is the MSR (Generation IV International Forum, 2020). The GIF was initiated by the US Department of Energy in 2000 and formally chartered in mid-2001. Several early innovators have entered the design and licensing arena, including Flibe Energy, ThorCon, Moltex, Seaborg Technologies, and Terrestrial Energy. The United States' Energy Department Advanced Reactor Development Program (ARDP) recently awarded funding for MSR related research and development activities in the US. Integral molten salt fast reactor (MSFR) Euratom projects named EVOL, SAMOFAR, and SAMOSAFER are examples of past and on-going MSR efforts by a European Consortium consisting of six European countries including France, Germany, Hungary, Italy, Netherlands, and United Kingdom. MSRs are considered among the Generation IV reactor concepts due to their potential to realize non-proliferation, manageable waste stream, and sustainable global energy needs around the globe (IAEA, 2013; SAMOFAR (Safety Assessment of the Molten Salt Fast Reactor), 2021; SAMOSAFER, 2021). New ideas are being pursued that were once thought to be too expensive or intractable. For instance, the improving availability of high-end computing resources and development of robust numerical multi-physics computational methods and applications have made design, evaluation, and demonstration of advanced nuclear concepts and engineering designs more achievable than ever before (Tiberga, 2020; Alsayyari, 2020; Altahhan, 2020). Concepts and designs can be modeled, analyzed, and evaluated long before resources are allocated for more cost-effective prototypic demonstrations.

This dissertation examines the performance of a new device that is used in or near pile to enhance neutronic and thermal-hydraulic performance of a liquid fueled reactor system. The device is referred to as a Wrapped Helix around an Inclined Plane or WHIP. Detailed study of the WHIP's performance will be presented in chapters three through five. Chapter one and two lay a directed foundation of current and past MSR work. This chapter will first present a more traditional MSR design currently being pursued by Terrestrial Energy, namely the Integral Molten Salt Reactor (IMSR). The IMSR design is shown in Fig. 1.1. The thermal spectrum IMSR design uses a molten salt fuel at low pressure and high temperature within a thermalized neutron spectrum reactor. The design consists of a self-contained primary system within a single structural vessel. The heat source, heat sink, primary pumps, and heat exchangers are contained within a single integral reactor vessel. The potential exists for future MSR designs to optimize core physics, fuel salts, and innovative fuel cycle management options. The author modeled the IMSR design using publicly available criterion and using MCNP achieved reasonable comparative results to the ORNL MSRE and FUJI-MSR conceptual design. These results were journal published and are described in the next section (Carter and Borrelli, 2020a).

Chapter two will focus on presenting research the author completed and published on MSR fuel salt temperature sensitivity. This work will explain the use of FLiNaK in the thermal spectrum Molten Salt Nuclear Battery (MsNB) and present an analysis of alternative fuel salts. The approach taken to evaluate fuel salt temperature sensitivity uses a novel Target Motion Sampling (TMS) approach developed for Serpent and the results are journal published (Carter and Borrelli, 2020b) .

Chapter three provides a neutronic and CFD study of the WHIP. In this work, the WHIP is placed in-core and in some cases utilized in-core and in the chimney volume in the MsNB. Experimental work that validates the WHIP hydraulic performance in a natural circulation, ohmic-heated, closed-loop system was tested by graduate schools at the University of Idaho in the Spring of 2020. Chapter three provides an outline of the experimentation results obtained. While much work remains integrated the WHIP into experimentation loop design, the work completed so far validates the WHIP's function as a thermal-hydraulic enhancement device. The WHIP is protected under a provisional patent held by the author and is currently being licensed by industry. Serpent and STAR-CCM+ were used to complete this modeling work. Chapter three is solely focused on the WHIP as a new invention and reports on the technical features that make it a unique device for engineering MSR core-physics.

Several published works provide fundamental research into neutron noise in a flowing fuel salt system (Pazsit et al., 2010a; Pazsit and Dykin, 2010b, 2016a; Dykin, 1977). Chapter four extends their work by applying this field of research to a micro-scale, thermal spectrum natural circulation MsNB being developed by the University of Idaho (diagrammed in Fig.1.2). Chapter four provides an analysis of how the WHIP device changes the system transfer function and neutron noise behavior. Neutron noise finite-element method modeling code developed by the author solves the first-order perturbation, coupled diffusion equations for a closed-loop, flowing fuel system using Python 3.7 script run on a Linux-based HPC system (Department of Energy, 2021) and the results have been journal published (Carter and Arcilesi, 2021). The code calculates neutron and precursor noise based on fluctuations in macroscopic fission and absorption cross-sections and fuel salt velocity. The code models a simple one-dimensional, bare core using one-group energy and one-group precursor, diffusion theory. The investigation is motivated by developing a more intuitive and fundamental understanding of neutron noise phenomena in natural circulation flow molten salt reactors. In particular, MSR systems that are driven by natural circulation flow might be more prone to flow and density variations throughout core life. Accumulation of insoluble fission products, corrosion products, and localized flow instabilities can produce neutron noise effects that must be analyzed and addressed in the design and safety basis analysis. This investigation helps build an intuitive understanding based on a simplified, one-dimensional modeling scheme (Dolan, 2017). Chapter four presents the basic aspects of neutron noise theory needed to develop a useful model to interpret calculated neutron noise results. After verifying agreement with earlier published work, the investigation is extended to evaluate neutron and precursor noise that results from fuel salt flow and density fluctuations in a micro-sized, natural circulation MSR design currently under conceptual design at the University of Idaho. Flow and density fluctuations are adopted based on qualitative, empirical functional forms.

Chapter five provides a novel investigation into the time-dependent, load-following behavior of the WHIP MsNB and specifically studies how the WHIP affects the transient characteristics of system response. The author wrote a computational inexpensive, finite-element modeling code in Python which replicates the results of STAR-CCM+ CFD code with 80% similarity in results. MsNB stability and operational safety limits are examined using these modeling codes. The physics for these codes was developed by the author using the static solution to the coupled neutron diffusion and delayed neutron precursor concentration equations. The approach and results are an extension of previous work. This work is pending journal publication in 2022.

1.2 An Integral Molten Salt Reactor Design study using MCNP 6

The Integral Molten Salt Reactor (IMSR) has multiple advantages over current Generation III designs including potentially lower capital costs, higher overall safety, and resource sustainability (World Nuclear Association, 2019; LeBlanc et al., 2018). The IMSR design uses a molten salt fuel at low pressure and high temperature within a thermalized neutron spectrum reactor. The design consists of self-contained primary system within a single structural vessel. As Fig. 1.1 shows, the heat source, heat sink, primary pumps, and heat exchangers are all contained within a single integral reactor vessel. This allows for single use and disposal of low enriched fissile fuel with the potential for self-burning long-lived actinides during reactor operations. The potential exists for future IMSR designs to optimize core physics, fuel salts, and innovative fuel cycle management options that have significant opportunities to reduce fuel loading to fractions of current Light Water Reactors (LWR) which may also reduce the Pu and Transuranic (TRU) waste products. This study aims to model the IMSR in MCNP based on commercially available Terrestrial Energy IMSR design data. While the author realizes that the MCNP code is not a validation standard for MSR application, results will help inform a future broader study of new and innovative molten salt fuels (such as using spent LWR fuel) and fuel cycles that improve the waste profile (waste burnup).

MSRs use eutectic mixtures of fissile and fertile fluorides (uranium, thorium, plutonium) with other carrier salts such as Lithium, Beryllium,or Sodium fluorides. Although fast molten salt designs have been pursued, thermalizing the neutron spectrum with graphite for instance has many advantages. Thermal spectrum MSRs provide improved reactor control, reduced detrimental neutron displacement per atom (dpa) material effects, and a reduced starting fissile inventory. Additionally, using carbon/graphite throughout the core permits over-moderating the outer zone which acts as a net absorber of neutrons and helps protect the reactor vessel wall from damaging neutron exposure. Zoning moderator radially to reduce thermal neutron density (fission density at core center) while enhancing thermal neutron density in the outer region to increase fission density at edges helps level power distribution and fission heating of the fuel salt rising in the core. This potentially helps promote more stable thermal hydraulic flow of the fuel salt. The reactor vessel and structural components have typically been proposed as being made of a high nickel alloy such as Hastelloy® N; however, new corrosion resistant materials are possible (McCoy et al., 2017).

MCNP stochastic particle transport code was used to evaluate core physics and nuclear heating phenomenology in the IMSR (Holden, 2018; Lucas, 2010; Brewer, 2019; Boston, 2014). To the maximum extent possible, the Terrestrial Energy IMSR was modeled using commercially available reactor design information (LeBlanc et al., 2018; LeBlanc, 2018; Dolan, 2017). Neutronic results were compared against nominal MSR performance established in the 1960s during Oak Ridge National Laboratory (ORNL) operation of the MSR Experiment (MSRE) (Robertson, 1965). One notable difference in structural design is the model uses Hastelloy® for primary and vessel structures and Oak Ridge studied 316 and 304 stainless steel. TEI has also mentioned publicly that stainless alloys as a preferred material due to code case development. A full description of this work can by found in the author's published journal manuscript (Carter and Borrelli, 2020a)

The research outlined in chapters one and two in the dissertation was motivated by collaborative discussions between the author and Dr. David LaBlanc, Chief Technology Officer at Terrestrial Energy, Inc. The author agreed to study the effects of alternative molten salts on fuel cycle and waste stream management as it pertains to the Terrestrial Energy IMSR design. Later work described in chapters three and beyond stemmed from a need to optimize fuel loading and reactor volume in a low hydraulic loss natural circulation MSR design concept. This chapter lays the foundation for the research work that follows in subsequent chapters.

This chapter aims to provide an overview of Integral Molten Salt Reactors and develop a nominal IMSR design that provides acceptable neutron physics for further fuel cycle and waste stream study (Carter and Borrelli, 2020). The IMSR design modeled in this chapter is a beginning-of-life (plutonium free, fission product poison free), carbon thermalized, 93% enriched uranium at 0.15 mol% UF4 in salt solution, FLiBe (66.85 mol% LiF - 33 mol% BeF2 - 0.15 mol% UF4) based fuel salt. Lithium is purified to 99.992 mol% lithium-7 to minimize the consequences of tritium generation. Physics and thermal results using MCNP 6.2 are compared with nominal MSRE and other MSR designs such as FUJI-MSR (Boston, 2014). This chapter will show the model results in acceptable physics that is consistent with other real and postulated designs. Specific physics, mechanical, and thermal performance results are presented and compared to other design benchmarks to show that the modeled design represents a physically credible IMSR design. Foremost, the author intends to use the core design presented in this chapter to conduct fuel burn-up evaluations using alternative carrier salts using Serpent 2 codes (Leppanen et al., 2013) in future studies. Burn-up study results are not reproduced in this dissertation as they were not the central focus in this work. The chapter three of this dissertation will study fuel salt alternatives from a neutronics and temperature reactivity perspective.

1.2.1 IMSR Model Development

Open source Terrestrial Energy, Inc (TEI) IMSR design parameters were obtained to help inform several aspects of the model in this study (Holden, 2018; Lucas, 2010; Brewer, 2019; Boston, 2014). Some performance parameters of the design were not available through open source and therefore conservative engineering assumptions were made to complete the nominal design. The modeled design includes an inlet and outlet fuel plenum, a rudimentary heat exchanger, reflectors, fuel downcomer region all contained within an integral reactor vessel. It is assumed that the TEI design has a hexagonal fuel lattice similar to the FUJI-MSR design (Banoori et al., 2016). In addition, it is assumed that material for structures such as reactor vessel, cladding, and reflector components are composed of $\mathrm{Hastelloy}^{®}$ N. Specific form factor dimensions were not fully available open source so assumptions were made to achieve suitable neutronic fundamental mode eigenvalues. Although not fully utilized in this study, the model can easily accommodate instrumentation assemblies.

Baseline IMSR design parameters including determination of core physical dimensions, basic core thermal-hydraulic constituents, basic core layout, core fuel/moderator/reflector materials, and core control materials were determined by completing a comprehensive literature review. Although detailed design information on the specific TEI IMSR core was scarce, information on the Terrestrial Energy website and publicly released reports of TEI's patent submission were available and used extensively in this study. The general TEI IMSR design was obtained online at the US Patent Office (LeBlanc, 2018). The lack of specific and detailed TEI design information was mitigated by leveraging use of ORNL MSRE and FUJI MSR core design concepts (Dolan, 2017; Robertson, 1965; Haubenreich et al., 1964; Banoori et al., 2016; Greaves et al., 2012; Robertson, 1965). For instance, fuel and moderator and structural composition and physical dimensions were extrapolated from MSRE and FUJI designs, where appropriate. Key reactor core design features included fuel, moderator, reflector structures, core materials, fuel inlet and outlet, and downcomer flow plenums. Core materials include a stainless steel reflector, $\mathrm{Hastelloy}^{®}$ N core basket and reactor vessel and plenum structures. ENDF70 B-VII.1 and thermal S(α, β) cross section libraries of molten salt fuel and moderator isotopic compositions were used for a 600 K operating core.

All MCNP tally and eigenvalue entropy checks passed statistical checks. Idaho National Laboratory's (INL) High Performance Computing (HPC) Falcon 2 was used to complete this study over 500,000 particles per cycle and 300 cycles per computational run. Falcon is a SGI ICE-X distributed memory system with 34,992 cores and a LINPACK rating of 1 Petaflop/s. The Idaho National Laboratory (INL) High Performance Computing (HPC) facility provides high-end computing capability to support this and follow-on studies (Department of Energy and Idaho National Laboratory, 2020).

The IMSR modeled in this section has a similar form factor to the Terrestrial Energy, Inc IMSR. Fuel assemblies are zoned into three regions. The fuel channels are zoned to a nominal 85 vol.% carbon moderator and 15 vol.% molten salt fuel (83.6 wt.% carbon - 16.4 wt.% fuel salt) which results in a nominal carbon-to-fuel salt ratio of about 5.1. This equates to about one uranium- 235 atom per 890 carbon atoms in the active core region. Region I fuel channels are zoned with 10% more carbon and less fuel while region III fuel channel are zoned with 10% less carbon and more fuel.

A carbon density of 1.84 g/cm3 at 300 K and a fuel salt density of 2.05 g/cm3 at 773 K (nominal melting point) was used in the model. This overall moderator zoning scheme results in an overall flattened neutron flux distribution. Specifically, the average region I-to-region III thermal neutron flux ratio is approximately 1.51. Over 97% of all fissions occur due to thermalized neutrons. The nuclear heat rate (power density) in region I, region II, and region III is 19.5 MW (0.52 MW/l), 212.9 MW (5.4 MW/l), and 127.8 MW (2.7 MW/l), respectively. The molten salt fuel flows upward through the flow metering channels into the outlet plenum at the top of the core. The salt then flows down the downcomer region into the inlet plenum. Structural components such as pumps, and heat exchangers are located above the reflected outlet plenum region of the core do not play a significant role in core neutronics and are therefore not modeled in detail in this study.

The core has six control rod assemblies which by MCNP results provide 10% shutdown margin and sufficient reactivity control to ensure critical operations throughout a projected core lifetime (5-7 years). In addition, the core has approximately 6% excess reactivity with rods fully withdrawn from the core. This excess allows for anticipated fuel burn-up over core life without the need for in-line fuel processing as some non-integral MSR designs are challenged to achieve estimated 30 year operating lifetimes (McCoy et al., 2017; LeBlanc, 2016). Control rods are composed of enriched Gadolinium (Gd-157) to about 55%. Critical control rod height is about 135 cm above the bottom of the core (65 cm below waist center line and 34% above core bottom). Non-Integral MSR concepts such as the MSRE had very small excess reactivity due to an in-line fuel reprocessing capability. MSRs are designed with very small excess reactivity ($\delta k/k < 0.01$) due to the ability to conduct in-process refueling. The IMSR has a slightly larger amount of excess reactivity at Beginning-of-Life (BOL) to permit critical core operations at End-of-Life (EOL) without the need for in-line reprocessing. Low excess reactivity is an advantage from a safety accident consequence perspective. The IMSR is unique in its design because it is an integral unit that does not refuel or have in-process refueling capability. This potentially allows end-of-life disposal or recapture (5-7 years) in a cost effective manner. As a result, the excess reactivity of the IMSR core is larger than a MSR with in-process refueling. The IMSR modeled in this design has approximately 6% excess reactivity and can be easily adjusted with changes to fuel core volume and molten salt weight percent loading.

The core has a carbon reflector region outside of the region III fuel assemblies. The reflector is indicated in light blue. Structural components are $\text{Hastelloy}^{®}$ N or Stainless Steel with $\text{Hastelloy}^{®}$ N cladding (Lindsay, 2018). $\text{Hastelloy}^{®}$ N has been shown to have effective corrosion resistance in molten salt environments throughout the 5-7 year design operating lifetime. Neutron flux profiles across the core indicate the reflector region is effective at minimizing fission heating of fuel in the downcomer region and also effective at reducing neutron fluence at the reactor vessel. In addition, reflector regions at the top of the core are effectively reduce neutron fluence to structures while enhancing neutronics in the core. Both of these features will promote vessel and structural integrity throughout the useful operating life of the core.

ENDF70 B-VII.1 cross section libraries representative of an operating temperature of 600^{o}C and appropriate S(α, β) thermal libraries were also utilized for moderator and fuel constituents. The hexagonal fuel and moderator assemblies have a pitch of 26.152 cm in a 21 by 21 hexagonal lattice. The core radius is 1.95 meters with an effective core height of 4.15 meters. A total core heating rate of about 400 MW resulted in an average thermal neutron flux of 5.6 x 10^{14} n/cm^2-s. This thermal output equates to an average power density of about 2.9 MW/ liter molten salt in the active core neglecting contributions from the plenum and downcomer. This result corresponds well to MSRE and FUJI-MSR design and operating parameters.

1.2.2 Thermal, Material, and Neutronic Performance

With control rods partially inserted, the radial thermal neutron distribution in the core at K_{eff}=1.01. The average neutron energy causing a fission is 635.3 eV and the energy corresponding to the average neutron lethargy causing fission is 0.0597 eV. The percentages of fissions caused by neutrons in the thermal region ($<$ 0.625 eV) is 97.15%. MCNP normalizes flux tallies per source neutron. Conversions based on operating core power (fissions per second) must be completed to derive meaningful physical core properties. Straight forward calculations shown below are required to convert the MCNP flux tallies to neutrons per square cm-seconds by multiplying by the number of source neutrons per second in the core operating at a given power level. As expected, the radial flux distribution is azimuthally symmetric and indicates strong attenuation in non-thermalized and rod poisoned regions. An additional advantage of the IMSR design is the lack of fission product buildup (which can be vented at the top of the integral vessel) (LeBlanc, 2018). By contrast, solid fueled reactors are constrained by fission product buildup in fuel structures over core life. This becomes problematic for waste disposal, limits EOL core performance from a thermal hydraulic perspective, and requires burnable poisons and zoned fuels to be loaded at BOL to compensate for fuel burn throughout core life. Conversely, MSRs can operate for sustained periods with lower fission product poison buildup and therefore can operate throughout core life with a lower fuel enrichment.

The MCNP flux and energy tallies have units of neutrons/cm^2 per source (fission) neutron and MeV/g per source (fission) neutron, respectively. To obtain the number of neutrons passing through a surface area per unit time (neutrons/cm^2-sec) and heat generated per gram of material, the tallies result must be multiplied by a neutron flux conversion factor and heat normalization factor using core power (user provided) and 2.437 neutrons per fission, 180.88 MeV per fission (MCNP provided). Actual flux values and the heat generation rates are calculated using MCNP flux and energy deposition tally results.

The axial flux distribution is symmetric about the core vertical centerline and is biased low in the core below the control rod height. The inlet plenum is not as well reflected as the outlet plenum and as a result more fission heating occurs in the inlet plenum low in the core. This design helps minimize the neutron fluence to structural components (pumps, heat exchangers, and rod drive assemblies) high in the core. The neutron flux is symmetrically distributed and show the effective reduction of flux in the rod poisoned regions.

In this core, in excess of 95$\%$ of fissions are caused by thermal neutrons. For the nominal 423 MW core, averaged thermal flux over the core is 5.6 x 10^{14} n/cm^2-s. The average fast flux ($>$ 0.1 MeV) over the core is approximately 4.8 x 10^{13} n/cm^2-s. This high degree of thermalization helps mitigate graphite strain effects that may contribute to limiting core life. It is important to characterize fast flux in a graphite moderated core due to graphite's shrink and swell when exposed to high energy neutron fluence. Graphite swell under a fast neutron fluence is a key engineering limitation when determining core lifetime in graphite moderated reactors.

Natural circulation of the molten salt in the event of a pump failure is a central safety feature in Generation IV cores. In order to enhance passive cooling (natural circulation), the thermal center of the heat sink (heat exchanger) is higher than the thermal center of the heat source (reactor) and minimal heating occurs in the downcomer region of the reactor. MCNP neutron and gamma energy/mass tallies provide energy deposition for each core material volume. MCNP normalizes energy/ mass tallies per source (fission) neutron. For a 400 MW thermal core, renormalization of the energy/mass tally yields the energy deposited into each of the structural and fuel volumes. Notably, the majority of heating occurs in the inlet plenum (low in the core). This provides for a low thermal center for the heat source which is ideal for promoting natural circulation flow in the event of loss of pumping action. Less than 9% of heating occurs in non-core structures.

The downcomer fuel region is effectively shielded from a neutron flux that would result in significant fission heating. Only 0.6% of core thermal output is generated in the downcomer region. This enables a greater thermal driving head in the downcomer region which enhances natural circulation flow in the event of integral primary pump failure. Additionally, 9.8% of the fission and gamma heating occurs in the inlet plenum which also enhances passive molten salt flow in the core. MCNP calculates mass in each user designed cell (volume) using user provided density and calculated or user provided volume. Using a molten salt density of 2.05 g/cm^3 and MCNP calculated cell volumes, the fuel mass contained within the core can be calculated. In a fission product poison free core, the total core mass (fuel, moderator, control rods, cladding) of 3.0 metric tons (this includes 278.5 kg of molten salt or about 0.42 kg of U-235 at 0.15 w% salt molar concentration in the core excluding downcomer and plenum regions). These mass calculations were independently verified by hand. Build up of fission product poisons and higher actinides such as plutonium and neptunium may significantly change core physics.

1.2.3 Comparative analysis to MSRE and FUJI-MSR

It is important to compare the modeled IMSR thermal and neutronic performance against established nominal MSRE and FUJI-MSR parameters (Yoshioka and Mitachi, 2010; International Thorium Molten- Salt Forum, 2016; Krepel, 2014). Examination shows that the modeled IMSR is representative of TEI's IMSR and scalable to similar MSR designs of MSRE and FUJI-

MSR.

The modeled IMSR form factor in terms of size, volume, fuel loading, and weight are commensurate with previous operational designs or in-line with design under development. Core thermal power values similar to TEI IMSR design values result in similar core power density for thermal neutron flux values. In contrast to this studies modeled design, the TEI design is intended to operate with control rods fully withdrawn to avoid unwanted issues associated with rod withdrawal accidents. The results of this study show uniform radial neutron distributions and axial neutron flux distributions in line with expected flux shapes for radially fuel zoned cores with absorbing control rods near core mid-waistline. The modeled core also has expected excess reactivity of 6% with control rods fully withdrawn and demonstrates a shutdown reactivity ($K_{eff} \approx$ 0.9) with the six control rods fully inserted. Model neutronics yield a thermal spectrum using a 5.7 (85/15) moderator- to-molten salt fuel ratio. This equals about 890 Carbon atoms per Uranium-235 atom in the active core. These factors taken in sum build confidence that this model produces results expected of a operational reactor is usable for further modeling and simulation study. For a given fuel loading, form factor, thermal neutron flux, and operating temperature; this evaluation shows the model has a neutron flux, power density, and core lifetime comparable to MSRE and FUJI-MSR. This is encouraging and motivates the notion that the model can be used to yield useful results to study fuel burn and fuel cycle alternatives.

1.3 MSR Temperature and Flow Reactivity Coefficients

If designed properly, the MSR has a negative temperature coefficient of reactivity to provide self-limiting power control. MSR temperature coefficients will be examinated in greater detail in chapter two using a novel sensitivity approach using Serpent 2. MSRs generally have a less negative temperature coefficient compared to traditional solid fueled reactor designs due to higher metal-to-moderator ratios that exist in solid fueled systems such as light water reactors. The temperature coefficient for the molten salt nuclear battery (MsNB) reactor design considered the primarily design of interest in this dissertation due to its inclusion of the Wrapped Helix around an Inclined Plane (WHIP) in-core device is about -3.5 pcm/K near nominal operating temperatures of 950 K, as calculuted using the Serpent modeling core physics parameters.

Flow reactivity is unique to liquid fueled reactor systems. Core reactivity is a function of fuel flow rate since delayed neutron precursors are transported out of the core with the fuel salt and generate their delayed neutrons outside the active region of the core. Delayed neutrons produced in the reactor loop are very unlikely to participate in the fission power production process and result in a negative reactivity contribution to the neutron economy. The WHIP can affect where, on average, delayed neutron precursors fire their neutrons and can therefore modify the core flow reactivity behavior. Chapter three and five study this effect in detail. Chapter four examines the MsNB neutron noise and system transfer function characteristic with the WHIP installed. This research will show that despite lengthening the hydraulic flow path length in the reactor, the core remains closely coupled from a neutronics stand point.

1.4 Future Work

MSR modeling has been conducted by many researchers using various compuational fluid dynamic, particle transport and fuel burn-up codes (Aufiero et al., 2013a, 2013b, 2014; Bauman et al., 1971; Betzler et al., 2017; Brown, 2016; Cammi, 2011; Cammi et al., 2012; Cheng and Dai, 2014; Diamond et al., 2018; Engel et al., 1980; Engel and Prince, 1967; Fiorina et al., 2013, 2014; Haag, 2005; Haubenreich and Engel, 1970; Kophazi et al., 2009; Krepel et al., 2007; Sheu et al., 2013; Shi et al., 2016a, 2016b; Taube, 1974; Xu et al., 2002; Yoshioka and Mitachi, 2010; Zhou et al., 2015; Greaves et al., 2012; Department of Energy and Idaho National Laboratory, 2020; LeBlanc, 2016; Giraud and Parry, 2019; International Thorium Molten-Salt Forum, 2016; Krepel, 2014; Lindsay, 2018; Betzler et al., 2018; Betzler et al., 2018; Rahnema et al., 2017; An Evaluation of the Molten Salt Breeder Reactor, 1972; Betzler et al., 2017; Rahnema et al., 2017; Ball and Kerlin, 1965; Kerlin and Ball, 1966; Haubenreich, 1962; He, 2016; Bowman, 2011; Heuer et al., 2010; Powers et al., 2013). Most of this previous work is not related to the MsNB design and a full multi-physics evaluation of the WHIP application in the MSR design might be considered fertile ground for further investigation. Detailed use of modeling codes to examine multi-physics phenomenology might lead to new understanding of MSR steady state and transient performance in normal and unusual operating modes.

Previous work includes MSR computational fluid dynamic (CFD) modeling using FLUENT, thermal-hydraulic modeling using RELAP5, and multi-physics modeling using SAM (NEAMS), DYN3D-MSR, TRACE, MOLTRES, and COMSOL. Several of these studies acknowledge that further optimization is required to yield full modeling maturity and qualification to inform safety studies.

Specific recommended work to this follow study might include:

- Incorporate this model into Serpent and complete fuel burnup studies on FLiBe and other viable alternative molten salt fuels. Williams et al. (2006) provides a helpful examination of candidate molten salt fuels. Analysis of optimum fuel cycle/waste stream concepts based on variance in modeling parameters, e.g. start-up fuel composition/ loading, core design, fuel reuse schemes merits additional work. Improved understanding of the viability of non-Lithium (mitigate need for highly enriched Li7 and Tritium handling schemes) and non-Beryllium (reduce human toxicity exposure and Tritium generation) LEU (low enrichment Uranium, less than 5 percent Uranium-235) and SEU (slightly enriched Uranium, less than 2 percent Uranium-235) fuel salts in an MSR is of special interest. Project results will inform future work and build on previous efforts by TEI using EQL0D and Serpent codes (LeBlanc et al., 2018).

- Improved understanding of IMSR fuel cycles may result in operational IMSR designs that generate significantly less TRU waste at end of Core-Unit life. Since plutonium (and other TRU constituents) dominate radiotoxicity well beyond 300 years, improved waste stream profiles (zero or near zero TRU) will greatly reduce the radiotoxicity and activity half-life of the waste stream. This has significant upside benefit by removing environmental risks while retiring legacy political liabilities and overcoming engineering challenges associated with long-term geologic sequestration of conventional LWR waste while providing future generations of safe, carbon-free energy. Alternative salt fuels might be evaluated on the merits of neutronic, eutectic, thermal-hydraulic performance as well as radiotoxicity and environmental impact. Validation of fuel performance from a physics, chemical, and radiotoxicity perspective is warranted. Molten fuel salts can generally be categorized into Alkali salts, $\mathrm{ZrF_4}$ salts, and $\mathrm{BeF_2}$ salts (Williams et al., 2006; Thoma, 1971; Zhu and Hawari, 2017). Leading alternative molten fuels include LiF-$\mathrm{BeF_2}$-(Th/U)$\mathrm{F_4}$ used by Flibe Energy in the LFTR, NaFBe$\mathrm{F_2}$-(Th/U)$\mathrm{F_4}$ in the Martingale Thorcon reactor, LiF-(U/Pu/La)$\mathrm{F_4}$, and TEI while not explicitly stating a preferred fuel salt candidate has mentioned NaF-RbF-U$\mathrm{F_4}$ as an example of non-Lithium and non-Beryllium fuel salt. Alternative Alkali salts include LiF-KF (50-50), LiF-RbF (44-56), Li-NaF-KF ”FLiNaK” (46.5-11.5-42), LiF-NaF-RbF (42-6-52). Alternative Zr$\mathrm{F_4}$ salts include candidates such as LiF-Zr$\mathrm{F_4}$ (51-49), NaF-Zr$\mathrm{F_4}$ (59.5-40.5), LiF-NaF-Zr$\mathrm{F_4}$ (42-29-29), LiF-NaFZr$\mathrm{F_4}$ (26-37-37), NaF-RbF-Zr$\mathrm{F_4}$ (33-24-43), RbF-Zr$\mathrm{F_4}$ (58-42), KFZr$\mathrm{F_4}$ (58-42). Be$\mathrm{F_2}$ salts candidates are LiF-Be$\mathrm{F_2}$ ”FLiBe” (67-33), LiF-BeF2-Zr$\mathrm{F_4}$ (64.5-30.5-5), NaF-Be$\mathrm{F_2}$ (57-43), and LiF-NaF-Be$\mathrm{F_2}$ (31-31-38) (LeBlanc et al., 2018; Dolan, 2017).

- A full physical study of the molten salt nuclear system must include a multi-physics approach to best understand delayed neutron precursor, salt transport and deposition, and fluid flow-thermal-neutronic stability. Several MOOSE (Multi-physics Object Oriented Simulation Environment) based applications exist that might be coupled in novel ways to create new and powerful simulation analysis. Cross section databases generated by Serpent can be used in follow-on work to link MOOSE multi-physics simulation applications in an object-oriented environment. Multi-physics analysis might include Serpent (for cross section database generation), RELAP 7 (for thermal-fluids, two phase flow modeling), and BISON (fuel and structure performance modeling) to fully integrate evaluation of reactor physics and thermal hydraulic performance of the MsNB in normal and abnormal casualty modes of reactor operation. In addition, coupling PRONGHORN (Porous Medium Flow MOOSE application) and GRIFFIN (Reactor Physics and Radiation Transport MOOSE application) would enable a robust study of transient analysis and design optimization and would allow first-ever predictive modeling of material behaviors at the micro and engineering materials scale. While MOOSE based computing frameworks recently received a significant certification that allows them to be used in safety software applications, much work remains to validate these code for use on MSRs. Much experimental work remains to understand thermo-physical and chemical properties of molten salts at various temperatures and densities under irradiation flux conditions to support model and code development.

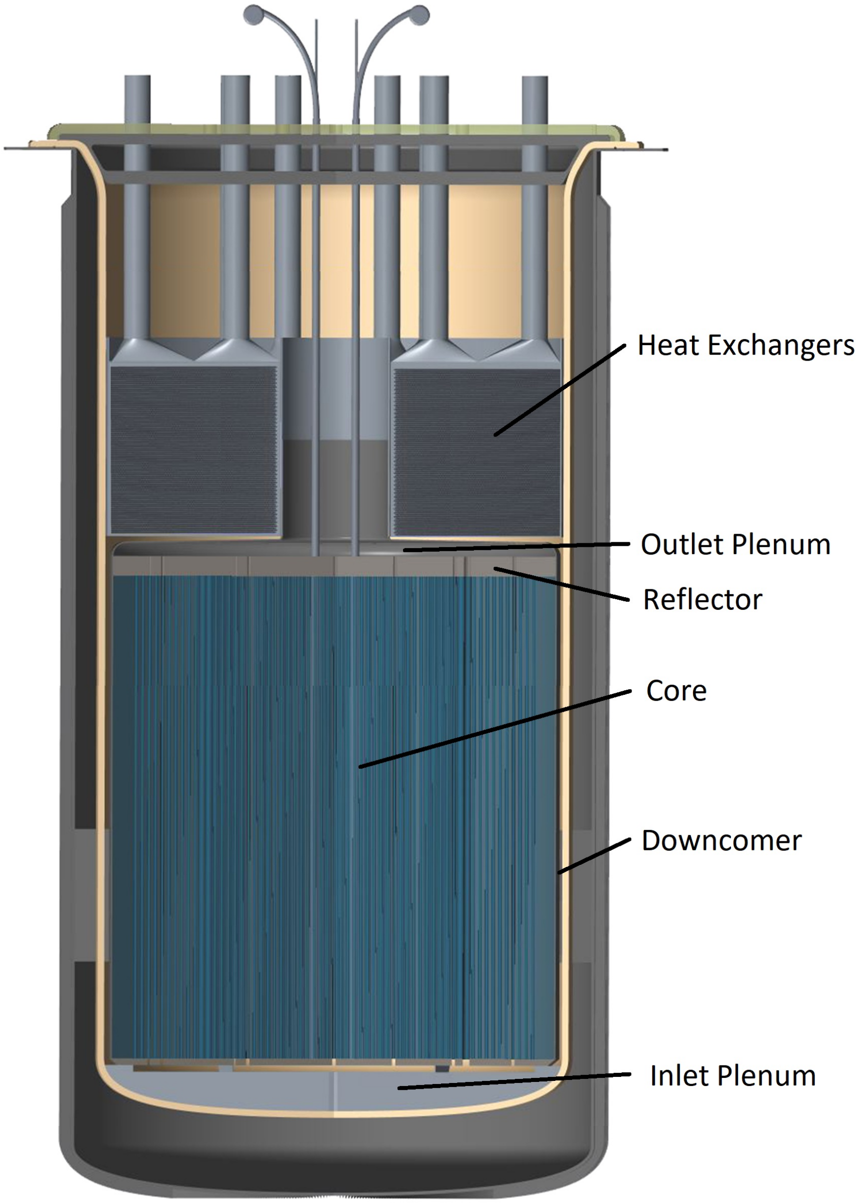


Figure 1.1: Terrestrial Energy, Inc IMSR diagram. (TEI, 2020). Primary systems are integral to reactor vessel structure. Fuel salt flow is up through the core, into the heat exchangers, downward into the downcomer region, and into the inlet plenum. Reflectors exist between core and top structures and downcomer.

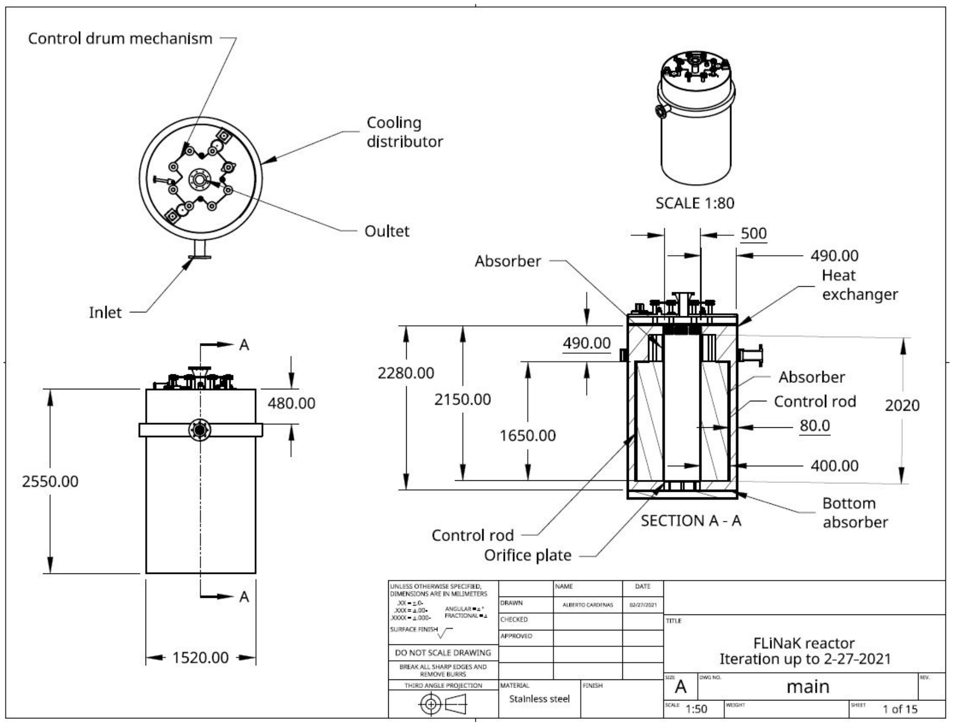


Figure 1.2: Nominal MsNB Drawing. Courtesy of Alberto Cardenas Melgar 2021.

CHAPTER 2

A Fuel Salt Temperature Sensitivity Investigation using Serpent - FLiNaK

2.1 Introduction

Methods and analyses presented in this chapter are used to determine the temperature coefficient of reactivity for the Molten Salt Nuclear Battery (MsNB). The WHIP was determined to not have a significant effect of reactor coefficients compared to traditional linear flow designs due to similarities in mean neutron thermalization time and fuel-to-moderator ratios. The results of this chapter are used throughout the remaining dissertation and chapter five in partipcular. Using a direct calculational method the temperature coefficient of the MsNB is determined be -3.5 pcm/K. This is a conservative evaluation as molten salt reactors generally have larger negative coefficients as noted by the target motion sampling (TMS) approach detailed in this chapter. FLiNaK fuel salt is specifically evaluated and compared against FLiBe and a Rubidium based Fluoride salt. A detailed discussion of TMS methodology is described in this chapter.

The MSR design uses a molten salt fuel at low pressure and high temperature within a thermalized neutron spectrum reactor. The IMSR design specifically consists of a self-contained primary system within a single structural vessel. The heat source, heat sink, primary pumps, and heat exchangers are all contained within a single integral reactor vessel (hence its name integral molten salt reactor) as shown in Fig. 2.1. The potential exists for future IMSR designs to optimize core physics, fuel salts, and innovative fuel cycle management options that have significant opportunities to reduce fuel loading to fractions of current Light Water Reactors (LWR) which may also reduce the Pu and Transuranic (TRU) waste products. This chapter aims to model the temperature coefficient behavior of the thermal spectrum IMSR in Serpent based on commercially available Terrestrial Energy IMSR design data. Similar approaches will be used to determine the temperature coefficient of the MsNB for use in this dissertation. While the author realizes that the Serpent code is not a validation standard for MSR application, results will help inform a future broader study of new and innovative molten salt fuels (such as using spent LWR fuel) and improve core performance design predictions.

The MsNB fuel salt temperature sensitivity is also of considerable interest. While the WHIP can be introduced in any engineering MSR design, the MsNB is the first reactor evaluated for WHIP use and for this reason confident knowledge of core-physics parameters including temperature coefficient is important to conducting preliminary analysis into the MsNB performance with a WHIP installed in-core and possibly in the chimney volume. Eutectic FLiNaK is the salt of choice in the MsNB and Serpent Target Motion Sampling (TMS) techniques allow temperature sensitivity analysis results obtained in modeling an IMSR can be applied to the MsNB design.

Serpent stochastic particle transport code was used in this investigation to evaluate core physics and temperature effects in the IMSR (Ville Valtavirta et al., 2019; Leppanen et al., 2013; Valtavirta and Leppanen, 2017; Aufiero et al., 2016). To the maximum extent possible, the Terrestrial Energy IMSR was modeled using commercially available reactor design information (LeBlanc et al., 2018) . One notable difference in structural design, however, is this study's model uses Hastelloy for primary and vessel structures while Oak Ridge studied 316 and 304 stainless steel. TEI has also mentioned publicly that stainless alloys as a preferred material due to benefits in code case development.

Monte Carlo methods are well suited to study the effects of changing a parameter P on a response function R. These methods have a strong equivalence to established Generalized Perturbation Theory (GPT) methods that perturb an independent parameter and evaluate the resulting dependence on a response function (Valtavirta, 2018). Although Serpent allows the user to define custom perturbations, pre-defined perturbations include nuclear reaction cross sections σ, fission spectrum χ, fission neutron yields $\bar{\nu}$, and Legendre moments of elastic scattering angular distribution. Perturbations can be applied to user specified isotopes and materials. Basic response functions include effective multiplication factor K_{eff}, effective delayed neutron fraction β_{eff}, effective prompt generation time l_{eff}, ratio of two quantities tallied by Serpent detectors, e.g. fission power and reaction rates, void reactivity coefficient, and delayed neutron precursor decay constants λ. This study will perturb temperature and cross section data and evaluate the sensitivity of K_{eff} to these perturbations.

The approach used in this chapter to evaluate K_{eff} sensitivities uses an established method that builds on a combination of collision history based sensitivities (Monte Carlo specific approach for GPT) and target motion sampling temperature treatment methods (Ville Valtavirta et al., 2019; Aufiero et al., 2015; Valtavirta et al., 2018). M. Aufiero *et al.* developed the collision history approach for sensitivity calculations and implemented additional features in the Serpent 2.1.31 release. Several perturbations and their effects on the response functions can be calculated during a single simulation. Traditional sensitivity methods would require multiple simulations. Sensitivities can also be discretely evaluated in energy (profile) or space (map).

To the first order, the unitless sensitivity of a response function R to a perturbation or small change in parameter P is defined as:

$$S_P^R = \frac{\frac{\partial R}{R}}{\frac{\partial P}{P}} = \frac{P}{R}\frac{\partial R}{\partial P}. \tag{2.1}$$

This sensitivity expression represents a dimensionless derivative of the response function with respect the perturbed parameter. The sensitivity of the response function R to a perturbation in parameter P can be described as the fractional change in the response function per fractional change in the perturbation. We take the effective multiplication factor K_{eff} to be the response function and temperature the perturbed parameter. This chapter will also examine perturbations in cross sections and are discussed in detail later.

Using basic definitions as provided in Ville Valtavirta et al. (2019), the pcm/K value for the sensitivity value or temperature coefficient of reactivity due to Doppler effects can be written,

$$\alpha_T|_{Doppler} \equiv \frac{\partial \rho}{\partial T} \quad = \quad \frac{1}{K_{eff}T}\, S_T^{K_{eff}}. \tag{2.2}$$

From a design perspective, the sensitivity calculation when applied to various isotopes and materials of interest (U^{235}, U^{238}, fuel salt, moderator, core structures) can assist with the evaluation of reactor temperature performance.

The target motion sampling (TMS) technique is a stochastic approach for nuclear cross section temperature treatment. The technique is based on calculating Doppler-broadening effects in nuclear cross sections. Temperature effects in cross sections can be determined by considering the interactions between neutrons and low temperature (lower base temperature; ideally 0 K) target nuclei and then including the effects of random thermal motion of target nuclei. The TMS approach calculates elevated temperature cross sections ”on-the-fly” by sampling target nuclei energies from a Maxwell energy distribution corresponding to local reactor component temperatures. It should be noted that the TMS approach does not currently consider the unresolved resonance region or the temperature effects on bound state data, i.e. S(α,β) data (Ville Valtavirta et al., 2019; Aufiero et al., 2016; Valtavirda, 2018; Aufiero et al., 2015; Valtavirta et al., 2018). For this reason, TMS treatment is not used to evaluate for moderator sensitivities and more direct calculations are used. Moreover, because the Doppler-broadening preprocessor routine does not currently address unresolved resonance range, study of a fast spectrum core would require production of temperature specific, point-wise and multi-group cross section libraries using a nuclear data processing system such as NJOY. In this study, temperature coefficients are evaluated over a range of core operating temperatures (773 K to 1600 K) for various materials. Lastly, while much of the focus of this study examines the sensitivity to perturbations in the thermal energy spectrum, TMS results in the fast spectrum show significant less sensitivity above 10 eV.

In practice, the core's overall temperature coefficient is a sum over all reactor materials, i, at their respective temperatures:

$$\alpha_T \rightarrow \sum_i \frac{\partial\rho}{\partial T_i} \quad = \quad \alpha_{T,\,\mathrm{fuel}} + \alpha_{T,\,\mathrm{moderator}} + \alpha_{T,\,\mathrm{structures}}. \tag{2.3}$$

The first term will be evaluated using TMS treatment and perturbing temperature (Doppler coefficient) and microscopic cross section (density coefficient). The second and third terms will be evaluated together using the direct approach using temperature pre-processor functionality in Serpent. Although TMS treatment was first implemented in version 2.1.27, this study uses Serpent release version 2.1.31. Some work using 2.1.32 developmental version to correct for some bias in uncertainty calculators was included in this study. To optimize Serpent resource and physics performance, typical Serpent physics settings included 240,000 particles in 65,000 cycles, the size of neutron buffer (nbuf) size was set at 20, the batching interval was set at 40, the number of generations used to calculate iterated fission probability (ifp) was 12, a default value of 1 was used for the number of independent parallel eigenvalue calculations, and Serpent optimization mode was set to 1 (slowest). These settings consistently resulted in approximately 32.6 million simulated particles histories per Serpent computation. A large number of particle histories provides improved confidence that the treatment approach will provide statistically significant results. All computational runs passed all internal Serpent eigenvalue statistical comparative checks.

As mentioned above, calculating the temperature coefficients of non-fuel material is performed using the direct method by changing pre-processed material density set in the mat card and temperatures set using temperature pre-processed (tmp) cards and conducting multiple simulation runs to determine K_{eff} variations. In addition to cross section tailoring, the direct method accounts for changes in physical material densities.

This chapter was motivated by collaborative discussions between the author and Dr. David LaBlanc, Chief Technology Officer at Terrestrial Energy, Inc. The author agreed to study the effects of molten salt fuels on the temperature coefficients of a near Terrestrial Energy IMSR design. This investigation uses publicly available IMSR design concepts and novel Serpent K_{eff} sensitivity treatments to conduct a study of this areas of interest. The numerical approach used to investigate the IMSR temperature coefficients was also used to determine the temperature coefficient of the MsNB (-3.5 pcm/K). This value is used extensively in the MsNB WHIP studies that follow in subsequent chapters to analysis for neutronic and thermal-hydraulic MsNB performance. The temperature coefficient of the MsNB is not expected to vary significantly across the range of nominal operating temperatures and for this reason is assumed to be constant between 550^{o}C and 800^{o}C. A time dependent power performance is completed in chapter five using this value. If system temperatures deviate significantly form this assumed range than additional analysis may be needed to determine the full temperature dependence of the MsNB temperature coefficient.

The goal of this chapter is to calculate spectral (energy) K_{eff} sensitivities (coefficients) of an IMSR design at nominal and off-nominal temperatures and determine the MsNB coefficient for use in subsequent chapters. The results are evaluated from an operational control stand point. While the TMS approach has been shown to produce results that correlate well with traditional methods, it should be emphasized the treatment is not yet certified for use from a nuclear quality assurance perspective. This study will also provide recommendation for future study on the IMSR to include the use of Serpent to conduct an evaluation of alternative molten salt fuel burnup in the IMSR design.

Open source Terrestrial Energy, Inc (TEI) thermal IMSR design parameters and basic physics behaviors were obtained from the TEI patent (LeBlanc et al., 2018) and a previous MCNP study (Carter and Borrelli, 2020a). As illustrated in Fig. 1.1, the modeled design includes an inlet and outlet fuel plenum, a rudimentary heat exchanger, reflectors, fuel downcomer region all contained within an multi-component integral reactor vessel. The hexagonal fuel and moderator assemblies have a pitch of 26.152 cm in a 21 by 21 hexagonal lattice. The core radius is 1.95 meters with an effective core height of 4.15 meters. The percentage of fissions caused by neutrons in the thermal region (< 0.625 eV) is 97.15%. ENDF/B-VII.1 and thermal $\mathrm{S}(\alpha, \beta)$ cross section libraries of molten salt fuel and moderator isotopic compositions were used for a range of operating core temperatures. A total core heating rate of about 400 MW resulted in an average thermal neutron flux of $5.6 \,\mathrm{x}\, 10^{14}$ n/(cm^2-s) (Carter and Borrelli, 2020a). This thermal output equates to an average power density of about 2.9 megawatt per liter molten salt in the active core neglecting contributions from the plenum and downcomer. The downcomer fuel region is effectively shielded from a neutron flux that would result in significant fission heating. Only 0.6% of core thermal output is generated in the downcomer region. This enables a greater thermal driving head in the downcomer region which enhances natural circulation flow capability in the event of integral primary pump failure. Fig. 2.2 and Fig. 2.3 provide graphical representation of select thermal neutron flux radial distributions (Andrew Johnson et al., 2020). To support the interpretation of the results for the three fuel salts presented in this study, a representative neutron flux spectrum are provided in Fig. 2.4. As modeled in this study, the core operates in the epithermal region with neutron flux between 0.01 - 1 eV. This flux spectrum does not vary significantly as a function of the three fuel salts examined in this chapter.

Idaho National Laboratory's (INL) High Performance Computing (HPC) Sawtooth was used to complete this study using over 240,000 particles per cycle and 65,000 cycles per computational run. Sawtooth is a HPE SGI 8600 distributed memory system with 96,480 cores and a LINPACK rating of 5587.93 Tflop/s. The Idaho National Laboratory (INL) High Performance Computing (HPC) facility provides high-end computing capability to support this and follow-on studies (Department of Energy, 2020).

2.2 Target Motion Sampling (TMS) temperature sensitivity treatment

The fuel salt temperature coefficient is comprised to two components, namely the Doppler components associated with cross section broadening and the changes in macroscopic cross section is a result of material density changes. The TMS treatment outlined below focuses on calculation of the Doppler component of fuel salt. Some calculation of the density component is also evaluated.

Understanding local spectral sensitivities of the effective neutron multiplication factor due to changes in core temperature (and cross sections) is essential for undertaking reactor control and design basis accident studies of core design. TMS provides an accurate treatment of K_{eff} sensitivities to perturbations in temperature and microscopic cross sections. The TMS approach involves using a Doppler-broadening pre-processor to set the temperature for all non-perturbed materials. Perturbed materials use run-time or ”on-the-fly” treatment to perturb the temperature. Run time cross section temperature effects are handled by sampling the target nuclei energy from a Maxwellian energy distribution. Changes in the Maxwellian energy distribution due to temperature changes are calculated analytically which allows a correlation between energy distribution perturbation sensitivities and material temperature sensitivities. For the isotopes in the perturbed materials (e.g. fuel materials), Serpent pre-broadens the cross section data to a defined minimum temperature, if 0 K cross sections are not used. If 0 K data is used then pre-broadening is not needed. For sensitivity calculations, it is desired to have the minimum amount of pre-broadening (use the lowest possible temperature cross section library for the nuclides in the materials to be perturbed, zero cross sections were used in this study) and the maximum amount of target motion sampling during particle history computations or ”on-the-fly”. Doubling the difference between base temperature (temperature of interest set by analyst) and temperature of cross section data will roughly reduce uncertainties in the calculated sensitivities by half. For example, if analyzing for a base temperature of 600 K, using 0 K cross section files over 300 K cross section files will reduce the statistical output uncertainty by about half (factor of 1.5 to 1.8). For the perturbed material, the TMS minimum and maximum temperatures are set manually using the tft (sets maximum and minimum temperature limits for material for coupled multi-physics calculations) Serpent command. The minimum temperature should be equal to the lowest cross section library temperature so that the pre-processor will not pre-broaden the cross sections. The maximum temperature is set to the maximum evaluated material nuclide temperature (set by analyst). This study used 0 K as the minimum and 1600 K as the maximum TMS temperature limits. The nominal material temperatures are set using a multi-physics interface file (.ifc). This file links the actual material temperature to the simulation. Core temperature distributions can be defined using regular mesh based interfaces in the file (Ville Valtavirda et al., 2019; Aufiero et al., 2015; Valtavirda et al., 2018; Viitanen, 2015). The computational performance of this approach is explicitly addressed in Ville Vardavirda et al. (2019).

Cross section data memory consumption may be an issue with TMS since energy grid thinning is switched off when TMS is in use. Lower optimization modes can mitigate the cross section data memory consumption problems. If event based scoring of the sensitivities is used the event bank size can also be adjusted to accommodate user needs. If direct scoring is used (which may be more efficient from a memory perspective), the user may adjust how many score matrices are pre-allocated. It should be noted again that the TMS sensitivity treatment does not consider the temperature effects in bound thermal scattering S(α, β) data nor cover the unresolved resonance probability tables. The effects of the unresolved resonance region is difficult to predict but effects are most likely less pronounced in thermal reactors and are therefore not considered in this study.

The multiple simulation direct approach uses the fastest Serpent optimization mode 4. If 0 K cross sections are used then Doppler-broadening rejection correction (DBRC) is enabled to improve accuracy of results. The single simulation TMS approach for fuel materials uses optimization mode 1 (slowest) and DBRC need not be set.

While K_{eff} sensitivities to temperature perturbations can be used to evaluate for the Doppler component of the temperature coefficient of reactivity, K_{eff} sensitivities to microscopic cross section perturbations can be used to evaluate for the density component of the temperature coefficient of reactivity. This is the case because in the system of neutron transport equations the nuclide density is always found as multiplier for some microscopic cross section. Therefore, any variations or perturbations of the microscopic cross sections can be viewed as a similar perturbation of the nuclide density.

For both thermal and fast systems, the relationship between $S_T^{K_{eff}}$ and $S_\sigma^{K_{eff}}$ can be shown to be,

$$S_T^{K_{eff}} = \frac{\partial \sigma}{\partial T}\frac{T}{\sigma}S_\sigma^{K_{eff}} \tag{2.4}$$

using the definition of the sensitivity value provided in Eq. 2.1. Therefore the density contribution to the temperature coefficient of reactivity can be written in terms of the cross section sensitivity as,

$$\frac{\partial \rho}{\partial T} = \frac{1}{K_{eff}\sigma}\frac{\partial \sigma}{\partial T}S_{\sigma}^{K_{eff}}. \tag{2.5}$$

Eqn. 2.5 has a similar form to Eqn. 2.2 but includes an additional derivative factor. In terms of a material density D, Eqn. 2.5 can be written

$$\frac{\partial \rho}{\partial T} = \frac{1}{K_{eff}D}\frac{\partial D}{\partial T}S_{\sigma}^{K_{eff}} \tag{2.6}$$

because from the perspective of the perturbation routines in Serpent, perturbations in microscopic cross section are equivalent to perturbations in material density.

Moreover, core reactivity can be expressed as an implicit function of neutron multiplication factor (K_{eff}), total neutron cross section, and temperature. Therefore, by chain rule differentiation, the temperature coefficient of reactivity in the thermal case can be expressed as:

$$\begin{aligned}\frac{\partial \rho}{\partial T} &= \frac{\partial \rho}{\partial K_{eff}}\frac{\partial K_{eff}}{\partial \sigma}\frac{\partial \sigma}{\partial T} && (2.7)\\ &= \left(\frac{1}{K_{eff}^{2}}\right)\left(\frac{K_{eff}}{\sigma}S_{\sigma}^{K_{eff}}\right)\left[\left(-\frac{\sigma_o}{2}\right)\left(\frac{T_o}{T^3}\right)^{\frac{1}{2}}\right] && (2.8)\end{aligned}$$

using $\sigma = \sigma_o\left(\frac{T_o}{T}\right)^{\frac{1}{2}}$ in the thermal region and where T_o and σ_o are reference cross section and temperature values (DOE Fundamentals Handbook, 1993). $S_{\sigma}^{K_{eff}}$ is the sensitivity of K_{eff} to perturbations in microscopic cross section, σ, obtained by the Serpent sensitivity treatment. If temperature and cross section equals those of the reference values which can be chosen to be the case then,

$$\alpha_T|_{Density} \equiv \frac{\partial \rho}{\partial T} \quad = \quad -\frac{1}{2K_{eff}T}S_{\sigma}^{K_{eff}}. \tag{2.9}$$

Eqn. 2.7 shows that cross section temperature dependence is important to understand K_{eff} sensitivity to cross section perturbations. For instance, due to the 1/v cross section behavior in the thermal case, positive cross section sensitivities result in negative temperature dependencies in reactivity coefficients. Serpent calculated core cross sections are shown in Fig. 2.5.

This relationship in Eqn. 2.6 and 2.9 holds for thermal systems and can be used to calculate the reactivity coefficient due to material density changes. The minus sign in Eqn. 2.9 is of particular interest in understanding the highly desired reactor characteristic of self-limiting control. In thermal regimes, for strong negative temperature coefficients, positive K_{eff} sensitivities to cross section perturbations are highly desirable. To exhibit a negative temperature coefficient, perturbations to cross sections pertinent to the neutron life cycle should have *net* positive sensitivity contribution to K_{eff}. Although fission and capture cross sections will generate opposite sensitivities in K_{eff}, it is desired the *net* sensitivity be positive for a reactor control perspective. This is principally true for thermal spectrum reactors where cross sections generally decrease with increasing core temperatures.

It should be noted that for non-equilibrium conditions when, for instance, during rapid power transients where core period is much less than the characteristic core thermal diffusion time can greatly affect these calculated results. The assumption that neutron energies are representative of core temperatures and respective neutron cross sections may not be valid and calculation of temperature coefficients is significantly more challenging than in the steady-state thermal case.

2.3 MSR temperature coefficients

IMSR and MsNB fuel salt and total reactor coefficients will be discussed in this section. Fuel salt Doppler coefficients for three salts (FLiBe, FLiNaK, and NaF-RbF) were determined using the Serpent TMS treatment. Individual fuel salt constituents are provided in Tables 2.1 through 2.3. For comparability purposes, the fuel salt mix concentration of U^{235} was maintained at 0.65 mol % for each salt. The K_{eff} sensitivities to temperature perturbations are examined over a temperature range of 773 K to 1600 K. The results show varying sensitivities in nominal operating and elevated temperature ranges. Fuel salt density coefficients for the three fuel salts are determined using K_{eff} sensitivities due to cross section perturbations. Total reactor temperature coefficient of reactivites are calculated using neutron physics data (K_{eff} six-factor contributions) provided by Serpent.

The ENDF libraries used by Serpent to calculate neutron capture cross sections for non-actinide fuel salt constituents are summarized in Table 2.4. The flourine capture is small and present in all the fuel salts. In addition, lithium and beryllium have negligible capture cross section at the relevant core energies. Potassium, rubidium, and sodium have the largest capture cross sections and to varying degrees their affect is evident in overall core reactivity.

The MsNB reactor temperature coefficient in the nominal operating range of 600^{o}C to 800^{o}C was determioned using the treatments described in this chapter. For a FLiNaK eutectic with 18% mol fuel loading the temperature coefficient is -3.5 pcm/K. It is also notable that the MsNB neutron energy spectrum mitigates the high Potassium capture cross section that is present in the thermal energy range (see Table 2.4). The MsNB average neutron energy is approximately 0.17 eV and the Potassium capture cross section has a factor of 10 reduction at these energies.

2.3.1 Fuel salt Keff sensitivities to temperature perturbations (Doppler coefficient)

The K_{eff} sensitivity to temperature perturbations is used to calculate the fuel salt Doppler component of the temperature coefficient of reactivity. Fig. 2.6 shows the K_{eff} sensitivity to temperature perturbations at 1000 K for the three fuel salts. Similar sensitivity calculations were completed across a temperature range of 773 K to 1600 K at 100 K increments. This range captures the operational envelop of interest.

Fig. 2.6 shows the calculated K_{eff} sensitivity profiles for temperature perturbations based at 1000 K. The sensitivities are a function of neutron energy at the perturbed temperature and the zero sensitivity crossover values represent the zero base perturbation temperature (1000 K in this case). Cross over at base temperature is a manifestation of Serpent's numerical perturbation routines and therefore higher base temperatures will move this crossover point to the right. While the general response shapes are similar between the fuel salts, the integrated energy sensitivities (area under the curve) vary and are used to explicitly calculate the Doppler coefficients using Eqn. 2.2. Temperature sensitivities above 1 eV are negligible (calculated as zero) for all three fuel salts. Notably, the FliBe differs from FLiNaK and NaF-RbF in that fuel salt K_{eff} spectral sensitivities are greater in magnitude than constituent uranium contributions. Specifically, the magnitude of total fuel temperature sensitivity exceeds the U^{235} temperature sensitivity in FLiBe (Fig. 2.6(a)) whereas the reverse is the case in FLiNaK and NaF-RbF (Fig. 2.6(b) and 2.6(c)). Fig. 2.6 also shows that temperature perturbations above the base temperature (higher neutron energies) result in a positive K_{eff} sensitivity and perturbations below the base temperature (lower neutron energies) result in negative K_{eff} sensitivities. This can the explained when considering Doppler broadening in the target motion sampling treatment. FLiBe has a total energy integrated K_{eff} sensitivity of -2.55317x10^{-2}. In contrast, at 1000 K FLiNaK and NaF-RbF have total energy integrated K_{eff} sensitivities of -4.57168x10^{-3} and

$-7.16196\text{x}10^{-3}$, respectively. In all three fuel salt cases, U^{238} has small negative contribution to sensitivity results (about $-2.0\text{x}10^{-3}$). This is expected due to the low concentration of this fertile constituent in the fuel salt. U^{238} becomes a larger contributor to the total fuel salt temperature sensitivity at lower fissile enrichment levels.

The calculated Doppler coefficients from these sensitivity results are shown in Table 2.5 and Fig. 2.7 through 2.9. Examination of these figures provides a good comparison between fuel salt Doppler coefficients over a nominal operating temperature range. FLiBe has strong negative Doppler coefficient across the evaluated temperature range (773 K to 1600 K). For FLiBe, the Doppler coefficient is -2.896 pcm/K at 773 K and monotonically increases to -2.048 pcm/K at 1600 K. This Doppler coefficient in conjunction with density coefficient in fuel results in an overall reactivity temperature coefficient of the FLiBe reactor of about -23.814 pcm/K. U^{235} does not have a significant Doppler contribution. U238 has a slight negative Doppler contribution in all three fuel salts studied (roughly -0.020 pcm/K to -0.008 pcm/K). FLiNaK has a moderately negative fuel salt Doppler coefficient across the temperature range of interest ($\approx$ -0.540 pcm/K). Furthermore, NaF-RbF also has a moderately negative Doppler coefficient (-0.816 pcm/K at 773 K and -0.652 pcm/K at 1600 K) and similar to FLiBe generally increases (less negative) as temperature increases. In all three fuel salts, uranium is not a significant contributor to the Doppler coefficient in the temperature range of interest. This is primarily due to the high enrichment of the uranium used in this simulation.

In terms of Doppler coefficient contribution to reactor self-limiting power control, FLiBe has the largest negative fuel salt Doppler coefficient. At the nominal reactor operating temperature of 1000 K, the FLiBe fuel salt Doppler coefficient is -2.562 pcm/K and FLiNaK and NaF-RbF are -0.540 pcm/K and -0.781 pcm/K, respectively. The fuel salt temperature coefficient of reactivity due to fuel salt density changes is equally important to fully understanding the temperature dependence of fuel salt reactivity.

2.3.2 Fuel salt Keff sensitivities to cross section perturbations (density coefficient)

The temperature coefficient of reactivity due to changes in material density can calculated using the temperature pre-processors to adjust cross sections. As discussed in further detail below, perturbations in material density are computationally equivalent to perturbations in total microscopic cross section. Serpent determines density coefficients by setting K_{eff} to be the response function and perturbing the total microscopic cross section. Fig. 2.10 shows the K_{eff} sensitivity to total cross section perturbations at a base perturbation temperature of 1000 K. The energy integrated K_{eff} sensitivities due to total fuel cross section perturbations is shown in the figures. Notice that FLiBe appears to have the highest positive energy integrated K_{eff} sensitivity. The sensitivities of both FLiNaK and NaF-RbF are roughly 63% of FLiBe. It is apparent from Fig. 2.10 that sensitivity analysis in the resolved and unresolved resonance regions has the potential to provide less accurate results despite large computational investment. The sensitivity maximum is centered about the user defined analysis base temperature (1000 K in this case). From Eqn. 2.9, positive sensitivities yield negative temperature coefficients. This inverse relationship is the result of the negative cross section 1/v slope in the thermal region.

At 1000 K, the temperature coefficient of reactivity due to fuel salt density changes are -2.186 pcm/K, -1.737 pcm/K, and -1.446 pcm/K for FLiBe, FLiNaK, and NaF-RbF, respectively. These values are similar in magnitude to the fuel salts Doppler coefficients. When combining both Doppler and density coefficients, FLiBe provides the best self-limiting fuel performance of the three salts studied. This will be true when considering the total reactor temperature coefficient (including moderator and structural material density effects).

2.3.3 Comparison of three fuel salt temperature coefficients

Three candidate fuel salts are compared on the basis of their K_{eff} sensitivities to temperature and cross section perturbations. The fuel salt temperature coefficients of ”FLiBe” $\mathrm{LiF-BeF_2-UF_6}$ (66.85-32.5-0.65 mol%), ”FLiNaK” $\mathrm{LiF-NaF-KF-UF_6}$ (45.85-11.5-42-0.65 mol %), and $\mathrm{NaF-RbF-UF_4}$ (73.35-26-0.65 mol%) were determined using the Serpent TMS treatment with informed thermo-physical data and a comparative evaluation is presented (Thoma, 1959; Sohal et al., 2013; Serrano-Lopez et al., Sept 2013; International Atomic Energy Agency (IAEA), 2013). Each fuel salt was evaluated near its respective eutectic composition with a similar fissile concentration. For ease of comparison, each fuel salt in this study has the same molar concentration of fissile uranium. Fuel salts without lithium and beryllium constituents are of unique interest due to the lack of radiotoxins such as tritium and hazardous materials such as beryllium in the waste stream. While TEI has not explicitly stated a preferred fuel, fuel salts without lithium and beryllium constituents are of special interest for this reason.

Reviewing the results summarized in Tables 2.5 and 2.6 provides additional insight. For instance, FLiBe is the best neutron economy followed by NaF-RbF second and FLiNaK a close third (this difference could be easily overcome by changing the fuel fissile composition). This is likely due to the neutron absorption cross sections of non-fissile fuel salt constituents as shown in Table 2.4. Secondly, FLiBe has the strongest negative Doppler and density coefficient contributions near nominal reactor operating temperatures of 1000 K. Total reactor coefficients determined at local discrete temperatures vary about $\pm 0.3\mathrm{pcm/K}$ from the averaged values provided in Table 2.6. Thirdly, FLiNaK has a relatively small Doppler contribution at 1000 K. Moreover, FLiBe has the most negative total average temperature coefficient of the three fuels studied (this difference is appreciable and may inform fuel salt design considerations). Lastly, the density and Doppler coefficients in FLiBe have a larger relative contribution to the fuel salts total average temperature coefficient compared to FLiNaK and NaF-RbF (20% compared to about 12-15%, respectively).

Although NaF-RbF does not have the most negative temperature coefficient of the fuel salts studied, NaF-RbF does have a reasonable self-limiting power characteristics. Considering the overall results in this study and the absence of lithium and beryllium in NaF-RbF, a stronger potential argument can be made for the use of NaF-RbF in thermal IMSR designs. Of course other factors such as thermo-physical and thermo-chemical properties must also be considered and evaluated. Nevertheless, this study provides information worthy of design consideration.

2.3.4 Average reactor temperature coefficient of reactivity

The average overall temperature coefficient of reactivity of the reactor can be calculated using a direct physics approach. Although this approach can be computationally expensive requiring several simulations at discrete temperatures to determine variance of several reactor kinetics parameters with respect to temperature, it is useful to briefly use this approach to provide an independent comparison to evaluate the TMS results.

The temperature coefficient of reactivity can be calculated using the K_{eff} six-factor formula. Namely, using the familiar six-factor formula definition of K_{eff}, we can derived the temperature coefficient of reactivity to be,

$$\frac{\partial \rho}{\partial T} = \frac{1}{K_{eff}} \left(\frac{1}{\eta} \frac{\partial \eta}{\partial T} + \frac{1}{f} \frac{\partial f}{\partial T} + \frac{1}{p} \frac{\partial p}{\partial T} + \frac{1}{\epsilon} \frac{\partial \epsilon}{\partial T} + \frac{1}{\mathscr{L}_{fnl}} \frac{\partial \mathscr{L}_{fnl}}{\partial T} + \frac{1}{\mathscr{L}_{tnl}} \frac{\partial \mathscr{L}_{tnl}}{\partial T} \right) \tag{2.10}$$

Each term in Eqn. 2.10 is calculated and provided in Tables 2.7, 2.8, and 2.9 for the three fuel salts. The derivatives in each of the coefficient terms are calculated as averages over the temperature range of interest (between 773 K and 1000 K in this case). The total temperature coefficient is calculated by summing the individual six-factor contributions. Review of the fractional changes in the neutron cycle as temperature changes is valuable in gaining a deeper understanding for why the temperature coefficients differ across the three fuel salts. Eqn. 2.10 shows that each stage of the neutron life cycle contributes to the temperature coefficient in different meaningful ways.

Average reactor temperature coefficients of reactivity are summed in Table 2.6. Reactor coefficients at specific temperatures may vary from those average calculated values due to local physics effects that involve neutron life cycle. In other words, coefficients at discrete temperatures may vary depending on core physics at that temperature. As discussed, moderator and structure coefficients (primarily moderator density factors) account for difference between fuel salt coefficients and the total reactor coefficient.

Table 2.6 summarizes the results of average reactor temperature reactivity behavior for the three fuel salts under consideration. The average temperature coefficients account for effects due to temperature changes in all reactor materials. The temperature coefficients for FLiBe, FLiNaK, and NaF-RbF fueled cores are -23.814 pcm/K, -17.927 pcm/K, and -13.478 pcm/K, respectively. The differences in the coefficients are primarily due to differences in the fractional changes in the thermal utilization factor, f, the resonance escape probability, p, and the fast fission factor, ϵ with respect to temperature. The fractional changes with respect to temperature in the thermal utilization factor contributes -9.457 pcm/K to the total reactor coefficient in the FLiBe fueled reactor (40%). This same contribution is -3.634 pcm/K and -1.486 pcm/K (20% and 11%) for FLiNaK and NaF-RbF, respectively. In FLiBe, this indicates that as temperature increases the probability that an absorbed neutron that is absorbed in fuel decreases at a greater fractional rate than compared to FLiNaK and NaF-RbF. As temperature increases FLiBe, fractionally fewer neutrons are absorbed in fuel compared to the other fuel salts. In FLiBe, the resonance escape probability contributes -2.931 pcm/K (12%) to the total reactor coefficient. This same contribution is -5.325 pcm/K and -8.394 pcm/K (30% and 62%) for FLiNaK and NaF-RbF, respectively. In FLiBe, this indicates that as temperature increases fractionally fewer neutrons are lost to resonance absorption compared to FLiNaK and NaF-RbF. This is likely due to FLiBe having a slightly softer neutron spectrum (better moderator ratio) and is less susceptible to resonance absorption. Lastly, the fractional change with respect to temperature in the fast fission factor contributes a +0.997 pcm/K to the total reactor coefficient in the FLiBe fueled reactor. This same contribution is +3.811 pcm/K and +5.695 pcm/K for FLiNaK and NaF-RbF, respectively. In FLiBe, this indicates that as temperature increases fractionally fewer total fission neutrons are produced from just thermal neutrons than compared to FLiNaK and NaF-RbF. Of all factors, the fractional change in the fast fission factor is the only positive contributor to the total reactor temperature coefficient. Fractional changes in the thermal fission factor and thermal and fast non-leakage factors do not contribute significantly to differences between the three fuel salts.

While all three fuel salt alternatives provide good self-limiting control behaviors from a temperature reactivity perspective, FLiBe provides the largest negative temperature coefficient and self-limiting control performance.

2.4 Future work and Summary

Specific recommended work to follow-up on this chapter might include:

- Evaluate these effects for a spent fuel composition to verify the effect of fission products and actinides on the temperature coefficient.
- Evaluate MSR reactor physics for HALEU fuels as well as Pu fuels.

- Couple distributed multi-physics dependencies within Serpent to allow thermal (e.g. temperature majorant) and neutronic performance studies to be evaluated in a simultaneous manner.

- Extend this study to evaluate fast IMSR concepts.

- Conduct a parametric temperature coefficient optimization study to obtain optimal negative total temperature coefficients to enhance power control and optimal self-limiting behaviors during postulated casualty conditions. Parameters might include fuel, moderator, and structure form factors and isotopic constituency.

- Conduct burnup of alternate salt fuels and evaluate waste stream benefits against operating physics performance to help down select a optimal fuel salt.

- Conduct a broad study of many candidate fuel salts to determine best IMSR fuel from a reactor control, core accident behavior, and waste stream manageability perspective. These down selected candidates should undergo further study to evaluate for thermal, chemical and metallurgical compatibility.

- Conduct multi-physics modeling and simulation that couples thermal hydraulic performance with nuclear physics performance. Coupling Navier-Stokes, delayed neutron precursor advection, and nuclear heating into a single, time dependent simulation will greatly improve prediction accuracy for use in MSR design, concept development, and safety analysis.

- Extend the analysis performed in this dissertation to include an analysis of alternative salts based on desirable thermo-chemical and thermo-physical and neutronic properties. While fuel economics may play a role in salt selection, a comprehensive approach is warranted to ensure life-cycle fuel performance and disposal are considered in the engineering down selection process.

In summary, the WHIP was determined to not have a significant effect of reactor coefficients. Temperature coefficient results are used throughout this dissertation and especially in chapter five. Using a direct calculational method the temperature coefficient of the MsNB is determined be -3.5 pcm/K. This is conservative evaluation as molten salt reactors generally have larger negative coefficients.

Serpent TMS treatment is an insightful tool for conducting reactor temperature performance studies. Core temperature reactivity behaviors are highly dependent on the effects of temperature on constituent isotopic cross sections. Although the IMSR appears to exhibit strong self-limiting power control in the three fuel salts cases studied, there are noteworthy differences in localized Doppler and density coefficients. In summary, FLiBe is the best neutron economy followed by NaF-RbF second and FLiNaK a close third (this difference would be overcome by increasing fuel concentration or enrichment in the salt). This is primarily due to the better moderating ratio of FLiBe and lower overall neutron absorption characteristics of salt constituents. Second, FLiBe has the strongest negative Doppler and density coefficient contributions near nominal reactor operating temperatures of 1000 K and the largest negative total average reactor temperature coefficient of the three fuel salts studied (this difference is appreciable and may inform fuel salt design considerations). Third, proportionally FLiNaK has the smallest Doppler and density contribution across the temperature range examined in the three fuel salts. This study provides insightful understanding of temperature coefficients of reactivity across a range of reactor temperatures in three representative fuel salts. The results highlight the value of the novel sensitivity treatments available in Serpent and in this case provides insightful design considerations for the MSR.

Table 2.1: FLiBe Constituents (at. %) in IMSR

U^{235} (93% enr.)	2.564x10^{-3}
U^{238}	1.930x10^{-4}
Flourine	5.758x10^{-1}
Lithium (99.992% Li^{7})	2.835x10^{-1}
Beryllium	1.379x10^{-1}

Table 2.2: FLiNaK Constituents (at. %) in IMSR

U^{235} (93% enr.)	2.643x10^{-3}
U^{238}	1.990x10^{-4}
Flourine	4.515x10^{-1}
Lithium (99.992% Li^{7})	2.005x10^{-1}
Sodium	5.029x10^{-2}
Potassium (93.258% K^{39})	1.713x10^{-1}

Table 2.3: NaF-RbF Constituents (at. %)in IMSR

U^{235} (93% enr.)	2.993x10^{-3}
U^{238}	2.253x10^{-3}
Flourine	5.048x10^{-1}
Sodium	3.632x10^{-1}
Rubidium (72.170% Rb^{85})	9.292x10^{-2}

Table 2.4: Non-Actinide Fuel Salt Capture Cross Sections

Fuel Salt Constituent	$\sigma_{capture}$ at 0.001 eV (b)	$\sigma_{capture}$ at 0.1 eV (b)
Rubidium (Rb^{85}, Rb^{87})	2.5	0.6
Flourine (F^{19})	0.07	0.008
Potassium (K^{39}, K^{41})	20.0	2.0
Sodium (Na^{23})	2.0	0.3
Lithium (Li^{6}, Li^{7})	0.6	0.02
Beryllium (Be^{9})	0.09	0.01

Table 2.5: Temperature Coefficients of Reactivity (pcm/K) at 1000 K ($\pm 1\sigma$)

Fuel Salt	K_{eff}	**Doppler**	**Fuel Salt Density**
FLiBe	0.99674 ± 0.00024	-2.562 ± 0.046	-2.186 ± 0.003
FLiNaK	0.85670 ± 0.00024	-0.540 ± 0.030	-1.737 ± 0.003
NaF-RbF	0.91756 ± 0.00024	-0.781 ± 0.027	-1.446 ± 0.003

Table 2.6: Total IMSR Temperature Coefficient (pcm/K) averaged over 773-1000 K ($\pm 1\sigma$)

Fuel Salt	**Average Temp. Coeff. of Reactivity (pcm/K)**
FLiBe	-23.814 ± 0.001
FLiNaK	-17.927 ± 0.001
NaF-RbF	-13.478 ± 0.001

Table 2.7: FLiBe Fueled IMSR - Six-Factor Physics ($\pm 1\sigma$)

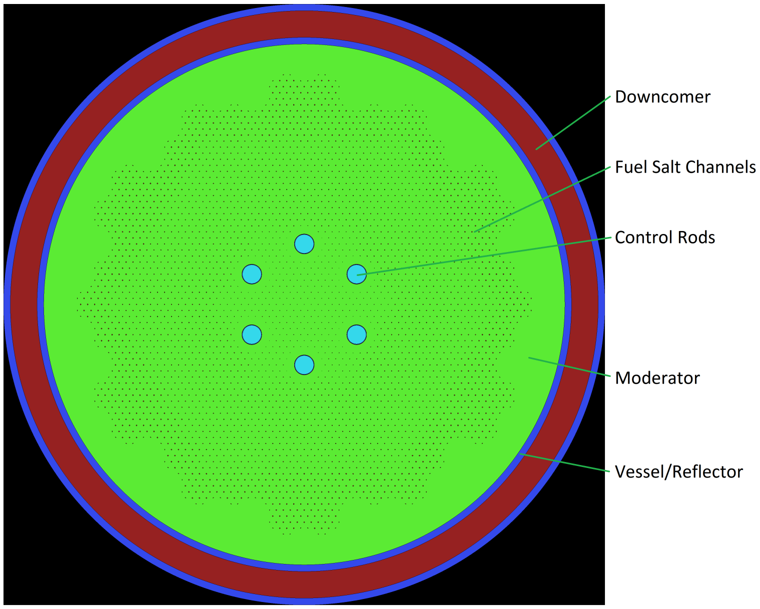
Factor	**FLiBe 773 K**	**FLiBe 1600 K**	**Temp. Coeff. (pcm/K)**
η	$1.99499 \pm 6.1\text{x}10^{-6}$	$1.97795 \pm 6.8\text{x}10^{-6}$	$-1.037 \pm < 0.001$
f	$6.40394\text{x}10^{-1} \pm 9.5\text{x}10^{-6}$	$5.92194\text{x}10^{-1} \pm 1.1\text{x}10^{-5}$	$-9.457 \pm < 0.001$
p	$7.19376\text{x}10^{-1} \pm 9.1\text{x}10^{-6}$	$7.02148\text{x}10^{-1} \pm 9.3\text{x}10^{-6}$	$-2.931 \pm < 0.001$
ϵ	$1.27895 \pm 8.9\text{x}10^{-6}$	$1.28954 \pm 9.3\text{x}10^{-6}$	$+0.997 \pm < 0.001$
$\mathscr{L}_{fnl}$	$8.34458\text{x}10^{-1} \pm 5.3\text{x}10^{-6}$	$7.76455\text{x}10^{-1} \pm 6.5\text{x}10^{-6}$	$-8.708 \pm < 0.001$
$\mathscr{L}_{tnl}$	$9.97586\text{x}10^{-1} \pm 4.4\text{x}10^{-7}$	$9.95436\text{x}10^{-1} \pm 6.2\text{x}10^{-7}$	$-0.260 \pm < 0.001$

Table 2.8: FLiNaK Fueled IMSR - Six-Factor Physics ($\pm 1\sigma$)

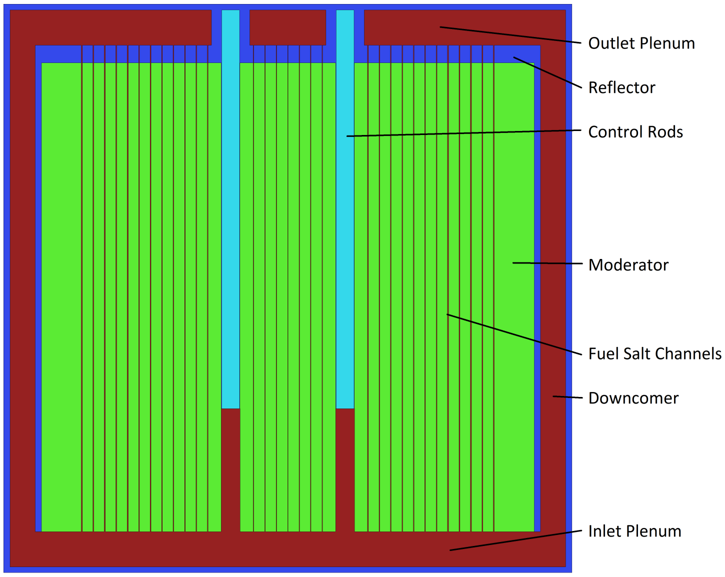
Factor	**FLiNaK 773 K**	**FLiNaK 1600 K**	**Temp. Coeff. (pcm/K)**
η	$1.52212 \pm 9.3\text{x}10^{-6}$	$1.49819 \pm 1.0\text{x}10^{-5}$	$-1.916 \pm < 0.001$
f	$6.67565\text{x}10^{-1} \pm 1.2\text{x}10^{-5}$	$6.47797\text{x}10^{-1} \pm 1.3\text{x}10^{-5}$	$-3.634 \pm < 0.001$
p	$7.63062\text{x}10^{-1} \pm 1.4\text{x}10^{-5}$	$7.30182\text{x}10^{-1} \pm 1.2\text{x}10^{-5}$	$-5.325 \pm < 0.001$
ϵ	$1.27179 \pm 1.5\text{x}10^{-5}$	$1.31252 \pm 1.3\text{x}10^{-5}$	$+3.811 \pm < 0.001$
$\mathscr{L}_{fnl}$	$8.85451\text{x}10^{-1} \pm 6.8\text{x}10^{-6}$	$8.29677\text{x}10^{-1} \pm 8.0\text{x}10^{-6}$	$-7.864 \pm < 0.001$
$\mathscr{L}_{tnl}$	$9.99775\text{x}10^{-1} \pm 1.3\text{x}10^{-7}$	$9.99309\text{x}10^{-1} \pm 2.3\text{x}10^{-7}$	$-0.189 \pm < 0.001$

Table 2.9: NaF-RbF Fueled IMSR - Six-Factor Physics ($\pm 1\sigma$)

Factor	**NaF-RbF 773 K**	**NaF-RbF 1600 K**	**Temp. Coeff. (pcm/K)**
η	$1.83344 \pm 6.9\text{x}10^{-6}$	$1.81009 \pm 7.5\text{x}10^{-6}$	$-1.550 \pm < 0.001$
f	$6.45650\text{x}10^{-1} \pm 1.3\text{x}10^{-5}$	$6.38603\text{x}10^{-1} \pm 1.2\text{x}10^{-5}$	$-1.486 \pm < 0.001$
p	$7.49130\text{x}10^{-1} \pm 1.3\text{x}10^{-5}$	$6.98873\text{x}10^{-1} \pm 1.2\text{x}10^{-5}$	$-8.394 \pm < 0.001$
ϵ	$1.19622 \pm 1.0\text{x}10^{-5}$	$1.25392 \pm 1.0\text{x}10^{-5}$	$+5.695 \pm < 0.001$
$\mathscr{L}_{fnl}$	$9.29117\text{x}10^{-1} \pm 4.1\text{x}10^{-6}$	$8.77936\text{x}10^{-1} \pm 5.3\text{x}10^{-6}$	$-6.850 \pm < 0.001$
$\mathscr{L}_{tnl}$	$9.99828\text{x}10^{-1} \pm 1.1\text{x}10^{-7}$	$9.99478\text{x}10^{-1} \pm 2.0\text{x}10^{-7}$	$-0.004 \pm < 0.001$

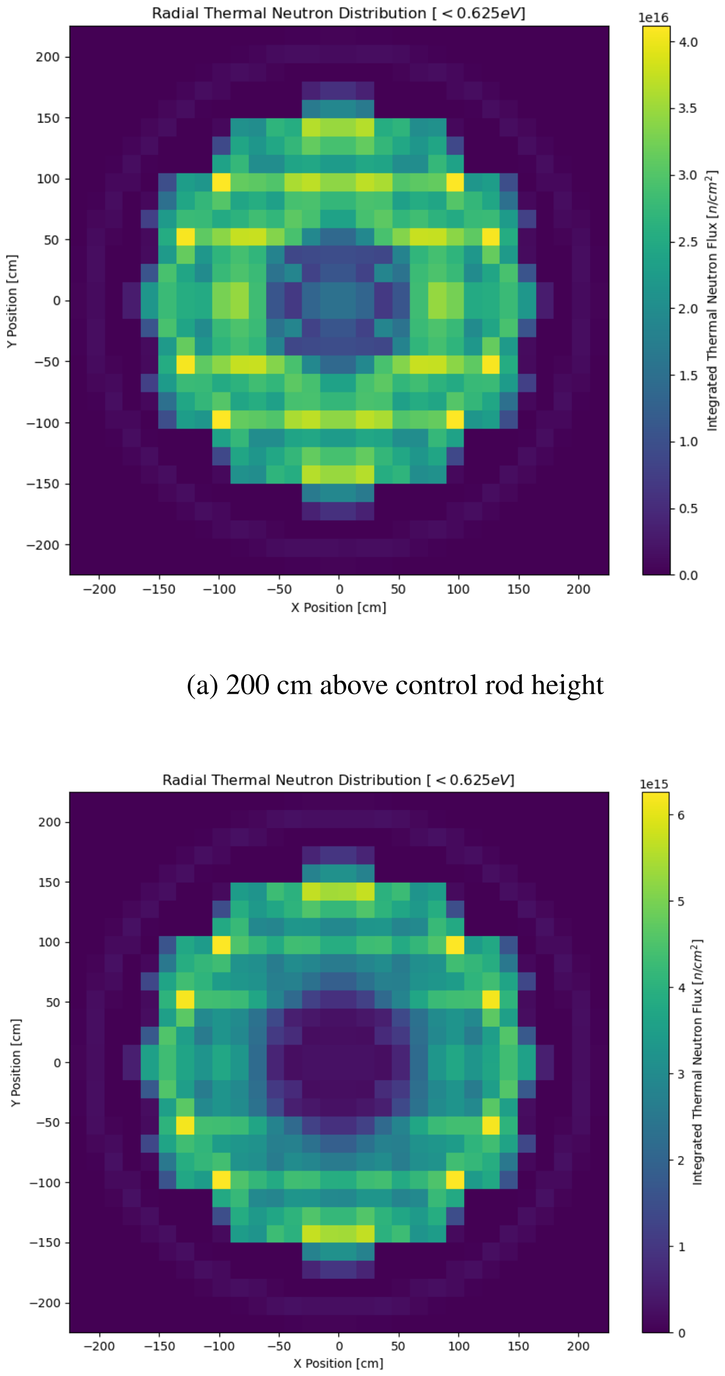


(a) Core Top View. Control rods are gadolinium, moderator is carbon, and structures and reflector is Hastelloy® N.



