VALIDATION & SENSITIVITY STUDY OF A 1D MULTIPHYSICS MODEL FOR GRAPHITE MODERATED MOLTEN SALT REACTOR (MSRE) USING GEN-FOAM

A Dissertation

by

LAITH JAMIL KHAMEES ZAIDAN

Submitted to the Graduate and Professional School of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Mark Kimber
Committee Members,	Karen Kirkland
	Pavel Tsvetkov
	Jorge Alvarado
Head of Department,	Michael Nastasi

May 2023

Major Subject: Nuclear Engineering

ABSTRACT

The Molten Salt Reactor (MSR) is one of multiple advanced reactor designs that has received increased attention due to its enhanced safety features, reduced waste production, and increased non-proliferation capabilities. However, further research is necessary to fully prepare for large-scale implementation and overcome regulatory hurdles. The complex nature of the thermal hydraulics/neutronics design in MSRs requires extensive Multiphysics modeling, which sets it apart from conventional solid fuel reactors like Light Water Reactors (LWRs). This research aimed to explore the physical phenomena related to thermal MSRs. A preliminary step in the research involved studying the effect of turbulence model variations on molten salt behavior in circular heated pipes. Five turbulence models were evaluated for their impact on capturing thermal hydraulic metrics for molten salt flow. This step laid the foundation for further exploration of the complex physics involved in MSRs through Multiphysics modeling using the open-source GeN-Foam code. GeN-Foam, a novel OpenFoam-based solver, was specifically developed for the nuclear community and equipped with a point kinetics solver. With the help of a validated 1D model of the MSRE, a comprehensive uncertainty analysis was performed to better understand the impact of input and numerical uncertainties. Grid Convergence Index (GCI) was used to evaluate the relative importance of the input variables, while the One Factor at a Time (OFAT) and Latin Hypercube Sampling (LHS) methods were used to study input uncertainty. This research provides a significant contribution to the field of thermal MSRs and is an important step towards establishing trusted recommendations for a fuel salt scenario.

DEDICATION

To my parents, wife, siblings, and children.

ACKNOWLEDGEMENTS

I would like to thank my committee chair and advisor Dr. Mark Kimber, and my committee members, Dr. Karen Kirkland, Dr. Pavel Tsvetkov, and Dr Jorge Alvarado for their guidance and support. I would also like to acknowledge Dr. Carlo Fiorina as he provided a consistent support throughout the course of this research.

Thanks also to my friends and colleagues, faculty and staff in the nuclear and aerospace engineering departments for making my time at Texas A&M University a great experience. Finally, a special thanks to my siblings, Mohanad, Mohammad, Zain, and Rahmeh for their support and to my mom and dad for their continuous encouragement and my wife for her love and patience.

This research would not have been possible without the invaluable support and guidance of Dr. Carlo Fiorina, who not only developed the code used in this study, but also provided priceless insights.

A special recognition is owed to my friend Dr. Abdullah Weiss for his assistance and productive discussions. His support and insightful discussions have been instrumental in furthering this research effort.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supervised by a dissertation committee consisting of Drs. Mark Kimber, Pavel Tsvetkov and Karen Kirkland of the Department of Nuclear Engineering, and Dr. Jorge Alvarado of the Department of Engineering Technology & Industrial Distribution, Manufacturing.

All other work conducted for the dissertation was completed by the student independently.

Funding Sources

The first year of graduate study was supported by the DOE Office of Nuclear Energy's Nuclear Energy University Program through NuSTEM project. The following years were made possible due to funding from Natura Resources. This project is one of many R&D studies supporting MSRs development.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

The United States was the first to investigate the concept of molten salt reactors late 1940's [1], which was motivated by the U.S. Air Force aiming to develop an engine that supplies nuclear energy for aircraft propulsion [2]. It began with a 2.5 MW_{th} thermal reactor at Oak Ridge National Laboratory (ORNL) as part of the Aircraft Reactor Experiment (ARE) that was supported by the U.S. Aircraft Nuclear Propulsion program. While the attempt was unsuccessful, it did position ORNL to then apply that knowledge toward ground-based molten salt reactors, which led to the construction of the Molten Salt Reactor Experiment (MSRE) in 1964. **Figure 1** shows a layout depicting the reactor primary and secondary loops, and the drain tank system, which are shielded and installed in containment vessels while the heavily shielded remote maintenance control room is located above the operating floor [3].

In liquid fuel reactors, molten salt is mixed with fluorides of fissile/fertile elements (UF_4, PuF_3 and/or ThF_4). This increases the complexity in predicting molten salt behavior, which highlights the need to understand the tight coupling between neutronics and thermal hydraulics. The fuel for the MSRE was LiF-BeF_2- ZrF_4-UF_4 (65.0/29.2/5.0/0.8 mole%), the secondary coolant was FLiBe (2LiF- BeF_2), and the core shown in **Figure 2** was graphite moderated. It operated for about 1.5 years of equivalent full power after it reached criticality in 1965.

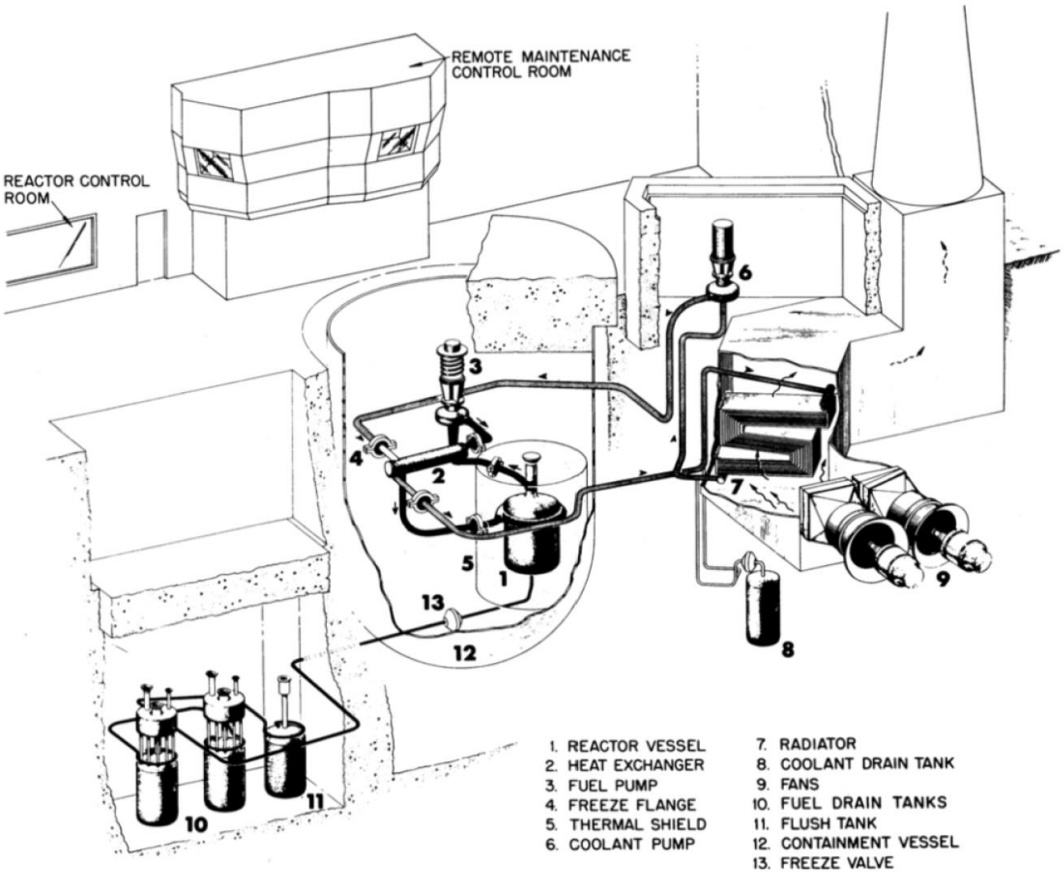


Figure 1: MSRE layout "Reprinted from [3]".

Figure 2 depicts the salt entering from the sides of the lower dividing plenum and flowing radially inward. Upon flowing inward, salt is discharged into vertical passages machined into the graphite matrix. The salt meets in the upper combining plenum and exits through the central fuel outlet header. The experiment produced a multitude of valuable data that paved the way for current research and supplied sound foundation to conduct successful verification and validation studies.

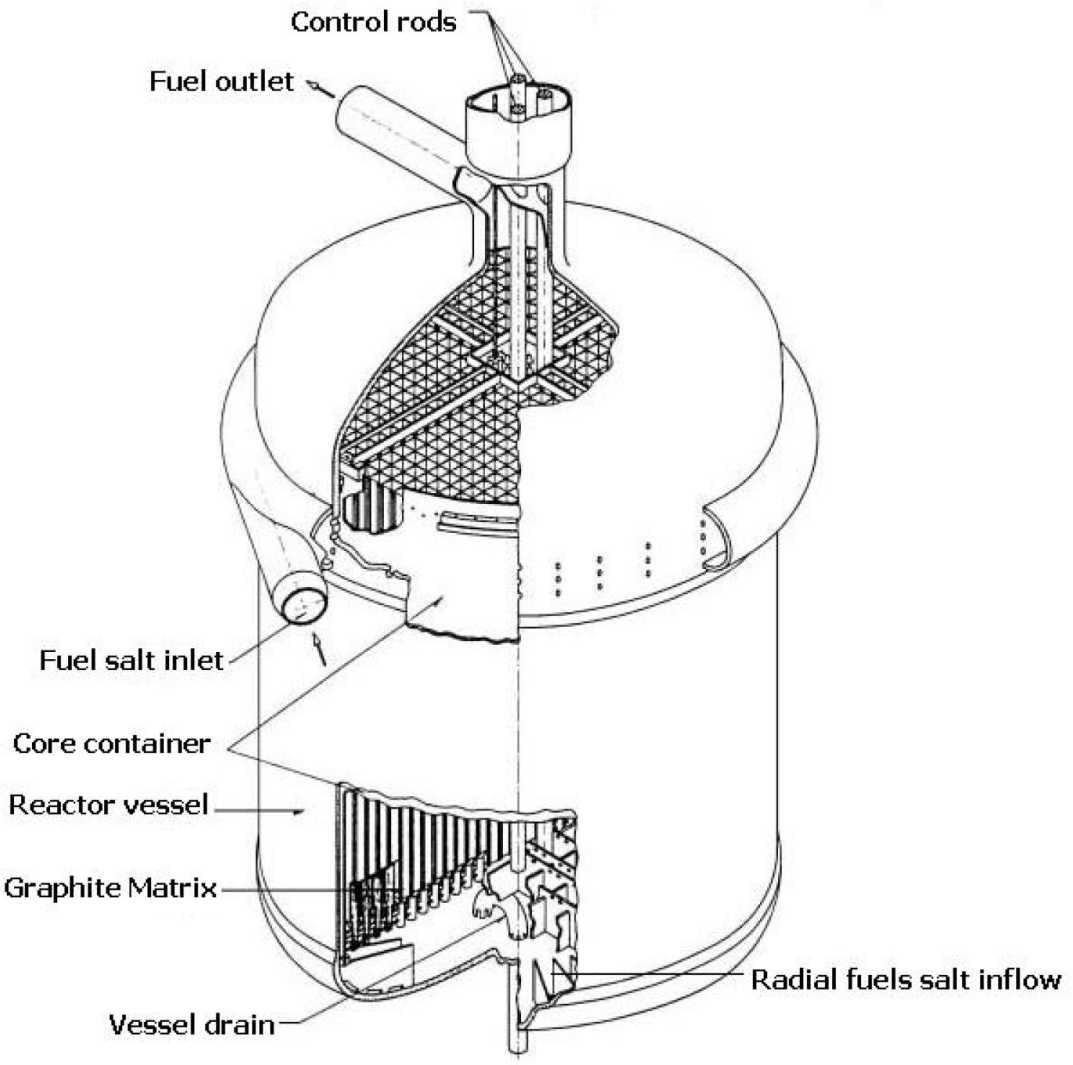


Figure 2: Overview of the core structure of the MSRE "Reprinted from[4]".

The MSRE characterized the practicality of the high temperature molten fuel concept that seemed very promising, having agreeable materials compatibility and fuel cycle cost. While government funding virtually disappeared for Molten Salt Reactors MSRs many decades ago, the success of the MSRE demonstration project has paved the way for current efforts along those same lines. Rosenthal *et al*. illustrated the origin of

the molten salt reactor concept and how the potential of MSRs for electricity production could be achieved [5].

Nowadays there are many different designs for MSRs, though they primarily split into two categories in terms of the nuclear fuel state. The first is solid fuel such as fluoride salt-cooled high-temperature reactor (FHR) with solid fuel, or pebble fuel (PB-FHR) [6], and the liquid fuel molten salt fast reactor (MSFR), or thermal MSR [7]. Salts considered as a coolant in the MSR concepts described above include lithium and beryllium fluoride FLiBe (66 LiF-34 BeF2 mol%), and lithium fluoride FLiNaK (46.5 LiF–11.5 NaF–42 KF mol%), among others [8].