(b) Core Side View. Control rods are gadolinium, moderator is carbon, and structures and reflector is Hastelloy® N.

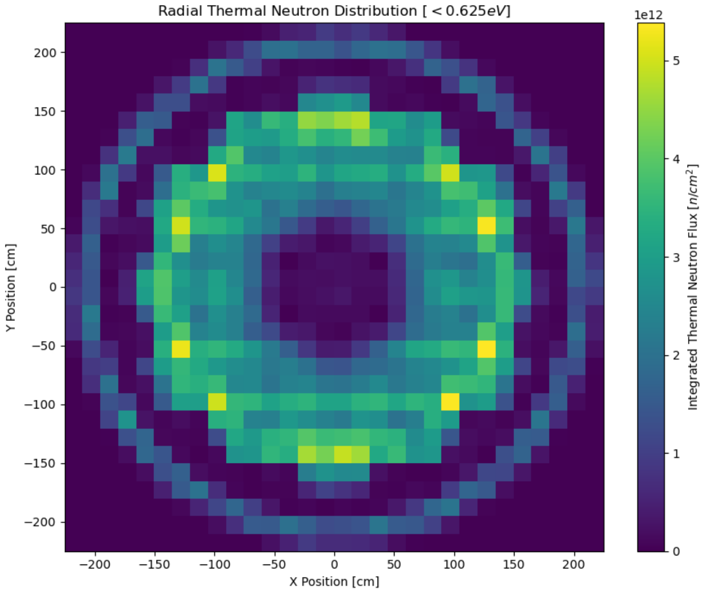
Figure 2.1: MCNP modeled IMSR geometric views. Fuel salt is indicated in red, metal structures and reflector regions in blue, graphite moderator in green, gadolinium control rods in light blue.



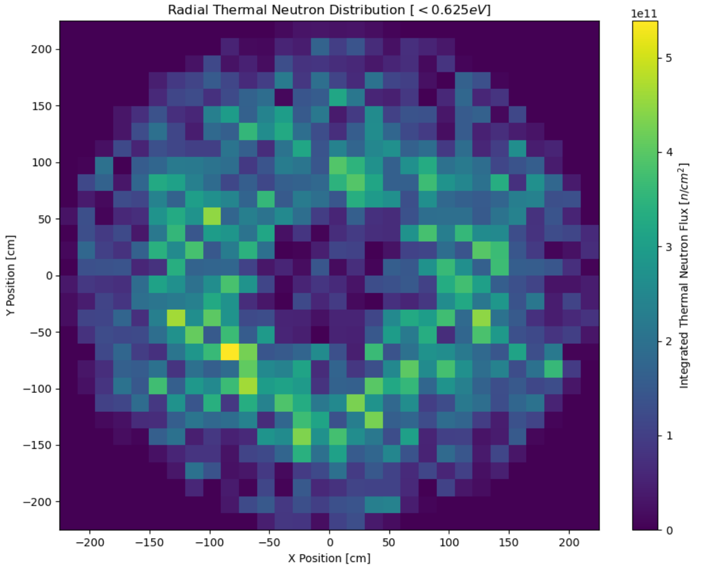
(a) 200 cm above control rod height

(b) 54 cm above control rod height. Notice the annular downcomer region is not a prominent thermal feature. Regions of greatest moderator have corresponding highest levels of thermal flux.

Figure 2.2: Axially integrated radial thermal flux distributions are shown at varying heights above the Gadolinium control rods. (Andrew Johnson et al., 2020)



(a) 92 cm below control rod height. Notice the annular downcomer region is clearly evident and the maximum thermal flux is present near the highest regions of greatest moderator and lowest in control channels.



(b) 124 cm below control rod height. Fuel inlet plenum.

Figure 2.3: Axially integrated radial thermal flux distributions are shown at varying heights below the Gadolinium control rods. (Andrew Johnson et al., 2020)

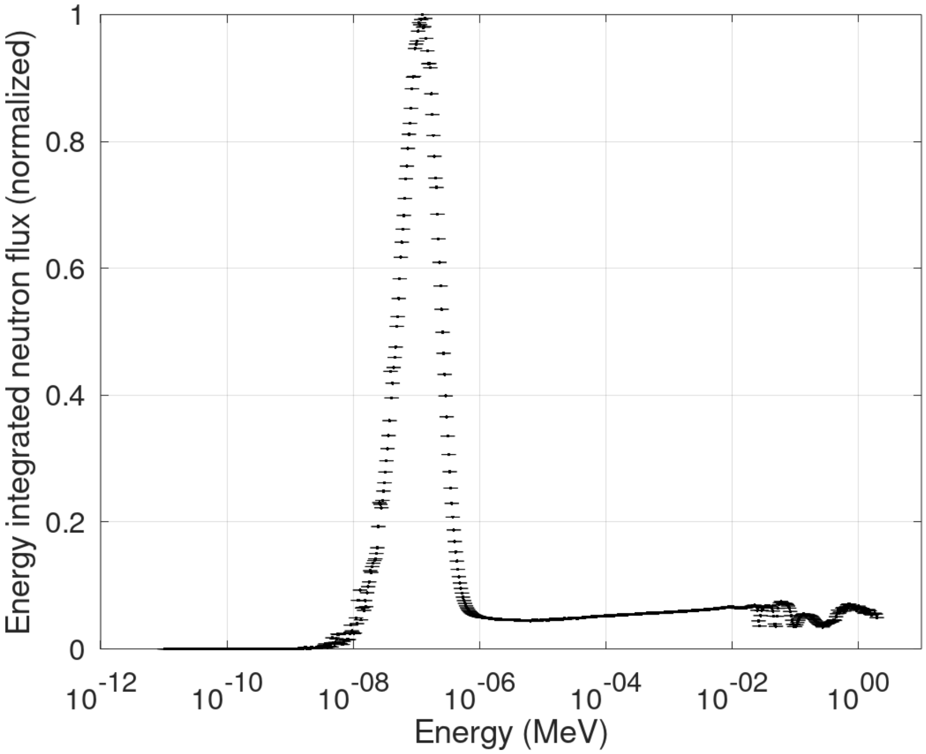


Figure 2.4: The energy integrated neutron flux spectrum for the FLiBe core with energy intervals have equal lethargy-width binning. FLiNaK and NaF-RbF fuel salt cores have similar epithermal flux spectrums in terms of peak and full width at half maximum flux. Most neutron flux lies between 0.01 eV and 1 eV.

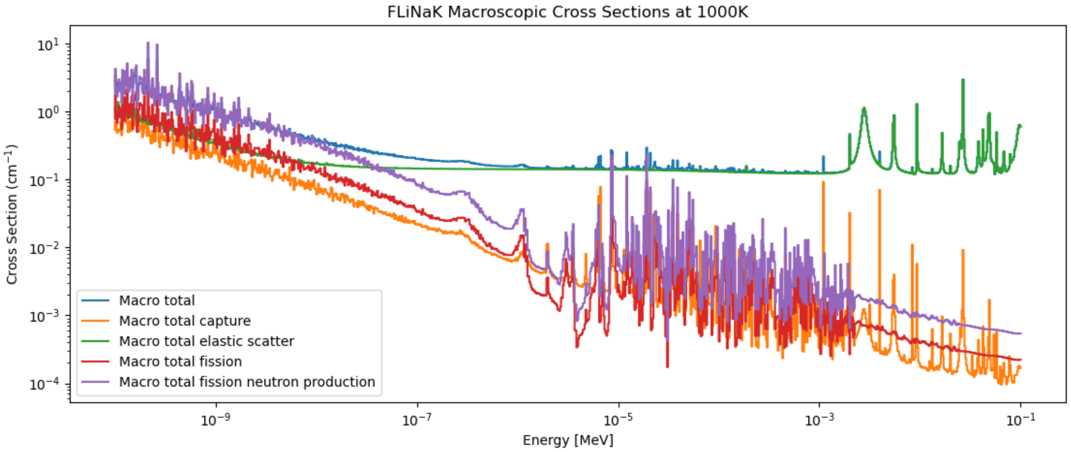
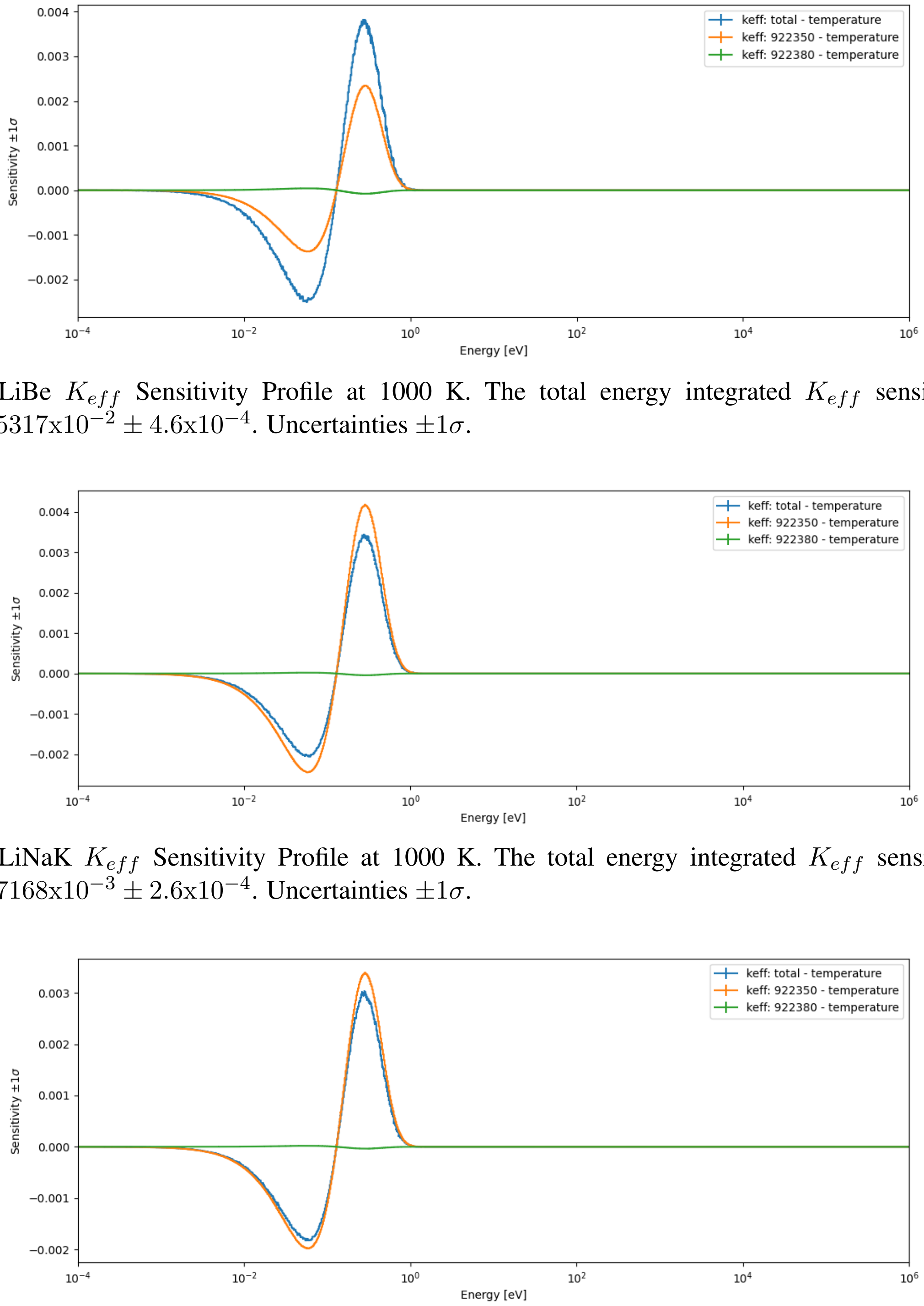


Figure 2.5: Macroscopic cross sections for the FLiNaK fuel-moderator IMSR core region. FLiBe and NaF-RbF fueled cores have very similar cross section profiles. The epithermal resonances for uranium just above 0.1 eV play an essential role in temperature behaviors in a thermalized system. Above 0.1 eV the macroscopic cross sections monotonically decrease into the fast spectrum.



(a) FLiBe K_{eff} Sensitivity Profile at 1000 K. The total energy integrated K_{eff} sensitivity of $-2.55317\text{x}10^{-2} \pm 4.6\text{x}10^{-4}$. Uncertainties $\pm 1\sigma$.

(b) FLiNaK K_{eff} Sensitivity Profile at 1000 K. The total energy integrated K_{eff} sensitivity of $-4.57168\text{x}10^{-3} \pm 2.6\text{x}10^{-4}$. Uncertainties $\pm 1\sigma$.

(c) NaF-RbF K_{eff} Sensitivity Profile at 1000 K. The total energy integrated K_{eff} sensitivity of $-7.16196\text{x}10^{-3} \pm 2.4\text{x}10^{-4}$. Uncertainties $\pm 1\sigma$.

Figure 2.6: K_{eff} sensitivity profiles to temperature perturbations for select fuel salts at 1000 K. Temperature perturbations of isotope ZAIs 922350 and 922380 contribute differently to sensitivities. The zero sensitivity positive cross over occurs at the base temperature (1000 K in this case).

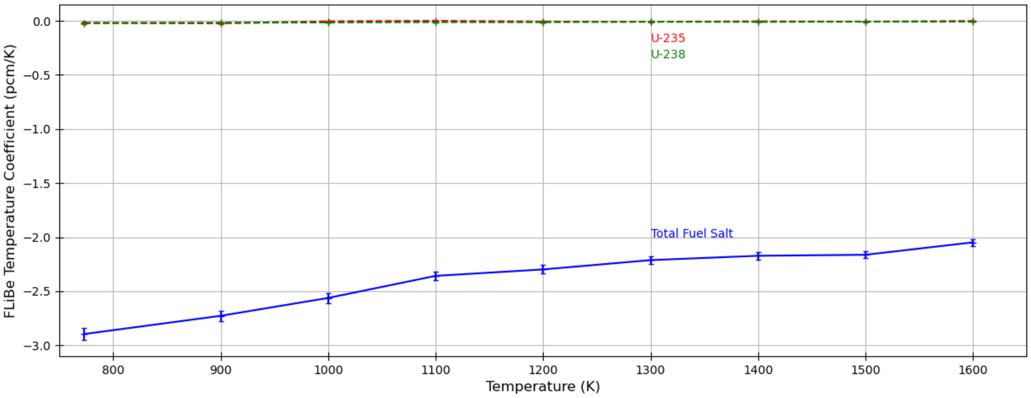


Figure 2.7: FLiBe Temperature Coefficient. FLiBe appears to have a strong negative but decreasing in magnitude Doppler temperature coefficient across the operating temperature range. U^{238} has a small negative (-0.020 pcm/K at 773 K to -0.008 pcm/K at 1600 K) but statistically significant contribution to the fuel salt coefficient. U^{235} has no statistically significant contribution.

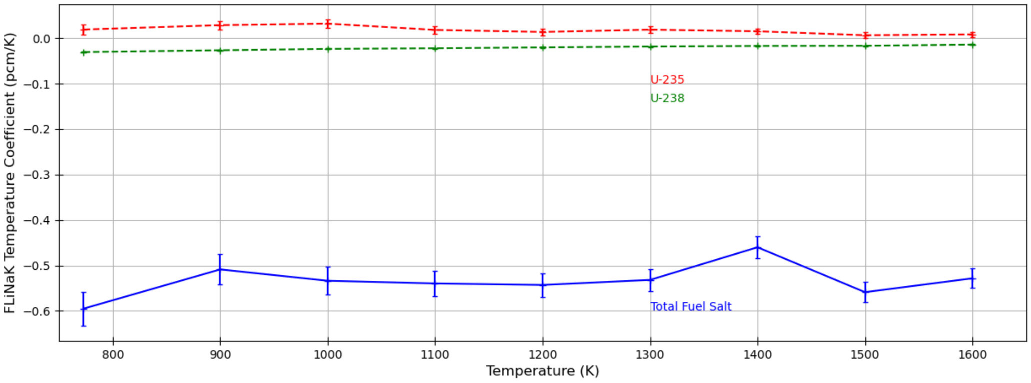


Figure 2.8: FLiNaK Temperature Coefficient. FLiNaK appears to have a relatively moderate negative Doppler temperature coefficient across the range of temperatures. U^{238} has a small negative (-0.019 pcm/K at 773 K to -0.008 pcm/K at 1600 K) but statistically significant contribution to the fuel salt coefficient, similar to FLiBe. Small contributions by U^{235} and U^{238} appear to be offsetting.

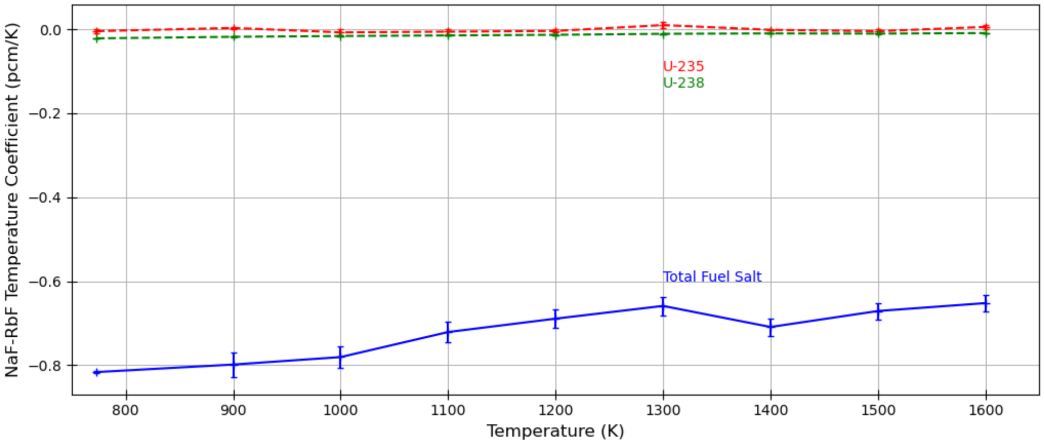
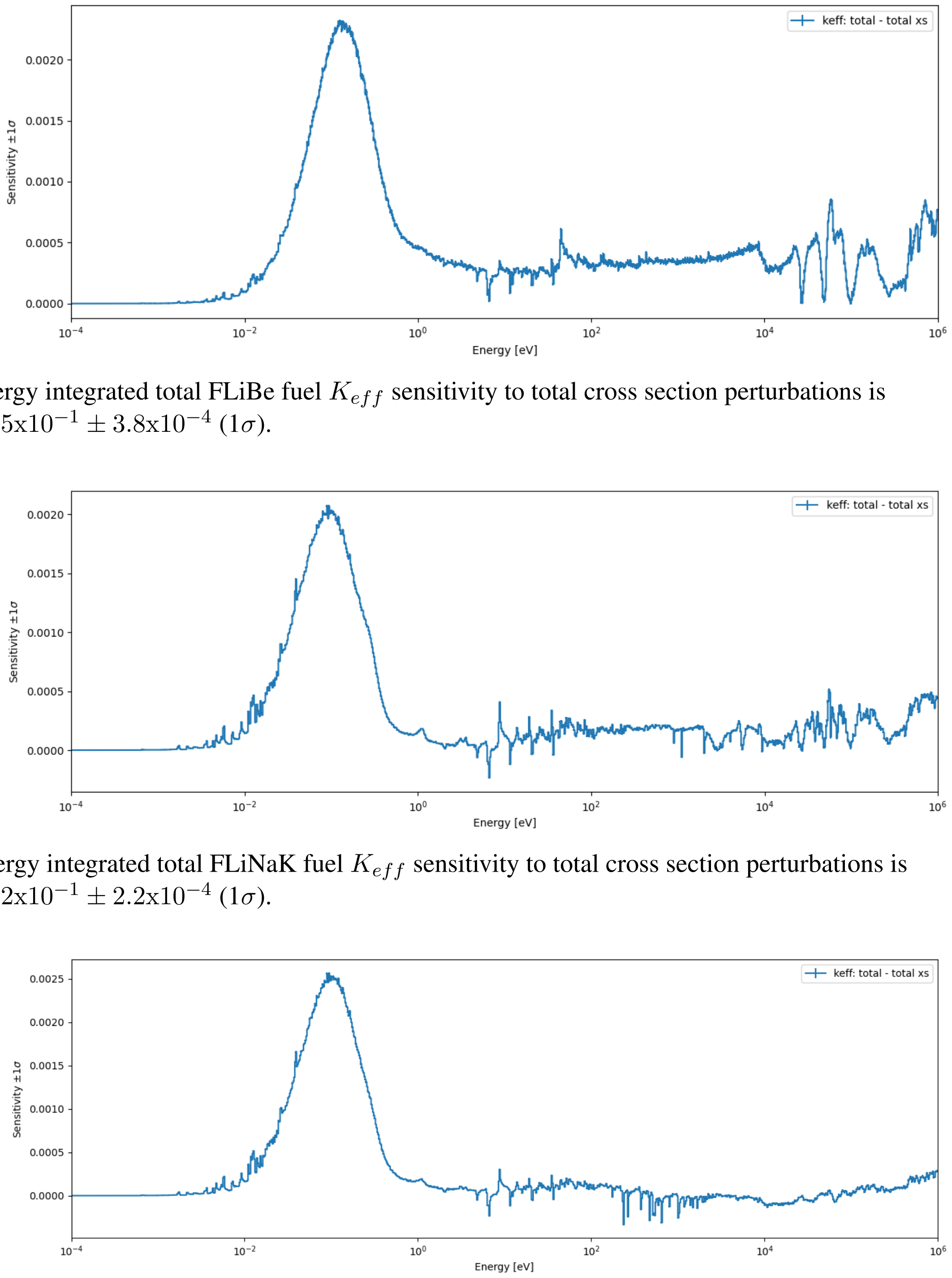


Figure 2.9: NaF-RbF Temperature Coefficient. NaF-RbF appears to have a moderate negative Doppler temperature coefficient across the range of temperatures. U^{238} has a small negative (-0.022 pcm/K at 773 K to -0.009 pcm/K at 1600 K) but statistically significant contribution to the fuel salt coefficient, similar to FLiBe. The negative coefficient appears to be influenced by non-uranium fuel salt isotopes.



(a) Energy integrated total FLiBe fuel K_{eff} sensitivity to total cross section perturbations is 4.35585x$10^{-1} \pm 3.8$x10^{-4} (1σ).

(b) Energy integrated total FLiNaK fuel K_{eff} sensitivity to total cross section perturbations is 2.93962x$10^{-1} \pm 2.2$x10^{-4} (1σ).

(c) Energy integrated total NaF-RbF fuel K_{eff} sensitivity to total cross section perturbations is 2.65358x$10^{-1} \pm 2.3$x10^{-4} (1σ).

Figure 2.10: K_{eff} sensitivity to cross section perturbations. The total integrated energy (0.0001 eV to 1 MeV) K_{eff} sensitivities are provide across the spectrum shown. Sensitivities are evaluated at 1000 K. Using Eqn. 2.9 the density coefficients at 1000 K are -2.186 pcm/K, -1.737 pcm/K, and -1.446 pcm/K for FLiBe, FLiNaK, and NaF-RbF respectively.

CHAPTER 3

Wrapped Helix around an Inclined Plane (WHIP) device

3.1 Motivation and Design Features

Fuel salt flow dynamics are strongly coupled to neutronic and thermal-hydraulic system performance. System pressure losses affect flow rate, flow rates affect core reactivity, core reactivity affects power and system temperatures, fuel salt temperatures affects core reactivity, fuel salt density and bouyant forces which in turn affects system fuel flow behaviors. The WHIP is a new invention that introduces the ability to shape the neutron spectrum, mitigate system hydraulic losses, manage in-core delayed neutron inventory, and constrain flow reactivity coefficients. The WHIP can be used in-core or near-core to affect changes to neutronics and thermal-hydraulic performance. A nominal drawing of the MsNB is shown in Fig. 3.1 and constituent material masses are provided in table 3.1. This dissertation examines the changes in reactor performance when the WHIP is installed in-core and in the riser chimney volume. General WHIP form nomenclature is outlined in Fig. 3.2. The application of this engineering form (located in-core and in exit chimney region) reduces fuel loading needs due to neutron thermalization effectiveness, reduces total reactor volume and complexity due to enhancing natural circulation flow and reducing the need to large differential thermal centers. The WHIP may also provide a practical means to monitor system performance, provide instrumentation and control functions, provide corrosion mitigations through material surface and electrochemical engineering applications. As noted in chapter four, an array of cross-correlated neutron detectors instrumented in the WHIP can provide neutronic and thermal-hydraulic system monitoring and diagnostic through-out core life. Lastly, if properly outfitted with heaters, the WHIP may provide the ability to manage fuel salt melting and cooling following transport, installation, initial critical operations, and end-of-life core cooldown and fuel salt solidification.

Some of these applications will be investigated and highlighted in this chapter. Chapters four and five will investigation the nature of neutron noise and power transient response in an MSR with a WHIP installed. Detailed discussion of the WHIP related to these topics is left to those respective chapters.

3.2 Form Factor

The WHIP has many forms but can generally be categorized by number of flights, rooted versus non-rooted, shape of flight (rectangular or cylinderical), open annular volume versus full flight radial volume as shown in Fig. 3.3. Each category can be varied by WHIP and root diameter, pitch, flight width, and helix angle. Design application of WHIP categories may depend on volumetric, insoluble deposition, chemistry management, moderator-to-fuel ratio, instrumentation, fabrication, and thermal-hydraulic (mass flow rate, heat transfer, pressure loss) considerations. Fig. 3.3(a) illustrates a generic rooted flight configuration. This configuration provides the option to package the root with instrumentation and monitoring system, fuel salt thermal management systems, and chemistry management systems, for instance. The flight number and helix angle can be design application adjusted to meet engineering requirements. Fig. 3.3(b) illustrates a non-rooted configuration that allows free vertical flow in the center core volume. This may be desirable as a hydraulic design feature. As with configurations in Fig. 3.3(a) and (c), the outer helical surface of the WHIP in this configuration would be affixed to the inner core vessel. Fig. 3.3(c) illustrates a rooted, 3-flight, cylindrical configuration. This configuration can be further modified using elliptical, oval, or pyriform features. These finer features provide potential benefit in engineering the proper radial flow distribution. Fig. 3.3(d) illustrates the rooted, 1-flight annular configuration which allows open free-flow of fuel salt in the outer radial core regions. Interaction between these two flow regions may provide thermal-hydraulic and neutronic engineering benefits. For instance, outer core regions may achieve higher flow velocities which may shape the radial neutron flux and power distribution due to changes in flow reactivity in the outer annulus. The degree of inner and outer radial flow mixing and neutron coupling would determine the magnitude of change in these important distributions.

The WHIP can be designed to have a variable number of helical rotations within a set vertical height. One full helical rotation over the full vertical height is referred to as a 1-pitch WHIP configuration. Two full rotations as 2-pitch and so forth. The hydraulic or streamline length of the fluid flow refers to the length or distance the fluid travels along its streamline path in the WHIP. This is different than the straight line or geodesic distance between two points in the WHIP. In traditional reactor designs, the fluid travels the equivalent of the vertical straight line distance up the core. The WHIP design, this flow path (streamline) is helical. Therefore, the hydraulic or streamline path evolves a larger distance than the traditional vertical geodesic taken by fluid in a traditional reactor design. Fig. 3.4 shows the hydraulic flow length as function of distance from the WHIP root for various WHIP pitch configurations. As pitch number increases, the hydraulic path increases as the hypotenuse of the radial and vertical distance traveled along the streamline. In the 6-pitch WHIP configuration, the hydraulic length can be as much as 5.3 times greater than the comparative vertical flow length.

Fig. 3.5 shows the distance between a point (referred to as the ”midpoint” in this discussion) in the fluid flow stream located half way up the core and half way between the WHIP root and flight radius. Appendix D provides the derivation of the relationship between WHIP path lengths. The blue curve represents the geodesic or straight line distance between a helical streamline and the midpoint. The distance form the midpoint is cyclic due to the helical nature of the flow path. The orange curve represents the vertical straight line distance from the midpoint. Fluid in a traditional reactor design would follow the orange curve. The gray curve represents the hydraulic distance the fluid travels toward and then away from the midpoint. For a 6-pitch WHIP the full hydraulic length of a WHIP with 179 cm vertical height (at r=15 cm) is approximately 600 cm. This figure shows that despite the large hydraulic size of the core (600 cm), the neutronic coupling more closely follows the traditional core design. In this way, a

WHIP core have the benefits associated with large flow length (flow reactivity - delayed neutron fraction) and remains closely coupled in a manner similar to a small core.

For a constrained moderator mass, the WHIP helix angle influences two principle parameters that are important to neutronics and thermal-hydraulic performance. First, the helix angle determines the net precursor outgoing flux. For small helix angles, the net out going precursor flux can be small compared to the net core channel mass flow rate. Conversely, if the helix angle is large (parallel to the WHIP axis in the limiting case), then the net out going precursor flux is equal to the net core mass flow rate. In this manner, the helix angle of the WHIP sets the effective net loss rate of precursors from the reactor core. The velocity component of the fuel salt that is parallel to the WHIP axis is denoted, $u_{\|}$. The mathematical relationship between the helix angle, the pitch, and channel width for a constant WHIP mass is given by,

$$\text{Pitch} = 2D_s\tan(\gamma), \tag{3.1}$$

where D_s is the WHIP diameter (50 cm), and γ is the helix angle. The WHIP mass is set as a constraint and is determined by Serpent 2 neutronics modeling to achieve criticality. An iterative solution method is then used to determine flight thickness for a given WHIP root diameter and number of desired pitches in the WHIP design. The WHIP diameter and number of desired pitches determines the pitch angle. The number of full pitch in the core is given by,

$$\text{Pitch Number} = \text{H/Pitch}, \tag{3.2}$$

where H is the core height (167 cm). The various 6-pitch WHIP configuration form dimensions are provided in chapter five, Table 5.3.

In practice, consecutive WHIP configurations can be utilized to ”trap” precursors near the core outlet to improve the core in-flux scattering probably. Specifically, a moderator WHIP nominally designed with several pitches resides in-core to soften the neutron spectrum, provide for a large effective core hydraulic height, reduce the asymptotic flow reactivity value, provide low hydraulic flow losses, and increase the effective delayed neutron fraction for a given flow condition. A second WHIP can be ”stacked” at the core outlet composed of a benign neutron absorbing material such a titanium and designed with a low helix angle. Although this second WHIP stage is essential transparent to neutrons, the relatively large streamline path due to the low helix angle ensures the precursors remain near the core outlet until they decay and generate their delayed neutrons. For practical purposes, the ”trap” would retain the precursors near the core for a period of about 3-5 half-lifes. Using the one-group approximation this time is about 20-35 seconds to capture 87-97$\%$ of the delayed neutrons near the core outlet. The third WHIP stage would be stacked above the second stage WHIP and purposed to reflect the delayed neutrons back into the core. This 3 stage WHIP design improves neutron economy, power stability, and reduce fuel loading for a given volume form.

The next section uses imported computer aided design (CAD) cells to validate the rectilinear mass volume approximation of the WHIP neutronics. Using curvilinear CAD forms in Serpent-2 will also provide a good steady state flux and power distribution in-core. Thsi result can then be used to gain an understanding how the effective delayed neutron precursor under circulating condition will be modified by the unique axial flux distribution imposed by the WHIP. Although the integrated flux across the core is essentially unchanged for a given core power output, the highly cross-pitch coupled flux distrubition will modify the precursor distributions exiting the core and hence modify the circulating β_{eff}.

In addition, section 4.4 will investigation the 3D thermal-hydraulic fields that exist in and around the WHIP. Axial and radial flow, pressure, and temperature effects must be considered to fully understand core physics. A capable computational fluid dynamic code such as STAR-CCM+ will be used to investigation these fields and the results will be used to inform the affect on neutronics (Siemens, 2020).

3.3 Neutronics Analysis - Serpent 2 STL geometry

The neutronic investigation was completed using Serpent 2.1.31 on Idaho National Laboratory HPC (Sawtooth). ENDF/B-VIII-0 with MCNP 6.2 photon libraries were used to complete photon coupled particle transport simulations. Appendix A provides the input files for the MsNB with CAD geometry forms. Eigenvalue were completed using material temperature and density corrections and included bound state thermal S($\alpha\,\beta$) libraries for moderator materials. Fast and thermal neutron flux, prompt and secondary emission photons were also tallied. Serpent runs included Photon-Neutron coupled interactions - (Rayleigh and Compton scattering, photoelectric effect, and electron-positron pair production) for photon energies ranging from 1 keV to100 MeV. Secondary photons are produced by atomic relaxation and bremsstrahlung, handled using the thick-target bremsstrahlung (TTB) approximation. Incorporating photoneutron interactions and production mechanisms into the transport code resulted in a +0.28% dk/k(+280 pcm) change.

Several detector reaction ENDF MTs were used in this study. Neutron reactions included total photon production, total triton production, total heat production, neutron density, and photon density. Results were also tallied at several locations of interest such as reactor top, bottom, and waistline surfaces including interior surfaces near absorber and reflectors. In all, 9 gamma tallies and 18 neutron tallies (9 thermal and 9 fast) were used in this study. Both volume and surface tallies were used where appropriate. Serpent burn runs were conducted to evaluate for fission product constituent buildup towards end-of-life (EOL) fuel burnup. Fission products were categorized by groups that included noble metals (salt insoluble), salt seeking alkalis (form stable soluble fluorides), neutron poisons, and higher actinides. These results are not presented in this dissertation but were instead used to inform a seperate project for University of Idado capstone engineering project **(Carter et al, 2021)**.

This natural circulation MsNB design was modified from previous course designs. The design was modified to minimize fuel loading by improving neutronic and thermal hydraulic performance. Specifically, the overall reactor volume was reduced by 30% by reducing the chimney height and the down comer flow area was reduced (through reduction of downcomer radial dimension to 3 cm) to maintain proper hydraulic flow system balance. This design change resulted in fuel savings of 29%. Total fuel salt mass is about 4150 kg. Total uranium fluoride mass in the fuel salt is 2240 kg. The change in excess reactivity over10 years of operation at 400 KW thermal is 0.96 %dk/k. Although this modification reduced the overall thermal driving head, overall convective flow performance was not impaired. The 6-pitch trapezoidal MsNB shown in Fig. 3.6 has a fuel salt mass of 4142.5 kg (1149.5 liters), core fuel salt mass is 2089.9 kg (581.8 liters) and the total reactor mass is 25.3 metric tons.

The Serpent model for the MsNB was also modified to include proper temperature gradient and pre-processed temperature cross section corrections for the operating temperatures of 600-700 degrees Celsius. Temperature dependent density corrections for UF4 were included for improved modeling accuracy. The fuel composition is 18mol % UF4 in a FLiNaK eutectic mixture (46.5-11.5-42) with a melting point of 454 degrees Celsius. The uranium is enriched to 19.75% w/o HALEU. Density temperature characteristics are provided by Manohar (Manohar, 2013). The fuel salt composition is tabulated in table 3.2 and WHIP core physics parameters are provided in table 3.3. UF4 solubility in the temperature range of 600-700 degrees Celsius is 23-44 mol % in the FLiNaK eutectic. With a fuel loading at 18mol% UF4, the solubility temperature is about 575 degrees Celsius. This provides a comfortable margin to fuel insolubility because of low-end temperature transient excursions. This also makes the fuel salt an attractive choice from a cost, mature knowledge of thermo-chemical properties, and counter-proliferation perspective. As noted in chapter two, there is a reactivity penalty associated with the use of FLiNaK over FLiBe, for example. The reactivity penalty can be up to 0.15 dk/k. Evaluating the risks of Beryllium's toxicity against the neutronic penalties associated with Potassium neutronics may be a design consideration.

Core structures are composed of 304 stainless steeland $\mathrm{Hastelloy}^{®}$ N, neutron reflectors in the core basket (1.5 cm thickness), chimney and control drums are Beryllium Oxide (BeO), neutron absorbers are composed of Boron Carbide ($\mathrm{B_4C}$) in the chimney and control drum volumes. The primary heat exchanger is an air-cooled Brayton cycle, offset finned heat exchanger. The primary flow is driven by natural circulation with a core differential pressure of 60 KPa which results in a fuel salt flow velocity of about 2-10 cm/s. The difference between thermal centers is about 109 cm.

Fuel Salt flow is buoyantly driven by density differences between the reactor hot and cold thermal centers. Consequently, reactor flow will be dependent on fuel salt density variations due to heat exchanger demand and system energy balance during operations. A down-power demand on the heat exchanger, for instance, will initial drive heat exchanger outlet temperature higher and gradually reducing the overall cold leg density. This gradually reduces the overall system differential pressure and consequently also reduces system and core flow until colder fuel temperatures arriving at the core results in a positive reactivity addition due the negative temperature coefficient of the core (-3.5 pcm/K). The initial reduction in core flow causes a positive reactivity effect due to fewer delayed neutron precursors exiting the core and producing fractionally more delayed neutrons inside the core. It is assumed these ”extra” neutrons would otherwise be produced outside the core and not contribute to the critical neutron economy. In a small core this may not be necessarily true as some ex-core neutron diffuse back into the core. This ”flow reactivity” effect is unique to MSR dynamic performance and must be considered in time-dependent power response of the system and safety basis analysis. Therefore, a down-power in system energy demand causes an initial reactor power increase. Conversely, if the heat exchanger undergoes a up-power demand, then the initial reactor power response is downward. This unstable, contra-variant nature of power response in a natural circulation MSR is of great interest and must be shown to be controllable under all credible operating conditions. Moreover, slow natural circulation transport times prolong reactor power response time and amplifies the power and temperature over and under shoot behaviors under dynamic powering conditions. The same is true for heat exchange down-power demands, that initially cause upward reactor power excursions due to increased system flow. Concerns for exceeding fuel salt solubility, fuel freezing, or fuel salt vaporization must be considered in the safety case basis. This time-dependent behavior will be analyzed and modeling in chapter five.

Once fuel salt flows through the core and is fission heated, the fuel flow vertically through the chimney where a fraction of neutrons that either leak from the core or are generated by the decay of delayed neutron precursors are reflected downward by beryllium reflectors that do not impede the convective flow. At the top of the chimney, the flow diverges into a horizontal plane and enters the primary heat exchanger inlet. Once in the heat exchanger, the flow continues radially outward and then downward where the flow stream is cooled, and fuel salt density increases to provide the net thermal driving head to continue the closed loop flow cycle.

Neutron density profiles (thermal flux if multiplied by average neutron velocity of 1.61x10^6 cm/s) is illustrated in Fig. 3.7 and 3.8. Regions of highest thermal flux are clearly apparent in a region just outside and concentric with the core vessel in moderating reflector material in the vicinity of the control drums. Peak thermal neutron flux is also evident in the WHIP moderator WHIP flights. The flattened and concentric thermal maximum flux (Fig. 3.8(a)) acts to improve the control drum reactivity to control core power and shutdown margin. The control drums are rotated in the reflector inward orientation and absorber outward. Rotation of the drums by 180 degrees, changes the core reactivity by -0.25 dk/k. This provides sufficiently large shutdown margin under all postulated accident conditions, i.e. loss of flow, loss of fuel salt, over-power demand, and freezing event. With a temperature reactivity in the operating range of -3.5 pcm/K there is ample shutdown margin to prevent a restart accident due to below normal temperatures down to room temperature. The axial thermal flux generally follows the traditional sinusoidal character with vanishing flux at the dore inlet and chimney inlet (core outlet). Thermal and epithermal flux peaks occur in WHIP flights. In the chimney the WHIP acts as a reflector to improve neutron economy, returning neutrons that would otherwise be lost to leakage or delayed neutrons generated in the chimney being reflected into the active core region.

At beginning-of-life (BOL), the core has approximately 0.3% excess reactivity. Moderator and fuel loading combined with in-line fuel processing to more effectively remove fission product poisons might still be optimized to provide for additional power capacity over a10-year operating cycle. Neutron flux is noticably attenuated in the chimney, the downcomer, the heat exchanger, and absorber regions. Reduced flux in these regions promotes convective flow, improves thermal efficiency, and reduces neutron fluence on structural and heat exchanger materials. Flux profiles depart significantly from those in traditional solid fueled reactor designs. The WHIP aids in shaping the neutron flux in a manner that enhances controllability, shutdown margin, control drum sensitivity, and neutron economy. While the fast neutron flux is four orders of magnitude smaller than the thermal flux, the fast flux is maximum in the active core while thermal flux is maximum at the core perimeter and WHIP flights. Maximum neutron flux for the 6-pitch trapezoidal flight WHIP MsNB configuration operating at 400 KW thermal is approximately $1.5\text{x}10^{14}$ n/cm^2-s. Maximum fast neutron fluxes at core center are on the order of $5.6\text{x}10^{8}$ n/cm^2-s. The mode neutron energy in the quasi-Boltzmann distribution is 0.167 eV (see Fig. 3.9). This corresponds to a temperature of approximately 2000 K. With core operating temperatures of 600-700 degrees Celsius, this average neutron energy is considered epithermal. Additional moderator can be added as a future design modification to evaluate for further improvement in neutronics performance.

Reactor macroscopic cross sections are shown in Fig. 3.10. The traditional 1/v thermal character is evident. The total macroscopic cross section at thermal energies indicates that the mean free path of a thermalized neutron is about 1 cm. An average neutron born fast undergoes approximately 9.5 collision between birth and thermalization. This implies that the fuel salt flow stream near the WHIP root and the outer radius of the WHIP near the flight ends (a distance of 20 cm) may be decoupled neutronically. As a result, reactivities effects due to differential flow velocities may induce radial variations in core reactivity. The next section discusses the radial dependence of fuel salt velocity and how these effects might be used to engineer optimized core physics performance.

The volumetric heat generation due to fission heating is depicted in Fig. 3.11. The hot color scheme (reds) indicates fission rate. The core and inlet plenum dominate in fission energy production. This is desirable to promote natural circulation for heat removal under nominal and accident conditions. The chimney and downcomer do not play a signifiant role in fission energy production. Decay heat at level of approximately 7% of core heat production act to mitigate flow circulation thermal driving head. Decay heat is not modeled in this investigation and may be an area ripe for future investigation particularly during shutdown and late-in-core-life accident scenarios. Figs 3.12 and 3.13 show spatially integrated thermal and fast neutron density distributions. Fig 3.14 shows the reactors temperature distribution based fission and gamma heating only. Effects due to heat exchanger cooling are not considered in this figures. The next section provides steady state temperature solutions using a CFD solver assuming uniform heat production in the core. Future work in this area might dynamically couple the multi-physics between the STAR-CCM+ solution and the Serpent fission heating solution.

Compared to a large scale Molten Salt Reactor similar to a design being pursued by Terrestrial Energy (the Integral Molten Salt Reactor, IMSR, discussed briefly in chapter one, for instance), the IMSR (larger $>$100 MW, forced circulation reactor) and MsNB (micro-sized $\approx$1 MW, natural circulation reactor) can have significant differences in noise characteristics (as described in chapter four), neutronics behaviors (as described in chapters three through five), and power transient behaviors (as described in chapter five) . The molten salt flow velocity is considerably larger due to the forced circulation flow. For larger forced circulation designs, the loop transport times are on the order of 6-7 seconds and as a result a significantly higher proportion of delayed neutron precursors decay in the loop. Delayed neutrons contribute less to the neutron economy in the large MSR design. Paradoxically, the flow reactivity is less sensitive to changes in fuel salt flow velocity in the large design compared to the MsNB because salt flow rate is much smaller in the MsNB. The physics of this phenomena will be explained in this chapter. The neutron diffusion length, D, for both MSR systems are roughly the same. This is expected given the similar absorption cross sections of the two fuel salts used in the design. Although the absorption cross section of FLiNaK is about twice as large compared to NaF-RbF-UF_4 fuel salt notionally used in IMSR. This is principally due to the large absorption cross section of potassium. The IMSR neutron energy is fully thermalize at 600 degrees Celsius while the MsNB is slightly higher as explained earlier.

The effective delayed neutron fraction is smaller in the MSR than in the traditional solid fueled system since a portion of the DNPs flow out of the core. Some fraction of precursors decay outside the core and these neutrons do not contribute to fission reactivity. In the case of the μscale-MsNB, where L = 382 cm and H = 166 cm, the effective delayed neutron fraction is about 30% of the analogous static fuel system. The static delayed neutron fraction calculated by Serpent is 0.00724. In the flowing system, the effective neutron fraction is closer to 0.00690. Approximately -34 pcm results from a nominal flow rate of 5 cm/s at 400 KW power demand. This reduced fraction has a small effect on reactor kinetics during transients as fewer prompt neutrons are required to achieve prompt criticality. The core transfer function amplitude is also approximately 3.3 times greater than the analogous traditional static fuel reactor system i.e., response to neutron flux noise is greater as discussed in chapter four.

Temperature coefficient of reactivity calculation shows MsNB is inherently self-limiting power behavior with a temperature coefficient of -3.5 pcm/K using direct six-factor neutron life cycle calculation. Additional methods using Serpent target motion sampling (TMS) techniques as discussed in chapter two. Compared to larger MSR designs where the temperature coefficient can be approximately -17.9 pcm/K. MSRs generally have smaller temperature coefficients compared to traditional solid fueled systems. Future work may include an investigation into how to improve the negative temperature coefficient in the operating off-nominal operating regions.

In concept, the reactor has five configurations that support a controlled and safe start-up. Configuration 1 (C1) is the reactor unloaded with absorber cans rotated inward for maximum neutron attenuation. This is the safest sub-critical reactor configuration. Configuration 2 (C2) is the reactor loaded with fuel salt with absorber cans rotated inward for maximum neutron attenuation. Neutronics analysis indicates the reactor has a Keff 0.75 in this configuration. Configuration 3 (C3) is the reactor loaded with fuel salt with absorber cans rotated inward for maximum neutron attenuation and internal electric heaters located in the downcomer, plena, and core WHIP stem and blades are energized and melt the fuel salt. Configuration 4 (C4) is the reactor loaded with fuel salt with control cans rotated to achieve criticality. Fission heating will continue the melting process and raising fuel salt temperatures to low in the nominal operating band, i.e. 600-620 deg Celsius. Electric heaters can be maintained ON or turned OFF during C4. Configuration 5 (C5) is the reactor fuel salt melted, critical at operating temperature, with no heat demand. Configuration 6 (C6) is similar to C5, but heat demand is introduced in the system to control natural circulation flow initiation. Each configuration is a progression from fuel salt loading to full power reactor operation. Transitions from Cn to Cn+1 is controlled by Documented Safety Basis (DSA) Administrative Controls (ACs).

A second master's student (Trevin Lasley, University of Idaho 2021) independently reviewed the Serpent input file and verified physics and fuel salt thermo-physical properties used in the analysis. The Serpent code was benchmarked with available Jezebel and Godiva eigenvalue benchmarks.

3.4 Thermal-Hydraulics Analysis - STAR-CCM+

Siemens Digital Industries Software STAR-CCM+, a commercially available computational fluid dynamic (CFD) simulation modeling tool, was used to conduct a thermal-hydraulic evaluation of the the MsNB with WHIP (Siemens, 2020). The evaluation includes a study of MsNB flow, temperature, and pressure fields present in the MsNB during buoyant convection flow conditions. STAR-CCM+ is a verified modeling tool that provides high confidence simulation results that can be used to inform core neutronic and system flow performance. The investigation involves using computer aided design of six variations of WHIP helix angles. Each WHIP variant maintains a constant mass form. The six variants include one helical pitch, two helical pitches, up to six helical pitches. This variation in design allows for a study of core differential pressure, differential temperature, flow velocity field, local effects pertinent to engineering design.

For a given MsNB WHIP design, a parametric study of changes in molten salt thermo-physical property effects on thermal-hydraulic performance will be examined. For instance, changes in core differential temperature, pressure, and flow performance will be studied using the 6-pitch WHIP MsNB by varying material fluid properties such as dynamic viscosity, thermal conductivity, specific heat. The database of thermal physical properties for Uranium Fluoride loaded FLiNaK is very limited. A parametric analysis will therefore very useful in determining solution sensitivity to variations in less well known fuel salt properties. This will bound system performance based on uncertainty in fluid properties and may help inform future research priorities.

The CFD investigation used the MsNB computer aided design (CAD) developed by Alberto Cardenas Melgar while attending the University of Idaho as a graduate student. The six WHIP variants were designed and engineered into CAD usable assemblies by the author. The WHIP assemblies were incorporated into the MsNB CAD model and complemented into the MsNB fluid body by Jan Lambrechsen also a graduate student at the University of Idaho. The six design variants were imported into the STAR-CCM+ CFD software using a 3D model file format called step (standard for the exchange of product data), an ISO standard exchange format. A volumetric heat source was used in the active core region of the fluid body while the heat exchanger-to-secondary heat transfer surfaces were used as the system heat sink. At the cross-flow heat exchanger, constant heat flux was the assigned as an acceptable boundary condition. System surfaces unrelated to active heat generations and removal were assigned adiabatic boundary conditions.

The STAR-CCM+ automated mesher was used to complete an adaptive mesh of the fluid geometry. A surface remesher, polyhedral mesher, and prism layer mesher were used with triangular curvature and proximity refinement. The mesh base size is 8 mm with a minimum surface size of 10% of base size setting. Target surface size was set to 100% of base size of 8 mm. This meshing regime resulted in an approximate 3.54 million cells, 17.23 million faces, and 12.73 million vertices per meshed MsNB pitch variant. This meshing scheme adequately modeled the necessary features of the MsNB design while avoid costly computational overhead. A mesh sensitivity was completed using Richardson extrapolation method (Richardson, 1910)(Ferziger J. H. et al. 2002) Using a mesh base size of 3 mm, 6 mm, and 12 mm. The discretization error for mass flow rate and core differential temperature at a mesh base size of 8 mm is 6.9% and 6.1%, respectively. The wall y+ value is a dimensionless parameter similar to a local Reynolds number that can used to characterize flow near a wall boundary surface in a meshed CFD model. The magnitude of the wall distance y+ value determines the relative importance of viscous and turbulent shear forces near wall boundaries. For y+ values less than 5, the frictional contributions to the total stress are due to viscous shear stresses near the wall region. The distribution of wall y+ values in the STAR-CCM+ model indicates that maximum y+ values are around 5 with a mode of 3.5. This implies shear stresses in the near wall regions are due to viscous forces, or a viscous sublayer and the fluid is dominated by viscous effect in the near wall. Reynolds shear stress is considered negligible. The wall function (boundary conditions applied to boundary patches of type wall) follows a linear velocity law. Therefore a uniform mesh scheme is appropriate for use in the MsNB model wall function (Lui F., 2016)(Salim S.M., 2009).

Modeled physics continua in the molten salt fluid accommodates tangential drag (viscous resistance) with a standard wall treatment. Prism layers are used to resolve near wall flow features accurately. To resolve the viscous sublayer near wall boundaries $y+$ wall treatment is used. Gravity is introduced into the model buoyant forces associated with natural circulation flow. The two-equation K-Omega turbulence model is used to solve the turbulent kinetic energy and specific dissipation rate to determine the turbulent eddy viscosity. The K-Omega turbulence model is well suited in low Reynold's number fluid environments. Exponential and polynomial functions are used to represent the thermal-physical fluid properties of dynamic viscosity, density, specific heat, thermal conductivity, and Prandtl number (ratio of momentum diffusivity-to-thermal diffusivity). Polynomial density relationships available in chapter five are used in this model. Other thermo-physical properties are provided by the compendium work of Manohar Sohal, et al. (Sohal, 2013) (Thoma, 1959). Segregated, Reynolds-Averaged Navier Stokes solvers were used to obtain steady-state, three-dimensional results for fluid temperature, velocity, and pressure fields in the MsNB. This dissertation focuses on the core thermal-hydraulics so much of the nuances associated with heat exchanger flow behaviors is not explicitly discussed in this manuscript. Reference physics values of gravity, density, and temperature are 9.81 $\mathrm{m/s^2}$, 3000.0 $\mathrm{kg/m^3}$, and 893 Kelvin, respectively.

Initial conditions include a uniform pressure field of 170 KPa, a uniform temperature field of 908.0 K, a dimensionless turbulent intensity value of 0.1, and velocity of $10\ \mathrm{cm/s}$ in the +z (upward) core direction. The volumetric heat generation source is defined to be a uniform (constant) heat source over the fluid volume of the WHIP. The assumption is local variations in the heat generation rate near boundaries does not have a significant effect on system flow, pressure, or temperature performance. The actual volumetric heat generation distribution is a function of the thermal neutron flux distribution. For this reason, the multi-physics coupling between neutronics and thermal-hydraulics is essential to improving on the modeling precision and accuracy of future work. The model consistently achieved acceptable solution convergence after 10,000 solution iterations. Residuals were typically on the order of 10^{-2} to 10^{-4}.

Reports and monitors were created on core differential pressure, core differential temperature, heat exchanger power (a field functional was used to tie heat generation with heat exchanger hear removal to ensure system heat balance was manifest), mass flow rate, 3-component momentum, single point fluid velocity, turbulent kinetic energy (Tke), and turbulence specific dissipation rate (Sdr) (Siemens, 2020) . Results were plotted in 2D and 3D formats including transverse and axial velocity in core and at core exit as well as radial and axial velocity, temperature, and pressure 3D plots and scenes.

The thermal-hydraulic investigation that follows includes the use of six WHIP design variants each with an integer increment in pitch. The WHIP material mass is maintained at 238.8 kg of graphite moderator. This mass values was obtained utilizing Serpent 2 mass equivalent geometry to solve the neutron multiplication factor eigenvalue. For each WHIP pitch configuration, the steady-state thermal hydraulic solution is obtained for reactor power levels from 400 KW to 2000 KW in 400 KW increments, e.g. 800 KW, 1200 KW, et cetera). Thermo-physical properties are also varied by a factor of 2 (50% to 100% of fuel free eutectic values) to determine system thermal-hydraulic sensitivities to changes in key properties. Results of varied steady-state simulations are presented in the sections that follow.

3.4.1 Flow Field

Extensive study was completed on the MsNB flow field. In particular, the nature of the fuel salt flow field near and in the core is essential to understanding delayed neutron precursor advection in and out of the active core region. Local and global system flow effects have a significant impact on core reactivity and therefore are of great interest in predicting reactor physics behavior. The introduction of the WHIP in the core imposes transitions between rectilinear and rotational flow regimes. These transitions occur at the inlet and outlet of the core and have the potential of producing flow turbulence and circulation eddies. The degree of turbulence in the flow is evaluated by calculating the local turbulent kinetic energy of the fluid. While laminar flow is typically preferred in well-behaved thermodynamic systems, the presence of turbulence and eddies might enhance heat transfer and fluid mixing and in the case of MSRs may promote delayed neutron precursor trapping in the chimney volume. This trapping effect may result in reduced non-leakage of delayed neutrons and improve overall neutron economy of the reactor system. Analysis of the CFD simulation results in the section will show that the MsNB flow is turbulent and experiences circulation eddies in localized regions. Further refinement of the closed-loop flow design may be warranted to mitigate or enhance some of the flow effects illustrated in these results. The investigation will be primarily interested in the component of fuel salt velocity that is normal to the inlet and outlet core plane. The normal component of the fluid velocity is most closely correlated to the rate of delayed neutron precursor loss from the active core. In addition, the normal component of fuel salt velocity is proportional to the change in fluid mass flux in the core control volume. For this reason, the z-component of the fluid velocity were evaluated under various power and thermo-physical environments. Vector representations of the velocity magnitude were also evaluated to determine fuel salt flow character globally (reactor cross section) and locally in the chimney at core outlet.

Important aspects of the fuel salt flow in the WHIP channels were also evaluated. Specifically, the radial and axial dependence of flow velocity along a representative monitor line was analyzed. The dependence of flow on the distance from the WHIP core may affect radial neutron flux distributions in cases where the core diameter is large compared to the characteristic neutron diffusion length. This is not expected to be the case in the MsNB due to the degree of fluid cross-streamline mixing and small circumferential waistline of the core.

At the system level, the reactor fuel salt flow field is evaluated by examining reactor cross section representations of the fuel salt velocity streamlines. Fig. 3.15 shows this representation for the six pitch variants evaluated at a steady-state system power of 400 KW. Each pitch variant has a mass equivalent moderator material to ensure core criticality is maintained. With increasing pitch number, the WHIP design incurs an additional whole number increment in the helix rotation. For instance, the 3-pitch WHIP has three helical rotations between core inlet and outlet. Increasing the number of pitches reduces the helix angle and the total streamline path length for the fuel salt in the core. While the color scale between illustrations in the figure vary between pitch configurations, the system flow velocity decreases as the pitch number increases due to the increase in the system flow resistance with higher pitch number. As the pitch number increases, the regions of highest system flow occur at the WHIP inlet. The region near the inner surface of the first full helical rotation experiences the highest steady-state system flow rate due to the flow transition between the inlet plenum and the rotational flow manifest in the WHIP. This flow transition induces an eddy that reduces channel effective flow area and results in increased flow velocity in this localized region low in the core. The radial dependence of the flow field with increasing pitch number changes in a non-linear manner. This makes flow behavior prediction challenging without evaluation of the specific design under consideration, i.e., small changes in the WHIP form factor may result in large changes in overflow field character.

As expected, the flow field is dominantly upward in the core and chimney regions, radial in the inlet and outlet plenum, and downward in the downcomer region. Results show that the flow is generally well-behaved (locally-irrotational). The offset-fin heat exchanger experiences higher than average flow velocities at the inlet and outlet (downcomer inlet) due to step reductions in the flow area near those control volumes. As the pitch number increase, flow vorticity increases and therefore centripetal forces promotes higher flow velocities on outer radial flow surfaces. This can be seen in Fig. 3.15(f) in the core and chimney regions.

At 400 KW, the system average flow rate is on the order of 2-4 cm/s which corresponds to the flow velocity results modeled by the transient power code discussed in chapter five. Velocity variance within the system can be large. For instance, in the 6-pitch configuration the minimum fuel salt velocities can be near zero in the turbulent chimney regions while the maximum velocities of 14-19 cm/s occur in first pitch channel, low in the core. This is shown in Fig. 3.15 (c)-(f). This volume of locally higher flow velocities does not result in adverse thermal conditions low in the core due to the relatively small amount of volumetric heating that has occurred low in the core.

The chimney experiences turbulent flow at all WHIP pitch configurations. While upward spiraling flow dominates near the chimney walls, eddies are prevalent throughout the interior chimney control volume. The separation of these distinct flow regimes becomes more apparent as the pitch number increases (flow vorticity increases). From a core physics perspective, this separation in flow regime may improve neutron utilization as delayed neutron precursors involved in the interior turbulent region of the chimney may have a higher resident time than precursors that spiral upward, relatively unimpeded, into the heat exchanger. This higher resident time translates to an increased probability of generating delayed neutrons that participate in active core region fissions. Using reflector materials in the chimney region (including the optional addition of a WHIP in the chimney fabricated of reflector materials) would further increase this neutron economy.

Localized areas of low flow can present thermodynamic and chemistry challenges over core life. Low flow areas can, for instance, lead to differential deposition of insoluble metallic fluorides which can lead to undesirable material chemical corrosion interactions or radiochemical deposition of local sources that might present costly and radiological challenges during end-of-life reactor vessel disposal. As illustrated by Fig. 3.15, localized flow velocity variations by up to a factor of five can occur in the closed loop. Although this dissertation will not delve into optimum design features to achieve long-term system wide flow balance, careful consideration of flow design features might include core inlet chimney features, curvilinear heat exchanger flow area profiles that mitigate inlet and outlet channeling, or helical vanes in the downcomer to mitigate flow transitions in inlet plenum and WHIP regions.

Fig 3.16 shows the reactor cross section representation of the axial component of the fuel salt velocity. The axial component is important to evaluating the fuel salt mass flux into and out of the active core volume. Although the average overall flow velocity decreases with pitch number for a constant power demand of 400 KW (due to increased system flow resistance), the maximum axial component of fuel salt flow velocity increases with pitch number (in localized regions). Two dominate flow features are worthy of mention. First, the lower third of the WHIP core experiences non-conforming helical flow due to the transition from inward radial flow in the inlet plenum to upward rotational flow in the core. This transition induces a torque on the newly formed rotational flow which may create an eddy in the axial plane soon after the flow transition occurs. The second feature is the rotational flow exiting the core and flowing up the chimney is dominate near the chimney walls. Flow in the interior of the chimney is more disorganized and subject to turbulent mixing. Both features can be mitigated, if desired, with the use of novel design features at the core inlet and outlet. Fig 3.17 shows the z-component (axial) of the fuel salt velocity at the mid-point in the chimney region.

Close examination of Fig 3.17 shows that as the pitch number increases, the positive axial component of fuel salt velocity in the chimney region is a maximum near the chimney walls. The upward flowing fuel salt layer thins with increasing pitch number due to the vorticity of the core exiting flow. Maximum velocities along the chimney walls can exceed three times to surface averaged fluid velocity. The interior regions of the chimney can have near zero or negative axial flow velocity as eddies and turbulent mixing dominate this interior volume. The turbulent kinetic energy in the chimney is illustrated in Fig 3.18. The radial distribution of turbulent kinetic energy is similar for the six WHIP pitch configurations examined. The maximum kinetic energy is also similar for pitch configurations examined. This interior chimney region also corresponds to the closed-loop system highest steady-state temperatures. While the turbulent region can have neutronic and thermal-hydraulic benefits, these flow effects can be modified by incorporating an ”inverted-cone” feature to the chimney to reduce local eddies and promote unidirectional flow into the downstream heat exchanger heat transfer surfaces. An inverted-cone feature centered on the core axis may also provide secondary engineering benefits that aid in reducing flow losses and active electrochemical management systems in the flow stream.

Fuel salt velocity integrated magnitudes are shown in Fig 3.19. Velocities in the chimney region vary from zero to approximately 14 cm/s near the chimney walls in the 6-pitch WHIP configuration. Maximum local velocities increase with pitch number. The azimuthal variance of fluid velocity decreases with pitch number. That is, the higher flow rates near the chimney wall at higher pitch numbers ensures the fluid is uniform in this annular flow volume. Close examination of the velocity streamlines shows evidence of interior rotational vortices and eddies. As expected, flow patterns near the chimney are concentric with the chimney wall. The axial representation of these eddies is shown in Fig. 3.17. While the general character of the velocity profiles is similar across different WHIP configurations, the localized velocity and turbulence behavior is strongly dependent on WHIP features and must be carefully considered in any final design.

The radial dependence of the z-component of the fuel salt velocity provides on understanding of the possible effects flow velocity has on modification of the radial neutron flux distribution in the core. In the one-dimensional case, the axial component of the fuel salt velocity is a factor in determining the flow reactivity. As the fuel salt velocity increases, the rate of loss of delayed neutron precursors from the active core volume also increases. This results in more delayed neutron precursors being transported out of the core and generating their delayed neutrons outside of the core. In a two-dimensional system where fuel salt velocity is a function of both axial and radial position in the WHIP channel, flow reactivity effects also have a potential radial dependence proportional to the amount of neutron transport coupling. For small, tightly coupled cores this radial dependence is likely negligible. For larger cores, where the core radius is several factors larger than the average neutron diffusion length, the degree of decoupling can be considerable and radial effects of flow reactivity must be considered. The diffusion length for the MsNB is on the order of 3 cm and therefore streamlines with a separation more than 15-20 cm may result in modifications to the radial flux distribution in the core. Figs 3.20, 3.21, and 3.22 show the radial and axial dependence of the axial fuel salt velocity in the WHIP channel. Fig 3.23 shows the dependence of inner wall fuel salt axial velocity as a function of power demand (system flow rate) and pitch configuration. For a constant power demand of 400 KW, an increase in pitch number typically results in more fluid shear and therefore the radial velocity dependence is increasingly negative. The 1-pitch WHIP configuration, for instance, has an inner wall average fuel salt velocity of approximately 4.2 cm/s and a rate of velocity change in the radial direction of 0.02 cm/s per cm from the WHIP core. This dependence is positive and therefore the axial component of fuel salt velocity increases as the distance from the WHIP core increases. Pitch number appears to be a poor predictor of radial velocity dependence. Specifically, the 3-pitch, 4-pitch and 6-pitch WHIP have strong negative radial flow dependencies while 1-pitch, 2-pitch, and 5-pitch have slight positive or neutral radial velocity dependencies. The 4-pitch WHIP configuration has the strongest negative radial velocity gradient that becomes more negative with increasing power demand on the system. Specifically, the 4-pitch WHIP has a radial velocity gradient of -0.12 cm/s per cm at 400 KW decreasing linearly to -0.5 cm/s per cm at 2000 KW power demand. Pitch number is also a poor predictor of inner channel axial velocity due to rotational flow formation in the WHIP. In all pitch number cases, the inner channel axial velocity increases with heat exchanger power demand. That is, inner channel velocity increases with power demand. Non-laminar flow would generally tend to increase cross streamline mixing and would increase cross stream shearing in the channel which would result in a negative radial velocity gradient in the channel.

The reactivity effects associated with this radial dependence of fuel salt velocity can be quantified using the work presented in chapter five. At 400 kW, the 1-pitch configuration has an inner axial fluid velocity of 4.2 cm/s and an outer velocity of 4.8 cm/s (Fig. 3.20(a)). In a large core with a fair degree of decoupling, this differential velocity may result in a reactivity difference of -16.5 pcm between the inner and outer core (inner stream less reactive). Although this would not likely be the case in a highly coupled, small MsNB reactor the effects of this radial dependence should be evaluated. These radial effects become less significant as pitch number increases (note in Fig. 3.22 that velocity gradient generally decreases with pitch number). In the 6-pitch configuration, the inner velocity is 3.5 cm/s and the outer velocity is 2.1 cm/s. This results in a difference in reactivity of +19.8 pcm (outer stream more reactive). These results assume a large core (radius much greater than the neutron diffusion length) and minimal mixing between adjacent fuel salt flow stream in the WHIP channel.

Fig. 3.24(a) and Table 3.4 illustrates the system mass flow rate dependence on both pitch number and heat exchanger power demand. Mass flow rate increases from 1-pitch to 3-pitch

WHIP configurations and decreases for all higher pitch configurations. The decrease in system flow rate at higher pitch numbers is primarily due to the increase in local channel turbulence at the core inlet. The 3-pitch WHIP configuration provides the highest system mass flow rate (75.8 kg/s) at 2000 KW power demand and core differential temperature of 21.7 C. Fig. 3.24(b) illustrates the commensurate change in core differential temperature for the various pitch and power configurations.

In principle, the WHIP's cross sectional flow area can be modified to promote higher flow rates near the WHIP root (inner surface). This may result in further flattening of the radial flux distribution that can promote improved control drum physics worth. The flow area depicted in Fig. 3.6(a) for instance is trapezoidal in form and promotes additional neutron moderation near core center.

In summary, the flow field tends to be non-linear, turbulent, with local eddies low in the core at high pitch numbers (>3). This result emphasizes the need to use accurate modeling and computational methods to evaluate system performance on a case basis. The MsNB seems to benefit neutronically and thermal-hydraulically by using the WHIP in-core. Flow fields have no adverse thermal effects in the core. Localized low flow regimes exist near the core inlet, in the interior (center line regions) of the chimney, and low inner heat transfer surfaces of the heat exchanger. These identified low flow regions can be easily modified by engineering design and engineered to maintain fuel salt velocity closer to nominal values to mitigate insoluble deposition and adverse thermal conditions near the heat exchanger.

3.4.2 Pressure Field

The system pressure field is dominated by the gravity pressure gradient on the fuel salt. The core differential pressure varies from 61.895 KPa at 400 KW and increases to 61.915 KPa at 2000 KW for the 1-pitch configuration. This differential pressure changes to 61.896 KPa at 400 KW to 61.883 KPa for the 2000 KW 6-pitch configuration. Although the flow resistance increases slightly as pitch numbers increase due to the increase in surface frictional forces, pressure loss in the system is dominated by the heat exchanger. For this reason, the core differential pressure varies only by approximately 20 Pa over a power demand change of 1600 KW. The nominal system pressure for the system is set at 170 KPa. The difference in working pressure across the core is about 2200 Pa.

3.4.3 Temperature Field

The temperature distribution in the reactor system at steady-state flow conditions is shown in Fig. 3.25. Nominal system temperature was set at 893 K (620 C). As expected, the highest system fluid temperatures are in the chimney. The uniform volumetric heat generation rate in the core results in an axial temperature increase as the fuel salt flows upward through the core and into the chimney. Due to the rotational flow profile in the chimney, the highest chimney temperatures are located in the interior (near the chimney center where eddy flow is dominate) of the chimney. As expected from the flow profile, as pitch number increases, the temperature distribution becomes more azimuthally symmetric. At 400 KW, highest system temperatures varied from 912 K to 921 K depending on WHIP pitch number.

The lowest system temperatures are located at the heat exchanger outlet at the exit, inner radius features of the off-set fins. The region is a low flow area and results in the lowest system temperatures under adiabatic conditions. Under conditions that consider internal heat conduction, lowest system temperatures would be higher. The simulations run for this analysis used adiabatic conditions as a limiting conservation case. Lowest system temperatures are 843 - 853 K at 400 KW depending on WHIP pitch number. A closer examination of the radial temperature distribution in the chimney region is shown in Fig. 3.26. Temperature variations in the chimney increase with pitch number. For the 1-pitch and 6-pitch configurations, the temperature variation is 0.5 C and 1.7 C, respectively. At low pitch numbers (1, 2-pitch) the temperature is asymmetric and conforms with the WHIP exit flow streamline (highest temps in the exit streamline). As pitch number increases, the temperature distribution becomes azimuthally symmetric as the vorticity of the flow increases. The lower chimney temperatures tends to correspond with regions of higher flow (improved convective heat transfer). The internal chimney regions are dominated by convective eddies that correspond to the higher fluid temperatures.

Temperature distributions in the reactor system also depend on heat exchanger power demand. As expected, system differential temperatures increase with power demand on the system (see Fig. 3.27 and Table 3.5). As power demand increases, differential temperatures across the heat exchanger increase which in turn drives higher system natural circulation flow. For the 1-pitch configuration, the core differential temperature is 9.3 C at 400 KW and increases to 26.0 C at 2000 KW. For the 6-pitch configuration, the differential temperature increases to 12.8 C at 400 KW and 37.3 C at 2000 KW. While system differential temperature also depends on pitch number, this dependence is not as easily discernible. Core differential temperature can have a complicated dependence on pitch number due to the non-linear nature of the flow character at the WHIP inlet. For instance, at 400 KW the 1-pitch configuration has a differential temperature of 9.3 C. This reduced to 7.9 C for the 3-pitch and increases again to 12.8 C for the 6-pitch. The 3-pitch WHIP configuration tends to have minimum differential temperatures compared to other configurations for all power demand levels evaluated (400 KW to 2000 KW).

3.4.4 Parametric Analysis varying FLiNaK Thermo-physical Properties

Thermal-hydraulic system performance varies as a function of thermo-physical fuel salt properties. Little is known about fuel salt thermo-physical properties at the temperatures and molarities encountered in MSR designs (Sohal, 2013; Thoma, 1959). In order to better quantify the varies in system performance (temperature and flow) as a result of varying thermo-physical properties of the fuel salt a parametric evaluation was conducted which examined system performance at 50%, 100%, 200%, and 500% of known fuel salt eutectic values for dynamic viscosity, specific heat capacity, and fluid thermal conductivity. The results are shown in Figs 3.28 and 3.29 and Tables 3.6 and 3.7. The sensitivity of the mass flow rate to changes in these properties reveals that system performance is relatively insensitive to changes in thermal conductivity. A factor of 5 increase in thermal conductivity results in a 2.3% increase in system mass flow rate. System thermal performance is moderately sensitive to changes in dynamic viscosity. A factor of 5 increase in dynamic viscosity results in a 12.7% reduction in steady-state mass flow rate. The system has considerable sensitivity to changes in specific heat capacity. A factor of 5 increase in specific heat capacity results in a 35.5% reduction in system mass flow rate. Additional laboratory measurement of actual fuel salt specific heat capacities at elevated temperatures is prudent to reducing this engineering uncertainty in design.

Core differential temperature varies with these thermo-physical properties as well. Fig 3.29 illustrates the change in core differential temperature due to changes in dynamic viscosity, specific heat capacity, and fluid thermal conductivity. Similar to the system mass flow rate, core differential temperature is least sensitive to changes in thermal conductivity and most sensitive to changes in specific heat capacity. For a factor of 5 change in dynamic viscosity, specific heat capacity, and thermal conductivity, core differential temperature changes by -2.2%, +14.1%, and -48.4%, respectively.

3.4.5 Conclusions

A basic examination of the the thermal hydraulic performance of the WHIP MsNB was conducted using STAR-CCM+ on a high performance computing system (Sawtooth) at the Idaho National Laboratory. A few interesting results are summarized in this section. Axial and radial flow, temperature, and pressure distributions were numerically calculated based on an adaptive mesh of the closed loop reactor system mesh. The flow results show turbulence increases as the pitch number of the WHIP increases. The turbulence is evident in the core inlet and interior chimney volume and heat exchanger interfaces. Resistance to flow also increases in the system as pitch number increases due to narrowing of core flow channel and increases in channel surface area. For a given power demand, the system achieves its highest flow rate with a 3-pitch WHIP configuration. At higher pitches the flow rate decreases due to these fractional effects. Localized regions worthy of additional design scrutiny include the WHIP inlet (first stage eddy formation), chimney vortex flow, and low and interior regions of the heat exchanger. Turbulence in the chimney may have neutronic benefits as mixing may keep delayed neutron precursors in the vicinity of the core to promote efficiency in the active core's neutron economy.

Fuel salt mass flux exiting the core is dependent on pitch number, system power demand, radial distance from the WHIP center line. Changes in fuel salt axial velocity with increased distance from the WHIP radial center may cause differential losses of precursors. For instance, if the fuel salt axial velocity is lower at the outer radius compared to the inner radius of the core then advection of delayed neutron precursors from the core will be higher in the core interior. This may cause a change in the core's radial neutron flux distribution and have reactivity effects up to 20 pcm at 400 KW during power operations.

Higher pitch numbers tend to produce more azimuthally symmetric flow. This is particularly the case at higher power demand levels (higher flow rates). This is caused by higher flow rates near the outer core and chimney walls which in turn results in higher flow vorticity.

A parametric study was completed to better understand sensitivity of thermal hydraulic performance to changes in fuel salt dynamic viscosity, specific heat capacity, and thermal conductivity. System thermal performance has the strongest sensitivity to changes in specific heat capacity and the least sensitivity to fluid. thermal conductivity. Additional investigation into fuel salt specific heat capacity properties at various compositional molarities and fuel loading values may be warranted.

3.5 OH-NC WHIP Experimental Data Evaluation

Graduate students at the University of Idaho conducted experimentation on a closed-loop system consisting of a ohmic heating source and tradition counter flow tubular heat sink (Geddes, 2021). Geddes experimental showed the viability of using ohmic heating to achieve natural circulation in a closed-loop system. The experiment was referred to as the ”clear vessel” ohmic heated natural circulation (OH-NC) apparatus. The OH-NC was instrumented with Omega Type J thermocouples with an accuracy of $\pm 2^{o}$C to measure heat source and sink temperatures to confirm thermodynamic parameters consistent with natural circulation flow. A data acquisition system with LabVIEW HMI (human machine interface) was used to monitor system performance (Bitter, 2006). The operating fluid was 25 w% salt water flowing in an inner and outer concentric tube configuration. The inner tube housed the heat source electrodes and a mock-up of a 5-pitch WHIP axially located in between the two heater electrodes. The hydraulic diameter of the inner flow tube was 0.0627 meters.

The overall system is show in Fig. 3.30. The ”clear vessel” OH-NC loop stands 246.4 cm in height. The heat sink extends about 150 cm into the vertical fluid column between the inner and outer flow tubes. The two heater electrodes are axially 20.32 cm and 77.5 cm from the bottom of the loop. The electrodes volumetrically heat the fluid and mimic fission heating the working fluid. The WHIP is located between the two electrodes in a volume that mimics the active core region of the loop. Thermocouples of interest are located at the heat sink inlet and outlet and cooling fluid (tap water) heat sink secondary side outlet. While several experimental runs were completed to calibrate thermocouples, test heat source variac performance, and establish baseline system performance and operational proficiency; this section will focus on the experiment run completed on 23 April 2020. This run included the WHIP and tested the loop natural circulation performance under three different heat source power load settings. Aside from the obvious fact that the ”clear vessel” OH-NC loop has no fission heating source and neutronics play no role in system behaviors, study of the OH-NC loop in it’s current design also differs from the MsNB in one important attribute - the power behavior of the heat source is NOT load following. This experiment studies the loop performance based on experimenter input power settling of ampere (at a certain voltage) in the heat source variac. Future enhancements to the OH-NC loop system might include a temperature and flow rate input control system that adjusts the power output based on neutronically relevant thermodynamic parameters. In this way, the neutronics can be simulated and converted to power control signals based on heat sink demand.

The 5-pitch WHIP is shown in Fig. 3.31. The clear tubing allows for qualitative observation of flow character during experimentation. The WHIP was 3D printed using commercially available off-the-shelve printers using nylon printing materials. These materials provided to requisite rigidity under the elevated salt water temperatures present in the OH-NC loop during operation. Neutrally buoyant particles were added to the working fluid during experimentation to provide an experimenter aid and visual contrast to help qualitatively evaluate for flow behavior.

One informative experimental run was completed using the 5-pitch WHIP 3D printed mimic device to validate natural circulation flow performance in the ”clear vessel” OH-NC loop. After calibrating loop thermocouples against water state change temperature conditions, the loop was heated using a heat source setting at 512.1 watts for 45 minutes to achieve thermal equilibrium. The power setting was then increased to 1542.5 watts for a duration of approximately 30 minutes to achieve thermal equilibrium. The power was then increased to a final setting of 5200 watts for a duration of approximately 45 minutes. This run profile is shown in Fig. 3.32. Geddes solved for losses to ambient using thermocouple data provided by difference between heat source outlet, heat sink inlet, heat sink outlet, and heat source inlet. Losses to ambient were determined to account for approximately 31% of the total heat source input. Using thermodynamic heat balance across the heat sink at various power levels, the cooling water inlet was calculated to be 14.1^o C and cooling water mass flow rate was calculated to be 0.44 kg/s. Fig. 3.32 illustrates the power test and the associated temperature by the system. As expected, system temperature increases as power input increases. Cooling water flow to the heat sink remains constant during these three power transients. Using the heat sink differential temperature at quasisteady-state conditions, the system mass flow rate can be calculated using the 25wt% specific heat capacities (Geddes, 2021).

For the low power setting ($\approx$ 510 watts), the cooling water and salt water differential temperatures of about 0.3 and 1.2 degrees Celsius, respectively, which results in a system mass flow rate of 0.038 kg/s and linear flow velocity of 1.2 cm/s in the inner flow tube above the WHIP. The total loop transport time at steady state for this power setting is approximately 734 seconds. Similarly, at a power setting of approximately 1540 watts, the heat sink differential temperature rises to 5.7 degrees Celsius which implies a mass flow rate of 0.046 kg/s and velocity of 1.5 cm/s. As expected, the cooling water outlet increases by 1.2 degrees Celsius at steady state. The loop transport time at this setting was 588 seconds. Finally, at 5200 watts the cooling water and heat sink differential temperatures rise to 3.0 and 10.8 degrees Celsius, respectively. This corresponds to a mass flow rate of 0.081 kg/s and flow velocity of 2.6 cm/s. The loop transort time at this higher power setting was 339 seconds. While the ”clear vessel” OH-NC loop had no flow detectors to measure actual flow velocity, qualitative observations and numerical buoyant flow solvers used by Geddes supports with calculated system flow rates at the various power settings observed (Geddes, 2021).