Liquid salts have become an attractive material for use as heat transfer fluids in several applications such as advanced oil recovery, solar, and nuclear as they possess unique thermal properties. Currently, many heat transfer systems using molten salts as a coolant are under investigation, motivated by the design and operation of the MSRE and the significant design advancements of MSRs were enabled during 1950's & 1960's [9].

Through the 1950's, forced convection experiments for molten salt through a heated pipe were conducted to assess the validity of applying predictive correlations to molten salt flows. Grele & Gedeon [10] and Hoffman & Lones [11] performed fundamental experiments and compared their results of Nusselt number Nu to the Dittus-Boelter correlation, which suggested a misprediction as high as 50% and was particularly prone to large errors for Inconel piping material (see **Figure 3**).

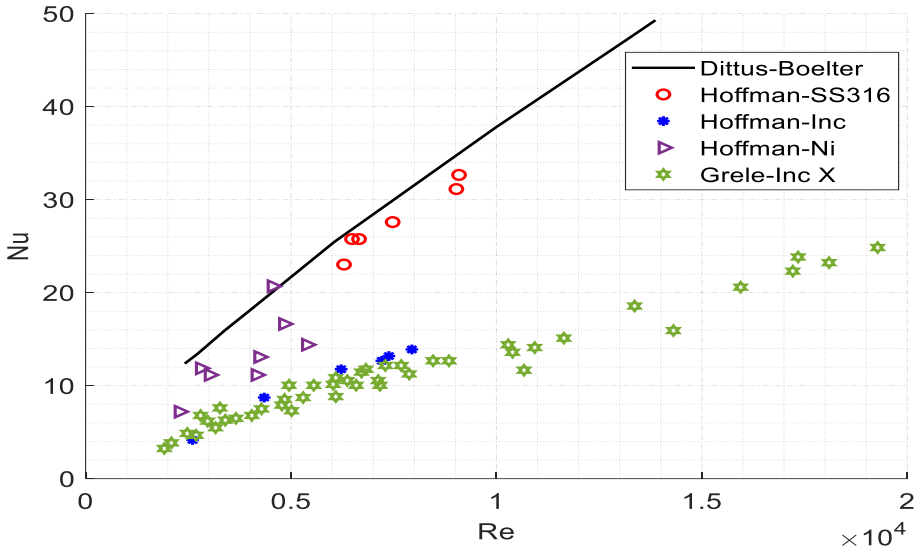


Figure 3: Experimental results obtained in the literature compared to Dittus-Boelter "Reprinted from [11] [10]".

This work was followed by Vriesema [12], then Ambrosek [13], who used Smirnov's correlation [14] for thermal conductivity to reanalyze each of the previous experiments as shown in **Figure 4**. These efforts revealed a better understanding of molten salt thermal properties and the behavior of molten salt in the presence of heating sources.

The growing interest in molten salts along with the experimental investigations have enhanced the research on molten salts. As a result, modeling and simulation have worked essentially to investigate molten salts by looking into system codes such as RELAP or Computational Fluid Dynamics (CFD). The following section will cover the layout of this dissertation, as it will include a detailed sequence of the overall effort performed in preparation of this work.

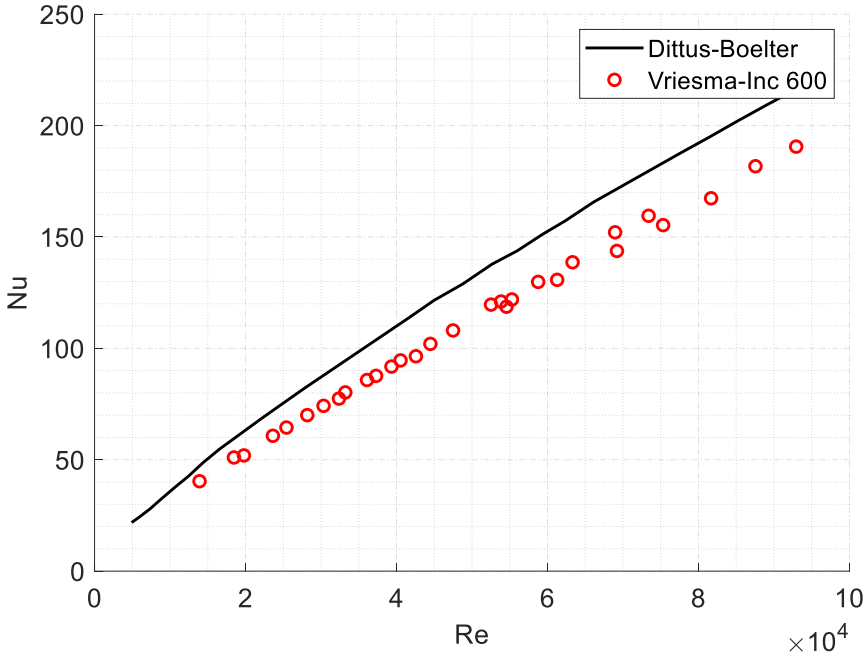


Figure 4: Experimental results obtained by [12] **compared to Dittus-Boelter "Reprinted from [12]".**

Layout & Contributions of Dissertation

Due to the diversity of topics that are covered in this work, each physics subcategory is generally addressed in its own chapter. Chapter II marks up an assessment of turbulence modeling on molten salt flowing in a pipe, while chapter III adds a nuclear aspect to the work done in chapter II, where all the physics are consolidated into a singular Multiphysics analysis framework. Chapter III includes Multiphysics modeling, methodology and results, which discusses the Multiphysics model built in GeN-Foam, where it acts as the primary motivation behind all this work. Chapter IV conducts sensitivity study to quantify input and numerical uncertainty for the work done in chapters II and III. The reader is encouraged to treat each chapter as its own independent

work with unique short introduction, methodology, and results sections. The contributions that this work includes are as follows:

1. Investigate how different turbulence models affect thermal hydraulic metrics in molten salt flows through a heated pipe and quantify input and numerical uncertainties through uncertainty analysis.
2. Utilize a 1D model to validate the point kinetics solver of Multiphysics tool (GeN-Foam) to conduct quick and efficient analysis on thermal MSR transients.
3. Employ the validated 1D model to perform a comprehensive sensitivity study to quantify the impact of input and numerical uncertainties when considering neutronics and thermal hydraulics designs of the reactor core and loop.

CHAPTER II

TURBULENCE MODELING

Fundamental metrics of primary interest in thermal-hydraulic predictions include pressure drop and heat transfer. Historically, experiments focused on flow through standard geometries (e.g., pipes) have enabled 1-D and 0-D systems level codes to capture these metrics. Such codes have been used to design and license most reactors currently in operation. For advanced reactors employing molten salt, conventional wisdom is to simply apply existing correlations covering Prandtl number ranges inclusive of molten salts. However, there is a need to further validate these current models to quantify uncertainties when applied to molten salt flows. Furthermore, additional investigations should be conducted in non-standard flow topologies, the numerical contribution of which would likely require computational fluid dynamics (CFD).

For both aforementioned needs, turbulence modeling and choice of model can greatly affect the thermal hydraulic metrics of highest importance such as the length required for the flows to reach fully developed conditions. Ferng *et al*. investigated the turbulent thermal-hydraulic characteristics of FLiNaK salt using CFD, and their model was tested by comparison with existing correlations and experimental data of friction factor (f), (Nu), and hydraulic and thermal entrance length (l_h, l_{th}) [15].

As expected, the friction factor trend with respect to streamwise direction revealed an initial sharp decrease at the pipe entrance. Soon after this a local minimum was observed, followed by what can be described as an oscillatory convergence to the

fully developed value sufficiently far downstream. They determined the fully developed length to be where the friction factor approached 99% of the fully developed value. Given the fact that this could occur at multiple locations for an oscillatory behavior, further consideration should be made concerning the prediction of this condition. Similar oscillatory behavior was observed in Nu trends as well, suggesting analogous research for l_{th} too. In that particular study, it was also unclear whether the oscillatory behavior was an artifact of the simulation approach since the authors only used a single turbulence model ($k-\epsilon$ two equation turbulence model).

Srivastava *et al*. studied the heat transfer and pressure drop characteristics of FLiNaK salt flowing in a circular pipe with the help of in-house developed CFD code, NAFA [16]. In their simulations, they unsurprisingly found good agreement with existing correlations for fully develop laminar and turbulent flows. They also investigated l_h , which was calculated as the downstream location where $\partial U/\partial x \leq 0.01$. They found over 30% error in l_h with laminar flow and considered the turbulent flow predictability a success since their data was within the large range of 10 to 60 diameters downstream. Although Srivastava *et al*. studied wide ranges of Re, they did not include investigations of heat transfer behavior in the developing region.

Zhang *et al*. performed numerical analysis for both laminar and turbulent flows to investigate thermal and hydrodynamic performance for multiple molten salts using STAR-CCM+ and verified the model using experimental data and convective heat transfer correlations including Dittus-Boelter, Gnielinski, Hausen, and Sieder-Tate correlations [17]. Similar to Srivastava *et al*., they found errors in the order of 30% for

laminar flow predictions, where errors of a similar magnitude were found for turbulent flows too. They used a single turbulence model ($k-\epsilon$ two layer), but unlike Ferng *et al*., they did not observe the oscillatory convergence of f or Nu to the fully developed value. The need for further investigations remains, as Zhang *et al*. employed a slightly different mathematical condition to define l_h.

While it is promising that these two studies [18], [19] share aspects of multiple conclusions, this work aims to provide a more thorough treatment of this topic by studying the impact of choice of turbulence models on the thermal-hydraulics metrics for FLiNaK flowing in a heated Inconel circular tube. The heat transfer coefficient h, its dimensionless form Nu, and l_h, l_{th} are the primary metrics of interest. The resulting Nu spread is compared over five different turbulence models to give better insight into the ramification of the modeling assumptions under the same flow and heat transfer conditions. Also, the l_h is compared over five different Re values in order to get a better insight of the salt's hydrodynamic and thermal developing characteristics.

Methodology

3D Navier-Stokes equations are able to describe turbulent flows completely. However, using the Navier-Stokes equations to resolve all scales of motion directly would require very large computational costs, even for relatively simple flows. One common way to overcome that is to describe turbulent motion in terms of averaged overall scales of motion instead of instantaneous values, which greatly reduces the computational cost. However, even under these simplifications, additional terms show up in the momentum and energy equations, requiring some treatment since mathematical

closure is no longer achievable. Models are formulated to provide the missing information from the time averaged equations by providing additional transport equations for some or all six independent Reynold stresses.

The continuity, momentum, and energy conservation equations are solved for an incompressible fluid, which under steady RANS formulations are expressed as shown in Eqs. 1-3.

$$\nabla.(\rho\bar{u}) = 0 \quad \text{(Eq.1)}$$

$$\nabla \cdot (\rho\bar{u} \times \bar{u}) = -\nabla P + \mu\nabla^2\bar{u} \quad \text{(Eq.2)}$$

$$\nabla \cdot (\rho\bar{u}E) = -\nabla P\bar{u} + \nabla \cdot (k\nabla T) + S \quad \text{(Eq.3)}$$

In the model, density variation is only due to temperature changes, and incompressibility is valid in this case. The following section entails a more detailed illustration of RANS models utilized in this work.

RANS Models

Turbulence models can be classified by the number of differential equations solved in addition to the mean flow equations [20], zero equation, one equation, two equation, and stress equation models. This study investigates five turbulence models while focusing on one equation and two equation models due to their ability in modeling mean flow quantities, and their availability within the commercial CFD software package Star-CCM+ (v. 14.06.012-R8) [21].

One-equation turbulence model is a type of turbulence model that uses a single transport equation for a variable that represents the intensity of turbulence in a flow field. This variable is usually the turbulent viscosity, and the equation for its transport is

derived based on assumptions and modeling of the turbulent flow physics. While the turbulent viscosity is a physical property that describes the actual effect of turbulence on momentum transport, it is not used directly in CFD simulations. The eddy viscosity is a modeling assumption used to represent the effect of turbulence on momentum transport, where eddy viscosity is modeled by $\nu_t = C_\mu k^{1/2} l$, where C_μ is a constant and the turbulent length scale l is flow dependent. Spalart-Allmaras model adopted in this work is an example of a one equation model [22]. Two equation models are prevalent in industry, where they have found success in many engineering applications. These models eliminate the need to specify l as a function of position, and instead employ a second transport equation which estimates it.

One variant of the popular two-equation $\kappa - \epsilon$ model, the realizable $\kappa - \epsilon$ [23] that is essentially designed to deliver a more adequate performance in the purpose of obtaining the mean quantities of complex flows involving strong adverse pressure gradients, flow separation, and secondary flows.

The following sections contain a more detailed illustration for RANS models applied in this work including the specific equations of turbulence quantities derived for each model. The used $k - \omega$ SST model is a modified version of the $k - \omega$ model, which uses a blending function to combine the capabilities of $k - \epsilon$ and $k - \omega$ to develope a widely used robust turbulence model. Another variant of $k - \epsilon$ model is $v^2 f$, which incorporates near wall turbulence anisotropy that avoids the use of wall functions and makes the model valid at the wall, precisely with the inclusion of non-local effects at the

wall region that makes the model superior over many models in the calculation of near wall behavior such as wall shear stress and heat transfer [24].

Spalart-Allmaras Model

Spalart-Allmaras is built for aerospace applications which includes the parameters related to space or aero bodies like airfoil [25]. However, various research has utilized the model for internal flows [26], [27]. As mentioned before in this chapter this model requires the solution of an equation for the turbulent kinetic energy k, and it solves a modeled transport equation for eddy viscosity $\mu_t = \rho\breve{\nu} f_{v1}$. In addition, it doesn't require calculating l related to the shear layer thickness as l is flow dependent.

The variable of interest in Spalart-Allmaras is $\breve{\nu}$, where it is incorporated for the spots with low viscous effects such as near wall region. The equation used for the model is shown in Eq. 4. The Spalart-Allmaras model was selected for this study due to its established effectiveness in simulating internal flows. In addition, it has the potential to significantly reduce computational costs if it proves to be valid for this application.

$$\frac{\partial(\bar{\rho}\cdot\breve{\nu})}{\partial t}+\frac{\partial(\bar{\rho}\,\breve{\nu}\,u_i)}{\partial x_i} = C_{b1}\left[S+\frac{\breve{\nu}}{k^2d^2}\left(1-\frac{X}{1+Xf_{v1}}\right)\right] + \frac{1}{\sigma_v}\left[\frac{\partial}{\partial x_i}\left((\mu+\rho\nu)\frac{\partial\nu}{\partial x_j}\right)+C_{b2}\rho\left(\frac{\partial\nu}{\partial x_j}\right)^2\right]-Y_\nu \quad \text{(Eq.4)}$$

Per Eq. 4, the damping function f_{v1}, Y_ν destruction of viscosity, and normalization of eddy viscosity X are expressed as follows per Eq. 5-7.

$$f_{v1} = \frac{X^3}{X^3 + C^3 v1} \quad \text{(Eq.5)}$$

$$Y_\nu = C_{\omega 1} \rho g \left[\frac{1 + C_{\omega 3}^6}{g^6 + C_{\omega 3}^6}\right]^{\frac{1}{6}} \quad \text{(Eq.6)}$$

$$X = \frac{\breve{\nu}}{\nu} \quad \text{(Eq.7)}$$

The Spalart-Allmaras model adopted in this work is identical to the one from literature, where the reader should refer to the paper by Spalart and Allmaras for more details on the model coefficients[22].

$k - \epsilon$ model and the realizable variant

Realizable $k - \epsilon$ model differs from the standard $k - \epsilon$ model in two ways. Firstly, it contains a new formulation for the turbulent viscosity ν_t, such that the constant of proportionality in the expression can take on a local value instead of one global value. The second difference is a revised transport equation for the dissipation rate, ϵ, that is derived from an exact equation for the transport of the mean-square vorticity fluctuation. Eq. 8 & Eq. 9 show the two transport equations for k and ϵ, respectively.

$$\frac{\partial}{\partial t}(\rho k) + \nabla \cdot (\rho \bar{u} k) = \nabla \cdot \left[\left(\mu + \frac{\mu_t}{\sigma_k}\right) \nabla k\right] + P_k - \rho\epsilon \quad \text{(Eq.8)}$$

$$\frac{\partial}{\partial t}(\rho\epsilon) + \nabla \cdot (\rho \bar{u} k) = \nabla \cdot \left[\left(\mu + \frac{\mu_t}{\sigma_\epsilon}\right) \nabla\epsilon\right] + \frac{\epsilon}{k}(C_{\epsilon 1} P_k - C_{\epsilon 2}\rho\epsilon) \quad \text{(Eq.9)}$$

Isotropic turbulent viscosity μ_t is introduced based on the Boussinesq hypothesis, which is used to close the momentum equations, where it accounts for the additional mixing

(momentum transfer) provided by the turbulent fluctuations. The μ_t is expressed as follows:

$$\mu_t = \rho C_\mu \frac{k^2}{\epsilon} \quad \text{(Eq.10)}$$

where C_μ comprises the strain rate S and vorticity tensors W as in Eq. 11, and production of turbulence is shown in Eq. 12, and the constants of Eq.12 are listed in Clifford [28]

$$C_\mu = \frac{1}{(4 + \sqrt{6(S:S + W:W)}\cos\zeta)} \quad \text{(Eq.11)}$$

$$P_k = \mu_t S^2 - \frac{2}{3}[\rho k(\nabla \cdot \bar{u}) + \mu_t(\nabla \cdot \bar{u})^2] \quad \text{(Eq.12)}$$

The parameter ξ, as well as the other turbulence model coefficients present in Eq.8 and Eq.9, are defined in **Table 1**.

Table 1: Realizable $k-\epsilon$ model constants "Reprinted from [23]".

$C_{\epsilon 1}$	$C_{\epsilon 2}$	σ_k	σ_ϵ	ξ
$\max(0.43, \frac{\eta}{5+\eta})$	1.9	1.0	1.2	$\frac{1}{3}\cos^{-1}\frac{S \cdot S \cdot S}{\sqrt[3]{S:S}}$

$k-\omega$ Standard Wilcox Model

After $k-\epsilon$ model was developed in 1973, numerous experiments have showed that the model is not accurate in predicting boundary layers with adverse pressure gradients, which means that lift and drag coefficients will be significantly incorrect. This motivated the research development of multiple turbulence models such as Spalart-Allmaras, Johnson -King, and $k-\omega$, where other models see less of an application than $k-\omega$ model. The model was developed in 1988 [29], where it attempts to predict turbulence

by the two partial differential equations for k and specific dissipation rate ω as can be seen in Eq. 13 and Eq. 14. μ_t as needed in two equation RANS models is given by $\mu_t = k/\omega$.

$$\frac{\partial}{\partial t}(\rho k) + \nabla \cdot (\rho \bar{u} k) = \rho P_k - \beta^* \rho \omega k + \nabla \cdot \left[\left(\mu + \sigma_k \frac{\rho k}{\omega}\right) \nabla k\right] \quad \text{(Eq.13)}$$

$$\frac{\partial}{\partial t}(\rho \omega) + \nabla \cdot (\rho \bar{u} \omega)$$

$$= \frac{\alpha \omega}{k} \rho P_k - \beta \rho \omega^2 + \nabla \cdot \left[\left(\mu + \sigma_\omega \frac{\rho k}{\omega}\right) \nabla \omega\right] + \frac{\rho \sigma_d}{\omega} \nabla k \nabla \omega \quad \text{(Eq.14)}$$

The turbulence model coefficients present in Eq.13 and Eq.14, are defined in **Table 2**.

Table 2: k − ω standard Wilcox model constants "Reprinted from [29]".

α	β	β^*	σ_k	σ_ω	ϵ
$\max(0.43, \frac{\nu}{5+\nu})$	1.9	1.0	1.2	$\frac{1}{3}\cos^{-1}\frac{S \cdot S \cdot S}{\sqrt[3]{S:S}}$	$\beta^* \omega k$

k − ω Shear-Stress Transport Model

Menter developed the k − ω SST model [30], which combined the two most popular two equation models, $k - \epsilon$ and $k - \omega$, by multiplying the two models with a blending function, and then taking their sum [31]. Blending is achieved through a hyperbolic function ranging from zero to one. This in turn allows $k - \omega$ to be predominant near the wall, and $k - \epsilon$ to be predominate in the free stream region. This allows the model to avoid the common $k - \omega$ problem of being more sensitive to the

inlet free stream turbulence properties, while also avoiding a common difficulty in the standard $k-\epsilon$ model in the near wall region.

$$\frac{\partial}{\partial t}(\rho k)+\nabla\cdot(\rho\bar{u}k)=\rho P_k-\beta^*\rho\omega k+\nabla\cdot[(\mu+\sigma_k\mu_t)\nabla k] \quad \text{(Eq.15)}$$

$$\frac{\partial}{\partial t}(\rho\omega)+\nabla\cdot(\rho\bar{u}\omega) = \alpha\rho S^2-\beta^*\rho\omega^2+\nabla\cdot[(\mu+\sigma_\omega\mu_t)\nabla\omega] + 2(1-F_1)\rho\frac{\sigma_{\omega 2}}{\omega}\nabla k\nabla\omega \quad \text{(Eq.16)}$$

v^2f Model

Durbin developed the v^2f model in order to capture the near-wall turbulence behavior and nonlocal effects by using an elliptic relaxation methodology [32]. It differs from the widely used two-equation models $k-\epsilon$ and $k-\omega$ in that two alternate turbulence quantities are solved, $\overline{v^2}$, the velocity fluctuation normal to the streamlines, where $v_t^{\overline{v^2}}=C_\mu\overline{v^2}T$, where the T is turbulence intensity. The second quantity is F22, which is called the elliptic relaxation function, and it models the anisotropic wall effects. In addition to the standard k and ϵ equations, the transport and elliptic equations are found as in the following Eqs.

$$\frac{\partial}{\partial t}\left(\rho\overline{v^2}\right)+\nabla\cdot\left(\rho\overline{v^2}\right)=\rho kf-\frac{\overline{v^2}}{k}\rho\epsilon+\nabla\cdot\left[\left(\mu+\frac{\mu_t}{\sigma_{\overline{v^2}}}\right)\nabla\overline{v^2}\right] \quad \text{(Eq.17)}$$

$$l^2+\nabla^2f-f=\frac{C_1-1}{T}\left(\frac{\overline{v^2}}{k}-\frac{2}{3}\right)-\frac{C_2P_k}{\epsilon} \quad \text{(Eq.18)}$$

8)

Where the turbulence length scale l and turbulence time scale T are shown in the Eqs. below.

$$l = C_l \cdot \max\left[\frac{k^{\frac{3}{2}}}{\epsilon}, C_\eta \left(\frac{\nu^3}{\epsilon}\right)^{\frac{1}{4}}\right] \quad \text{(Eq.19)}$$

$$T = \max[\frac{k}{\epsilon}, C_T \left(\frac{\nu}{\epsilon}\right)^{\frac{1}{2}}] \quad \text{(Eq.20)}$$

The following section will provide a description of the geometry that was established to conduct the necessary investigations. This section follows the introduction of the general RANS equation form and a comprehensive explanation of each of the turbulent models used in this work.

Thermal Hydraulic Metrics

One of the most important metrics in thermal hydraulic analysis is Nusselt number. It represents ratio of convective to conductive heat transfer at a boundary in a fluid. In this work, Nu is calculated at the interface between the molten salt and the solid Inconel and is quantified via the following expression:

$$Nu = \frac{h \cdot D}{K_f} \quad \text{(Eq. 21)}$$

Heat transfer coefficient h in Eq. 21 is the only unknown variable, which could be determined at the solid/fluid interface as a conjugate heat transfer boundary condition

is applied on that surface to allow the heat transfer mechanism to switch from conduction to convection.

The absence of a fixed free stream temperature necessities using a mean or bulk temperature, which is simply the integration of radial temperature of the flow that is determined as follows:

$$T(r) = \frac{T_f - T_s}{ln\left({r_f}/{r_s}\right)} ln\left(\frac{r}{r_s}\right) + T_s \quad \text{(Eq. 22)}$$

Where T_f and T_s represent fluid surface temperature and the solid temperature near the surface respectively. Similarly, r_f and r_s are the radii of the fluid and solid at the interface and near the surface (the first solid cell from the surface).

The bulk temperature is calculated as follows:

$$T_{bulk} = \frac{2}{u_{bulk} \cdot r^2} \int_0^r u \cdot T(r) \cdot r dr \quad \text{(Eq. 23)}$$

u_{bulk} is the bulk velocity, which is expressed in Eq. 24 as follows.

$$u_{bulk} = \frac{2}{r^2} \int_0^r u(r, x) r \cdot dr. \quad \text{(Eq. 24)}$$

Now substituting the radial temperature from Eq. 22 into the general form of conduction equation results in the following expression

$$q_r = \frac{2\pi \cdot l \cdot k(T_f - T_s)}{ln\left({r_s}/{r_f}\right)} \quad \text{(Eq. 25)}$$

h is found by solving conduction and convection equations at the interface, which results as follows.

$$h = \frac{k_s(T_s - T_f)}{(T_f - T_{bulk}) \cdot r_f \cdot ln\left({}^{r_s}/_{r_f}\right)} \quad \text{(Eq. 26)}$$

Both Hoffman & Lones [11] and Sohal *et al* [33] proved that radiation can be neglected because temperatures reached are below the range where radiation is known to impact the behavior, and radiation is generally important more important in laminar flows, which is not the case in this study. In addition, the diameter of the pipe in the current case is less than 1cm, which diminishes the effect of radiation too.