Discussion with Geddes regarding her specific qualitative observations of the WHIPs performance include noting that the flow was more stable (tended to result in less variability in the radial dimension during power setting changes. She also observed that more buoyant particles were entrained in the bulk fluid when using the WHIP as compared to only using a straight vertical empty tube. Geddes noted a radial velocity profile to the internal WHIP flow. The flow near the outer tube boundaries tended to be lower than the flow in the center flow stream and near the inner WHIP root. This radial flow profile was also seen in STAR-CCM+ simulations using certain WHIP pitch configurations. In summary, Geddes' qualitative observations provide initial validation of STAR-CCM+ simulations of a closed-loop natural circulation system using the WHIP to direct hydraulic flow. Geddes did not observe a tangential component to the flow velocity at the WHIP exit. The flow was predominantly axial. While tangential components are predicted by the STAR-CCM+ simulations for the full sized MsNB. The absence of a tangential flow component in the ”clear vessel” OH-NC apparatus was likely due to the small axial length of the ”core” WHIP and the reduced thermal driving head present in the smaller system.

3.6 Summary

The WHIP is a novel device invented by the author and is protected under a provisional patent filed with the USA Patent Office on 28 September 2021 under application number 63/261,776, BEA docket number BA-1254. The WHIP offers the potential for new and innovative MSR design applications. This chapter reports the initial examination on the WHIP motivation and design features, a neutronics analysis, a CFD thermal-hydraulic analysis, and evaluation of existing experimental data to validate the performance and viability of this device in the application of a micro-natural circulation molten salt nuclear battery design concept. The experimental data shows that the WHIP promotes natural circulation flow and improves flow stability in a closed-loop volumetrically heated natural circulation laboratory engineering scale system. CFD and neutronics analysis confirmations satisfactory physics performance and provides evidence of unique performance enhancements associated with MSR unique core physics parameters such as k_{eff}, flow reactivity and β_{eff} without detriment to hydraulic flow performance. These results affirm the potential advantage of achieving fuel and reactor system performance requirements with minimal fuel loading and size/weight constraints. The WHIP decouples the hydraulic size and neutronic size of the core which enhances control and stability features of the design. The WHIP introduces new and different design features to help mitigate flow patterns that might promote partial stagnation points, insoluble depositions, and chemical concentrations within the reactor system. In conjunction with other work in this dissertation, namely chapters four and five, the unique advantages the WHIP may provide a designer are presented.

3.7 Tables

Table 3.1: MsNB Material Masses (* U-235 included in fuel salt)

Material	**Mass (kg)**
304 SS	12,766
Hastelloy® N cladding	195
BeO reflector	6,937
B_4C absorber	678
Fuel Salt (FLiNaK-UF_4)	4,140
Graphite	554
U-235*	1,910
Total	25,270

Table 3.2: MsNB Fuel Salt Composition - FLiNaK at eutectic (46.5-11.5-42 mol%) 18 mol% UF_4

Constituent Flouride Salt	**Weight %**
NaF	4.38
LiF	10.96
KF	22.11
UF_4	62.55

Table 3.3: 6-pitch MsNB WHIP core physics Serpent parameters. Effective delayed neutron fraction is calculated by adjoint weighted time constant perturbation techniques. Prompt neutron lifetime calculated using Nauchi method.

Parameter	Value	Uncertainty
η	1.943	0.00136
f	0.4905	0.00189
p	0.3777	0.00130
ϵ	2.857	0.00254
L_f	0.9771	0.00025
L_{th}	0.9971	0.000063
K_{eff}^{∞}	1.0287	0.00151
K_{eff}	1.0023	0.00151
$\bar{\nu}$	2.453	1.6x10^{-5}
$\bar{E}_f$	202.41 MeV	1.8x10^{-6}
E_n	0.166 eV	0.00224
λ	0.485 s^{-1}	0.03111
$\beta_{eff,static}$	0.00724	0.01859
$\beta_{eff,flow}$	0.00690	0.01859
l^*	9.577x10^{-5} s	0.00564
1/v	6.2128x10^{-7} s/cm	0.00163
t_{thermal}	3.102x10^{-4} s	0.00189
N_{th}	9.46	0.00094
$P_{density}$	2.1x10^{-4} kW/g	-
ϕ_n	6.0338x10^{18} cm/s	0.0158
Fissile Mass	1905.12 kg	-
Avg.Temperature	914.65 K	-

Table 3.4: System Mass Flow Rate (kg/s) at various powers and pitches

Power (KW)	1-pitch	2-pitch	3-pitch	4-pitch	5-pitch	6-pitch
400	22.15	21.27	44.12	40.74	38.14	16.10
800	28.54	27.24	55.71	51.53	47.87	20.14
1200	32.97	31.39	63.96	58.90	54.82	23.01
1600	36.48	34.69	69.86	64.95	60.72	25.32
2000	39.43	37.45	75.82	69.82	65.98	27.29

Table 3.5: Core Differential Temperature (in Celsius) at various powers and pitches

Power (KW)	1-pitch	2-pitch	3-pitch	4-pitch	5-pitch	6-pitch
400	9.29	9.65	7.85	8.63	9.30	12.81
800	14.39	14.96	12.05	13.09	14.44	20.23
1200	18.66	19.44	15.49	16.85	18.69	26.293
1600	22.46	23.41	18.87	20.47	22.42	31.97
2000	25.96	27.10	21.68	23.14	26.01	37.26

Table 3.6: Parametric Evaluation: Core Differential Temperature

Property Fraction	Viscosity	Sp. cp	Th. Cond.
0.5	0.965	1.192	1.003
1.0	1	1	1
2.0	1.038	0.781	0.992
5.0	1.141	0.516	0.978

Table 3.7: Parametric Evaluation: System Mass Flow Rate

Property Fraction	Viscosity	Sp. cp	Th. Cond.
0.5	1.038	1.120	0.997
1.0	1	1	1
2.0	0.962	0.853	1.009
5.0	0.873	0.645	1.023

3.8 Figures

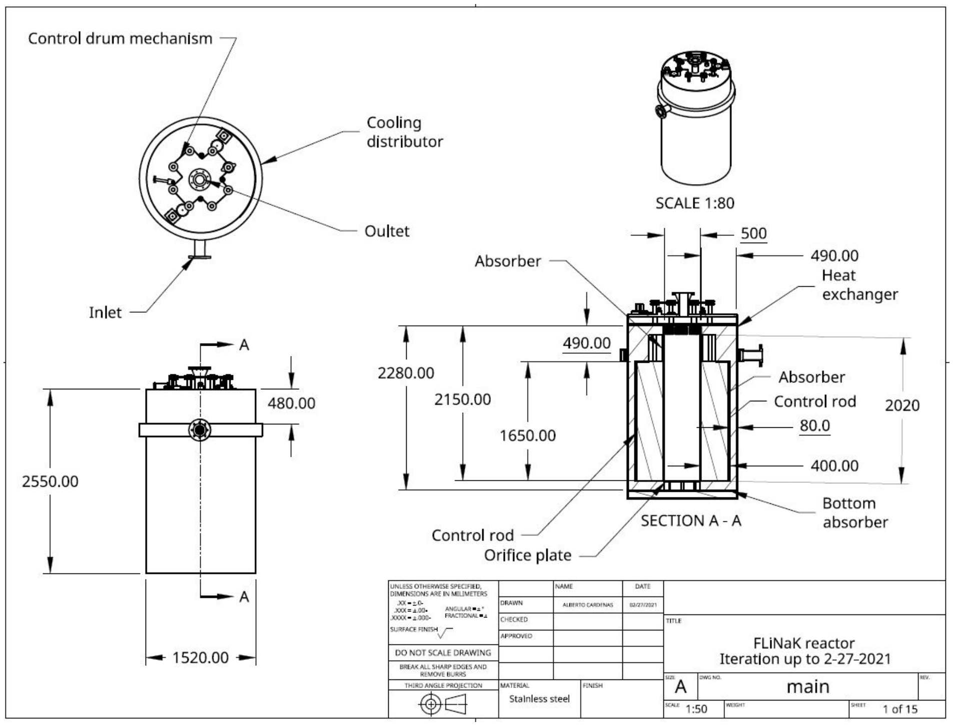


Figure 3.1: Nominal MsNB Drawing. Courtesy of Alberto Cardenas Melgar 2021.

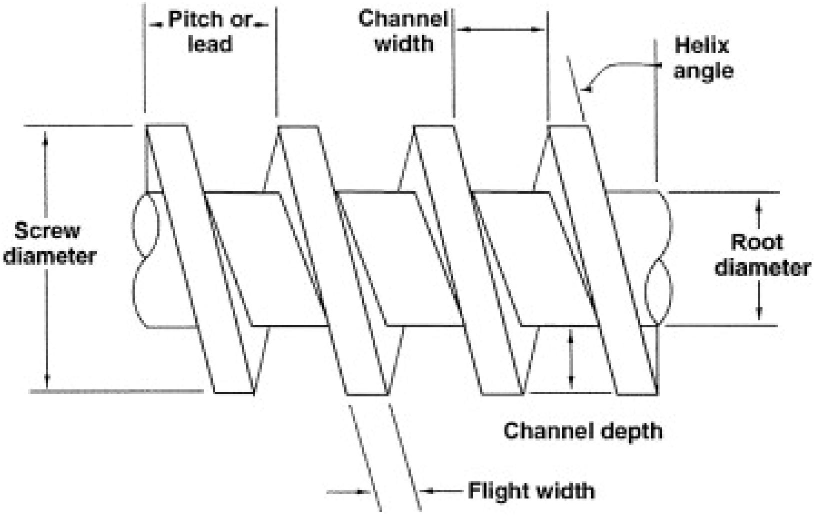
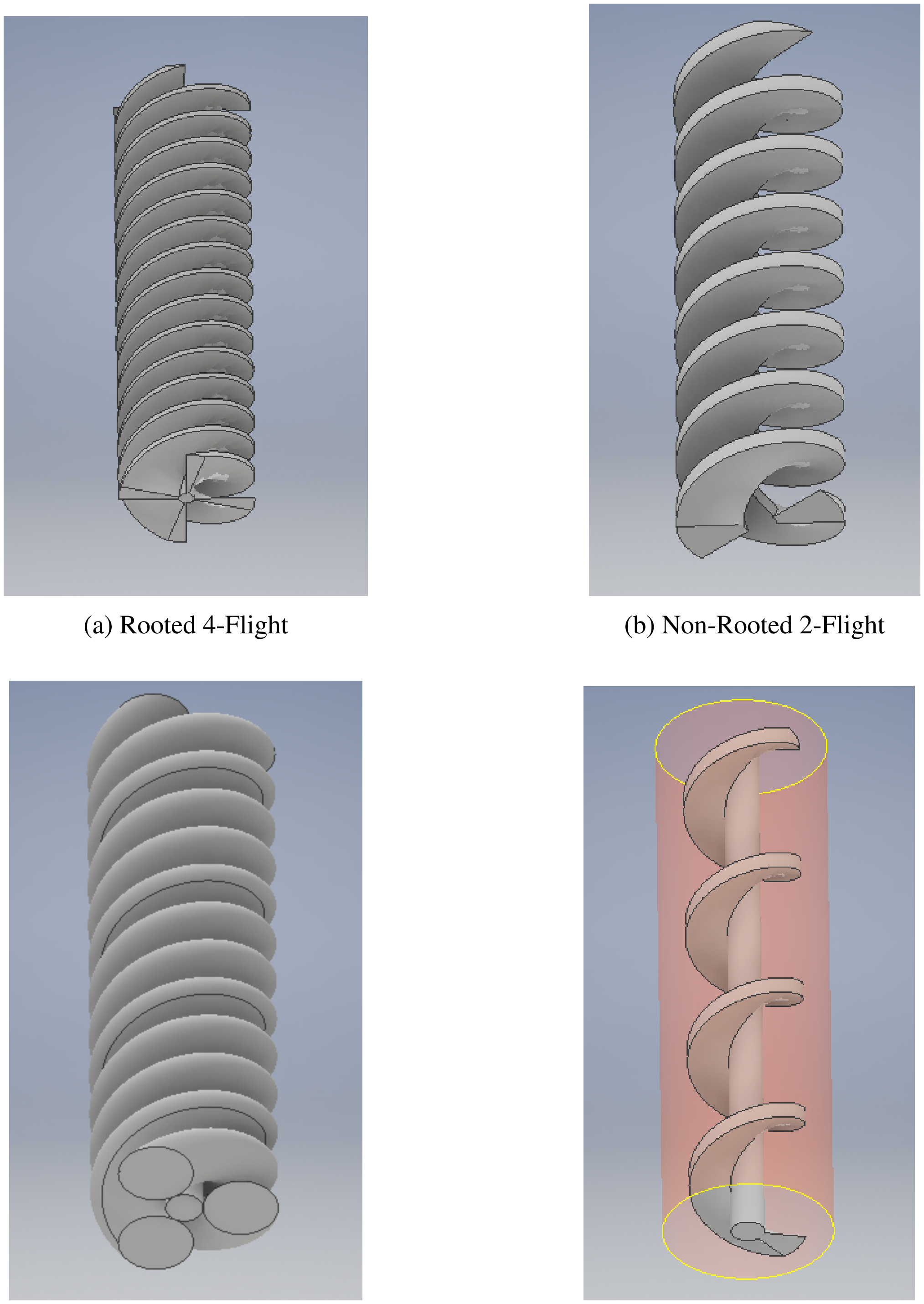


Figure 3.2: Nominal WHIP Diagrammatic Nomenclature. The example shown has a rectangular channel cross section. In practice, channel cross sections can take on variable forms. For instance, in Fig. 3.6 the trapezoidal form was used to shape the radial velocity profile of the fluid to affect reactivity and control drum neutronic worth.



(a) Rooted 4-Flight

(b) Non-Rooted 2-Flight

(c) Rooted 3-Flight Cylinder

(d) Rooted 1-Flight Annulus

Figure 3.3: Variety of possible WHIP configurations. Rooted versus non-rooted, number of flights, annular versus full, cylinder versus rectangular.

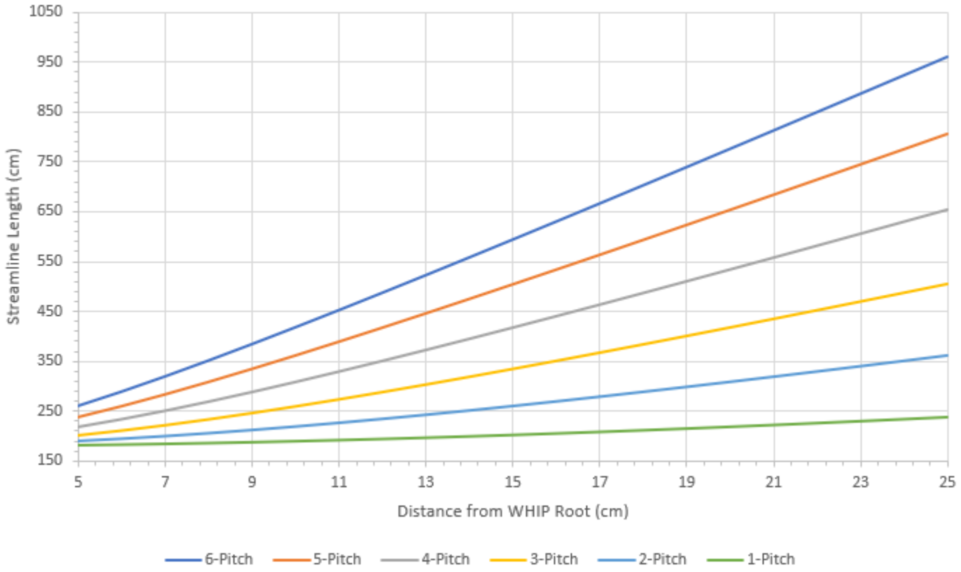


Figure 3.4: Hydraulic cord length in MsNB WHIP: 1-6 pitches. Vertical core height is 179 cm. Root radius is 5 cm. WHIP outer flight radius is 25 cm. In a 6-Pitch WHIP the hydraulic flow length can be 5.3 times the vertical core height. This additional length affects the proportional delayed neutron precursor residence time in core.

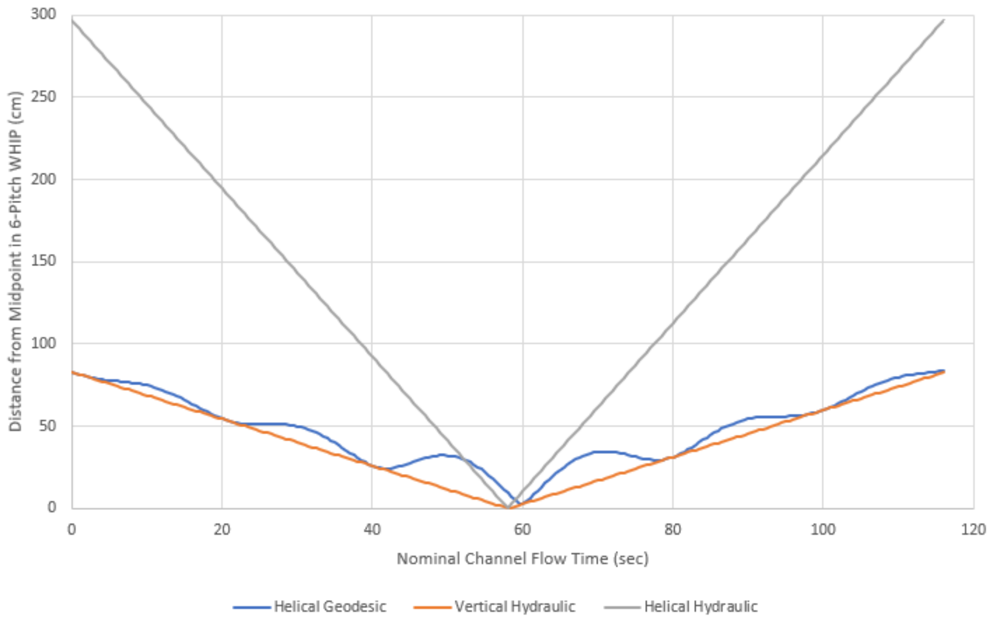
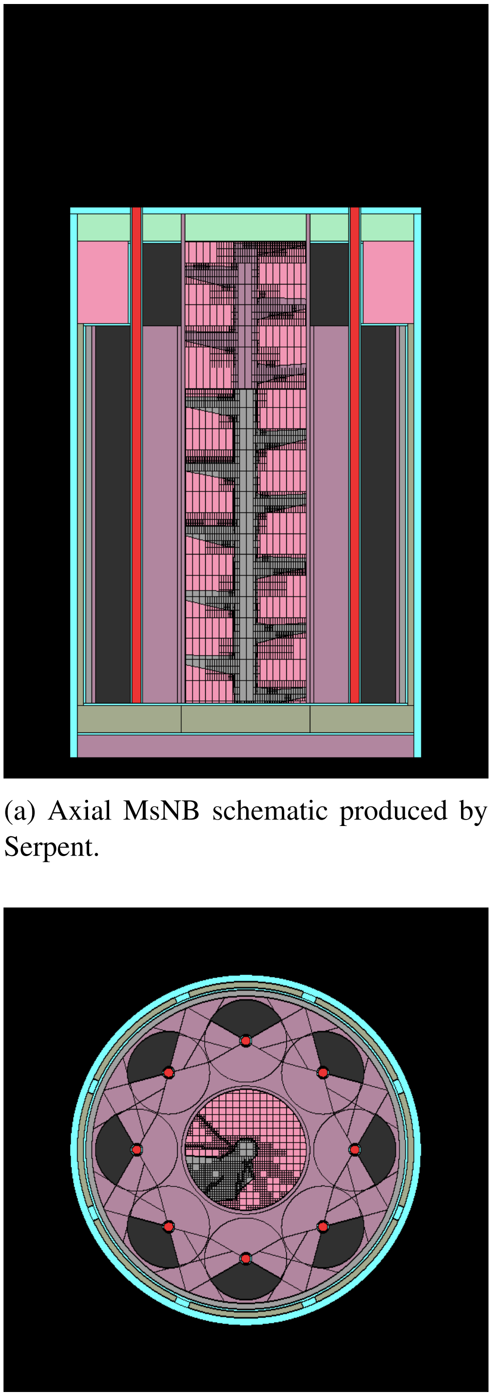


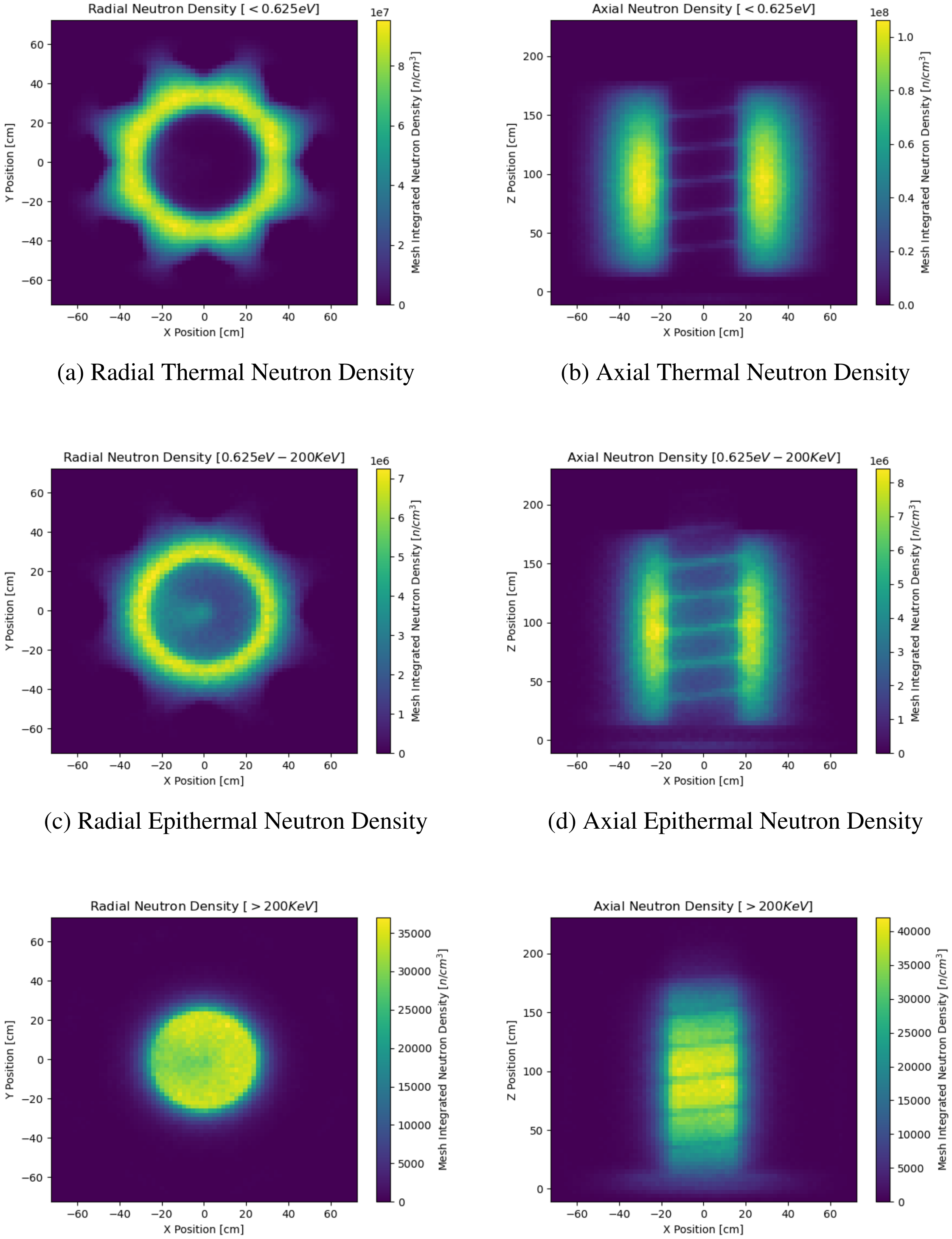
Figure 3.5: Straight line (geodesic) and hydraulic flow cord distances in a 6-Pitch WHIP from a chosen helical midpoint, r=15 cm and z=89.5 cm. The helical geodesic periodically cycles to a minimum equal to the vertical straight line distance as fluid passes directly above or below the midpoint. The helical hydraulic length represents the fluids full helical flow distance from the midpoint. Adjacent pitch fluid streams experience recurring near passes which neutronically enable physics coupling to occur. For instance, the geodesic distance at 30 cm from the midpoint occurs at approximately 38, 48, 68, and 76 sec channel flow time. Fluid in the WHIP experiences a neutronic coupling similar to the smaller traditional vertical flow design.



(a) Axial MsNB schematic produced by Serpent.

(b) Radial MsNB schematic produced by Serpent.

Figure 3.6: Serpent MsNB schematics. Colors differentiate material cards. 6-pitch WHIP configuration with trapezoidal channel cross sections to enhance radial flow gradient to assist in radial flux dampening to affect control drum neutron worths.



(a) Radial Thermal Neutron Density

(b) Axial Thermal Neutron Density

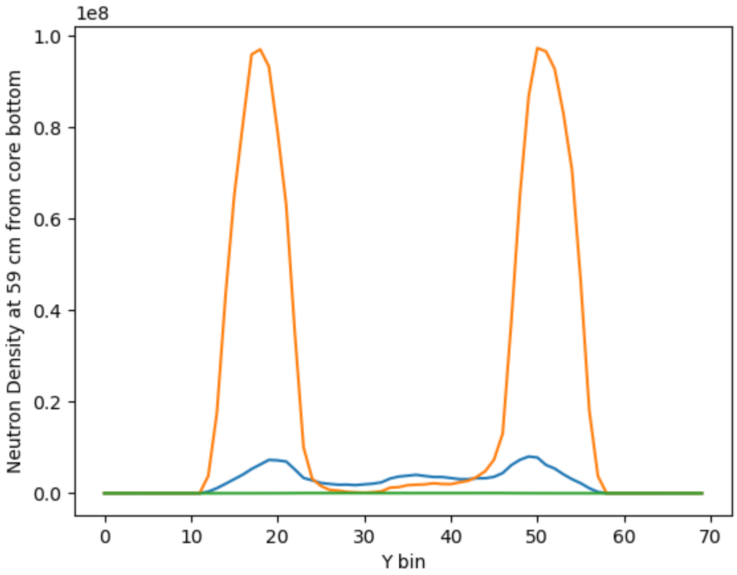
(c) Radial Epithermal Neutron Density

(d) Axial Epithermal Neutron Density

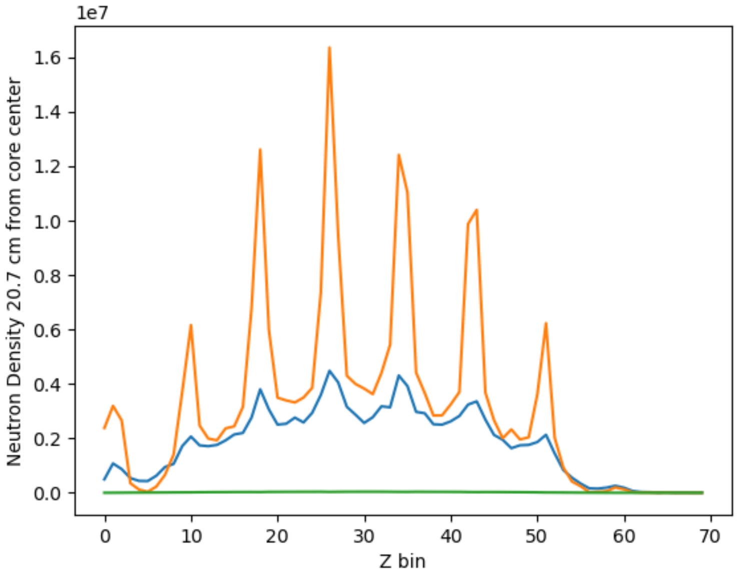
(e) Radial Fast Neutron Density

(f) Axial Fast Neutron Density

Figure 3.7: Serpent Neutron Density Distributions for the critical (K_{eff}=1.00250), 6-pitch, trapezoidal WHIP configuration operating at 400 KW. Thermal energies are <0.625 eV, epithermal are between 0.625 eV and 200 KeV, and fast spectrum neutrons are >200 KeV.



(a) Radial Neutron Density Profile at about 59 cm up from bottom of active core region (1/3 of core axial height). Slightly asymmetry exists about bin 35 (core center line) due to presence of the WHIP flight cross section. Control rod reactivity is greatly increased due to this distribution.



(b) Axial Neutron Density Profile at 21.7 cm (roughly flow channel midpoint) from core centerline. Z bin 0 is located at the inlet plenum and Z bin 70 is located at the top of the outlet plenum.

Figure 3.8: Radial and axial neutron density profiles for a 6-pitch WHIP configuration with trapezoidal channel cross sections to enhance radial flow gradient to assist in radial flux dampening to affect control drum neutron worths. Yellow is the thermal spectrum, blue is the epithermal spectrum, and green is the fast spectrum.

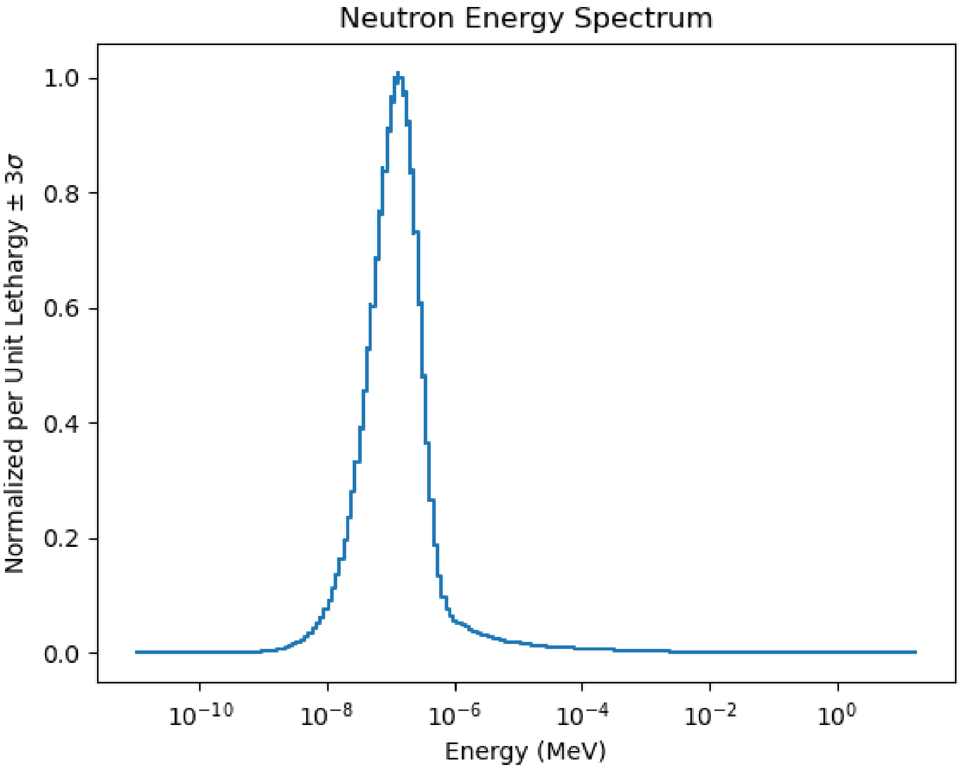


Figure 3.9: MsNB neutron energy spectrum per unit lethargy. Mode of energy distribution is 0.167 eV (slightly under moderated).

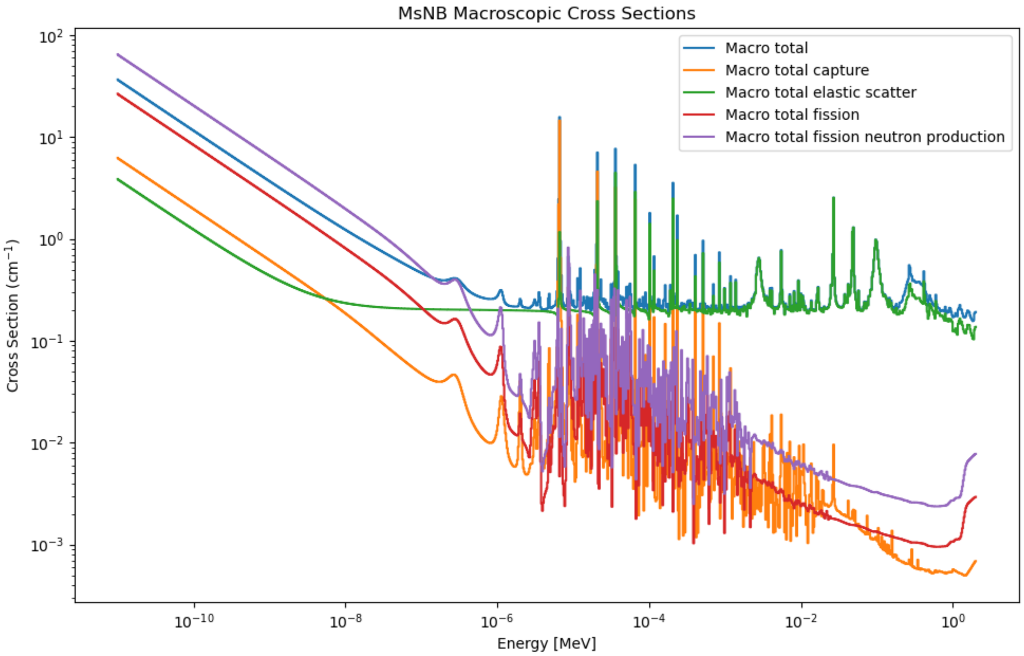
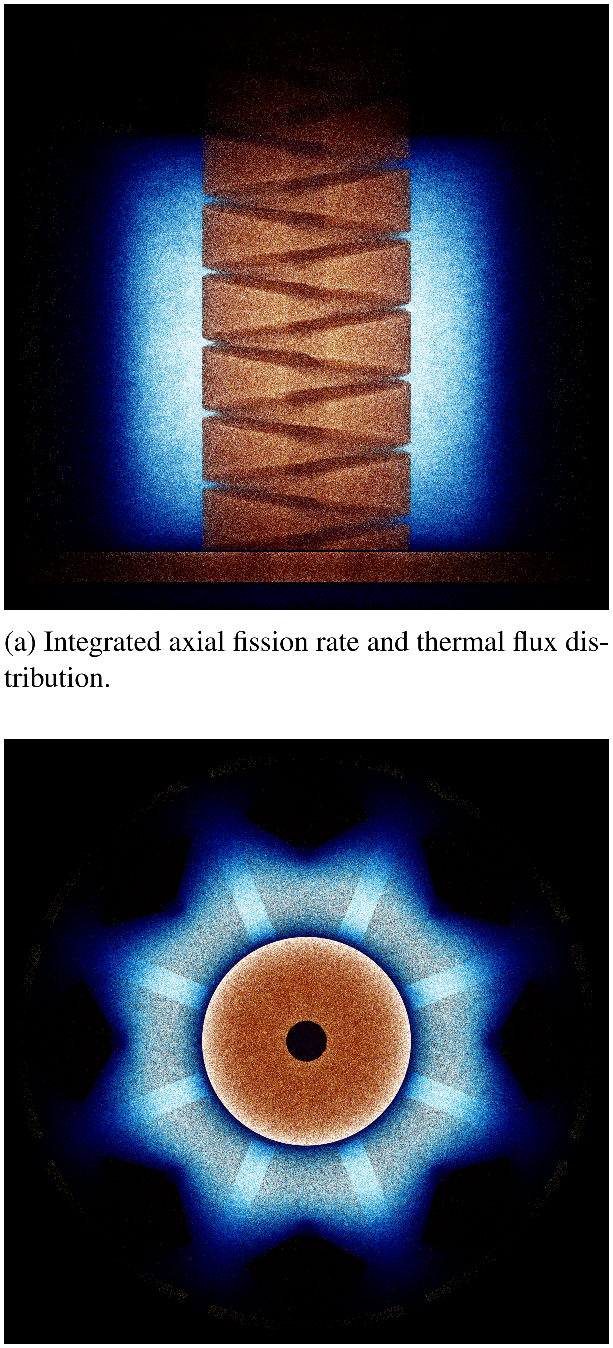


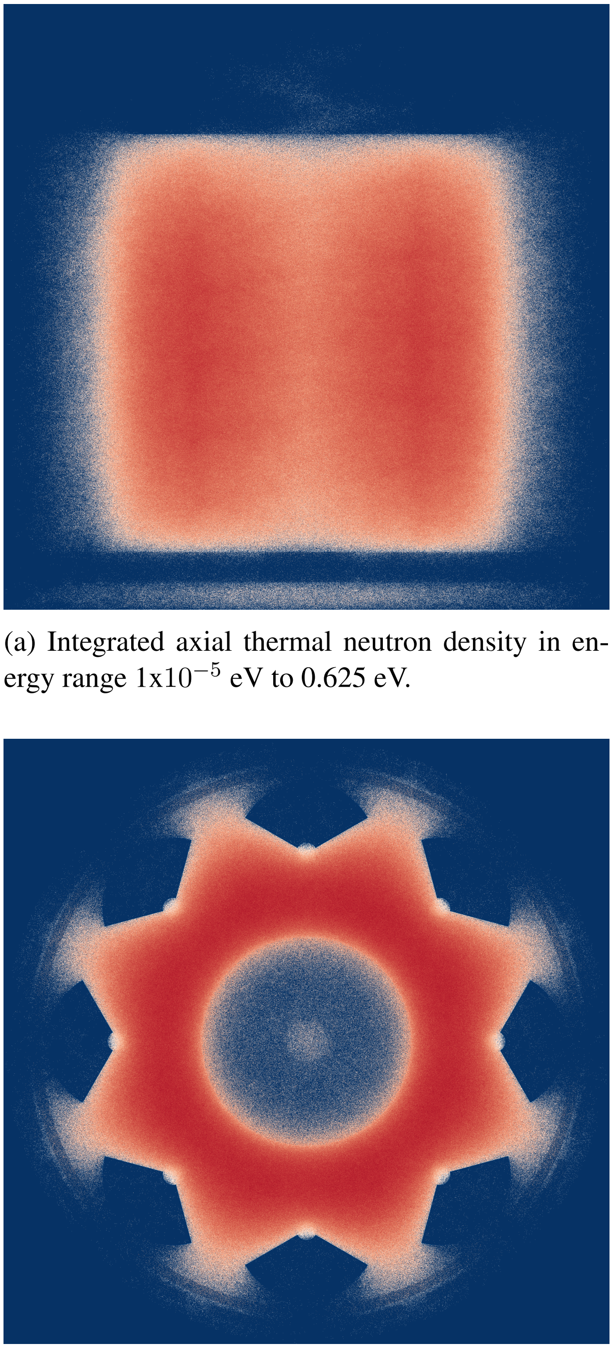
Figure 3.10: Macroscopic Cross Sections of the MsNB with WHIP.



(a) Integrated axial fission rate and thermal flux distribution.

(b) Integrated radial fission rate and thermal flux distribution.

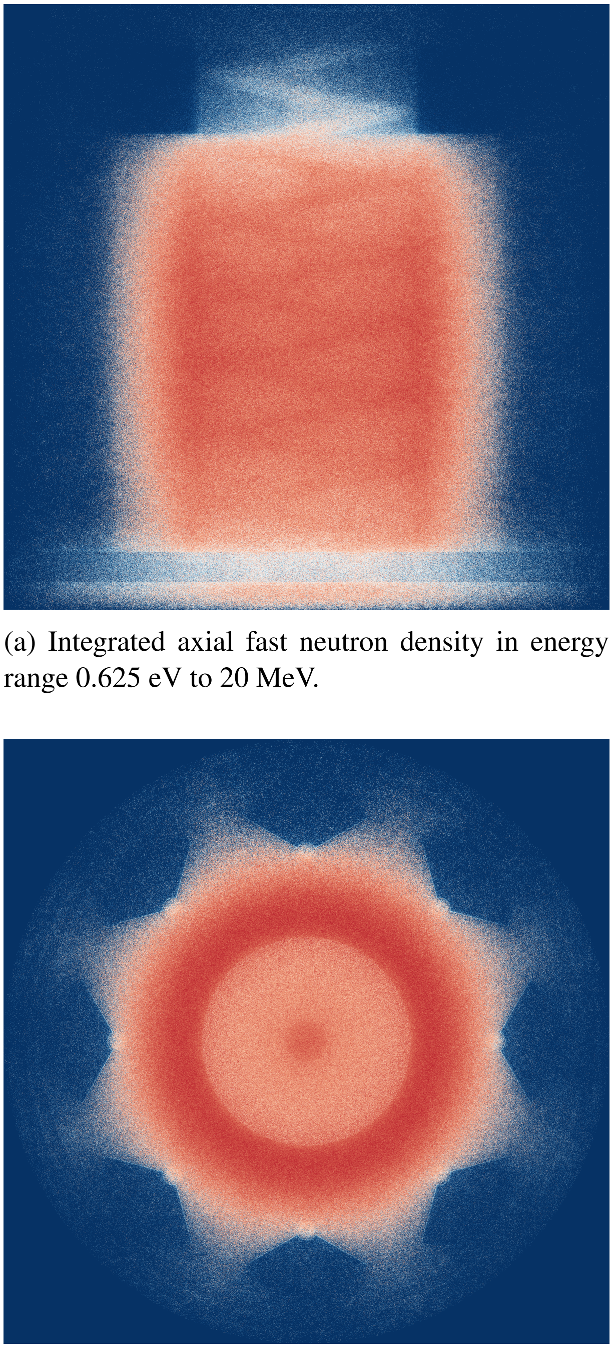
Figure 3.11: Fission rate and thermal flux distribution is in hot and cold color schemes, respectively. This a 6-pitch WHIP configuration with trapezoidal channel cross sections to enhance radial flow gradient to assist in radial flux dampening to affect control drum neutron worths.



(a) Integrated axial thermal neutron density in energy range 1x10^{-5} eV to 0.625 eV.

(b) Integrated radial thermal neutron density in energy range 1x10^{-5} eV to 0.625 eV.

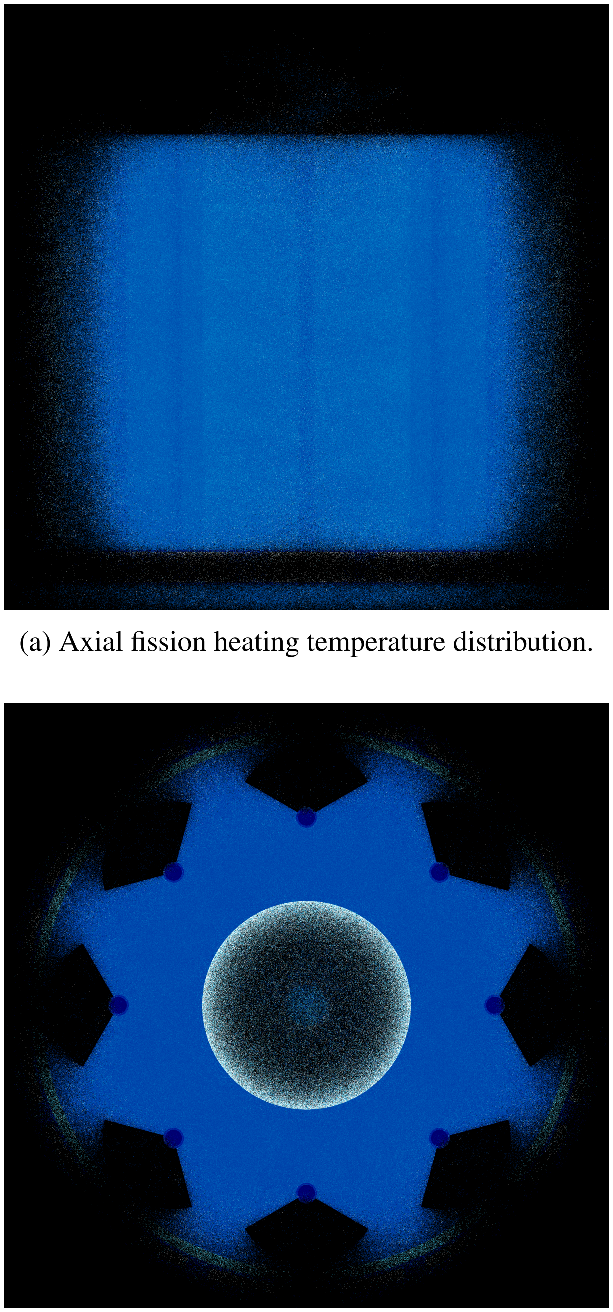
Figure 3.12: Thermal neutron density in energy range 1x10^{-5} eV to 0.625 eV. This a 6-pitch WHIP configuration with trapezoidal channel cross sections to enhance radial flow gradient to assist in radial flux dampening to affect control drum neutron worths.



(a) Integrated axial fast neutron density in energy range 0.625 eV to 20 MeV.

(b) Integrated radial fast neutron density in energy range 0.625 eV to 20 MeV.

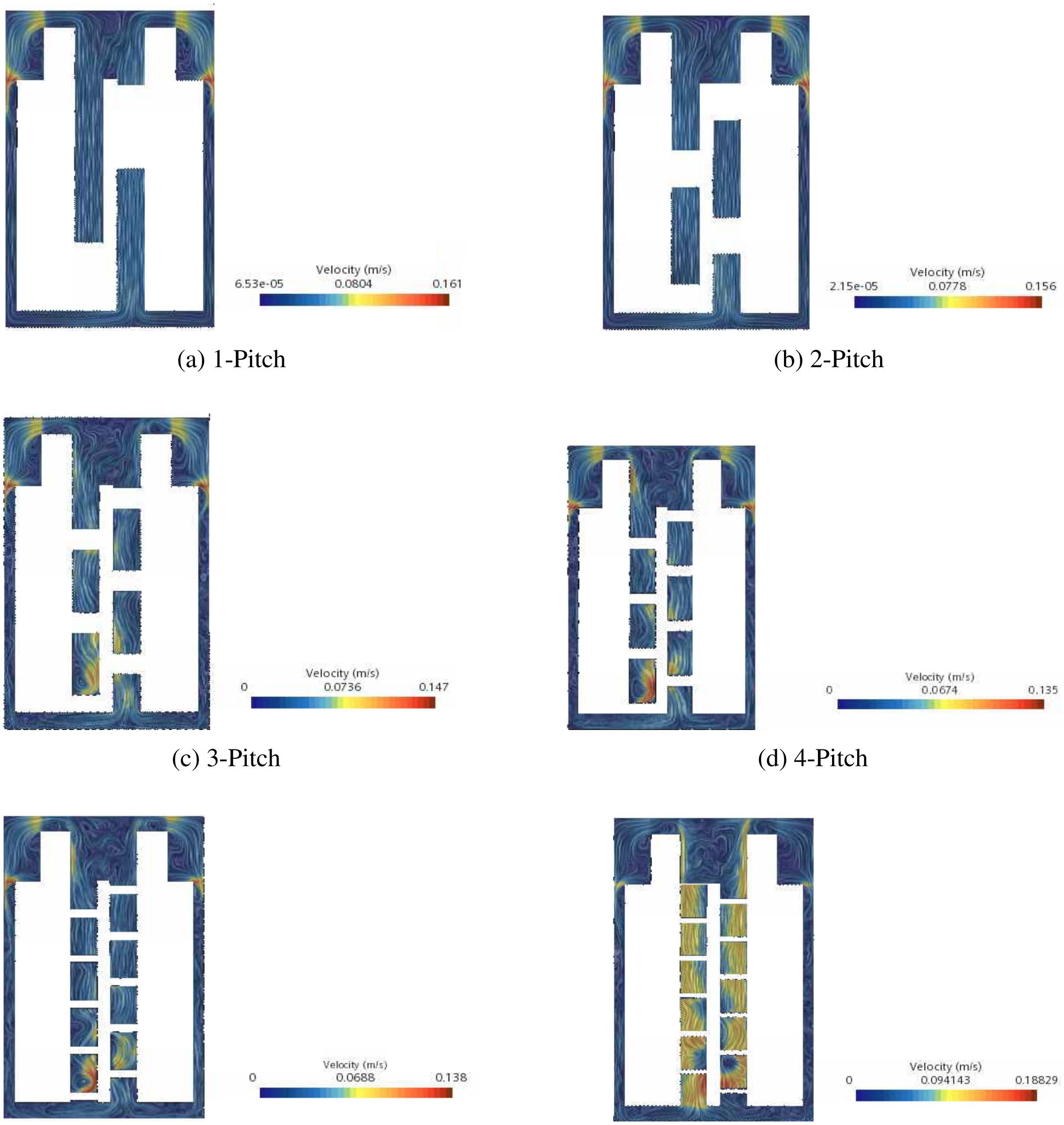
Figure 3.13: Fast neutron density in energy range 0.625 eV to 20 MeV. This a 6-pitch WHIP configuration with trapezoidal channel cross sections to enhance radial flow gradient to assist in radial flux dampening to affect control drum neutron worths.



(a) Axial fission heating temperature distribution.

(b) Radial fission heating temperature distribution in a 6-pitch WHIP configuration with trapezoidal channel cross sections to enhance radial flow gradient to assist in radial flux dampening to affect control drum neutron worths.

Figure 3.14: MsNB fission heating temperature distribution.



(a) 1-Pitch

(b) 2-Pitch

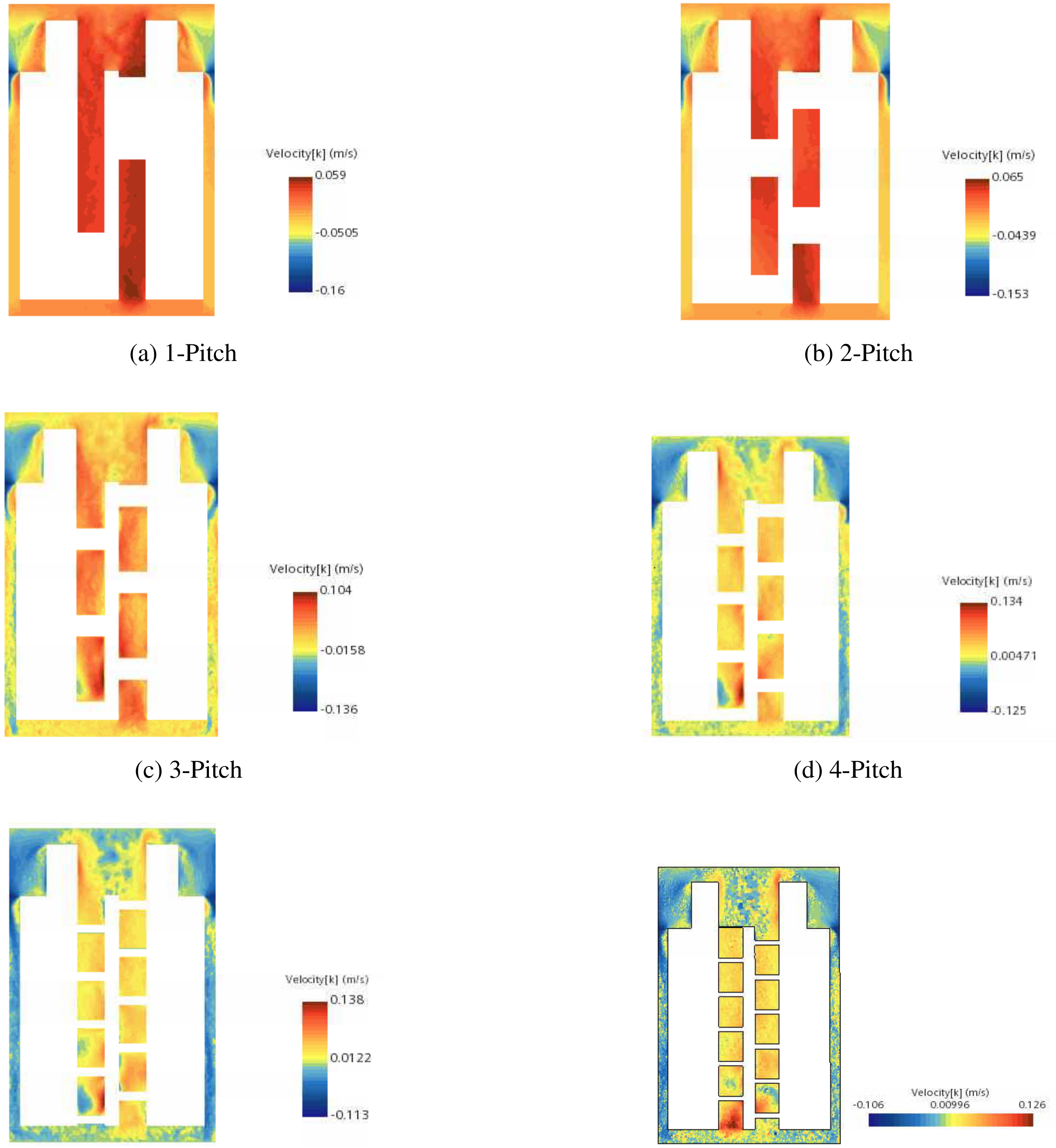
(c) 3-Pitch

(d) 4-Pitch

(e) 5-Pitch

(f) 6-Pitch

Figure 3.15: Reactor cross section representation of fuel salt velocity streamlines. Eddies and turbulent flow are evident in the chimney and the first lower WHIP pitch for WHIPs with pitch numbers above 2. A flow eddy occurs in the first lower pitch which constraints the fuel salt flow to the inner WHIP core surface and results in higher flow velocities low in the core. This artifact has no impact on fluid temperature due to the low integrated volumetric heat generation low in the core. The reactor system generally has higher flow rates at lower pitch number due to the lower surface flow area and lower frictional facts present in the low pitch number configurations.



(a) 1-Pitch

(b) 2-Pitch

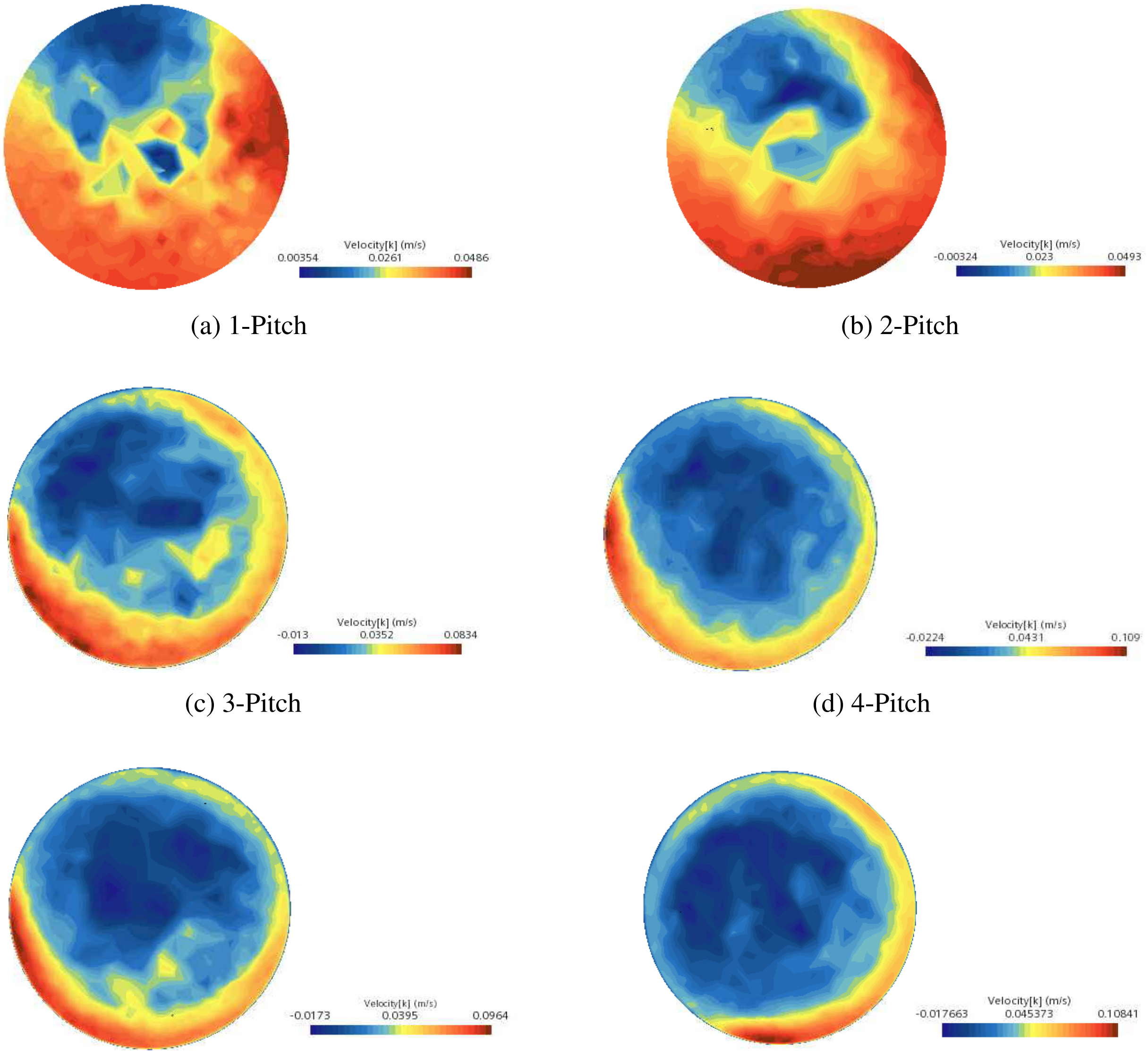
(c) 3-Pitch

(d) 4-Pitch

(e) 5-Pitch

(f) 6-Pitch

Figure 3.16: Reactor cross section representation of fuel salt vertical flow component. Fuel salt flow velocity in the z-direction increases with pitch number. Upward flow in the core and chimney and downward flow in the downcomer is illustrated. While interesting flow features exist at the inlet and outlet of the heat exchanger, this examination will focus on the flow field in and near the core and chimney regions. Increasing pitch number produces increased rotational flow in the core and chimney. The buoyant flow generally requires one full pitch to establish rotational flow. The high vertical fluid velocities indicated in the lower, inner surface of the first pitch (for 3-pitch through 6-pitch configurations) is due to the fluid reorienting from the inlet plenum to establish rotational flow up the WHIP. This artifact has no impact on fluid temperatures due to their location low in the core because any significant volumetric heating has occurred.



(a) 1-Pitch

(b) 2-Pitch

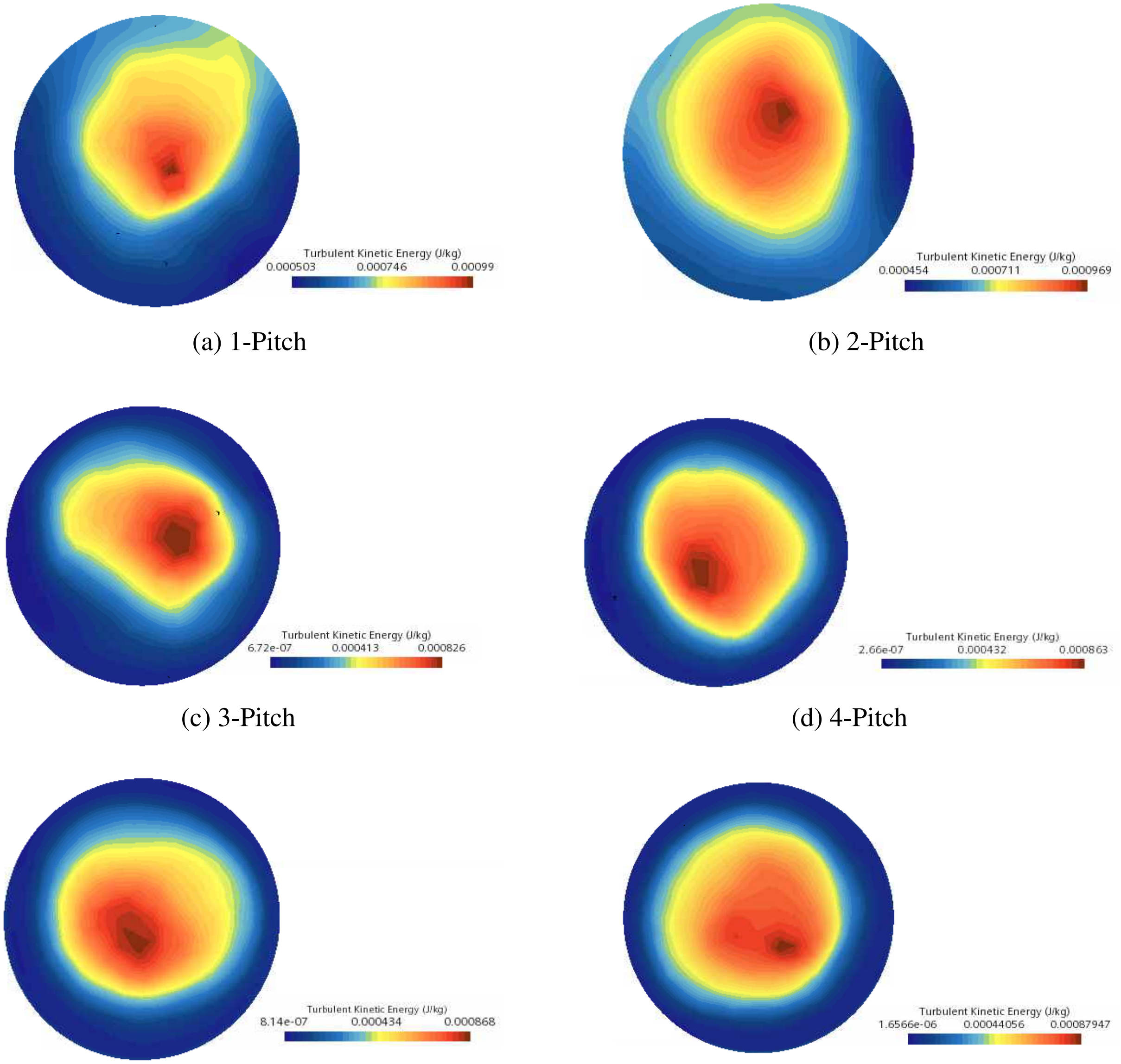
(c) 3-Pitch

(d) 4-Pitch

(e) 5-Pitch

(f) 6-Pitch

Figure 3.17: Vertical component of fuel salt velocity in the chimney. This vertical component of fluid velocity contributes to the radial dependence of flow reactivity in the core. Fuel salt exits the core at a higher velocity near the walls of the chimney. Flow vorticity is greater for higher pitch numbers and contributes to higher exit flow rates. Negative velocities are present in the chimney center due to vortex mixing and eddy formation.



(a) 1-Pitch

(b) 2-Pitch

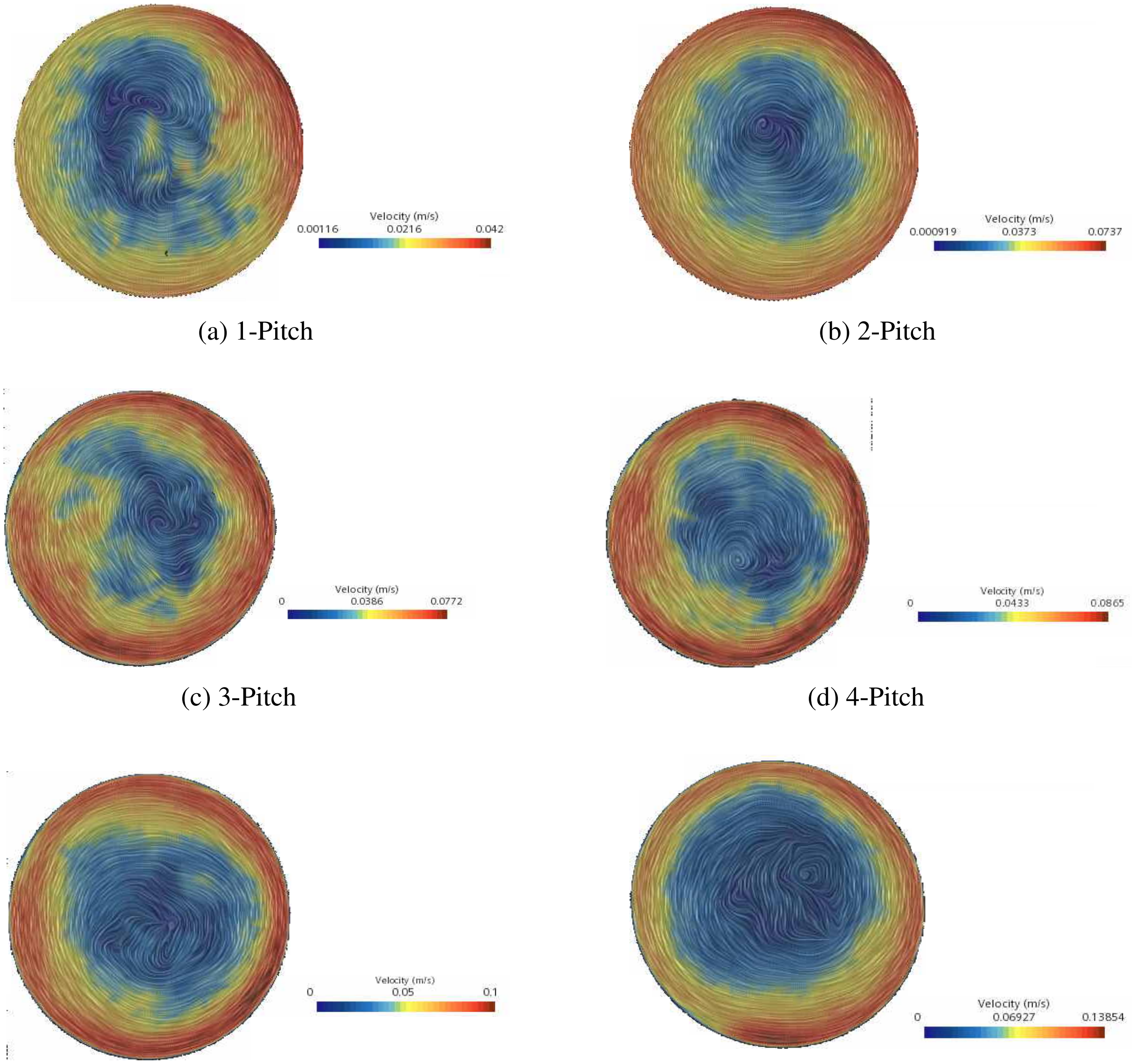
(c) 3-Pitch

(d) 4-Pitch

(e) 5-Pitch

(f) 6-Pitch

Figure 3.18: Turbulent kinetic energy in the chimney. For all pitch configurations examined the maximum turbulence is at the center of the chimney. This location corresponds to the center of vortex flow. A potential benefit of greater turbulence near the center of the chimney riser and less turbulence near the chimney walls is advection of delayed neutron precursors tend to be trapped in the turbulent flow in the center and are more likely to contribute their delayed neutrons to the to the core compared to a purely axial vertical flow with low turbulence.



(a) 1-Pitch

(b) 2-Pitch

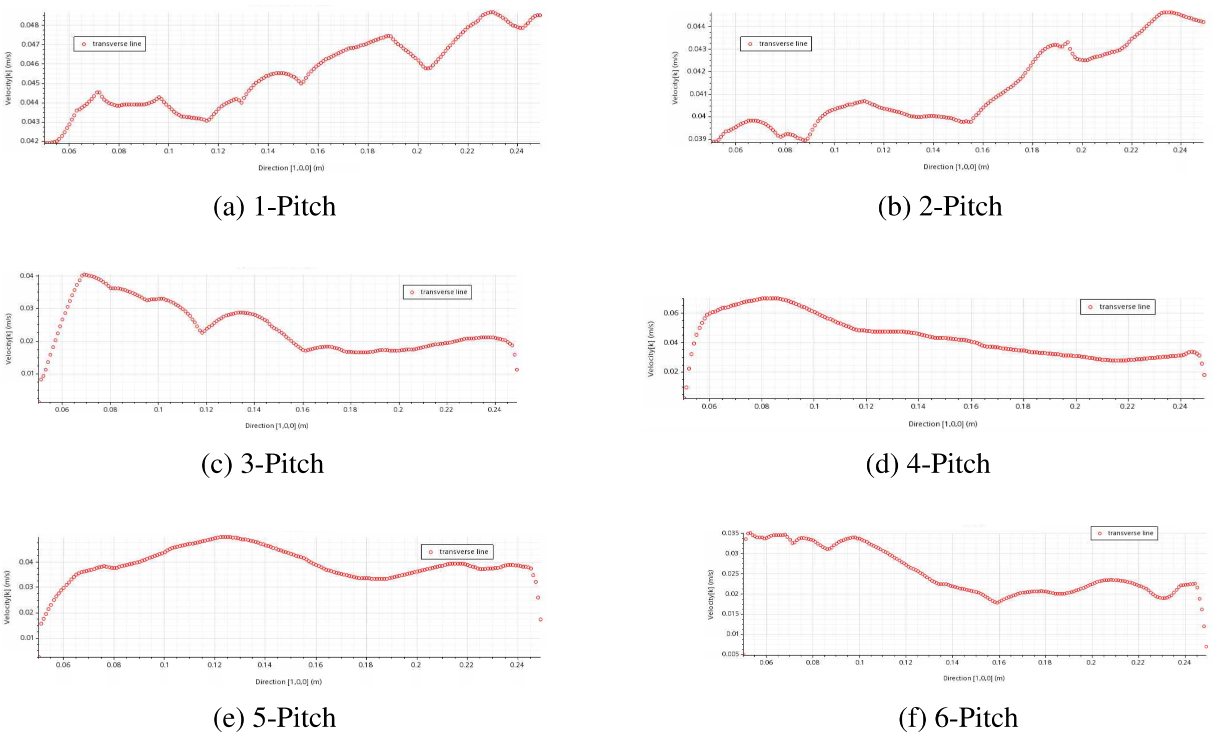
(c) 3-Pitch

(d) 4-Pitch

(e) 5-Pitch

(f) 6-Pitch

Figure 3.19: Vector representation of the magnitude of fuel salt velocity on the chimney. Maximum velocity near the chimney walls increases with pitch number. The magnitude of the fuel salt velocity decreases to near zero in the chimney center due to the rotational flow.



(a) 1-Pitch

(b) 2-Pitch

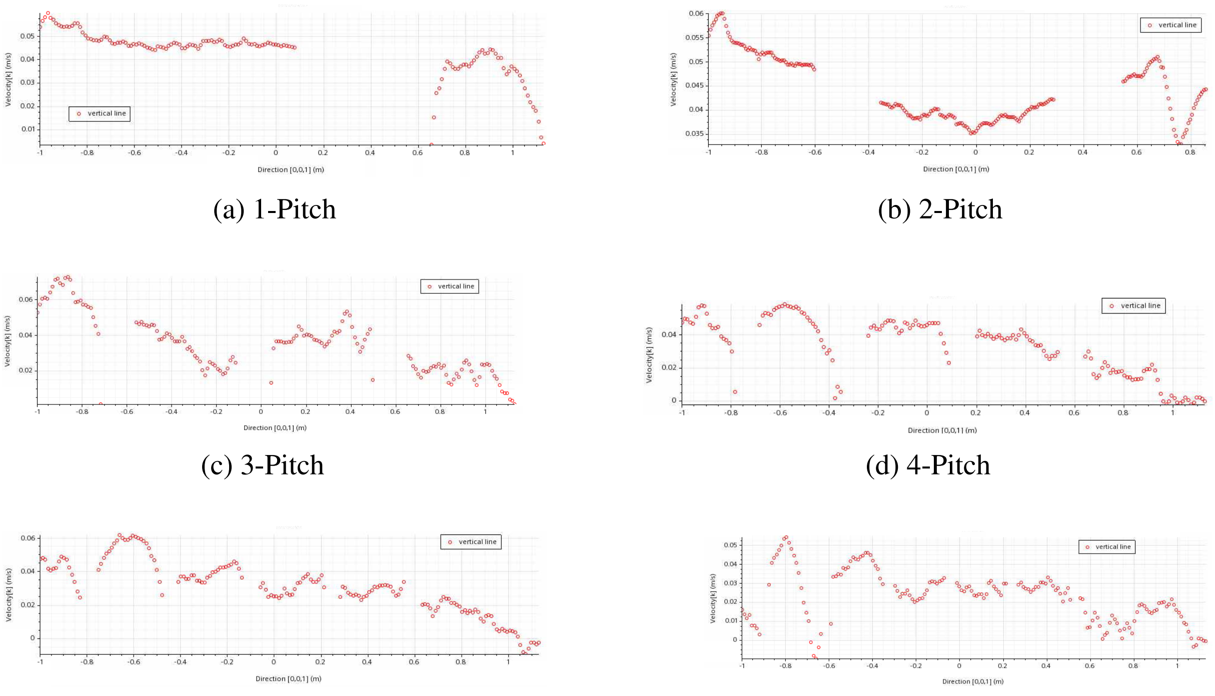
(c) 3-Pitch

(d) 4-Pitch

(e) 5-Pitch

(f) 6-Pitch

Figure 3.20: Vertical z-component of fuel salt velocity in the WHIP channel as a function of distance from the WHIP core at 400 KW. At low pitch numbers the vertical component of fuel salt velocity increase with distance from the WHIP core. Velocity in the channel is not a smooth function of channel position.



(a) 1-Pitch

(b) 2-Pitch

(c) 3-Pitch

(d) 4-Pitch

(e) 5-Pitch

(f) 6-Pitch

Figure 3.21: Vertical z-component of fuel salt velocity as a function of axial distance up the reactor. Direction -1 corresponds to the core inlet and +1 corresponds to the top of the outlet plenum. Discontinuities in the data represent WHIP flights. At high pitch numbers the inlet flow might experience eddies which reverse flow in the channel. This can be seen in (f).

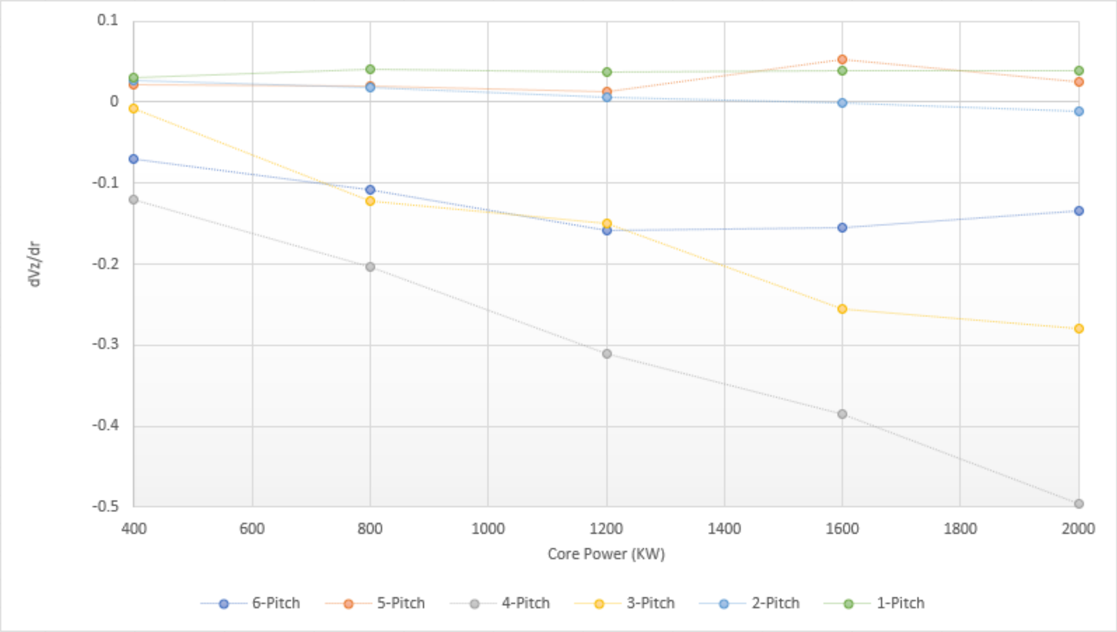


Figure 3.22: Parametric evaluation of $\frac{dV_z}{dr}$ for various core power values and WHIP pitch numbers. WHIPs with low (1-2) or high (>5) pitch numbers have a low sensitivity to radial v_z changes. A WHIP with 4 pitches has the highest degree of radial velocity sensitivity. v_z radial sensitivities tend to be negative. That is, as the distance from the WHIP core increases the axial component of the fuel salt velocity decreases. In a large neutronically decoupled MSR, this may induce positive reactivity effect near the outer radius of the core.

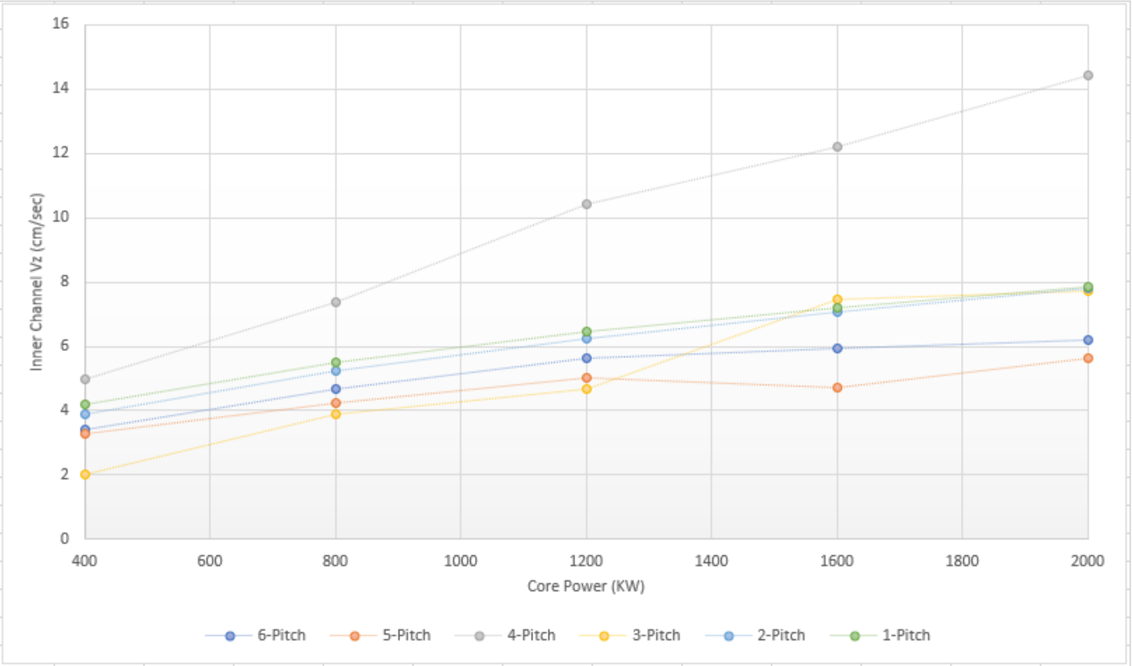
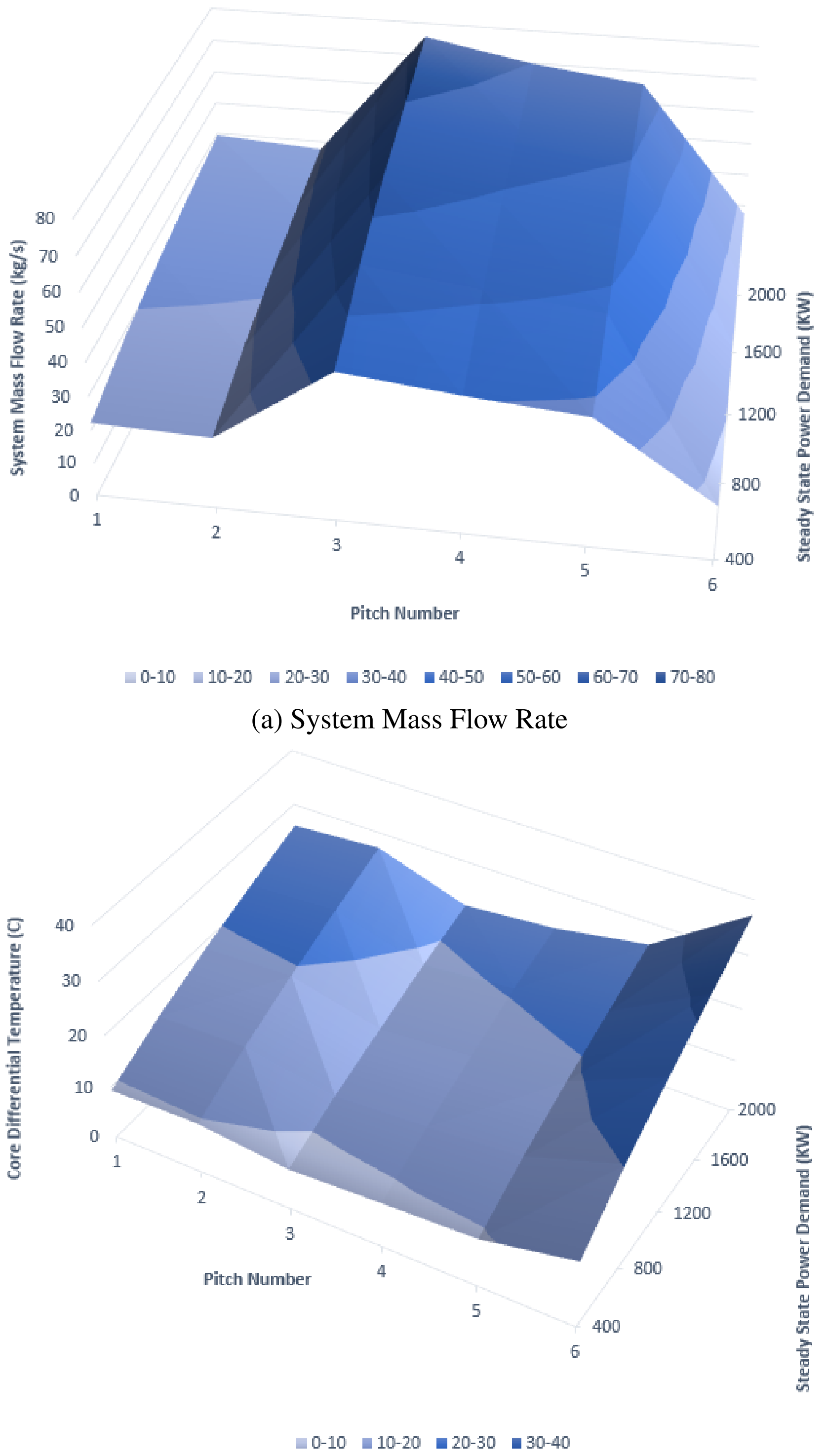


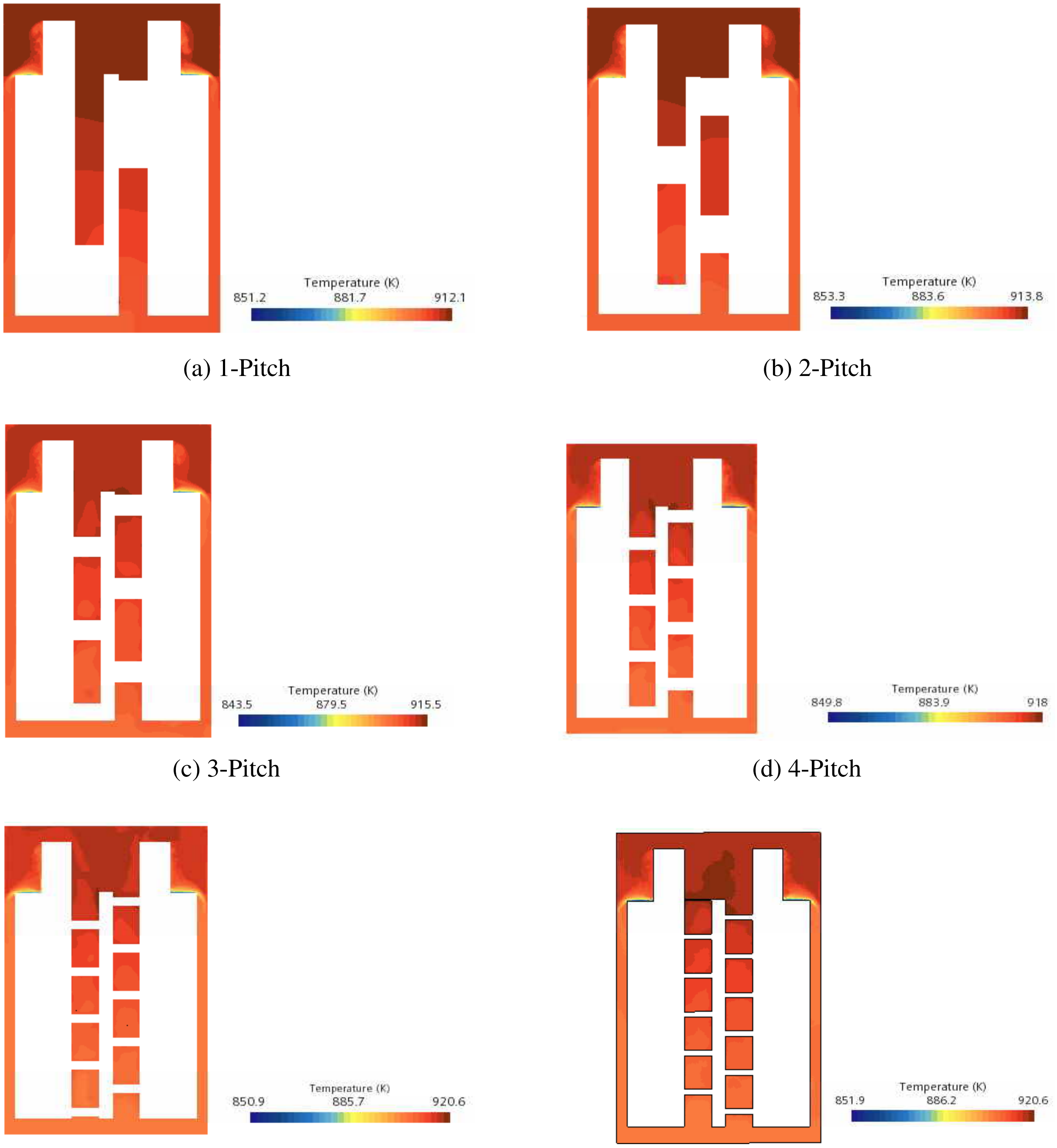
Figure 3.23: v_z near the WHIP core surface as a function of pitch number and core power. Naturally the higher buoyant forces present at higher core powers drive higher system flow rates. Channel turbulence and inlet eddies provide for a more interesting result. The 4-pitch WHIP appears to have unique flow characteristics that differ from higher or lower pitch numbers.