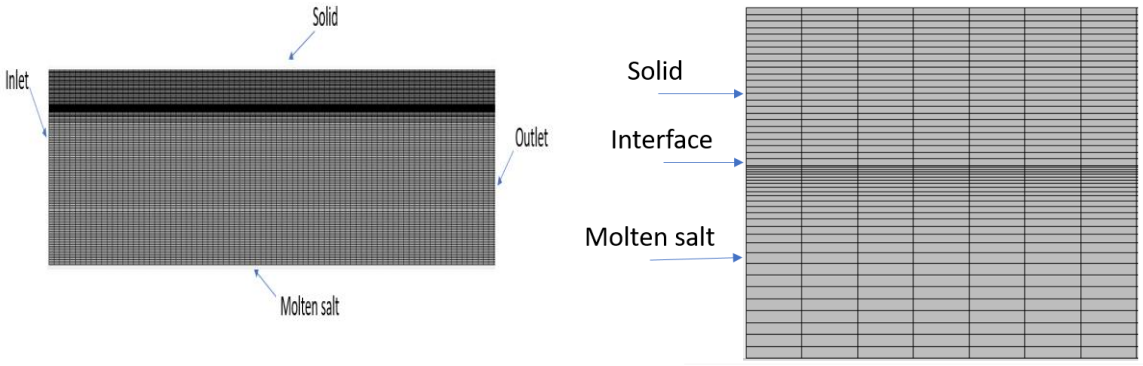
The heat source was calculated depending on the so called "best run" from [34] using the temperatures at the inlet and outlet, as can be seen in Eq. 27, and thermophysical properties for this equation are presented in **Table 6**.

$$Q = \rho \cdot C_p \cdot V_{avg}(T_{out} - T_{in}) \quad \text{(Eq. 27)}$$

The second metric of interest is hydrodynamic entry length l_h. It is considered a crucial parameter for design engineers as it is the variable that determines fully developed region for fluid flow. So far, there is no satisfactory general expression for the entry length in turbulent flow. However, it is known that its value is approximately $10 \leq \frac{X_{fd,h}}{D} \leq 60$. Where $x_{fd,h}$ is the hydrodynamic fully developed length, and D is the pipe diameter. A wide review of experimental studies for turbulent flow developing was conducted by Klein *et al*. [35], where the term blockage factor was introduced to express the flow development along the pipe. This factor is simply the ratio of centerline velocity to the mean stream velocity.

Geometry

A 2D mesh motivated by the experimental test section in Grele and Gedeon experiment [36] has been created using ICEM CFD 2019 R3. **Figure 5** shows the 2D mesh in Star CCM+. The upper region in the figure is Inconel X pipe wall with a 0.375-in and 0.310-in outside and inside diameter, respectively, and the lower region presents the molten salt flowing through the Inconel pipe [10].



(a) (b)

Figure 5: (a) 2D representation of the computational domain, and (b) illustration of refined mesh near the solid-fluid interface.

Table 3 shows the number of elements for solid and fluid in the computational domain. A growth ratio of 1.1 from the wall was applied in the fluid region, while the first cell size is set to 1E-05 m. This refinement in the fluid region near the wall is made so that the model could resolve the equations for the temperatures and velocities up to the viscous sub-layer.

Wall y^+ is measured to be 0.625, as y^+ is a non-dimensional distance (based on the velocity of the fluid in local cell) from the wall to the first mesh node. In order to ensure that results are not sensitive to y^+, Grid Convergence Index study (GCI) is conducted,

and the results are discussed in Chapter IV. The aspect ratio might concern the reader while looking at **Figure 5. b**, this concern is addressed and discussed in Chapter IV.

Table 3: Number of elements in the computational domain

Region	Radial	Axial
Solid	24	2098
Fluid	74	2098

The geometry created in ICEM CFD was in 3D structure as it allows creating blocks for meshing purposes. However, the mesh was converted to 2D when imported to Star CCM resulting in a reduction of the total number of elements from ~5,500,000 cells down to 205,506.

In the simulations, Star CCM+ utilizes a finite volume approach that solves the flow field iteratively using SIMPLE algorithm [37] to link the pressure and velocity fields, where it uses a second order upwind scheme for momentum and energy equations.

Boundary Conditions

The hydrodynamic and thermal boundary conditions applied to the fluid and solid regions in the domain are standard conditions for internal pipe flow. Uniform velocity and temperature distributions are set at the inlet for boundary conditions, and the pressure is fixed at zero for the outlet boundary condition. The inlet velocity varied from 3.0539 m/s to 6.1903 m/s, in order to match part of Re range in the literature.

The inlet temperature for all simulations is set at 989 K, which falls in the range of temperatures for MSRs being developed nowadays, and the pressure is 1 atm. A

uniform heat flux of 0.637 MW/m^2 was applied on the upper surface of the solid, which serves the purpose of including heaters around the primary cooling pipes in MSRs. The values for temperature and heat flux were pulled out from Grele and Gedeon [10]in order to be consistent with the work done by [8] Freile and Kimber as they succeeded in capturing the experimental data within the range of simulations the conducted. Fluid boundary conditions are presented in **Table 4**, while solid boundary conditions are shown in **Table 5**. In this work, the default solver and discretization options found in StarCCM+ were used, where the default solver is the Pressure-Based Solver (PISO). As for the discretization option, a second-order accurate spatial discretization is used.

Table 4: Hydrodynamic and energy boundary conditions for fluid.

Fluid B.C.	**Inlet**	**Outlet**	**Wall**
Hydrodynamic B.C	Velocity Inlet	Pressure outlet	No-slip condition
Energy B.C	T_{in} = 989K	Adiabatic	Interface

Table 5:Energy boundary conditions for solid.

Solid B.C	**Exterior walls**	**Interior walls**	**External surface**	**Internal surface**
Energy B.C	Adiabatic	Heat flux (undefined)	Heat flux (calculated)	Conjugate heat transfer

Temperature dependent thermophysical properties employed in this work are shown in **Table 6** and are employed for each turbulence model to remove their impact on the results while still accommodating temperature dependence.

Table 6: Thermophysical properties used in the simulations.

Thermophysical property	Correlation (T in Kelvin)	Reference
Specific heat	$c_p = 1905$	[38]
Density	$\rho = 2729 - 0.73T$	[39]
Thermal conductivity	$k = 0.36 + 5.6^{-4T}$	[40]
Dynamic viscosity	$\mu = (1.1E-4)^{(3379/T)}$	[41]

Results

The main metrics that dictate thermal hydraulic behavior are velocities and temperatures, which are first analyzed in detail. **Figure 6** shows velocity profiles for the realizable $k-\epsilon$ model at multiple axial locations for the lowest Re considered ($Re = 14{,}800$). This begins with a nearly uniform radial profile at $X = 0$, and eventually transitions to a fully developed velocity profile at approximately the $X = 50.8\,D$ location. Note that r/r_0 is a normalized measured of the radial location where $r = 0$ is the centerline of the pipe and $r = 1$ is the wall.

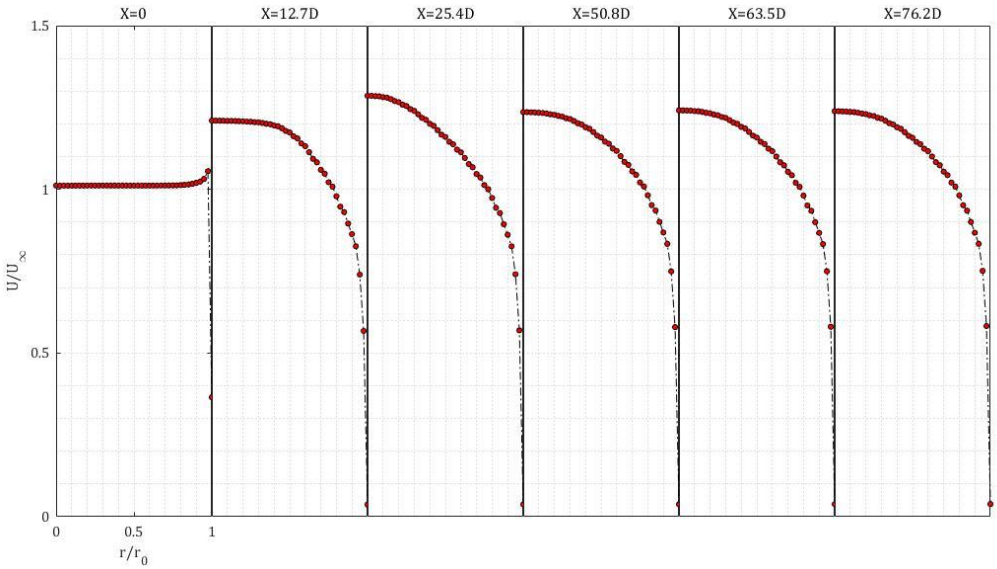


Figure 6: Velocity profiles for realizable $k-\epsilon$ model.

The velocity at the inlet ($X = 0$) shows a constant value $\frac{U}{U_\infty} = 1$, which confirms with uniform velocity inlet boundary condition defined in the section of boundary conditions. Velocity starts developing as the value rises around $X = 25.4D$, then similar velocity profile appears starting at $X = 50.8D$ downstream. As expected, for other turbulence models, similar behaviors are observed, namely a transition from a uniform to a fully develop velocity profile. However, differences do exist when comparing results of one turbulence model to another, especially in the region where flow is still developing. This will be further discussed later in the results.

Also of interest in the current study is the behavior of the thermal development of the flow. Fully developed thermal profiles are achieved when the shape of the temperature distribution no longer changes, and a self-similar condition is met. **Figure 7** provides the temperature profiles at the same axial locations. Similar to **Figure 6**, these

conditions are achieved as the flow progress downstream, where differences between models are greatest in the developing region.