(a) System Mass Flow Rate

(b) Core Differential Temperature

Figure 3.24: Core Differential Temperature and System Mass Flow Rate for Power demand and pitch combinations. Differential temperatures are a minimum for the 3-pitch WHIP configuration and corresponding mass flow rates are highest for this configuration.



(a) 1-Pitch

(b) 2-Pitch

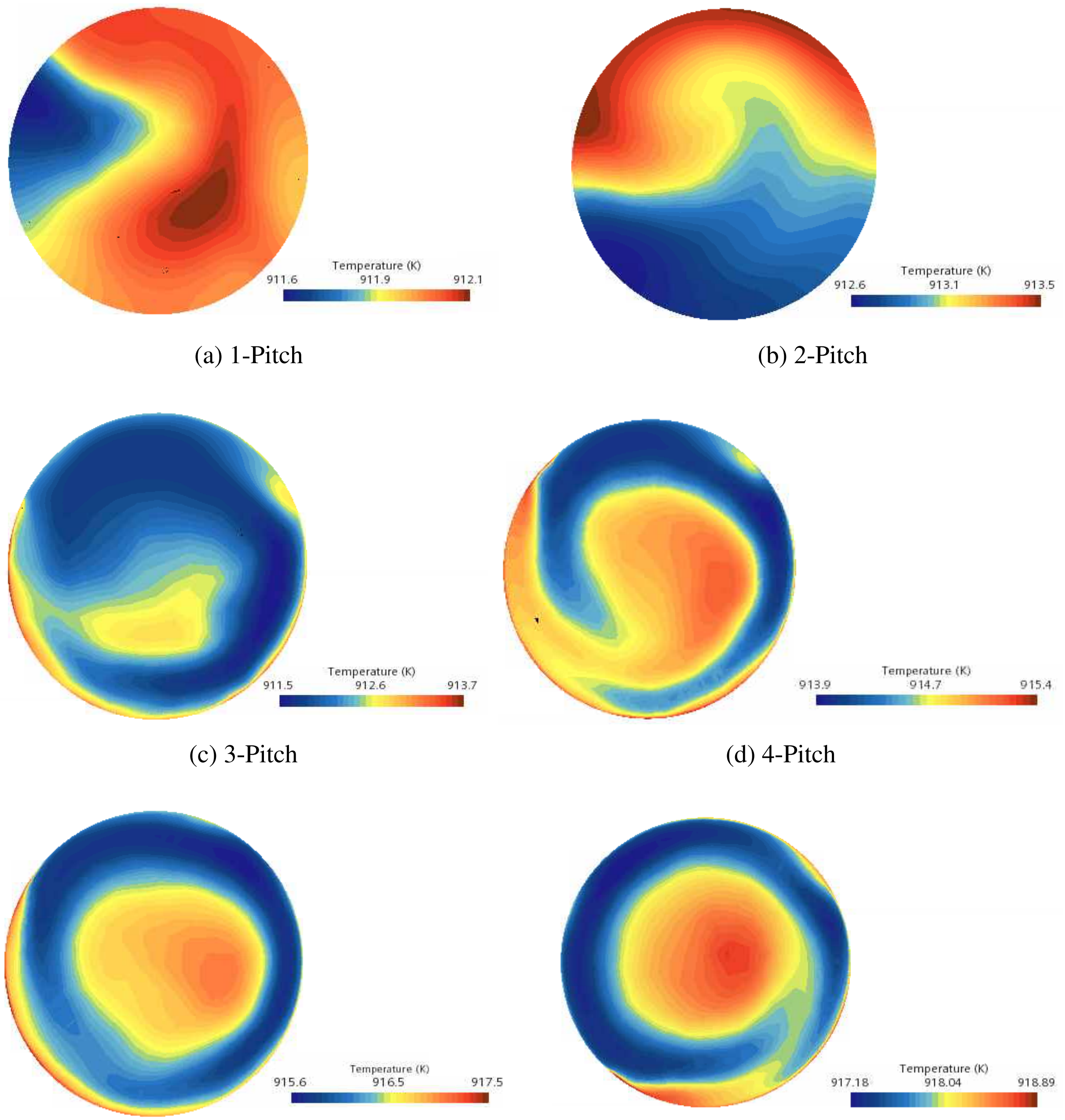
(c) 3-Pitch

(d) 4-Pitch

(e) 5-Pitch

(f) 6-Pitch

Figure 3.25: Reactor cross section representation of fuel salt temperature. At steady-state, temperatures are highest in the chimney and upper core volumes. Highest temperatures increase with pitch number due to the decrease in system flow due to increases in frictional factors. Lowest system temperatures are present in the lower features of the heat exchanger due to the low flow conditions that exist in those regions. A heat exchanger design modification would likely be warranted to mitigate the effects of this lower temperature volume. The low temperature calculated is a conservative result due to the adiabatic wall conditions that were assumed in the CFD model.



(a) 1-Pitch

(b) 2-Pitch

(c) 3-Pitch

(d) 4-Pitch

(e) 5-Pitch

(f) 6-Pitch

Figure 3.26: Fuel salt temperature distribution in the chimney at a core power of 400 KW. As pitch number increases, the radial temperature profile is increasingly symmetric about the chimney center. Maximum temperatures are present at the chimney wall near the WHIP outlet and near the chimney centerline. The differential temperature in the chimney is 1-2 degrees Celsius depending on the pitch number.

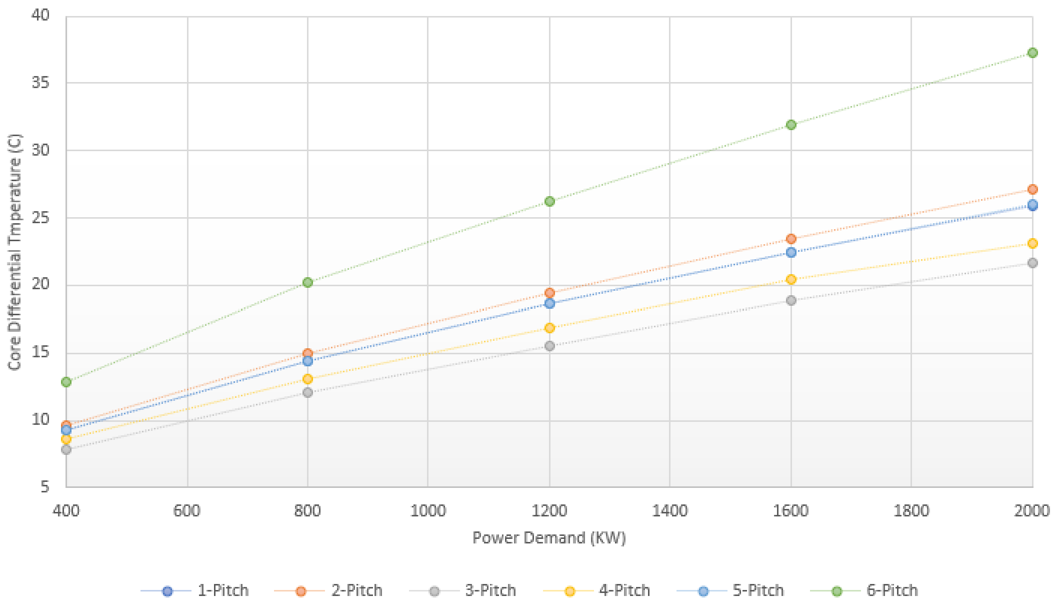


Figure 3.27: Core Differential Temperature as a function of Heat Demand and WHIP configuration.

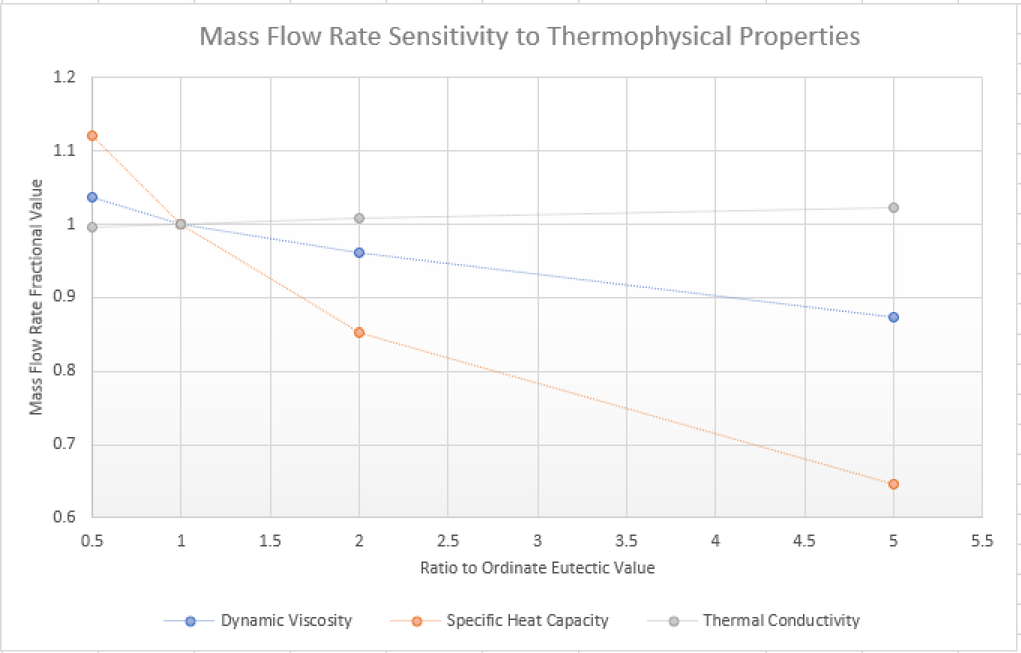


Figure 3.28: Parametric evaluation of mass flow rate due to 0.5, 1.0, 2.0, and 5.0 factor changes in the thermo-physical property ordinate value at the eutectic.

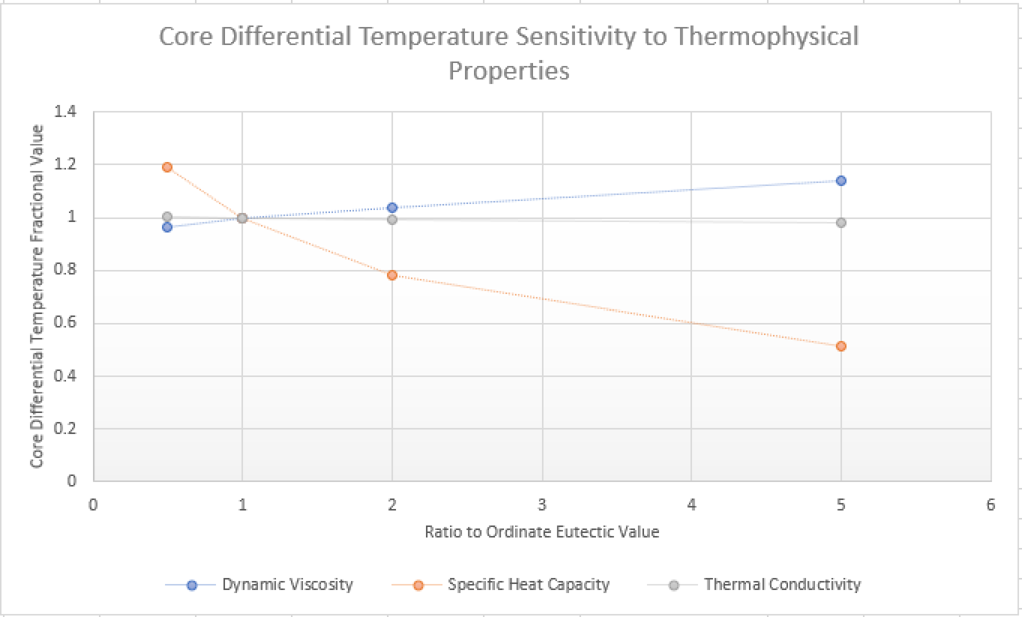


Figure 3.29: Parametric evaluation of core differential temperature due to 0.5, 1.0, 2.0, and 5.0 factor changes in the thermo-physical property ordinate value at the eutectic.

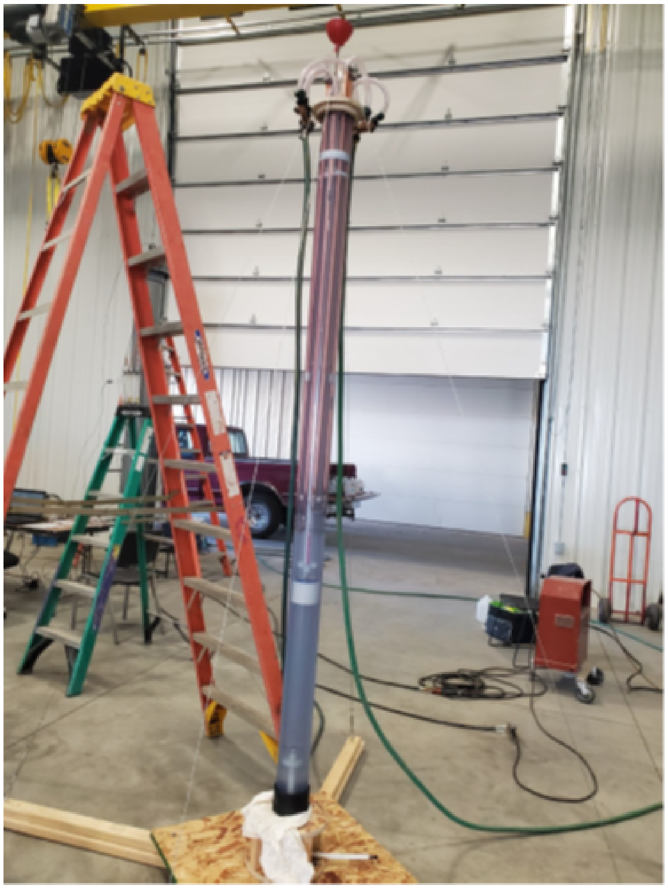


Figure 3.30: Photograph of the Ohmic Heated ”clear vessel”. The heat sink can be seen in the upper half of the assembly as copper piping. The lower portion of the tube column contains the ohmic heat source electrodes. The WHIP is not present in this configuration. Courtesy of Kristen Geddes, Graduate Student at University of Idaho, Nuclear Engineering.

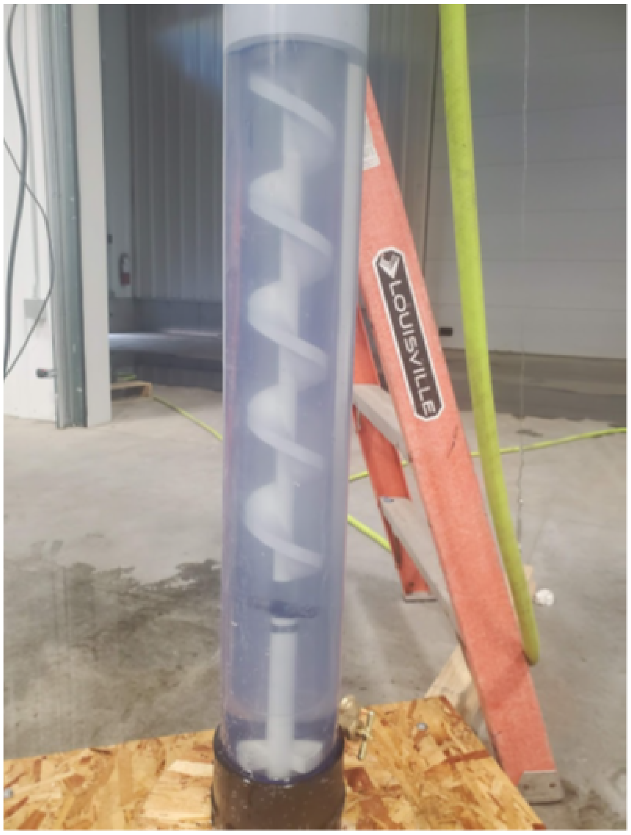


Figure 3.31: Photograph of the Ohmic Heated ”clear vessel” with the 5-pitch WHIP installed. Courtesy of Kristen Geddes, Graduate Student at University of Idaho, Nuclear Engineering.

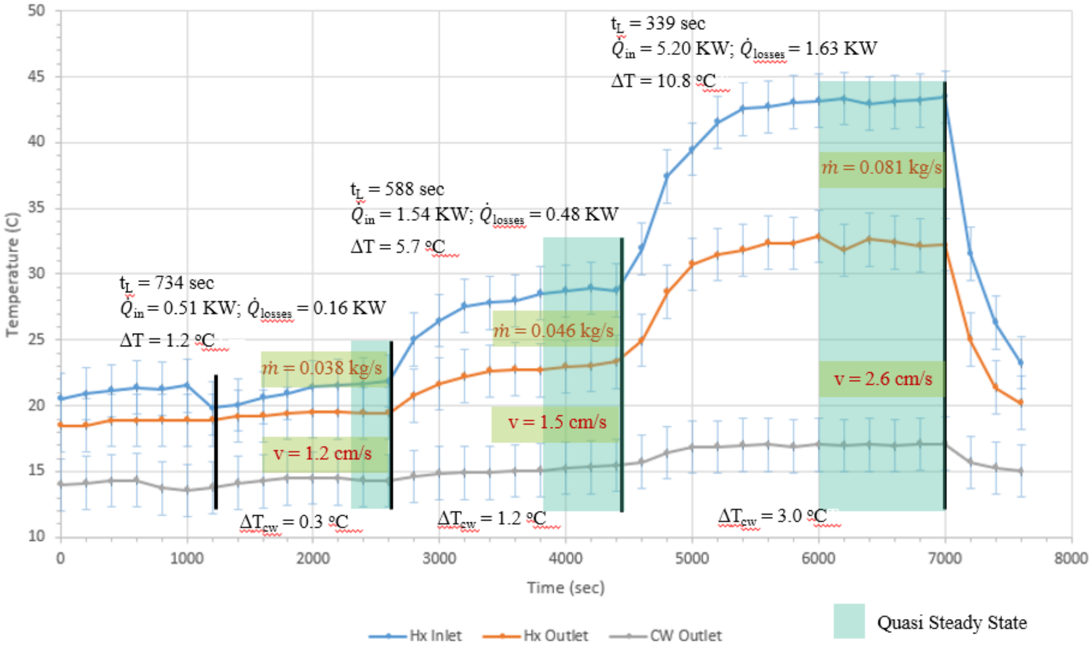


Figure 3.32: "Clear vessel" OH-NC loop data run during 3 heat source power settings. With constant heat removal, increasing input power also results in an increase in system temperature and system mass flow rate as buoyant forces increase due to density differences between hot and cold legs of the loop create the necessarily thermal driving head to support stable flow.

CHAPTER 4

MSR Neutron Noise

4.1 Introduction

The origins of neutron noise theory started at the Clinton Pile or X-10 reactor at Oak Ridge National Laboratory (ORNL). It was observed early on that neutron fluctuations were considerably more complex than simply reactivity-induced ”point-kinetic” power oscillations. Measured neutron flux behaviors include complex, localized, spatial dependencies that dominate near the noise source. In the early 1970s, vibrating control rod induced neutron noise was later observed at the Oak Ridge Research Reactor (ORR) and the High Flux Isotope Reactor (HFIR). Observed flow-induced vibrations at 5-6 Hz were caused by a failed bearing in one of the reactor control rod mechanisms. In the mid-1970s, cross-correlated in-core temperature measurements permitted the calculation of coolant flow, and ex-core neutron detectors were used to determine moderator temperature coefficients in Pressurized Water Reactors (PWRs) (Pazsit, 1977). While application of noise techniques are relevant to MSRs, little practical experience exists in this area. The study of MSR reactor noise involves evaluation of neutron fluctuations induced by perturbations in reactor properties such as core components and fuel salt temperature and density changes.

Monitoring neutron noise in a MSR system and using this data to perform remote diagnostics can be a cost-effective alternative to mechanical and direct human involved maintenance protocols. The WHIP may present a novel host for instrumentation in measuring and diagnosing neutron noise during initial reactor start-up and follow-on throughout core life. MSR digital twin technology and advanced modeling can aid as a predictive tool for ensuring cost constrained life-cycle maintenance. This chapter is motivated by presenting neutron noise in the application of MSRs and briefly describing how the MsNB with WHIP would be expected to modify the traditional solid fueled or larger MSRE-type reactor transfer function. Releasable results are journal published with David Arcilesi (Carter and Arcilesi, 2021).

4.2 General Principles

Any temporal changes in these core properties manifest themselves in corresponding changes in the core cross sections. Concomitant changes in the coefficients of the neutron transport equations are manifest. Changes in cross sections are most commonly caused by the unique random character of turbulent flow or flow-induced vibrations. The associated neutron noise will also have unique random behavior. Changes in the cross section can be a random or deterministic process (Pazsit, 2010). This study will examine the cases specific to flowing fuel MSRs only, and include changes consist of small variations about an expected value. The time-averaged values of the variation are zero. The small variations are referred to as fluctuations, perturbations, or noise source and the induced variations in neutron flux is referred to as neutron noise. The fluctuating cross sections are considered to be stochastic and can be described by their auto-correlations and auto-spectra and the neutron noise can be determined to an extent which is predicated on the fluctuations of the cross sections.

Solutions to the time-independent static equations are important to calculating a complete perturbative solution. Specifically, the static equations are given by,

$$D_o \frac{d^2\phi_o(x)}{dx^2} + \left[(1-\beta)\nu\Sigma_f^o - \Sigma_a^o\right]\phi_o(x) + \lambda C_o(x) \quad = \quad 0 \tag{4.1}$$

and

$$u_o \frac{dC_o(x)}{dx} \quad = \quad \beta\nu\Sigma_f^o\phi_o(x) - \lambda C_o(x) \tag{4.2}$$

with boundary conditions similar to Eqs. 4.7 and 4.8. Eq. 4.2 can be solved by integrating and using the delayed neutron precursor boundary condition which gives,

$$C_o(x) \;=\; e^{-\frac{x\lambda}{u}}\frac{\beta\nu\Sigma_f}{u}\left[\frac{1}{e^{\lambda\tau}-1}\int_{-a}^{a} e^{\frac{x^{'}\lambda}{u}}\phi_o(x^{'})dx^{'} + \int_{-a}^{x} e^{\frac{x^{'}\lambda}{u}}\phi_o(x^{'})dx^{'}\right] \qquad (4.3)$$

where a is the half core height. The first term is the contribution due to all past full-core passes, the second term is the last partial pass-through to core height x. Eqs. 4.1 and 4.2 are coupled ordinary differential equations and their closed-form solutions are left for the literature (Paszit 2011, NS&E, 167, 61-76). In this study, the static flux and delayed neutron precursor concentration are solved numerically and used to calculate the delay neutron precursor source noise given by Eq. 4.15. Fig. 4.1 was generated by the modeling code and used in calculating system noise. The figures show that contrary to traditional solid fueled system, the precursor concentration are skewed above the core axial center line. This result agrees with a previous study by Dulla (Dulla, 2005). The spatial gradient of delayed neutron precursors in the core are an important component to propagating noise in the MSR system.

Using Green's function technique for solving the coupled transport equations, the induced neutron noise can be obtained by a convolution over the noise source with the system Green's (transfer) function. The noise source is a function of cross section or velocity variations and is unique to the source behavior, i.e. flow, density or temperature variations. The system transfer function is independent of sources and is representative of core physics parameters. The transfer function at a given frequency gives the response of the neutron noise at a point x, due to a perturbation at point x' with same frequency ω. Noise due to a fluctuating source has a global or point kinetic (PK) component and a local or space-dependent (sd) component and can be written as,

$$\delta\phi(x,\omega) \;=\; \delta\phi_{PK}(x,\omega) + \delta\phi_{sd}(x,\omega), \tag{4.4}$$

in the frequency domain. The first term is referred to as the point kinetic component and the second component is the space-dependent term. Total noise, $\delta\phi$, can now be interpreted as the complex addition or ”interference” of the PK mode and space-dependent solutions. Unlike solid fueled reactors, the PK component is not the static flux multiplied by a frequency dependent factor. Instead, the PK component in MSRs is a more involved expression (Pazsit et al, 2014). As noted by Pazsit, the phase of the space dependent term closely follows the phase of the noise source (Pazsit, 2010). This convenient relationship makes noise diagnostics possible in practice. The phase of the point kinetic term is usually constant throughout the core. The constructive and destructive interference between the two terms generates the core spatial noise amplitude modulation when neither the PK or space-dependent term dominate. Of course these terms are complex-valued and therefore both their amplitudes and phases play a significant role in interpreting the results in this present study. The ratio of the two terms determines the prevalence of the interference pattern and the modulation of the noise amplitude. At low frequencies ($\omega < \lambda$), the PK term generally dominates. The two terms are more equally weighted in the plateau region where interference effects are observed at non-sink frequencies. For propagating perturbations, sink frequencies are identified as those frequencies where the point kinetic term is zero. At higher frequencies ($\omega > \frac{\beta}{\Lambda}$), the contribution of the PK term is diminished and the phase of the space-dependent term oscillates rapidly. This results in a rapid oscillation in the total noise amplitude that smooths as frequency increases. In MSRs, the amplitude of the zero power transfer function, $G_0(\omega)$, in the plateau region goes as $1/\beta_{eff}$ where $\beta_{eff} = \beta\frac{H}{H+L}$ (Pazsit, 2010a). Therefore, an MSR design that maximizes the ratio of core flow length-to-loop flow length can reduce the Green’s function amplitude.

In small systems or at low frequencies, global neutron noise behavior is point kinetic. Table 4.1 provides a comparison between baseline Dykin and MsNB reactor parameters used in this investigation (Dykin, 2016a). At larger spatial scales and higher frequencies, the neutron noise behavior becomes more dominated by the space-dependent term. As shown in Eq. 4.4, neutron noise can be viewed as the sum of a point reactor noise term driven by the reactivity effect of the perturbation source, and a space-dependent noise term that is driven by the spatial non-uniform character of the perturbation. In most cases, using one-group theory to calculate both the PK term and the space-dependent term are sufficient for noise diagnostics since the system relaxation length (neutron mean free path or characteristic neutron diffusion length) is on the same order as detector and source spacing. Describing neutron noise in the functional form of a PK term and a space-dependent term is also useful for interpreting the results. Two-group theory is needed to resolve localized effects near the source and will therefore not be addressed in this study. In conceptual neutron noise diagnostic applications, the use of homogeneous, bare-core, one-group theory is satisfactory and for this reason will suffice for the purposes of this study (Carter and Arcilesi, 2021). In this study, empirically motivated functional forms of velocity or cross section fluctuations will be used to evaluate the resulting neutron noise in a natural circulation, closed-loop MsNB system.

One-dimensional, one-energy, one-precursor group, coupled diffusion equations in a bare, homogeneous MSR system are given by,

$$\frac{1}{v}\frac{\partial\phi(x,t))}{\partial t} = D_o\frac{\partial^2}{\partial x^2}\phi(x,t) + \left[(1-\beta)\,\nu\Sigma_f(x,t) - \Sigma_a(x,t)\right]\phi(x,t) + \lambda C(x,t) \tag{4.5}$$

and

$$\frac{\partial C(x,t)}{\partial t} + u(x,t)\frac{\partial C(x,t)}{\partial x} = \beta\nu\Sigma_f(x,t)\phi(x,t) - \lambda C(x,t) \tag{4.6}$$

with closed loop MSR boundary conditions,

$$\phi(-a,t) = \phi(a,t) = 0, \tag{4.7}$$

$$C(-a,t) = C(a,t-\tau_L)e^{-\lambda\tau_L}, \tag{4.8}$$

where $2a$ (-a at the inlet and +a at the outlet) defines the reactor axial height. While this simplification may overlook additional physical effects in an operational reactor, this simplified model provides an intuitive interpretation and understanding of fundamental physics present in a real system and is therefore worthy of study and examination as principle-based work. Appendix C provides the mathematical derivation used by the Python program to solve the system of linear equations.

Noise or perturbations in macroscopic cross sections (k is a place holder for interaction type, for instance a is absorption and f is fission), velocity, neutron flux, and delayed neutron precursor concentration can be written in the usual functional form,

$$\Sigma_k(x,t) = \Sigma_k^o + \delta\Sigma_k(x,t), \tag{4.9}$$

$$u(x,t) = u_o + \delta u(x,t), \tag{4.10}$$

$$\phi(x,t) \;=\; \phi_o(x) + \delta\phi(x,t), \tag{4.11}$$

and

$$C(x,t) \;=\; C_o(x) + \delta C(x,t). \tag{4.12}$$

The time-averaged values of these parameters and their fluctuations are the first and second terms on the right-hand side, respectively. The equations are linearized by neglecting products of first order and higher fluctuating quantities. The following Fourier transformed boundary conditions are used, $\delta\phi(-a,\omega) = \delta\phi(a,\omega) = 0$ and $\delta C(-a,\omega) = \delta C(a,\omega)e^{-(\lambda+i\omega)\tau_L}$. Fourier transformed coupled equations can be written in matrix form as,

$$\begin{pmatrix} \partial_x^2 + B_o^2(\omega) & \frac{\lambda}{D_o} \\ -\beta\nu\Sigma_f^o & u_o\partial_x + (\lambda + i\omega) \end{pmatrix} \begin{pmatrix} \delta\phi(x,\omega) \\ \delta C(x,\omega) \end{pmatrix} = \begin{pmatrix} S_\phi(x,\omega) \\ S_C(x,\omega) \end{pmatrix} \tag{4.13}$$

where $B_o^2 = \frac{(1-\beta)\nu\Sigma_f^o - \Sigma_a^o - \frac{i\omega}{v}}{D_o}$. S_ϕ and S_C are the flux and delayed neutron precursor noise sources, respectively, and are explicitly given by,

$$S_\phi(x,\omega) \;=\; \frac{1}{D_o}\left[\delta\Sigma_a(x,\omega) - (1-\beta)\delta\nu\Sigma_f(x,\omega)\right]\phi_o(x) \tag{4.14}$$

and

$$S_C(x,\omega) \;=\; \beta\delta\nu\Sigma_f(x,\omega)\phi_o - \delta u_o(x,\omega)\frac{dC_o(x)}{dx}. \tag{4.15}$$

The second term in Eq. 4.15 is unique to liquid fuel salt systems and notably involves the spatial gradient of the precursor concentration. For propagating (with velocity $u(x,\omega)$) fluctuations in the macroscopic cross sections ($\delta\nu\Sigma_f(x,\omega)$ or $\delta\Sigma_a(x,\omega)$) or fuel salt velocity, $u(x,\omega)$, the expressions can be written as,

$$\delta\nu\Sigma_f(x,\omega) \;=\; \delta\nu\Sigma_f(-a,\omega)e^{-i\omega\frac{x+a}{u_o}}, \tag{4.16}$$

$$\delta\Sigma_a(x,\omega) \;=\; \delta\Sigma_a(-a,\omega)e^{-i\omega\frac{x+a}{u_o}}, \tag{4.17}$$

and

$$\delta u(x,\omega) \;=\; \delta u(-a,\omega)e^{-i\omega\frac{x+a}{u_o}}. \tag{4.18}$$

The mathematical form of the above transformed fluctuations are derived from the propagating fluctuation in the time domain; namely, $\delta P(x,t) = \delta P(0,t-\frac{x}{u})$. If the exponential coefficients are constants, then the noise at the core entrance is random ”white noise”, i.e. all frequency components are equally weighted. Of course this doesn’t have to be the case and appropriately motivated physical shaping functions may be used to appropriately represent phenomena such as temperature fluctuations or flow turbulence induced vibrations in fluid density or velocity. Fluctuations in fuel salt density are manifested as changes in the macroscopic cross section and flow turbulence are manifested as velocity fluctuations. For this reason, perturbations in fuel salt density can be evaluated by including fluctuations in $\delta\nu\Sigma_f$ or $\delta\Sigma_a$. Fuel salt density or velocity changes can be produced by flow turbulence in the reactor system. The spectral character of these phenomena are well understood and can be used to shape the neutron flux and delayed neutron precursor concentration sources, S_ϕ and S_C, respectively. Flow turbulence, for instance, has a well-defined $\omega^{-5/3}$ spectral behavior while a density resonance may have a peaked spectral resonance of form $S_{pk}e^{-\omega^2/2\sigma^2}\omega^{-\alpha}$. As mentioned earlier, in small systems or at low frequencies, the global behavior is point kinetic. In large systems or at higher frequencies, however, there is a transition from point kinetic behavior to space-dependent behavior.

The frequency dependence of the system's transfer function (amplitude and phase) generally has a plateau region between the one-group delayed neutron precursor decay parameter, λ, and the effective delayed neutron fraction-to-prompt neutron generation time ratio. In this region, the transfer function amplitude is nearly constant and proportional to the reciprocal of the effective delayed neutron precursor fraction. The transfer function phase is near zero in this region and equal to $\pi/2$ elsewhere. Outside the plateau region the transfer function generally decreases as the inverse of frequency. The physical interpretation of the plateau region can be explained in terms of the transfer function's ability to bound the dynamic system time scales. The system tends to be most responsive to perturbations on time scales greater than the delayed neutron precursor half-life. Conversely, perturbations that occur on time scales less than the prompt neutron generation time have reduced effect on system response. On time scales between these two plateau bounds, the system is effectively tuned to couple noise sources to concomitant effects in neutron noise. The amplitude of the plateau increases as the fraction of prompt neutrons that comprise all fissions increases. In other words, in the plateau region, source coupling to neutron flux noise increases with the fraction of fissions caused by prompt neutrons.

The matrix Green's functions associated with this system of coupled equations can be written as,

$$\begin{pmatrix} \partial_x^2 + B_o^2(\omega) & \frac{\lambda}{D_o} \\ -\beta\nu\Sigma_f^o & u_o\partial_x + (\lambda + i\omega) \end{pmatrix} \begin{pmatrix} G_{\phi\phi} & G_{\phi C} \\ G_{C\phi} & G_{CC} \end{pmatrix} = \begin{pmatrix} \delta(x - x^{'}) & 0 \\ 0 & \delta(x - x^{'}) \end{pmatrix} \tag{4.19}$$

where x and $x^{'}$ are the core detector and source positions, respectively. Using the Green's function boundary conditions provided by V. Dykin (Dykin, 2016a), the Green's functions and hence the full core space-frequency neutron flux noise, $\delta\phi$, and full core space-frequency delayed neutron precursor fluctuations, δC, can be determined by integrating the source terms, S_ϕ and S_C, times the Green's function over all possible core source positions as follows,

$$\delta\phi(x,\omega) = \sum_{n=1}^{2} \int_{-a}^{a} G_{1,n}(x,x^{'},\omega)S_n(x^{'},\omega)dx^{'} \tag{4.20}$$

and

$$\delta C(x,\omega) = \sum_{n=1}^{2} \int_{-a}^{a} G_{2,n}(x,x^{'},\omega)S_n(x^{'},\omega)dx^{'} \tag{4.21}$$

where n = 1,2 are the neutron flux ϕ and delayed neutron precursor concentration C, respectively (Dykin, et al., 2016). Calculated Green's functions for the Dykin baseline case are shown in Fig. 4.2.

4.3 Validation to Published Works

Analytically and numerically calculated dynamic Green's functions (transfer functions) for a one-group, one-dimensional homogeneous system have been used to determine neutron noise induced by changes in fuel salt cross sections. This previously published work presents both point-kinetic and space dependent terms of the neutron noise (Dolan, 2017; Pazsit et al., 2012; Dulla, 2005). The Python-code based model developed for this investigation provides consistent calculations with previously published results. The point-kinetic and space-dependent terms of the neutron noise are in good agreement with previously published results. Some transfer function visualizations produced by the modeling code of previously published 1-D results are shown in Figs. 4.2 and 4.3. This provides confidence that the model calculations can be extended to investigate MSR phenomena that induce neutron noise.

Figs. 4.2 and 4.3 show that key properties of the Green's function amplitudes noted in previous work are reproduced in the present Python-code results (Dykin, 2016a). Specifically, the frequency dependence of all amplitudes diverge as ω^{-1} and have a plateau region ($\lambda \ll \omega \ll \frac{\beta}{\Lambda}$) where regular peak patterns due to delayed neutron precursors are observed. In the specific case of the MsNB, the plateau region lies between 0.1 to 43.0 radians per second. As expected, the lowest frequency peak corresponds to the reciprocal of the total system recirculation time. Also present in model results shown in Fig. 4.2 and 4.3(d) is a constant G_{22} kinetic transfer function component for frequencies above the plateau region. G_{22} relates the delayed neutron precursor source S_C with the delayed neutron precursor density fluctuations, δC. In a qualitative sense, delayed neutron precursor decays are negligible during periods of fluctuation less than the decay half-life and therefore the amplitude is constant above those frequencies. This non-decaying behavior of G_{22} is due to precursor transport and is a unique characteristic of flowing fuel systems. Also consistent is the result that the zero power reactor transfer function $G_o(\omega)$ of one-group diffusion theory has an amplitude that varies as the inverse of the delayed neutron fraction, β. Figs. 4.2, 4.3, and 4.4 show similar results and provides confidence that the Python-code is consistent with previously published results.

The model code will be used in this investigation to extend upon previous work to introduce the study of velocity and cross section induced fluctuations caused by flow and temperature changes.

4.4 Noise Sources

Eqs. 4.16 through 4.18 each have an arbitrary frequency dependent coefficient that uniquely defines the nature of the perturbation. If the coefficients are constant in time or frequency, the representative perturbation represents white noise at the core entrance. A frequency dependent coefficient as described below for turbulence or fuel salt temperature transients will manifest as unique spectrum neutron noise.

A diagnostic problem typically results in the transforms being solved in the reverse direction. That is, one or more arrayed neutron detectors measure cross-correlated neutron noise signals. Using informed assumptions about the nature of the neutron noise source measured at discrete spatial locations within and around the core, and knowledge of the system transfer function, the convolution can be solved in reverse to provide a good characterization of the source frequency and location. This study will only examine the nature of three sources in a simplified one-dimensional case to gain an intuitive understanding of how these sources are manifest in neutron noise. This intuitive understanding can be extended into higher dimensional analysis.

In the sections that follow, simple models for flow and density fluctuations are introduced. Localized velocity fluctuations caused by flow turbulence and critically damped temperature fluctuations at the core inlet are investigated. Temperature fluctuations at the core inlet might be caused by such effects as changing heat demand on the system. Sources of noise can be displacement-type motion such as vibration of a solid reactor, or core component or temporal changes in the cross sections due to flow velocity or temperature (density) changes. To provide a comparison case, propagating velocity fluctuations with constant power spectrum coefficients are calculated. This comparison case represents a propagating white noise source at the core inlet. This baseline case will be used as a comparison to the noise response of the MSR to functional noise sources representing flow and density noise sources. Green's transfer amplitudes for the baseline results are shown in Fig. 4.3. Baseline neutron and precursor noise results are shown in Fig. 4.4. Contours show the expected transfer function behavior explicitly. Loop recirculation peaks and nulls in precursor gradients are also noticeable.

4.4.1 Turbulent Flow (High Re) induced Velocity Fluctuations, δu

Natural circulation flow of the fuel salt in the MSR system can be turbulent in nature during most all operational modes and times of core life. Calculations of system Reynolds numbers is approximately 3000. Moreover, there may be localized flow regions during end-of-life (EOL) core power transients that induce localized intermittent turbulent flow due to variations in kinematic viscosity and flow uniformity as insoluble fission-product fluorides deposit and accumulate on flow surfaces. Localized flow conditions may develop due to the different speeds and conditions of the fluid. The effect of these localized variations may propagate in the fluid system. Fuel salt velocity fluctuations due to flow turbulence may have a Kolmogorov (Balk, 2000; Hunt, 1991; Frisch, 1995) spectral shaping function of the form,

$$\delta u(-a,\omega) \quad = \quad S_{pk}\omega^{-\frac{5}{3}}. \tag{4.22}$$

This investigation will use a source peak amplitude of 100 $\mathrm{cm/s}^{7/3}$.

4.4.2 Variations in Fuel Salt Concentration induced Absorption Cross-Section Fluctuations $\delta\Sigma_a$ and $\delta\Sigma_f$

Fuel salt density fluctuations are expected to be a likely source of neutron noise in a flowing fuel system. Such fluctuations might result, for instance, from fuel salt density variations due to thermodynamic heat transfer transients to the secondary cooling system. Temperature transients would manifest as perturbations in the macroscopic fission and absorption cross sections, $\delta\nu\Sigma_f(-a,\omega)$ and $\delta\Sigma_a(-a,\omega)$. As noted by Dykin, it is sufficient to only consider fluctuations in the absorption cross section (Dykin et al., 2016a). It is sufficient to assume that the ratio of fluctuations in the thermal absorber density to the fluctuations in the fuel density can be described as a constant over the frequency range of interest. This ratio is defined by γ = $\delta\Sigma_a$/$\delta\nu\Sigma_f$ and is taken to be unity for this study. Fuel salt velocity fluctuations as a result of mechanically induced resonance might have a spectral shaping function of the form,

$$\delta\Sigma_a(-a,\omega) \quad = \quad S_{pk}\frac{e^{-\frac{(\omega-\omega_S)^2}{\sigma^2}}}{\omega}. \tag{4.23}$$

where ω_s is the source resonance frequency, S_{pk} is the source peak amplitude, and σ is the 1/e source frequency drop-off.

This investigation will use a transient duration, τ of 2400 seconds, a temperature fluctuation frequency, ω_s of 5.31x10^{-4} sec^{-1} (this corresponds to about 8 temperature cycles in a period of 40 minutes), and a peak source change of 1.0 $\frac{1}{\mathrm{cm-s}}$. This value of S_{pk} is representative of a nominal change in cross section at nominal reactor neutron energies ($\approx$ 0.175 eV). The 1/e drop-off is assumed to be on the order of the source resonance.

As noted in Eqs. 4.20 and 4.21, although perturbations in velocity and density have similar forms, the convolution with the source and system transfer function produce distinct differences in the calculated neutron noise.

4.5 WHIP incorporation

Employing an in-core WHIP feature has only a slight effect on the core Green's function. The core remains closely coupled neutronically as discussed in chapter five. While the bare Buckling factor, B_o, and effective delayed neutron fraction, β_{eff} do change as a result of the WHIP configuration as will be discussed in chapter five, the overall effect is negligible. β_{eff}, for instance, changes by approximately 4 % (0.88 to 0.92 of the static β_{eff}). Although this does result in a change to the Buckling and delayed neutron effects, these changes are small to do not significantly alter the calculated kinetic transfer function of the system. This is to be expected since the core with or without a WHIP remains neutronically closely coupled and physics perturbations transfer similarly in both systems.

4.6 Results and Summary

The functional form of some example propagating fluctuations are given by Eqs. 4.22 and 4.23. The coefficient to the exponential (for example, $\delta u(-a, \omega)$) may have many physically motivated forms that empirically represent flow turbulence or system density fluctuations. Fig. 4.6 shows the MsNB Green's functions for a case where the source position is at core center line and the coefficient in the velocity fluctuation, $\delta u(-a, \omega)$, is a constant. For propagating white noise at the reactor inlet, the system response to local perturbations is more localized at higher frequencies. The system response below 0.1 rad/sec is spread throughout the volume of the core and aided by the precursor transport in a liquid fueled system. Fig. 4.7 illustrates the calculated neutron and precursor noise in the core as a function of noise frequency for the constant coefficient to the perturbation. The core exhibits a strong interference pattern between the point kinetic and space dependent terms. The frequency dependence of the phase of the fluctuation and the pure space dependent noise term are similar while the point kinetic phase is constant throughout the core. The maximum near 1 rad/s is associated with the first recirculation of precursors to the core. At high frequencies, the contribution of the point kinetic term diminishes compared to the space-dependent term but is not negligible. Since the phase frequency of the space dependent term is high, the total noise amplitude also oscillates at a high frequency. This is observed at frequencies greater than approximately 3 rad/s. The amplitude minimum at about 25 cm is due to the vanishing precursor axial gradient at that location in the core. High frequency peaks in the precursor noise are due to in-phase constructive interference between the G_{CC} and $G_{C\phi}$ transfer function and propagating source fluctuation phases. Notice that maximum noise amplitude is on the same order as the core neutron flux at $10^{13}\ \mathrm{neutrons/cm^2\,s}$. Generally, sharp peaks with soft digs are associated with recirculation in the system loop while sharp dips with soft peaks result from propagation within the core.

While each of these perturbations are worthy of detailed independent study, this paper will only briefly address each phenomenon to highlight the potential for future study to improve intuitive understanding of neutron noise characteristics in MSR cores.

4.6.1 Flow-Turbulence induced Noise

Turbulence-induced flow fluctuations are empirically represented using Eq. 4.22. Results are shown in Fig. 4.7. While the source peak amplitude was chosen as an order of magnitude estimation, it is noted that the sensitivity on low-frequency noise is significant. The subtle features associated with the point kinetic and space-dependent terms are still observable, the flow turbulence has the effect of dominating the low-frequency noise response in the core. Neutron and precursor noise both have frequency mid-range relative maximum near 1 rad/sec and have noise nulls at the precursor gradient zero. Space-dependent interference patterns are dominant across the precursor noise spectrum.

In practice, precursor noise is not easily measurable. Therefore, except for noise amplitude, a spectrum perspective turbulence is not easily discernible from the constant coefficient velocity fluctuation case shown in Fig. 4.6.

4.6.2 Fuel Salt Density Fluctuations induced Noise

The Fourier transform of a natural circulation MSR reactor system temperature response to a power up-demand is modeled using Eq. 4.23. System responses to power demand changes are critically damped sinusoids. Oscillation and dampening form constants were chosen based on numerical results for the MsNB. The thermal-hydraulic transient that results from a power demand change can take on the order of a few hours before quasi-equilibrium conditions are re-established in the system. This results in a very-low frequency source in the cross-section fluctuation provided by Eqs. 4.16 and 4.17. The form of the source is given by Eq. 4.23. The mean temperature fluctuation frequency, ω_s, is 5.31x10^{-4} sec^{-1} (5 minutes per thermal cycle), the transient duration, τ, of 2400 seconds (0.7 hours), and peak source amplitude, S_{pk}, of 1.0 $\frac{1}{\mathrm{cm-s}}$. These values are based on numerical results for a MsNB that experiences a 300% up-power demand.

Noise response to a nominal thermal-hydraulic system transient (25% to 100% rated power) is shown in Fig. 4.8. Low frequency noise amplitudes in response to the cross-section fluctuation are significant. All frequencies are dominated by the space dependent term. Many of the finer features present in the baseline constant coefficient case are no longer observable due to the presence of the transient source. Notably, the neutron noise has a sharp minimum in the frequency mid-band above 25% of the core axial height and a relative maximum in precursor noise in this same frequency mid-band. The high frequency precursor noise spectrum becomes more detailed than the case with turbulence. Noise resulting from density fluctuations also do not exhibit the familiar noise null caused by the null in the precursor gradient at 25 cm. While precursor noise measurements are challenging in practice, cross-correlated neutron noise across multiple axial detectors can be used to characterize the nature and location of the neutronic noise.

In general, with knowledge of the system transfer functions, the noise character and location can be determined using the measured noise spectrum using a discrete number of detector locations and numerically solving for the source term in Eqs. 4.20 and 4.21. As shown in the two cases above, knowledge of the noise spectrum even at discrete locations in the core can be used to discern the source of noise, i.e. propagating turbulence or density fluctuations. Time dependent density changes in the core have a very different noise character then flow induced noise.

Investigation into the noise behavior of the molten salt reactor system is essential to understanding core physics and thermal-hydraulic performance of core life. Study of MSR neutron noise provides an informed understanding of MSR physics to build confidence in the safety basis case. In particular, this investigation presented a simplified model that solves the coupled MSR diffusion equations to first order perturbations. The investigation shows the model results are consistent with previous published work.

Furthermore, result show that point kinetic behavior is present at higher frequencies in the MsNB system and that because of low system flow, the micro-scale ($\sim$ 2 meters), natural circulation MsNB noise behavior is similar to a medium to large sized ($\sim$ 4-5 meters) MSR systems, e.g., the molten salt reactor experiment (MSRE). Periodic sink structure of the point kinetic term is visible at higher frequencies than expected for small systems, indicating that a natural circulation small system has the spectral noise characteristics of a much larger MSR system. In addition, fluctuations induced in an MSR by a propagating source have noise amplitudes larger than comparable traditional reactor systems due to the enhanced neutronic coupling obtained by fuel convection.

Turbulent MSR flow can potentially induce large amplitude, low-frequency noise into the system. The spectral character of turbulence induced noise is similar to other noise that might be induced by velocity fluctuations in the system. Noise induced by perturbations in fuel salt density manifest a different spectral character than velocity perturbations. Noise induced by density fluctuations do not exhibit a noise null at the precursor gradient zero. Density induced noise also has a broad space dependent interference behavior with a sharp constant minimum at mid-range neutron noise frequencies. Noise diagnostics on MSR systems offers the potential for effective and inexpensive preventative monitoring for a wide range of local and system performance attributes without the need for intrusive sampling.

4.7 Recommendations for Further Work

Specific recommendations for follow-on work to this study might include:

- The author has developed a time-dependent, finite-element neutronic and thermal-hydraulic model of the closed-loop MSR system. The model calculates time-dependent flow and temperature reactivity effects on reactor power due to changes in system heat demand. Fourier transformed variations of reactor temperature might provide additional insight to dynamic sources of neutron noise in a MSR.
- Conduct a dedicated study of neutron multiplication factor dependence on flow velocity by iteratively solving the MSR criticality equation numerically for various velocities. The result would inform an improved time-dependent power transient model that might provide practical neutron noise predictions.

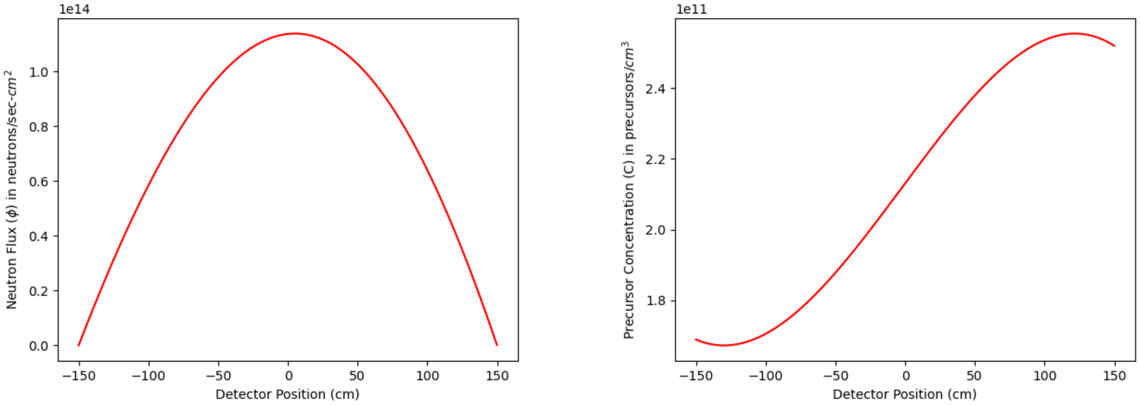
- Develop a diagnostic system based on an array of cross-correlated neutron noise measurements during reactor operation. Improve upon methods to determine nature and location of sources and characterize unique features of spectral signatures associated with sources. As a mitigation and monitoring against thermo-chemical effects late in core life, a system of cross-correlated neutron detectors embedded in in-core moderator features might be very useful to diagnose and monitor MSR thermal-hydraulic performance over core life.

- Expand the investigation into MSR noise sources of interest including fuel salt density and temperature fluctuations. Late in-core life thermal-hydraulic flow performance may manifest interesting and important neutron noise behaviors. Understanding these behaviors is essential to establishing confidence in a safety basis analysis.

- Expand the numerical code to include two-group calculations to help conduct investigations into localized neutron noise source diagnostics. Pazsit et al. have studied noise phenomenology in PWRs and BWRs (Pazsit, 2017; Behringer, 1979).

Table 4.1: Dykin Baseline and Micro-Scale MsNB Molten Salt System Parameters (Dykin, 2016a, 2016b; Carter, 2020)

Parameter	**Baseline (Dykin, 2016a)**	**MsNB**
$2a$ (cm)	300.0	166.0
L (cm)	400.0	405.0
u_o (cm/s)	50.0	3.0
τ_L (sec)	8.0	135.0
D (cm)	0.33	0.874
$\nu\Sigma_f^o$ (cm^{-1})	0.00100774	0.366084
Σ_a^o (cm^{-1})	0.001	0.366
λ (sec^{-1})	0.1	0.0928
Λ (sec)	1.21x10^{-3}	1.63x10^{-4}
β	0.0065	0.0070
v $(\mathrm{cm/s})$	1.7549x10^5	1.1765x10^6



(a) Axial Flux Distribution.

(b) Axial Precursor Concentration Distribution.

Figure 4.1: Baseline axial Delayed Neutron Precursor Concentration Distribution for the baseline case presented by Dykin (Dykin, 2016a). At baseline loop size and flow velocity, precursors returning from the loop contribute delayed neutrons to core inlet.

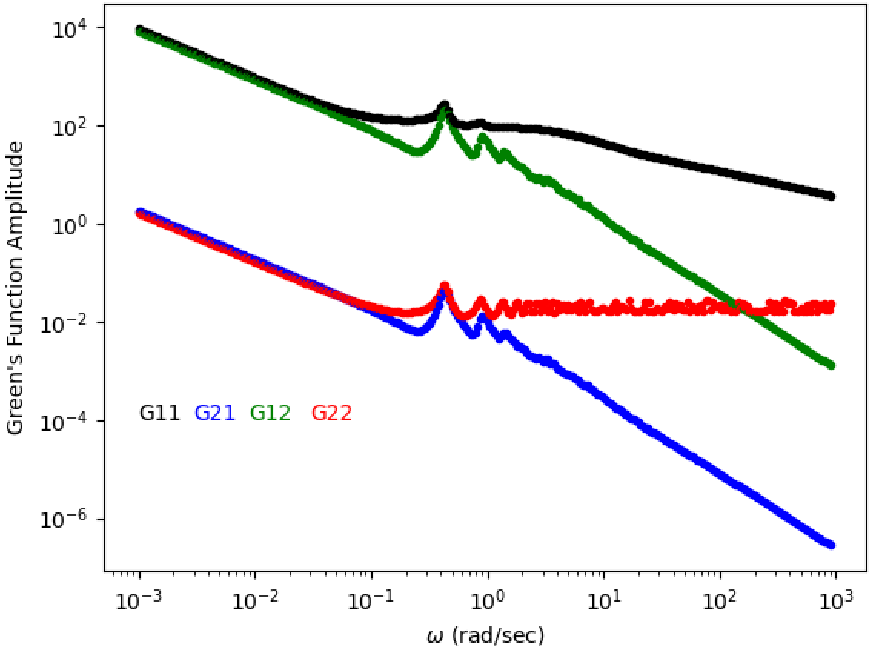
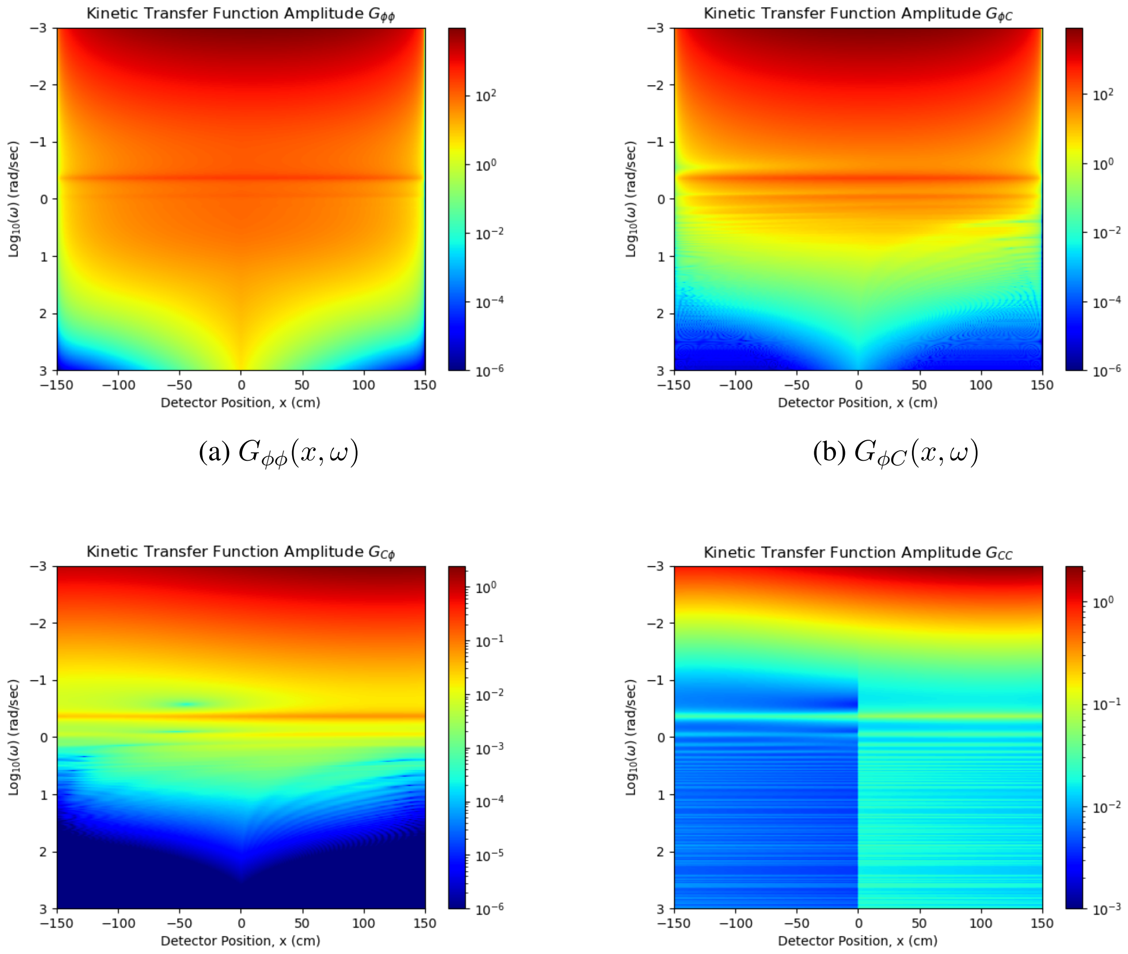


Figure 4.2: Baseline Transfer Functions illustrating $1/\omega$ dependence and precursor loop return peaks for the baseline case presented by Dykin (Dykin, 2016a). Constant G_{CC} (G_{22}), shown in red, at higher frequencies is a unique result for flowing fuel systems. $x = x' = 0$

.



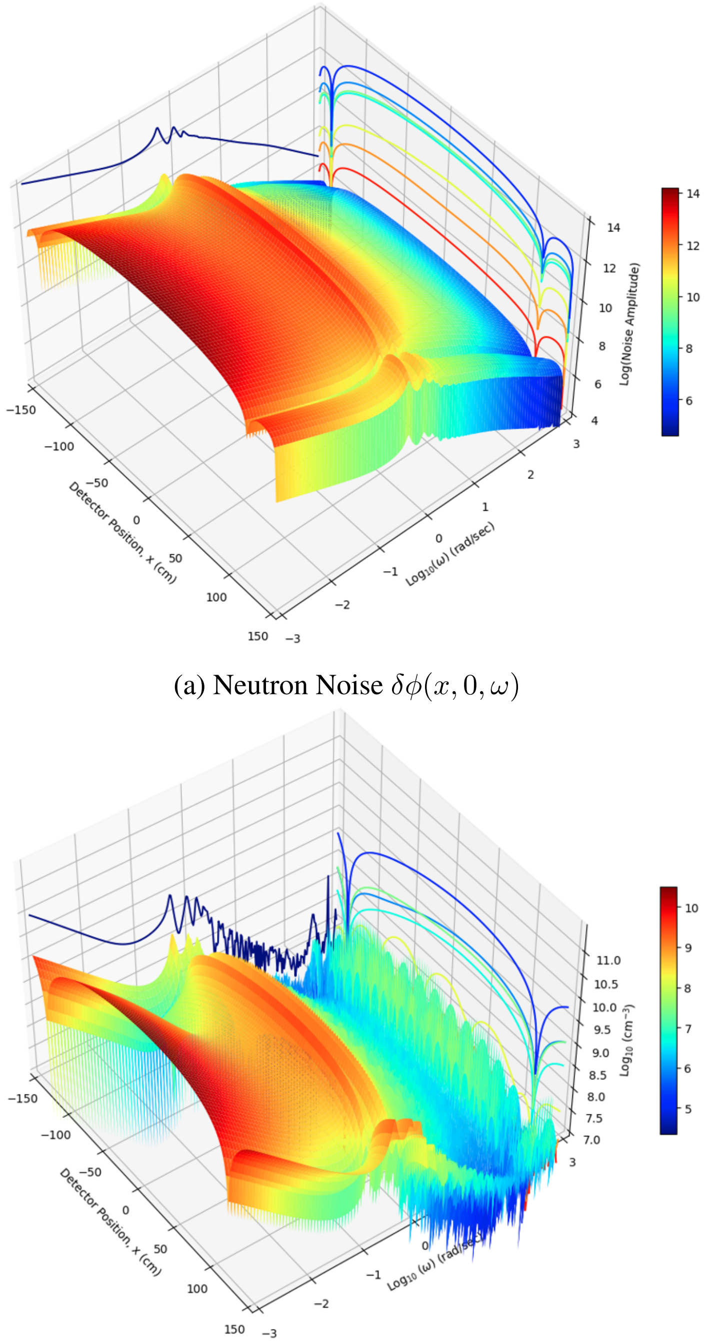
(a) $G_{\phi\phi}(x,\omega)$

(b) $G_{\phi C}(x,\omega)$

(c) $G_{C\phi}(x,\omega)$

(d) $G_{CC}(x,\omega)$

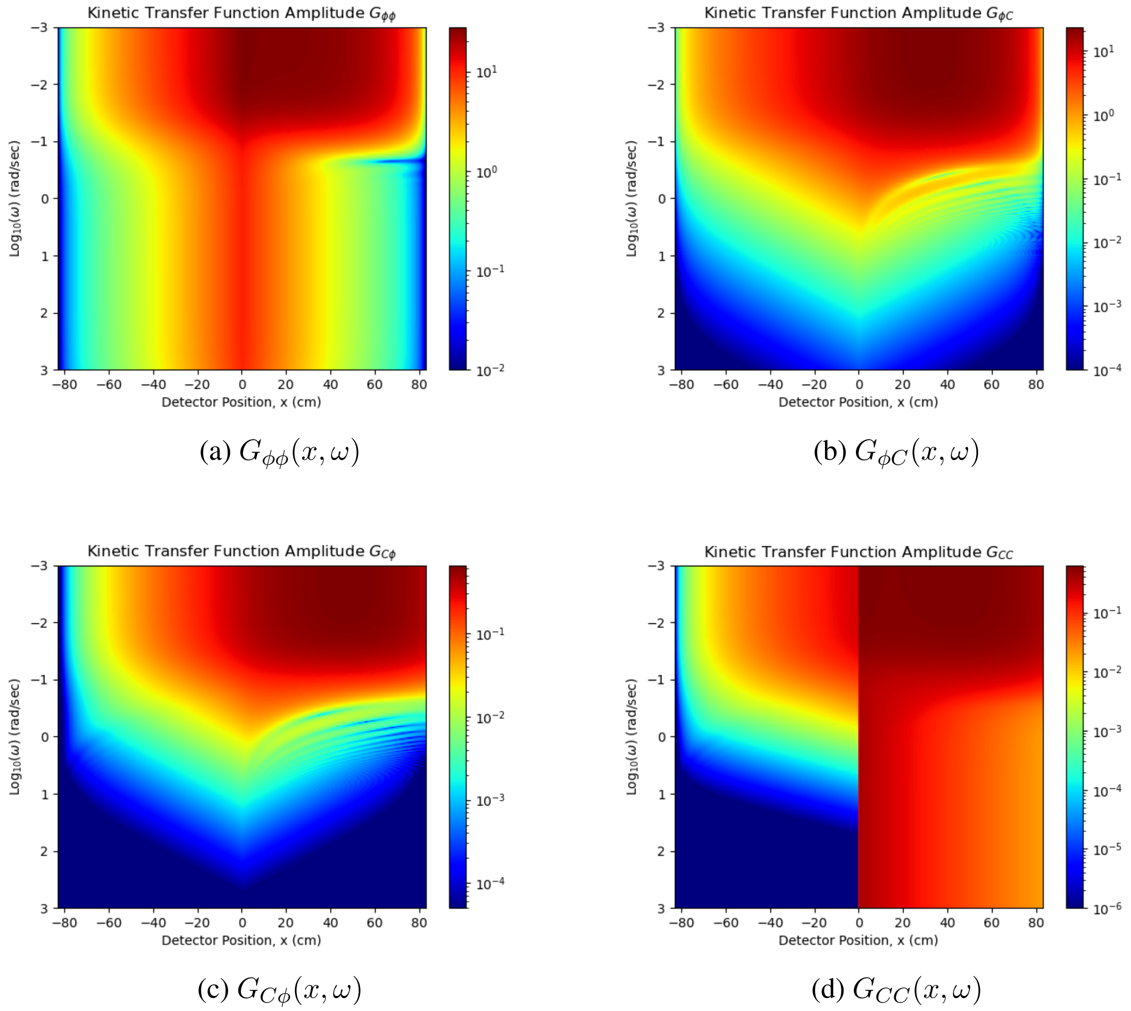
Figure 4.3: Green's Transfer Functions for the baseline case presented in Dykin located at $x^{'}$ = 0 cm (Dykin, 2016a). $1/\omega$ dependence is discernible and consistent with previous published work. The frequency dependence varies with location in the core. At higher frequencies, the noise contribution due to velocity fluctuations is localized to the source. At frequencies below the plateau this is not the case. The G_{CC} transfer function discontinuity at the source location is due to the G_{CC} boundary condition at the source location. That is, $G_{CC}(x,x^{'},\omega)|_{x=x^{'}_{+}}$ = $G_{CC}(x,x^{'},\omega)|_{x=x^{'}_{-}}$ + $1/u_0$.)



(a) Neutron Noise $\delta\phi(x, 0, \omega)$

(b) Fluctuations in Precursor Concentration $\delta C(x, 0, \omega)$

Figure 4.4: Neutron and precursor noise for the baseline case presented by Dykin located at $x^{'}$ = 0 cm (Dykin, 2016a). Inlet and outlet nulls in noise are caused by vanishing precursor gradients. Local noise peaks at about Log$_{10}\omega$ = 0.3 rad/sec and 0 rad/sec are caused by precursor loop recirculation. Space dependent phase interference patterns are evident in the precursor noise at higher frequencies. Notice the 1/ω behavior outside the plateau range is present.



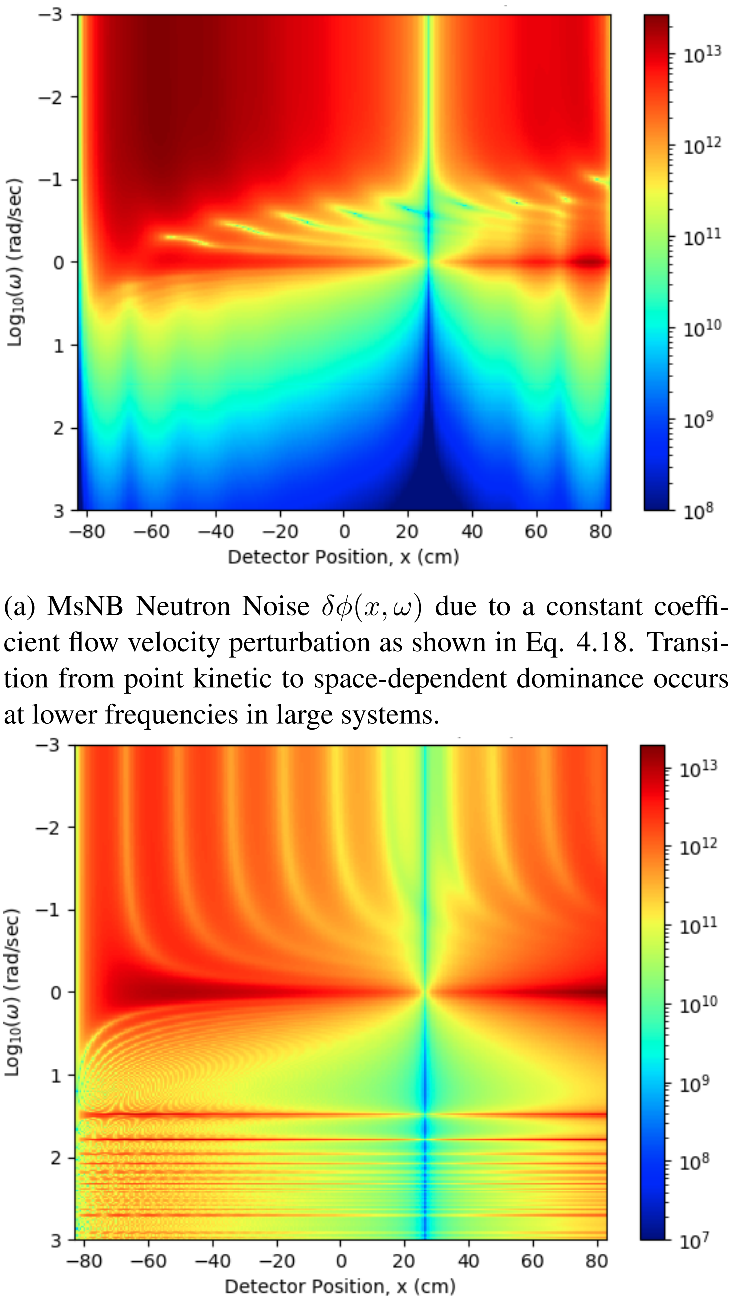
(a) $G_{\phi\phi}(x, \omega)$

(b) $G_{\phi C}(x, \omega)$

(c) $G_{C\phi}(x, \omega)$

(d) $G_{CC}(x, \omega)$

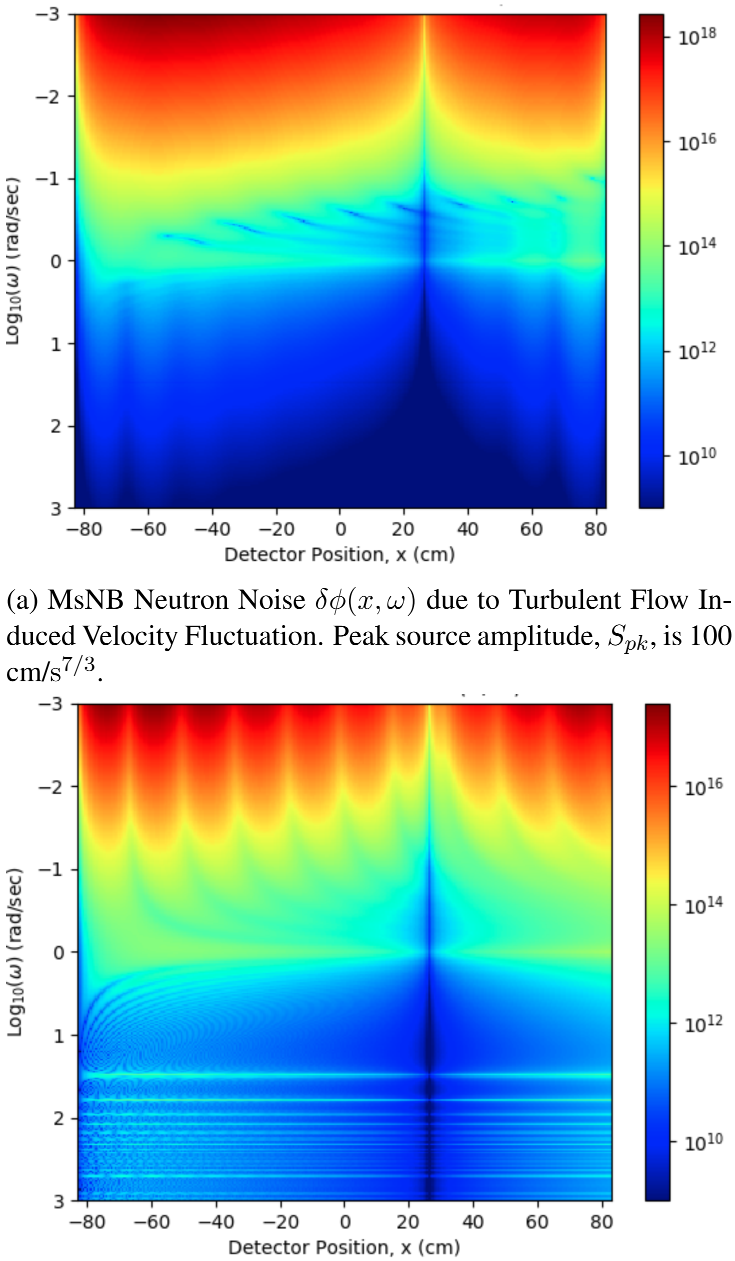
Figure 4.5: MsNB Green's Transfer Functions at $x^{'}$ = 0 cm. The spatial dependence of the Green's function becomes more localized above $\omega > 0.1$ rad/sec. This implies that system response to local perturbations becomes more localized at higher frequencies. This is due to the greater neutronic coupling (precursor transport) in MSR systems. The G_{CC} transfer function discontinuity at the source location is due to the G_{CC} boundary condition at the source location. That is, $G_{CC}(x, x^{'}, \omega)|_{x=x^{'}_{+}}$ = $G_{CC}(x, x^{'}, \omega)|_{x=x^{'}_{-}}$ + $1/u_0$.



(a) MsNB Neutron Noise $\delta\phi(x,\omega)$ due to a constant coefficient flow velocity perturbation as shown in Eq. 4.18. Transition from point kinetic to space-dependent dominance occurs at lower frequencies in large systems.

(b) MsNB Precursor Noise $\delta C(x,\omega)$ due to a constant coefficient flow velocity perturbation as shown in Eq. 4.18. Rapid oscillations of small amplitude correspond to the periodicity of the phase of the fluctuating noise source. Higher frequency peaks are caused by precursor gradient and transfer function phase interference.

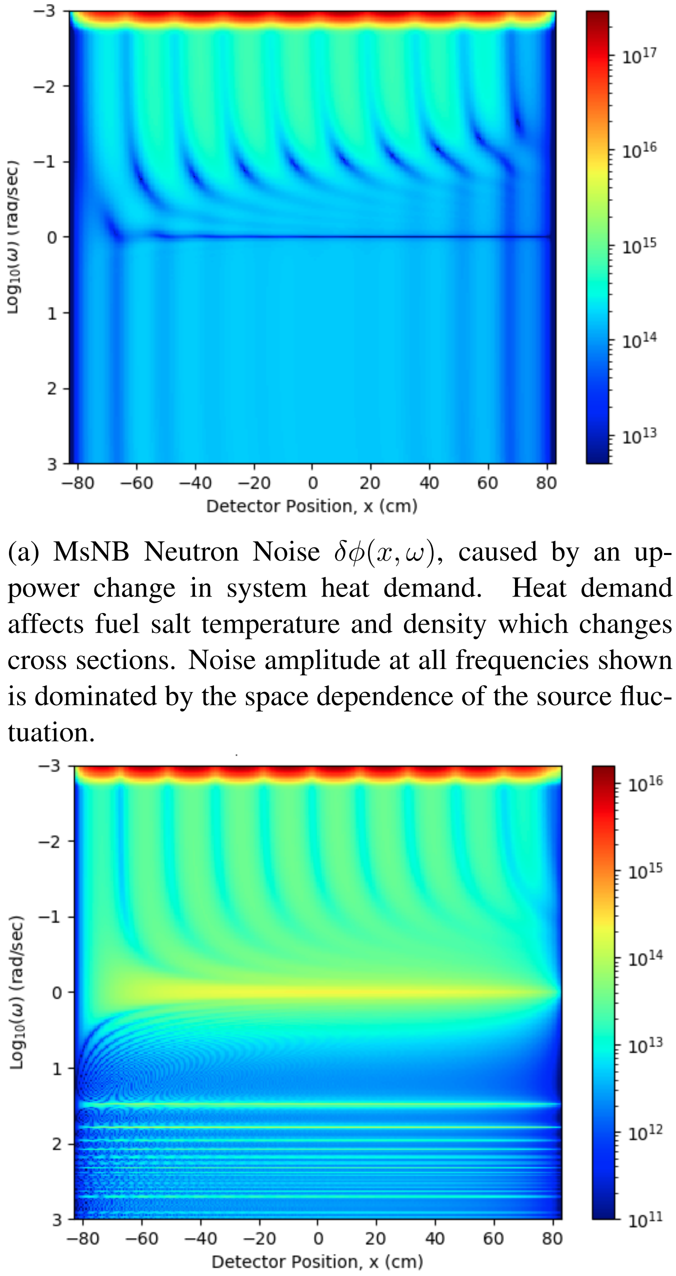
Figure 4.6: MsNB Neutron and Precursor Noise for a constant coefficient velocity perturbation. $\delta u(-a,\omega)$ = constant.



(a) MsNB Neutron Noise $\delta\phi(x,\omega)$ due to Turbulent Flow Induced Velocity Fluctuation. Peak source amplitude, S_{pk}, is 100 cm/s$^{7/3}$.

(b) MsNB Precursor Noise $\delta C(x,\omega)$ due to Turbulent Flow Induced Velocity Fluctuation. Peak source amplitude, S_{pk}, is 100 cm/s$^{7/3}$.

Figure 4.7: MsNB Neutron and Precursor Noise for a Turbulent Flow Induced Velocity Fluctuation. Peak source amplitude, S_{pk}, is 100 cm/s$^{7/3}$.



(a) MsNB Neutron Noise $\delta\phi(x, \omega)$, caused by an up-power change in system heat demand. Heat demand affects fuel salt temperature and density which changes cross sections. Noise amplitude at all frequencies shown is dominated by the space dependence of the source fluctuation.

(b) MsNB Precursor Noise $\delta C(x, \omega)$, caused by an up-power change in system heat demand. Heat demand affects fuel salt temperature and density which changes cross sections. The peak near 1 rad/sec due to recirculation of precursors. Higher frequency peaks are caused by precursor gradient and transfer function phase interference.

Figure 4.8: MsNB Neutron and Precursor Noise for an up-power change in system heat demand. Heat demand affects fuel salt temperature and density which changes cross sections. The mean temperature fluctuation frequency, ω_s, is 5.31x10^{-4} sec^{-1}, the transient duration, τ, is 2400 seconds, and the peak source amplitude, S_{pk}, is 1.0 1/cm-s.

CHAPTER 5

WHIP MsNB Load-Following Power Transient Behavior

5.1 Passive Load Following

Understanding the time-dependent behavior of this liquid fueled natural circulation reactor system is of particular interest since the *flow reactivity* causes an inherently destabilizing effect on reactor power. For instance, as heat exchanger power demand decreases, the heat exchanger outlet and average cold leg temperature initially raises which causes a reduction in downcomer fuel salt density. This reduction in density causes an accompanying reduction in hot and cold leg differential pressure and system flow rate. Decreasing system flow has a positive reactivity effect on the core because proportionally fewer delayed neutron precursors advect out the core and produce added delayed neutrons in the core. Therefore, as heat exchanger power demand on the reactor system decreases the reactor power generation increases due to fuel salt flow changes. The reverse effect occurs for increases in system power demand. This mechanism is referred to as flow reactivity. A time-delayed temperature mechanism associated with the arrival of the colder fuel at the core inlet inserts negative reactivity and eventually corrects the opposing power demand behavior. This temperature reactivity effect is common in traditional solid fuel designs, the flow reactivity mechanism is unique to liquid fueled reactors and as a volumetric effect is less well understood. These two counter-acting compound dynamic feedback reactivity effects are modeled using a Python program to investigate the stability of such a thermal-neutronic system.

The implementation of passive cooling and heat transfer in thermal engineering has resulted in high reliability and compact size in system design. Many studies have been conducted to understand the stability and characterization of flow dynamics in natural circulation systems in single phase and two-phase flow regimes. In the mid-1960s, Welander and others published seminal works studying the flow stability of differentially heated loops (Welander, 1967). Many extended this work and further characterized flow dynamics and the nature of instabilities of natural circulation fluid systems in the application of passive energy removal in nuclear applications. The phenomenology of flow instabilities is manifestly chaotic due to the non-linear nature of the energy and momentum conservation equations that govern the fluid behavior. For this reason, numerical analysis is often well suited to provide insight into dynamic system behaviors. Pini, for instance, showed that stability improves with internal heated fluids (Pini, 2016). Using numerical analysis, Desrayand and others showed the significant effect of the Reynolds number on the flow characteristics in the loop (Desrayandi et al, 2005)(Desrayandi et al, 2013). Nayak also used numerical techniques to investigate stability characteristics of a natural circulation loop and the effects of nanofluids on flow behaviors which is germane to the ”ionic soap” that is the salt fuel, i.e. fission product nanofluid (Nayak, 1995)(Nayak et al, 2005).

This investigation uses a one-dimensional simplified model to evaluate the effects of two opposing, time-delayed reactivity mechanisms that act as the principle feedback to enable autonomous, power-following reactor control. The compound dynamic feedback is unique to liquid fueled reactors and natural circulation during power operations makes flow reactivity a significant power control feature. Basic reactor power response to heat demand transients is central to this investigation. Thermal transient behavior will provide an evaluation of proximity to thermo-physical fluid limitations.

Fuel Salt flow is convective driven by density differences between the reactor hot and cold thermal centers. Consequently, reactor flow will be dependent on fuel salt density variations due to heat exchanger demand and system energy balance during operations. A down-power demand on the heat exchanger, for instance, will initially drive heat exchanger outlet temperature higher and gradually reduce the overall cold leg density. This gradually reduces the overall system differential pressure and consequently also reduces system and core flow until colder fuel temperatures arriving at the core result in a positive reactivity addition due the negative temperature coefficient of the core (-3.5 pcm/K). The initial reduction in core flow causes a positive reactivity effect due to fewer delayed neutron precursors exiting the core and producing fractionally more delayed neutrons inside the core (Pazsit, 2017). These neutrons would otherwise be produced outside the core and are assumed to not contribute to the critical neutron economy. This flow reactivity effect is unique to MSR dynamic performance and must be considered in time-dependent power response of the system and safety basis analysis. Therefore, a down-power in system energy demand causes an initial reactor power increase. Conversely, if the heat exchanger undergoes a up-power demand, then the initial reactor power response is downward. This unstable, contravariant nature of power response in a natural circulation MSR is of great interest and must be shown to be controllable or limited under all credible operating conditions. Moreover, slow natural circulation transport times prolong reactor power response time and amplifies the power and temperature over and under shoot behaviors under dynamic powering conditions. The same is true for heat exchange down-power demands, that initially cause upward reactor power excursions due to increased system flow. Concerns for the potential for fuel salt solubility, fuel freezing, or fuel salt vaporization must be considered in the safety case basis. This time-dependent behavior will be further analyzed and modeling in this dissertation.

Once fuel salt flows through the core and is fission heated, the fuel flow vertically through the chimney where a fraction of neutrons that either leak from the core or are generated by the decay of delayed neutron precursors are reflected downward by beryllium reflectors that do not impede the convective flow. At the top of the chimney the flow diverges into a horizontal plane and enters the primary heat exchanger inlet. Once in the heat exchanger, the flow continues radially outward and then downward where the flow stream is cooled, and fuel salt density increases to provide the net thermal driving head to continue the closed loop flow cycle.

An investigation of the power transient behavior of the MsNB with an in-core WHIP was undertaken in this dissertation. Power reactor traditionally have a power following behavior between the heat exchanger and the core. As power demand on the system changes, reactor power follows to meet heat exchanger demand. In a traditional solid fueled reactor system with negative temperature coefficients this following behavior is automatic and inherent to core physics behavior. The MSR is fundamentally different in power response due to flow reactivity effects. In a natural or convective flow MSR system such as the MsNB flow reactivity effects can dominate the reactor's response to power transients. For instance, take a down power demand on the heat exchanger. A down power on the heat exchanger results in an increase in heat exchanger outlet temperature that propagates down the downcomer and eventually enters the core. In the MsNB this flow transfer takes about 30-60 seconds. Since the downcomer temperatures decrease, the corresponding downcomer densities increase. This increases the differential pressure between the hot core leg and the colder downcomer region. The increases differential pressure results in increase flow in the system. The resulting increase in system flow produces a negative reactivity in the core and a decrease in reactor power. So, the reactor power follow behavior manifest in traditional reactor system is not true in MSR convective system. This downward power trend will continue until the colder fuel salt enters the core in sufficient rate and volume to turn reactor power upward due to the negative temperature coefficient. Once the temperature coefficient turns reactor power the downward power excursion may be significant due to the slow loop transport times. Eventually reactor power will recover and overshoot to help restore the system heat balance. This overshoot results in continuing flow and power oscillations that may or may not be stable. Loop transport time, flow and temperature coefficients play a significant role in system stability.

Flow reactivity can be viewed as an effect caused by global (loop-wide) thermal properties which affect the flow velocity. Temperature reactivity on the other hand is a local effect based only on the temperatures in the core. The interplay of these two core physics effects determine reactor power response to thermal-hydraulic dynamics in the MSR system.

5.2 Fuel Salt Flow Phenomenology

5.2.1 Thermal Hydraulics in a Natural Circulation MSR System

Buoyant convective flow in a closed loop system is driven by fluid average density differences between hot and cold legs of the system. Density differences are established when the cold thermal center (region of lowest average system temperature) is placed above the hot thermal center (region of highest average system temperature). In the MSnB design the reactor core is placed low in the loop system just above the core inlet plenum region and the primary heat exchanger is placed high in the loop at the reactor top outlet plenum region, see Fig. 5.1. Due to the offset placement of the heat exchanger the hot and cold legs are asymmetric in their flow cord lengths. The effective vertical height difference between core center and heat exchanger center in the MsNB design is about 1.09 meters. An average density difference between hot and cold leg fuel salt is about 10-20 kg/m^3 which establishes a differential pressure in the system of about 200-2500 Pascal. This results in a flow rate of about 2.0-5.0 cm/s in the system assuming the pressure loss coefficient, ξ, on the order of 25.

STAR-CCM+ CFD code was used to model the thermal-hydraulic performance of the closed-loop MsNB natural circulation system using the core physics outlined in this chapter. STAR-CCM+ provided a high fidelity understanding of the temperature, flow, and pressure fields in the system. These results were then used to ”fine tune” the 1D Python program results. For instance, a pressure loss coefficient of 10-15 was found to be appropriate for the 6-pitch WHIP configuration operating at 400 KW. Using the CFD code results to modify the 1D core and heat exchanger fluid mixing behaviors for the system configuration being modeled ensured similar results between the two codes. Using a high fidelity CFD tool such as STAR-CCM+ to calculate the thermo-physical system performance helped improve the prediction capability of the computationally inexpensive 1D Python code (Carter, Richardson, Yoon, 2022).

Generally the hydraulic relationships that apply in the convective flow case are,

$$P_{\text{out}} \;=\; P_{\text{in}} + \rho\left(\frac{v_{\text{in}}^2 - v_{\text{out}}^2}{2}\right) + \rho g\left(z_{\text{in}} - z_{\text{out}}\right) - \xi\frac{\rho\bar{v}^2}{2} \tag{5.1}$$

and

$$h_{\text{out}} \;=\; h_{\text{in}} + \left(\frac{v_{\text{in}}^2 - v_{\text{out}}^2}{2}\right) + g\left(z_{\text{in}} - z_{\text{out}}\right). \tag{5.2}$$

For this model, fluid height or velocity are assumed to have negligible change in fluid enthalpy.

The differential pressure of the closed-loop, buoyant convective flow system can be written in terms of the average hot and cold leg density,

$$\Delta P(t) \;=\; \left(\bar{\rho}_{\text{cold}}(t) - \bar{\rho}_{\text{hot}}(t)\right) h_c g \tag{5.3}$$

where ΔP is the system differential pressure, ρ is the average hot or cold leg density, h_c is the difference in system thermal centers, and g is the gravitational constant. This differential pressure establishes flow in the loop at a magnitude sufficient to overcome the resistive drag force in the system. Friction losses can result from forces on flow surfaces and the flow system geometry that result in changes in flow direction or can present hydraulic resistance to flow inertia, i.e. heat exchanger, core internals, elbows, and flow direction devices (IAEA, 2005)(Manohar, 2013). System resistance can be written as,

$$\Delta P(t) = \left(f\frac{L}{D_H} + \sum K\right)\frac{\bar{\rho}v^2(t)}{2} \tag{5.4}$$

$$= \xi\,\frac{\bar{\rho}v^2(t)}{2}. \tag{5.5}$$

The first term is the straight pipe resistance where f is the Darcy friction factor, L is the pipe length, and D_H is the hydraulic diameter of the flow volume. The second term are local form losses, K, due to flow restrictions such as elbows, circuitous heat exchanger flow path, and changes in inlet and outlet plenum flow direction. The two terms combined are the system pressure loss coefficient (PLC), ξ. An estimate of the PLC was used in this investigation based on standard pipe fitting coefficients (Janna, 1998) and previous analysis of the heat exchanger PLC of 7.0 (Cardenas-Melgar, 2021).

The system velocity can then be determined by,

$$v(t) = \sqrt{\frac{2\,\Delta P(t)}{\xi\,\bar{\rho}}} \tag{5.6}$$

where $\bar{\rho}$ is the system average fuel salt density. System flow varies as the inverse square root of the system pressure loss coefficient. Therefore, as pressure losses increase system flow decreases in an inverse square root proportion. Since this parameter affects system flow, it also may affect power over and under shoots during power demand transients. This will be examined in the results section.

A constant heat exchanger wall temperature and uniform volumetric internal heating rate is assumed. As a result, temperature variations and heat transfer within the reactor core and the primary heat exchanger are assumed to change linearly over fluid travel distance in the heat source or sink. Reactor core and heat exchanger differential temperatures are approximated using heat transfer equations,

$$\dot{Q}(t) \;=\; \dot{m}(t)\, c_p(\bar{T}(t))\ [T_{\mathrm{out}}(t) - T_{\mathrm{in}}(t)]\,. \tag{5.7}$$

Eqn. 5.7 is used in the model to determine the core and heat exchange outlet temperature after time-step values for reactor power, heat exchanger power, system mass flow rate, and specific heat capacity are calculated. $\bar{T}$ is the heat source or sink average temperature. All the quantities in Eqn. 5.7 are functions of time.

Fuel salt density is a function of composition and temperature. Salt density can be determined using the weight percent relationship,

$$\frac{1}{\bar{\rho}} \;=\; \sum_i \frac{w_i}{\rho_i} \tag{5.8}$$

where $\bar{\rho}$ is the mixed fuel salt density, w_i is the weight percent of constituent i, and ρ_i is the density relationship of constituent i. Using this relationship for 15 mol% UF_4 in a FLiNaK eutectic composition mix (46.5-11.5-42 mol%) gives a temperature dependent density relationship of,

$$\rho_{\mathrm{density}}\ [\mathrm{g/cm^3}] \;=\; -9.4601046\,\mathrm{x}\,10^{-4}\ \mathrm{T[K]} + 4.6820365 \tag{5.9}$$

where density is in $\mathrm{g/m^3}$ and fuel salt temperature, T, is in Kelvin. The specific heat capacity of FLiNaK fuel salt is,

$$c_p\ [\mathrm{J/kg\cdot K}] \;=\; 1.0634\ \mathrm{T[K]} + 976.78 \tag{5.10}$$

where c_p is in J/kg-K. Mass flow rates are easily calculated using the $\dot{m} = \rho A v$, where A is the MsNB cross sectional flow area ($\approx 0.4\mathrm{m}^2$) (Manohar,2013)(Guifeng, 2019)(Lizin, 2013).

Unless otherwise stated, this investigation was completed using the parameters provided in Table 1. Table 1 neutronic values were obtaining using Serpent 2.1.31 particle transport code on the micro-volume molten salt nuclear battery design. Flow reactivity coefficients were derived using prior numerical work by Dulla (Dulla, 2005).

5.2.2 Density Wave Oscillations

The direction and stability of flow is determined by the temperature difference between varying locations in the loop where buoyant forces are relevant. These propagating temperature variations result in fluid density variations that can induce flow and power oscillations. Temperature oscillations driven by changes in core reactivity result in fluid density and pressure changes in the system. For this reason, flow velocity can oscillate or reverse direction depending on external factors that influence system energy. The goal of most past studies was to evaluate for the predictive operating conditions where instabilities can be expected (Luzzi, 2017). Avoiding or implementing mitigating controls near unstable or chaotic flow regimes is desirable from a design safety standpoint.

While significant work has been completed to examine the mechanisms of single phase and two-phase flow in light water reactor system, virtually no work has been conducted to investigate the flow stability of a closed-loop, natural circulation molten salt reactor system. The engineering study of such a system is currently under investigation with the MsNB at the University of Idaho.

Thermal hydraulic instabilities that result from closed-loop, natural circulation flow are generally classified into two categories: static and dynamic. The MSR has a unique instability mechanism caused by reactivity effects coupled to delayed neutron precursor advection. Namely, molten salt velocity in an MSR system influences the fraction of transported precursors from the core and therefore affect core reactivity. As a result, reactor power and local system temperatures respond to flow velocity changes. The contribution of precursor advection to the system's thermal hydraulic feedback is unique and not well studied experimentally. Furthermore, traditional density wave oscillations are caused by time delayed feedback effects between flow, temperature, and pressure fields in the closed loop. Temperature oscillations in the closed loop cause in-phase time variation of fluid density. Density wave oscillations are well understood in closed-loop, natural circulation systems (Zou, 2017)(Saha, 2018)(Vijayan, 2010). Density wave oscillations coupled with power changes induced by flow reactivity effects result in compound dynamic interactions that have both temporal and spatial delayed feedback contributions to reactor power and have been studied less. Combined, both interact in a time-dependent manner to generate a combined dynamic instability unique to molten salt fueled systems.

5.2.3 Compound Dynamic Effects Unique to MSR - Flow Reactivity Coefficient

Compound dynamic effects (CDE) generate a unique class of time-dependent power behaviors that occur when in addition to fundamental effects such as density variations, secondary effects are involved that couple to and significantly modify the fundamental system response. A typical case of a fundamental effect is density wave oscillations caused by the delay and feedback between fluid flow, density, and pressure in a closed-loop system. The MSR has a characteristic CDE that is unique to flowing fuel systems that results from two reactivity affecting changes, namely, average core temperature (fundamental) and delayed neutron precursor advection (secondary). Precursor advection results in delayed neutrons being generated in locations other than where the precursor is produced. This non-local contribution to the neutron economy introduces flow dependent reactivity effects, i.e. *flow reactivity*. This specific CDE as it relates to the MSR will be referred to as the *compound reactivity effect*. Compound reactivity effects are similar to the interaction between void reactivity and flow dynamics and heat transfer in a boiling water reactor (BWR). In a MSR, flow reactivity couples to the thermal-hydraulics of the system.

The compound reactivity effect has two opposing reactivity mechanisms. The first mechanism is related to the temperature coefficient of reactivity that also acts in traditional solid fueled reactor systems. As core average temperature changes, the core reactivity changes due to Doppler absorption and material density changes. For the MsNB, the temperature coefficient was determined to be about -3.5 pcm/K across the 600-700 degrees Celsius operating temperature range of the reactor (Carter, 2021). The temperature reactivity in the core can be calculated by,

$$\Delta\rho_T(T) \quad = \quad \alpha_T(T)\,\Delta T \tag{5.11}$$

where α_T is the temperature coefficient of reactivity and ΔT is the change in average core temperature from a defined steady state initial temperature value. The temperature coefficient is assumed constant over the range being considered (800-1000 K).

The second reactivity mechanism is unique to flowing fuel systems. As fission products advect out of the core, delayed neutron precursors are transported from point of generation to the point of decay. The point of decay may be out of the active core region (ex-core) which results in the potential loss of neutron inventory to sustain criticality. In this analysis it is assumed that delayed neutrons that are generated outside the core are lost to the active core. In actuality, consideration of a spatial weighted adjoint flux across the fuel salt regions of the reactor system would be appropriate to evaluate for the ex-core neutron contributions to the active core. Using a one-group delayed neutron precursor (DNP) model, on average the number of DNPs that decay outside of the core is a function of time spent out of the core versus time spent in the core. The relative times the DNPs spend in the transport cycle are determined by loop geometry and fuel salt transport velocity. In the one dimensional case, near critical conditions, the core neutron multiplication factor can be expressed as,

$$K_{eff} \;\simeq\; 1 - f(u_{\|})\beta_{eff}, \tag{5.12}$$

and hence,

$$\rho_f(u_{\|}) \;=\; -\frac{f(u_{\|})\beta_{eff}}{1 - f(u_{\|})\beta_{eff}}, \tag{5.13}$$

where $f(u_{\|})$ is a normal to core exit plane velocity dependent function for the fraction of precursors outside of the core and β_{eff} is the static delayed neutron fraction. The normal component of fluid velocity, $u_{\|}$, is the calculated by the sine of the helix angle, α. The normal outward precursor mass flux determines the net loss of precursors from the core per unit area per unit time. The function $f(u_{\|})$ can be thought of as one minus the ratio of the β_{eff} for the circulating system to the β_{eff} for the static case as shown in Eqn. 5.18. The departure from criticality is proportional to the fraction of precursors that advect out of the active core region. In a static fuel condition, all precursors remain in the core and the reactor remains critical. As fuel salt velocity increases, more precursors decay outside of the core and do not contribute neutrons to criticality. As flow velocity increases to large values such that loop recirculation time is much shorter than precursor group decay half-life, an asymptotic neutron loss is reached as precursors that leave the reactor return before decay occurs. In the limit of large flow velocities, the fraction of delayed neutrons produced outside of the core is the time spent in the loop outside the core over the total loop recirculation time. For simplicity it is assumed that flow velocity is the same throughout the system loop at any location and moment in time (ρA = constant; where A is the cross sectional flow area), therefore the fraction of delayed neutrons produced outside of the core is directly related to the loop and core flow cord lengths. Specifically, in the limiting case for large fuel salt velocities, K_{eff} is given by,

$$K_{eff} \simeq 1 - \left(\frac{L}{L+H}\right)\beta_{eff}, \tag{5.14}$$

where L is the ex-core loop length and H is the in-core flow cord length. As the ex-core loop length becomes very large such that recirculation time is large compared to precursor half-lives then the multiplication factor approaches 1-β_{eff}. The expression for the large velocity limit for core reactivity is then given by,

$$\rho_{\infty} = -\frac{\beta_{eff}}{(1-\beta_{eff})+\frac{H}{L}}, \tag{5.15}$$

where for large L the reactivity penalty at high flow rates is a function of the delayed neutron fraction only. Dulla's previously published work shows that the functional form for reactivity changes in a MSR core due to velocity changes (initial stationary fuel flow) can be expressed as,

$$\Delta\rho_f(u_\parallel)\,|_{0\to u_\parallel} \;\simeq\; \rho_\infty\left(1 \,-\, e^{-\alpha_f u_\parallel}\right), \tag{5.16}$$

where α_f is referred to as the flow reactivity exponent and is determined by linear regression of Dulla numerical results (Dulla, 2005) and is provided in Table 5.1. It is assumed that the value of α_f is the same between the Dulla's results and the MsNB. The two thermal systems have similar β_{eff} and system loop length ratios and at velocities above 4 cm/sec the flow reactivity approaches a constant contribution to the total core reactivity. While flow reactivity effects dominate core power response early in a heat exchanger power demand transient (system flow changes before core temperatures change) until core temperatures change a short time after the transient starts. The flow reactivity exponent is a function of core physics. The change in flow reactivity between an initial velocity, u_i and final velocity, u_f can be written,

$$\Delta\rho_f\,|_{u_i\to u_f} \;\simeq\; \rho_\infty\left(e^{-\alpha_f u_i} \,-\, e^{-\alpha_f u_f}\right). \tag{5.17}$$

Analytically, the fraction of precursors that advect outside the active core volume can be expressed as a normalized ratio of precursor concentrations integrated over the core volume as a function of fuel salt velocity. Specifically,

$$f(u_\parallel) \;=\; 1 \,-\, \left(\frac{\int C_o(u_\parallel,\vec{\mathbf{r}})\,dV}{N}\right), \tag{5.18}$$

where C_o is the static (time-independent) precursor concentration distribution function (normalized to core integrated flux). The second term is the circulation β_{eff} to static β_{eff} ratio. For a 1-D model, this relationship reduces to a definite integral over the axial dimension, $-a$ to a. N represents the precursor number at zero flow or,

$$N \;=\; \int C_o(0,\vec{\mathbf{r}})\,dV. \tag{5.19}$$

The static flux and precursor equations can be numerically solved to provide C_o. Although it can be shown that the approach used in Eqn. 5.18 and 5.19 provides a statistical coefficient of determination $>$ 0.99 R^2 fit to analytical, deterministic, and Monte Carlo agreements in previous published work by M. Aufiero, this paper will use a regression fit of Dulla's results (fit with $>$ 0.98 R^2) and shown by Eqn. 5.16 (Aufiero, 2014; Dulla, 2005). This provides simplification without significant loss in technical rigor in the results. For the parameters provided in Table 5.1 and used in this analysis, ρ_∞ = 499 pcm. Notice that as velocity becomes large, the change in flow reactivity becomes small for incremental changes in velocity. This is because at larger velocities (velocities that allow the fuel salt to traverse the loop within the one-group decay constant) the precursors are effectively distributed through the loop and the rate of precursor loss from the core is offset by the precursors returning to the core and no net additional neutron loss is noted. Both reactivity mechanisms affect core power response in a coupled, non-linear manner. Fig.5.2 diagrammatically illustrates this compound reactivity effect.

In the WHIP MsNB case, the fraction of delayed neutron precursors present outside of the core as a function of normal fluid flow velocity, $f(u_\|)$, can be numerically determined for each WHIP pitch configuration using the velocity-dependent, static, delayed neutron precursor concentration, C_o, using Eqn. 5.18. For each velocity, a fraction of precursors outside the core can be determined. The linear regression fit of results are used to calculate a function form of $f(u_\|)$ to a coefficient of determination $R^2 >$ 0.98. The analytical form of $f(u_\|)$ is,

$$f(u_\|) \;\simeq\; f_0\left(1 \,-\, e^{-f_1 u_\|}\right), \tag{5.20}$$

where the leading coefficient, f_0, and the exponent, f_1, are determined using high confidence ($R^2 > 0.98$) linear regression methods. The leading coefficient f_0 represents the fraction of delayed neutron precursors outside the core as the fluid velocity approaches large values. Since the effective circulation length of the closed loop is biased to in-core for increasing pitch numbers this leading coefficient increases with pitch number, as expected. The exponent factor, f_1, represents how quickly the asymptotic value of the fraction is approached as fluid velocity increases and is related to the slope of the static precursor concentration, C_o, near the core outlet. Table 5.4 provides a tabulation of Eqn. 5.20 coefficients for the 1-pitch through 6-pitch WHIP configurations. As the helix angle decreases or pitch number increases, the leading coefficient f_0 and exponent f_1 both decrease. This derivation for $f(u_{||})$ is used in the Python and STAR-CCM+ modeling code as a key kinematic relationship characterizing flow reactivity.

As can be seen, the flow reactivity coefficient is strongly dependent system geometry and fuel salt flow velocity. This opens a rich area for engineering inputs that can affect core performance outcomes.

5.3 Modeling Approach

The sum of flow and temperature reactivity affects core power. Each Euler method time-step performs a calculation to determine the reactor period, τ. This can be written as,

$$P[t+1] \quad = \quad P[t]\, e^{[t]/\tau} \tag{5.21}$$

the square brackets denote numerical time-step iterations. One-group kinematics provides a formulation for calculating the flow dependent reactor period, namely

$$\tau(u,t) \;=\; \frac{l^*}{\rho(t)} + \frac{(1-f(u_{\|}))\beta_{eff}\,-\,\rho(t)}{\lambda\rho(t)+\dot{\rho}(t)}, \tag{5.22}$$

where l^* is the prompt neutron generation time, ρ is the total core reactivity (flow and temperature reactivity summed), λ is the delayed neutron precursor one-group decay constant, $(1-f(u))$ is the fraction of precursors in the core (1.0 when u=0 cm/s) and $\dot{\rho}$ is the time rate of change of reactivity (Lamarsh, 2001; Duderstadt, 1976). It should be noted that the reactor period kinetics expression in the liquid fuel case is a function of fuel salt velocity. This makes sense because the velocity dependence of the effective delayed neutron precursor fraction. Precursors advecting out of the core have the effect of reducing the circulating β_{eff} and diminishes the contribution the second term has on reactor period. The time dependence of each factor is made explicit in the equation to emphasize implications on the time-step numerical solution method. The first term provides the prompt neutron contribution to the time rate of change of reactor power while the second term provides the delayed neutron contribution. The time rate of change of reactivity in the second term plays a significant role in the compound dynamic effects in the MSR.

Python 3.7 was used on a linux, 12-core, Lenovo ideapad 330S to develop a time dependent finite-element code to model the thermal hydraulic and neutronic coupled flowing loop. This section outlines the approach taken to solve the time iterated system frames. Fig.5.3 outlines the model coding methodology. Appendix B provides the Python code used in completing this investigation. Constants in this model included the temperature and flow coefficients of reactivity, the prompt neutron generation time, the effective delayed neutron fraction, the one-group delayed neutron decay constant, the system pressure loss coefficient (PLC), constants associated with fuel salt density computations, thermal center heights, and core dimensions. Global (time-dependent) and local (time and space-dependent) array variables and functions were used in this application. Global variables and functions are reactor power, heat exchanger power demand, differential pressure between hot and cold legs, system fuel salt velocity, flow and temperature reactivity in the system, reactor period, average specific heat capacity of the hot and cold legs, average reactor and heat exchanger temperature, time rate of change of reactivity, and differential temperature across the reactor and heat exchanger. Local variables included temperature and density. The effect DNPs have on core reactivity is modeled by the influence of system mass flow rate (fuel salt velocity). The time rate of change of reactivity in the system was determined by standard finite difference approximation.

Differential temperatures across the reactor and heat exchanger and outlet temperatures were calculated at each time step using Eqn. 5.7 based on time-step values of inlet temperature, system mass flow rate, and hot/cold leg specific heat capacity. MsNB reactor circulation loop dimensions are provided in Table 5.2 and various pitch WHIP physical form parameters are provided in Table 5.3.

5.3.1 Discretized Spatial Propagators

At each time-step, global variables are calculated based on the results of a nested spatial computation loop that updates local variables. Specifically, local variables such as temperature and density are space-advanced (incremented) based on system mass flow rate, $u(t)dt$. As shown in Fig. 5.4, advancing by $u(t)dt$ each time step mimics the convective transport of the fuel salt in the system loop. For example, reactor outlet temperature cell is advanced toward the heat exchanger inlet by a distance the volume would travel in a time-step interval.

After the spatial local variables are spatially advanced, the code calculates updated average reactor and heat exchanger inlet temperature values. Inlet temperatures are calculated using an averaging routine that accounts for heat exchanger fluid mixing at the inlet. The code then calculates updated average vertical hot and cold leg densities. The densities are assumed to change linearly in the reactor and heat exchanger. Spatially advancing reactor and heat exchanger exit temperatures and densities ensures fluid propagation at rate commensurate with fluid velocity to the down stream heat device.

The code then updates all global variables for time step [t] and then calculates updated reactor and heat exchanger outlet temperatures using time history heat balance,

$$T_{\mathrm{Rx,Outlet}}[t] = T_{\mathrm{Rx,Inlet}}[t-1] + \Delta T_{\mathrm{Rx}}[t-1] \tag{5.23}$$

and,

$$T_{\mathrm{Hx,Outlet}}[t] = T_{\mathrm{Hx,Inlet}}[t-1] - \Delta T_{\mathrm{Hx}}[t-1]. \tag{5.24}$$

Once all global and local array variables are computed the time-step is incremented and the computation is repeated until the model simulation is complete. Generally, at 400 KW thermal output the steady-state system parameters are loop differential pressure is 230 Pa, flow velocity is 5 cm/s, and reactor differential temperature is 32 degrees Celsius.

5.3.2 Boundary Conditions

Time delayed boundary conditions are appropriately applied to ensure energy conservation and flow is recursive in the closed loop. Two temporal boundary conditions are important is this model. First, fluid arriving at the reactor inlet originated at an earlier time, depending on flow rate leaving the heat exchanger outlet. Similarly, fluid arriving at the heat exchanger inlet originates at an earlier time leaving the reactor outlet region. For the purposes of simplification, flow velocity is assumed to be constant throughout the closed-loop circulation path. Changes in flow velocity due to changes in flow area, for instance, can be scaled into the model if needed. One spatial boundary condition is imposed to ensure that loop is closed. Mathematically these boundary conditions are written in the local temperature arrays as,

$$T[RxInlet, t] \;=\; T[HxOutlet, t - \frac{l_d}{u}], \tag{5.25}$$

$$T[HxInlet, t] \;=\; T[RxOutlet, t - \frac{l_c}{u}], \tag{5.26}$$

and

$$T[RxInlet0,\ t] \;=\; T[RxOutlet1,\ t]. \tag{5.27}$$

where l_d and l_c are the downcomer and chimney flow length, respectively.

5.3.3 Assumptions

Simplifying assumptions are introduced without significant lose of physical rigor. First, the model is one-dimensional and assumes adiabatic conditions in piping between the heat exchangers. Finite losses to the environment would likely occur due to radiation and convection at the reactor vessel and cold leg which would promote natural circulation flow. Adiabatic conditions to the environment are a bounding buoyant flow condition. Delayed neutrons generated outside of the active core are assumed to not play a role in the core kinematics. The model also conservatively assumes no internal conduction between reactor structures. Fission heating is assumed to be uniform in the core and results in a linear temperature rise in the fuel salt through the core. Actual fuel salt temperature is a function of axial height in the core due to varying distributions of thermal flux and fission cross sections. Volumetric heat generation in the system due to decay heating is assumed to be negligible. The core outlet temperature calculation is assumed to be a simple function of inlet temperature, average core fuel salt heat capacity, mass flow rate, and reactor power. In addition, average core temperature is weighted to 60% toward outlet to accommodate axial flux skewing to upper core due to precursor convection. The core is assumed to have constant temperature reactivity coefficient across the range of investigated temperatures (values provided in Table 5.1). This was shown to be a valid assumption by the author, see chapter three (Carter, et al., 2020). The core and heat exchanger inlet temperatures are assumed to be a 20 centimeter spatial average of the fluid temperature. This accounts for an expected amount of inlet mixing that would be present in the system. Core reactivity is assumed to be low (¡ 150 pcm) so that the limiting reactor period (10 seconds) and kinematic time-scales are an order of magnitude larger than the Euler time step calculation. Insertion of a higher reactivity would result in reactor power change on a time-scale that make the modeling time-step of one second inaccurate. If higher fidelity results are needed, then the time-steps must be reduced appropriately.

Changes in spatial and temporal values over successive finite time steps are small compared to rates of convective and reactor period changes and for this reason errors inserted due to Euler methodology are assumed small. The 571 cm loop is divided into equally spaced one millimeter spatial elements while temporal computations are divided into one second intervals. The selected element sizes are reasonable based on average thermal and temporal gradients calculated in the model. Additionally, axial fluid heat conduction is assumed negligible and fluid properties with exception of density and heat capacity are assumed to be constant. Fluid axial heat conduction must be considered if the Peclet number is on the order of one or less. The Peclet number, Pe, for the hot and cold legs is provided by,

$$Pe = \frac{\rho_0 c_p u L}{k}, \tag{5.28}$$

where ρ_0 is the reference fluid density, c_p is the specific heat capacity, u is the fluid speed, L is the fluid segment characteristic length, and k is the thermal conductivity. The Peclet number is a dimensionless representation of convection-to-conduction energy transport in the fluid. For the MsNB using FLiNaK the Peclet number is in excess of 25,000 and therefore heat transfer by conduction is negligible. It should be noted than at low power heating conditions conduction may have a much larger contribution to system heat transport and may need to be considered. Low power flow modes are beyond the scope of this investigation and left for future work. Finally, this model assumes constant pipe size, uniform heat flux in the heat and cooling section, and combines friction pressure losses due to varying pipe sizes into a single pressure loss coefficient K as defined in Eqn. 5.4.

The form of the flow reactivity is provided by Eqn. 5.20 and is empirically obtain by the author developed static DNP flux solver (Python programming) for various loop geometry dependent configurations. It is assumed that this form holds in the physical case. The value of flow reactivity coefficients, f_0 and f_1, used in this dissertation are included in Table 5.4. In summary, these assumptions lay the ground work for further work and research.

5.4 WHIP Phenomenology - a comparative analysis of one up to six pitches in a WHIP

There are several aspects of the WHIP form factor that affects both the neutronics and hydraulic performance of the core. Specifically, the WHIP's helix angle determines the flow component parallel to the flow exit axis of the core (see Fig. 3.2) . A smaller helix angle results in a reduced flow component normal to the core exit plane. This parallel component determines the rate at which precursors exit the core. For this result, the helix angle affects the effective delayed neutron fraction during circulation, the reactor period, and the flow reactivity of the system for a given change in system flow rate. Secondly, the fuel salt flow channel is constrained within the channel depth and width of the WHIP. This flow area will likely be different than the flow area in the plena and downcomer. In fact, the WHIP flow area may be designed for a set core flow velocity which affects where on the flow reactivity curve the core operates under nominal temperature steady-state and transient conditions. Third, the WHIP effectively couples neutronic effects across adjacent pitches. Although separated in terms of stream line path distance, the adjacent pitch distance can be significantly smaller. For this reason, the traditional neutron flux distribution present of stationary fuel systems is very different in the MSR. Although the total integrated flux for a given core fission heat generation power is similar between cores, the flowing liquid fueled core with a WHIP will tend to have a flattened peak near core axial center and have higher average fluxes toward core inlets and outlet regions. For this reason, the MSR with a WHIP tends to have a greater flux coupling than even the traditional MSR design (see Figs. 3.7 and 3.8)

Fig. 5.5 shows the results of a parametric analysis of WHIP pitch (helix angle). Static flux and delayed neutron precursor distributions under varying fuel salt velocities were solved to determine the effective delayed neutron fraction and flow reactivity under varying circulation conditions. Six WHIP configurations (parameters provided in Table 5.3) were evaluated using this method and the results are presented in Fig. 5.5. Results are plotted against a dimensionless parameter, λT, and aids in comparative analysis between the different configurations. T is the total loop transport time and λ is the one-group delayed neutron precursor decay constant. Larger values of λT are representative of slower flow velocities. Nominal values of λT for the MsNB are on the order of 10 to 15. Fig. 5.5 shows that at large flow velocities, the effective delayed neutron fraction under circulation conditions is proportional to system geometry.

Namely, β_{eff} is proportional to the ratio of time precursors spend in the loop external to the core per transport cycle to the total loop transport time. Therefore, at high flow velocities the limiting value of β_{eff} circulation will increase with larger core resident times. Comparing a single pitch (1-pitch) WHIP configuration to a 6-pitch configuration increases β_{eff} under circulation conditions by over 70%. In addition, under nominal flow conditions β_{eff} under circulation condition only varies by 5% over the range of flow rates and WHIP configurations. The effects are more sever at higher flow velocities and would, for instance, be more dominant in a force circulation application where λT is on the order of unity.

Moreover, as precursors spend proportionally more time in the core per transport cycle, the maximum flow reactivity effect is reduced. This is due to more delayed neutrons participating in the fissions in the active regions of the core. Fig. 5.6 shows flow reactivity as a function of core flow exit velocity. At flow rates expected in the MsNB during nominal operating conditions, flow reactivity changes by about 50% for a given WHIP configuration (-140 pcm to -200 pcm for the single pitch configuration). Changing WHIP pitch configurations can reduce the effects of flow reactivity by a factor of 2.8 (-170 pcm for single pitch to -70 pcm for a 6-pitch configuration). This result highlights the potential benefit of the WHIP to improve reactor control during start-up, shutdown, and transient power operations.

5.4.1 Helix Angle and $\dot{m}_{||}$

The helix angle for the WHIP of length 179 cm with a single pitch revolution is 60.7 degrees. The helix angle decreases to a value of 16.6 degrees for a six pitch revolution WHIP. This reduction increases the streamline flow length at a radius of 15 cm (half split between WHIP core radius of 5 cm and WHIP outer radius of 25 cm) from 202.5 cm for the single-pitch configuration to 593.5 cm for the six-pitch configuration. That is, increasing from one to six pitch revolutions increases the flow streamline length by about 2.9 times. In addition, the single-pitch

WHIP has a normal component of the mass outlet flux, $\dot{m}_{||}$, that is 87.2% of the channel mass flux and for the six-pitch configuration the normal component of the mass flux reduces to 28.6% of the channel mass flux. This reduction in fuel salt mass flux exiting normal to the core outlet surface results in a proportional decrease in precursor flux from the core. Therefore, reduction in helix angle also reduced the effects of flow reactivity and mitigates reduction in the circulation effective delayed neutron fraction. This effect can be seen in results shown in section 6.5.3.

5.4.2 In-core Flow Area and Fuel Salt Velocity, u

It should also be noted that the channel width decreases as the number of pitch revolutions increases in the WHIP (for a constant mass device). Specifically, the channel width decreases from 118 cm for a single-pitch configuration to 26 cm for a six-pitch configuration. This is a reduction by a factor of 4.5 and by mass conservation implies that the six-pitch configuration will have a channel linear flow rate about 4.5 times higher than the one-pitch configuration. Due to the reduced helix angle, this higher flow rate does not translate to higher precursor losses at the core outlet, however. In short, precursor losses at the core outlet are a function of the mass flux normal to the core exit surface but the fuel salt velocity.

5.4.3 Adjacent Pitch Neutron Coupling

As described in 6.4.1 above, increasing the number of pitch revolutions to the WHIP configuration increases the stream line flow length substantially. Due to the helical flow path, however, neutronically the fuel salt in any given pitch is closer the fuel salt in adjacent pitches. This effect greatly enhances the axial neutron transport coupling in the core. For a given fission power generation, the integrated core flux can be considered constant. Therefore, enhanced neutron transport coupling results in a flatter flux distribution near core center and higher flux gradients near core boundaries to adjacent structures. In the traditional solid fueled reactor systems, the axial flux distribution is nominally has a sinusoidal dependence. While is can not be the case in MSR systems due to the advection of precursors, the addition of the WHIP device further flattens the flux and increase gradients near core boundaries. The physical effect of this flattening is that the circulation delayed neutron fraction and flow reactivity has a strong velocity dependence at lower velocities than without the WHIP.

To best understand how the WHIP improves neutron coupling, the geodesic distance between two point on a helical path can be derived (appendix D). Fig. 5.7 shows the coupling comparison between the various core geometries using parameters for a 6-pitch MsNB WHIP. Neutronic coupling in the core is proportional to the proximity between local reactive regions. The orange curve in Fig. 5.7 represents the straight-line vertical distance from a core axial midpoint. The gray curve represents the distance from a core midpoint following the fuel salt helical streamline (much longer path length due to the low helix angle in the 6-pitch configuration). The blue curve represents the geodesic distance (or neutron straight-line transport path length). Because neutrons are not constrained the fluid path, neutronic coupling in the WHIP core closely resembles that of the small, highly coupled traditional vertical flow core while the helical streamline (gray curve) constrains the delayed neutron precursor transport in core. For this reason, the WHIP maintains a high degree of neutron coupling while hydraulically transporting delayed neutron precursors along a path that increases the likelihood of in or near core decay and thereby increasing the effective delayed neutron fraction under circulation conditions.

5.5 Results and Summary

5.5.1 Power Demand Transients in a non-WHIPed MSR

The total flow loop length is 5.71 meters. For nominal evaluated flow rates, the fuel salt loop transport time is about 115 seconds or 1.9 minutes. This causes transient system response to take several loop transport cycle times to stabilize. Simulations show that quasi-steady state conditions following power demand changes can be achieved in about 40 minutes. Initial conditions are assigned in the code based on an analytical estimate of steady state conditions in the closed-loop system. A simplified natural circulation, closed-loop system is illustrated in Fig. 5.8. Any offsets to actual system steady state conditions introduce an initial run time transient. This transient takes the standard time to ”ring down” until numerical steady state system values are used by the code. This initial ”ring down” time can be minimized by adjusting initial conditions to match the numerically derived steady state values. This code tuning process allows the code to settle on steady state values in the shortest possible time. For this reason, it is always a good practice to run the code for a minimum of 25,000 runtime seconds to ”ring out” any transients that may be in the numerical calculations and ensure results more accurately reflect transient being investigated. The initial spatial temperature profile is shown in Fig. 5.9. The initial temperature rise is due to reactor heating which is assumed linear throughout the core. The system is assumed to be adiabatic and therefore temperatures do not change between system heat source and heat sink. At each time step, the reactor and heat exchanger outlet temperatures are spatially advanced by a distance proportional to the salt flow rate. In this way, outlet temperatures arrive at the downstream inlet at the proper time. Updated outlet temperatures are calculated at each time step based on time updated reactor power and heat exchanger power demand. Hot and cold leg densities that drive salt flow are calculated based on a spatial-averaged density value at each time step. In addition to density, time step propagated local temperatures are used to calculate updated specific heats, flow driving differential pressure, mass flow rate, and core reactivities.

Under certain conditions of low Reynolds number and appreciable internal volumetric heating, the system is unstable and may experience flow instabilities or un-damped flow oscillations or reversals due to density inversions in the hot and cold legs of the loop (Luzzi, 2017). Flow instabilities may be noted during specific transients that challenge the inherent system temperature and flow feedback. An example of unstable regimes include low flow, low temperature, and homogeneous volumetric heat generation within a natural circulation loop. Such conditions might be more likely during a shutdown (low flow) shortly following high power operations (high decay heat). Detailed evaluation of instabilities is left for future study and not addressed in this paper. Further investigation is recommended so that designs can be modified to accommodate, mitigate, and eliminate instabilities during power operations. Future work may support safety basis engineering analysis prior to fielding a design for demonstration for both natural and forced circulation flow modes.

5.5.2 STAR-CCM+ modeling of MsNB transient behavior - comparison to 1D Python result

A STAR-CCM+ java macro script developed in collaboration with SuJong Yoon of the Idaho National Laboratory using the core physics modeled in the Python program. The macro calaulated a time step value for flow and temperature reactivity using STAR-CCM+ field function calculated values for average volumetric core temperature and average core outlet surface averaged axial fluid velocity in the buoyant, closed-loop convective system. STAR CFD results were then used to ”fine tune” the 1D Python program thermal-hydraulics to more closely generate the high fidelity CFD results. This numerical approach produced meaningful simulated results without the expense computational overhead associated with STAR-CCM+. A typical 2400 second physical time simulation using STAR would take 40 cores and a wall time of approximately 100 hours to complete a transient simulation. The equivalently ”tuned” 1D Pytho code can run the same transient in roughly 10 minutes on a 10 core laptop while providing improved confidence in results. For this reason, this report will present Python code results that have been ”tuned” based on CFD informed simulations results using a java macro to calculate core physics.

Three specific modifications were made to the author developed Python program that simulates the MsNB power following, natural circulation power transient code. First the system pressure loss coefficient (PLC), ξ, was adjusted to replicate system flow velocities provided by STAR-CCM+ using smooth wall surface conditions. The PLC was calculated by STAR-CCM+ to be approximately 11.0 for the 6-pitch WHIP configuration operating at 400 KW. Secondly, the Python program was modified slightly to accommodate some degree of fluid mixing at the heat inlets. This aided in smoothing the thermal-hydraulic results that more closely represented the STAR-CCM+ thermal hydraulic calculations in pressure, flow, and temperature. Third, the Python code was modified to account for the full hydraulic flow length of the WHIP configured reactor. Specifically, the 6-pitch WHIP increases the hydraulic flow length in the core to approximately 593.5 cm. This increase in flow path results in longer transport times of the closed-loop system and are accommodated for by inserting an equivalent length in two equal, horizontal segments between the top of the chimney and the heat exchanger, and the bottom of the downcomer and the reactor inlet. This effectively increased the hydraulic flow path of the loop without affecting buoyancy forces and core physics calculations.

While modeling and simulation verification, validation and accreditation of the author’s Python code is difficult if not possible at this time due to the absence of real-world experimental data to compare simulation results, the author’s intent in this dissertation is to show the Python program provides reasonable simulation results commensurate with the accredited CFD code such as STAR-CCM+ under limited evaluated situations. Using this approach, the author considers the STAR-CCM+ simulation results a high fidelity thermal-hydraulic simulation that provides confidence in CFD computed results with the physics provided. Under these limited conditions, the author considers his Python program a viable option that provides reasonable results at significantly less computational expense.

A comparison of simulated results between the author's Python program and STAR-CCM+ for a 400 to 800 KW power demand transient response in a 6-pitch WHIP MsNB configuration are shown in Fig. 5.10. The Python program generally provides a faster power response with reduced power overshoots and faster power damping than the CFD code in STAR-CCM+. The physics code is the same in both codes and therefore differences can be attributed to the improved fidelity of the STAR-CCM thermal-hydraulic computations. While it is ideally desirable to use a coupled multi-physics simulations using STAR-CCM+ to accomplish detailed studies to MsNB reactor transient response, this approach can be computationally expensive as each 2400 sec real time simulation can take up to a week using 40 cores on a HPC system. By contrast, a Python code simulation on a work station can provide results in under an hour total run time.

To better understand system performance as a result of a significant over power demand from the heat exchanger, Fig. 5.11 illustrates the results of a STAR-CCM+ time-dependent transient power simulation for a 6-pitch WHIP MsNB configuration. Initial system power is 400 KW steady state. At time zero the system experiences a prompt power demand increase to 4 MW for a duration of 130 seconds which corresponds to the time needs to flow reactivity dominance to subside. At time 130 seconds, the system power demand decreases to 0 KW. The initial reactor power response decreases to about 370 KW due to flow reactivity effects. At approximately 20 seconds, colder fuel salt arrives in the reactor which contributes positive reactivity. Temperature effects dominate the core power response to the remainder of the transient. The mass flow rate more than doubles in a short period of 40 seconds and decays to about 1 kg/sec after a period of approximately 10 minutes concomitant with reactor shutdown. Severe power rates do result in significant system temperature transients which contribute to high thermal stresses particularly during cooldown transient where stresses are more limiting to the reactor vessel structures. As expected, the heat exchanger has the coldest system temperatures and severe power transient can challenge both fuel salt solubility (approximately 823 K for 18 mol% UF_4) and solidus conditions (freezing temperature of approximately 726 K) of the fuel in local regions. This is an important result that informed safety analysis and design basis accident evaluation. To prevent off-nominal system performance in such a case, for example, an active control systems may be appropriate to mitigate the severity of such transients.

These models assume adiabatic losses at system fluid boundaries and no decay generation upon loss of heat sink accidents. The adiabatic assumption may need be conservative from a safety perspective in some cases. While the adiabatic assumption provides a lower bounding temperature case for heat sink surfaces (particularly near low, inner surfaces), the heat losses to the environment combined with thermal conduction of heat across the downcomer control volume may or may not promote natural circulation depending on net heat flux conditions in the downcomer. A full thermal analysis is needed to determine net heat balance conditions in the system due to internal conduction and losses to the environment.

Fig. 5.12 illustrates a STAR-CCM+ simulation highlighting the effect of loss of heat sink followed by rapid restart. This transient illustrates the system performance during down power events followed by power demand restoration in a short period of time. This transient establishes system response to a short duration shutdown and subsequent restart. Initial system power is 400 KW steady state. At time zero the system experiences a prompt loss of heat sink for 1000 seconds (approximately 17 minutes). A duration of 1000 seconds was selected for this transient to allow the system to establish a quasi-equilibrium at low power levels prior to a significant over power demand. At time 1000 seconds, the system power demand increases rapidly to 1 MW. On loss of the heat sink, initial core power peaks at approximately 422 KW due to flow reactivity effects. Maximum reactor outlet temperatures approach about 940 K. Upon reactor rapid startup at 1000 sec, core power reaches a minimum at approximately 1200 sec at about 388 KW before rising rapidly to a peak core power of approximately 3.8 MW. System mass flow rate response autonomously to the heat sink power demand. As in the python model, core reactivity is dominated by the temperature coefficient. Thermal transients in the heat exchanger are severe and must be evaluated in detail to ensure yield strength limitation are not approached at temperatures above the reference transition temperatures of the material involved in the design (316 SS in this case).

5.5.3 MSR Compound Dynamic Effect

Compound dynamic feedback effects that are unique to a MSR system occur during time-dependent power transients. As shown in Fig. 5.11(a), at early times, core power does not follow power demand due to flow reactivity effects that dominate the system before local temperature changes arrive at the core inlet. Convection of relatively colder fuel salt temperatures arriving at the core inlet insert positive reactivity and turn core power to follow demand at roughly one minute into the transient. The figure shows the reactivity effects and the over taking effect of temperature. The strong negative temperature reactivity gradient after the initial flow component turns the positive reactor period is evident around 4 minutes into the transient.

A time-dependent, finite-element python code was developed to model the MSR power transient behavior in natural circulation flow. Opposing reactivity effects associated with flow velocity and temperature act on changing reactor power. Flow related neutronics effects cause reactor power to decrease initially after an up-power demand from the heat exchanger. Likewise, reactor power increases initially on down-power demands. This is unique to MSR systems and counter to traditional lead-follow heat balance systems. Although the model has no current benchmark, the python model developed by the author to study this transient behavior is supported by thermodynamic and neutronic principles and informed intuitive reasoning. STAR-

CCM+ simulations using an equivalent neutronic physics is the only comparative analysis tool available at this time. Results show that the natural circulation MSR system does exhibit critical damped characteristics but may also have instabilities that warrant further study.

While the Python program generates damped results compared to STAR-CCM+ with equivalent physics, the Python code provides an 80% solution in temporal and amplitude predictive behavior and will be used in this study to provide computationally inexpensive results that are strongly supported by physical principles to help build an intuitive learning foundation. Fig. 5.13 depicts the Fourier transform of the reactivity components in the frequency domain. Both temperature and flow components have maximum amplitudes at low frequencies. This is expected due to the loop transport time (115 sec). This implies a natural response frequency of about 0.01 Hz, well below the scale in Fig. 5.13. Across the spectrum of frequencies shown in Fig. 5.13, the flow component leads the temperature component by π radian. This is expected, since the flow component is driven by global density variations in the hot and cold loop segments whereas the temperature component is a local function of core temperature.

Figs. 5.14 through 5.16 are phase space diagrams generated by the author's Python program from up-power and down-power transients. The blue curves represent the transient from 250KW to 1000KW and the red curve represents power change from 1000KW to 250KW. A slightly reduced intial power level was slected for this analysis to show the reader the difference in initial conditions. Power-Velocity transitions tend to closely follow geodesic trajectories between initial and final state. Reactivity-Velocity trajectories tend to be less constrained because core reactivity changes do not linearly result in system flow changes. This can be seen in the flat (horizontal) trajectories near initial and final state points. Temperature-Flow reactivity trajectories both begin at the origin. Again, approach trajectories to final state tend to involve flat (horizontal) approach due to insensitivity of the flow component.

The flow reactivity coefficient is unique to MSRs and is strongly dependent system geometry and fuel salt flow velocity. Forced circulation MSR designs with forced circulation flow have largely constant (assuming constant speed pumps) negative flow reactivity coefficients that are largely independent of power changes. Natural circulation operating regimes or loss of force flow accident scenarios makes this technical analysis important to safety basis development for all MSRs. This opens a rich area for engineering inputs that can affect core performance outcomes.

5.5.4 Effects of the WHIP on MSR Power Transient Performance

A MSR core modified by an in-core WHIP device substantially changes the physics and thermal-hydraulic performance of the system. The fuel salt flow stream is significantly lengthened due to the circular helix flow path. Fuel flow velocity becomes a strong function of distance from WHIP root center and channel flow area and the outlet mass flow rate is modified by the WHIP helix angle. These geometric factors change the core's circulation effective delayed neutron fraction, flow reactivity coefficient, flux and delayed neutron precursor distribution, and temperature and pressure distributions. The WHIP can be design to help mitigate the lose of delayed neutrons from the core, improve neutron efficiency, and reduce fuel loading needs to achieve thermal performance requirements. Figs. 5.17 through 5.20 illustrate the results of power transient Python modeling code using a six pitch WHIP configuration. The figures show the comparison of the single pitch (1-pitch) WHIP in pastel colors against the 6-pitch WHIP in primary colors.

5.5.4.1 Power and Reactivity Response

Figs. 5.17 and 5.18 show the time-dependent power and reactivity response of the 1-pitch and 6-pitch WHIPs to an up-power and down-power demand (shown in blue) transient between

250 KW and 1000 KW. The average total system loop transport time in these transients is about 115 seconds or just under 2 minutes. The characteristic period of the power oscillations is about 4 loop transport times. Short period oscillations on the order of a single loop transport cycle can be seen on the flow reactivity response in Fig. 5.18. Comparison of Fig. 5.18.(a) and 5.18.(b) shows the up-power transient dampen more quickly than down-power transients. The 6-pitch reactor power response is indicated in red and the 1-pitch configuration is indicated in pastel color tones. The average transient response period (seconds) of the 6-pitch configuration is 7 minutes. This is slightly longer than the 1-pitch configuration period due to the slower mass flow rate using the 6-pitch WHIP (higher pressure losses). The slower flow also results in slightly higher power oscillation amplitudes because temperature reactivity effects dominant core physics. The initial power response in the 1-pitch configuration opposes the power demand signal. This can be seen by the decrease in reactor power response in the first 3 minutes from the start of the transient. This opposing power behavior, in the up-power case, is effectively eliminated in the 6-pitch configuration. The initial opposing power response in the down-power case is greatly reduced in both configurations. In both the up-power and down-power cases, the maximum power over-shoot is about 10% of the highest power value. In the up-power transient case, the power oscillations are essentially dampened approximately 40 minutes after transient initiation. Down-power transients take about 100% longer to dampen regardless of WHIP configuration.

The oscillatory behavior of the power response is dominated by the temperature reactivity feedback in the flowing system. This can clearly be seen in fig. 5.17. The flow reactivity contribution to transient power response generally opposes the temperature reactivity contribution. Therefore, any effect that reduces the flow reactivity contribution results in slightly more pronounced oscillations in power response following a power transient. Pastel color tones indicate the 1-pitch WHIP configuration. For both the up-power and down-power transients, final steady state reactivity values for the 1-pitch case are settle on equal but opposite values of about 60 pcm. This divergence at steady state is $<$ 10 pcm for the 6-pitch configuration. The WHIP has the affect of minimizing the system temperature and flow changes between initial and final steady state conditions. The flow reactivity in both WHIP configurations and transient conditions changes monotonically while temperature reactivity changes are oscillatory. As noted in the power changes in Fig. 5.18, the period of the temperature reactivity oscillations are slightly shorter in the 1-pitch configuration due to the slightly higher system flow rates as a result of lower pressure losses across the 1-pitch WHIP.

Oscillations during the down-power transient persist about 100% longer than the up-power case. Although the peak-to-peak total reactivity change is roughly equivalent between the two transients and WHIP configurations, the oscillation period is increase from about 6 minutes to 12 minutes for the down-power transient. Therefore, the same number of cycles are required to dampen the up or down transient and the oscillation period accounts for the longer time to achieve steady state conditions.

Fig. 5.19 shows that the system seeks the final stead-state condition along a reactivity phase-space geodesic. Dark red and blue indicate the 6-pitch WHIP response and the pastel color palet indicates the 1-pitch configuration. Flow reactivity oscillations occur on a time scale representative of a total loop transport time (2 minutes). Temperature reactivity oscillations occur on a period of 3-4 loop transport times and persist for a much longer time than flow effects. Fig. 5.19 shows that the 6-pitch configuration reduces deviation from initial system reactivity conditions and reduces flow reactivity oscillations in the core (flattened spirals in dark red and blue).

5.5.4.2 Spectral Response

A fast Fourier transform (FFT) using Python blackman methodology to minimize spectrum leakage due to truncation of the transform at 49900 seconds after the transient. Fig. 5.20 shows the FFT results for the 1- pitch (pastel color tones) and 6-pitch (primary color tones) WHIP configurations during the up-power (Fig. 5.20(a)) and down-power (Fig. 5.20(b)). Spectral analysis shows that for up-power transients the temperature reactivity leads the flow reactivity response by π radians. The flow reactivity leads the temperature reactivity by π radians during a down-power transient. Considering up-power transients (shown in Fig. 5.20(a)) first, as expected both the flow and temperature spectral amplitudes are largest at low frequencies. The total reactivity component amplitude is lower than both flow and temperature constituents due to the π radian phase difference between the two components. As frequency increase, the flow spectral component amplitude decreases for physical frequencies of interest. The temperature spectral amplitude remains essentially constant above 20 Hz. Comparing the 1-pitch to 6-pitch configuration, the single pitch WHIP has higher spectral component amplitudes across the frequency range of interest. Compared to the single pitch configuration, the 6-pitch WHIP reduces spectral amplitudes at all frequencies. This implies the 6-pitch WHIP acts to dampen the spectral response of the system to power transients by one order of magnitude in the frequency range of interest (below 10 Hz).

For down-power transients, the spectral amplitude character of the system is similar but different in a few significant ways. For down-power transients, the 6-pitch WHIP also acts to dampen the spectral response of the system in the frequency range of interest. The 6-pitch configuration has a noticeable local flow and temperature spectral peaks near 1-2 Hz. Notice that the spectral components are π radians out of phase and do not contribute to the total spectral response of the system. A local maximum in the temperature spectral amplitude is also observable at about 65 Hz and 73 Hz for the 6-pitch and 1-pitch configurations, respectively. For the down-power transient, the total spectral amplitude is essential constant with the exception of this local max. As will be described in the next section, down-power transients are generally closer to regions of flow instability than up-power transients. Nuances in the down-power spectrum are indicative of these additional system energy states.

5.5.4.3 Instabilities

Compound dynamic instabilities have been characterized and studied in prior work (Watanabe et al, 2008)(Nayak, 1995)(Nayak et al, 2009)(Saha, 2018)(Sabharwall, 2012)(Dass, 2009)(Vijayan, 2010)(Zou, 2017)(Anbrosini, 1998). Despite the wide and thorough study of stabilities of this type, few actually study the unique features associated with the compound effects of flow and temperature reactivity on a flowing fuel nuclear reactor system. A thorough examination of system stability under natural circulation liquid fuel flow conditions is key to creating a high confidence documented safety design basis case for licensed reactor operations.

Natural circulation systems are inherently more unstable due to their low thermal driving head. Disturbances in the thermal driving head, e.g. changes in fluid density, affects the flow which in turn affects the thermal driving head. This strong coupling between flow and the thermal driving head is nonlinear and can result in chaotic flow instabilities. Single phase flow instabilities will be examined in the MsNB. As defined by Vijayan (Vijayan, 2010), instabilities in natural circulation systems is defined as the system returning to the original state following a perturbation. In the case of power demand changes, an instability is understood to mean the system achieves a final steady-state power level equal to the demand placed on the system. If the system continues to oscillate or experiences oscillations with increasing amplitude, the system is neutrally stable or unstable, respectively. Unstable flow typically results in chaotic flow or flow reversals.

Instability in system flow is not desirable because it can cause forced mechanical vibration of reactor components, can results in undesirable power excursions, and can challenge thermo-chemical salt limits and design basis casualties. Regions of flow instability should be avoided by design and control systems. It may be necessary to provide safety significant control and design features to actively prevent flow instabilities from occurring during reactor start-up, steady-state, and shutdown power operations.

The nature of instability thresholds are typically related to the relationship between the average cold and hot leg average fuel salt densities. Namely, for stable operations at time t, the average cold leg density should exceed the average hot leg density,

$$\int_{\text{Rx Inlet}}^{\text{Chimney Outlet}} \rho(x,t)\,dx \quad < \quad \frac{L_h}{L_c}\int_{\text{Hx Inlet}}^{\text{Downcomer Outlet}} \rho(x,t)\,dx \tag{5.29}$$

where L_h and L_c are the vertical linear length of the hot and cold legs, respectively. Asymmetric loop designs that incorporate unequal hot and cold vertical loop lengths can improve flow stability. Moreover, instabilities in the MsNB are typically due to density wave instabilities (DWI) as a primary effect compounded by a secondary effect associated with the flow reactivity mechanism. Density wave and flow effects are related in a time delayed manner which further complicates the compounding nature of the feedback. The proximity to flow instability in the single-phase MsNB system is dependent on operating conditions and loop geometry (including WHIP features that affect flow reactivity feedback contributions). A parametric investigation was completed to characterize stability regions when varying power transients (magnitude and duration) and WHIP geometry (pitch number). Increasing the difference between the initial and final power demand and reducing the duration over which the power demand change occurs both pose a relatively more severe transient on the system and reduces the proximity to instability.

5.5.4.4 Summary

Inserting a WHIP in-core can have significant damping effects on power demand transients imposed on an operating MsNB. These benefits will also manifest during startup and shutdown reactor operations. The WHIP provides enhanced stability against flow oscillations and reversals by increasing the system pressure losses and increasing the effective delayed neutron precursor fraction under circulating conditions. This improves controllability and natural circulation flow stability under power operations. Increasing the streamline flow path length seems to further improve these positive performance traits. Increasing the streamline path length is a design feature engineered by changing the WHIP pitch and helix angle. In addition, the WHIP acts to mitigate the flow reactivity contribution to power feedback. As a result, initial power changes that counter heat exchanger power demand are mitigated. Although not stability limiting, there are regions that may impose thermo-physical limitations on the fuel salt particularly late in core life. Fuel salt fission product and fuel solubility limits may be exceeded during significant transient which causes precipitation of solids on colder heat transfer surface near or in the heat exchanger, for instance.

5.6 Future Work

Specific recommended work to follow-up on this study might include:

- Determination of actual MsNB design system friction and pressure loss coefficients.
- Determine flow reactivity coefficient using refined numerical approach. This can be achieved by solving the criticality equation for a range of fuel salt velocities. This is relationship then numerically solve for the flow coefficient.
- Verification and validation of python program that models power demand transients against bench marked equivalence code, using pre-conditioned Jacobian free Newton Krylov

(PJFNK) methodology for instance.

- Consider proximity to fuel salt freezing, solubility, and vaporization under nominal, off-nominal, and accident conditions during limiting power transient conditions.
- Evaluate for the bounding power transient safety case that inserts the maximum credible instability into the system.

Literature review indicates little prior work has been completed in the area of MSR power transient behavior especially in the area of natural circulation. This is a rich field for research and will help inform MSR safety basis analysis in support of concept demonstration and commercial licensing.

Table 5.1: Model Thermal and Neutronics Parameters

Parameter	
α_T (pcm/K)	-3.5
l^* (sec)	9.65 x 10^{-5}
β	6.96 x 10^{-3}
λ (sec)	0.1
h (m)	1.09
ξ	25.0
α_f (s/cm)	3.3976917 x 10^{-1}
dt (sec)	1.0
dx (mm)	1.0

Table 5.2: MsNB Reactor Linear Integrated Path Dimensions

Parameter	**Path Length (cm)**
Reactor Inlet	0.0
Reactor Outlet	166.0
Loop Top Height	209.0
Heat Exchanger (Hx) Inlet	234.0
Hx Loop Top	282.5
Hx Outlet	282.5
Loop Bottom	497.5
Reactor Inlet	571.0

Table 5.3: WHIP Six Pitch Configurations (lengths in cm) for 238.8 kg moderator, 179 cm total length, 50 cm diameter, and 5 cm root radius

Parameter	**1-Pitch**	**2-Pitch**	**3-Pitch**	**4-Pitch**	**5-Pitch**	**6-Pitch**
# pitches	1	2	3	4	5	6
Flight Width (cm)	60.16	26.47	15.16	9.52	6.31	3.85
Helix Angle (γ)	60.7	41.8	30.8	24.1	19.7	16.6
Channel Width (cm)	118.04	62.94	44.45	35.22	29.68	25.96
r=15 cm Streamline (cm)	202.49	260.08	334.86	417.46	504.03	593.54

Table 5.4: $f(u_\parallel)$ coefficients (see Eqn. 5.20)

WHIP configuration	f_0	f_1
1-pitch	0.6700	0.0985
2-pitch	0.6170	0.0884
3-pitch	0.5630	0.0785
4-pitch	0.5140	0.0707
5-pitch	0.4740	0.0623
6-pitch	0.4390	0.0584

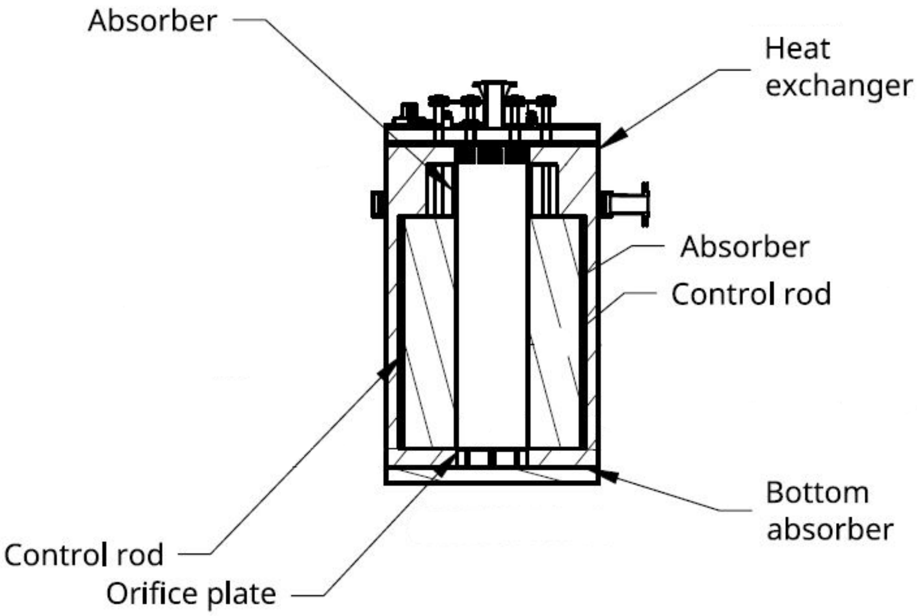


Figure 5.1: Axial cross section diagram of the FLiNaK molten salt nuclear battery. Flow propagates vertically up through the core and chimney, enters the outlet plenum, flows radially out through the heat exchanger, downward through the downcomer, collects below the core in the inlet plenum, and repeats the flow cycle. Total loop linear flow distance is 571 cm. Nominal transport time is 130 seconds. (Cardenas Melgar, 2021).

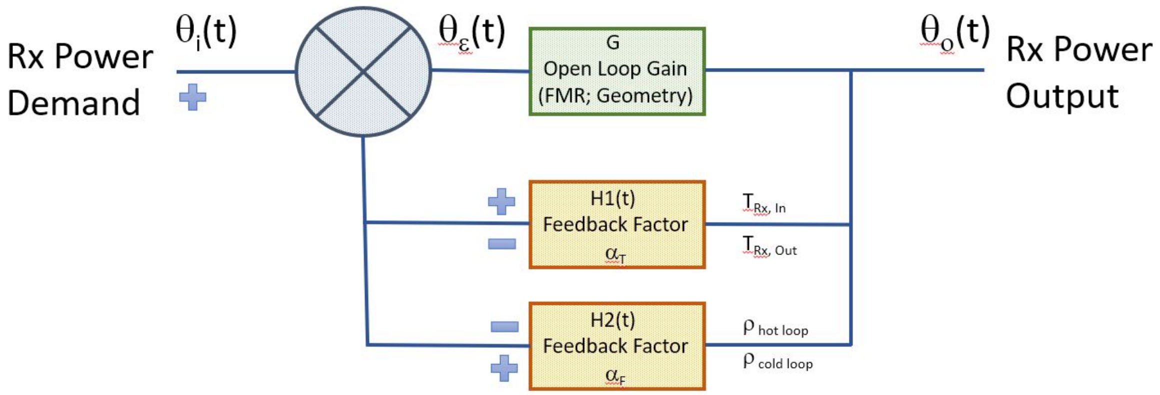


Figure 5.2: Compound Dynamic Power Control of a Natural Circulation Molten Salt Reactor System. Compound time-delayed feedback factors make the MSR power control system unique. In natural circulation modes of operation, the time-dependent flow reactivity feedback factor (H2) is a significant contribution to power response behavior and defines transitions to instabilities such as flow oscillations and reversals.

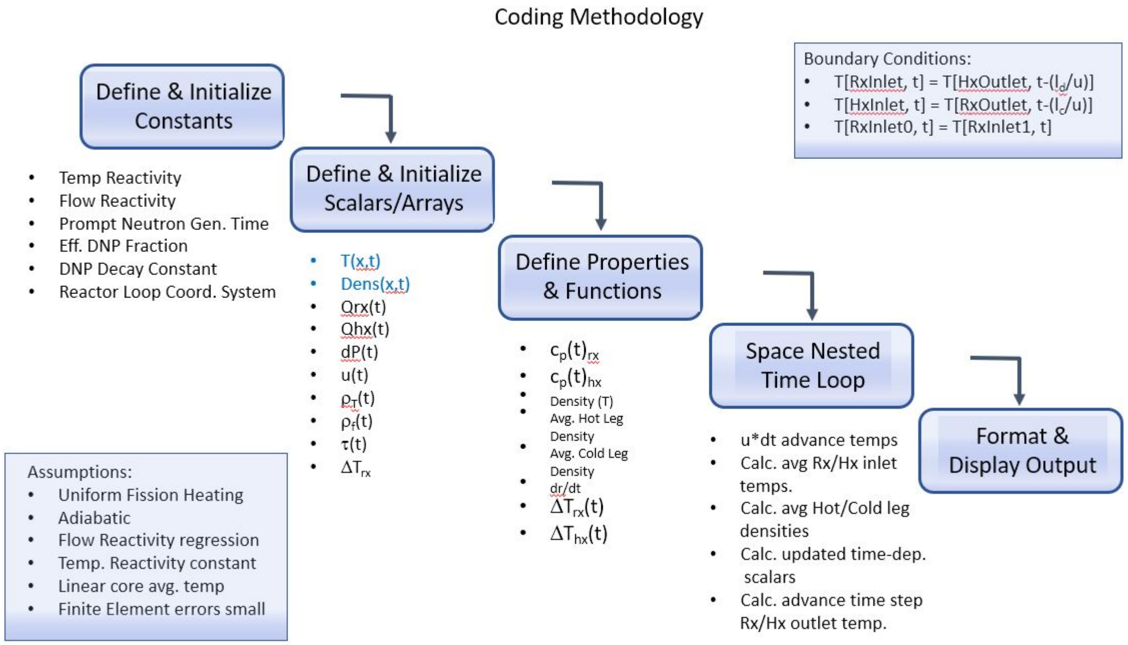


Figure 5.3: Coding Methodology.

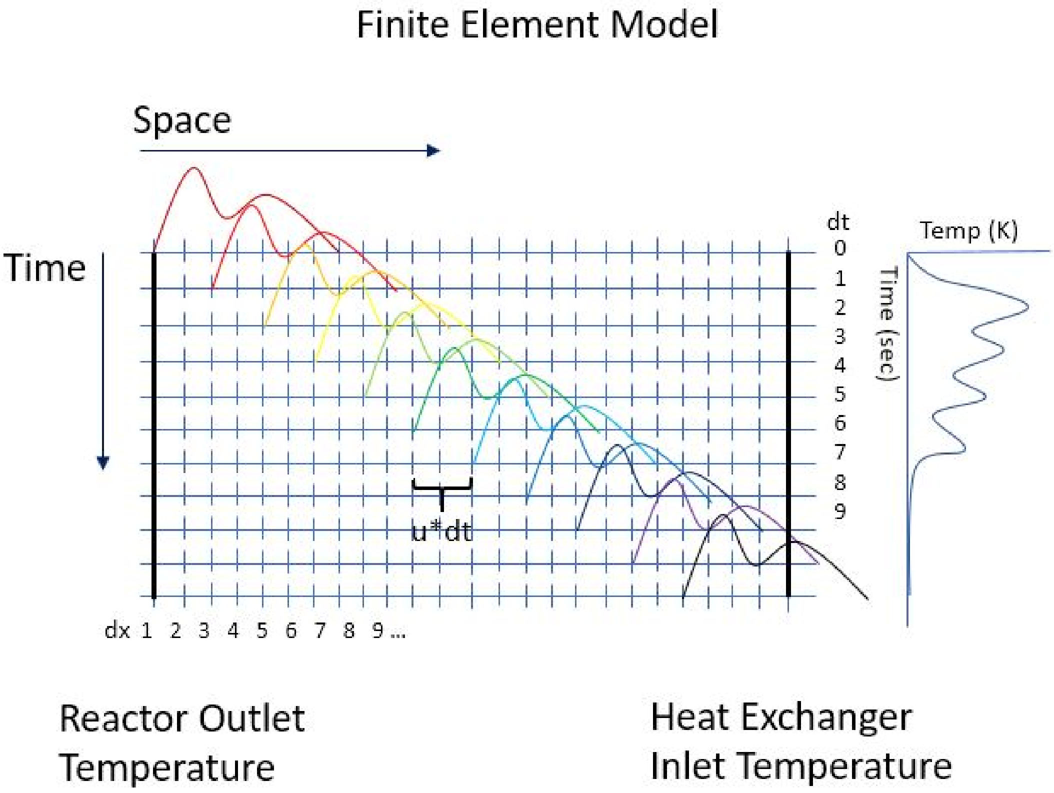


Figure 5.4: Forward Euler Approach.

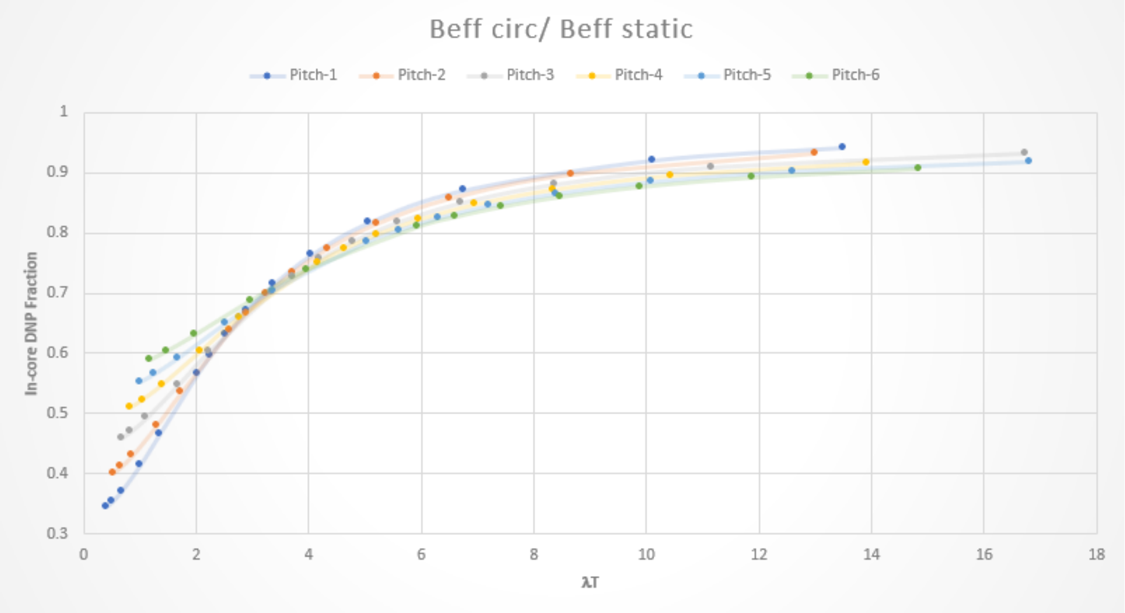


Figure 5.5: βeff circ/β_{eff} static for various WHIP pitch configurations. Configuration differences become more pronounced at high flow velocities. Interestingly, there is a flow velocity at λT = 3.2 where there is no difference between WHIP pitch configurations in terms of its affect on β_{eff} under circulation conditions.

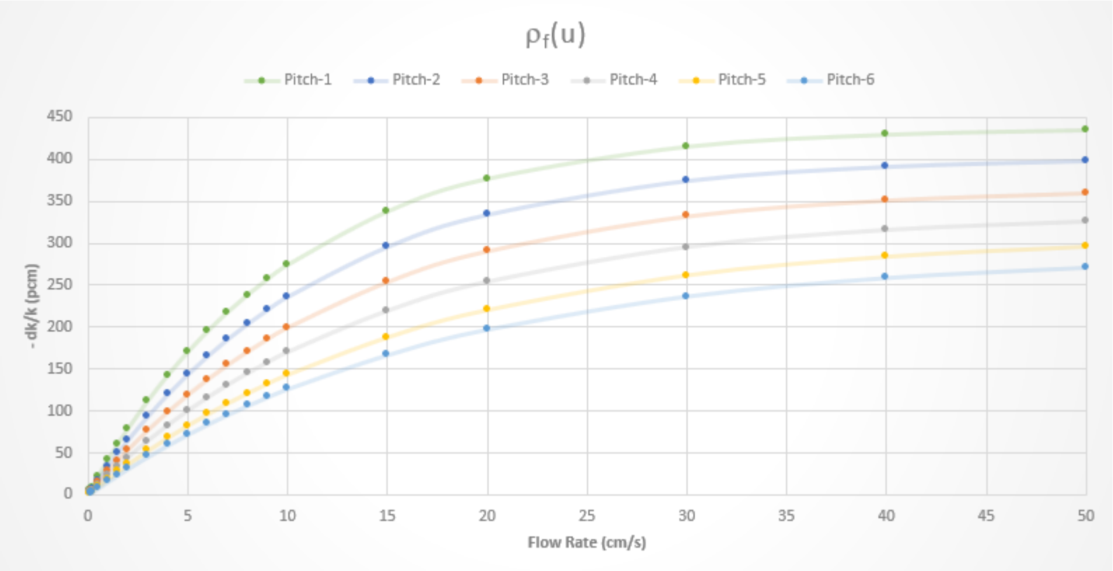


Figure 5.6: Flow reactivity for various WHIP pitch configurations. Flow reactivities at asymptotic flow values are proportional to the delayed neutron loss fraction in the loop external to the core.

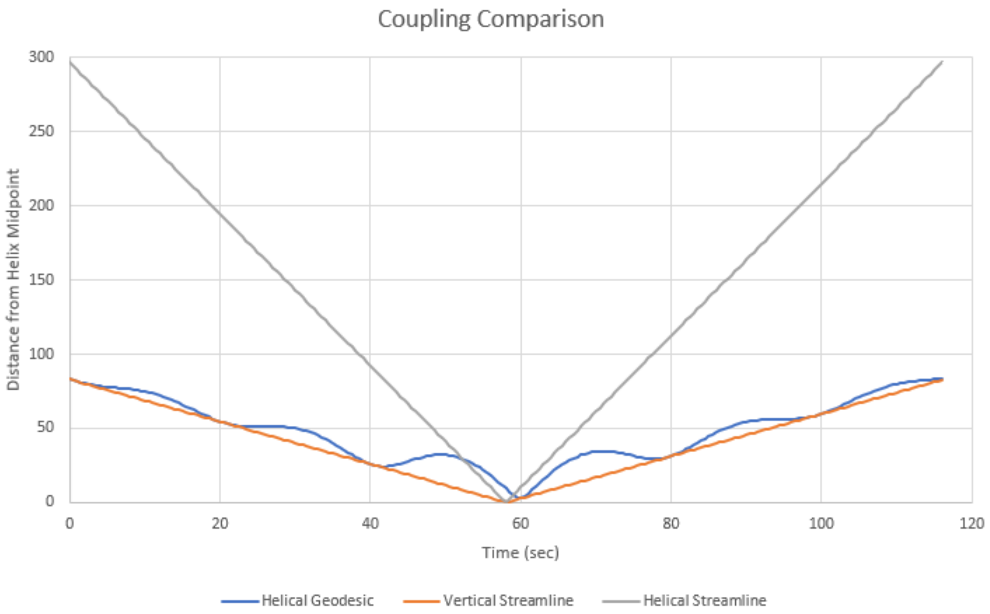


Figure 5.7: Coupling Comparison in a 6-pitch WHIP configuration. Neutronic coupling in the core is proportional to the proximity between fissile regions. The orange curve represents the straight line vertical distance from a core axial midpoint. The gray curve represents the distance from a core midpoint following the fuel salt helical streamline (longer path length). The blue curve represents the geodesic distance (or neutron straight line transport path length). Because neutrons are not constrained the fluid path, neutronic coupling in the WHIP core closely resembles that of the traditional vertical flow core while the helical streamline (gray curve) constrains the delayed neutron precursor transport in core.

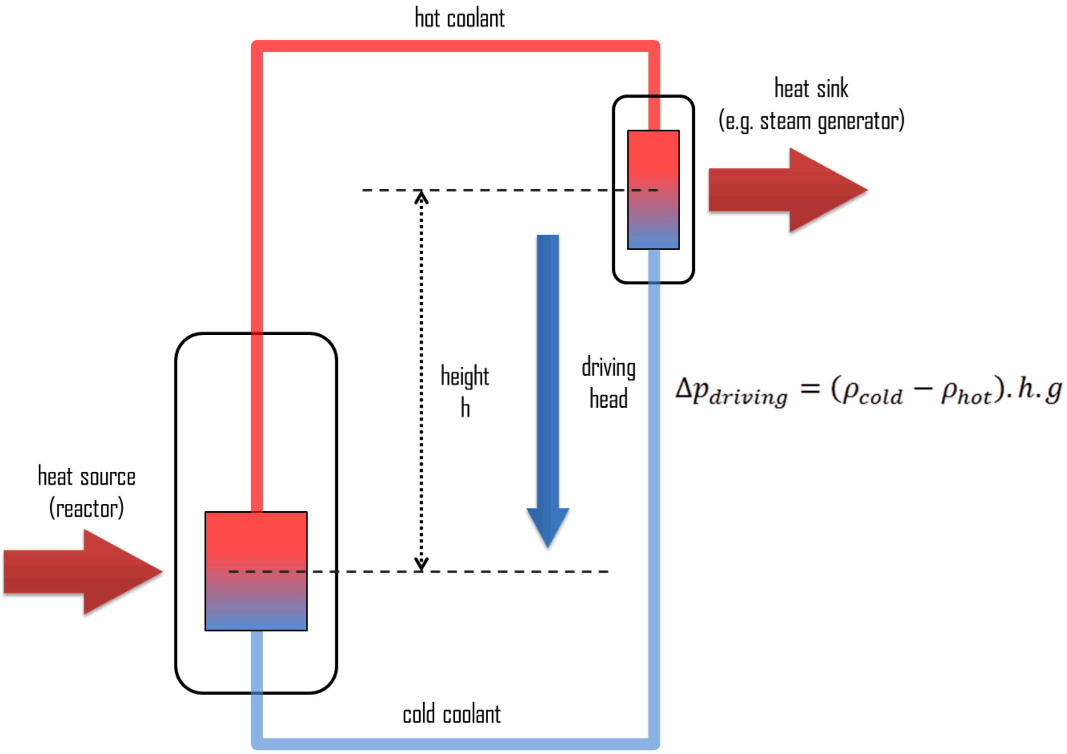


Figure 5.8: Simplified Natural Convection Closed Loop System. (IAEA, 2005)

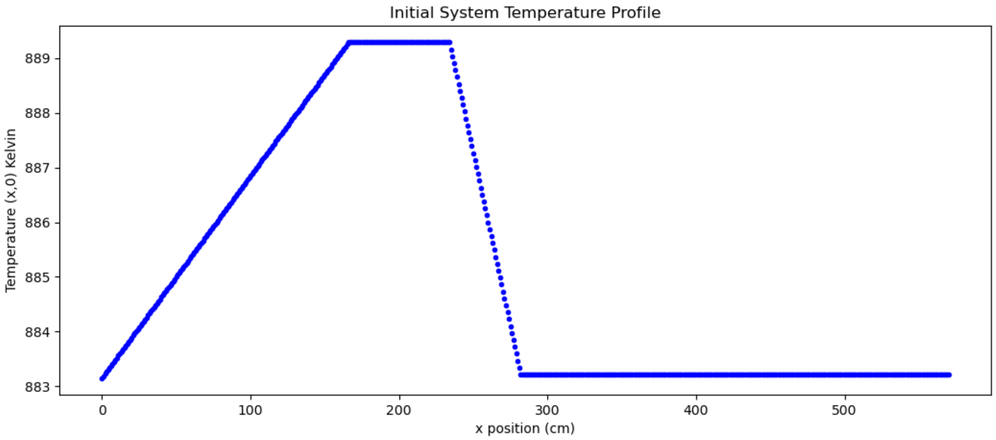


Figure 5.9: Initial reactor loop temperature profile. Nominal differential temperature across the reactor at 400 KWth power operation is about 5 degree Celsius. Heat generation and removal are assumed to be linear within the reactor and heat exchanger. The reactor thermodynamic control volume is assumed to be adiabatic and no preheating occurs in the downcomer flow region.