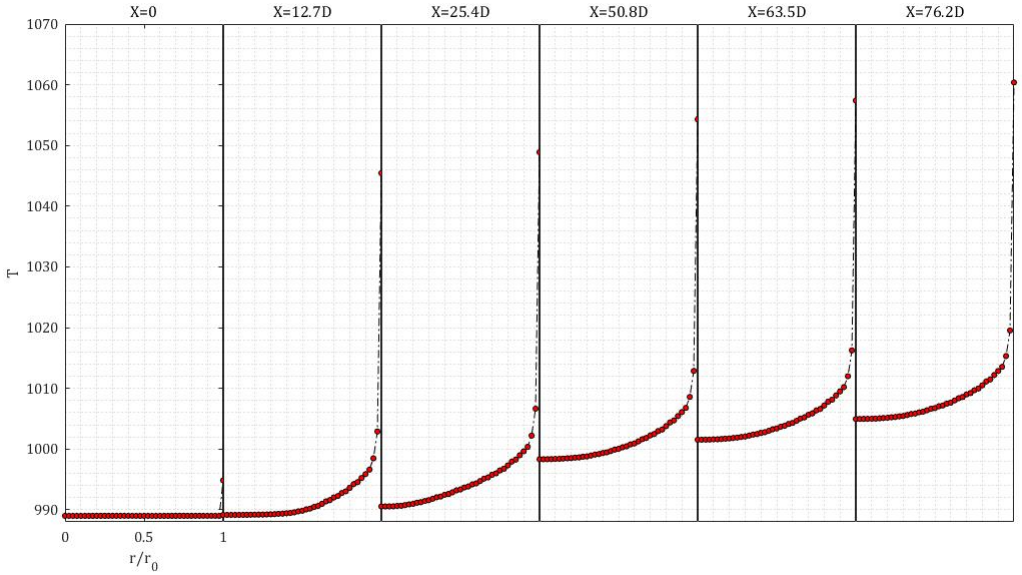


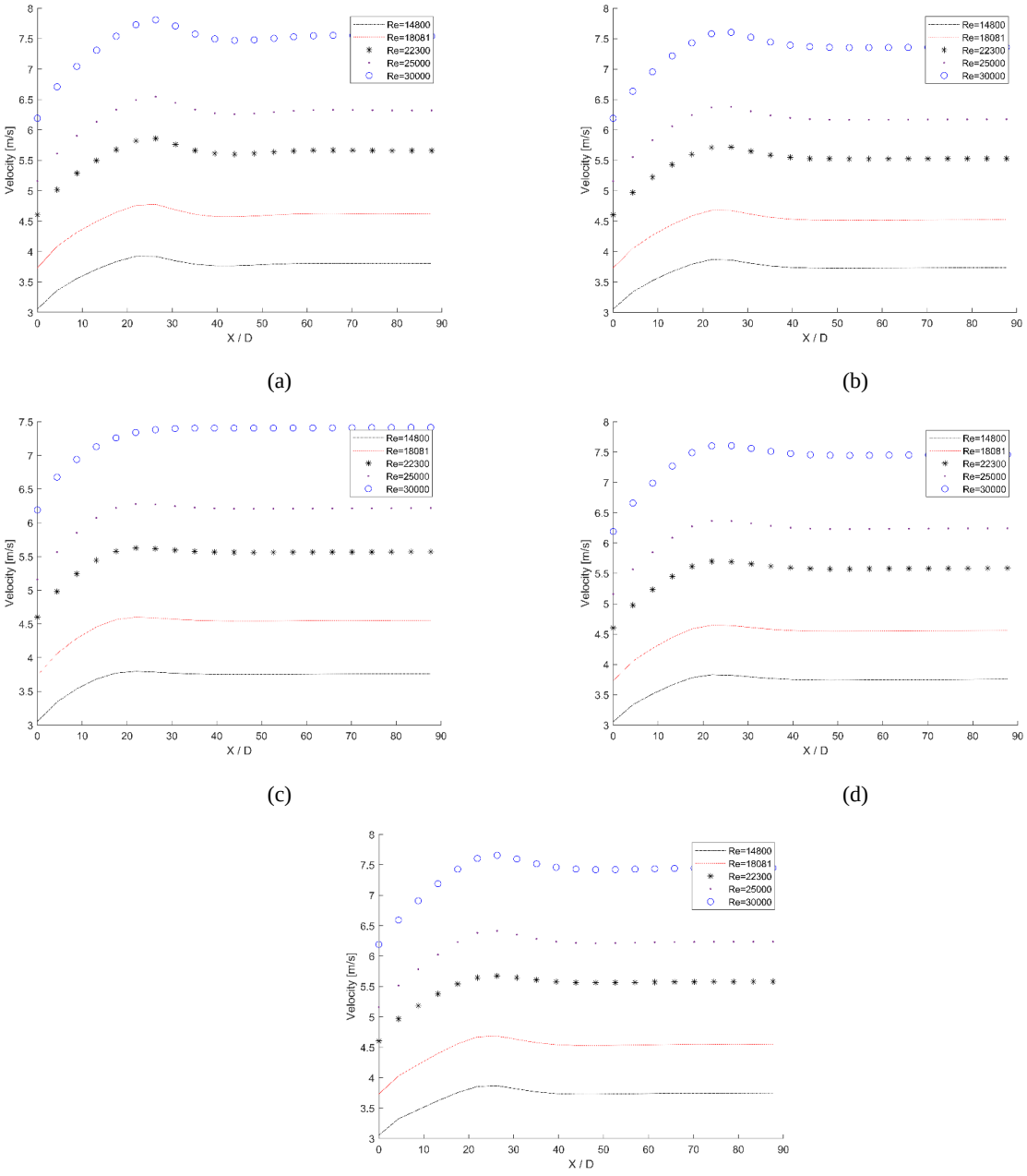
Figure 7: Temperature profiles for realizable $k-\epsilon$ model.

The flow development from a hydrodynamic viewpoint is better illustrated in **Figure 8**, where the centerline velocity is shown as a function of axial position. The location at which the centerline velocity reaches a maximum is an important metric to compare between turbulence models. While differences do exist, this peak occurs between 20D < X < 30D for nearly all Re values of each model. Also, for a given turbulence model, there seems to be little dependence on the Re value itself. In other words, the results are more sensitive to the model chosen than to the Re magnitude. That is not entirely surprising since the low end of the Re range considered (Re = 14,800) is roughly five times larger than where one would expect a transition from laminar to turbulence to occur. Therefore, if the flow physics are well beyond that transition,

similar results are expected as long as the high end doesn't approach the regime where compressibility starts to play a role.

The difference observed between turbulence models is then primarily due to the assumptions tied to that particular model. It is interesting to note that for such a fundamental scenario (i.e., turbulent flow through a pipe), the peak in centerline velocity is only predicted within 20-25% when considered all turbulence models. It is also worth noting that for the largest Re value using the $k-\omega$ model (top curve in **Figure 8 (c)**), the peak seems to have disappeared altogether.

This peak is mentioned in the literature [42] as velocity overshooting and it reflects the flow tendency to laminarization. It can be reduced by increasing turbulence intensity, which is not the scope of this study. Peak disappearance in the case of $k-\omega$ model shows how the model fails for high Re values, where the model can be used only for flows with low Re where the boundary layer is relatively thick, and the viscous sublayer can be resolved.



(a)

(b)

(c)

(d)

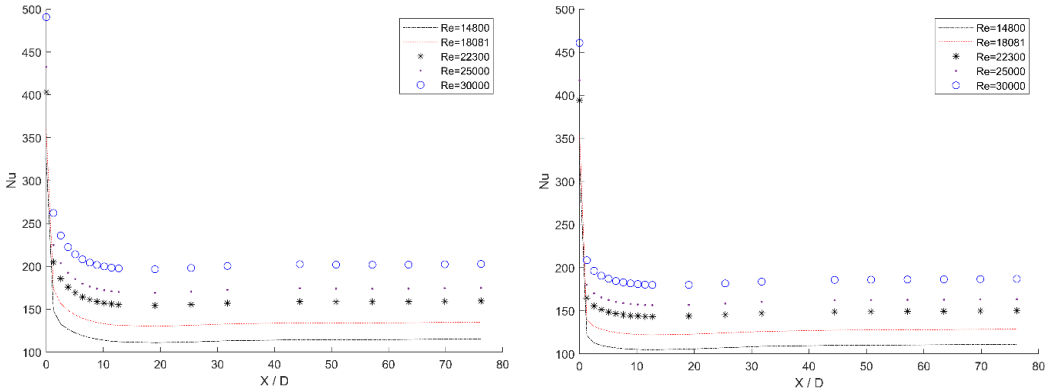
(e)

Figure 8: Centerline velocity vs X/D for five different models, (a) k-ϵ, (b) k-ω SST, (c) k-ω, (d) v^2f, (e) SA.

Nusselt number is one of the key design parameters to be considered in the case of internal flow as it quantifies the ratio of convective to conductive heat transfer at the boundary. **Figure 9** shows Nu behavior along the heated pipe for FLiNaK. While differences do exist between the models, Nu ranges between (450-500) at the inlet, which falls within X/D=3, then it drops after X/D=15 to a value that ranges between (100-220) near the outlet, except for v^2f model.

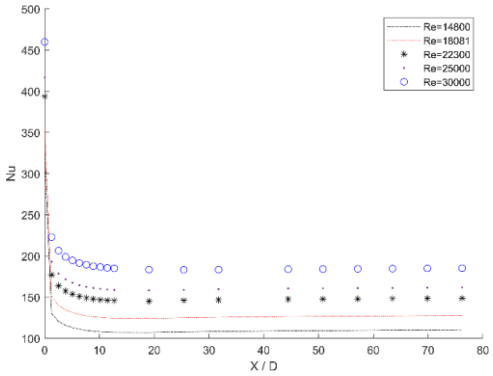
The decay trend of Nu at the entry region X/D <15 is a result of thermal boundary layer growing with downstream distance. As the salt flows downstream, boundary layers merge, and the fully developed condition is reached, causing Nu to be invariant along the pipe. **Figure 9** also reveals that the fully developed Nu is predicted to increase with increasing Re number due to the proportionality between Nu and Re in Dittus Boelter equation.

It is worth noting that Nu in v^2f shows a dip near the entry region unlike the other models. In addition, the lowest Re in v^2f model has scored Nu value below 100. A similar behavior was found in the literature [43] and the difference recorded by v^2f was due to the treatment of length scale l, where the authors stated that $k-\epsilon$ unlike v^2f artificially specified l in its equations. This refers to the difference observed between turbulence models that is primarily due to the assumptions tied to that particular model.

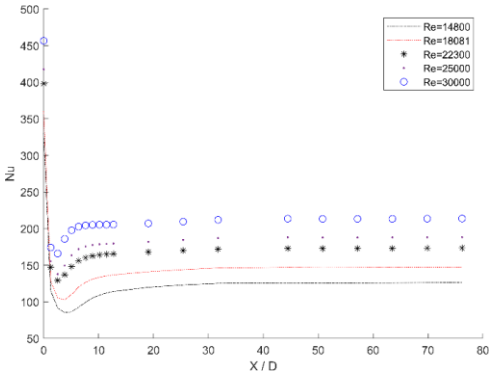


(a)

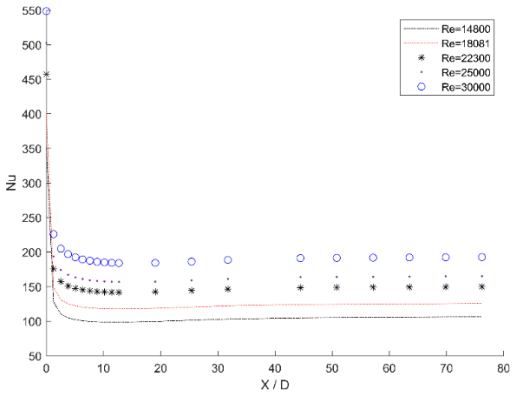
(b)



(c)



(d)



(e)

Figure 9: Nusselt number along the axial direction for different inlet Re number, (a) k-ϵ, (b) k-ω SST, (c) k-ω, (d) v^2f, (e) SA.

Realizable $k-\epsilon$ acts as expected, because it is generally suitable for high Re values, in other words it is appropriate for simulation of free flow as it relates the rate of kinematic dissipation to the turbulent viscosity. $k-\omega$ and $k-\omega\ SST$ behavior is almost identical for all Re. Spalart-Allmaras scored the highest Nu value at the entry region, however, the value of Nu downstream fill within the same range of other models.

The flatline behavior in both, **Figure 8** and **Figure 9** indicate that fully developed conditions from a hydrodynamic and thermal perspective has been reached, where **Figure 9** shows that the thermal entrance length l_{th} is ranging from 10D to 15D. The thermal entrance length for $k-\epsilon$ model is provided with respect to Re in **Table 7**. Their values vary across the different turbulence models and show small dependence on Re as well. The predicted values in this table were calculated based on the equations in [44].

Table 7: Thermal entry length (l_{th})

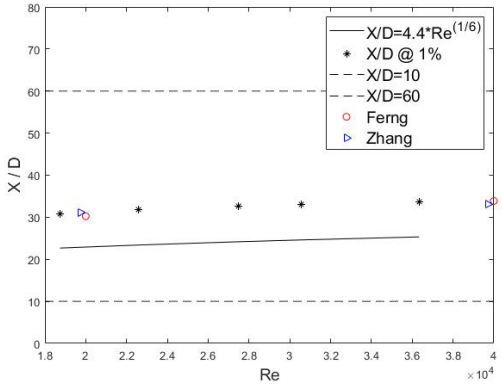
Re	Thermal entry length
14800	11.8
18081	12.1
22300	12.9
25000	13.2
30000	13.5

The hydrodynamic entry length (l_h) was studied thoroughly in three stages, where the first stage was the most common method adapted in literature, in which l_h is found by computing the value that represents less than a 1% change compared to the

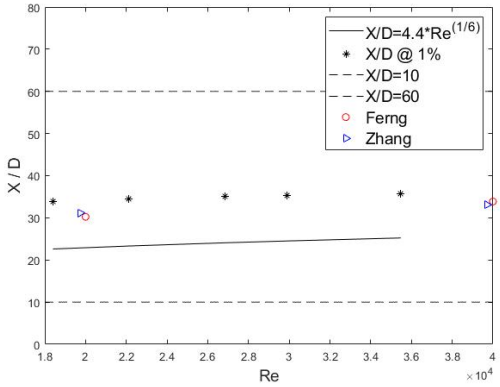
value at x = L,($(l - l_\infty)/l_\infty < 1\%$), which meets the criterion in [18], [19]. **Figure 10** shows the spread of l_h along the pipe as a function of Re. The figure shows that l_h falls in the range of 30D < X < 38D except for $k - \omega$, where the model expectedly underestimates the value of l_h as $k - \omega$ struggles to capture the bulk flow behavior because of the higher Re away from the wall. This is consistent with previous observations shown in **Figure 8 (c)** where $k - \omega$ is seen to perform unlike the other models especially at higher Re. It's seen that the trend l_h conforms to $4.4 \cdot \mathrm{Re}^{1/6}$ unlike the other models.

The other models showed a similar behavior in predicting l_h and the values were in good agreement with the literature [18], [19]. This similarity is anticipated because the adopted method in this work is identical to the literature where $k - \epsilon$ was used. Figure 8 (a) demonstrates this best where it can be seen that $k - \epsilon$ results in this work match best with the literature. Considering the similarity between $k - \epsilon$ and $v^2 f$, the results of both models are expected to yield comparable results.