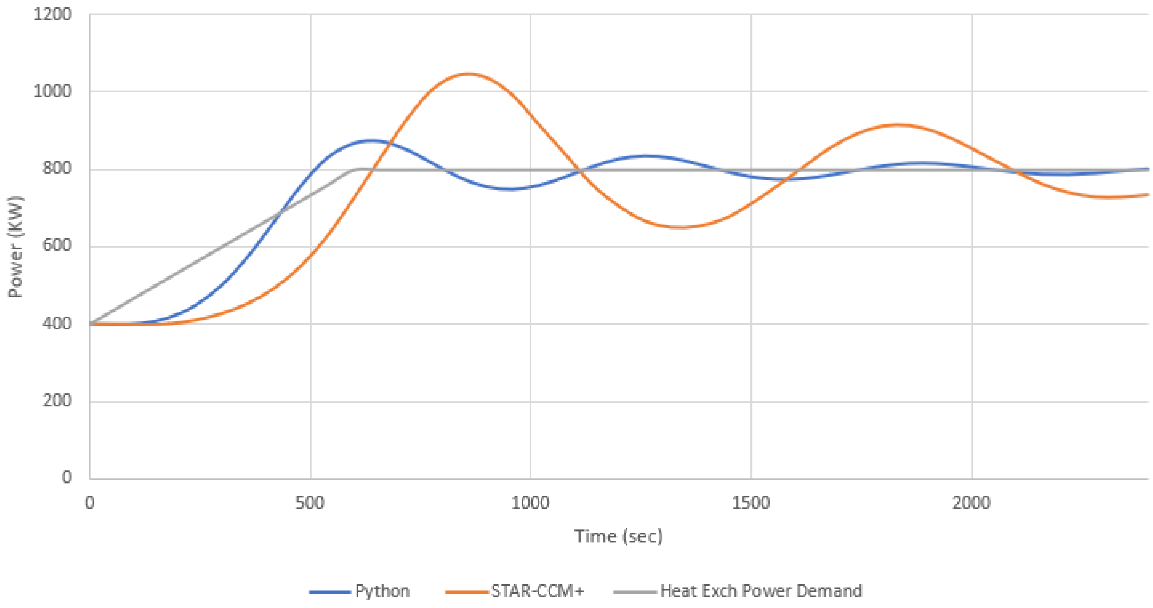
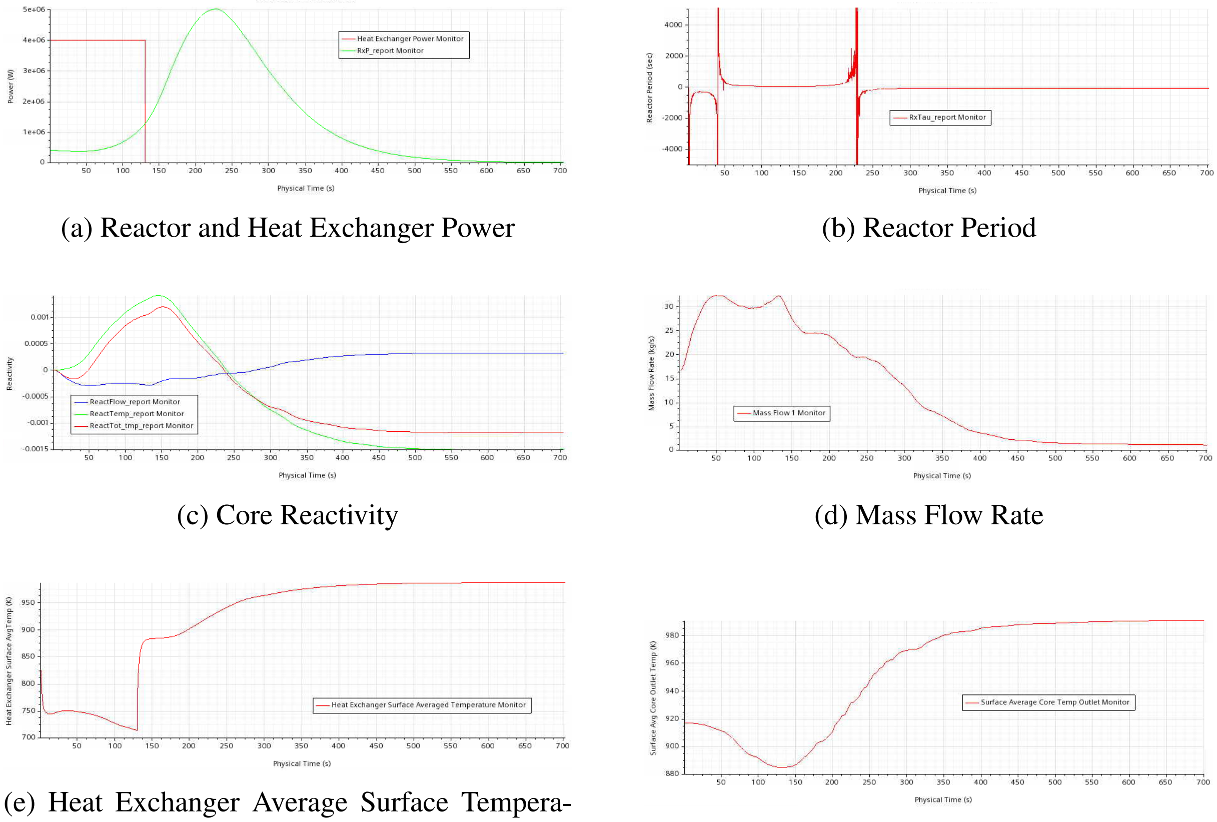


Figure 5.10: Comparison of simulated results for a 400 to 800 KW power demand transient response. The Python code generally provides a faster power response with reduced power overshoots and faster power damping than the CFD code in STAR-CCM+. The neutronics are the same in both codes and therefore differences can be attributed to the improved fidelity of the STAR-CCM thermal-hydraulic computations.



(a) Reactor and Heat Exchanger Power

(b) Reactor Period

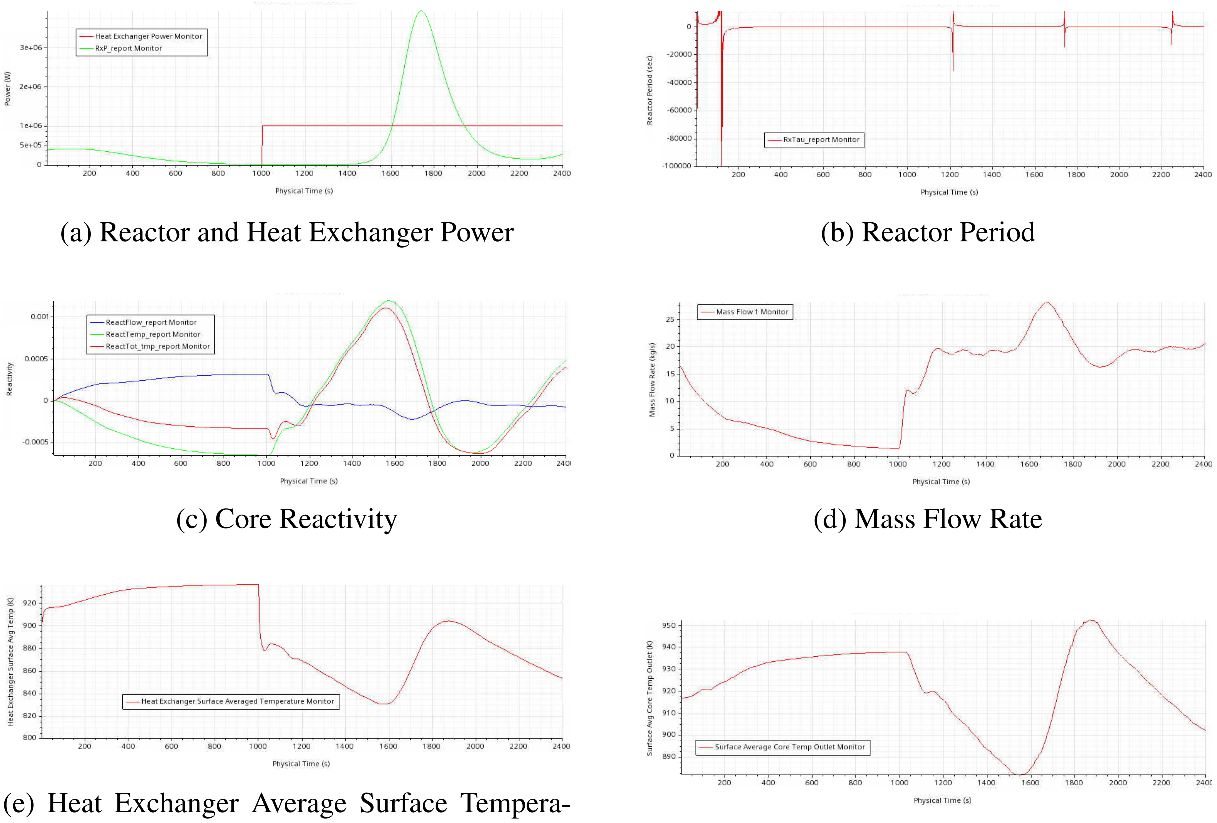
(c) Core Reactivity

(d) Mass Flow Rate

(e) Heat Exchanger Average Surface Temperature

(f) Reactor Outlet Surface Averaged Temperature

Figure 5.11: STAR-CCM+ simulation showing overpower demand followed by loss of heat sink. Initial system power is 400 KW stead state. At time zero, the system experiences a prompt power demand increase to 4 MW for a duration of 130 seconds which corresponds to the time needed for flow reactivity dominance to subside. At time 130 seconds, the system power demand decreases to 0 KW. The initial reactor power response decreases to about 370 KW due to flow reactivity effects. At approximately 20 seconds, colder fuel salt arrives in the reactor which contributes positive reactivity. Temperature effects dominate the core power response to the remainder of the transient. The mass flow rate more than doubles in a short period of 40 seconds and decays to about 1 kg/sec after a period of approximately 10 minutes concombinate with reactor shutdown. Severe power rates do result in significant system temperature transients which contribute to high thermal stresses particularly during cooldown where stresses can be significantly limiting to thick walled the reactor vessel structures. As expected, the heat exchanger has the coldest system temperatures and severe power transient can challenge both fuel salt solubility (approximately 823 K for 18 mol% UF_4) and solidus conditions (freezing temperature of approximately 726 K) of the fuel in local regions (see fig. (e)). For this reason, active control systems must be used to mitigate the severity of such transients.



(a) Reactor and Heat Exchanger Power

(b) Reactor Period

(c) Core Reactivity

(d) Mass Flow Rate

(e) Heat Exchanger Average Surface Temperature

(f) Reactor Outlet Surface Averaged Temperature

Figure 5.12: STAR-CCM+ simulation showing effect of loss of heat sink followed by rapid restart. Initial system power is 400 KW stead state. At time zero the system experiences a prompt loss of heat sink for 1000 seconds (approximately 17 minutes). At time 1000 seconds, the system power demand increases rapidly to 1 MW. On loss of heat sink, initial core power peaks at approximately 422 KW due to flow reactivity effects. Maximum reactor outlet temperatures approach about 940 K. Upon reactor rapid startup at 1000 sec, core power reaches a minimum at approximately 1200 sec at about 388 KW before rising rapidly to a peak core power of approximately 3.8 MW. System mass flow rate response autonomously to the heat sink power demand. As in the python model, core reactivity is dominated by the temperature coefficient. Thermal transients in the heat exchanger are severe and must be evaluated in detail to ensure yield strength limitation are not approached at temperatures above the reference transition temperatures of the material involved in the design (316 SS in this case).

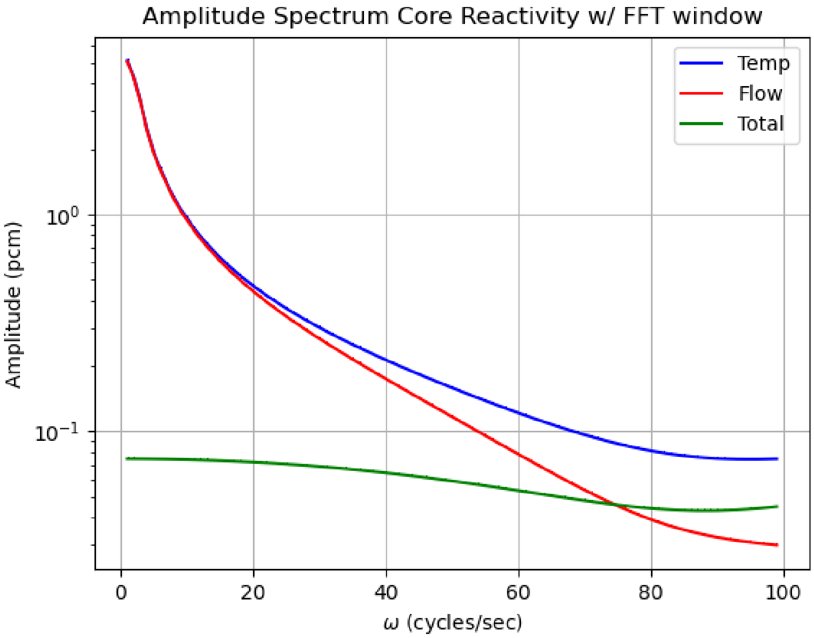


Figure 5.13: Stable Power Transient from 250 KW to 1000 KW over 10 minutes (Python program). Fast Fourier Transform of reactivity components. At low frequencies the temperature and flow reactivity components are out of phase. As frequency increases, the temperature component becomes the dominant contributor to overall core power behavior.

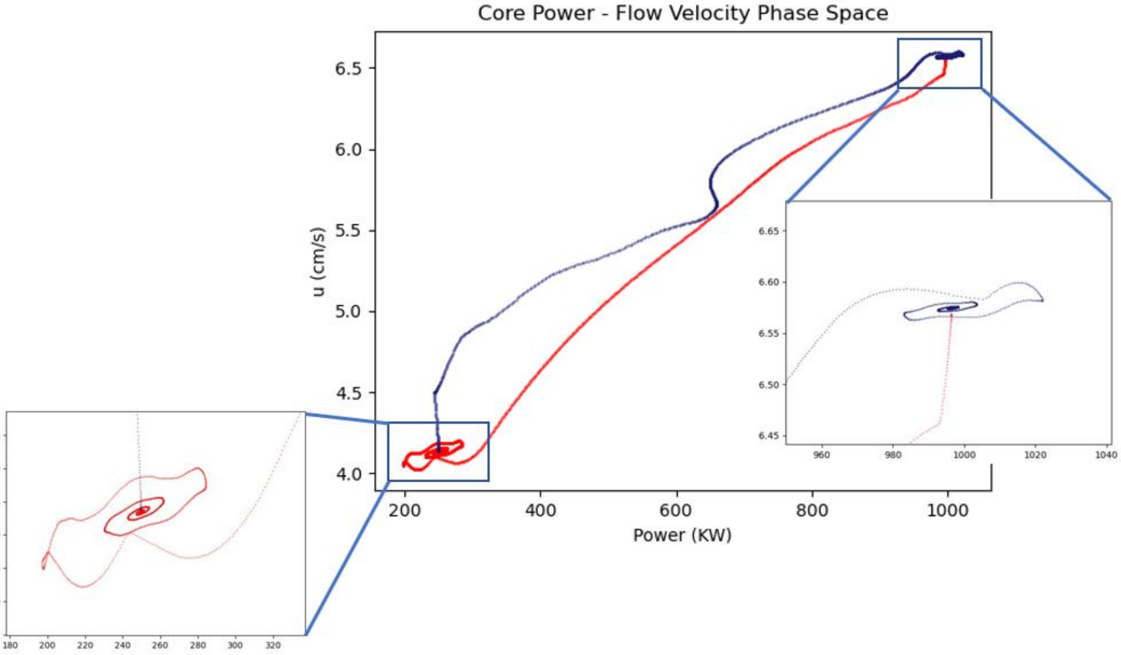


Figure 5.14: Power - Velocity phase space representation of a stable power transient (Python program). Up-power transient from 250KW to 1000KW indicated in blue. Down-power transient from 1000KW to 250KW indicated in red.

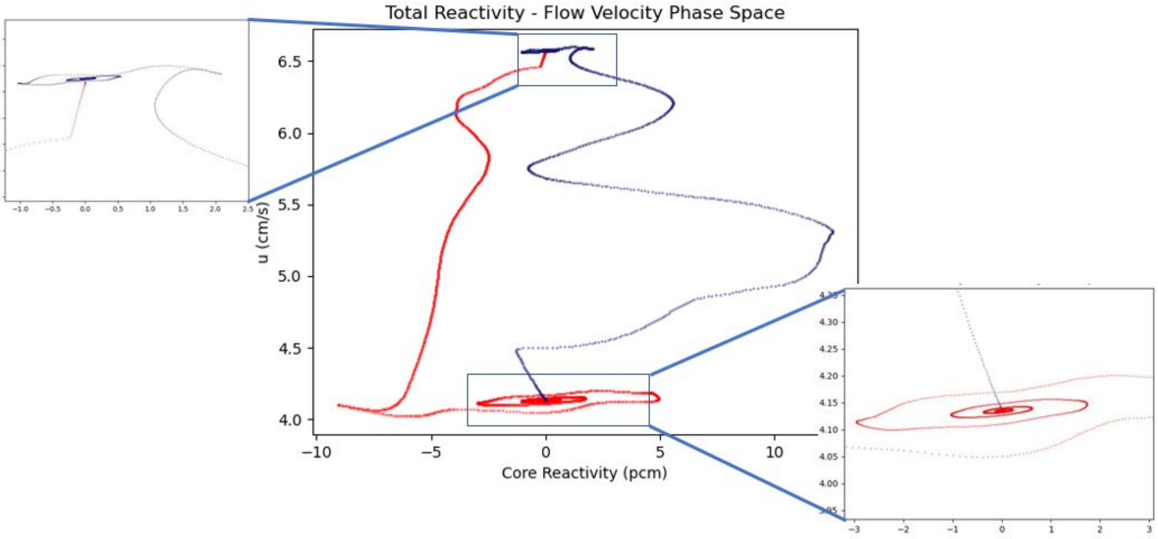


Figure 5.15: Reactivity - Velocity phase space representation of a stable power transient (Python program). Up-power transient from 250KW to 1000KW indicated in blue. Down-power transient from 1000KW to 250KW indicated in red.

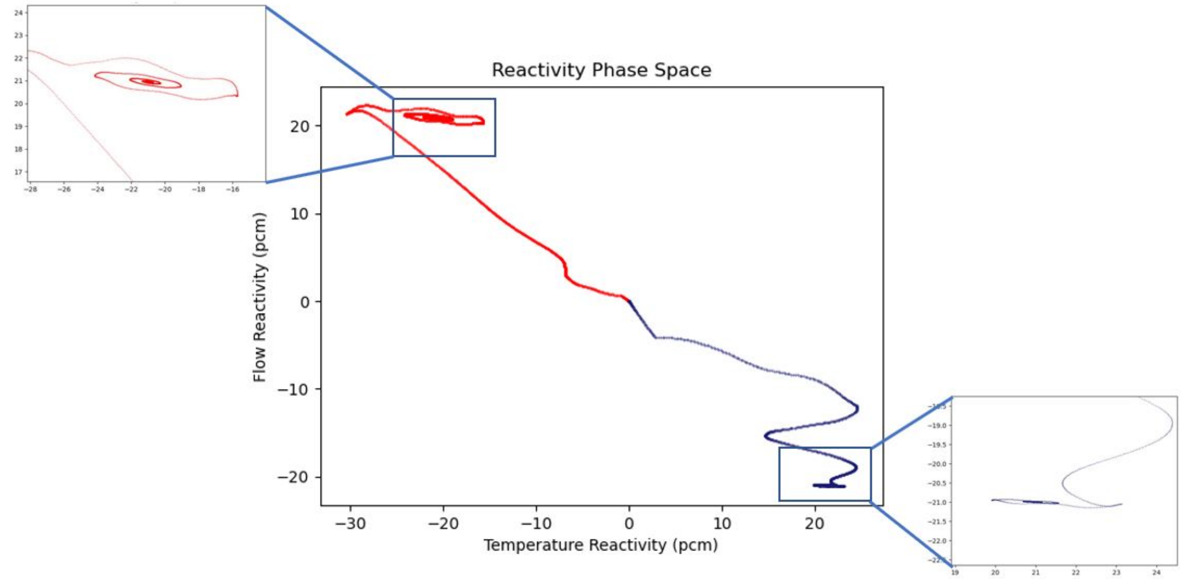
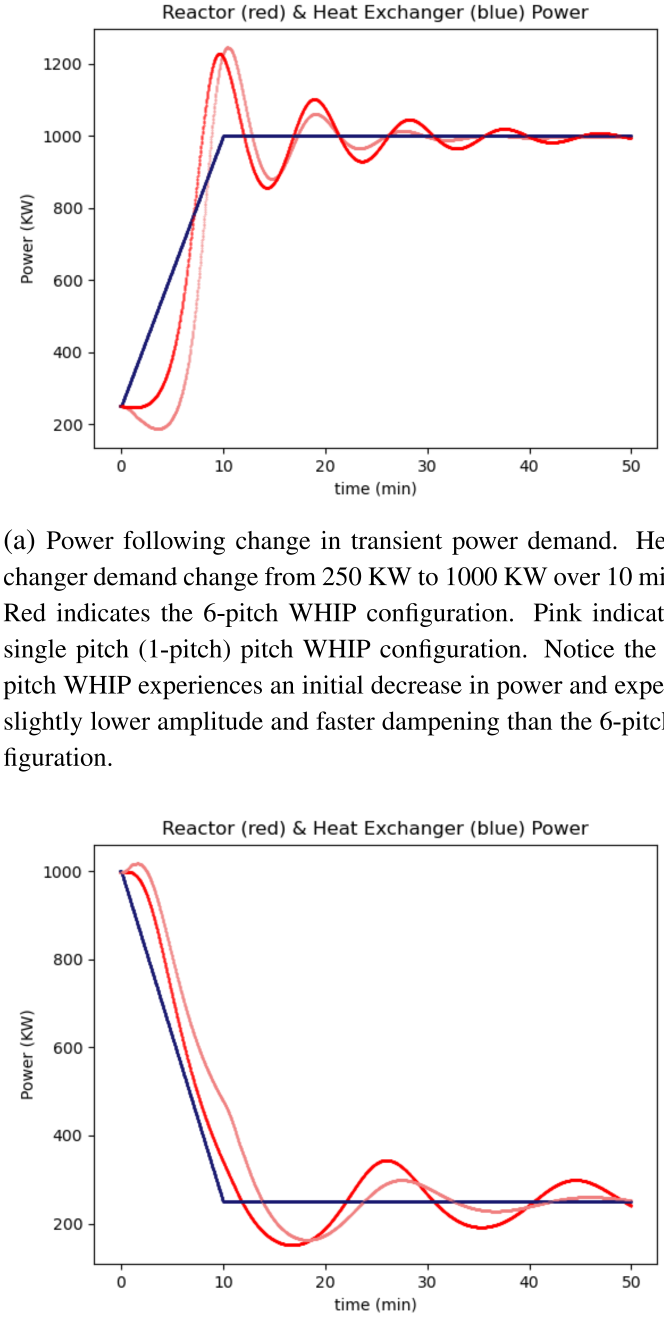


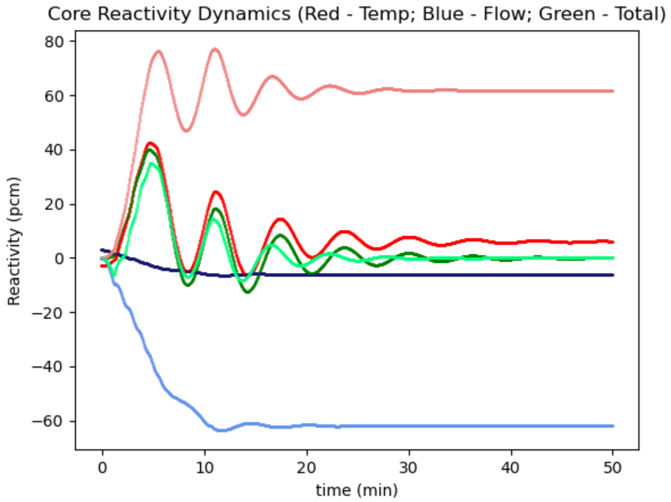
Figure 5.16: Temperature - Flow Reactivity phase space representation of a stable power transient (Python program). Up-power transient from 250KW to 1000KW indicated in blue. Down-power transient from 1000KW to 250KW indicated in red.



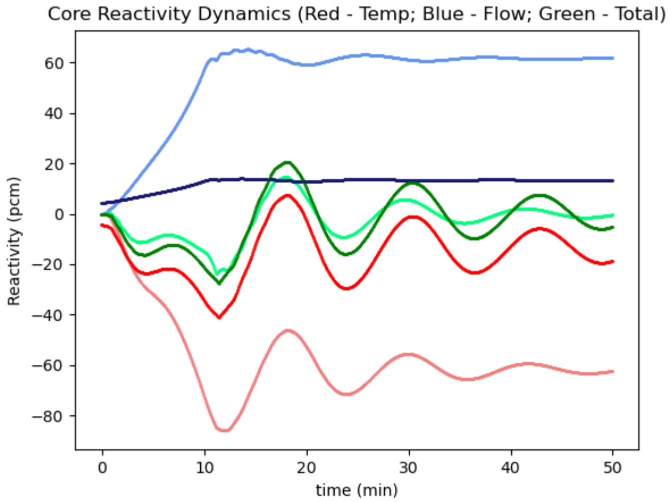
(a) Power following change in transient power demand. Heat exchanger demand change from 250 KW to 1000 KW over 10 minutes. Red indicates the 6-pitch WHIP configuration. Pink indicates the single pitch (1-pitch) pitch WHIP configuration. Notice the single pitch WHIP experiences an initial decrease in power and experience slightly lower amplitude and faster dampening than the 6-pitch configuration.

(b) Power following change in transient power demand. Heat exchanger demand change from 1000 KW to 250 KW over 10 minutes. Red indicates the 6-pitch WHIP configuration. Pink indicates the single pitch (1-pitch) pitch WHIP configuration. Notice the single pitch WHIP experiences an initial increase in power and experience slightly lower amplitude and faster dampening than the 6-pitch configuration.

Figure 5.17: Up and down power demand transients. Power versus time after transient initiation.



(a) Reactivity following change in transient power demand. Heat exchanger demand change from 250 KW to 1000 KW over 10 minutes. Temp (red), Flow (blue), and Total reactivity (green) indicates the 6-pitch WHIP configuration. The single pitch WHIP configuration is shown in pastel color scheme, i.e. temp (pink), light blue (flow), and total reactivity (light green).



(b) Reactivity following change in transient power demand. Heat exchanger demand change from 1000 KW to 250 KW over 10 minutes. Temp (red), Flow (blue), and Total reactivity (green) indicates the 6-pitch WHIP configuration. The single pitch WHIP configuration is shown in pastel color scheme, i.e. temp (pink), light blue (flow), and total reactivity (light green).

Figure 5.18: Up and down power demand transients. Reactivity versus time after transient initiation.

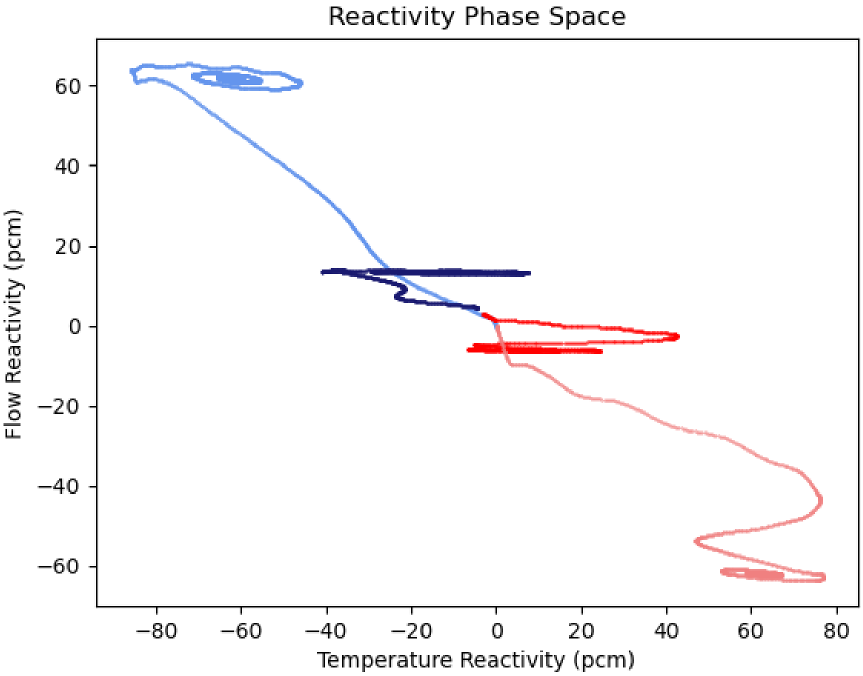
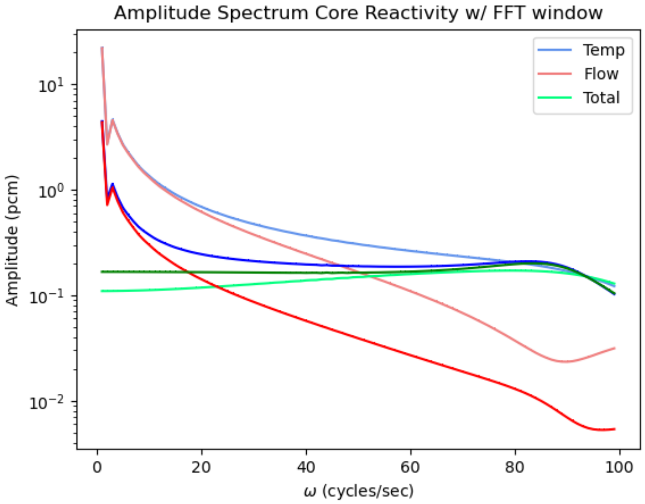
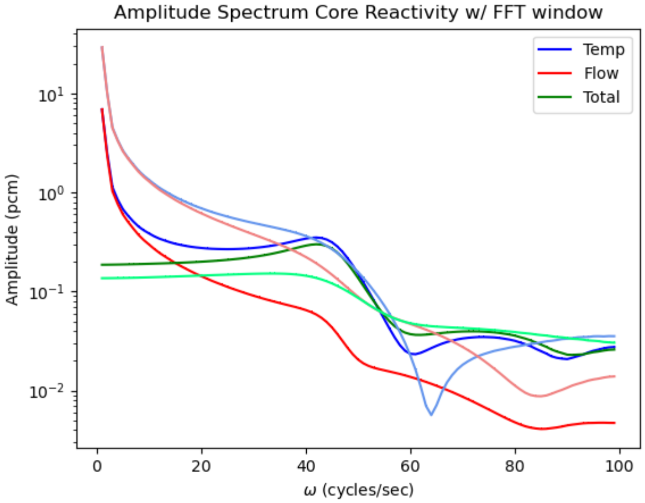


Figure 5.19: Temperature versus Flow Reactivity Phase Space Diagram (Python program). The 6-pitch WHIP configuration is represented in the strong red and blue colors. The pastel color scheme represents the single pitch WHIP configuration. The reddish curves depict a 250 KW to 1000 KW up power demand transient while the blueish curves depict a 1000 KW to 250 KW down power demand transient.



(a) Fourier Transform of Reactivity Response function following a power transient from 250 KW to 1000 KW. For the 6-pitch WHIP configuration, temp amplitude is in red, flow is in blue, and total amplitude is in green. The single pitch WHIP is indicated in the pastel color scheme, i.e. temp amplitude in pink, flow amplitude is in light blue, and the total amplitude is in light green.



(b) Fourier Transform of Reactivity Response function following a power transient from 1000 KW to 250 KW. For the 6-pitch WHIP configuration, temp amplitude is in red, flow is in blue, and total amplitude is in green. The single pitch WHIP is indicated in the pastel color scheme, i.e. temp amplitude in pink, flow amplitude is in light blue, and the total amplitude is in light green.

Figure 5.20: Up and down power demand transients. Fourier Transform Amplitude Spectrum versus Frequency.

References

[1] AEC, 1972. An Evaluation of the Molten Salt Breeder Reactor, 1972. WASH-1222, U.S. Atomic Energy Commission.

[2] Alsayyari F., 2020. A nonintrusive adaptive reduced order modeling approach for a molten salt reactor system, Annals of Nuclear Energy, Volume 141, 107321.

[3] Altahhan M., 2020. Preliminary design and analysis of Liquid Fuel Molten Salt Reactor using multi-physics code GeN-Foam, Nuclear Engineering and Design, Volume 369, 110826.

[4] Ambrosini W., et al, 1998. Numerical Analysis of single-phase, natural circulation in a simple closed loop, Dipartimento di Construzioni Meccaniche e Nucleari, UniversitÃ¡ di Pisa.

[5] Aufiero, M., Cammi, A., Fiorina, C., Ricotti, M.E., 2013. 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 (1), 473-486.

[6] Aufiero, M., Brovchenko, M., Cammi, A., Clifford, I., Geoffroy, O., Heuer, D., Laureau, A., Losa, M., Luzzi, L., Merle-Lucotte, E., Ricottia, M.E., Rouch, H., 2014. Calculating the effective delayed neutron fraction in the molten salt fast reactor: analytical, deterministic and Monte Carlo approaches. Ann. Nucl. Energy 65, 78-90.

[7] Aufiero, M., Martin, M. and Fratoni, M. 2016. XGPT: Extending Monte Carlo Generalized Perturbation Theory capabilities to continuous-energy sensitivity functions. Ann. Nucl. Energy, 96 295-306.

[8] Aufiero, M. et al. 2015. A collision history-based approach to sensitivity/perturbation calculations in the continuous energy Monte Carlo code Serpent, Ann. Nucl. Energy, 152, 245-258.

[9] Balk A., 2000. On the Kolmogorov-Zakharov spectra of weak turbulence, Physica D 139 137-157.

[10] Ball, S.J., Kerlin, T.W., 1965. Stability Analysis of the Molten Salt Reactor Experiment, ORNL-TM-1070, Oak Ridge National Laboratory.

[11] Banoori, S., Reitsma, F., 2016. International Thorium Molten Salt Forum: ITMSF, Advances in Small Modular Reactor Technology Developments 2016 Edition booklet, Status Report - FUJI-MSR.

[12] Bauman, H.F., Cunningham III, G.W., Lucius, J.L., Kerr, H.T., Craven Jr., C.W., 1971. ROD: A Nuclear and Fuel-Cycle Analysis Code for Circulating-Fuel Reactors ORNL TM-3359. Oak Ridge National Laboratory.

[13] Behringer K., Kosaly G., Pazsit I., 1979. Linear Response of the Neutron Field to a propagating Perturbation of Moderator Density (two group theory of BWR noise), Nuclear Science Engineering, 72, 304.

[14] Betzler, B.R., Robertson, S., Davidson (nÃ©e Sunny), E.E., Powers, J.J., Worrall, A., Dewan, L., Massie, M., 2018. Fuel cycle and neutronic performance of a spectral shift molten salt reactor design, Ann. Nucl. Energy 119, 396-410.

[15] Betzler, B.R., Brown, N.R., Feng, B., Heidet, F., Rabiti, C., Sofu, T., Wysocki, A.J., 2018. Molten Salt Reactor Modeling and Simulation Functional Needs, Molten Salt Reactor Campaign Report, NTRD-MSR-2018-000450, Nuclear Technology Research and Development, US Department of Energy (DOE).

[16] Betzler, B.R., Powers, J.J., Worrall, A., 2017. Molten salt reactor and fuel cycle modeling and simulation with SCALE. Ann. Nucl. Energy 101, 489-503.

[17] Betzler, B.R., et al., 2017. Two-Dimensional Neutronic and Fuel Cycle Analysis of the Transatomic Power Molten Salt Reactor, ORNL/TM-2016/742, Oak Ridge National Laboratory.

[18] Bitter, R., Mohiuddin, T., Nawrocki, M., 2006. LabVIEW: Advanced programming techniques. Crc Press.

[19] Boston, Andy, 2014. Introduction to MCNP - The Monte Carlo Transport Code. University of Livermore, Department of Physics, Oliver Lodge Laboratory, Print.

[20] Bowman, S.M., 2011. SCALE 6: comprehensive nuclear safety analysis code system. Nucl. Technol. 174.

[21] Brewer, Roger, 2019. Criticality Calculations with MCNP 5: A Primer. LA-UR-09-00380 Los Alamos National Laboratory. Print.

[22] Brown, N.R., et al., 2016. Complete Sensitivity/Uncertainty Analysis of LR-0 Reactor Experiments with MSRE FLiBe Salt and Perform Comparison with Molten Salt Cooled and Molten Salt Fueled Reactor Models, ORNL/TM-2016/729. Oak Ridge National Laboratory.

[23] Cammi, A., Fiorina, C., Guerrieri, C., Luzzi, L., 2012. Dimensional effects in the modeling of MSR dynamics: moving on from simplified schemes of analysis to a multi-physics modeling approach. Nucl. Eng. Des. 246, 12-26.

[24] Cammi, A., et al., 2011. A multi-physics modeling approach to the dynamics of Molten Salt Reactor. Ann. Nucl. Energy 38, 6.

[25] Cardenas-Melgar, A., et al, 2021. Report on the Friction Factor for the Offset-Strip Fin Heat Exchanger", unpublished manuscript.

[26] Carter J., Borrelli R., 2020a. Integral molten salt reactor neutron physics study using Monte Carlo N-particle code, Nuclear Engineering and Design, Vol. 365, 110718.

[27] Carter J., Borrelli R., 2020b. Integral molten salt reactor temperature sensitivities using Serpent target motion sampling, Nuclear Engineering and Design, Vol. 369, 110863.

[28] Carter J., Arcilesi D. Jr, 2021. Investigation of neutron noise in a micro-scale, natural circulation molten salt fission battery system, Nuclear Engineering and Design 282 (2021) 111437.

[29] Carter J., Richardson R., Yoon S,. 2022. Numerical Analysis of Dynamic Load Following Response in a Natural Circulation Molten Salt Power Reactor System. Unpublished Manuscript.

[30] Cheng, M., Dai, Z., 2014. Development of a three dimension multi-physics code for molten salt fast reactor. Nucl. Sci. Technol. 25 010601.

[31] Dass A., et al, 2021. Stability analysis of a single-phase rectangular Coupled Natural Circulation Loop system employing a Fourier series based 1-D mode", retrieved from https://arxiv.org/abs/2012.11003 on 14 June 2021.

[32] Department of Energy, Idaho National Laboratory, 2021. *High Performance Computing*, Retrieved from https://inl.gov/article/inl-joins-high-performance-computing-consortium on 12 May 2021.

[33] Department of Energy, DOE-HDBK-1019/1-93, 1993. DOE Fundamentals Handbook: Nuclear Physics and Reactor Theory, Vol. 1 of 2.

[34] Desrayaudi G., et al, 2005. Steady and unsteady natural circulation in a single-phase 2D-annular loop, Proceedings of the 3rd IASME/WSEAS Int. Conf. on Heat Transfer, Thermal Engineering and Environment, Corfu, Greece.

[35] Desrayaudi, G., et al, 2013. Two-dimensional numerical analysis of a rectangular closed-loop thermosiphon, Appl. Therm. Eng. 50: 187-196.

[36] Dolan T., 2017. Molten Salt Reactors and Thorium Energy, Woodhead Publishing.

[37] Diamond, D.J., Brown, N.R., Denning, R., Bajorek, S., 2018. Phenomena important in modeling and simulation of molten salt reactors, BNL-114869-2018-IR. Brookhaven National Laboratory (BNL).

[38] Dolan, Thomas J., 2017. Molten Salt Reactors and Thorium Energy, Chapter 18. Cambridge, MA: Wood head Publishing. Print.

[39] Duderstadt, James, J., Nuclear Reactor Analysis, John Wiley & Sons, Inc., 1976.

[40] Dulla S., 2005. Models and methods in the neutronics of fluid fuel reactors, (PhD Thesis), University of Turin.

[41] Dykin V., Pazsit I., Sanchez R., 2016a. Remarks on the neutron noise induced by propagating perturbations in a MSR, Annals of Nuclear Energy 90 93-105.

[42] Dykin V., Pazsit I., 2016b. The Molten Salt Reactor Point-Kinetic Component of Neutron Noise in two-group diffusion Theory, Nuclear Technology, Volume 193, 404-415.

[43] Engel, J.R., Grimes, W.R., Bauman, H.F., McCoy, H.E., Dearing, J.F., Rhoades, W.A., 1980. Conceptual design characteristics of a denatured molten-salt reactor with oncethrough fueling, ORNL/TM-7207. Oak Ridge National Laboratory (ORNL).

[44] Engel, J.R., Prince, B.E., 1967. The Reactivity Balance in the MSRE, ORNL-TM-1796. Oak Ridge National Laboratory.

[45] Fergizer, J.H., Peric, M. 2002. Computational Methods for Fluid Dynamics, third ed. Springer, 58-60.

[46] Fiorina, C., Lathouwers, D., Aufiero, M., Cammi, A., Guerrieri, C., Kloosterman, J.L., Luzzi, L., Ricotti, M.E., 2014. Modeling and analysis of the MSFR transient behaviour. Ann. Nucl. Energy 64, 485-498.

[47] Fiorina, C., Aufiero, M., Cammi, A., Ricotti, M.E., 2013. Investigation of the MSFR core physics and fuel cycle characteristics. Prog. Nucl. Energy 68, 153-168.

[48] Frisch U., 1995. Turbulence: the legacy of A.N. Kolmogorov, Cambridge University Press.

[49] Geddes, K. 2021. ”The Ohmic Heating Reactor Experiment”, University of Idaho, Mechanical Engineering course work degree project report (unpublished manuscript).

[50] Generation IV International Forum, 2020. https://www.gen-4.org/gif/jcms/c_9260/public retrieved on 12 August 2021.

[51] Giraud, K.M., Parry, J.R., 2019. TREAT Fisssion Wire Physics Analysis, ECAR-4693, Revision 1, Idaho National Laboratory.

[52] Greaves, E.D., et al., 2012. The Case for the Thorium Molten Salt Reactor. AIP Conf. Proc. 1423 453(2012). https://doi.org/10.1063/1.3688845.

[53] Guifeng, Z., 2019. Low enriched uranium and thorium fuel utilization under once-through and offline reprocessing scenarios in small modular molten salt reactor, International Journal of Energy Research 43(2).

[54] Haag, Gerd, 2005. Properties of ATR-2E Graphite and properties changes due to fast neutron irradiation, Berichte des Firschungzentrums. Julich; 4183. Institut fur Sicherheitsforschung und Reacktortechnik Jul.

[55] Haubenreich, P.N., Engel J.R., Prince, B.E., Claiborne, H.C., 1964. MSRE Design and Operations Report Part III: Nuclear Analysis (ORNL-TM-730). Retrieved from U.S. Department of Energy Office of Scientific and Technical Information website: http://www.osti.gov/servlets/purl/4114686.

[56] Haubenreich, P.N., Engel, J.R., 1970. Experience with the Molten-Salt Reactor Experiment. Nucl. Technol. 8 (2), 118-136.

[57] Haubenreich, P.N., J.R. 1962. Temperatures in the MSRE during Steady-State Power Operation (ORNL-TM-378). Retrieved from U. S. Department of Energy Office of Scientific and Technical Information website: https://www.osti.gov/servlets/purl/ 4749259.

[58] He, X., 2016. Validation of the TRACE Code for the System Dynamic Simulations of the Molten Salt Reactor Experiment and the Preliminary Study on the Dual Fluid Molten Salt Reactor. Ph. D. Thesis Retrieved from. Technical University of Munich.

[59] Heuer, D., Merle-Lucotte, E., Allibert, M., Doligez, X., Ghetta, V., 2010. Simulation tools and new developments of the molten salt fast reactor. In: Contribution A0115, Proceedings of the European Nuclear Conference ENC2010, Barcelona, Spain.

[60] Holden, KaeCee M., 2018. A Quick Guide to MCNP. Lexinfton, KY.

[61] Hunt J., Phillips O., Williams D., 1991. Turbulence and Stochastic Processes: Kolmogorov's Ideas 50 Years On, Proc. Roy. Soc. London 434 1 - 240.

[62] International Atomic Energy Agency (IAEA), 2013. Challenges Related to the Use of Liquid Medal and Molten Salt Coolants in Advanced Reactors, IAEA-TECDOC-1696.

[63] IAEA, 2005. Natural circulation in water cooled nuclear power plants, IAEA-TECDOC-1474. . ISBN 92-0-110605-X. Image retrieved from https://www.nuclear-power.net/nuclear-engineering/heat-transfer/convection-convective-heat-transfer/natural-convection-free-convection/natural-circulation/ on 12 April 2021.

[64] International Thorium Molten-Salt Forum: ITMSF, Status Report - MSR-FUJI - IAEA ARIS Database, Advances in Small Modular Reactor Technology Developments 2016 Edition booklet, Retrieved from https://aris.iaea.org/PDF/MSR-FUJI.

[65] Janna, W., 1998. Design of Fluid Thermal Systems, 2nd Ed, PWS Publishing Co.

[66] Johnson Andrew, Kotlyar Dan, Terlizzi Stefano, and Ridley Gavin, 2020. serpentTools: A Python Package for Expediting Analysis with Serpent. Nuclear Science and Engineering (in press).

[67] Kerlin, T.W., Ball, S.J., 1966. Experimental Dynamic Analysis of the Molten Salt Reactor Experiment, ORNL-TM-1647, Oak Ridge National Laboratory.

[68] Kophazi, J., Lathouwers, D., Kloosterman, J.L., 2009. Development of a threedimensional time-dependent calculation scheme for molten salt reactors and validation of the measurement data of the molten salt reactor experiment. Nucl. Sci. Eng. 163 (2), 118-131.

[69] Krepel, J., 2014. Molten Salt Reactor: sustainable and safe reactor of the future?, Paul Scherr Institute, NES Colloquium.

[70] Krepel, J., Rohde, U., Grundmann, U., Weiss, F., 2007. DYN3D-MSR spatial dynamics code for molten salt reactors. Retrieved from Ann. Nucl. Energy 34, 449-462.

[71] Lamarsh, John R., Introduction to Nuclear Engineering, 3rd Edition, Addison-Wesley Company, 2001.

[72] LeBlanc, David, 2016. An Overview of the Integral Molten Salt Reactor, Presentation to ORNL MSR Workshop.

[73] LeBlanc, D. et al., 2020. Fuel Cycle Flexibility of Terrestrial Energy's Integral Molten Salt Reactor (IMSR), 38th Annual Conference of the Canadian Nuclear Society and 42nd Annual CNS/CNA Student Conference. Retrieved from https://www.terrestrialenergy.com/media/cns-conference-paper-fuel-cycle-flexibility-of-terrestrial-energys-imsr on 21 Aug 2020.

[74] LeBlanc, David, Terrestrial Energy, Inc., 2018, US Patent US20150036779A1, Retrieved from http://patents.google.com/patent/US20150036779A1/en on 21 Aug 2020.

[75] LeppÃ¤nen J., Pusa M., Viitanen T., Valtavirta V., and Kaltiaisenaho T., 2015. The Serpent Monte Carlo code: Status, development and applications in 2013. Ann. Nucl. Energy, 82 142-150.

[76] Lindsay, 2018. Alexander Introduction to Moltres - an Application for Simulation of Molten Salt Reactors. University of Illinois, Department of Nuclear, Plasma, and Radiological Engineering. Print.

[77] Lizin, A., 2013. UF4, ThF4 solubility in LiF-NaF-KF MELT, Atomic Energy, Vol. 115, No. 1 (Russian Original Vol. 115, No. 1).

[78] Lui, F. 2016. A thorough description of how wall functions are implemented in OpenFOAM. Chalmers University of Technology, Sweden.

[79] Lucas, Greg, 2010. MCNP Tutorial. UW - Engineering Physics Department: Spring - Print.

[80] Luzzi, L., et al., 2017. Thermal hydraulics of liquid-fueled MSRs, Chapter 6: Molten Salt Reactors and Thorium Energy, Woodhead Publishing Series in Energy. Cambridge, MA.

[81] Manohar S. Sohal, et al, 2013. Engineering Database of Liquid Salt Thermophysical and Thermochemical Properties, INL/EXT-10-18297 Rev.1.

[82] McCoy, H.E., et al., 2017. New developments in materials for molten salt reactors. Nucl. Appl. Technol. 8(1970) 156-169.

[83] Nayak, A.K., 1995. Mathematical modelling of the stability characteristics of a natural circulation loop, Math. Comput. Model. 22: 77-87.

[84] Nayak, A.K., et al, 2009. Thermal-hydraulic characteristics of a single-phase natural circulation loop with water and Al_2O_3 nanofluids, Nucl. Eng. Des. 239: 526-540.

[85] Pazsit I., et al, 2010a. Handbook of Nuclear Engineering, Vol. 3 Reactor Analysis, Ch 14: Noise Techniques in Nuclear Systems, Springer Publishing.

[86] Pazsit I., Dykin V., 2010b. Investigation of the space-dependent noise induced by propagating perturbations, Annals of Nuclear Energy 37 1329-1340.

[87] Pazsit I., et al, 2012. Analytical solutions of the molten salt reactor equations, Annals of Nuclear Energy 50 206-214.

[88] Pazsit I., Jonsson A., 2011. Reactor Kinetics, Dynamic Response, and Neutron Noise in Molten Salt Reactors, Nuclear Science and Engineering: 167, 61-76.

[89] Pazsit I., 1977. Investigation of the space-dependent noise induced by a vibrating absorber, Atomkernenergie (ATKE) Bd.30 Lfg.1.

[90] Pazsit I. et al., 2014. The point kinetic component of neutron noise in an MSR, Annals of Nuclear Energy 64 344-353.

[91] Pazsit, I., 2017. Kinetics and dynamics (including noise analysis) of Molten Salt Reactors, microsoft power point from Chalmers University of Technology.

[92] Pini A., et al, 2016. Analytical and numerical investigation of the heat exchange effect on the dynamic behaviour of natural circulation with internally heated fluids, Chem. Eng. Sci. 145: 108-125.

[93] Powers, J.J., Harrison, T.J., Gehin, J.C., 2013. A new approach for modeling and analysis of molten salt reactors using SCALE. In: Proc. Int. Conf. Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2013), Sun Valley, Idaho.

[94] Rahnema, F., et al., 2017. The Challenges in Modeling and Simulation of Fluoride-Salt-Cooled High Temperature Reactors, CRMP-2017-9-001, Georgia Institute of Technology.

[95] Richardson, L.F. 1910. The Approximate Arithmetical Solution by Finite Differences of Physical Problems involving Differential Equations, with an Application to the Stresses in a Masonry Dam. Philos. Trans. Roy. Soc. Lord Ser A210, 307-357.

[96] Robertson, R.C., 1965. MSRE Design and Operations Report Part I: Description of Reactor Design (ORNL-TM-728). Retrieved from U.S. Department of Energy Office of Scientific and Technical Information website: https://osti.gov/servlets/purl/4654707.

[97] Sabharwall, P., Yoo, Y., Wu Q., Sienicki, J., 2012. Natural Circulation and Linear Stability Analysis for Liquid-Metal Reactors with the Effect of Fluid Axial Conduction, Nuclear Technology, vol 178.

[98] Saha R., et al, 2018. Flow reversal prediction of a single-phase square natural circulation loop using symbolic time series analysis, International Conference on Mechanical Engineering held at Jadavpur University, Kolkata, India during January 4-6.

[99] Salim, M. Cheah, S. 2009. Wall y+ strategy for dealing with wall-bounded turbulent flows. Proceedings of the International Multiconference of Engineers and computer scientists Vol II IMECS during 18-19 March.

[100] SAMOFAR (Safety Assessment of the Molten Salt Fast Reactor), 2021. http://samofar.eu retrieved on 12 August 2021.

[101] SAMOSAFER, 2021. https://samosafer.eu retrieved on 12 August 2021.

[102] Serrano-Lopez R., Fradera J., Cuesta-Lopez S., 2013. Molten Salts Database for Energy Applications, Chemical Engineering and Processing: Process Intensification, Universidad de Burgos, Spain.

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

[104] Shi, C., Cheng, M., Liu, G., 2016. Development and application of a system analysis code for liquid fueled molten salt reactors based on RELAP5 code. Retrieved from Nucl. Eng. Des. 305, 378-388. https://ac.els-cdn.com/S0029549316301674/1-s2.

[105] Siemens Digital Industries Software, 2020, " Simcenter STAR-CCM+ Documentation version 2020.3, Product Overview and User Guide", www.siemens.com/mdx.

[106] Simon, C., 2020. Evaluation and Viability of Liquid Fuel Fast Reactor System, Centre National De La Recherche Scientifique CNRS. Retrieved from https://cordis.europa.eu/project/id/249696/reporting on 25 Aug 2020.

[107] Taube, M., 1974. A Molten Salt Fast Thermal Reactor System with no Waste, EIR Bericht Nr. 249, Eidg. Institut fÃ¼r Reaktorforschung WÃ¼renlingen Schweiz.

[108] Thoma, R.E., 1971. Chemical Aspects of MSRE Operations, ORNL-4658. Oak Ridge National Laboratory.

[109] Thoma, R.E., 1959. Phase Diagrams of Nuclear Reactor Materials, Oak Ridge National Laboratory ORNL-2548, TID-4500 (15th Edition).

[110] Tiberga, M., 2020. Results from a multi-physics numerical benchmark for codes dedicated to molten salt fast reactors, Annals of Nuclear Energy, Volume 142, 107428.

[111] Valtavirta V., Aufiero M., and LeppÃ¤nen J., 2019. A Novel Temperature Sensitivity Calculation Methodology for the Serpent Monte Carlo Code, VTT Technical Research Centre of Finland Ltd, P.O. Box 1000, FI-02044 VTT, FINLAND.

[112] Valtavirta V. and Leppanen J., 2017. Estimating the effects of homogenized fuel temperature in group constant generation using Serpent 2, Annals of Nuclear Energy, 105, 79-94.

[113] Valtavirta, V. 2018. Nuclear data uncertainty propagation to Serpent generated group and time constants, Research report VTT-R-04681-18.

[114] Valtavirta, V., Aufiero, M. and LeppÃ¤nen, J. 2018. Collision-history based sensitivity/perturbation calculation capabilities in Serpent 2.1.30, In Proc. BEPU 2018, Lucca, Italy.

[115] Viitanen T., 2015. Development of a stochastic temperature treatment technique for Monte Carlo neutron tracking, Ph.D. Thesis, Aalto University.

[116] Vijayan P.K., 2010. Joint ICTP-IAEA Course on Natural Circulation Phenomena and Passive Safety Systems in Advanced Water Cooled Reactors, International Center for Theoretical Physics, No. 2152-22.

[117] Watanabe N., et al, 2008. Thermal Hydraulic Flow Oscillation Characteristics in Multiformed Channels under Natural Circulation and Low-Pressure Conditions, Journal of Nuclear Science and Technology, 45:2, 160-170, DOI: 10.1080/18811248.2008.9711425.

[118] Welander P., 1967. On the oscillatory instability of a differentially heated fluid loop. J. Fluid Mech. 29(1).

[119] Williams, D.F., Toth, L.M., Clarno, K.T., 2006. Assessment of Candidate Molten Salt Coolants for the Advanced High-Temperature Reactor (AHTR), ORNL/TM-2006/12. Oak Ridge National Laboratory (ORNL).

[120] World Nuclear Association, 2019. Generation IV Nuclear Reactors, Tower House, 10 Southhampton Street, London, WC2E 7HA, United Kingdom.

[121] Xu, Z., Hejzlar, P., Driscoll, M.J., Kazimi, M.S., 2002. An Improved MCNP-ORIGEN Depletion Program (MCODE) and Its Verification for High Burnup Applications. PHYSOR, Seoul, Korea.

[122] Yoshioka, R., Mitachi, K., 2010. Controllability of FUJI-MSR International Thorium Molten-Salt Forum. Thorium Energy Conference.

[123] Zhou, J., Zhang, D., Qiu, S., Su, G., Tian, W., Wu, Y., 2015. Three dimensional neutronic/ thermal-hydraulic coupled simulation of MSR in transient state condition. Nucl. Eng. Des. 282, 93-105.

[124] Zhu, Y., Hawari, A.I., 2017. Thermal neutron scattering cross section of liquid FLiBe. Prog. Nucl. Energy.

[125] Zou L., et al, 2017. Numerical study on the Welander oscillatory natural circulation problem using high-order numerical methods, Progress in Nuclear Energy 94, 162-172.

Appendix A: Serpent Input Files

```
set title "FLiNaK-UF4 WHIP-Nuclear Battery-MicroVol -
Reflected configuration - stl WHIP eigenvalue calc rev 0"
%
% ________________Cell cards____________________
_______________________________
%
%   FUEL/Archimedes Screw Blades
%
/*  Core and Chimney WHIP rectilinear segments
 are commented out
cell 111 0 m10 14 -21 -30 39    % NaF-RbF in core- slice 1
%cell 112 0 m11 21 -22 -30    % NaF-RbF in core- slice 2
cell 112 0 ASgr 21 -22 -30 39   % AS Blade#1
cell 113 0 m12 22 -23 -30 39   % NaF-RbF in core- slice 3
%cell 114 0 m13 23 -24 -30    % NaF-RbF in core- slice 4
cell 114 0 ASgr 23 -24 -30 39   % AS Blade#2
cell 115 0 m14 24 -25 -30 39    % NaF-RbF in core- slice 5
%cell 116 0 m15 25 -26 -30    % NaF-RbF in core- slice 6
cell 116 0 ASgr 25 -26 -30 39   %AS Blade#3
cell 117 0 m16 26 -27 -30 39    % NaF-RbF in core- slice 7
%cell 118 0 m17 27 -28 -30    % NaF-RbF in core- slice 8
cell 118 0 ASgr 27 -1001 -30 39
%AS Blade#4 Graphite lower half
cell 130 0 ASBe 1001 -28 -30 39
```

```
%AS Blade#4 Be Refl upper half
cell 119 0 m18 28 -15 -30 39    % NaF-RbF in core- slice 9
cell 120 0 ASBe 15 -1000 -30 39   % AS BeO Blade#1
cell 121 0 m19 1000 -100 -30 39   % NaF-RbF in chimney
cell 122 0 ASBe 100 -101 -30 39   % AS BeO Blade#2
%3cell 123 0 m19 101 -102 -30 39
%3cell 124 0 ASBe 102 -103 -30 39
% AS BeO Blade#3
%3cell 125 0 m19 103 -104 -30 39
%3cell 126 0 ASBe 104 -105 -30 39
% AS BeO Blade#4
%3cell 127 0 m19 105 -106 -30 39
cell 127 0 m19 101 -106 -30 39
cell 128 0 m19 106 -19 -30 39
*/

% Chimney fuel cells - stem cell 39 removed
% cell 120 0 m19 15 -1000 -30
% cell 121 0 m19 1000 -100 -30
% cell 122 0 m19 100 -101 -30
% cell 127 0 m19 101 -106 -30
cell 128 0 m19 106 -19 -30

% Core Fuel cell
% cell 111 0 m12 14 -15 -30
 %single fuel volume in core
```

```
% Fuel in Inlet, Outlet, and Downcomer
cell 102 0 m1 12 -13 31 -36 41 42
 % Fuel in lower plenum
%cell 102 0 m1 12 -13 111 -36 41 42
 % Fuel in lower plenum
cell 103 0 m1 13 -16 35 -36 41 42
% Fuel around core
cell 104 0 m19 18 -19 -36 31 41 42 60 61
62 63 64 65 66 67 70 71 72 73 74 75 76 77
% Fuel in upper plenum
%cell 104 0 m19 18 -19 -36 111 41 42 60
61 62 63 64 65 66 67 70 71 72 73 74 75 76 77
% Fuel in upper plenum
cell 105 0 m14 16 -18 33 -36 41 42 60 61 62 63
64 65 66 67 70 71 72 73 74 75 76 77 % Fuel around chimney
%cell 106 0 m1 12 -13 -111 % Fuel in centre
of lower plenum
cell 106 0 m1 12 -13 -31 % Fuel in centre of
 lower plenum
% cell 107 0 ASgr  14 -15 -39 % AS center stem
 in core - Graphite
% cell 108 0 ASB4C 15 -19 -39 % AS center stem
 in chimney - Boron Carbide
%
%   STRUCTURE
%
cell 200 0 m2  11 -12 -36    % base
cell 201 0 m2  13 -14 -35    % core reflector lower
 clad / orifice plate
cell 202 0 Be  14 -19 30 -31 % core vessel-to-BeO
 shell inner cladding
%cell 209 0 Be  14 -19 110 -31 % core vessel BeO
 shell
%cell 210 0 m2  14 -19 31 -111 % core vessel-to-BeO
 shell outer cladding
cell 203 0 m2  14 -29 34 -35 % core reflector outer clad
cell 204 0 m2  29 -16 32 -35 60 61 62 63 64 65 66 67 70
 71 72 73 74 75 76 77 % chimney step
cell 205 0 m2  17 -18 31 -33 60 61 62 63 64 65 66 67 70
 71 72 73 74 75 76 77 % chimney top clad
%cell 205 0 m2  17 -18 111 -33 60 61 62 63 64 65 66 67
 70 71 72 73 74 75 76 77 % chimney top clad
cell 206 0 m2  16 -17 32 -33 60 61 62 63 64 65 66 67 70
 71 72 73 74 75 76 77 % chimney absorber clad
cell 207 0 m2  10 -19 36 -37 % RV cylinder
cell 208 0 m2  19 -20 -37    60 61 62 63 64 65 66 67 70
 71 72 73 74 75 76 77 % RV lid
%
%   REFLECTOR / ABSORBER
%
cell 300 0 Be 10 -11 -36     % reflector below core
cell 301 0 Be 14 -15 31 -38 60 61 62 63 64 65 66 67 70
 71 72 73 74 75 76 77 80 81 82 83 84 85 86 87 90 91
 92 93 94 95 96 97 % reflector around core
```

```
%cell 301 0 Be 14 -15 111 -38 60 61 62 63 64 65 66 67
70 71 72 73 74 75 76 77 80 81 82 83 84 85 86 87 90 91
 92 93 94 95 96 97 % reflector around core
cell 302 0 m4 29 -17 -32 31 60 61 62 63 64 65 66 67 70
 71 72 73 74 75 76 77  % absorber around chimney
%cell 302 0 m4 29 -17 -32 111 60 61 62 63 64 65 66 67
 70 71 72 73 74 75 76 77  % absorber around chimney
cell 303 0 m3 14 -15 38 -34  % safety reflector around
core reflector
cell 304 0 m4 15 -29 31 -34 60 61 62 63 64 65 66 67 70
 71 72 73 74 75 76 77  % upper (drumless) absorber
%cell 304 0 m4 15 -29 111 -34 60 61 62 63 64 65 66 67
 70 71 72 73 74 75 76 77  % upper (drumless) absorber
%
%   CONTROL DRUM REFLECTORS
%
cell 400 0 Be 14 -15 -90 60    % drum 1
cell 401 0 Be 14 -15 -91 61    % drum 2
cell 402 0 Be 14 -15 -92 62    % drum 3
cell 403 0 Be 14 -15 -93 63    % drum 4
cell 404 0 Be 14 -15 -94 64    % drum 5
cell 405 0 Be 14 -15 -95 65    % drum 6
cell 406 0 Be 14 -15 -96 66    % drum 7
cell 407 0 Be 14 -15 -97 67    % drum 8
%
%   CONTROL DRUM ABSORBERS
%
```

```
cell 500 0 m4 14 -15 -80 60    % drum 1
cell 501 0 m4 14 -15 -81 61    % drum 2
cell 502 0 m4 14 -15 -82 62    % drum 3
cell 503 0 m4 14 -15 -83 63    % drum 4
cell 504 0 m4 14 -15 -84 64    % drum 5
cell 505 0 m4 14 -15 -85 65    % drum 6
cell 506 0 m4 14 -15 -86 66    % drum 7
cell 507 0 m4 14 -15 -87 67    % drum 8
%
%   CONTROL ROD DRIVESHAFTS
%
cell 600 0 m2 -60 70 14 -20 % driveshaft tube 1
cell 601 0 m2 -61 71 14 -20 % driveshaft tube 2
cell 602 0 m2 -62 72 14 -20 % driveshaft tube 3
cell 603 0 m2 -63 73 14 -20 % driveshaft tube 4
cell 604 0 m2 -64 74 14 -20 % driveshaft tube 5
cell 605 0 m2 -65 75 14 -20 % driveshaft tube 6
cell 606 0 m2 -66 76 14 -20 % driveshaft tube 7
cell 607 0 m2 -67 77 14 -20 % driveshaft tube 8
cell 700 0 m20 -70 14 -20 % driveshaft tube 1 fill
cell 701 0 m20 -71 14 -20 % driveshaft tube 2 fill
cell 702 0 m20 -72 14 -20 % driveshaft tube 3 fill
cell 703 0 m20 -73 14 -20 % driveshaft tube 4 fill
cell 704 0 m20 -74 14 -20 % driveshaft tube 5 fill
cell 705 0 m20 -75 14 -20 % driveshaft tube 6 fill
cell 706 0 m20 -76 14 -20 % driveshaft tube 7 fill
cell 707 0 m20 -77 14 -20 % driveshaft tube 8 fill
```

```
%
% Flow separators
%
cell 800 0 m2 -41 12 -13 31 -36 #102
% lower plenum dividers 1
%cell 800 0 m2 -41 12 -13 111 -36 #102
% lower plenum dividers 1
cell 801 0 m2 -41 13 -16  35 -36 % core
 surround dividers 1
cell 802 0 m2 -41 16 -18 33 -36 % chimney
surround-RbFd dividers 1
%cell 803 0 m2 -41 18 -19 111 -36 #104 %
upper plenum dividers 1
cell 803 0 m2 -41 18 -19 31 -36 #104 % upper
 plenum dividers 1
%cell 804 0 m2 -42 12 -13 111 -36 #102 %
 lower plenum dividers 2
cell 804 0 m2 -42 12 -13 31 -36 #102 % lower
 plenum dividers 2
cell 805 0 m2 -42 13 -16  35 -36 % core
 surround-RbFd dividers 2
cell 806 0 m2 -42 16 -18 33 -36 % chimney
 surround-RbFd dividers 2
cell 807 0 m2 -42 18 -19 31 -36 #104 % upper
 plenum dividers 2
%cell 807 0 m2 -42 18 -19 111 -36 #104 % uppe
r plenum dividers 2
```

```
%
%   VOID
%
cell 900 0 outside -10       % void below
cell 901 0 outside 10 37 -20 % void around vessel
cell 902 0 outside 20        % void above

%
%  cells for stl WHIP
%
cell 111 0 fill 1 -111
cell 112 2 m14    -112
%
%  Create STL-based universe for WHIP moderator
%
solid 2 1 2  % universe=1, background universe=2
10 4 5 4 3 2        % mesh_splitting, mesh_dim, and SZ1
1 1E-5       % fast mode=1, two points joined into
 one if inside radius 0.01 cm
body whipmod whipmod ASgr  %body name,
 cell name, material
%file whipmod "WHIP6.stl" 1.00 0 0 -13   %body name,
 file path and name, scale, coord shift

file whipmod "WHIP6v1.stl" 1.00 0 0 -13   %trapezoidal flights

%
```

```
%  cells for stl WHIP reflector in CHIMNEY
%
cell 113 0 fill 3 -113
cell 114 4 m14 -114
%
%  Creat STL-based universe for WHIP REFLECTOR
%
solid 2 3 4
10 4 5 4 3 2
1 1E-5
body whiprefl whiprefl Be
%file whiprefl "WHIP6ch.stl" 1.00 0 0 -49
%file whiprefl "WHIP6.stl" 1.00 0 0 -49
file whiprefl "WHIP6v1.stl" 1.00 0 0 -49
trans U 3 rot 0 0 0 0 0 1 90

% _________________Surface cards_______________
________________________________
%
%  stl surfaces for WHIP core
%
%surf 111 cyl 0.0 0.0 25.0 13.0 179.0
surf 111 cyl 0.0 0.0 25.0 13.0 151.0
surf 112 inf
%
%  stl surfaces for WHIP reflector
%
```

```
%surf 113 cyl 0.0 0.0 25.0 179.0 216.0
surf 113 cyl 0.0 0.0 25.0 151.0 216.0
surf 114 inf
%
%  Reactor planes
%
surf 10 pz -11  % lower reflector base
surf 11 pz -1   % ground
surf 12 pz 0    % RV base inner (floor)  (FIX)
surf 13 pz 12   % reflector clad outer  (FIX)
surf 14 pz 13   % reflector clad inner
surf 15 pz 179  % core / chimney divider
surf 16 pz 180  % step clad upper  (FIX)
%3surf 17 pz 287  % chimney clad inner
%3surf 18 pz 288  % chimney clad outer  (FIX)
%3surf 19 pz 300  % top plate (RV) inner (FIX)
%3surf 20 pz 303  % top plate outer (ceiling)
surf 17 pz 215  % chimney clad inner
surf 18 pz 216  % chimney clad outer  (FIX)
surf 19 pz 228  % top plate (RV) inner (FIX)
surf 20 pz 231  % top plate outer (ceiling)
surf 29 pz 179  % step clad lower
%3surf 1000 pz 187
surf 1000 pz 187.77777
%
%  Fuel stratification
%
```

```
surf 21 pz 31.4444  %top of slice 1  %31.4444
surf 22 pz 49.8889  %top of slice 2  %49.8889
surf 23 pz 69.3333  %top of slice 3  %68.3333
surf 24 pz 86.7778  %top of slice 4  %86.7778
surf 25 pz 105.2222  %top of slice 5  %105.2222
surf 26 pz 123.6667  %top of slice 6  %123.6667
surf 27 pz 142.1111  %top of slice 7  %142.1111
surf 1001 pz 151.3334 % divider in refl/abs blade
surf 28 pz 160.5556  %top of slice 8
%3surf 100 pz 207 %top of 1st Chimney Blade
surf 100 pz 206.22222 %top of 1st Chimney Blade
surf 101 pz 215  %top of 2nd BeO chimney blad
%3surf 102 pz 235 %btm of 3rd chimney Blade
%3surf 103 pz 243 %btm of 3rd chimney Blade
%3surf 104 pz 263 %top of 3rd chimney Blade
%3surf 105 pz 271 %btm of 4th chimney Blade
surf 106 pz 216 %chimney clad upper
%
%  Reactor cylinders
%
surf 30 cyl 0 0 25   % core vessel inner (FIX) 25
%surf 110 cyl 0 0 25.1  %core vessel cladding
surf 31 cyl 0 0 26.5 % core vessel liner 25.1
%surf 111 cyl 0 0 26.8  %core vessel reflector
cladding
surf 32 cyl 0 0 47.55   % chimney reflector outer
surf 33 cyl 0 0 48.55   % absorber liner (FIX)
surf 34 cyl 0 0 66   % safety absorber outer
surf 35 cyl 0 0 67   % reflector liner (FIX)
%3surf 36 cyl 0 0 75   % RV inner (FIX)
%3surf 37 cyl 0 0 76   % RV outer
surf 36 cyl 0 0 69.5   % RV inner (FIX)
surf 37 cyl 0 0 72.5   % RV outer
surf 38 cyl 0 0 63.5   % reflector outer
surf 39 cyl 0 0 2.0  % AS center stem
%
% Control drum absorber pads, numbered clockwise
 from 9 o'clock
%
surf 80 pad -45.05 0 2.5 17 120 240
 % drum 1
surf 81 pad -31.8551605 31.8551605 2.5
 17 165 285 % drum 2
surf 82 pad 0 45.05 2.5 17 210 330
  % drum 3
surf 83 pad 31.8551605 31.8551605 2.5 17
 255 15   % drum 4
surf 84 pad 45.05 0 2.5 17 300 60
 % drum 5
surf 85 pad 31.8551605 -31.8551605 2.5 17
345 105 % drum 6
surf 86 pad 0 -45.05 2.5 17 30 150
  % drum 7
surf 87 pad -31.8551605 -31.8551605 2.5 17
```

```
 75 195 % drum 8
%
% Control drum reflector pads, likewise
%
surf 90 pad -45.05 0 2.5 17 240 120
   % drum 1
surf 91 pad -31.8551605 31.8551605 2.5 17
285 165 % drum 2
surf 92 pad 0 45.05 2.5 17 330 210
   % drum 3
surf 93 pad 31.8551605 31.8551605 2.5 17
 15 255   % drum 4
surf 94 pad 45.05 0 2.5 17 60 300
  % drum 5
surf 95 pad 31.8551605 -31.8551605 2.5 17
 105 345 % drum 6
surf 96 pad 0 -45.05 2.5 17 150 30
    % drum 7
surf 97 pad -31.8551605 -31.8551605 2.5
17 195 75 % drum 8
%
% Control drum driveshafts
%
surf 60 cyl -45.05 0 2.5                % drum
 drive tube outer 1
surf 61 cyl -31.8551605 31.8551605 2.5
 % drum drive tube outer 2
```

```
surf 62 cyl 0 45.05 2.5                 % drum
 drive tube outer 3
surf 63 cyl 31.8551605 31.8551605 2.5
 % drum drive tube outer 4
surf 64 cyl 45.05 0 2.5                 % drum
 drive tube outer 5
surf 65 cyl 31.8551605 -31.8551605 2.5
% drum drive tube outer 6
surf 66 cyl 0 -45.05 2.5                % drum
 drive tube outer 7
surf 67 cyl -31.8551605 -31.8551605 2.5
% drum drive tube outer 8
surf 70 cyl -45.05 0 2                 % drum
 drive tube inner 1
surf 71 cyl -31.8551605 31.8551605 2
% drum drive tube inner 2
surf 72 cyl 0 45.05 2                  % drum
 drive tube inner 3
surf 73 cyl 31.8551605 31.8551605 2
% drum drive tube inner 4
surf 74 cyl 45.05 0 2                  % drum
 drive tube inner 5
surf 75 cyl 31.8551605 -31.8551605 2
 % drum drive tube inner 6
surf 76 cyl 0 -45.05 2                 % drum
 drive tube inner 7
surf 77 cyl -31.8551605 -31.8551605 2
```

```
 % drum drive tube inner 8
%
% Flow separators
%
surf 41 cross 0 0 75.5 3
strans 41 0 0 1 0 0 22.5
surf 42 cross 0 0 75.5 3
strans 42 0 0 1 0 0 -22.5
%
% _________________Material cards____
__________________________________________
%
% LiF-NaF-KF-UF4
mat m1 -3.643499  tmp 873 %burn 1 rgb 123
 173 0
%mat m1 -3.643499  tmp 873 rgb 123 173 0
%92235.06c  2.48750e-2            % NaF-RbF-UF4
  ********************
%92238.06c  1.00125e-1
%11023.06c  1.37500e-1
%37085.06c  3.60850e-2
%37087.06c  1.39150e-2
%9019.06c   6.87500e-1

%11023.06c  2.35294e-1    % NaCl-PuCl3 3%
 enriched (64-36 salt)
%17035.06c  4.79071e-1
```

```
%17037.06c  1.53282e-1
%94239.06c  3.97059e-3
%94238.06c  1.28382e-1

%3006.06c  1.57173e-5     % LiF-NaF-KF-PuF4
%3007.06c  0.19645059
%9019.06c  0.460525284
%19039.06c  0.165489528
%19041.06c  0.119426164
%11023.06c  0.048588442
%94239.06c  0.009504274

%3006.06c  9.0398707e-6     % LiF-NaF-KF-UF4
21% UF4
%3007.06c  1.1298934e-1
%9019.06c  6.3058851e-1
%19039.06c  9.5181965e-2
%19041.06c  6.8688437e-2
%11023.06c  2.7945837e-2
%92235.06c  1.2757882e-2
%92238.06c  5.1838989e-2

3006.06c  1.49954e-5     % LiF-NaF-KF-UF4
18% UF4
3007.06c  1.49939e-1
9019.06c  6.06524e-1
19039.06c  1.26311e-1
```

```
19041.06c  9.11554e-3
11023.06c  3.70854e-2
92235.06c  1.40246e-2
92238.06c  5.69859e-2

%
%mat m10 -3.635826  tmp 881.33 %burn 1
%vol 2.356E+5 rgb 127 181 0
mat m10 -3.635826  tmp 881.33
%vol 2.356E+5 rgb 127 181 0
%92235.06c  2.48750e-2
% NaF-RbF-UF4  *********************
%92238.06c  1.00125e-1
%11023.06c  1.37500e-1
%37085.06c  3.60850e-2
%37087.06c  1.39150e-2
%9019.06c   6.87500e-1

%11023.06c  2.35294e-1
% NaCl-PuCl3 3% enriched (64-36 salt)
%17035.06c  4.79071e-1
%17037.06c  1.53282e-1
%94239.06c  3.97059e-3
%94238.06c  1.28382e-1

%3006.06c  1.57173e-5     % LiF-NaF-KF-PuF4
%3007.06c  0.19645059
```

```
%9019.06c  0.460525284
%19039.06c  0.165489528
%19041.06c  0.119426164
%11023.06c  0.048588442
%94239.06c  0.009504274

%3006.06c  9.0398707e-6
% LiF-NaF-KF-UF4 21% UF4
%3007.06c  1.1298934e-1
%9019.06c  6.3058851e-1
%19039.06c  9.5181965e-2
%19041.06c  6.8688437e-2
%11023.06c  2.7945837e-2
%92235.06c  1.2757882e-2
%92238.06c  5.1838989e-2

3006.06c  1.49954e-5
% LiF-NaF-KF-UF4 18% UF4
3007.06c  1.49939e-1
9019.06c  6.06524e-1
19039.06c  1.26311e-1
19041.06c  9.11554e-3
11023.06c  3.70854e-2
92235.06c  1.40246e-2
92238.06c  5.69859e-2
%
mat m11 -3.628150  tmp 889.66
```

```
%burn 1 %vol 2.356E+5 rgb 132 185 0

%mat m11 -3.628150  tmp 889.66

%vol 2.356E+5 rgb 132 185 0

%92235.06c  2.48750e-2

% NaF-RbF-UF4  *********************

%92238.06c  1.00125e-1

%11023.06c  1.37500e-1

%37085.06c  3.60850e-2

%37087.06c  1.39150e-2

%9019.06c   6.87500e-1


%11023.06c  2.35294e-1

% NaCl-PuCl3 3% enriched (64-36 salt)

%17035.06c  4.79071e-1

%17037.06c  1.53282e-1

%94239.06c  3.97059e-3

%94238.06c  1.28382e-1


%3006.06c  1.57173e-5     % LiF-NaF-KF-PuF4

%3007.06c  0.19645059

%9019.06c  0.460525284

%19039.06c  0.165489528

%19041.06c  0.119426164

%11023.06c  0.048588442

%94239.06c  0.009504274


%3006.06c  9.0398707e-6
```

```
% LiF-NaF-KF-UF4 21% UF4
%3007.06c  1.1298934e-1
%9019.06c  6.3058851e-1
%19039.06c  9.5181965e-2
%19041.06c  6.8688437e-2
%11023.06c  2.7945837e-2
%92235.06c  1.2757882e-2
%92238.06c  5.1838989e-2

3006.06c  1.49954e-5
% LiF-NaF-KF-UF4 18% UF4
3007.06c  1.49939e-1
9019.06c  6.06524e-1
19039.06c  1.26311e-1
19041.06c  9.11554e-3
11023.06c  3.70854e-2
92235.06c  1.40246e-2
92238.06c  5.69859e-2
%
mat m12 -3.620470  tmp 897.99
%burn 1 %vol 2.356E+5 rgb 137 196 0
%mat m12 -3.620470  tmp 897.99
%vol 2.356E+5 rgb 137 196 0
%92235.06c  2.48750e-2
% NaF-RbF-UF4  *********************
%92238.06c  1.00125e-1
%11023.06c  1.37500e-1
```

```
%37085.06c  3.60850e-2

%37087.06c  1.39150e-2

%9019.06c   6.87500e-1


%11023.06c  2.35294e-1

% NaCl-PuCl3 3% enriched (64-36 salt)

%17035.06c  4.79071e-1

%17037.06c  1.53282e-1

%94239.06c  3.97059e-3

%94238.06c  1.28382e-1


%3006.06c  1.57173e-5     % LiF-NaF-KF-PuF4

%3007.06c  0.19645059

%9019.06c  0.460525284

%19039.06c  0.165489528

%19041.06c  0.119426164

%11023.06c  0.048588442

%94239.06c  0.009504274


%3006.06c  9.0398707e-6

% LiF-NaF-KF-UF4 21% UF4

%3007.06c  1.1298934e-1

%9019.06c  6.3058851e-1

%19039.06c  9.5181965e-2

%19041.06c  6.8688437e-2

%11023.06c  2.7945837e-2

%92235.06c  1.2757882e-2
```

```
%92238.06c  5.1838989e-2

3006.06c  1.49954e-5     % LiF-NaF-KF-UF4 18% UF4
3007.06c  1.49939e-1
9019.06c  6.06524e-1
19039.06c  1.26311e-1
19041.06c  9.11554e-3
11023.06c  3.70854e-2
92235.06c  1.40246e-2
92238.06c  5.69859e-2
%
%mat m13 -3.612788  tmp 906.32
%burn 1 %vol 2.356E+5 rgb 143 201 0
mat m13 -3.612788  tmp 906.32
%vol 2.356E+5 rgb 143 201 0
%92235.06c  2.48750e-2
% NaF-RbF-UF4  *********************
%92238.06c  1.00125e-1
%11023.06c  1.37500e-1
%37085.06c  3.60850e-2
%37087.06c  1.39150e-2
%9019.06c   6.87500e-1

%11023.06c  2.35294e-1
% NaCl-PuCl3 3% enriched (64-36 salt)
%17035.06c  4.79071e-1
%17037.06c  1.53282e-1
```

```
%94239.06c  3.97059e-3
%94238.06c  1.28382e-1

%3006.06c  1.57173e-5     % LiF-NaF-KF-PuF4
%3007.06c  0.19645059
%9019.06c  0.460525284
%19039.06c  0.165489528
%19041.06c  0.119426164
%11023.06c  0.048588442
%94239.06c  0.009504274

%3006.06c  9.0398707e-6
% LiF-NaF-KF-UF4 21% UF4
%3007.06c  1.1298934e-1
%9019.06c  6.3058851e-1
%19039.06c  9.5181965e-2
%19041.06c  6.8688437e-2
%11023.06c  2.7945837e-2
%92235.06c  1.2757882e-2
%92238.06c  5.1838989e-2

3006.06c  1.49954e-5
% LiF-NaF-KF-UF4 18% UF4
3007.06c  1.49939e-1
9019.06c  6.06524e-1
19039.06c  1.26311e-1
19041.06c  9.11554e-3
```

```
11023.06c  3.70854e-2
92235.06c  1.40246e-2
92238.06c  5.69859e-2
%
mat m14 -3.605102  tmp 914.65
%burn 1 %vol 2.356E+5 rgb 160 225 0
%mat m14 -3.605102  tmp 914.65
%vol 2.356E+5 rgb 160 225 0
%92235.06c  2.48750e-2
% NaF-RbF-UF4  *********************
%92238.06c  1.00125e-1
%11023.06c  1.37500e-1
%37085.06c  3.60850e-2
%37087.06c  1.39150e-2
%9019.06c   6.87500e-1

%11023.06c  2.35294e-1
% NaCl-PuCl3 3% enriched (64-36 salt)
%17035.06c  4.79071e-1
%17037.06c  1.53282e-1
%94239.06c  3.97059e-3
%94238.06c  1.28382e-1

%3006.06c  1.57173e-5     % LiF-NaF-KF-PuF4
%3007.06c  0.19645059
%9019.06c  0.460525284
%19039.06c  0.165489528
```

```
%19041.06c  0.119426164
%11023.06c  0.048588442
%94239.06c  0.009504274

%3006.06c  9.0398707e-6
% LiF-NaF-KF-UF4 21% UF4
%3007.06c  1.1298934e-1
%9019.06c  6.3058851e-1
%19039.06c  9.5181965e-2
%19041.06c  6.8688437e-2
%11023.06c  2.7945837e-2
%92235.06c  1.2757882e-2
%92238.06c  5.1838989e-2

3006.06c  1.49954e-5
% LiF-NaF-KF-UF4 18% UF4
3007.06c  1.49939e-1
9019.06c  6.06524e-1
19039.06c  1.26311e-1
19041.06c  9.11554e-3
11023.06c  3.70854e-2
92235.06c  1.40246e-2
92238.06c  5.69859e-2
%
mat m15 -3.597414  tmp 922.98
%burn 1 %vol 2.356E+5 rgb 182 255 0
%mat m15 -3.597414  tmp 922.98
```

```
%vol 2.356E+5 rgb 182 255 0
%92235.06c  2.48750e-2
% NaF-RbF-UF4  *********************
%92238.06c  1.00125e-1
%11023.06c  1.37500e-1
%37085.06c  3.60850e-2
%37087.06c  1.39150e-2
%9019.06c   6.87500e-1

%11023.06c  2.35294e-1
% NaCl-PuCl3 3% enriched (64-36 salt)
%17035.06c  4.79071e-1
%17037.06c  1.53282e-1
%94239.06c  3.97059e-3
%94238.06c  1.28382e-1

%3006.06c  1.57173e-5     % LiF-NaF-KF-PuF4
%3007.06c  0.19645059
%9019.06c  0.460525284
%19039.06c  0.165489528
%19041.06c  0.119426164
%11023.06c  0.048588442
%94239.06c  0.009504274

%3006.06c  9.0398707e-6
% LiF-NaF-KF-UF4 21% UF4
%3007.06c  1.1298934e-1
```

```
%9019.06c  6.3058851e-1
%19039.06c  9.5181965e-2
%19041.06c  6.8688437e-2
%11023.06c  2.7945837e-2
%92235.06c  1.2757882e-2
%92238.06c  5.1838989e-2

3006.06c  1.49954e-5
% LiF-NaF-KF-UF4 18% UF4
3007.06c  1.49939e-1
9019.06c  6.06524e-1
19039.06c  1.26311e-1
19041.06c  9.11554e-3
11023.06c  3.70854e-2
92235.06c  1.40246e-2
92238.06c  5.69859e-2
%
mat m16 -3.589722  tmp 931.31
%burn 1 %vol 2.356E+5 rgb 196 255 0
%mat m16 -3.589722  tmp 931.31
 %vol 2.356E+5 rgb 196 255 0
%92235.06c  2.48750e-2
 % NaF-RbF-UF4  *********************
%92238.06c  1.00125e-1
%11023.06c  1.37500e-1
%37085.06c  3.60850e-2
%37087.06c  1.39150e-2
```

```
%9019.06c   6.87500e-1

%11023.06c  2.35294e-1
% NaCl-PuCl3 3% enriched (64-36 salt)
%17035.06c  4.79071e-1
%17037.06c  1.53282e-1
%94239.06c  3.97059e-3
%94238.06c  1.28382e-1

%3006.06c  1.57173e-5     % LiF-NaF-KF-PuF4
%3007.06c  0.19645059
%9019.06c  0.460525284
%19039.06c  0.165489528
%19041.06c  0.119426164
%11023.06c  0.048588442
%94239.06c  0.009504274

%3006.06c  9.0398707e-6
% LiF-NaF-KF-UF4 21% UF4
%3007.06c  1.1298934e-1
%9019.06c  6.3058851e-1
%19039.06c  9.5181965e-2
%19041.06c  6.8688437e-2
%11023.06c  2.7945837e-2
%92235.06c  1.2757882e-2
%92238.06c  5.1838989e-2
```

```
3006.06c  1.49954e-5
% LiF-NaF-KF-UF4 18% UF4
3007.06c  1.49939e-1
9019.06c  6.06524e-1
19039.06c  1.26311e-1
19041.06c  9.11554e-3
11023.06c  3.70854e-2
92235.06c  1.40246e-2
92238.06c  5.69859e-2
%
mat m17 -3.582028  tmp 939.64
%burn 1 %vol 2.356E+5 rgb 207 255 91
%mat m17 -3.582028  tmp 939.64
%vol 2.356E+5 rgb 207 255 91
%92235.06c  2.48750e-2
% NaF-RbF-UF4  *********************
%92238.06c  1.00125e-1
%11023.06c  1.37500e-1
%37085.06c  3.60850e-2
%37087.06c  1.39150e-2
%9019.06c   6.87500e-1

%11023.06c  2.35294e-1
% NaCl-PuCl3 3% enriched (64-36 salt)
%17035.06c  4.79071e-1
%17037.06c  1.53282e-1
%94239.06c  3.97059e-3
```

```
%94238.06c  1.28382e-1

%3006.06c  1.57173e-5
% LiF-NaF-KF-PuF4
%3007.06c  0.19645059
%9019.06c  0.460525284
%19039.06c  0.165489528
%19041.06c  0.119426164
%11023.06c  0.048588442
%94239.06c  0.009504274

%3006.06c  9.0398707e-6
% LiF-NaF-KF-UF4 21% UF4
%3007.06c  1.1298934e-1
%9019.06c  6.3058851e-1
%19039.06c  9.5181965e-2
%19041.06c  6.8688437e-2
%11023.06c  2.7945837e-2
%92235.06c  1.2757882e-2
%92238.06c  5.1838989e-2

3006.06c  1.49954e-5
% LiF-NaF-KF-UF4 18% UF4
3007.06c  1.49939e-1
9019.06c  6.06524e-1
19039.06c  1.26311e-1
19041.06c  9.11554e-3
```

```
11023.06c  3.70854e-2
92235.06c  1.40246e-2
92238.06c  5.69859e-2
%
mat m18 -3.574330  tmp 947.97
%burn 1 %vol 2.356E+5 rgb 218 255 127
%mat m18 -3.574330  tmp 947.97
 %vol 2.356E+5 rgb 218 255 127
%92235.06c  2.48750e-2
 % NaF-RbF-UF4  *********************
%92238.06c  1.00125e-1
%11023.06c  1.37500e-1
%37085.06c  3.60850e-2
%37087.06c  1.39150e-2
%9019.06c   6.87500e-1

%11023.06c  2.35294e-1
% NaCl-PuCl3 3% enriched (64-36 salt)
%17035.06c  4.79071e-1
%17037.06c  1.53282e-1
%94239.06c  3.97059e-3
%94238.06c  1.28382e-1

%3006.06c  1.57173e-5
% LiF-NaF-KF-PuF4
%3007.06c  0.19645059
%9019.06c  0.460525284
```

```
%19039.06c  0.165489528
%19041.06c  0.119426164
%11023.06c  0.048588442
%94239.06c  0.009504274

%3006.06c  9.0398707e-6
 % LiF-NaF-KF-UF4 21% UF4
%3007.06c  1.1298934e-1
%9019.06c  6.3058851e-1
%19039.06c  9.5181965e-2
%19041.06c  6.8688437e-2
%11023.06c  2.7945837e-2
%92235.06c  1.2757882e-2
%92238.06c  5.1838989e-2

3006.06c  1.49954e-5
% LiF-NaF-KF-UF4 18% UF4
3007.06c  1.49939e-1
9019.06c  6.06524e-1
19039.06c  1.26311e-1
19041.06c  9.11554e-3
11023.06c  3.70854e-2
92235.06c  1.40246e-2
92238.06c  5.69859e-2
%
mat m19 -3.566629  tmp 956.3
%burn 1 %rgb 227 255 152
```

```
%mat m19 -3.566629  tmp 956.3
 %rgb 227 255 152
%92235.06c  2.48750e-2
% NaF-RbF-UF4  *********************
%92238.06c  1.00125e-1
%11023.06c  1.37500e-1
%37085.06c  3.60850e-2
%37087.06c  1.39150e-2
%9019.06c   6.87500e-1

%11023.06c  2.35294e-1
% NaCl-PuCl3 3% enriched (64-36 salt)
%17035.06c  4.79071e-1
%17037.06c  1.53282e-1
%94239.06c  3.97059e-3
%94238.06c  1.28382e-1

%3006.06c  1.57173e-5
% LiF-NaF-KF-PuF4
%3007.06c  0.19645059
%9019.06c  0.460525284
%19039.06c  0.165489528
%19041.06c  0.119426164
%11023.06c  0.048588442
%94239.06c  0.009504274

%3006.06c  9.0398707e-6
```

```
 % LiF-NaF-KF-UF4 21% UF4
%3007.06c  1.1298934e-1
%9019.06c  6.3058851e-1
%19039.06c  9.5181965e-2
%19041.06c  6.8688437e-2
%11023.06c  2.7945837e-2
%92235.06c  1.2757882e-2
%92238.06c  5.1838989e-2

3006.06c  1.49954e-5
% LiF-NaF-KF-UF4 18% UF4
3007.06c  1.49939e-1
9019.06c  6.06524e-1
19039.06c  1.26311e-1
19041.06c  9.11554e-3
11023.06c  3.70854e-2
92235.06c  1.40246e-2
92238.06c  5.69859e-2
%
% 304 Stainless Steel
%
mat m2 -7.5983  tmp 923  rgb 127 255 255
26054.06c -0.037589 %54-Fe
26056.06c -0.611384 %56-Fe
26057.06c -0.014379 %57-Fe
26058.06c -0.001932 %58-Fe
7014.06c  -0.000995 %14-N
```

```
7015.06c  -0.000004 %15-N
28058.06c -0.070485 %58-Ni
28060.06c -0.028087 %60-Ni
28061.06c -0.001241 %61-Ni
28062.06c -0.004025 %62-Ni
28064.06c -0.001058 %64-Ni
24050.06c -0.008338 %50-Cr
24052.06c -0.167231 %52-Cr
24053.06c -0.019327 %53-Cr
24054.06c -0.004902 %54-Cr
14028.06c -0.006881 %28-Si
14029.06c -0.000363 %29-Si
14030.06c -0.000248 %30-Si
16032.06c -0.000284 %32-S
16033.06c -0.000002 %33-S
16034.06c -0.000014 %34-S
15031.06c -0.000020 %31-P
25055.06c -0.019980 %55-Mn
6000.06c  -0.000799 %(Carbon)

mat m2a -7.5983 tmp 923
%mat used for heating det dr 301
26056.06c  -1.0

mat act304ss -7.5983  tmp 923
%burn 1 vol 1 rgb 127 255 255
26054.06c -0.037589 %54-Fe
```

```
26056.06c -0.611384 %56-Fe
26057.06c -0.014379 %57-Fe
26058.06c -0.001932 %58-Fe
27059.06c -0.001 %Co-60
7014.06c  -0.000995 %14-N
7015.06c  -0.000004 %15-N
28058.06c -0.070485 %58-Ni
28060.06c -0.028087 %60-Ni
28061.06c -0.001241 %61-Ni
28062.06c -0.004025 %62-Ni
28064.06c -0.001058 %64-Ni
24050.06c -0.008338 %50-Cr
24052.06c -0.167231 %52-Cr
24053.06c -0.019327 %53-Cr
24054.06c -0.004902 %54-Cr
14028.06c -0.006881 %28-Si
14029.06c -0.000363 %29-Si
14030.06c -0.000248 %30-Si
16032.06c -0.000284 %32-S
16033.06c -0.000002 %33-S
16034.06c -0.000014 %34-S
15031.06c -0.000020 %31-P
25055.06c -0.019980 %55-Mn
6000.06c  -0.000799 %(Carbon)
%92235.06c -0.01
%
% Graphite
```

```
%
mat m3 -1.8650  tmp 923 moder
 grph 6000 rgb 160 160 160
6000.06c -1 %(Carbon)
therm grph 1010 gre7.00t gre7.26t
%therm grph 1010 grph.03t grph.20t

mat ASgr -1.8650  tmp 923 moder
 grph 6000 rgb 160 160 160
6000.06c -1 %(Carbon)
%therm grph 1010 gre7.00t gre7.26t
%therm grph 1010 grph.03t grph.20t
%
% Boron Carbide (10-B enriched)
%
mat m4 -2.52  tmp 923  rgb 48 48 48
5010.06c 0.8 %10-B
6000.06c 0.2 %(Carbon)

mat ASB4C -2.52  tmp 923  rgb 48 48 48
5010.06c 0.8 %10-B
6000.06c 0.2 %(Carbon)
%
% "Air"
%
mat m5 -1.225  tmp 923 rgb 255 233 127
18036.06c 0.021
```

```
18038.06c 0.043
18040.06c 0.936 %Argon
%
% Beryllium Oxide Reflector
mat Be -3.01 tmp 1010
4009.06c  0.5
8016.06c  0.5

mat ASBe -3.01 tmp 1010
4009.06c  0.5
8016.06c  0.5
% Detector material

mat m20 -8.89 tmp 956  rgb 235 52 52
 % Hastelloy CHIMNEY REFLECTOR
28058.06c -0.48
24052.06c -0.07
26056.06c -0.05
14028.06c -0.01
28060.06c -0.23
42092.06c -0.02344
42094.06c -0.014704
42095.06c -0.025392
42096.06c -0.026672
42097.06c -0.015328
42098.06c -0.038864
42100.06c -0.0156
```

```
%
mat m6 -19.2  tmp 956 % 235-U
92235.06c -1
% mat m6 -0.059 % Helium-3
% 2003.06c -1

/*
mat MgO -3.58 tmp 1010
12024.06c 0.78990
12025.06c 0.1
12026.06c 0.1101
8016.06c  1.0
*/

% _______________________Detectors___
__________________________________________
%
% ene 1 1 1e-11 0.625e-6
% ene 2 1 0.625e-6 2e1

ene 1 1 1e-11 0.625e-6
ene 2 1 0.625e-6 0.2
ene 3 1 0.2 2e1

% ene 1 1 1e-11 2e1
% ene GammaEspect 3 500 1e-11 2e1
% Neutron inverse speed 6.21283E-07

det meshth n dr -15 void
%dv 3.996145e6
de 1
dx -72.5 72.5 70 dy -72.5 72.5 70 dz -11 231 70

det meshepi n dr -15 void
%dv 3.996145e6
de 2
dx -72.5 72.5 70 dy -72.5 72.5 70 dz -11 231 70

det meshf n dr -15 void
%dv 3.996145e6
de 3
dx -72.5 72.5 70 dy -72.5 72.5 70 dz -11 231 70

%det axialth           n de 1 dr -15 m2 dn 1 0 25
 1 0 360 1 -11 231 242
%det axialf            n de 2 dr -15 m2 dn 1 0 25
 1 0 360 1 -11 231 242
%det gammaax           p dr -15 void dn 1 75 76
 1 0 360 1 -11 231 242 dv 1.1480008e5
%det gammarad          p dr -15 void dn 1  0 76
 76 0 360 1 228 231 1 dv 5.44375e4
%det act304            n de 1 da act304ss -1 dv 1
%det TotPhoton          n de 1 dr -5 void dn 1 0 76
```

```
 1 0 360 1 -11 231 242 dv 1e6
%det PhotonSrcRate      p dr -11 void dn 1 0 76
 1 0 360 1 -11 231 242 dv 1e6
%det gammaEspecFuel      n dr -5 m10 de GammaEspect
 dn 1 0 76 1 0 360 1 -11 231 1 dv 1e6
%det gammaEspec304SS      n dr -5 m2 de GammaEspect
 dn 1 0 76 1 0 360 1 -11 231 1 dv 1e6

%det HastNth            n de 1 dc 700 dr -15 m20
dv 2.86513e3
%det HastNf             n de 2 dc 700 dr -15 m20
 dv 2.86513e3

/*
det RVBaseEnDepG12        p  de 1 dc 200 dr -12 void
dv 1.51747e4
det RVWallEnDepG12        p  de 1 dc 207 dr -12 void
 dv 3.198581e5
det RVTopEnDepG12         p  de 1 dc 208 dr -12 void
dv 4.953899e4
det RVBaseEnDepG301        p  de 1 dc 200 dr 301
m2a dv 1.51747e4
det RVWallEnDepG301        p  de 1 dc 207 dr 301
 m2a dv 3.198581e5
det RVTopEnDepG301         p  de 1 dc 208 dr 301
m2a dv 4.953899e4
det RVBaseEnDepN301        n  de 1 dc 200 dr 301
```

```
m2a dv 1.51747e4
det RVWallEnDepN301        n  de 1 dc 207 dr 301
m2a dv 3.198581e5
det RVTopEnDepN301         n  de 1 dc 208 dr 301
 m2a dv 4.953899e4
det RVBaseEnDepN80        n  de 1 dc 200 dr -80
void dv 1.51747e4
det RVWallEnDepN80        n  de 1 dc 207 dr -80
void dv 3.198581e5
det RVTopEnDepN80         n  de 1 dc 208 dr -80
void dv 4.953899e4
*/

/*
det BaseN          n  de 1  dc 200 dr -15 m2 dv 1.51747e4
det BaseNf         n  de 2 dc 200 dr -15 m2 dv 1.51747e4
det BaseGamma      p  dc 200 dr -15 m2 %dv 1.51747e4

det CoreVesselN    n  de 1 dc 202 dr -15 m2 dv 1.19619e3
det CoreVesselNf   n  de 2 dc 202 dr -15 m2 dv 1.19619e3

det RVN            n  de 1 dc 207 dr -15 m2 dv 3.198581e5
det RVNf           n  de 2 dc 207 dr -15 m2 dv 3.198581e5
det RVGamma        p  dc 207 dr -15 m2 %dv 3.198581e5

det RVTopN         n  de 1 dc 208 dr -15 m2 dv 4.953899e4
det RVTopNf        n  de 2 dc 208 dr -15 m2 dv 4.953899e4
det RVTopGamma      p  dc 208 dr -15 m2  %dv 4.953899e4

det BeCoreBeltN     n  de 1 dc 301 dr -15 Be dv 5.6856487e5
det B4CBeltN        n  de 1 dc 302 dr -15 m4 dv 1.7880604e5

det OutsdRVBtmG    p  dc 900 dr -15 void
det OutsdRVCLG     p  dc 901 dr -15 void
det OutsdRVTopG    p  dc 902 dr -15 void

det OutsdRVBtmNf    n  de 2 dc 900 dr -15 void
det OutsdRVCLNf     n  de 2 dc 901 dr -15 void
det OutsdRVTopNf    n  de 2 dc 902 dr -15 void

det TopSurfN       n  de 1 ds 20 -2 dv 1.6512996e4
det TopSurfG       p  ds 20 -2 dv 1.6512996e4
det SideSurfN      n  de 1 ds 37 -2 dv 1.093274e5
det SideSurfG      p  ds 37 -2 dv 1.093274e5
det BtmSurfN       n  de 1 ds 11 -2 dv 1.6512996e4
det BtmSurfG       p  ds 11 -2 dv 1.6512996e4

det TotTritProdMacroXS n dr -55 void
*/

/*
det EnergyDetector de MyEnergyGrid
ene MyEnergyGrid 3 500 1e-11 2e1
%
```

```
det inner_lower_plenum dc 106 dr 18 m6
%
det outer_lower_plenum dc 105 dr 18 m6
%
det slice_1 dc 111 dr 18 m6
%
det slice_2 dc 112 dr 18 m6
%
det slice_3 dc 113 dr 18 m6
%
det slice_4 dc 114 dr 18 m6
%
det slice_5 dc 115 dr 18 m6
%
det slice_6 dc 116 dr 18 m6
%
det slice_7 dc 117 dr 18 m6
%
det slice_8 dc 118 dr 18 m6
%
det slice_9 dc 119 dr 18 m6
%
det riser dc 127 dr 18 m6
%
det upper_plenum dc 104 dr 18 m6
*/
%
```

```
% ______________________Physics cards_____
_____________________________________
%
set pop 20000 40 20 1
set ngamma 1         %invokes production of prompt
 gammas in neutron reactions

set ures 1
%set acelib "sss_endfb7u.xsdata"
set acelib "s2v0_endfb71.xsdata" "photon_data.
xsdata" % ENDF/B-VII.1 library
set pdatadir "/home/jc/SERPENT2/xsdata/photon_data"

/*
set opti 1        % minimum opti and small
 memory usage
set mcvol 10000000  % mc determination of
material volumes for burn
set fpcut 1e-2    % FP yield cutoff 1e05 = 0.001%
set pcc 1         % time integration method 1= predictor
method const extrapl corrector method linear interpl
set bumode 2 14   % burnup calc mode Chebbyshev
Rational Approx Method (CRAM) order 14
set depout 3      % both parents and partials
printed to output

set nbuf 15      % sets neutron tracking memory
```

```
 buffer size (5, 100) default
*/

% Prints cross section data to [input]_xs0.m file.
set xsplot 100 1e-11 2.0  %comment out for temp sens

set power 0.4e6  % 400 KW thermal output

% _____________________ Depletion cards____
_____________________________________

/* all depletion/burnup calcs and isotopes are
commented out in this version of the code
set declib "sss_endfb7.dec"
set nfylib "sss_endfb7.nfy"

% --- Depletion Steps
% set powdens 40.0e-3    %40 KW/KgU
div m1  sep 0
div m10 sep 0
%div m11 sep 0
div m12 sep 0
%div m13 sep 0
div m14 sep 0
%div m15 sep 0
div m16 sep 0
%div m17 sep 0
div m18 sep 0
div m19 sep 0

dep daytot

%0.1
%183
%365
%548
%731
%1095
%1278
%1460
%1642
%1825
%2190
%2372
%2555
%2737
%2920

%0.1
%365
%731
%1095
%1460
%1825
%2190
```

```
%2555
%2920
%3285
3650

% --- Isotope list for Inventory calculation:

set inventory

%   Actinides

982520   % Californium
982510
982500
982490
972470   % Berkelium
962500   % Curium
962480
962470
962460
962450
962440
962430
952430   % Americium
952410
942440   % Plutonium
942420
```

```
942400
942390
942380
942360
932370   % Neptunium
932360
932350
922380   % Uranium
922360
922350
922340
922330
922320
912310   % Palladium
902320   % Thorium
902300
902290
902280
892270   % Actinium

%  Noble Metals

531290   % Iodine
531270
511250   % Antimony
511210
%% 511220   % does not exist in the library
```

```
511230
501240   % Tin
501120
501130
501140
501200
%% 501210   % does not exist in the library
501220
491150   % Indium
491130
%% 491140   % does not exist in the library
481160   % Cadmium
481080
%% 481090   % does not exist in the library
481100
481140
%% 481150   % does not exist in the library
471090   % Silver
471070
461100   % Palladium
461020
%% 461030   % does not exist in the library
461040
461060
451030   % Rhodium
441040   % Ruthenium
440960
```

```
%% 440970   % does not exist in the library
440980
441020
441030
430990   % Technetium
%% 430970   % does not exist in the library
421000   % Molybdenum
420920
%% 420930   % does not exist in the library
420940
420950
420960
420970
420980
420990
410950   % Niobium
%% 410910   % does not exist in the library
%% 410920   % does not exist in the library
410930
410940
521200   % Tellurium
%% 521210   % does not exist in the library
521220
521260
521280
521300
```

```
%  Salt Seeking

400960   % Zirconium
400940
400930
400920
400910
400900
390890   % Yittrium
380900   % Strontium
380890
380880
380870
370870   % Rubidium
370850
581420   % Curium
581400
581380
581360
561380   % Barium
561370
561360
561350
561340
561330
561320
```

```
561300

551370   % Cesium

551360

551350

551340

551330


% Neutron Poisons

441010   % Ruthenium

461050   % Palladium

461070

360830   % Krypton

360850

420950   % Molybdenum

531310   % Iodine 8 day t1/2

601430   % Neodymium

611470   % Promethium

621490   % Samerium

621510



% Cover Gas System Gaseous Source Term

541240   % Xenon

541210

541340

541350

541370
```

```
541380
541400
420990   % Molydenum  easily shielded
430990   % Technicium easily shielded
441030   % Ruthenium  after SD
441050
441010
531320   % Iodine after SD
531300
531310
531350
521270   % Tellurium
521290
521310
521320
521330
521340
521350
521360
360830   % Krypton
360850

% Misc
10030   % Tritium

%% 400880   % Zirconium  xs doesnt exist
*/
```

```
%
% _________________________Plotter_________
____________________________________
%
plot 23 640 1024 [0 -100 100 -20 320]
%plot 33 640 640 [7 -100 100 -100 100]
%plot 33 640 640 [30 -100 100 -100 100]
%plot 33 640 640 [41 -100 100 -100 100]
%plot 33 640 640 [52 -100 100 -100 100]
plot 33 640 640 [63 -100 100 -100 100]
%plot 33 640 640 [74 -100 100 -100 100]
%plot 33 640 640 [85 -100 100 -100 100]
%plot 33 640 640 [96 -100 100 -100 100]
%plot 33 640 640 [106 -100 100 -100 100]
%plot 33 640 640 [111 -100 100 -100 100]
%plot 33 640 640 [116 -100 100 -100 100]
%plot 33 640 640 [121 -100 100 -100 100]
%plot 33 640 640 [126 -100 100 -100 100]
%plot 33 640 640 [131 -100 100 -100 100]
%plot 33 640 640 [136 -100 100 -100 100]
%plot 33 640 640 [141 -100 100 -100 100]


mesh 1 3000 3000
%mesh 2 3000 3000
mesh 3 3000 3000
mesh 10 1 3000 3000
%mesh 10 2 3000 3000
mesh 10 3 3000 3000

mesh 8 -13 1 3 3000 3000 0
mesh 8 -13 1 2 3000 3000 0
mesh 8 -13 2 3 3000 3000 0
mesh 8 -13 2 2 3000 3000 0

det 1 n                     % Thermal Flux Detector
%dx -225.0 225.0 1
%dy -225.0 225.0 1
%dz -235.0 250.0 1
dr -15 void
de 6

/*
ene 4 3 100 1e-11 0.625e-6
ene 5 3 100 0.625e-6 2e1
*/
ene 6 3 200   1e-11 2e1


det 2 n                     % Epi Flux Detector
%dx -225.0 225.0 1
%dy -10.0 10.0 1
%dz -10.0 10.0 1
```

```
dr -15 void

de 6


det 3 n                    % Fast Flux Detector

dr -15 void

de 6
```

Appendix B: Python Programs

Appendix B.1. Static Flux and Precursor Solver Code

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Dec 27 12:21:00 2020

@author: jc
"""
# Neutron Noise Development Code
# Written by John P. Carter on December 2020

'''
V2 added f(u) calculation
'''

import numpy as np
import cmath as cm
#import math
#import matplotlib.colors as colors
#import matplotlib.cbook as cbook
#import sympy as sp
#from numpy.linalg import inv
#from sympy import core
#import sympy.core as sp
```

```
#from sympy.matrices import Matrix
#from sympy import nroots
import matplotlib.pyplot as plt
#from mpl_toolkits import mplot3d
#from mpl_toolkits.mplot3d import Axes3D
#import matplotlib.colors as colors
#from sympy.abc import x
#w = sp.symbols('w')

#fig, ax = plt.subplots()
#extent = (0,300,0,180)



def my_range(start, end, step):
    while start <= end:
        yield start
        start += step

w = np.zeros(1000)
x1 = np.zeros(1000)
i = 0
B0_w2 = 0 + 0j
P = np.array([0,0,0,0])
G11 = np.zeros((1000, 1000), dtype=complex)
G12 = np.zeros((1000, 1000), dtype=complex)
G21 = np.zeros((1000, 1000), dtype=complex)
G22 = np.zeros((1000, 1000), dtype=complex)
```

```
delPhi = np.zeros((1000, 1000), dtype=complex)
delC = np.zeros((1000, 1000), dtype=complex)

'''
# test case f(u) coefficients
a4 = 6.51971530e-6
a3 = -3.07499255e-4
a2 = 4.06437259e-3
a1 = 1.16939349e-2
a0 = 3.08531213e-3
'''
'''
# MSnB Properties Pitch-1
l = 0.0928 #delayed neutron decay constant sec^-1
L = 392.0
D0 = 0.874 #diffusion coefficient cm
b = 0.00690  #delayed neutron fraction
nu = 2.452 #reproduction factor
#Nf = 1.0e22 # number density of fissile isotope
Sigf = 0.004164#macroscopic cross-section fission cm^-1
Siga = 0.01#0.038 #macroscopic cross section absorption cm^-1
F = 1e14 #phi_0; unperturbed core flux
v = 1.1765e6
u0 = 500.0
tau_L = L/u0  #loop transport time sec
a = 202.4938/2  #179/2 #179 full core height
tau = tau_L + (2*a)/u0  #total loop transport time
```

```
'''

'''
# MSnB Properties Pitch-2
l = 0.0928 #delayed neutron decay constant sec^-1
L = 392.0
D0 = 0.874 #diffusion coefficient cm
b = 0.00690  #delayed neutron fraction
nu = 2.452 #reproduction factor
#Nf = 1.0e22 # number density of fissile isotope
Sigf = 0.004130#macroscopic cross-section fission cm^-1
Siga = 0.01#0.038 #macroscopic cross section absorption cm^-1
F = 1e14 #phi_0; unperturbed core flux
v = 1.1765e6
u0 = 0.2
tau_L = L/u0  #loop transport time sec
a = 260.082/2  #179/2 #179 full core height
tau = tau_L + (2*a)/u0  #total loop transport time
'''
'''
# MSnB Properties Pitch-3
l = 0.0928 #delayed neutron decay constant sec^-1
L = 392.0
D0 = 0.874 #diffusion coefficient cm
b = 0.00690  #delayed neutron fraction
nu = 2.452 #reproduction factor
#Nf = 1.0e22 # number density of fissile isotope
```

```
Sigf = 0.0041095#macroscopic cross-section fission cm^-1
Siga = 0.01#0.038 #macroscopic cross section absorption cm^-1
F = 1e14 #phi_0; unperturbed core flux
v = 1.1765e6
u0 = 500.0
tau_L = L/u0  #loop transport time sec
a = 334.8458/2  #179/2 #179 full core height
tau = tau_L + (2*a)/u0  #total loop transport time
'''
'''
# MSnB Properties Pitch-4
l = 0.0928 #delayed neutron decay constant sec^-1
L = 392.0
D0 = 0.874 #diffusion coefficient cm
b = 0.00690  #delayed neutron fraction
nu = 2.452 #reproduction factor
#Nf = 1.0e22 # number density of fissile isotope
Sigf = 0.0040985#macroscopic cross-section fission cm^-1
Siga = 0.01#0.038 #macroscopic cross section absorption cm^-1
F = 1e14 #phi_0; unperturbed core flux
v = 1.1765e6
u0 = 500.0
tau_L = L/u0  #loop transport time sec
a = 417.4635/2  #179/2 #179 full core height
tau = tau_L + (2*a)/u0  #total loop transport time
'''
'''
```

```
# MSnB Properties Pitch-5
l = 0.0928 #delayed neutron decay constant sec^-1
L = 392.0
D0 = 0.874 #diffusion coefficient cm
b = 0.00690  #delayed neutron fraction
nu = 2.452 #reproduction factor
#Nf = 1.0e22 # number density of fissile isotope
Sigf = 0.0040921#macroscopic cross-section fission cm^-1
Siga = 0.01#0.038 #macroscopic cross section absorption cm^-1
F = 1e14 #phi_0; unperturbed core flux
v = 1.1765e6
u0 = 500.0
tau_L = L/u0  #loop transport time sec
a = 504.0252/2  #179/2 #179 full core height
tau = tau_L + (2*a)/u0  #total loop transport time
'''


# MSnB Properties Pitch-6
l = 0.0928 #delayed neutron decay constant sec^-1
L = 392.0
D0 = 0.874 #diffusion coefficient cm
b = 0.00690  #delayed neutron fraction
nu = 2.452 #reproduction factor
#Nf = 1.0e22 # number density of fissile isotope
Sigf = 0.0040883#macroscopic cross-section fission cm^-1
Siga = 0.01#0.038 #macroscopic cross section absorption cm^-1
F = 1e14 #phi_0; unperturbed core flux
```

```
v = 1.1765e6

u0 = 500.0

tau_L = L/u0  #loop transport time sec

a = 593.5378/2  #179/2 #179 full core height

tau = tau_L + (2*a)/u0  #total loop transport time



'''

# MSnB Properties REFERENCE BENCH

l = 0.0928 #delayed neutron decay constant sec^-1

L = 392.0

D0 = 0.874 #diffusion coefficient cm

b = 0.00690  #delayed neutron fraction

nu = 2.452 #reproduction factor

#Nf = 1.0e22 # number density of fissile isotope

Sigf = 0.004187 #macroscopic cross-section fission cm^-1

Siga = 0.01#0.038 #macroscopic cross section absorption cm^-1

F = 1e14 #phi_0; unperturbed core flux

v = 1.1765e6

u0 = 0.1

tau_L = L/u0  #loop transport time sec

a = 179/2 #179 full core height

tau = tau_L + (2*a)/u0  #total loop transport time
'''

'''

# IMSR Properties
```

```
l = 0.0904 #delayed neutron decay constant sec^-1
L = 665.0
D0 = 1.018 #diffusion coefficient cm
b = 0.0067  #delayed neutron fraction
nu = 2.4378 #reproduction factor
#Nf = 1.0e22 # number density of fissile isotope
Sigf = 0.114 #macroscopic cross-section fission cm^-1
Siga = 0.066 #macroscopic cross section absorption cm^-1
v = 8.0532e5
F = 1e14 #phi_0; unperturbed core flux
u0 = 100.0
tau_L = L/u0  #loop transport time sec
a = 400.0/2 #half core height
tau = tau_L + (2*a)/u0  #total loop transport time
'''

'''
# Test Case Properties
l = 0.1 #delayed neutron decay constant sec^-1
L = 400 #400 non-core loop length cm
u0 = 30 #100 # 50 cm/sec fuel salt speed cm/sec
a = 150.0 #core half height
D0 = 0.33 #0.33 diffusion coefficient cm
b = 0.0065 # delayed neutron fraction
nu = 1.0 #reproduction factor
Sigf = 1.0035e-2  #1007.74e-5 #macroscopic cross-section
fission cm^-1
```

```
Siga = 1.0e-2 #macroscopic cross section absorption cm^-1
v = 1.7549e5 #neutron velocity (energy)
 cm/sec
F = 1e14 #phi_0; unperturbed core flux
#Cu = 1.0 # Velocity Fluctuation constant
#CSigf = 1.0 # Cross Section Fluctuation
constant (Density)
#g = 1.0  # gamma, ratio of absorption xs
 fluct-to-nu*fission xs
fluct
#g1 = (1-b-g)/D0
tau_L = L/u0  #8.0 #loop (non-core portion)
 transport time sec
tau = tau_L + (2*a)/u0  #total loop transport
 time
'''

print('Integration valid if', tau_L*l/np.log(2),
 '>> 1.')
print('Loop Transport Factor is:', tau_L*l/
np.log(2))
print('vSigf/Siga ratio:', nu*Sigf/Siga)
print('Fuel x-core Recirc Time (sec):', tau_L)

# Define variables for Static Flux and DNP
 conc calc
B02 = (nu*Sigf*(1-b)-Siga)/D0
```

```
#C0 = (l*b*nu*Sigf)/(D0*u0)
B0s2 = (l*((nu*Sigf)-Siga))/(u0*D0)
P0 = np.array([0,0,0,0])
Roots0 = np.array([0,0,0])
#x0 = np.array([0,0,0])
#A0 = np.array([0,0,1])
P0 = np.array([1, l/u0, B02, B0s2])
Roots0 = np.roots(P0)

#c2 = (b*nu*Sigf)/(u0*Roots0[1] +l)
#c3 = (b*nu*Sigf)/(u0*Roots0[2] +l)
#Q11 = cm.exp(-Roots0[0]*a); Q12 = cm.
exp(-Roots0[1]*a);
Q13 = cm.exp(-Roots0[2]*a)
#Q21 = cm.exp(Roots0[0]*a); Q22 = cm.
exp(Roots0[1]*a);
Q23 = cm.exp(Roots0[2]*a)
#Q31 = c1/(Roots0[0] + l/u0); Q32 = c2/
(Roots0[1] + l/u0);
Q33 = c3/(Roots0[2] + l/u0)

'''
def Phi0(chi):
    A2 = 1e14
    dk12 = Roots0[0]-Roots0[1]
    dk13 = Roots0[0]-Roots0[2]
    edelk = (cm.exp(2*a*dk12)-1)/(cm.ex
p(a*(dk12+dk13))-1)
    phi0 = cm.polar(A2*((edelk*cm.exp(a*
dk13) -
cm.exp(a*dk12))*cm.exp(Roots0[0
]*chi) + \
                cm.exp(Roots0[1]*chi) -
edelk*cm.exp(Roots0[2]*chi)))[0]
    return phi0
'''

A2 = F/1.073
dk12 = Roots0[0] - Roots0[1]
dk13 = Roots0[0] - Roots0[2]
e1 = (1 - cm.exp(2*a*dk12))/(1 - cm.
exp(2*a*dk13))
e2 = cm.exp(a*Roots0[1])/cm.exp(a*
Roots0[2])
#e3 = (cm.exp(2*a*Roots0[1]-l*tau_L)-1)/
(cm.exp(2*a*Roots0[0]-l*tau_L))
#e4 = (cm.exp(2*a*Roots0[2]-l*tau_L)-1)/
(cm.exp(2*a*Roots0[0]-l*tau_L))
#A2_A1 = 1/(e1*e2*cm.exp(a*dk13) - cm.
exp(a*dk12))
#A3_A1 = (e1*e2)/(cm.exp(a*dk12) - e1*
e2*cm.exp(a*dk13))

#c1 = -A2_A1*e3*c2 - A3_A1*e4*c3
```

```
# 100 linearly spaced numbers
#size = int(2*a+1)
size = 3000
phi0 = np.zeros((size, ), dtype=complex)
phi1 = np.zeros((size, ))  # array for Static
 Neutron Flux, Phi_0(x)
C0 = np.zeros((size, ), dtype=complex)
C1 = np.zeros((size, )) # array for C0(x)
C2 = np.zeros((size, )) # array for dC0/dx
x = np.arange(-a,a, 0.197845933333333334)
 #3000 steps

i = 0
while i < 3000:

    phi0[i] = A2*((e1*e2*cm.exp(a*dk13) - cm.
exp(a*dk12))*
cm.exp(Roots0[0]*x[i]) \
                    + cm.exp(Roots0[1]*x[i]) - e1*e2*
cm.exp(Roots0[2]*x[i]))
    phi1[i] = cm.polar(phi0[i])[0]
#    phi1[i] = phi0[i].real

#    C0[i] = A2*((e1*e2*cm.exp(a*dk13) - cm.exp
(a*dk12))*c1*
cm.exp(Roots0[0]*x[i]) \
#                    + c2*cm.exp(Roots0[1]*x[i]) - c3*
e1*e2*
cm.exp(Roots0[2]*x[i]))
#    C1[i] = cm.polar(C0[i])[0]
#    phi0[i] = phi0[i].real
    i += 1

# Calculation of Statis DNP Conc C0(x)
Cft = 0.0  # Cft = Conc First Term - full core
 C0 integration
n = 0
for i in my_range(0, 2999, 1):  #z' integration
 full core - first term
    Cft = Cft + (cm.exp((l/u0)*((a/1500)*i - a))
*phi1[n]*1.0)
    n += 1
Cft = cm.polar(Cft*(1/(cm.exp(l*tau)-1)))[0]

m = 0
for i in my_range(0,2999,1):  # i is the z
 coordinate index
    n = 0
    for j in my_range(0,i,1): # z' integration - second term
    C0[m] = C0[m] + (cm.exp((l/u0)*((a/1500)
*j - a))*phi1[n]*1.0)
        n = n + 1
    C0[m] = (cm.exp((-l/u0)*((a/1500)*i - a))
*(b*nu*Sigf/u0)*
(Cft + C0[m]))
    C1[m] = cm.polar(C0[m])[0]
    m = m + 1

#Determine dC0/dx
m = 1
dx = 1.0
for i in my_range(1,2998,1):
    C2[m] = (C1[i+1]-C1[i-1])/(2*dx)
 #Central Difference
Differentiation
    C2[0] = C2[1] + (C2[2] - C2[3])
    C2[2999] = C2[2998] - (C2[2997] - C2[2998])

    m += 1

#Determine integrated C over core volume
Cint = 0.0
for i in my_range(1,2998, 1):
    Cint = Cint + C1[i]
Cint = Cint*0.1
print(’u =’, u0, ’and Cint =’, Cint)

#determine integrated flux over core
phiInt = 0.0
for i in my_range(1,2998,1):
    phiInt = phiInt + phi1[i]
phiInt = phiInt*0.1
print(’Integrated flux =’, phiInt)

#determine integrated flux over core
phi2Int = 0.0
for i in my_range(1,2998,1):
    phi2Int = phi2Int + phi1[i]*phi1[i]
phi2Int = phi2Int*0.01
print(’Integrated flux Squared =’, phi2Int)

#calc max Phi
phimax = 0.0
for i in my_range(1,2998,1):
    if phimax <= phi1[i]:
        phimax = phi1[i]
print(’Max flux =’, phimax)



#pr = phi0.real
#pii = phi0.imag
#phi0mag = (pr**2 + pii**2)**0.5
#print(phi0mag)
#phi0 = x**2
# the function, which is y = x^2 here
#y = Phi0(chi)
```

```
# setting the axes at the centr


fig = plt.figure(1)
ax = fig.add_subplot(1, 1, 1)

ax.xaxis.set_ticks_position(’bottom’)
ax.yaxis.set_ticks_position(’left’)

# plot the function
# C2 is the DNP derivative dC/dx
# C1 is the DNP conc C(x)
# phi1 is the Neutron Flux $\phi(x)$
#x = np.arange(-5, 15, 2)
#y = np.sin(x / 2.) * np.exp(x / 4.) + 6. * np.exp(-x / 4.)
plt.plot(x, C1, ’tomato’)
#plt.title(’MSnB Precursor Noise
$\delta C$ (x, $\omega$)’)
plt.ylabel(’Precursor Concentration (C)
in precursors/$cm^3$’)
plt.xlabel(’Detector Position (cm)’)

# show the plot
plt.show()


fig = plt.figure(2)
```

```
ax = fig.add_subplot(1, 1, 1)

ax.xaxis.set_ticks_position('bottom')
ax.yaxis.set_ticks_position('left')

# plot the function
# C2 is the DNP derivative dC/dx
# C1 is the DNP conc C(x)
# phi1 is the Neutron Flux $\phi(x)$
#x = np.arange(-5, 15, 2)
#y = np.sin(x / 2.) * np.exp(x / 4.) + 6. *
 np.exp(-x / 4.)
plt.plot(x, phi1, 'tomato')
#plt.title('MSnB Precursor Noise $\delta C$ (x, $\omega$)')
plt.ylabel('Neutron Flux ($\phi$) in neutrons/sec-$cm^2$')
plt.xlabel('Detector Position (cm)')

# show the plot
plt.show()
```

Appendix B.2. Green’s Function Code

```
import numpy as np
import cmath as cm
import math
import matplotlib.colors as colors
import matplotlib.cbook as cbook
#import sympy as sp
from numpy.linalg import inv
#from sympy import *
#from sympy import core
#import sympy.core as sp
#from sympy.matrices import Matrix
#from sympy import nroots
import matplotlib.pyplot as plt
#from mpl_toolkits import mplot3d
#from mpl_toolkits.mplot3d import Axes3D
#import matplotlib.colors as colors
#from sympy.abc import x
#w = sp.symbols(’w’)

#fig, ax = plt.subplots()
#extent = (0,300,0,180)


def my_range(start, end, step):
    while start <= end:
        yield start
        start += step

w = np.zeros(1000)
x1 = np.zeros(1000)
i = 0
B0_w2 = 0 + 0j
P = np.array([0,0,0,0])
G11 = np.zeros((1000, 1000), dtype=complex)
G12 = np.zeros((1000, 1000), dtype=complex)
G21 = np.zeros((1000, 1000), dtype=complex)
G22 = np.zeros((1000, 1000), dtype=complex)

'''
# MSnB Properties Test
l = 0.2885 #delayed neutron decay constant sec^-1
tau_L = 150.0 #163.0 #loop transport time sec
D0 = 0.33 #diffusion coefficient cm^-1
b = 1.36842e-3  #delayed neutron fraction
nu = 1.0 #reproduction factor
#Nf = 1.0e22 # number density of fissile isotope
#sigf = 584e-24 #micro cross section for fission
Sigf = 1007.74e-5 #macroscopic cross-section
fission cm^-1
Siga = 1.0e-2 #macroscopic cross section absorption
cm^-1
```

```
v = 1.611494e6
u0 = 9.59
#u0 = 9.6
#a = 50.0
a = 90.0 #half core height
'''

# MSnB Properties Actual Design
l = 0.0928 #delayed neutron decay constant sec^-1
L = 382.0
D0 = 0.874 #diffusion coefficient cm^-1
b = 0.0070  #delayed neutron fraction
nu = 2.452 #reproduction factor
#Nf = 1.0e22 # number density of fissile isotope
Sigf = 0.1493  #0.15042  #0.1493 #macroscopic
cross-section fission cm^-1
Siga = 0.366 #macroscopic cross section absorption
cm^-1
v = 1.1765e6
u0 = 2.6
tau_L = L/u0  #loop transport time sec
a = 166/2 #166 full core height

'''
# Test Case Properties
l = 0.1 #delayed neutron decay constant sec^-1
tau_L = 8.0 #loop transport time sec
```

```
D0 = 0.33 #diffusion coefficient cm^-1
b = 0.0065
nu = 1 #reproduction factor
Sigf = 1007.74e-5 #macroscopic cross-section fission
cm^-1
Siga = 1.0e-2 #macroscopic cross section absorption
cm^-1
v = 1.7549e5 #neutron velocity (energy) cm/sec
u0 = 50.0 #fuel salt speed cm/sec
a = 150.0 #core half height
’’’

x = 0.0 # detector position
a1 = 2.0*a #full core height
xp = 0.0 # **********   source position ***********

B = np.array([0,0,0,0,1,0])
D = np.array([0,0,0,1/u0,0,0])
#B = sp.Matrix([0, 0, 0, 0, 1, 0])
#D = sp.Matrix([0, 0, 0, 1/u0, 0, 0])
print(’Loop Transport Factor is:’, tau_L*l/math.log(2))
m = 0
#x1[0] = -a
for x in my_range(-a, a, 0.55333333333334):
#position calcs; step is 2a/300
    print("x equals", x)
    n = 0
    for i in my_range(-3.0, 3.0, 0.02): #freq calcs

#       w[n] = i
        w[n] = 10**i
        B0_w2 = 0 + 0j
        P = np.array([0,0,0,0])
        Roots = np.array([0,0,0])
        F1_w = 0 + 0j
        F2_w = 0 + 0j
        F3_w = 0 + 0j
        E1_w = 0 + 0j
        E2_w = 0 + 0j
        E3_w = 0 + 0j
        delay = 0 + 0j
        A11 = 0+0j; A12 = 0+0j; A13 = 0+0j;
A14 = 0+0j; A15 = 0+0j; A16 = 0+0j
        A21 = 0+0j; A22 = 0+0j; A23 = 0+0j;
A24 = 0+0j; A25 = 0+0j; A26 = 0+0j
        A31 = 0+0j; A32 = 0+0j; A33 = 0+0j;
A34 = 0+0j; A35 = 0+0j; A36 = 0+0j
        A41 = 0+0j; A42 = 0+0j; A43 = 0+0j;
A44 = 0+0j; A45 = 0+0j; A46 = 0+0j
        A51 = 0+0j; A52 = 0+0j; A53 = 0+0j;
A54 = 0+0j; A55 = 0+0j; A56 = 0+0j
        A61 = 0+0j; A62 = 0+0j; A63 = 0+0j;
A64 = 0+0j; A65 = 0+0j; A66 = 0+0j
```

```
        C11 = 0+0j; C12 = 0+0j; C13 = 0+0j;
C14 = 0+0j; C15 = 0+0j; C16 = 0+0j
        C21 = 0+0j; C22 = 0+0j; C23 = 0+0j;
C24 = 0+0j; C25 = 0+0j; C26 = 0+0j
        C31 = 0+0j; C32 = 0+0j; C33 = 0+0j;
C34 = 0+0j; C35 = 0+0j; C36 = 0+0j
        C41 = 0+0j; C42 = 0+0j; C43 = 0+0j;
C44 = 0+0j; C45 = 0+0j; C46 = 0+0j
        C51 = 0+0j; C52 = 0+0j; C53 = 0+0j;
C54 = 0+0j; C55 = 0+0j; C56 = 0+0j
        C61 = 0+0j; C62 = 0+0j; C63 = 0+0j;
C64 = 0+0j; C65 = 0+0j; C66 = 0+0j

        B0_w2 = ((1-b)*nu*Sigf - Siga - (1/v)*w[n]*1j)/D0

#       Polynomial Coefficient Array
#       x^3, X^2, x, constant
        P = np.array([u0,l+w[n]*1j,B0_w2*u0,((l+w[n]*1j)
*B0_w2
 + (l*b*nu*Sigf/D0))])

        Roots = np.roots(P)

#       print("For w equals", w[n])
#       print("Polynomial Roots:", Roots)
#       print("Iteration", n+1)
```

```
        F1_w = -D0*(Roots[0]**2 + B0_w2)/l

        E1_w = F1_w

        F2_w = -D0*(Roots[1]**2 + B0_w2)/l

        E2_w = F2_w

        F3_w = -D0*(Roots[2]**2 + B0_w2)/l

        E3_w = F3_w


        delay = cm.exp(-(l*tau_L + tau_L*w[n]*1j))


        A11 = cm.exp(Roots[0]*a); A12 = cm.exp(
Roots[1]*a);
        A13 = cm.exp(Roots[2]*a); A14 = 0.0; A15 =
 0.0;
 A16 = 0.0


        A21 = delay*F1_w*cm.exp(Roots[0]*a); A22 =
delay*F2_w*
cm.exp(Roots[1]*a);
        A23 = delay*F3_w*cm.exp(Roots[2]*a); A24
 -F1_w*
cm.exp(-Roots[0]*a);
        A25 = -F2_w*cm.exp(-Roots[1]*a); A26 = -F3
_w*
cm.exp(-Roots[2]*a)


        A31 = cm.exp(Roots[0]*xp); A32 = cm.exp(
Roots[1]*xp);
```

```
        A33 = cm.exp(Roots[2]*xp); A34 = -cm.exp
(Roots[0]*xp);
        A35 = -cm.exp(Roots[1]*xp); A36 = -cm.exp(
Roots[2]*xp)

        A41 = F1_w*cm.exp(Roots[0]*xp); A42 =
 F2_w*
cm.exp(Roots[1]*xp);
        A43 = F3_w*cm.exp(Roots[2]*xp); A44 =
 -F1_w*
cm.exp(Roots[0]*xp);
        A45 = -F2_w*cm.exp(Roots[1]*xp); A46 =
 -F3_w*
cm.exp(Roots[2]*xp)

        A51 = Roots[0]*cm.exp(Roots[0]*xp); A52 =
 Roots[1]*
cm.exp(Roots[1]*xp);
        A53 = Roots[2]*cm.exp(Roots[2]*xp); A54 =
 -Roots[0]*
cm.exp(Roots[0]*xp);
        A55 = -Roots[1]*cm.exp(Roots[1]*xp); A56 =
 -Roots[2]*
cm.exp(Roots[2]*xp)

     A61 = 0.0; A62 = 0.0; A63 = 0.0; A64 = cm.exp(
-Roots[0]*a);
```

```
        A65 = cm.exp(-Roots[1]*a); A66 = cm.exp
(-Roots[2]*a)

        A = np.array([[A11, A12, A13, A14, A15,
A16],
                   [A21, A22, A23, A24, A25, A26],
                   [A31, A32, A33, A34, A35, A36],
                   [A41, A42, A43, A44, A45, A46],
                   [A51, A52, A53, A54, A55, A56],
                   [A61, A62, A63, A64, A65, A66]])

        A_inv = inv(A)

        A_coeff = A_inv.dot(B)

    # linalg.solve is the function of NumPy to solve
 a system
of linear scalar equations
    # Ax=B; print x

    #print("A Solution Vector:\n", A_coeff)

        C11 = cm.exp(Roots[0]*a); C12 = cm.exp(
Roots[1]*a);
        C13 = cm.exp(Roots[2]*a); C14 = 0; C15 = 0;
 C16 = 0
```

```
        C21 = delay*E1_w*cm.exp(Roots[0]*a);
C22 = delay*E2_w*cm.exp(Roots[1]*a);
        C23 = delay*E3_w*cm.exp(Roots[2]*a);
C24 = -E1_w*cm.exp(-Roots[0]*a);
        C25 = -E2_w*cm.exp(-Roots[1]*a);
C26 = -E3_w*cm.exp(-Roots[2]*a)

        C31 = cm.exp(Roots[0]*xp);
C32 = cm.exp(Roots[1]*xp);
        C33 = cm.exp(Roots[2]*xp);
C34 = -cm.exp(Roots[0]*xp);
        C35 = -cm.exp(Roots[1]*xp);
C36 = -cm.exp(Roots[2]*xp)

        C41 = E1_w*cm.exp(Roots[0]*xp);
C42 = E2_w*cm.exp(Roots[1]*xp);
        C43 = E3_w*cm.exp(Roots[2]*xp);
C44 = -E1_w*cm.exp(Roots[0]*xp);
        C45 = -E2_w*cm.exp(Roots[1]*xp);
C46 = -E3_w*cm.exp(Roots[2]*xp)

        C51 = Roots[0]*cm.exp(Roots[0]*xp);
C52 = Roots[1]*cm.exp(Roots[1]*xp);
        C53 = Roots[2]*cm.exp(Roots[2]*xp);
C54 = -Roots[0]*cm.exp(Roots[0]*xp);
        C55 = -Roots[1]*cm.exp(Roots[1]*xp);
 C56 = -Roots[2]*cm.exp(Roots[2]*xp)
```

```
        C61 = 0; C62 = 0; C63 = 0;
C64 = cm.exp(-Roots[0]*a);
        C65 = cm.exp(-Roots[1]*a);
C66 = cm.exp(-Roots[2]*a)

        C = np.array([[C11, C12, C13, C14, C15, C16],
                   [C21, C22, C23, C24, C25, C26],
                   [C31, C32, C33, C34, C35, C36],
                   [C41, C42, C43, C44, C45, C46],
                   [C51, C52, C53, C54, C55, C56],
                   [C61, C62, C63, C64, C65, C66]])

        C_inv = inv(C)

        C_coeff = C_inv.dot(D)

        # linalg.solve is the function of NumPy to solve
 a system of linear scalar equations
        # Ax=B; print x

        #print("C Solution Vector:\n", C_coeff)

        if x >= xp:
            G11[n][m] = A_coeff[0]*cm.exp(Roots[0]*x) +
A_coeff[1]*cm.exp(Roots[1]*x) +
A_coeff[2]*cm.exp(Roots[2]*x)
```

```
            G21[n][m] = F1_w*A_coeff[0]*cm.exp(
Roots[0]*x) +
F2_w*A_coeff[1]*cm.exp(Roots[1]*x) +
F3_w*A_coeff[2]*cm.exp(Roots[2]*x)
            G12[n][m] = C_coeff[0]*cm.exp(Roots[0]
*x) +
C_coeff[1]*cm.exp(Roots[1]*x) +
C_coeff[2]*cm.exp(Roots[2]*x)
            G22[n][m] = E1_w*C_coeff[0]*cm.exp(
Roots[0]*x) +
E2_w*C_coeff[1]*cm.exp(Roots[1]*x) +
E3_w*C_coeff[2]*cm.exp(Roots[2]*x)
        elif x < xp:
            G11[n][m] = A_coeff[3]*cm.exp(Roots[0]
*x) +
A_coeff[4]*cm.exp(Roots[1]*x) +
A_coeff[5]*cm.exp(Roots[2]*x)
            G21[n][m] = F1_w*A_coeff[3]*cm.exp(
Roots[0]*x) +
F2_w*A_coeff[4]*cm.exp(Roots[1]*x) +
 F3_w*A_coeff[5]*cm.exp(Roots[2]*x)
            G12[n][m] = C_coeff[3]*cm.exp(Roots[0]
*x) +
C_coeff[4]*cm.exp(Roots[1]*x) +
C_coeff[5]*cm.exp(Roots[2]*x)
            G22[n][m] = E1_w*C_coeff[3]*cm.exp(
Roots[0]*x) +
E2_w*C_coeff[4]*cm.exp(Roots[1]*x) +
E3_w*C_coeff[5]*cm.exp(Roots[2]*x)

    #print("G11 equals:", G11)
    #print("G21 equals:", G21)
    #print("G12 equals:", G12)
    #print("G22 equals:", G22)

        n = n + 1  #increments w (freq) index
    m = m + 1   #increments x (detector) position
 index
#    x1[m] = x1[m-1]+1



# **** Plot Green Function Amplitudes vs.
Detector Position x ****
'''
plt.figure(1)
k = 0 #reset detector position counter
r = 185 #set freq counter to (log(freq)-start)/
step
for k in range(m-1):

    plt.plot(0.553333334*k-a, cm.polar(G11[r][int(k)]
)[0], 'k.')
 plt.plot(0.553333334*k-a, 50000*cm.polar(G21[r
][int(k)])[0], 'b.')
```

```
    plt.plot(0.553333334*k-a, 25*cm.polar(G12[r]
[int(k)])[0], ’g.’)
 plt.plot(0.553333334*k-a, 3000*cm.polar(G22[r]
[int(k)])[0], ’r.’)

plt.ylabel(’Green\’s Function Amplitude’)
plt.xlabel(’Detector Position (cm)’)
plt.text(100, 105, ’G11’, color=’k’)
plt.text(100, 95, ’G21’, color=’b’)
plt.text(100, 85, ’G12’, color=’g’)
plt.text(100, 75, ’G22’, color=’r’)
plt.show()



# **** Plot Green Function Phase vs. Detector
 Position x ****
plt.figure(2)
k = 0 #reset detector position counter
r = 185 #set freq counter to int((log(freq)-start)/
step)
for k in range(m-1):

    plt.plot(0.553333334*k-a, cm.polar(G11[r][k
])[1], ’k.’)
    plt.plot(0.553333334*k-a, cm.polar(G21[r][k
])[1], ’b.’)
    plt.plot(0.553333334*k-a, cm.polar(G12[r][k
])[1], ’g.’)
    plt.plot(0.553333334*k-a, cm.polar(G22[r][k
])[1], ’r.’)

plt.ylabel(’Green\’s Function Phase’)
plt.xlabel(’Detector Position (cm)’)
plt.text(-110, 0, ’G11’, color=’k’)
plt.text(-110, -0.5, ’G21’, color=’b’)
plt.text(-110, -1, ’G12’, color=’g’)
plt.text(-110, -1.5, ’G22’, color=’r’)
plt.show()


# **** Plot Green Function Amplitudes vs freq
 (w) ****
plt.figure(3)
x = 0.0  # set detector position
k = int(x+a) # calc detector position index
r = 0 # reset freq index
for r in range(n-1):

    plt.loglog(w[r], cm.polar(G11[r][k])[0], ’k.’)
    plt.loglog(w[r], cm.polar(G21[r][k])[0], ’b.’)
    plt.loglog(w[r], cm.polar(G12[r][k])[0], ’g.’)
    plt.loglog(w[r], cm.polar(G22[r][k])[0], ’r.’)

plt.ylabel(’Green\’s Function Amplitude’)
plt.xlabel(’$\omega$ (rad/sec)’)
plt.text(1e-3, 1e-4, ’G11’, color=’k’)
plt.text(3e-3, 1e-4, ’G21’, color=’b’)
plt.text(9e-3, 1e-4, ’G12’, color=’g’)
plt.text(3e-2, 1e-4, ’G22’, color=’r’)
plt.show()

# **** Plot Green Function Phase vs freq (w)
 ****
plt.figure(4)
x=0.0  # set detector position
k = int(x+a)  #calc detector position index
r = 0  #reset freq index
for r in range(174): #Index 174 equates to
 about 3 rad/sec

    plt.plot(w[r], cm.polar(G11[r][k])[1], ’k.’)
    plt.plot(w[r], cm.polar(G21[r][k])[1], ’b.’)
    plt.plot(w[r], cm.polar(G12[r][k])[1], ’g.’)
    plt.plot(w[r], cm.polar(G22[r][k])[1], ’r.’)

plt.ylabel(’Green\’s Function Phase’)
plt.xlabel(’$\omega$ (rad/sec)’)
plt.text(0.75, -2.5, ’G11’, color=’k’)
plt.text(1.25, -2.5, ’G21’, color=’b’)
plt.text(1.75, -2.5, ’G12’, color=’g’)
plt.text(2.25, -2.5, ’G22’, color=’r’)
plt.show()


# G Amplitude vs. Freq vs. x
plt.figure(5)
AmpMax = 0.0
J = [[0 for m in range(300)] for n in range(300)]
def f(n, m):
 ampl = cm.polar(G22[n][m])[0]   #set to Gxy
 ampl of choice
    return ampl
m = 0
while m < 300:  #180 max position index
    n = 0
    while n < 300: #x=a max freq index
        J[n][m] = f(n, m)
        if AmpMax < J[n][m]:
            AmpMax = J[n][m]
        else:
            AmpMax = AmpMax
        n += 1
    m += 1


c = plt.imshow(J, cmap=’jet’, extent=[-a,a,3.0
,-3.0],
aspect=’auto’, norm=colors.LogNorm(
vmin=1e-3,
vmax=AmpMax))
```

```
plt.colorbar(c)
plt.title('Kinetic Transfer Function Amplitude
 $G_{C C}$')
plt.xlabel('Detector Position, x (cm)')
plt.ylabel('$\mathrm{Log_{10}}$($\omega$)
(rad/sec)')
plt.show()

# G Amplitude vs. Freq vs. x
plt.figure(6)
AmpMax = 0.0
J = [[0 for m in range(300)] for n in range(300)]
def f(n, m):
    ampl = cm.polar(G11[n][m])[0]   #set to Gxy
amplitude of choice
    return ampl
m = 0
while m < 300:  #180 max position index
    n = 0
    while n < 300: #x=a max freq index
        J[n][m] = f(n, m)
        if AmpMax < J[n][m]:
            AmpMax = J[n][m]
        else:
            AmpMax = AmpMax
        n += 1
    m += 1
```

```
c = plt.imshow(J, cmap=’jet’, extent=[-a,a,
3.0,-3.0],
aspect=’auto’, norm=colors.LogNorm
(vmin=1e-6, vmax=AmpMax))
plt.colorbar(c)
plt.title(’Kinetic Transfer Function Amplitude
$G_{\phi \phi}$’)
plt.xlabel(’Detector Position, x (cm)’)
plt.ylabel(’$\mathrm{Log_{10}}$($\omega$)
(rad/sec)’)
plt.show()
’’’

# G Amplitude vs. Freq vs. x
plt.figure(7)
AmpMax = 0.0
J = [[0 for m in range(300)] for n in rang
e(300)]
def f(n, m):
    ampl = cm.polar(G11[n][m])[0]   #set to Gxy
amplitude of choice
    return ampl
m = 0
while m < 300:  #180 max position index
    n = 0
    while n < 300: #x=a max freq index
```

```
        J[n][m] = f(n, m)
        if AmpMax < J[n][m]:
            AmpMax = J[n][m]
        else:
            AmpMax = AmpMax
        n += 1
    m += 1

c = plt.imshow(J, cmap=’jet’, extent=[-a,a,
3.0,-3.0],
aspect=’auto’, norm=colors.LogNorm
(vmin=1e-6, vmax=AmpMax))
plt.colorbar(c)
plt.title(’Kinetic Transfer Function Amplitude
 $G_{\phi C}$’)
plt.xlabel(’Detector Position, x (cm)’)
plt.ylabel(’$\mathrm{Log_{10}}$($\omega$)
 (rad/sec)’)
plt.show()

’’’
# G Amplitude vs. Freq vs. x
plt.figure(8)
AmpMax = 0.0
J = [[0 for m in range(300)] for n in range
(300)]
def f(n, m):
    ampl = cm.polar(G21[n][m])[0]   #set
to Gxy
amplitude of choice
    return ampl
m = 0
while m < 300:  #180 max position index
    n = 0
    while n < 300: #x=a max freq index
        J[n][m] = f(n, m)
        if AmpMax < J[n][m]:
            AmpMax = J[n][m]
        else:
            AmpMax = AmpMax
        n += 1
    m += 1

c = plt.imshow(J, cmap='jet', extent=[-a,a,
3.0,-3.0],
aspect='auto', norm=colors.LogNorm
(vmin=1e-6, vmax=AmpMax))
plt.colorbar(c)
plt.title('Kinetic Transfer Function Amplitude
 $G_{C \phi}$')
plt.xlabel('Detector Position, x (cm)')
plt.ylabel('$\mathrm{Log_{10}}$($\omega$)
 (rad/sec)')
plt.show()
```

```
# G Amplitude vs. Freq vs. x
plt.figure(9)
AmpMax = 0.0
J = [[0 for m in range(300)] for n in range
(300)]
def f(n, m):
    ampl = cm.polar(G22[n][m])[0]   #set
 to Gxy
amplitude of choice
    return ampl
m = 0
while m < 300:  #180 max position index
    n = 0
    while n < 300: #x=a max freq index
        J[n][m] = f(n, m)
        if AmpMax < J[n][m]:
            AmpMax = J[n][m]
        else:
            AmpMax = AmpMax
        n += 1
    m += 1

c = plt.imshow(J, cmap=’jet’, extent=[-a,a,
3.0,-3.0],
aspect=’auto’, norm=colors.LogNorm
(vmin=1e-3, vmax=AmpMax))
```

```
plt.colorbar(c)
plt.title('Kinetic Transfer Function Amplitude
 $G_{C C}$')
plt.xlabel('Detector Position, x (cm)')
plt.ylabel('$\mathrm{Log_{10}}$($\omega$)
 (rad/sec)')
plt.show()

# G Phase vs. Freq vs. x
plt.figure(10)
PhaseMax = 0.0
J = [[0 for m in range(300)] for n in range
(174)]
#w index 174 is 3 rad/sec
def f(n, m):
    phase = cm.polar(G11[n][m])[1]
    return phase
m = 0
while m < 300:  #x=a max position index
    n = 0
    while n < 174: #174 max freq index
        J[n][m] = f(n, m)
        if PhaseMax < J[n][m]:
            PhaseMax = J[n][m]
        else:
            PhaseMax = PhaseMax
        n += 1
    m += 1

c = plt.imshow(J, cmap='jet', extent=
[-a,a,0.47712,-3.0], aspect='auto')
plt.colorbar(c)
plt.title('Kinetic Transfer Function Phase $G
_{\phi \phi}$')
plt.xlabel('Detector Position, x (cm)')
plt.ylabel('$\mathrm{Log_{10}}$($\omega$)
 (rad/sec)')
plt.show()

# G Phase vs. Freq vs. x
plt.figure(11)
PhaseMax = 0.0
J = [[0 for m in range(300)] for n in range
(174)]
#w index 174 is 3 rad/sec
def f(n, m):
    phase = cm.polar(G12[n][m])[1]
    return phase
m = 0
while m < 300:  #x=a max position index
    n = 0
    while n < 174: #174 max freq index
        J[n][m] = f(n, m)
        if PhaseMax < J[n][m]:
            PhaseMax = J[n][m]
        else:
            PhaseMax = PhaseMax
        n += 1
    m += 1

c = plt.imshow(J, cmap=’jet’, extent=[-a,a,
0.47712,-3.0],
aspect=’auto’)
plt.colorbar(c)
plt.title(’Kinetic Transfer Function Phase $G
_{\phi C}$’)
plt.xlabel(’Detector Position, x (cm)’)
plt.ylabel(’$\mathrm{Log_{10}}$($\omega$)
(rad/sec)’)
plt.show()



# G Phase vs. Freq vs. x
plt.figure(12)
PhaseMax = 0.0
J = [[0 for m in range(300)] for n in range
(174)]
#w index 174 is 3 rad/sec
def f(n, m):
    phase = cm.polar(G21[n][m])[1]
    return phase
m = 0
while m < 300:  #x=a max position index
    n = 0
    while n < 174: #174 max freq index
        J[n][m] = f(n, m)
        if PhaseMax < J[n][m]:
            PhaseMax = J[n][m]
        else:
            PhaseMax = PhaseMax
        n += 1
    m += 1

c = plt.imshow(J, cmap=’jet’, extent=[-a,a,
0.47712,-3.0],
aspect=’auto’)
plt.colorbar(c)
plt.title(’Kinetic Transfer Function Phase $G
_{C \phi}$’)
plt.xlabel(’Detector Position, x (cm)’)
plt.ylabel(’$\mathrm{Log_{10}}$($\omega$)
(rad/sec)’)
plt.show()


# G Phase vs. Freq vs. x
plt.figure(13)
```

```
PhaseMax = 0.0
J = [[0 for m in range(300)] for n in range
(174)]
#w index 174 is 3 rad/sec
def f(n, m):
    phase = cm.polar(G22[n][m])[1]
    return phase
m = 0
while m < 300:  #x=a max position index
    n = 0
    while n < 174: #174 max freq index
        J[n][m] = f(n, m)
        if PhaseMax < J[n][m]:
            PhaseMax = J[n][m]
        else:
            PhaseMax = PhaseMax
        n += 1
    m += 1

c = plt.imshow(J, cmap='jet', extent=[-a,a,0.
47712,-3.0],
aspect='auto')
plt.colorbar(c)
plt.title('Kinetic Transfer Function Phase $G_
{C C}$')
plt.xlabel('Detector Position, x (cm)')
plt.ylabel('$\mathrm{Log_{10}}$($\omega$)
 (rad/sec)’)
plt.show()
’’’



h = np.zeros((300,300))

for n in my_range(0,299,1):
    for m in my_range(0,299,1):
        if m == 0:
            h1 = cm.polar(G22[n][m+1])[0]
        else:
            h1 = cm.polar(G22[n][m])[0]
        h[n][m] = np.log10(h1)

#plt.figure(1)
fig = plt.figure(14, figsize=(6,6))
#ax = fig.gca(projection=’3d’)

x = np.linspace(-150,150,300)
y = np.linspace(-3,3,300)

#x = np.arange(-40,40,0.5)
#y = np.arange(-40,40,0.5)
X, Y = np.meshgrid(x, y)

Z = h
```

```
#fig = plt.figure()
ax = plt.axes(projection='3d')
#mycmap = plt.get_cmap('viridis')
#ax.contour3D(X,Y,Z,1,cmap=mycmap)

ax.set_title('Kinetic Transfer Function
Amplitude
$G_{C C}$')



surf1 = ax.plot_surface(X, Y, Z, rstride=3
 cstride=3,
alpha=1.0, cmap='jet')
#cset = ax.contour(X, Y, Z, zdir='z', offset=5
 cmap=
'viridis')
cset = ax.contour(X, Y, Z, zdir='x', levels=[0],
 offset=-150,
 cmap='jet')
cset = ax.contour(X, Y, Z, zdir='y', offset=3,
cmap='jet')

fig.colorbar(surf1, ax=ax, shrink=0.4, aspect=15,
anchor=(1.0, 1.0))
```

```
ax.set_xlabel('Detector Position, x (cm)')
ax.set_xlim(-150,150)
ax.set_ylabel('$\mathrm{Log_{10}}$($\omega$)
 (rad/sec)')
ax.set_ylim(-3,3)
ax.set_zlabel('Log(Transf Funct Amplitude)')
ax.set_zlim(-9,0)

plt.show()
```

Appendix B.3. Neutron Noise Code

```
# Neutron Noise Development Code rev 2
 (incl Static Fluc/DNP calculator)
# Written by John P. Carter on December 2020
'''
Setup for 300 grid size. Loop step increments
 must
 be recalculated and code updated if
core size or frequency wondow of interested is
changed, e.g. increment equals 2a/300
for core size changes and (Logwmax –
 Logwmin)/
300 for frequency window changes.
'''

import numpy as np
import cmath as cm
#import math
import matplotlib.colors as colors
#import matplotlib.cbook as cbook
#import sympy as sp
from numpy.linalg import inv
#from sympy import *
#from sympy import core
#import sympy.core as sp
```

```
#from sympy.matrices import Matrix
#from sympy import nroots
import matplotlib.pyplot as plt
#from mpl_toolkits import mplot3d
#from mpl_toolkits.mplot3d import Axes3D
#import matplotlib.colors as colors
#from sympy.abc import x
#w = sp.symbols('w')

#fig, ax = plt.subplots()
#extent = (0,300,0,180)



def my_range(start, end, step):
    while start <= end:
        yield start
        start += step

w = np.zeros(1000)
x1 = np.zeros(1000)
i = 0
B0_w2 = 0 + 0j
P = np.array([0,0,0,0])
G11 = np.zeros((1000, 1000), dtype=complex)
G12 = np.zeros((1000, 1000), dtype=complex)
G21 = np.zeros((1000, 1000), dtype=complex)
G22 = np.zeros((1000, 1000), dtype=complex)
```

```
delPhi = np.zeros((1000, 1000), dtype=complex)
delC = np.zeros((1000, 1000), dtype=complex)
mu = 0.1  #Sigf FWHM in Fuel Salt Density
Flunctuation
# BELOW - Density Oscillation Pararmeters
tpr = 0.0
tamp = 1.0 # 0.1
tom = 2.693e-4 #0.5
tmu = 6000 #100
#  define resonance freq parameters
ctapr = 1.0    #freq taper amplitude
cres = 100.0    #resonance amplitude coefficient
cfreq = 10.0   #center resonance frequency
cwidth = 1.0   #HWHM of resonance


'''
# MSnB Properties
l = 0.0928 #delayed neutron decay constant
 sec^-1
L = 382.0
D0 = 0.874 #diffusion coefficient cm^-1
b = 0.0070  #delayed neutron fraction
nu = 2.452 #reproduction factor
#Nf = 1.0e22 # number density of fissile isotope
Sigf = 0.1495 #macroscopic cross-section
 fission cm^-1
```

```
Siga = 0.366 #macroscopic cross section
absorption cm^-1
v = 1.1765e6
u0 = 2.6   #fuel salt velocity cm/s
tau_L = L/u0  #loop transport time sec
a = 166/2 #166 half core height
'''

'''
# IMSR Properties
l = 0.0904 #delayed neutron decay constant
sec^-1
L = 665.0
D0 = 1.018 #diffusion coefficient cm^-1
b = 0.0067  #delayed neutron fraction
nu = 2.4378 #reproduction factor
#Nf = 1.0e22 # number density of fissile
 isotope
Sigf = 0.114 #macroscopic cross-section
 fission cm^-1
Siga = 0.066 #macroscopic cross section
 absorption cm^-1
v = 8.0532e5
u0 = 100.0
tau_L = L/u0  #loop transport time sec
a = 400.0/2 #half core height
'''
```

```
#  Test Case Properties
l = 0.1 #delayed neutron decay constant
 sec^-1
L = 400 #non-core loop length cm
a = 150.0  #150.0 #core half height
u0 = 50 # 50 cm/sec fuel salt speed cm/sec
tau_L = L/u0  #8.0 #loop transport time sec
D0 = 0.33 #diffusion coefficient cm^-1
b = 0.0065 # delayed neutron fraction
nu = 1.0 #reproduction factor
Sigf = 1007.74e-5 #macroscopic cross-section
fission cm^-1
Siga = 1.0e-2 #macroscopic cross section
 absorption cm^-1
v = 1.7549e5 #neutron velocity (energy)
cm/sec


F = 1e14 #phi_0; unperturbed maximum core flux
Cm = 1.0
Cu = 1.0 # Velocity Fluctuation constant
CSigf = 1.0 # Cross Section Fluctuation
 constant (Density)
g = 1.0  # gamma, ratio of absorption
 xs fluct-to-nu*fission
xs fluct
```

```
g1 = (1-b-g)/D0
tau = tau_L + (2*a)/u0

print(’Out of Core Loop Transport Time = ’,
 tau_L, ’sec.’)
print(’Loop Transport Factor is:’, tau_L*
l/np.log(2))

’’’Define variables for Static Flux and
Static DNP conc calc’’’
B02 = (nu*Sigf*(1-b)-Siga)/D0
C0 = (l*b*nu*Sigf)/(D0*u0)
B0s2 = (l*(nu*Sigf-Siga))/(u0*D0)
P0 = np.array([0,0,0,0])
Roots0 = np.array([0,0,0])
# x0 = np.array([0,0,0])
# A0 = np.array([0,0,1])
P0 = np.array([1, l/u0, B02, B0s2])
Roots0 = np.roots(P0)

#c2 = (b*nu*Sigf)/(u0*Roots0[1] +l)
#c3 = (b*nu*Sigf)/(u0*Roots0[2] +l)

A2 = F/0.445
dk12 = Roots0[0] - Roots0[1]
dk13 = Roots0[0] - Roots0[2]
e1 = (1 - cm.exp(2*a*dk12))/(1 - cm.
exp(2*a*dk13))
e2 = cm.exp(a*Roots0[1])/cm.exp(a
*Roots0[2])
#e3 = (cm.exp(2*a*Roots0[1]-l*tau_L)
-1)/(cm.exp(2*a*
Roots0[0]-l*tau_L))
#e4 = (cm.exp(2*a*Roots0[2]-l*tau_L)
-1)/(cm.exp(2*a*
Roots0[0]-l*tau_L))
#A2_A1 = 1/(e1*e2*cm.exp(a*dk13) -
cm.exp(a*dk12))
#A3_A1 = (e1*e2)/(cm.exp(a*dk12) -
 e1*e2*cm.exp(a
*dk13))

#c1 = -A2_A1*e3*c2 - A3_A1*e4*c3

# 100 linearly spaced numbers
#size = int(2*a+1)
size = 300
phi0 = np.zeros((size, ), dtype=complex)
phi1 = np.zeros((size, ))  # array for
 Static Neutron Flux
, Phi_0(x)
C0 = np.zeros((size, ), dtype=complex)
C1 = np.zeros((size, )) # array for C0(x)
C2 = np.zeros((size, )) # array for dC0/dx
```

```
x = np.arange(-a,a,1.00333333333333)
 #300 steps

’’’Calculation of Static Neutron Flux, Phi_0(x)
This calculation valid in tau_L*l>>1 regime’’’
i = 0
while i < 300:
    phi0[i] = A2*((e1*e2*cm.exp(a*dk13)
- cm.exp(a*dk12
))*cm.exp(Roots0[0]*x[i]) \
                    + cm.exp(Roots0[1]*x[i]) - e1
*e2*cm.exp(
Roots0[2]*x[i]))
    phi1[i] = cm.polar(phi0[i])[0]
#    C0[i] = A2*((e1*e2*cm.exp(a*dk13) -
 cm.exp(a*dk12
))*c1*cm.exp(Roots0[0]*x[i]) \
#                    + c2*cm.exp(Roots0[1]*x[i])
 - c3*e1*e2*
cm.exp(Roots0[2]*x[i]))
#    C1[i] = cm.polar(C0[i])[0]
#    phi0[i] = phi0[i].real
    i += 1

’’’Calculation of Statis DNP Conc (x) using
 Phi_0(x)
solution’’’
```

```
Cft = 0.0  # First term – full core C
 integration
n = 0
for i in my_range(0, 299, 1):  #z' integration
 full core
 – first term
    Cft = Cft + (cm.exp((l/u0)*((a/150)*i – a))
*phi1[n]
*1.0)
    n += 1
Cft = cm.polar(Cft*(1/(cm.exp(l*tau)-1)))[0]

m = 0
for i in my_range(0,299,1):  # i is the z
coordinate index
    n = 0
    for j in my_range(0,i,1): # z' integration
– second term
        C0[m] = C0[m] + (cm.exp((l/u0)*(
(a/150)*j – a))
*phi1[n]*1.0)
        n = n + 1

    C0[m] = (cm.exp((-l/u0)*((a/150)*i – a))
*(b*nu*Sigf/
u0)*(Cft + C0[m]))
    C1[m] = cm.polar(C0[m])[0]
    m = m + 1

’’’Determine dC0/dx from Static DNP Conc (x)
solution
determined above.’’’
m = 1
dx = 1.0
for i in my_range(1,298,1):
    C2[m] = (C1[i+1]-C1[i-1])/(2*dx)
 #Central
Difference Differentiation
    C2[0] = C2[1] + (C2[2] - C2[3])
    C2[299] = C2[298] - (C2[297] - C2[298])
    m += 1

’’’Plot Phi1, C1, or C2.’’’
’’’
fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)

ax.xaxis.set_ticks_position(’bottom’)
ax.yaxis.set_ticks_position(’left’)

# plot the function
plt.plot(x, C2, ’r’)

# show the plot
```

```
plt.show()
'''


'''Neutron and/or DNP Conc Noise Calculation
using
 dC0/dx calculated in 173-180'''
x = 0.0 # detector position
a1 = 2.0*a #full core height
xp = 0.0 # ***********************
*******
  source position ***********


#Com = (b*Sigf*nu*F)/l  # Max DNP
 Conc C2


B = np.array([0,0,0,0,1,0])
D = np.array([0,0,0,1/u0,0,0])


#m = 0
for xp in my_range(-a, a,
1.00333333333333):
 # source position, xp, integration loop to
 calc delPhi
and delC
    print('xp =', xp)
    m = 0
    for x in my_range(-a, a, 1.00333333333333):
```

```
   #detector position integration loop
#       print("x equals", x)
        n = 0
        for i in my_range(-3.0, 3.0, 0.02):
 #freq calcs
#        for i in my_range(-2.0, 2.0, 0.01333333
33333334): #freq calcs

    #       w[n] = i
            w[n] = 10**i
            B0_w2 = 0 + 0j
            P = np.array([0,0,0,0])
            Roots = np.array([0,0,0])
            F1_w = 0 + 0j
            F2_w = 0 + 0j
            F3_w = 0 + 0j
            E1_w = 0 + 0j
            E2_w = 0 + 0j
            E3_w = 0 + 0j
            delay = 0 + 0j
            A11 = 0+0j; A12 = 0+0j; A13 = 0+0j;
A14 = 0+0j; A15 = 0+0j; A16 = 0+0j
            A21 = 0+0j; A22 = 0+0j; A23 = 0+0j;
A24 = 0+0j; A25 = 0+0j; A26 = 0+0j
            A31 = 0+0j; A32 = 0+0j; A33 = 0+0j;
A34 = 0+0j; A35 = 0+0j; A36 = 0+0j
```

```
            A41 = 0+0j; A42 = 0+0j; A43 = 0+0j;
 A44 = 0+0j; A45 = 0+0j; A46 = 0+0j
            A51 = 0+0j; A52 = 0+0j; A53 = 0+0j;
 A54 = 0+0j; A55 = 0+0j; A56 = 0+0j
            A61 = 0+0j; A62 = 0+0j; A63 = 0+0j;
A64 = 0+0j; A65 = 0+0j; A66 = 0+0j


            C11 = 0+0j; C12 = 0+0j; C13 = 0+0j;
C14 = 0+0j; C15 = 0+0j; C16 = 0+0j
            C21 = 0+0j; C22 = 0+0j; C23 = 0+0j;
C24 = 0+0j; C25 = 0+0j; C26 = 0+0j
            C31 = 0+0j; C32 = 0+0j; C33 = 0+0j;
 C34 = 0+0j; C35 = 0+0j; C36 = 0+0j
            C41 = 0+0j; C42 = 0+0j; C43 = 0+0j;
C44 = 0+0j; C45 = 0+0j; C46 = 0+0j
            C51 = 0+0j; C52 = 0+0j; C53 = 0+0j;
C54 = 0+0j; C55 = 0+0j; C56 = 0+0j
            C61 = 0+0j; C62 = 0+0j; C63 = 0+0j;
C64 = 0+0j; C65 = 0+0j; C66 = 0+0j


            B0_w2 = ((1-b)*nu*Sigf - Siga - (1/v)*
w[n]*1j)/D0


    #       Polynomial Coefficient Array
    #       x^3, X^2, x, constant
            P = np.array([u0,l+w[n]*1j,B0_w2*u0,
((l+w[n]*1j)*B0_w2 + (l*b*nu*Sigf/D0))])
```

```
            Roots = np.roots(P)

    #       print("For w equals", w[n])
    #       print("Polynomial Roots:", Roots)
    #       print("Iteration", n+1)

            F1_w = -D0*(Roots[0]**2 + B0_w2)/l
            E1_w = F1_w
            F2_w = -D0*(Roots[1]**2 + B0_w2)/l
            E2_w = F2_w
            F3_w = -D0*(Roots[2]**2 + B0_w2)/l
            E3_w = F3_w

            delay = cm.exp(-(l*tau_L + tau_L*w[n]
*1j))

            A11 = cm.exp(Roots[0]*a); A12 =
cm.exp(Roots[1]*a);
            A13 = cm.exp(Roots[2]*a); A14 = 0.0;
A15 = 0.0; A16 = 0.0

            A21 = delay*F1_w*cm.exp(Roots[0]*a);
A22 = delay*F2_w*cm.exp(Roots[1]*a);
            A23 = delay*F3_w*cm.exp(Roots[2]*a);
A24 = -F1_w*cm.exp(-Roots[0]*a);
            A25 = -F2_w*cm.exp(-Roots[1]*a); A26
```

```
 = -F3_w*cm.exp(-Roots[2]*a)

            A31 = cm.exp(Roots[0]*xp); A32 =
cm.exp(Roots[1]*xp);
            A33 = cm.exp(Roots[2]*xp); A34 =
 -cm.exp(Roots[0]*xp);
            A35 = -cm.exp(Roots[1]*xp); A36 =
 -cm.exp(Roots[2]*xp)