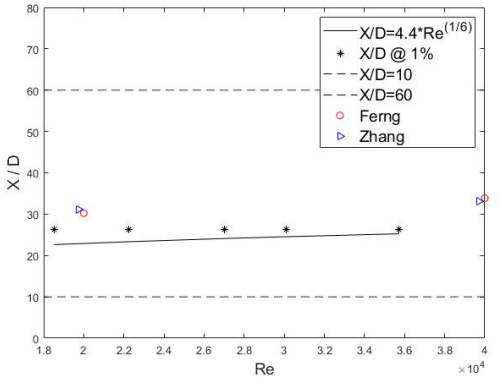
In order to accurately predict l_h, the use of standard deviation is proposed as a second stage of this investigation. In this work, the standard deviation of the velocity is computed using two points in space with a set distance between them (~3.5 hydraulic diameters). This approach is deemed more accurate in predicting l_h, due to the oscillatory behavior of the velocity. The velocity values at certain points might be within 1% with respect to the velocity at the end of the pipe $X = L$. This is going to be proven later in this work.



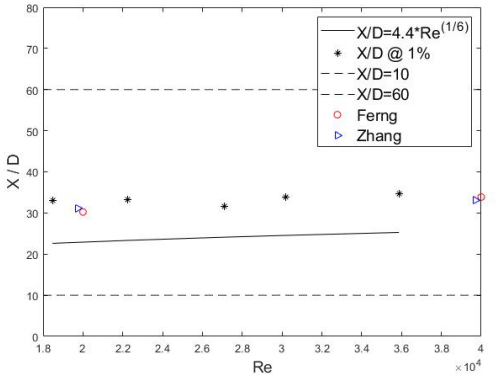
(a)



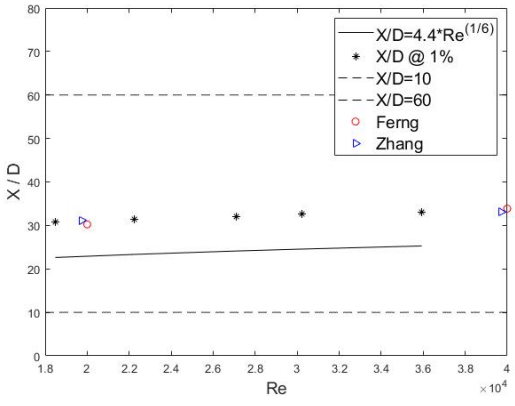
(b)



(c)

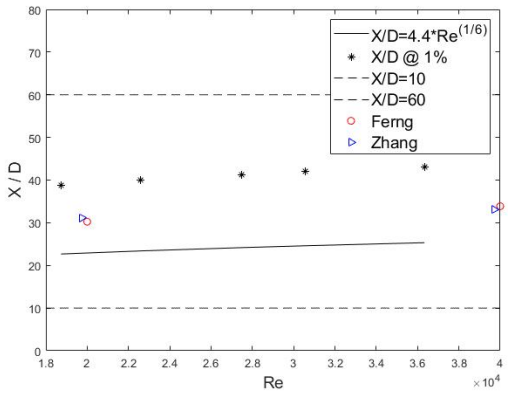


(d)

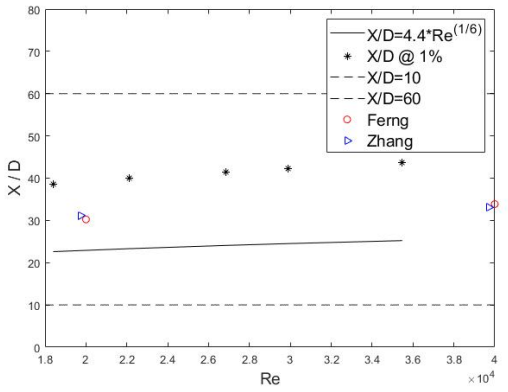


(e)

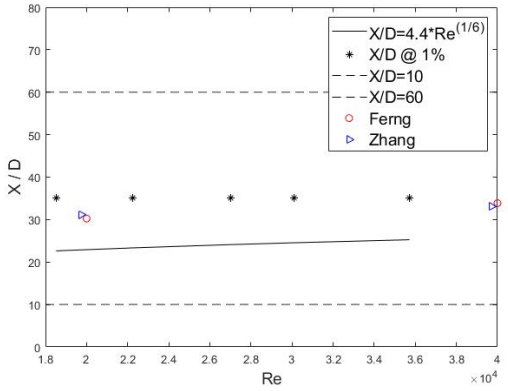
Figure 10: Entry length using criterion in [16, 17], (a) k-ϵ, (b) k-ω SST, (c) k-ω, (d) v^2f, (e) SA.



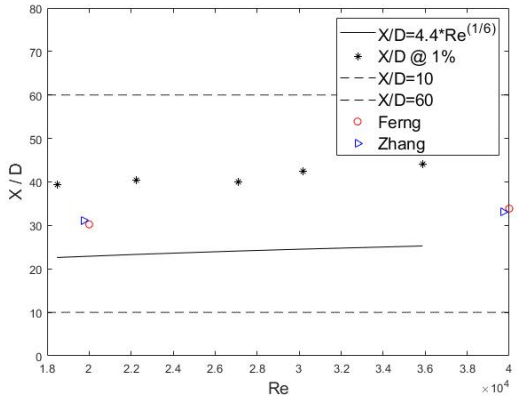
(a)



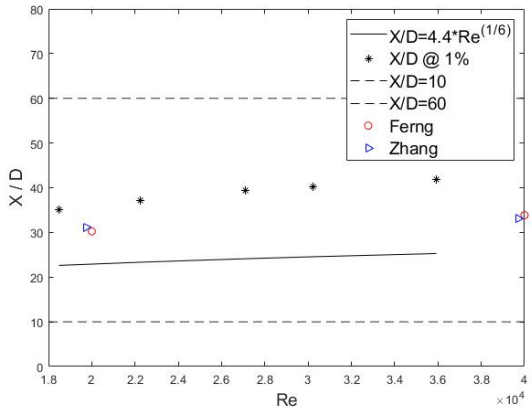
(b)



(c)



(d)



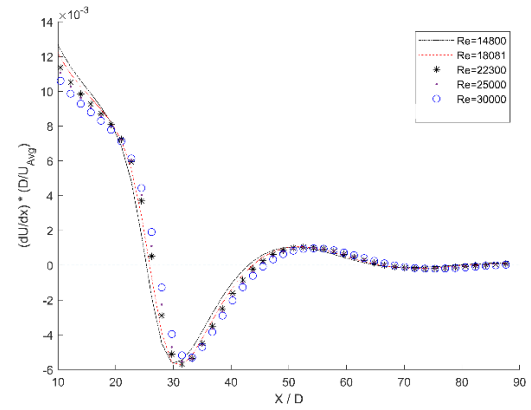
(e)

Figure 11: Entry length using standard deviation, (a) k-ϵ, (b) k-ω SST, (c) k-ω, (d) v^2f, (e) SA.

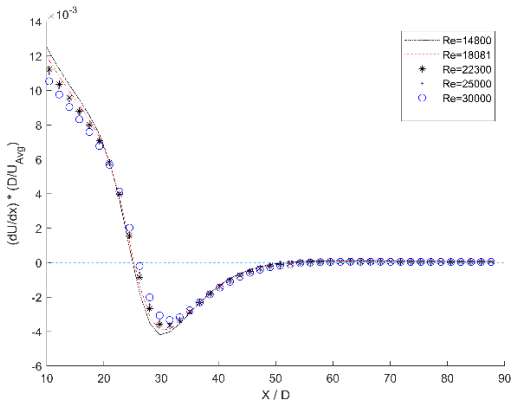
[18], [19]The third and last stage proposed in this work applies alternative fully developed mathematical definition ($\partial \mathrm{U}/\partial \mathrm{x} = 0$), which showed that models behave differently in approaching the fully developed region. Also, the entry lengths were all above the values found in the previous two stages, and they ranged from 45 up to 68 as can be seen in **Figure 12**.

It can be seen that oscillations differ from one model to another, which verifies that each model has a different behavior. The $k-\epsilon$ model predicted the highest value of l_h, while $k-\omega$ expected the lowest. In all cases, changing Re number didn't significantly change the value of l_h(variation was within 2 hydraulic diameters), unlike the previous two stages where the l_h varied between 3 and 7 hydraulic diameters such that higher Re yielded greater variations. **Figure 12** shows that $k-\epsilon$ and the v^2f models record the highest value of l_h, while other models show smaller development lengths and less oscillatory behaviors further downstream in the flow.

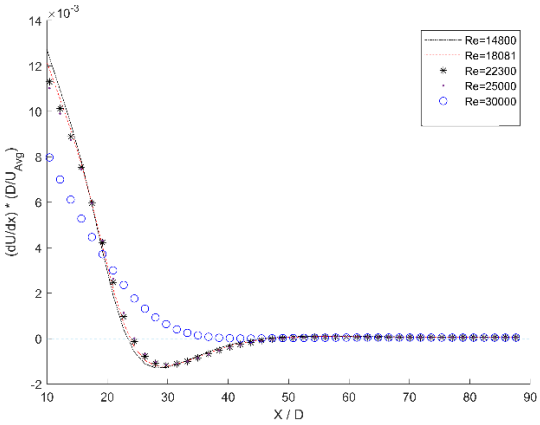
The similar behavior of the two models is because v^2f model is similar to $k-\epsilon$, however, it additionally incorporates some near-wall turbulence anisotropy, and that's why **Figure 12** (d) shows lower perturbations than **Figure 12**(a) of the normalized velocity value. With this analysis the accuracy of predicting $l_h$was reduced two orders of magnitude comparing to the criterion mentioned before in the literature.



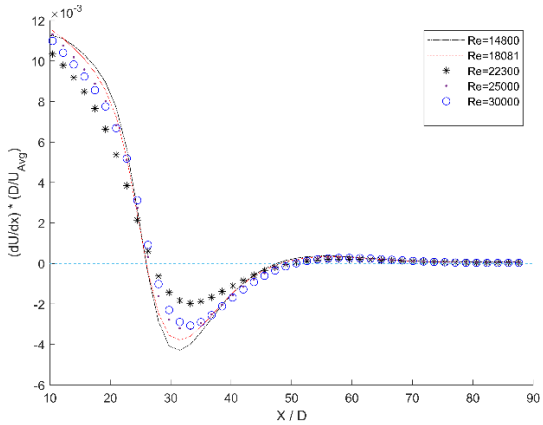
(a)



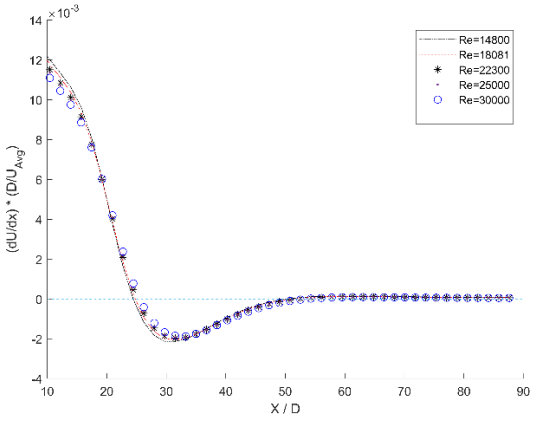
(b)



(c)



(d)



(e)

Figure 12: Entry length using fully developed definition, (a) k-ϵ, (b) k-ω SST, (c) k-ω, (d) v^2f, (e) SA.

Entry length variability among the models indicates that each model predicts different behavior of the molten salt flow, which confuses the determination of entry length. This is considered as an obstacle in the way to predict flow behavior for commercial reactors, as it's challenging to predict the entry length and define the fully developed region for molten salts, it is even harder to predict the developing region for a complicated flow as in the case of a nuclear reactor.

Another way of illustrating the difference among turbulent models concludes in verifying the thermal hydraulic metrics of interest against famous correlations for internal flows from literature [45]. **Figure 13** shows a comparison between the two correlations (Sieder-Tate [46] and Dittus-Boelter [47]) and the turbulent models adopted in this work. Experimental data by Vriesema *et al*. [39] was also plotted in order to validate the results. This analysis relies on a comparison of Nu computed at each Re simulated in this work. For greater generalizability, Nu is divided by the Prandtl number (Pr) per Eq. 28 to yield the fluid-agnostic Nusselt number (Nu^*).

$$Nu^* = \frac{Nu}{\mathrm{Pr}^{0.4}} \qquad \text{(Eq. 28)}$$

The fully developed Nu^* vs Re is presented in **Figure 13** for the five turbulent models and they show a good agreement with Sieder-Tate and Dittus-Boelter.[30] The v^2f and $k-\epsilon$ models overpredict Nu^*, while the other three models were almost matching to Dittus-Boelter for lower Re values but they underpredict for higher Re values. Spalart-Allmaras is slightly unique in that it notably underpredicts the Nu^* at lower Re values but slightly overpredicts at higher Re values (relative to Dittus-Boelter);

however, it followed the trends of v^2f and $k-\epsilon$. For $k-\omega$ and $k-\omega\ SST$, Nu^* spread was closer to Dittus-Boelter correlation in comparison to Sieder-Tate correlation.

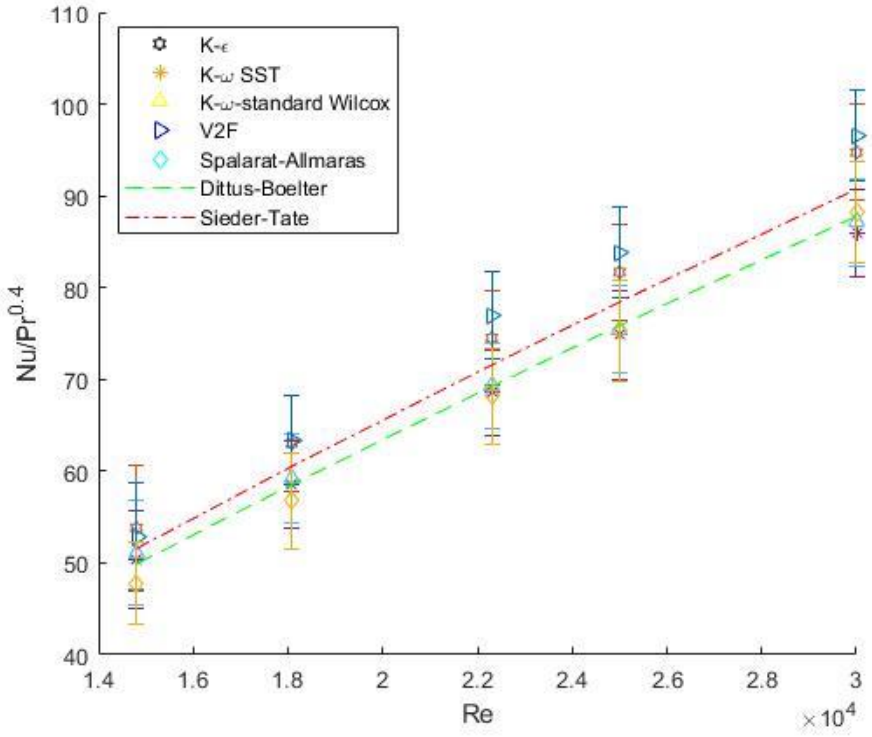


Figure 13: Comparison of Nu^* between turbulent models for the fully developed region.

For the highest Re (3×10^4) on average, the simulation results showed a 0.7% discrepancy when compared to Dittus-Boelter correlation and a 1.1% when compared to Sieder-Tate correlation for $k-\omega$, while results from $k-\omega\ SST$ was 2.1%, and 5.3%, respectively. Spalart-Allmaras model showed for the same Re a difference of 0.5%, and 2.7%, respectively. In the case of $k-\epsilon$, the simulation results showed 7.9% when compared to Dittus-Boelter correlation, and 4.4% difference when compared to Sieder-

Tate correlation. Finally, v^2f showed 9.9% and 6.4% when compared to the two correlations.

This range of discrepancies (1-10%) shows that all results are in a good agreement with the correlations and the experiment, where v^2f model for the highest Re scored the highest difference of 9.9% comparing to Dittus-Boelter correlation. $k-\omega$ and $k-\omega\ SST$ were the closest among the other models when comparing to the correlations. On the other hand, the other models acted differently for different Re values. Some of them were closer to the correlations for low Re such as V2F, and $k-\epsilon$, after that, they drifted away with the increase of Re. The Spalarat-Allmaras results became closer to the established correlations as Re increased, but the slope of its trends may hint that it would overpredict at much larger Re values (outside the scope of this work).

Figure 14 shows that the spread is larger at the entry region than the fully developed region, where the v^2f and $k-\epsilon$ models differed by a maximum of 28%. This is due to the fact that heat transfer is at its highest and the thermal boundary layer is quite thin in the entry region. It should be noted that for many flow geometries and conditions, fully developed analysis might not be appropriate; therefore, the quality of the CFD prediction in the more commonly found developing flow can largely depend on the turbulence model chosen.

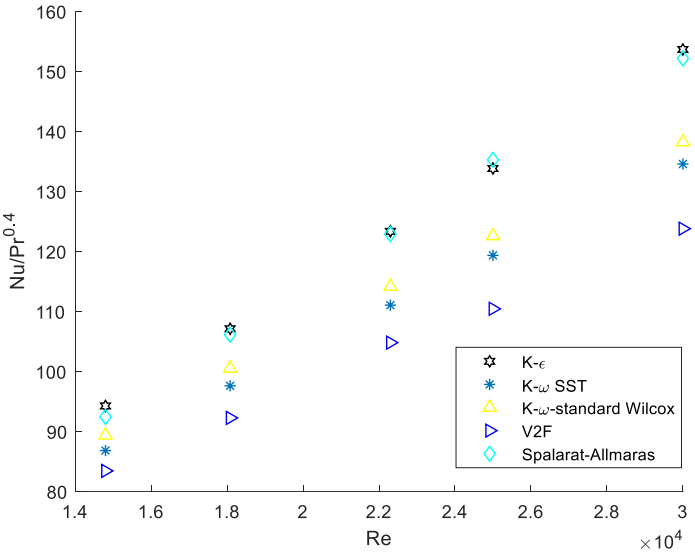


Figure 14: Comparison of Nu^* Nusselt number between turbulent models for the developing region.

The investigation conducted in this chapter represented the data in a dimensionless form except for **Figure 8,** which was presented this way for the purpose of illustration in order to discriminate the overshooting phenomenon in terms of each Re value. The nondimensional representation of the data generalizes the investigated case for any flow in a pipe. In addition, the investigation included one type of molten salts (FLiNaK), which has similar properties to other fluoride salts such as FLiBe. The similarity in thermal properties and the certification of the Department of Energy DOE for these two salts to be used in MSRs motivated a comparison analysis for the properties of the two salts.

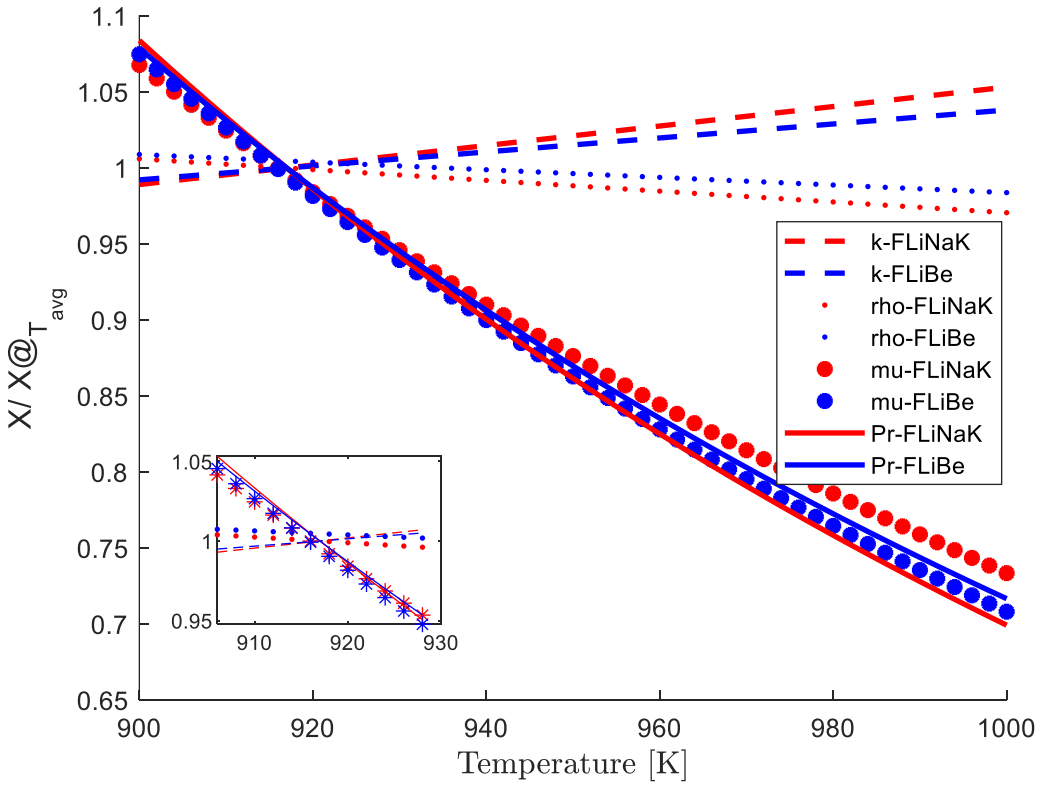


Figure 15: Normalized thermophysical properties for FLiNak and FLiBe.

Figure 15 shows the thermophysical properties for both salts of interest in this work. The plot was extended over the temperature range of interest as the annotation shown in the figure (905K<T<930) represents thermophysical properties over the temperature range in MSRE core. The properties were normalized over each property value found at the average temperature of MSRE core (T~916K).

Based on the comparison provided in **Figure 15**, it can be observed that the thermophysical properties, such as ρ and K, remain relatively constant within a certain percentage range over the given temperature range $\sim 2\% \; and \; 5\%$ respectively. However, μ shows a significant variation over the temperature range, and the Prandtl

number as it has a proportional relationship with μ. As a result, it is possible for future modeling efforts to simplify the analysis by focusing on the variations in μ and Prandtl number while ignoring the variations in ρ and K. Comparison of the properties for both salts shows very close normalized values over the temperature range of interest, which allows to generalize the results for FLiBe too.

Conclusions

All investigated turbulence models showed good agreement with Dittus-Boelter and Sieder-Tate correlations when predicting Nu^* in the fully developed region. In the developing region, the models showed notably high differences in the Nu^* predictions, which may present challenges for picking an appropriate turbulence model for more realistic reactor flow topologies.

The fact that such differences are observed in the developing region for something as fundamental as flow through a round pipe might yield even more discrepancies when more complex reactor geometries are considered. Moreover, when neutronics are included, and the nuclear fuel mixed with the salt is the heat source, which changes thermophysical properties of the salt, the added complexities will require more detailed investigations.

The physics of convective heat transfer fluids is generally well understood. However, because geometry and physical properties are important in convective heat transfer, and the database for molten salts is limited, this is a process that is expected to require some experimental attention in future works. Channel flow design will depend in part on simulation of convective heat transfer processes. In order to determine the

appropriate thermophysical correlations, and to validate CFD models for molten salt, more detailed analysis is needed. The deep investigation on l_h showed that more research is needed at the entrance region, because of the noticeable variability if different models were tested.

The main reason behind using RANS is the approved applicability and efficiency of RANS models in various applications in the industry. RANS requires the lowest computational effort when compared to Large Eddy Simulation (LES) and Direct Numerical Simulation (DNS), and as the study showed, it provides excellent agreement with literature. Consequently, RANS is often the preferred choice for system codes that are currently being developed to include solvers for molten salts. This study was initiated as part of a plan to perform Multiphysics analysis using GeN-FOAM, an OpenFOAM-based solver, to examine how changes in turbulence models can impact the behavior of molten salt.

CHAPTER III

MULTIPHYSICS

The preliminary step done in Chapter II provided a more thorough evaluation of this topic by studying the impact of choosing turbulent models on the thermal-hydraulic metrics for molten salt flowing in a heated Inconel circular tube. The resulting Nusselt number spread was compared over five different turbulent models to give better insight into the ramification of the modeling assumptions under the same flow and heat transfer conditions. In addition, entrance length was compared over five different Reynolds numbers in order to provide a better image of the salt's hydrodynamic characteristics.

The work done in Chapter II proved the applicability of utilizing the conventional Nu correlations (Dittus-Boelter & Sieder-Tate) for molten salts as shown in **Figure 13**. This outcome provided a confidence in implementing Dittus-Boelter/Sieder-Tate in the Multiphysics approach adopted in this chapter.

Multiple innovative concepts of Nuclear Power Plants (NPP) were proposed by the Generation IV International Forum (GIF-IV) in 2002. The forum discussed future production of nuclear energy with multiple benefits in order to give nuclear energy a significant role nationally and internationally [48]. MSRs are considered a potential candidate as they overcome the drawbacks of current Light Water Reactors (LWRs) such as their ambient operating pressure and higher thermal efficiency [49]. As such, various investigations have been conducted to analyze the applicability and liability of these reactors to act as a better replacement for the current fossil fuel plants, which inspired the work presented in this chapter.