            A41 = F1_w*cm.exp(Roots[0]*xp);
A42 = F2_w*cm.exp(Roots[1]*xp);
            A43 = F3_w*cm.exp(Roots[2]*xp);
A44 = -F1_w*cm.exp(Roots[0]*xp);
            A45 = -F2_w*cm.exp(Roots[1]*xp);
A46 = -F3_w*cm.exp(Roots[2]*xp)

            A51 = Roots[0]*cm.exp(Roots[0]*xp);
A52 = Roots[1]*cm.exp(Roots[1]*xp);
            A53 = Roots[2]*cm.exp(Roots[2]*xp);
 A54 = -Roots[0]*cm.exp(Roots[0]*xp);
            A55 = -Roots[1]*cm.exp(Roots[1]*xp);
A56 = -Roots[2]*cm.exp(Roots[2]*xp)

            A61 = 0.0; A62 = 0.0; A63 = 0.0;
A64 = cm.exp(-Roots[0]*a);
            A65 = cm.exp(-Roots[1]*a);
A66 = cm.exp(-Roots[2]*a)
```

```
            A = np.array([[A11, A12, A13,
A14, A15, A16],
                       [A21, A22, A23, A24,
 A25, A26],
                       [A31, A32, A33, A34,
 A35, A36],
                       [A41, A42, A43, A44,
 A45, A46],
                       [A51, A52, A53, A54,
A55, A56],
                       [A61, A62, A63, A64,
A65, A66]])

            A_inv = inv(A)

            A_coeff = A_inv.dot(B)

        # linalg.solve is the function of NumPy
to solve a system of linear scalar equations
        # Ax=B; print x

        #print("A Solution Vector:\n", A_coeff)

            C11 = cm.exp(Roots[0]*a); C12 =
cm.exp(Roots[1]*a);
            C13 = cm.exp(Roots[2]*a); C14 =
0; C15 = 0; C16 = 0

            C21 = delay*E1_w*cm.exp(Roots[0]*a);
C22 = delay*E2_w*cm.exp(Roots[1]*a);
            C23 = delay*E3_w*cm.exp(Roots[2]*a);
 C24 = -E1_w*cm.exp(-Roots[0]*a);
            C25 = -E2_w*cm.exp(-Roots[1]*a); C26
= -E3_w*cm.exp(-Roots[2]*a)

            C31 = cm.exp(Roots[0]*xp); C32 =
cm.exp(Roots[1]*xp);
            C33 = cm.exp(Roots[2]*xp); C34 =
 -cm.exp(Roots[0]*xp);
            C35 = -cm.exp(Roots[1]*xp); C36 =
 -cm.exp(Roots[2]*xp)

            C41 = E1_w*cm.exp(Roots[0]*xp);
C42 = E2_w*cm.exp(Roots[1]*xp);
            C43 = E3_w*cm.exp(Roots[2]*xp);
C44 = -E1_w*cm.exp(Roots[0]*xp);
            C45 = -E2_w*cm.exp(Roots[1]*xp);
C46 = -E3_w*cm.exp(Roots[2]*xp)

            C51 = Roots[0]*cm.exp(Roots[0]*xp);
C52 = Roots[1]*cm.exp(Roots[1]*xp);
            C53 = Roots[2]*cm.exp(Roots[2]*xp);
C54 = -Roots[0]*cm.exp(Roots[0]*xp);
```

```
            C55 = -Roots[1]*cm.exp(Roots[1]*xp);
C56 = -Roots[2]*cm.exp(Roots[2]*xp)

            C61 = 0; C62 = 0; C63 = 0; C64 =
cm.exp(-Roots[0]*a);
            C65 = cm.exp(-Roots[1]*a); C66 =
cm.exp(-Roots[2]*a)

            C = np.array([[C11, C12, C13,
C14, C15, C16],
                       [C21, C22, C23, C24, C25,
 C26],
                       [C31, C32, C33, C34, C35,
C36],
                       [C41, C42, C43, C44, C45,
C46],
                       [C51, C52, C53, C54, C55,
 C56],
                       [C61, C62, C63, C64, C65,
C66]])

            C_inv = inv(C)

            C_coeff = C_inv.dot(D)

            # linalg.solve is the function of NumPy
to solve a system of linear scalar equations
```

```
            # Ax=B; print x

            #print("C Solution Vector:\n", C_coeff)

            if x >= xp:
                G11[n][m] = A_coeff[0]*cm.exp
(Roots[0]
*x) + A_coeff[1]*cm.exp(Roots[1]*x) +
A_coeff[2]
*cm.exp(Roots[2]*x)
                G21[n][m] = F1_w*A_coeff[0]*
cm.exp(
Roots[0]*x) + F2_w*A_coeff[1]*cm.exp(
Roots[1]*x)
+ F3_w*A_coeff[2]*cm.exp(Roots[2]*x)
                G12[n][m] = C_coeff[0]*cm.exp
(Roots[0]*x)
 + C_coeff[1]*cm.exp(Roots[1]*x) + C_
coeff[2]*cm.exp
(Roots[2]*x)
                G22[n][m] = E1_w*C_coeff[0]
*cm.exp(Roots[0]
*x) + E2_w*C_coeff[1]*cm.exp(Roots[1]
*x) + E3_w*C_
coeff[2]*cm.exp(Roots[2]*x)
            elif x < xp:
                G11[n][m] = A_coeff[3]*cm.exp
```

```
(Roots[0]*x) +
 A_coeff[4]*cm.exp(Roots[1]*x) + A_coeff
[5]*cm.exp(
Roots[2]*x)
                G21[n][m] = F1_w*A_coeff[3]*cm
.exp(Roots[0]
*x) + F2_w*A_coeff[4]*cm.exp(Roots[1]*x
) + F3_w*A_
coeff[5]*cm.exp(Roots[2]*x)
                G12[n][m] = C_coeff[3]*cm.exp
(Roots[0]*x) +
C_coeff[4]*cm.exp(Roots[1]*x) + C_coeff[5
]*cm.exp(
Roots[2]*x)
                G22[n][m] = E1_w*C_coeff[3
]*cm.exp(Roots[0]*
x) + E2_w*C_coeff[4]*cm.exp(Roots[1]*x)
+ E3_w*C_
coeff[5]*cm.exp(Roots[2]*x)

        #print("G11 equals:", G11)
        #print("G21 equals:", G21)
        #print("G12 equals:", G12)
        #print("G22 equals:", G22)
            # Integrate over xp to get flux and
delayed
 neutron precursor noise
```

```
            delU = Cu*cm.exp(-((a+xp)/u0)*1j)  #
Velocity Fluct
#            delU = 0
#            delU = (Cm*w[n]**(-5.0/3.0)+Cu)*
cm.exp(-((a+xp)/u0)*1j)
#            delU = (Cm*(w[n]**(-1)*cm.ex
p(-(w[n]-
cfreq)**2/cwidth**2))+Cu)*cm.exp(-((a+
xp)/u0)
*1j)
#            delS = CSigf*cm.exp(-((a+xp)/u0)
*1j)
 # Density (macroXS) fluct)
#            delS = CSigf*mu*cm.exp(-mu**2
*w[n]**
2/(2+tpr*w[n]*1j))*cm.exp(-((a+xp)/u0)*1j)
  #Gaussian Density (macroXS) fluct
#            delS = (Cu*(ctapr*w[n]**(-5.0/3.0)
+cres
*cm.exp(-(w[n]-cfreq)**2/cwidth**2))
+CSigf)*cm.
exp(-((a+xp)/u0)*1j)  #Gaussian Density
 (macroXS)
fluct
#            delS = ((Cm*0.5*tamp*(tmu
/cm.sqrt(2)))*(
(cm.exp(-0.25*tmu**2*(tom-w[n])**2))
```

```
 + \
#                                           (cm.exp(-0.2
5*tmu**2*(
tom+w[n]))**2))+CSigf)*cm.exp(-((a+xp
)/u0)*1j)
 #Oscillating Density (macroXS) fluct
            delS = 0.0 # set to 0.0 if no
 DelSiga
,f fluctuations



# Flux form is cosine
# DNP form is cosine
#           delPhi[n][m] = delPhi[n][m] + (G12[n][m]
*delU*Com*(np.pi/(2.0*a))*np.sin(np.pi*xp/(2.0*a))
*1.0)

            delPhi[n][m] = delPhi[n][m] + ((-(g1*G11[n]
[m]-b*G12[n][m])*delS*phi1[m] - \
                                            G12[n][m]*delU*C2[m])
*1.0)

#           delPhi[n][m] = delPhi[n][m] + ((-(g1*G11[n]
[m]-b*G12[n][m])*delS*F*np.cos(np.pi*xp/(2.0*a))
 + \
#                                            G12[n][m]*delU*Com*(
np.pi/(2.0*a))*np.sin(np.pi*xp/(2.0*a)))*1.0)
```

```
#           delC[n][m] = delC[n][m] + (G22[n][m]*delU*
Com*(np.pi/(2.0*a))*np.sin(np.pi*xp/(2.0*a))*1.0)

            delC[n][m] = delC[n][m] + ((-(g1*G21[n][m]-
b*G22[n][m])*delS*phi1[m] - \
                                        G22[n][m]*delU*C2[m])*1.0)

#           delC[n][m] = delC[n][m] + ((-(g1*G21[n][m]-b
*G22[n][m])*delS*F*np.cos(np.pi*xp/(2.0*a)) + \
#                                        G22[n][m]*delU*Com*(np.pi
/(2.0*a))*np.sin(np.pi*xp/(2.0*a)))*1.0)

            n = n + 1  #increments w (freq) index
        m = m + 1   #increments x (detector) position
 index

#    x1[m] = x1[m-1]+1

'''
# **** Plot Green Function Amplitudes vs. Detector
 Position x ****
plt.figure(1)
k = 0 #reset detector position counter
r = 185 #set freq counter to (log(freq)-start)/step
for k in range(m-1):
    plt.plot(k-a, cm.polar(G11[r][k])[0], ’k+’)
    plt.plot(k-a, 70000*cm.polar(G21[r][k])[0], ’b+’)
    plt.plot(k-a, 15*cm.polar(G12[r][k])[0], ’g+’)
    plt.plot(k-a, 3000*cm.polar(G22[r][k])[0], ’r+’)

plt.ylabel(’Green\’s Function Amplitude’)
plt.xlabel(’Detector Position (cm)’)
plt.text(75, 275, ’G11’, color=’k’)
plt.text(75, 250, ’G21’, color=’b’)
plt.text(75, 225, ’G12’, color=’g’)
plt.text(75, 200, ’G22’, color=’r’)
plt.show()



# **** Plot Green Function Phase vs. Detecto
r Position x ****
plt.figure(2)
k = 0 #reset detector position counter
r = 185 #set freq counter to int((log(freq)-start)/
step)
for k in range(m-1):

    plt.plot(k-a, cm.polar(G11[r][k])[1], ’k.’)
    plt.plot(k-a, cm.polar(G21[r][k])[1], ’b.’)
    plt.plot(k-a, cm.polar(G12[r][k])[1], ’g.’)
    plt.plot(k-a, cm.polar(G22[r][k])[1], ’r.’)
plt.ylabel(’Green\’s Function Phase’)
plt.xlabel(’Detector Position (cm)’)
plt.text(-90, 0, ’G11’, color=’k’)
plt.text(-90, -0.5, ’G21’, color=’b’)
plt.text(-90, -1, ’G12’, color=’g’)
plt.text(-90, -1.5, ’G22’, color=’r’)
plt.show()



# **** Plot Green Function Amplitudes vs freq
 (w) ****
plt.figure(3)
x = 150.0  # set detector position
k = int(x-a) # calc detector position index
r = 0 # reset freq index
for r in range(n-1):

    plt.loglog(w[r], cm.polar(G11[r][k])[0], ’k.’)
    plt.loglog(w[r], cm.polar(G21[r][k])[0], ’b.’)
    plt.loglog(w[r], cm.polar(G12[r][k])[0], ’g.’)
    plt.loglog(w[r], cm.polar(G22[r][k])[0], ’r.’)

plt.ylabel(’Green\’s Function Amplitude’)
plt.xlabel(’$\omega$ (rad/sec)’)
plt.text(90, 1, ’G11’, color=’k’)
plt.text(150, 1, ’G21’, color=’b’)
plt.text(300, 1, ’G12’, color=’g’)
plt.text(600, 1, ’G22’, color=’r’)
plt.show()



# **** Plot Green Function Phase vs freq
 (w) ****
plt.figure(4)
x=150.0  # set detector position
k = int(x-a)  #calc detector position index
r = 0  #reset freq index
for r in range(174): #Index 174 equates to
about 3 rad/sec

    plt.plot(w[r], cm.polar(G11[r][k])[1], ’k.’)
    plt.plot(w[r], cm.polar(G21[r][k])[1], ’b.’)
    plt.plot(w[r], cm.polar(G12[r][k])[1], ’g.’)
    plt.plot(w[r], cm.polar(G22[r][k])[1], ’r.’)

plt.ylabel(’Green\’s Function Phase’)
plt.xlabel(’$\omega$ (rad/sec)’)
plt.text(0.5, -1, ’G11’, color=’k’)
plt.text(1.0, -1, ’G21’, color=’b’)
plt.text(1.5, -1, ’G12’, color=’g’)
plt.text(2.0, -1, ’G22’, color=’r’)
plt.show()
```

```
# G Amplitude vs. Freq vs. x
plt.figure(5)
AmpMax = 0.0
J = [[0 for m in range(300)] for n in range(300)]
def f(n, m):
    ampl = cm.polar(G11[n][m])[0]
    return ampl
m = 0
while m < 300:  #180 max position index
    n = 0
    while n < 300: #x=a max freq index
        J[n][m] = f(n, m)
        if AmpMax < J[n][m]:
            AmpMax = J[n][m]
        else:
            AmpMax = AmpMax
        n += 1
    m += 1

c = plt.imshow(J, cmap='jet', extent=[-a,a,3.0
,-3.0], aspect='auto', norm=colors.LogNorm(vmin
=0.001, vmax=AmpMax))
plt.colorbar(c)
plt.title('Kinetic Transfer Function Amplitude $G
_{\phi \phi}$')
```

```
plt.xlabel('Detector Position, x (cm)')
plt.ylabel('$\mathrm{Log_{10}}$($\omega$)
 (rad/sec)')
plt.show()



# G Phase vs. Freq vs. x
plt.figure(6)
PhaseMax = 0.0
J = [[0 for m in range(300)] for n in range(174)]
 #w index 174 is 3 rad/sec
def f(n, m):
    phase = cm.polar(G22[n][m])[1]
    return phase
m = 0
while m < 300:  #x=a max position index
    n = 0
    while n < 174: #174 max freq index
        J[n][m] = f(n, m)
        if PhaseMax < J[n][m]:
            PhaseMax = J[n][m]
        else:
            PhaseMax = PhaseMax
        n += 1
    m += 1
```

```
c = plt.imshow(J, cmap='jet', extent=[-a,a,0
.47712,-3.0], aspect='auto')
plt.colorbar(c)
plt.title('Kinetic Transfer Function Phas
e $G_{C C}$')
plt.xlabel('Detector Position, x (cm)')
plt.ylabel('$\mathrm{Log_{10}}$($\omega$)
 (rad/sec)')
plt.show()
'''



# delPhi/delC vs. Freq vs. x
plt.figure(1)
AmpMax = 0.0
AmpMin = 1e15
J = [[0 for m in range(300)] for n in range(300)]
 #m is position index, n is freq index
def f(n, m):
    ampl = cm.polar(delC[n][m])[0]
    return ampl
m = 0
while m < 300:  #180 max position index
    n = 0
    while n < 300: #x=a max freq index
        J[n][m] = f(n, m)
        if AmpMax < J[n][m]:
            AmpMax = J[n][m]
        else:
            AmpMax = AmpMax
        if AmpMin > J[n][m]:
            AmpMin = J[n][m]
        else:
            AmpMin = AmpMin
        n += 1
    m += 1

c = plt.imshow(J, cmap=’jet’, extent=[-a,a,3.0,
-3.0], aspect=’auto’, norm=colors.LogNorm(vmin
=1e7, vmax=AmpMax))
plt.colorbar(c)
#plt.title(’MSnB Neutron Noise $\delta \phi$ (x,
 $\omega$)’)
#plt.title(’MSnB DNP Noise $\delta C#$ (x,
$\omega$)
 w/ $\omega^{-5/3}$ Turbulence’)
plt.title(’Precursor $\delta C$ (x, $\omega$)’)
plt.ylabel(’$\mathrm{Log_{10}}$($\omega$)
 (rad/sec)’)
plt.xlabel(’Detector Position, x (cm)’)
plt.show()


# 3D plot with 2D contours
```

```
h = np.zeros((300,300))
for n in my_range(0,299,1):
    for m in my_range(0,299,1):
        if m == 0:
            h1 = cm.polar(delC[n][m+1])[0]
        else:
            h1 = cm.polar(delC[n][m])[0]
        h[n][m] = np.log10(h1)
fig = plt.figure(2, figsize=(6,6))
#ax = fig.gca(projection=’3d’)

x = np.linspace(-150,150,300)
y = np.linspace(-3,3,300)

X, Y = np.meshgrid(x, y)
Z = h

ax = plt.axes(projection=’3d’)

ax.set_title(’Precursor Fluctuations $\delta
 C$ (x, $\omega$)’)


surf1 = ax.plot_surface(X, Y, Z, rstride=3,
 cstride=3, alpha=1.0, cmap=’jet’)
#cset = ax.contour(X, Y, Z, zdir=’z’, offset=
5, cmap=’viridis’)
```

```
cset = ax.contour(X, Y, Z, zdir=’x’, levels
=[0], offset=-150, cmap=’jet’)
cset = ax.contour(X, Y, Z, zdir=’y’, offset=
3, cmap=’jet’)

fig.colorbar(surf1, ax=ax, shrink=0.4, aspec
t=15, anchor=(1.0, 1.0))

ax.set_xlabel(’Detector Position, x (cm)’)
ax.set_xlim(-150,150)
ax.set_ylabel(’$\mathrm{Log_{10}}$
($\omega$) (rad/sec)’)
ax.set_ylim(-3,3)
ax.set_zlabel(’$\mathrm{Log_{10}}$
 ($\mathrm{cm^{-3}}$)’)
ax.set_zlim(7,11.48)

plt.show()
```

Appendix B.4. MSR Transient Power Code

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Mon Mar 29 08:57:54 2021

@author: jc
"""
# Transient Power MSR Development Code
 rev0
# Written by John P. Carter on March 2021
'''
This code is developmental for the purpose
 of evaluating MSR transient power
response in forced and natural circulation
 modes of operation.  The model
assumes no heating due to velocity o
r pressure drop changes in the closed-
loop system.  The model also assumes linear
heat generation and removal rates
in the reactor and heat exchanger.  The
 code uses flow reactivity results
provided by Dulla, S. PhD thesis and
 Pazsit, I. (2019).  A logamithmic fit
to the numerically derived reactivity
```

```
 results was modified and applied by the
code author for the purpose of modeling
the MSnB form factor.

Vers 7 - addresses propagator adjustment
 and changes dx from 1.0 cm to 1.0 mm.
Corrects flow area calculation.
Added dual transient abaility at tst1 and tst2
'''

'''
Assumptions:
- Rx average temp is biased 60\% to
outlet temp.
- Flow reactivity modeled per Dulla
 log regression relationship.
- Changes in core inlet temp affect
Rx power during next time step.
- System Pressure Loss Coeff is 25.
'''

#import math
#import matplotlib.cbook as cbook
#import sympy as sp
#from sympy import *
#from sympy import core
#import sympy.core as sp
```

```
#from sympy.matrices import Matrix
#from sympy import nroots
#from mpl_toolkits import mplot3d
#from mpl_toolkits.mplot3d import Axes3D
#import matplotlib.colors as colors
#from sympy.abc import x



#fig, ax = plt.subplots()
#extent = (0,300,0,180)

import numpy as np
import cmath as cm
import matplotlib.colors as colors
from numpy.linalg import inv
import matplotlib.pyplot as plt
from scipy.fft import fft, ifft, fftfreq
from scipy.signal import blackman
import pandas as pd
import math
import matplotlib.animation as
 animation



# ---  Define Constants ---
# Linear flow dimensions of the
 reactor system defined.
```

```
RxInlet0 = 0.0
RxOutlet = 1660.0
LpTop = 2090.0
HxInlet = 2340.0
HxLpTop = 2825.0
HxOutlet = 2825.0  # 289.5
LpBtm = 4975.0
RxInlet1 = 5710.0
x = 0.0
t = 0.0

densHleg = 0.0
densCleg = 0.0

dt = 1  # second
dx = 1  # mm
# dP = 0.0 #i.c. diff pressure
# u = 0.0  #i.c. fuel salt velocity
# Tave = 0.0  #average raector
 temp.

# Thermal-hydraulic and neutronic
 constants defined.
dthermCent = 1.09  # distance between
 thermal centers in meters
HelixAngle = 58.5402*(np.pi/180)
 #pitch angle of WHIP in rad
```

```
# Vpc = 1.0
Vpc = np.sin(HelixAngle)
  #component of fluid velocity parallel to WHIP
 center axis
FrCoeff = 25.0  # system
 friction loss coefficient

TmpCoeff = -3.5  # pcm/K
alpha = -3.3976917e-1  # s/cm from
 Dulla thesis regression
RxEffHeight = 1500.383   # Rx
 effective height with WHIP cord length H
PNGT = 1.63e-4  # prompt neutron
 generation - seconds
Beff = 6.96e-3*(RxEffHeight/((RxInlet1
-RxOutlet)+RxEffHeight))
DNPLeff = 0.1
DNPFrac = (RxInlet1-RxOutlet)/((RxInlet1
-RxOutlet)+RxEffHeight)

# Define Arrays (x, t) format
T = np.zeros((7000, 50000))
  # Temperature x in mm increments; time in sec
Dens = np.zeros((7000, 50000))
# Density
QRx = np.zeros(50000)
 # Reactor Power
```

```
QHx = np.zeros(50000)
  #Heat exchanger power
dP = np.zeros(50000)
  #differential pressure
u = np.zeros(50000)
  #fuel salt velocity
drhou = np.zeros(50000)
  #differential salt velocity
Tave = np.zeros(50000)
 #average system Temperature
RxTave = np.zeros(50000)
  #average Rx temperature
HxTave = np.zeros(50000)
 #average Hx temperature
drhoT = np.zeros(50000)
  #change in Temp reactivity
cpRxave = np.zeros(50000)
 #average Rx sp. heat capacity
cpHxave = np.zeros(50000)
 #average Hx sp. heat capacity
Rxperiod = np.zeros(50000)
 #Rx Period
rhodot = np.zeros(50000)
  #time rate of change of total reactivity
du = np.zeros(50000
)  #differential salt velocity
dTRx = np.zeros(50000)
```

```
 #change in Rx average temp between t
 and t+1
DTRx = np.zeros(50000)
 #differential temp across the Rx
DensH = np.zeros(50000)
  #avg Hot leg density at time t
DensC = np.zeros(50000)
 #avg Cold leg density at time t

# Define functions

def my_range(start, end, step):
    while start <= end:
        yield start
        start += step

# define Initial Conditions
Qrxinit = 1000.0  #250.0
 #400.0  # initial rx power KW
Qhxinit = 1000.0  #250.0
 #400.0  # initial Hx power demand in KW
cpinit = 1940  #1945  #1940
 # initial system Cp in J/Kg-K
densinit = 3823.33  #3819.75
 #3823.33 # initial density in kg/m3
uinit = 6.5682  #4.1351 #6.5682
 #2.3974  # initial system mass flow rate in cm/s
```

```
FlowArea = 0.06305 #0.4
 # flow area in m2
T1 = 613.905544
 # Initial Rx inlet temp in C
T0 = T1 + 273.15
 # Initial Rx inlet temp in K



# --- initialize the local Temperature,
 Density, and Cp arrays (x,t) for all x and t
for j in my_range(0, 1, dt):  # i is the
 x position in the loop in mm (use 5999)
    for i in my_range(0, 6999, 1):  # j is th
e time in sec (use 999)
        if i <= int(RxOutlet):
            T[i][j] = T0 + ((1000*Qrxinit)
/(cpinit*(densinit*FlowArea*uinit/100)))*(i/
(RxOutlet))
        if i > int(RxOutlet) and i < int(HxInlet):
            T[i][j] = T[int(RxOutlet)][j]
        if i >= int(HxInlet) and i <= int(HxOutlet):
            slope = (1000*Qhxinit)/(cpinit
*(densinit*FlowArea*uinit/100)) / \
                     ((HxInlet-HxOutlet))
            T[i][j] = T[int(RxOutlet)][j] +
 slope*(i-(HxInlet))
        if i > int(HxOutlet) and i <= int(RxInlet1):
            T[i][j] = T[int(HxOutlet)][j]
        Dens[i][j] = 1000*(-9.4601046e-4*
T[i][j] + 4.6820365)  # density in kg/m3

'''
Assumptions:
Hx and Rx has same flow area.
Flow rate is constant througout the loop.
flow exit velocity goes as sin (helix angle).
core flow area goes as cos (helix angle).
'''

# --- initialize the global arrays at t=0
QRx[0] = 1000.0  #250.0  #1000.0 #400.0
QHx[0] = 1000.0  #250.0  #1000.0 #400.0
dP[0] = 223.22066  #88.31 #223.22066
u[0] = 6.5682  #4.1351 #6.5682 #2.39742
drhou[0] = 0.0002046
Tave[0] = 882
RxTave[0] = 893.35  #912.37  #893.35
HxTave[0] = 891.004  #903.14  #891.004
drhoT[0] = -0.0002046
cpRxave[0] = 1941.2594  #1946.9958
  #1941.2594
cpHxave[0] = 1937.8124  #1945.6267
 #1937.8124
Rxperiod[0] = 4.5e12
```

```
rhodot[0] = 0.0
du[0] = 0.114424
dTRx[0] = 0.0
DTRx[0] = 32.2153  #12.874  #32.2153


'''
# --- initialize the Density array (x,t)
for i in my_range(0, 5999, 1):  # i is the
 x position in the loop in mm
    for j in my_range(0, 9999, 1):
  # j is the time in sec
        Dens[i][j] = 1000*(-9.4601046e-4
*T[i][j] +
                           4.6820365)
 # density in kg/m3
'''
# --- initialize the Hx power array
 QHx(t); linear up/down power demand
qhxfperc1 = 0.25  #4.0  #0.25
 #0.435918764500  #fraction of
 initial power after first transient
qhxfperc2 = 1.0  #fraction of
 initial power after second transient
xdur1 = 600  #transient duration in sec
tdur2 = 600
tst1 = 35000  # transient 1 start time in sec
```

```
tst2 = 39200  # transient 2 start time in sec
m1 = (Qhxinit-(qhxfperc1*
Qhxinit))/(tst1-(tst1+xdur1))
b1 = Qhxinit - (m1*tst1)

for j in my_range(0, 49999, 1):
 # j is the time in sec
#    QRx[j] = Qrxinit  # Q(t) in KW
    if j < tst1:
        QHx[j] = Qhxinit
    else:
        if j>= tst1 and j <= (tst1+xdur1):
            QHx[j] = m1*j + b1  # Q(t) in KW
        else:
            QHx[j] = qhxfperc1*Qhxinit
m2 = (QHx[tst2]-(QHx[tst2]
*qhxfperc2))/(tst2-(tst2+tdur2))
b2 = QHx[tst2] - (m2*tst2)
for j in my_range(tst2, 49999, 1):
  # adds second transient at time tst2
    if j >= tst2 and j <= (tst2+tdur2):
        QHx[j] = m2*j + b2
    else:
        QHx[j] = qhxfperc2*QHx[tst2]


'''
```

```
# --- initialize Cp array
for i in my_range(0, 599, 1):
 # i is the x position in the loop in mm
    for j in my_range(0, 999, 1):
  # j is the time in sec
        Cp[i][j] = 1.0634*T[i][j] + 976.78
 # Cp in units J/Kg-K
’’’
’’’
# --- Heat Exchanger Power Demand (t)
for j in my_range(0, 999, 1):
 # j is the time in sec
    if j <= 100:
        QHx[j] = Qhxinit  # Q(t) in KW
    else:
        QHx[j] = 2*Qhxinit
’’’

print(’finished initializing arrays’)

’’’
# **** Plot init QHx(t) ****
k = 0 #reset counter
for k in range(49999):
    plt.plot(k, QHx[k], ’b.’)
plt.ylabel(’Initial Hx Heat Removal (t)’)
plt.xlabel(’time (sec)’)
plt.show()
’’’
’’’
# **** Plot init T, Dens, Cp[n][m] ****
k = 0  # reset detector position counter
for k in range(int(RxInlet1)):
    plt.plot(k, T[k][0], ’b.’)
    plt.title(’Initial System Temperature Profile’)
plt.ylabel(’Temperature (x,0) Kelvin’)
plt.xlabel(’x position (cm)’)
plt.show()
’’’
’’’
def Densf(T, n, m):
    nx = 0
    for i in my_range(0, RxInlet, 1):
        Dens[nx][m] = -9.460104e-4*T[nx][m]
+ 4.6820365
        nx += 1
’’’
def Df(Temp):
    Density = 1000*(-9.4601046e-4*Temp
+ 4.6820365)
    return Density


def denslegf(my_range, Dens, Inlet, Outlet, j):
    m = int(Inlet)
    densleg1 = 0.0
    for i in my_range(Inlet, Outlet, dx):
        densleg1 = densleg1 + Dens[m][j]
        m += 1
    densleg = (densleg1)/(Outlet-Inlet)
 # calc avg (Hot/Cold) Leg Density in kg/m3
    return densleg


def dPf(my_range, denslegf, Dens, j):
    # calc diff pressure in Pa in system at time j
    dP[j] = (denslegf(my_range, Dens

, HxLpTop, LpBtm, j)-denslegf(my_range,
Dens, RxInlet0, LpTop, j)) * dthermCent *
 9.80665
    return dP[j]


def uf(my_range, Dens, dPf, denslegf, j):
    # calc flow in cm/s at time j
    u[j] = np.sqrt((2*dPf(my_range,
denslegf, Dens, j))/(FrCoeff*denslegf
(my_range, Dens, RxInlet0, RxInlet1, j)))*
100
    return u[j]
```

```
def duf(my_range, Dens, dPf, denslegf, uf, j):
    if j == 0:
        j=1
    du[j] = uf(my_range, Dens, dPf, denslegf, j)
-uf(my_range, Dens, dPf, denslegf, 0)
 #change in
 velocity in cm/s compared to prior to the
 transient.
    return du[j]


def drhouf(my_range, Dens, denslegf, uf,
 dPf, duf, j):
    drhou[j] = -(DNPFrac*Beff)*((1-np.
exp(-0.3397617
*uf(my_range, Dens, dPf, denslegf, j)
*Vpc))-(1-
np.exp(-0.3397617*uf(my_range, Dens,
dPf, denslegf,
 0)*Vpc))) # flow reactivity in dk/k a
t time j
    return drhou[j]


def Tavef(T, Inlet, Outlet, j):
#    temp1 = 0.0
#    for i in my_range(int(Inlet), int(Outlet), 1):
```

```
#        temp1 = temp1 + T[i][j]
#    Tave[j] = temp1/(Outlet-Inlet)  # calc avg
 (Hot/Cold) Leg Temp in K at time j
    Tave[j] = (4*T[int(Inlet)][j]+6*T[int(Outlet)
][j])/10
    return Tave[j]

def dTRxf(my_range, T, Tavef, j):
    if j == 0:
        j=1
    dTRx[j] = Tavef(T, RxInlet0, RxOutlet, j) -
Tavef(T, RxInlet0, RxOutlet, j-1)
 #calc change in
ave Rx temp between time steps.
    return dTRx[j]



def drhoTf(my_range, T, Tavef, j):
    # calc temp Coeff Reactivity in pcm
    if j == 0:
        j = 1
    drhoT[j] = 1e-5*TmpCoeff*(Tavef(T,
 RxInlet0,
RxOutlet, j)- \
                        Tavef(T, RxInlet0, RxOutlet, 0))
  # change in temp reactivity compared to prior to
 transient.
```

```
    return drhoT[j]



def cpRxavef(my_range, T, Tavef, j):
    cpRxave[j] = 1.0634*Tavef(T, RxInlet0,
                              RxOutlet, j) + 976.78
  # calc Cp in J/Kg-K
    return cpRxave[j]



def cpHxavef(my_range, T, Tavef, j):
    cpHxave[j] = 1.0634*Tavef(T, HxInlet,
                              HxOutlet, j) + 976.78
  # calc Cp in J/kg-K
    return cpHxave[j]


def rhodotf(my_range, T, drhouf, duf, drhoTf,
 dt, j):
    if j == 0:
        j=1
    rhoTot0 = (drhouf(my_range, Dens, denslegf,
 uf, dPf, duf, j)+drhoTf(my_range, T, Tavef, j))
    rhoTot1 = (drhouf(my_range, Dens, denslegf
, uf, dPf, duf, j-1)+drhoTf(my_range, T, Tavef, j-1))
    rhodot[j] = (rhoTot0-rhoTot1)/dt  # calc change
 in reactivity dk/k per unit time at time j
    return rhodot[j]
```

```
def Rxperiodf(my_range, T, drhoTf, drhouf, duf
, rhodotf, j):
    if j == 0:
        j=1
    rhoTot0 = (drhouf(my_range, Dens, denslegf
, uf, dPf, duf, j)+drhoTf(my_range, T, Tavef, j))
    Rxperiod[j] = (PNGT/rhoTot0)+((Beff-rhoTot0)
/((DNPLeff*rhoTot0)+rhodotf(my_range, T, drhouf,
 duf, drhoTf, dt, j)))
    return Rxperiod[j]

def QRxf(my_range, T, drhouf, drhoTf, rhodotf,
 duf, Rxperiodf, j):
    if j == 0:
        j=1
    QRx[j] = QRx[j-1]*np.exp(dt/Rxperiodf(my
_range,
 T, drhoTf, drhouf, duf, rhodotf, j))
    return QRx[j]

def DTRxf(QRxf, my_range, T, drhouf, drhoTf,
 rhodotf, duf, Rxperiodf, cpRxavef, denslegf,
 Tavef,
Dens, dPf, u, j):
    DTRx[j] = (1000*QRxf(my_range, T, drhouf,
drhoTf, rhodotf, duf, Rxperiodf, j))/(cpRxavef
```

```
(my_range,
 T, Tavef, j)*(denslegf(my_range, Dens, RxInlet0,
RxOutlet, j)*FlowArea*uf(my_range, Dens, dPf,
denslegf, j)/100.0))  # calc temp across core in K
 at time j.
    return DTRx[j]

# --- At each time increment, full syste
m parameters are calculated.
’’’
a = 0

densHleg = denslegf(my_range, Dens,
 RxInlet0, LpTop, a)
print(densHleg, ’Hot Leg Avg Density’)

densCleg = denslegf(my_range, Dens,
HxLpTop, LpBtm, a)
print(densCleg, ’Cold Leg Avg Density’)

dP[a] = dPf(my_range, denslegf, Dens, a)
print(dP[a], ’Differential Press. (Pa)’)

u[a] = uf(my_range, Dens, dPf, denslegf, a)
print(u[a], ’Fuel Salt Flow Velocity (cm/s)’)

du[a] = duf(my_range, Dens, dPf, denslegf,
uf, a)
print(du[a], ’Change in fuel salt flow in cm/s
 between time j and time 0’)

RxTave[a] = Tavef(T, RxInlet0, RxOutlet, a)
print(RxTave[a], ’Avg Rx Temperature in K’)

HxTave[a] = Tavef(T, HxInlet, HxOutlet, a)
print(HxTave[a], ’Avg HX temperature in K’)

dTRx[a] = dTRxf(my_range, T, Tavef, a)
print(dTRx[a], ’Change in avg Rx temp in
 K between time steps’)

drhoT[a] = drhoTf(my_range, T, Tavef, a)
print(drhoT[a], ’Temp Reactivity in core in
dk/k (0 dk/k at time = 0)’)

drhou[a] = drhouf(my_range, Dens,
denslegf, uf, dPf, duf, a)
print(drhou[a], ’Flow Reactivity in core
 in dk/k (0 dk/k at time = 0)’)

cpRxave[a] = cpRxavef(my_range, T
, Tavef, a)
print(cpRxave[a], ’Avg Rx Cp in J/kg-K’)
```

```
cpHxave[a] = cpHxavef(my_range, T,
Tavef, a)
print(cpHxave[a], 'Avg Hx Cp in J/kg-K')

rhodot[a] = rhodotf(my_range, T, drhouf,
 duf, drhoTf, dt, a)
print(rhodot[a], 'change in reactivity dk/k
 per unit time ')

Rxperiod[a] = Rxperiodf(my_range, T,
drhoTf, drhouf, duf, rhodotf, a)
print(Rxperiod[a], 'Rx period in sec at time j')

QRx[a] = QRxf(my_range, T, drhouf,
drhoTf, rhodotf, duf, Rxperiodf, a)
print(QRx[a], 'Rx power in KW at time j')

DTRx[a] = DTRxf(QRxf, my_range, T,
 drhouf, drhoTf, rhodotf, duf, Rxperiodf,
 cpRxavef, denslegf, Tavef, Dens, dPf, u, a)
print(DTRx[a], 'Differential Temp across the
 Rx at time j')
'''



print('start time iterations')
```

```
t = 0
# --- Calculate all global variables for time
 0 based on initial local arrays + QRx/QHx.
densHleg = denslegf(my_range, Dens,
 RxInlet0, LpTop, t)
DensH[t] = densHleg
’’’
print(densHleg, ’Hot Leg Avg Density’)
’’’
densCleg = denslegf(my_range, Dens,
 HxLpTop, LpBtm, t)
DensC[t] = densCleg
’’’
print(densCleg, ’Cold Leg Avg Density’)
’’’

’’’
dP[t] = dPf(my_range, denslegf, Dens, t)
print(dP[t], ’Differential Press. (Pa)’)

u[t] = uf(my_range, Dens, dPf, denslegf, t)
print(u[t], ’Fuel Salt Flow Velocity (cm/s)’)

drhou[t] = drhouf(my_range, Dens,
 denslegf, uf, dPf, duf, t)
print(drhou[t], ’Flow Reactivity in core in
 dk/k (0 dk/k at time = 0)’)

print(T[int(RxInlet0)][t], ’Rx Inlet Temp
 at time ’, t)
print(T[int(RxOutlet)][t], ’Rx Outlet Temp
 at time ’, t)
print(T[int(HxInlet)][t], ’Hx Inlet Temp
 at time ’, t)
print(T[int(HxOutlet)][t], ’Hx Outlet
Temp at time ’, t)

RxTave[t] = Tavef(T, RxInlet0, RxOutlet, t)
print(RxTave[t], ’Avg Rx Temperature in K’)

HxTave[t] = Tavef(T, HxInlet, HxOutlet, t)
print(HxTave[t], ’Avg HX temperature in K’)

drhoT[t] = drhoTf(my_range, T, Tavef, t)
print(drhoT[t], ’Temp Reactivity in core in
 dk/k (0 dk/k at time = 0)’)

cpRxave[t] = cpRxavef(my_range, T, Tavef, t)
print(cpRxave[t], ’Avg Rx Cp in J/kg-K’)

cpHxave[t] = cpHxavef(my_range, T, Tavef, t)
print(cpHxave[t], ’Avg Hx Cp in J/kg-K’)
```

```
Rxperiod[t] = Rxperiodf(my_range, T, drhoTf
, drhouf, duf, rhodotf, t)
print(Rxperiod[t], ’Rx period in sec at time 0’)

QRx[t] = QRxf(my_range, T, drhouf, drhoTf
, rhodotf, duf, Rxperiodf, t)
print(QRx[t], ’Rx power in KW at time 0’)

DTRx[t] = DTRxf(QRxf, my_range, T, drhouf,
 drhoTf, rhodotf, duf, Rxperiodf, cpRxavef,
 densleg f, Tavef, Dens, dPf, u, t)
print(DTRx[t], ’Differential Temp across the
 Rx at time 0’)
’’’

# --- initialize T array for first 100 mm into
 Rx/Hx, this support Rx/Hx inlet avg calc.
for i in my_range(0, 49900, dt):
    for j in my_range(1, 100, dx):
        T[int(RxInlet1+j)][i] = T[int(RxInle
t1-100+j)][i]
        T[int(HxInlet+j)][i] = T[int(HxInlet-
100+j)][i]

# --- TIME STEP CALC for time t.

for t in my_range(1, 49900, dt):
    print(’t = ’, t)

    T[int(RxOutlet)][t-1] = T[int(RxInlet0)]
[t-1]+DTRx[t-1]
    T[int(HxOutlet)][t-1] = T[int(HxInlet)]
[t-1]-(1000*QHx[t-1])/(cpHxave[t-1]*densCleg
*FlowArea*u[t-1]/100.0)

# --- PROPAGATOR - time-step advance
 transfer-legs from Rx-to-Hx and Hx-to-Rx

#    for q in my_range(0, int(round(u[t-1]
*dt*10)), dx):
    q = int(round(u[t-1]*dt*10))
    for i in my_range(RxOutlet, HxInlet+
100, dx):
        if i < (RxOutlet+q):
            T[int(i)][t] = T[int(RxOutlet)][t-1]
        else:
#            T[int(i+1)][t] = T[int(i)][t-1]
#            T[int(i+q)][t] = T[int(i+q-1)][t-1]
#            T[int(i+q+1)][t] = T[int(i+q)][t-1]
            T[int(i)][t] = T[int(i-q)][t-1]
            ’’’
            print(i, q, T[int(i)][t])
            ’’’
    for i in my_range(HxOutlet, RxInlet1+100, dx):
        if i < (HxOutlet+q):
            T[int(i)][t] = T[int(HxOutlet)][t-1]
        else:
#            T[int(i+1)][t] = T[int(i)][t-1]
#            T[int(i+q)][t] = T[int(i+q-1)][t-1]
#            T[int(i+q+1)][t] = T[int(i+q)][t-1]
            T[int(i)][t] = T[int(i-q)][t-1]

# --- Calc Rx/Hx Inlet as average of u*dt values

    inavg = 0.0
    for i in my_range(-99, 99, 1):
        inavg = inavg + T[int(RxInlet1+i)][t]
    T[int(RxInlet0)][t] = inavg/199

#    for i in my_range(0, int(round(u[t-1]*dt-1)), dx):
#        inavg = inavg + T[int(RxInlet1+i)][t]

#    T[int(RxInlet0)][t] = inavg/round(u[t-1]*dt)

#    T[int(RxInlet0)][t] = T[int(RxInlet1)][t]

    inavg = 0.0
    for i in my_range(-99, 99, 1):
        inavg = inavg + T[int(HxInlet+i)][t]
    T[int(HxInlet)][t] = inavg/199
#    for i in my_range(0, int(round(u[t-1]*dt-1)), dx):
```

```
#        inavg = inavg + T[int(HxInlet+i)][t]
#    T[int(HxInlet)][t] = inavg/round(u[t-1]*dt)

    ’’’
    print(T[int(RxInlet0)][t], T[int(RxInlet0)
][t-1], ’Rx Inlet0 Temp at time ’, t, t-1)
    print(T[int(RxInlet1)][t], T[int(RxInlet1)]
[t-1],’Rx Inlet1 Temp at time ’, t, t-1)
    print(T[int(RxOutlet)][t], T
[t-1],’Rx Outlet Temp at time ’, t, t-1)
    print(T[int(HxInlet)][t], T[int(HxInlet)][t-1
],’Hx Inlet Temp at time ’, t, t-1)
    print(T[int(HxOutlet)][t], T[int(HxOutlet)
][t-1],’Hx Outlet Temp at time ’, t, t-1)
    ’’’
# --- calc updated arrays based on step
 advanced transfer-legs and updated QRx and QHx

# --- Calc Average Hot leg density above
 Rx to LpTop
    densavg = 0.0
    for d in my_range(RxOutlet+1, LpTop, dx):
        densavg = densavg + Df(T[int(d)][t])

    densavg = densavg/(LpTop-(RxOutlet+1))

    densHleg = (RxOutlet*Df((4*T[int(RxInlet0)
][t]+6*T[int(RxOutlet)][t])/10) + \
               ((LpTop-RxOutlet)*densavg))/LpTop
#dens of hot let bias 60/40
    DensH[t] = densHleg

# --- Calc Average Cold leg density from
 HxLpTop down to LpBtm
    densavg = 0.0
    for d in my_range(HxOutlet+1, LpBtm, dx):
        densavg = densavg + Df(T[int(d)][t])

    densavg = densavg/(LpBtm-(HxOutlet+1))

    densCleg = densavg
    DensC[t] = densCleg

    dP[t] =  (densCleg-densHleg) * dthermCent
 * 9.80665

    densavg = 0.0
    for d in my_range(RxInlet0, RxInlet1-1, dx):
        densavg = densavg + Df(T[int(d)][t])

    AveD = densavg/((RxInlet1-1)-(RxInlet0))
  # Calc system avg density

    #AveD = Df((T[int(RxInlet0)][t]+T[int(
RxOutlet)][t])/2)  # this is an average system
 density approximation
    u[t] = np.sqrt((2*dP[t])/(FrCoeff*AveD)
)*100
    du[t] = u[t]-u[0]
    drhou[t] = -(DNPFrac*Beff)*((1-np.exp(
alpha*u[t]*Vpc))-(1-np.exp(alpha*u[0]*Vpc)))
   #-6e-5
    RxTave[t] = (4*T[int(RxInlet0)][t]+6*T
[int(RxOutlet)][t])/10
    HxTave[t] = (T[int(HxInlet)][t]+T[int(HxOutlet)]
[t])/2
    drhoT[t] = 1e-5*TmpCoeff*(((4*T[int(RxInlet0)]
[t]+6*T[int(RxOutlet)][t])/10)-((4*T[int(RxInlet0)
][0]+6*T[int(RxOutlet)][0])/10))
    cpRxave[t] = 1.0634*((4*T[int(RxInlet0)][t]+6
*T[int(RxOutlet)][t])/10) + 976.78
    cpHxave[t] = 1.0634*((T[int(HxInlet)][t]
+T[int(HxOutlet)][t])/2) + 976.78
    rhoTot0 = drhou[t]+drhoT[t]
    Rxperiod[t] = (PNGT/rhoTot0)+((Beff-rhoTot0)
/((DNPLeff*rhoTot0)+((drhou[t]+drhoT[t])-
(drhou[t-1]+drhoT[t-1])/dt)))
    QRx[t] = QRx[t-1]*np.exp(dt/Rxperiod[t])
    DTRx[t] = (1000*QRx[t])/(cpRxave[t]*
densHleg*FlowArea*u[t]/100.0)
```

```
    ’’’
    print(densHleg, ’Hot Leg Avg Density’)
    print(densCleg, ’Cold Leg Avg Density’)
    print(dP[t], ’Differential Press. (Pa)’)
    print(u[t], ’Fuel Salt Flow Velocity (cm/s)’)
    print(drhou[t], ’Flow Reactivity in core in
 dk/k (0 dk/k at time = 0)’)
    print(RxTave[t], ’Avg Rx Temperature in K’)
    print(HxTave[t], ’Avg HX temperature in K’)
    print(drhoT[t], ’Temp Reactivity in core in
dk/k (0 dk/k at time = 0)’)
    print(cpRxave[t], ’Avg Rx Cp in J/kg-K’)
    print(cpHxave[t], ’Avg Hx Cp in J/kg-K’)
    print(Rxperiod[t], ’Rx period in sec at time ’, t)
    print(QRx[t], ’Rx power in KW at time’ , t)
    print(DTRx[t], ’Differential Temp across
 the Rx at time ’, t)
    ’’’

# --- calc updated differential temps across
 Rx and Hx based on temp, flow, demand changes
’’’
    T[int(RxOutlet)][t] = T[int(RxInlet0)][t]
+DTRx[t]
    T[int(HxOutlet)][t] = T[int(HxInlet)][t]
-(1000*QHx[t])/(cpHxave[t]*densCleg*
FlowArea*u[t]/100.0)
```

```
’’’




’’’
for t in my_range(0, 999, dt):
    print(’t = ’, t, ’j = ’, t1)
    i = 0

    # --- x loop to calculate time-step advanced
 u*dt T, Dens, Cp arrays for time t.
    for x in my_range(RxInlet0, RxInlet1, dx):
        x1 = int(i-int((u[t1]*dt)))
        if x1 < 0:
            x1 = int((RxInlet1 - u[t1])*dt + i)

        T[i][t1] = T[x1][t1]
        Dens[i][t1] = 1000*(-4.4601046e-4
*T[i][t1]+4.6820365)
        Cp[i][t1] = 1.0634*T[i][t1]+976.78
        i += 1

    # --- calculate global parameters for time
 advanced T, Dens, Cp profile for time t.
    densHleg = denslegf(my_range, Dens,
 RxInlet0, LpTop, t1)
    densCleg = denslegf(my_range, Dens,
HxLpTop, LpBtm, t1)
    dP[j] = dPf(densHleg, densCleg, t1)
    u[j] = uf(dP, densHleg, densCleg, t1)
    drhou[j] = drhouf(u, t1)
    RxTave[j] = Tavef(my_range, T, RxInlet0,
 RxOutlet, t1)
    HxTave[j] = Tavef(my_range, T, HxInlet,
 HxOutlet, t1)
    drhoT[j] = drhoTf(Tavef, t1)
    cpRxave[j] = cpRxavef(Tavef, t1)
    cpHxave[j] = cpHxavef(Tavef, t1)
    QRx[j] = QRxf(QRx, drhouf, drhoTf, rhodotf,
 PNGT, Beff, DNPLeff, dt, t1)

# --- Recalculate T, Dens, Cp arrays to linear
 QRx, Hx profiles based on updated QRx, QHx
 arrays.
    for i in my_range(0, 599, 1):  # i is the x
 position in the loop in mm
        for j in my_range(0, 999, 1):  # j is th
e time in 0.1 sec
            if i <= int(RxOutlet - 1):
                T[i][j] = T[int(RxInlet1)][j] +
 ((1000*QRx[j])/(cpRxave[j]*\
                (denslegf(my_range, Dens, RxInlet0,
 RxOutlet, j)*\
                 FlowArea*u[j]/100)))*(i/(RxOutlet-1))
            if i > int(RxOutlet - 1) and i <= int(HxInlet - 1):
                T[i][j] = T[int(RxOutlet - 1)][j]
            if i >= int(HxInlet-1) and i <= int(HxOutlet - 1):
                slope = ((1000*QHx[j])/(cpHxave[j]*
(denslegf(my_range, Dens, HxInlet, HxOutlet, j)
*FlowArea*u[j]/100))) / \
                     ((HxInlet-1)-(HxOutlet - 1))
                T[i][j] = slope*i + (T[int(HxInlet-1)
][j] - slope*(HxInlet-1))
            if i >= int(HxOutlet - 1) and i <= int(RxInlet1):
                T[i][j] = T[int(HxOutlet - 1)][j]
            Dens[i][j] = 1000*(-9.4601046e-4*
T[i][j] + 4.6820365)  # density in kg/m3
            Cp[i][j] = 1.0634*T[i][j] + 976.78
 # Cp in units J/Kg-K

    t1 += 1
'''

'''
    # --- Calculate all local variables for all x
at time t.
    for x in my_range(RxInlet0, RxInlet1, dx):
        # --- calc of system Temps for all x a
nd time t
        if i <= int(RxOutlet - 1):
            T[i][j] = T[int(RxInlet0-1)][j] + \
                ((1000*QRx[j])/(cpRxave[j]*u[j])
)*(i/(RxOutlet-1))
        if i > int(RxOutlet - 1) and i <= int(HxInlet - 1):
            T[i][j] = T[int(RxOutlet - 1)][j]
        if i >= int(HxInlet-1) and i <= int(HxOutlet - 1):
            slope = (QHx[j]/(cpHxave[j]*u[j]))/
                     ((HxInlet-1)-(HxOutlet - 1))
            T[i][j] = slope*i + (T[int(HxInlet-1)]
                                 [j] - slope*(HxInlet-1))
        if i >= int(HxOutlet - 1) and i <= int(RxInlet1):
            T[i][j] = T[int(HxOutlet - 1)][j]
        # ---- calc of system Temps for all x
 at time t complete.
        # --- calc of system Densities for all
 x at time t.
        Dens[i][j] = 1000*(-9.4601046e-4*
T[i][j] +
                           4.6820365)  # density in kg/m3
        # --- calc of system Densities for all
 x at time t.
        # --- calc of system sp. heat capacities
 for all x at time t.
        Cp[i][j] = 1.0634*T[i][j] + 976.78
  # Cp in units J/Kg-K
        # --- calc of system sp. heat capacities
for all x at time t complete.
```

```
        i += 1

    j += 1
'''

# Calc min and max Rx inlet temp after the transient

j = 0
trmax = 0
prmax = 0
#trmin = 10000
thmin = 10000
prmin = 2000
dPmin = 200
umin = 200
for j in my_range(34990, 38000, 1):
    if T[int(RxOutlet)][j] > trmax:
        trmax = T[int(RxOutlet)][j]
    if T[int(HxOutlet)][j] < thmin:
        thmin = T[int(HxOutlet)][j]
#    if T[int(RxOutlet)][j] < trmin:
#        trmin = T[int(RxOutlet)][j]
    if QRx[j] < prmin:
        prmin = QRx[j]
    if QRx[j] > prmax:
        prmax = QRx[j]
    if dP[j] < dPmin:
        dPmin = dP[j]
    if u[j] < umin:
        umin = u[j]

print(T[int(RxOutlet)][34995]-T[int(RxInlet0)]
[34995
],T[int(HxInlet)][34995]-T[int(HxOutlet)][34995],
 dP[34995], u[34995], )
print(T[int(RxOutlet)][38000], T[int(RxInlet0)]
[38000], T[int(HxOutlet)][38000], T[int(HxInlet)
][38000], dP[38000], u[38000], DTRx[38000]
, cpRxave[38000], RxTave[38000])
print(qhxfperc1, T[int(RxOutlet)][34995], prmax,
prmin, trmax, thmin, umin, dPmin)



# **** Plot Temps at defined position(t) ****
plt.figure(1)
j = 0 #reset counter
for j in my_range(34990, 38000, 1):
    plt.plot((j-tst1)/60, T[int(RxInlet1)][j], ms=0.7,
 marker=’.’, color=’midnightblue’, label=’Rx Inlet’)
 #olivedrab
    plt.plot((j-tst1)/60, T[int(HxInlet)][j], ms=0.7,
 marker=’.’, color=’orangered’, label=’Hx Inlet’)
  #darkcyan
```

```
    plt.plot((j-tst1)/60, T[int(HxOutlet)][j], ms=0.7,
marker=’.’, color=’dodgerblue’, label=’Hx Outlet’)
  #orchid
    plt.plot((j-tst1)/60, T[int(RxOutlet)][j], ms=0.7,
 marker=’.’, color=’crimson’, label=’Rx Outlet’)  #c

plt.title(’Reactor and Heat Exchanger In/Outlet
 Temperatures’)
#plt.legend(loc=’upper right’)
plt.ylabel(’Temp (K)’)
plt.xlabel(’time (min)’)
plt.grid()
plt.show()

j = 0 #reset counter
plt.figure(2)
for j in my_range(34990, 38000, 1):
    plt.plot((j-tst1)/60, DensH[j], ms=0.7,
 marker=’.’, color=’red’) #fuchsia
    plt.plot((j-tst1)/60, DensC[j], ms=0.7,
marker=’.’, color=’midnightblue’)  #steelblue

plt.title(’Hot (red) and Cold Leg (blue)
Average Density’)
plt.ylabel(’Density (kg/$mˆ3$)’)
plt.xlabel(’time (min)’)
plt.grid()
plt.show()


plt.figure(3)
j = 0 #reset counter
for j in my_range(34990, 38000, 1):
    plt.plot((j-tst1)/60, QRx[j], ms=0.7,
marker=’.’, color=’red’)  #fuchsia
    plt.plot((j-tst1)/60, QHx[j], ms=0.7,
marker=’.’, color=’midnightblue’)  #steelblue

plt.title(’Reactor (red) & Heat Exchanger
(blue) Power’)
plt.ylabel(’Power (KW)’)
plt.xlabel(’time (min)’)
plt.show()


plt.figure(4)
j = 0 #reset counter
for j in my_range(34990, 38000, 1):
    plt.plot((j-tst1)/60, u[j], ms=0.7,
marker=’.’, color=’midnightblue’)  #olivedrab

plt.title(’System Flow Rate’)
plt.ylabel(’Flow Rate (cm/sec)’)
plt.xlabel(’time (min)’)
plt.grid()
plt.show()
```

```
# --- Plot reactivity (t), QRx(t), etc...
plt.figure(5)
j = 0 #reset counter
for j in my_range(34990, 38000, 1):
    plt.plot((j-tst1)/60, 1e5*drhoT[j],
 ms=0.7, marker='.', color='red')  #firebrick
    plt.plot((j-tst1)/60, 1e5*drhou[j],
 ms=0.7, marker='.', color='midnightblue')
 #dodgerblue
    plt.plot((j-tst1)/60, 1e5*(drhoT[j]
+drhou[j]), ms=0.7, marker='.', color='green')
 #darkgreen
plt.title('Core Reactivity Dynamics (Red -
 Temp; Blue - Flow; Green - Total)')
plt.ylabel('Reactivity (pcm)')
plt.xlabel('time (min)')
plt.show()

plt.figure(6)
j = 0 #reset counter
for j in my_range(34990, 38000, 1):
    plt.plot(QRx[j], u[j], ms=0.7, marker='.'
, color='red')  #firebrick

plt.title('Core Power - Flow Velocity Phase Space')
plt.ylabel('u (cm/s)')
plt.xlabel('Power (KW)')
plt.show()

plt.figure(7)
j = 0 #reset counter
for j in my_range(34990, 38000, 1):
    plt.plot(1e5*(drhoT[j]+drhou[j]), u[j],
ms=0.7, marker='.', color='red')  #firebrick

plt.title('Total Reactivity - Flow Velocity
 Phase Space')
plt.ylabel('u (cm/s)')
plt.xlabel('Core Reactivity (pcm)')
plt.show()

plt.figure(8)
j = 0 #reset counter
for j in my_range(34990, 38000, 1):
    plt.plot(1e5*drhoT[j], 1e5*drhou[j],
 ms=0.7, marker='.', color='red')  #firebrick

plt.title('Reactivity Phase Space')
plt.ylabel('Flow Reactivity (pcm)')
plt.xlabel('Temperature Reactivity (pcm)')
plt.show()

# plot FFT (PSD if squared) of Reactivity
```

```
j = 0 #reset counter
N = 49900
Ni = 100  # sets the a-axis max scale (Hz)
T = 1/N   #interval between frequency point
w1 = blackman(N)  # FFT window -
minimize spectrum leakage due to data truncation

drhoT = np.resize(drhoT, 49900)
drhou = np.resize(drhou, 49900)
fftdrhoT = fft(1e5*drhoT)
fftdrhoTw = fft(1e5*drhoT*w1)
fftdrhou = fft(1e5*drhou)
fftdrhouw = fft(1e5*drhou*w1)

xf = fftfreq(N, T)[:N//2]
plt.figure(9)
plt.semilogy(xf[1:Ni], 2/N*abs(fftdrhoTw[1:Ni]),
 linestyle=’-’, marker=’,’, color=’b’)  #m
plt.semilogy(xf[1:Ni], 2/N*abs(fftdrhouw[1:Ni]),
 linestyle=’-’, marker=’,’, color=’r’)  #sienna
plt.semilogy(xf[1:Ni], 2/N*abs(fftdrhoTw[1:Ni]+
fftdrhouw[1:Ni]), linestyle=’-’, marker=’,’, color=’g’)
 #teal

#for j in my_range(0, 30, 1):
#    plt.plot(j, abs(fftdrhoT[j]), ’.b-’)
#    plt.plot(j, abs(fftdrhou[j]), ’.r-’)
#    plt.plot(j, abs(fftdrhoT[j]+fftdrhou[j]), '.g-')
plt.title('Amplitude Spectrum Core Reactivity w/
 FFT window')
plt.ylabel('Amplitude (pcm)')
plt.xlabel('$\omega$ (cycles/sec)')
plt.legend(['Temp', 'Flow', 'Total'])
plt.grid()
plt.show()




# plot FFT of Reactivity Phase
j = 0 #reset counter
N = 49900
Ni = 100
T = 1/N
w1 = blackman(N)  #minimizes spectrum leakage
 due to truncated FFT at 49900
w2 = np.hamming(N)

drhoT = np.resize(drhoT, 49900)
drhou = np.resize(drhou, 49900)
fftdrhoT = fft(1e5*drhoT)
fftdrhoTw = fft(1e5*drhoT*w1)
fftdrhou = fft(1e5*drhou)
fftdrhouw = fft(1e5*drhou*w1)
```

```
xf = fftfreq(N, T)[:N//2]
plt.figure(10)
plt.plot(xf[1:Ni], 180*(np.angle(fftdrhoT[1:Ni])-np
.angle(fftdrhou[1:Ni]))/3.1415926, linestyle=’-’,
 marker=’,’, color=’b’)  #fuchsia
#plt.plot(xf[1:Ni], 180*(np.angle(fftdrhoTw[1:Ni]-
fftdrhouw[1:Ni]))/3.1415926, linestyle=’-’, marker=’,’,
 color=’b’)
#plt.plot(180*(np.angle(fftdrhoTw[1:Ni])-np.angle
(fftdrhouw[1:Ni]))/3.1415926, 2/N*abs(fftdrhoTw[1:Ni]
+fftdrhouw[1:Ni]), linestyle=’-’, marker=’,’, color=’b’)

plt.title(’FFT Reactivity Phase Difference ($\phi_T$
 - $\phi_f$)’)
plt.ylabel(’$\phi$ (degrees)’)
plt.xlabel(’$\omega$ (cycles/sec)’)
plt.grid()
plt.show()

’’’
# Atrempt to create an animation of transient
dependnencies

fig = plt.figure()
ax = plt.axes()
points = []
#for i in range(6010):
```

```
#    x = 1e5*drhoT[24990+int(i)]
#    y = 1e5*drhou[24990+int(i)]
#    points.append(ax.plot(x, y, marker=’o’)[0])

def init():
    points.set_data([], [])
    return points

def animate(time, drhoT, drhou):

    for i in range(6010):
        x = 1e5*drhoT[24990+int(i)]
        y = 1e5*drhou[24990+int(i)]
        points[i].set_data(x, y)
    return points

ani = animation.FuncAnimation(
    fig, animate,
    init_func=init,
    frames=6010, interval=1,
    blit=True)

#animation.save(’Reactivity_Anim.mp4’, fps=30)

plt.show()

’’’
```

Appendix D. Neutron Noise Transfer Function Coefficients Derivation

The completeness in deriving the Green's function solution for the MSR system, the following is provided in addition to the approach outlined by Dykin (Dykin, 2016a).

The third order characteristic equation of the Fourier transformed homogeneous MSR Green's matrix equation is given by,

$$u_o k^3 + (\lambda + i\omega)k^2 + B_0^2(\omega)u_o k + (\lambda + i\omega)B_o^2(\omega) + \frac{\lambda}{D_o}\beta\nu\Sigma_f^o \;=\; 0. \tag{5.30}$$

The characteristic form of the solution involves the three roots (k_1, k_2, k_3) and associated three solution term coefficients for regions $x' > x$ and $x < x'$, separately. As a reminder, x' is the perturbation source position and x is the detector position. This solution form results in six total unknown coefficients. A matrix solution method is used with boundary conditions to resolve the solution coefficients and roots. Specifically, the explicit form of the Green's function solutions are,

$$G_{11} = \begin{cases} A_1^+(x')e^{k_1(\omega)x} \;+\; A_2^+(x')e^{k_2(\omega)x} \;+\; A_3^+(x')e^{k_3(\omega)x}, & \text{for x>x'} \\ A_1^-(x')e^{k_1(\omega)x} \;+\; A_2^-(x')e^{k_2(\omega)x} \;+\; A_3^-(x')e^{k_3(\omega)x}, & \text{for x<x'} \end{cases} \tag{5.31}$$

$$G_{21} = \begin{cases} F_1(\omega)A_1^+(x')e^{k_1(\omega)x} \;+\; F_2(\omega)A_2^+(x')e^{k_2(\omega)x} \;+\; F_3(\omega)A_3^+(x')e^{k_3(\omega)x}, & \text{for x>x'} \\ F_1(\omega)A_1^-(x')e^{k_1(\omega)x} \;+\; F_2(\omega)A_2^-(x')e^{k_2(\omega)x} \;+\; F_3(\omega)A_3^-(x')e^{k_3(\omega)x}, & \text{for x<x'} \end{cases} \tag{5.32}$$

$$G_{12} = \begin{cases} C_1^+(x')e^{k_1(\omega)x} \,+\, C_2^+(x')e^{k_2(\omega)x} \,+\, C_3^+(x')e^{k_3(\omega)x}, & \text{for x>x’} \\ C_1^-(x')e^{k_1(\omega)x} \,+\, C_2^-(x')e^{k_2(\omega)x} \,+\, C_3^-(x')e^{k_3(\omega)x}, & \text{for x<x’} \end{cases} \tag{5.33}$$

$$G_{22} = \begin{cases} E_1(\omega)C_1^+(x')e^{k_1(\omega)x} \,+\, E_2(\omega)C_2^+(x')e^{k_2(\omega)x} \,+\, E_3(\omega)C_3^+(x')e^{k_3(\omega)x}, & \text{for x>x’} \\ E_1(\omega)C_1^-(x')e^{k_1(\omega)x} \,+\, E_2(\omega)C_2^-(x')e^{k_2(\omega)x} \,+\, E_3(\omega)C_3^-(x')e^{k_3(\omega)x}, & \text{for x<x’} \end{cases} \tag{5.34}$$

where,

$$F_n(\omega) \;=\; E_n(\omega) \;=\; -\frac{D_0(k_n^2 \,+\, B_0^2(\omega))}{\lambda} \tag{5.35}$$

for n = 1, 2, and 3.

Boundary conditions are explicitly written out as,

$$G_{11}^{x<x'}(-a,x',\omega) \;=\; G_{11}^{x>x'}(a,x',\omega) \;=\; 0 \tag{5.36}$$

$$= \; A_1^+(x')e^{k_1(\omega)a} \,+\, A_2^+(x')e^{k_2(\omega)a} \,+\, A_3^+(x')e^{k_3(\omega)a} \tag{5.37}$$

$$= \; A_1^-(x')e^{-k_1(\omega)a} \,+\, A_2^-(x')e^{-k_2(\omega)a} \,+\, A_3^-(x')e^{-k_3(\omega)a} \tag{5.38}$$

$$G_{12}^{x<x'}(-a,x',\omega) \;=\; G_{12}^{x>x'}(a,x',\omega) \;=\; 0 \tag{5.39}$$

$$= \; C_1^+(x')e^{k_1(\omega)a} \,+\, C_2^+(x')e^{k_2(\omega)a} \,+\, C_3^+(x')e^{k_3(\omega)a} \tag{5.40}$$

$$= \; C_1^-(x')e^{-k_1(\omega)a} \,+\, C_2^-(x')e^{-k_2(\omega)a} \,+\, C_3^-(x')e^{-k_3(\omega)a} \tag{5.41}$$

$$\begin{align}
G_{21}^{x<x'}(-a,x',\omega) &= G_{21}^{x>x'}(a,x',\omega)e^{-(\lambda+i\omega)\tau_L} \tag{5.42}\\
&= F_1A_1^-(x')e^{-k_1(\omega)a} + F_2A_2^-(x')e^{-k_2(\omega)a} + F_3A_3^-(x')e^{-k_3(\omega)a} \tag{5.43}\\
&= (F_1A_1^+e^{k_1(\omega)a} + F_2A_2^+e^{k_2(\omega)a} + F_3A_3^+e^{k_3(\omega)a})e^{-(\lambda+i\omega)\tau_L} \tag{5.44}
\end{align}$$

$$\begin{align}
G_{22}^{x<x'}(-a,x',\omega) &= G_{22}^{x>x'}(a,x',\omega)e^{-(\lambda+i\omega)\tau_L} \tag{5.45}\\
&= E_1C_1^-(x')e^{-k_1(\omega)a} + E_2C_2^-(x')e^{-k_2(\omega)a} + E_3C_3^-(x')e^{-k_3(\omega)a} \tag{5.46}\\
&= (E_1C_1^+e^{k_1(\omega)a} + E_2C_2^+e^{k_2(\omega)a} + E_3C_3^+e^{k_3(\omega)a})e^{-(\lambda+i\omega)\tau_L} \tag{5.47}
\end{align}$$

$$\begin{align}
G_{11}^{x<x'}(x'_-,x',\omega) &= G_{11}^{x>x'}(x'_+,x',\omega)e^{-(\lambda+i\omega)\tau_L} \tag{5.48}\\
&= A_1^+(x')e^{k_1(\omega)x'_+} + A_2^+(x')e^{k_2(\omega)x'_+} + A_3^+(x')e^{k_3(\omega)x'_+} \tag{5.49}\\
&= A_1^-(x')e^{-k_1(\omega)x'_-} + A_2^-(x')e^{-k_2(\omega)x'_-} + A_3^-(x')e^{-k_3(\omega)x'_-} \tag{5.50}
\end{align}$$

$$\begin{align}
G_{21}^{x<x'}(x'_-,x',\omega) &= G_{21}^{x>x'}(x'_+,x',\omega)e^{-(\lambda+i\omega)\tau_L} \tag{5.51}\\
&= F_1A_1^-(x')e^{k_1(\omega)x'_-} + F_2A_2^-(x')e^{k_2(\omega)x'_-} + F_3A_3^-(x')e^{k_3(\omega)x'_-} \tag{5.52}\\
&= F_1A_1^+e^{k_1(\omega)x'_+} + F_2A_2^+e^{k_2(\omega)x'_+} + F_3A_3^+e^{k_3(\omega)x'_+} \tag{5.53}
\end{align}$$

$$\begin{align}
\partial_x G_{11}^{x>x'}(x'_+,x',\omega) &= \partial_x G_{11}^{x<x'}(x'_-,x',\omega) + 1 \tag{5.54}\\
&= k_1A_1^+(x')e^{k_1(\omega)x'_+} + k_2A_2^+(x')e^{k_2(\omega)x'_+} + k_3A_3^+(x')e^{k_3(\omega)x'_+} \tag{5.55}\\
&= k_1A_1^-e^{k_1(\omega)x'_-} + k_2A_2^-e^{k_2(\omega)x'_-} + k_3A_3^-e^{k_3(\omega)x'_-} + 1 \tag{5.56}
\end{align}$$

$$\begin{align}
G_{12}^{x<x'}(x'_-,x',\omega) &= G_{12}^{x>x'}(x'_+,x',\omega) \tag{5.57}\\
&= C_1^+(x')e^{k_1(\omega)x'_+} + C_2^+(x')e^{k_2(\omega)x'_+} + C_3^+(x')e^{k_3(\omega)x'_+} \tag{5.58}\\
&= C_1^-(x')e^{k_1(\omega)x'_-} + C_2^-(x')e^{k_2(\omega)x'_-} + C_3^-(x')e^{k_3(\omega)x'_-} \tag{5.59}
\end{align}$$

$$
\begin{aligned}
G_{22}^{x>x'}(x'_+, x', \omega) &= G_{22}^{x<x'}(x'_-, x', \omega) + \frac{1}{u_0} && (5.60)\\
&= E_1 C_1^- e^{k_1(\omega)x'_-} + E_2 C_2^- e^{k_2(\omega)x'_-} + E_3 C_3^- e^{k_3(\omega)x'_-} + \frac{1}{u_0} && (5.61)\\
&= E_1 C_1^+ e^{k_1(\omega)x'_+} + E_2 C_2^+ e^{k_2(\omega)x'_+} + E_3 C_3^+ e^{k_3(\omega)x'_+} && (5.62)
\end{aligned}
$$

$$
\begin{aligned}
\partial_x G_{12}^{x<x'}(x'_-, x', \omega) &= \partial_x G_{12}^{x>x'}(x'_+, x', \omega) && (5.63)\\
&= k_1 A_1^+(x') e^{k_1(\omega)x'_+} + k_2 A_2^+(x') e^{k_2(\omega)x'_+} + k_3 A_3^+(x') e^{k_3(\omega)x'_+} && (5.64)\\
&= k_1 A_1^- e^{k_1(\omega)x'_-} + k_2 A_2^- e^{k_2(\omega)x'_-} + k_3 A_3^- e^{k_3(\omega)x'_-} && (5.65)
\end{aligned}
$$

For brevity E_n and F_n are implicitly understood to be functions of ω and A_n and C_n are implicitly understood to be functions of x'. Representation into matrix format results in,

$$
\begin{pmatrix}
e^{k_1 a} & e^{k_2 a} & e^{k_3 a} & 0 & 0 & 0\\
F_1 e^{k_1 a} e^{-(\lambda+i\omega)\tau_L} & F_2 e^{k_2 a} e^{-(\lambda+i\omega)\tau_L} & F_3 e^{k_3 a} e^{-(\lambda+i\omega)\tau_L} & -F_1 e^{-k_1 a} & -F_2 e^{-k_2 a} & -F_3 e^{-k_3 a}\\
e^{k_1 x'_-} & e^{k_2 x'_-} & e^{k_3 x'_-} & -e^{k_1 x'_-} & -e^{k_2 x'_-} & -e^{k_3 x'_-}\\
F_1 e^{k_1 x'_+} & F_2 e^{k_2 x'_+} & F_3 e^{k_3 x'_+} & -F_1 e^{k_1 x'_-} & -F_2 e^{k_2 x'_-} & -F_3 e^{k_3 x'_-}\\
k_1 e^{k_1 x'_+} & k_2 e^{k_2 x'_+} & k_3 e^{k_3 x'_+} & -k_1 e^{k_1 x'_-} & -k_2 e^{k_2 x'_-} & -k_3 e^{k_3 x'_-}\\
0 & 0 & 0 & e^{-k_1 a} & e^{-k_2 a} & e^{-k_3 a}
\end{pmatrix}
\begin{pmatrix} A_1^+\\ A_2^+\\ A_3^+\\ A_1^-\\ A_2^-\\ A_3^- \end{pmatrix}
=
\begin{pmatrix} 0\\ 0\\ 0\\ 0\\ 1\\ 0 \end{pmatrix}
\qquad (5.66)
$$

and

$$
\begin{pmatrix}
e^{k_1 a} & e^{k_2 a} & e^{k_3 a} & 0 & 0 & 0\\
E_1 e^{k_1 a} e^{-(\lambda+i\omega)\tau_L} & E_2 e^{k_2 a} e^{-(\lambda+i\omega)\tau_L} & E_3 e^{k_3 a} e^{-(\lambda+i\omega)\tau_L} & -E_1 e^{-k_1 a} & -E_2 e^{-k_2 a} & -E_3 e^{-k_3 a}\\
e^{k_1 x'_+} & e^{k_2 x'_+} & e^{k_3 x'_+} & -e^{k_1 x'_-} & -e^{k_2 x'_-} & -e^{k_3 x'_-}\\
E_1 e^{k_1 x'_+} & E_2 e^{k_2 x'_+} & E_3 e^{k_3 x'_+} & -E_1 e^{k_1 x'_-} & -E_2 e^{k_2 x'_-} & -E_3 e^{k_3 x'_-}\\
k_1 e^{k_1 x'_+} & k_2 e^{k_2 x'_+} & k_3 e^{k_3 x'_+} & -k_1 e^{k_1 x'_-} & -k_2 e^{k_2 x'_-} & -k_3 e^{k_3 x'_-}\\
0 & 0 & 0 & e^{-k_1 a} & e^{-k_2 a} & e^{-k_3 a}
\end{pmatrix}
\begin{pmatrix} C_1^+\\ C_2^+\\ C_3^+\\ C_1^-\\ C_2^-\\ C_3^- \end{pmatrix}
=
\begin{pmatrix} 0\\ 0\\ 0\\ \frac{1}{u_0}\\ 0\\ 0 \end{pmatrix}
\qquad (5.67)
$$

The Python matrix solver is used to determine the coefficients A_n and C_n.

Appendix D. Derivation of distance between Two Points on a Helix

The geodesic between two points on a helix is used to understand the axial coupling of core physics parameters in the WHIP. Unlike purely vertical core flow that is present in traditional solid and liquid fueled designs which results in linear diffusion gradients, the periodic azimuthal flow patterns in the WHIP result in non-linear periodic coupling in neutron transport diffusion. The distance between point on a helix provides a first order approximation of this enhanced coupling effect. The distance between two points on a helix can be derived by first defining a time-parameterized vector, $\vec{\mathbf{r}}$, that depicts the fuel salt flow path. This vector has the form,

$$\vec{\mathbf{r}} = R\cos(\omega t)\vec{\mathbf{i}} + R\sin(\omega t)\vec{\mathbf{j}} + (u_{\|}t - a)\vec{\mathbf{k}} \quad \text{for } t \in \left[0, \frac{2a}{u_{\|}}\right], \tag{5.68}$$

where ω denotes the angular velocity of the propagating fuel salt and is given by $u_{\perp}/R$, R is the radius of the helical path of interest, α is the helix angle, and u is the fuel salt channel velocity. A midpoint on the helix can be defined at z=0 or t=$a/u_{\|}$,

$$\vec{\mathbf{r}'} = R\cos\left(\frac{a\cot(\alpha)}{R}\right)\vec{\mathbf{i}} + R\sin\left(\frac{a\cot(\alpha)}{R}\right)\vec{\mathbf{j}} + 0\vec{\mathbf{k}}. \tag{5.69}$$

The magnitude of the difference between these two vectors denotes the distance between these points. Specifically,

$$\|\vec{\mathbf{dr}}\| = \|\vec{\mathbf{r}} - \vec{\mathbf{r}'}\|, \tag{5.70}$$

where $\vec{\mathbf{dr}}$ denotes the vector from the helix midpoint, $\vec{\mathbf{r}'}$, to another point of interest on the helix, $\vec{\mathbf{r}}$. Solving for $\|\vec{\mathbf{dr}}\|$, we have,

$$\|\vec{\mathbf{dr}}\| = \left[R^2\left(\cos\left(\frac{u_\perp t}{R}\right) - \cos\left(\frac{a\cot\alpha}{R}\right)\right)^2 + R^2\left(\sin\left(\frac{u_\perp t}{R}\right) - \sin\left(\frac{a\cot\alpha}{R}\right)\right)^2 + \left(u_\parallel t - a\right)^2\right]^{\frac{1}{2}} \quad (5.71)$$

where $u_\perp$=$u$cosα and $u_\parallel$=$u$sinα, the component of the fuel salt velocity perpendicular and parallel to the WHIP axis, respectively.

Appendix E. Acronyms

AC	Administrative Control
BOL	Beginning of Life
BWR	Boiling Water Reactor
CAD	Computer Aided Design
CDE	Compound Dynamic Effect
CFD	Computational Fluid Dynamics
DBRC	Doppler Broadening Rejection Correction
DNP	Delayed Neutron Precursors
DSA	Documented Safety Analysis
DWI	Density Wave Instabilities
EOL	End of Life
FLiBe	Fluoride Lithium Beryllium
FLiNaK	Fluoride Lithium Sodium Potassium Salt
GIF	Generation IV International Forum
GPT	Generalized Perturbation Theory
HALEU	High Assay Low Enriched Uranium
HFIR	High Flux Isotope Reactor
HPC	High Performance Computing
IAEA	International Atomic Energy Agency
IMSR	Integral Molten Salt Reactor
INL	Idaho National Laboratory
LEU	Low Enriched Uranium
LWR	Light Water Reactor
MCNP	Monte Carlo N-Particle

MeV	Mega Electron Volt
MOOSE	Multi-Physics Object Oriented Simulation Environment
MSFR	Molten Salt Fast Reactor
MsNB	Molten Salt Nuclear Battery
MSR	Molten Salt Reactor
MSRE	Molten Salt Reactor Experiment
OH-NC	Ohmic Heated Natural Circulation
ORNL	Oak Ridge National Laboratory
ORR	Oak Ridge Research Reactor
PJFNK	Pre-conditioned Jacobian Free Newton Krylov
PK	Point Kinetic
PLC	Pressure Loss Coefficient
PWR	Pressurized Water Reactor
SAMOFAR	Safety Assessment of the Molten Salt Fast Reactor
sd	Space Dependent
Sdr	Turbulence Specific Dissipation Rate
SEU	Slightly Enriched Uranium
TEI	Terrestrial Energy Incorporated
Tke	Turbulent Kinetic Energy
TMS	Target Motion Sampling
TRU	Transuranic
TTB	Thick Target Bremstrahlung
WHIP	Wrapped Helix around an Inclined Plane

Appendix F. Constants and Symbols

ϕ_o	Static neutron flux ($\mathrm{cm}^{-2}\ \mathrm{s}^{-1}$)
D_o	Diffusion Coefficient (cm)
β	Delayed Neutron Precursor fraction
Σ	Macroscopic cross section (cm^{-1})
ν	Average number of neutrons produced per fission
τ	Total loop recirculation time (sec)
τ_L	External core loop recirculation time (sec)
H	Core length (cm)
L	External core loop length (cm)
v	Neutron average velocity (cm/s)
Λ	Prompt neutron generation time (sec)
S	Neutron noise source ($\frac{\#}{\mathrm{cm}^3-\mathrm{s}}$) (subscripted ϕ and C for flux and DNP contribution)
G	Green's function or Transfer function
G_o	Zero Power Transfer Function
β_{eff}	Effective Delayed Neutron Precursor fraction
λ	One-group delayed neutron precursor decay constant (sec^{-1})
C_o	Static Delayed Neutron Precursor concentration (cm^{-3})
u_o	Static molten salt fuel flow velocity (cm s^{-1})
a	Core half height (cm)
τ	Reactor Period (sec)
L	External core loop length (cm)
P	Pressure (Pa)
ρ	Density (kg m^{-3})

g	Gravitational Constant
z	Fluid height (cm)
$v_{\mathrm{in/out}}$	Flow velocity (cm s^{-1})
$\bar{v}$	Average fuel salt flow velocity (cm s^{-1})
ξ	Pressure Loss Coefficient
$h_{\mathrm{in/out}}$	Fluid specific enthalpy (J kg^{-1})
h_c	Distance between thermal centers (cm)
f	Darcy friction factor
D_H	Hydraulic diameter (cm)
K	Hydraulic loss factor
$\dot{Q}$	Heat production rate (W)
$\dot{m}$	Mass flow rate (kg s^{-1})
c_p	Specific heat capacity (J kg^{-1} K^{-1})
$T_{\mathrm{in/out}}$	Temperature (K)
w_i	Weight percent fraction
α_T	Temperature coefficient of reactivity (pcm K^{-1})
α_f	Flow reactivity exponent (sec cm^{-1})
ρ_f	Reactivity due to flow (pcm)
ρ_T	Reactivity due to temperature (pcm)
ρ_∞	Flow reactivity at infinite flow velocity (pcm)
K_{eff}	Neutron multiplication factor
$f(u)$	Precursor fraction function
l^*	Prompt neutron lifetime (sec)
Pe	Peclet number
k	Thermal conductivity (J m^{-1} K^{-1})
S_P^R	Sensitivity parameter to perturbation P and response function R.
$S(\alpha, \beta)$	Thermal bound cross section libraries.

σ	Microscopic cross section cm^{-1} or spectral resonance width in chapter four.
D	Density (kg m^{-3})
η	Thermal fission factor in 6-factor formula
f	Thermal utilization factor in 6-factor formula
p	Resonance escape probably in 6-factor formula
$\mathscr{L}_f$	Fast non-leakage factor in 6-factor formula
$\mathscr{L}_t$	Thermal non-leakage factor in 6-factor formula
ϵ	Fast fission factor in 6-factor formula
D_s	WHIP diameter (cm)
γ	Helix Angle
H	Core Height (cm)
$u_{\parallel}$	Fuel salt speed component parallel to WHIP longitudinal axis
$u_{\perp}$	Fuel salt speed component perpendicular to WHIP longitudinal axis
l_d	Downcomer flow length (cm)
l_c	Chimney flow length (cm)