Brown *et al*. discussed the importance of thermal hydraulic analysis of liquid fuel MSRs [50]. Afterwards, neutronic phenomena was added to the work, where Brown *et al*. emphasized that modeling is deeply needed for these systems, which will enable the U.S. Nuclear Regulatory Commission (NRC), U.S. Department of Energy (DOE), and industry to identify modeling tools that would be approved in licensing analyses

Rubiolo *et al*. described two main challenges of MSR thermal hydraulics including modeling of molten salt phenomena accounting for coupling between neutronics and thermal hydraulics, and developing experimental techniques for molten salt, mainly, due to the reactivity of the salt and the high temperature, which requires advanced instrumentation [51]. In 2005 Krepel *et al.* [52] developed a 1D code called DYN1D-MSR and validated the data for MSRE at zero power and for natural circulation, then proved the ability of the code to perform analysis for the liquid fuel reactor. In 2007, DYN3D-MSR code was developed and validated against experimental results from MSRE and the 1D version of the code[53].

Cammi *et al*.[54] analyzed MSR primary circuit dynamics with reference to the MSRE. They investigated several simplified models with different complexities varying between 0D and 1D for both delayed neutron precursors and heat convection models. Then these models were compared to a two-dimensional model called Multiphysics model MPM. This work was followed by a 3D model to investigate transient analysis of MSRE[55]. Their approach utilized a Multiphysics code that solves the partial differential equations that represent the different phenomena in the reactor, which requires a huge amount of computational power. The amount of computational power

was reduced by assigning a 3D model for the core, and 0D model for the remaining parts of the system.

Other 3D models [56], [57] were developed with different approaches to investigate the complex and highly coupled behavior of the fuel in MSRs. In [56], DNP drift model for GeN-Foam was verified based on MSRE. Recently, Mattioli *et al*.[49] developed a point kinetics model for fluid fuel nuclear systems and verified the model by comparing to an analytical approach. The point kinetics solver derived by Mattioli *et al*. is utilized in this work.

System codes had their own share of MSRs investigation as well, where Shi *et al*.[58] developed a system analysis code using Reactor Excursion and Leak Analysis Program (RELAP5) suitable for liquid fueled MSRs. Their code was extended and verified by MSRE benchmarks and four transients of Molten Salt Breeder Reactor (MSBR). They also pointed the advantages of liquid fuel system over solid fuel system as follows:

- A large negative temperature coefficient of reactivity because of a high coefficient of thermal expansion.
- The possibility of continuous fission-product removal using physical and pyrochemical process in an online mode.
- More efficient resource utilization by achieving higher fuel burnup than with conventional uranium reactors using solid fuel.
- The reduction of the expense of manufacture of new fuel elements.

ORNL created a model with the Modelica-based modeling library TRANSFORM (Transient Simulation Framework of Reconfigurable Models) [59],

where they compared the model results to MSRE data, and justified the misprediction of gain magnitude in frequency response calculations.

After MSR-specific modeling features utilized in a delayed neutron precursor drift model and a modified point kinetics model were implemented in SAM. Fei *et al*. [60] used SAM to evaluate the precursor concentration in the core as a function of time, and the resulting changes in reactivity. The code was involved in benchmarking simulations based on three MSRE pump related tests. Their results showed good agreement with the experimental measurements and recommended some approaches to achieve better agreement in future work. Recently, Mochizuki [61] used RELAP5 to test MSRE transients over different reactor states and the results were in good agreement with the MSRE. **Table 8** shows previous MSR modeling, part of this table was quoted from Cammi *et al*. [54].

Table 8: Previous MSR modeling summary "Modified from [54]".

Author	Reactor	Geometry	Analysis type/code	Neutronics/fuel and graphite cross-section treatment	Thermal Hydraulics
Lecarpentier and Carpentier [62] **(2003)**	AMSTER	1D	Transient Two-group Cinsf1D	Two-group diffusion theory Homogeneous approach	Prescribed uniform velocity field Empirical correlation for heat transfer fuel/graphite
Yamamoto *et al.* [63] **(2006)**	SMSR	2D, axial-symmetric	Extension of SIMMER-III transient	Two-group diffusion theory Heterogeneous approach	Navier-Stokes equations Empirical correlation for heat transfer fuel/graphite
Krepel *et al.* **(2007** [52]**, 2008** [53]**)**	MSRE and MSBR	3D (neutronics) 1D (thermal hydraulics)	Transient DYN3D-MSR	Two-group diffusion Theory Homogeneous approach	Navier-Stokes equations Empirical correlations for heat transfer fuel/graphite
Nicolino *et al.* [64] **(2008)**	MOSART	2D, axial-symmetric	Transient	Multi-group diffusion Theory Generic cross-sections as a function of the fuel density	Navier-Stokes equations (stream function/vorticity form)

Table 8: Previous MSR modeling summary "Modified from [54]".

Author	Reactor	Geometry	Analysis type/code	Neutronics/fuel and graphite cross-section treatment	Thermal Hydraulics
Kophazi *et al.* [57] (2009)	MSRE	3D (neutronics) 1D (thermal hydraulics)	Transient Neutron kinetics + thermos-hydraulics coupled codes (DALTON + THERM)	Multi-group diffusion Theory Homogeneous approach	Prescribed uniform velocity field Empirical correlation for heat transfer fuel/graphite
Zhang *et al.* [65] (2009a)	Generic thermal MSR	2D, axial-symmetric	Transient	Two-group diffusion Theory Homogeneous approach	Prescribed uniform velocity field
Zhang *et al.* [66] (2009b)	MOSART	0D	Transient	Point-kinetics	Prescribed uniform velocity field Empirical correlation for heat transfer fuel/graphite

Table 8: Previous MSR modeling summary "Modified from [54]".

Author	Reactor	Geometry	Analysis type/code	Neutronics/fuel and graphite cross-section treatment	Thermal Hydraulics
Zhang *et al.* [67] (2009c)	Generic thermal MSR	2D, axial-symmetric	Steady	Two-group diffusion Theory Homogeneous approach	Navier-Stokes equations (RANS/k-ε model)
Cammi *et al.* [54] (2011)	Model MSRE	0D, 1D, 2D	Transient	0D 2D diffusion model for an average channel	Navier-Stokes and energy equations for an average channel Empirical correlation for heat transfer fuel/graphite
Zanetti *et al.* [55] (2015)	MSRE	3D (neutronics) 0D (thermal hydraulics)	Transient	3D diffusion model for a channel	Zero-dimensional. Empirical correlations for heat transfer from graphite to fuel
Shi *et al.* [58] (2016)	MSRE MSBR	1D Nodal analysis	Transient RELAP	Point Kinetics	Single phase flow
Bao *et al.* [56] (2016)	MSRE	3D (neutronics) 1D (thermal hydraulics)	Steady GeN-Foam	3D diffusion model for full core	Navier-Stokes equations (porous/k-ε model) Empirical correlations for heat transfer fuel/graphite

Table 8: Previous MSR modeling summary "Modified from [54]".

Author	Reactor	Geometry	Analysis type/code	Neutronics/fuel and graphite cross-section treatment	Thermal Hydraulics
Wet & Greenwood [59] (2019)	MSRE	1D Nodal analysis	Steady TRANSFORM	Diffusion	Single phase flow
Fei *et al*. [60] (2021)	MSRE	1D Nodal analysis	Transient SAM	Point Kinetics	
Mattioli *et al*. [68] (2021)	MSFR	1D	Transient GeN-Foam	Point kinetics	Navier-Stokes equations (porous/k-ε model) Empirical correlations for heat transfer fuel/graphite
Mochizuki [61] (2022)	MSRE	1D Nodal analysis	Transient RELAP	Point Kinetics	Single phase flow

In fact, developing modeling and simulation capabilities is an important aspect of the Nuclear Energy Advanced Modeling and Simulation program within the U.S. DOE Office of Nuclear Energy (DOE-NE) in the purpose of foreseeing the performance and safety behavior for a variety of nuclear reactor concepts. There has been a focus on assessing whether these advanced tools developed in the program are applicable for modeling steady-state and time-dependent conditions for MSRs with flowing fuel, as a result of the increasing interest in MSRs by the vendor industry, such as Natura Resources, which is currently focusing on designing and building a molten salt research reactor (MSRR) at Abilene Christian University (ACU). This will enlarge molten salt reactor R&D infrastructure and develop a world-class educational center that is capable of supporting US molten salt reactor design and development, which will eventually contribute to commercialize the reactor.

With all the effort employed in the past in order to create different models for MSRs, each analysis faced different issues, and they were mostly either high computational time and power or inaccuracy comparing to the experiments. Also, the readiness of system codes in modeling MSRs is a hurdle for the time being. For instance, at PSI-FAST group, before developing GeN-Foam, they conducted the transient analysis of a reactor in a classical way of coupling legacy codes for neutron diffusion and 1-D or sub-channel thermal hydraulics, which has some limitations, such as the errors from the data exchange interface and the inflexibility of parallel computing. Additionally, thermal-mechanical analysis also was not included in a transient problem. Hence, an

open-source code would be of great importance as current simulations for the new generation of reactors require more advanced tools for more complex problems.

The purpose of this work is to utilize a useful, very simple and easy model to investigate multiple transient cases for MSRs, which will have an added benefit in further advancing the role that CFD plays in licensing efforts of the NRC. A 1D model is created and validated based on the MSRE system data described in ORNL's past reports and recently edited reports for benchmark analysis. This model with the assumptions adopted in it, which will be covered in detail later, can capture the physics of a transient in few minutes and it can produce results that are closer to the benchmarks than any other Multiphysics code both, in value and behavior.

MSRE Components

Core

The core forms the section where fission occurs, where fuel salt enters the flow distributer through the upper side of the vessel with $908.15\ K$, which spreads the salt evenly around the circumference of the vessel flowing downward, then the salt reflects the direction of the flow upward through the graphite channels shown in **Figure 16**.

Graphite forms a matrix of $0.05 \times 0.05 \times 1.7\ m$, and the fuel salt flows within $0.01 \times 0.03\ m$ channels that are cut through the Graphite matrix, which enhances the moderation of fission neutrons, see Figure 17. The core that is $\sim 2.5\ m^3$ total volume has 1140 fuel channels, and these channels form about 0.225 % of the total volume.

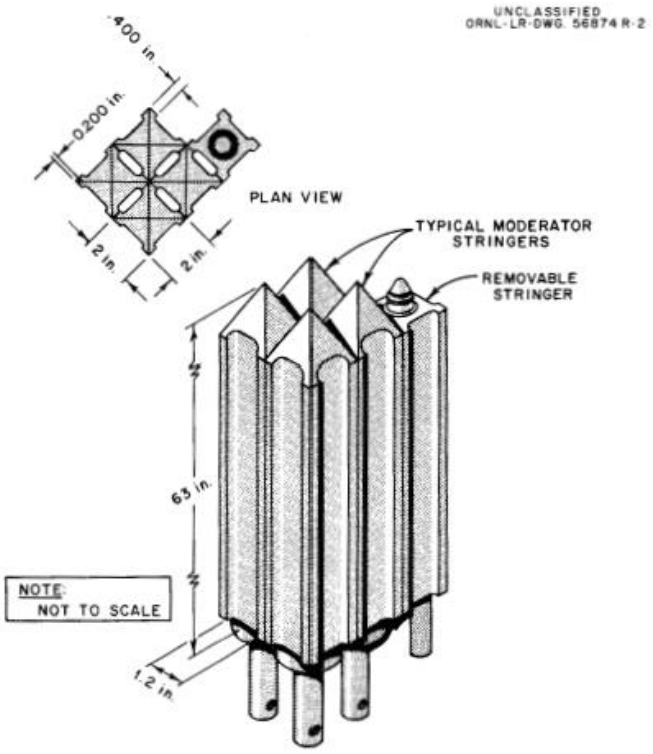


Figure 16: MSRE Graphite matrix "Reprinted from [69]".

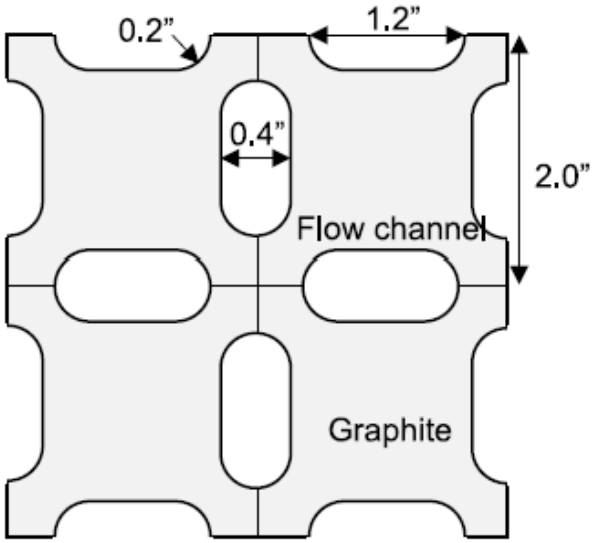


Figure 17: MSRE fuel channels "Reprinted from [61]".

Fuel Pump

The fuel exiting the reactor flows to the centrifugal pump shown in **Figure 18**. The pump operates at a speed of $1160\ rpm$ to deliver $0.0757\ m/s$ with a head of $14.93\ m$ [70]. The diameter of pump bowl is $0.91\ m$, while pump height is 8.4 .

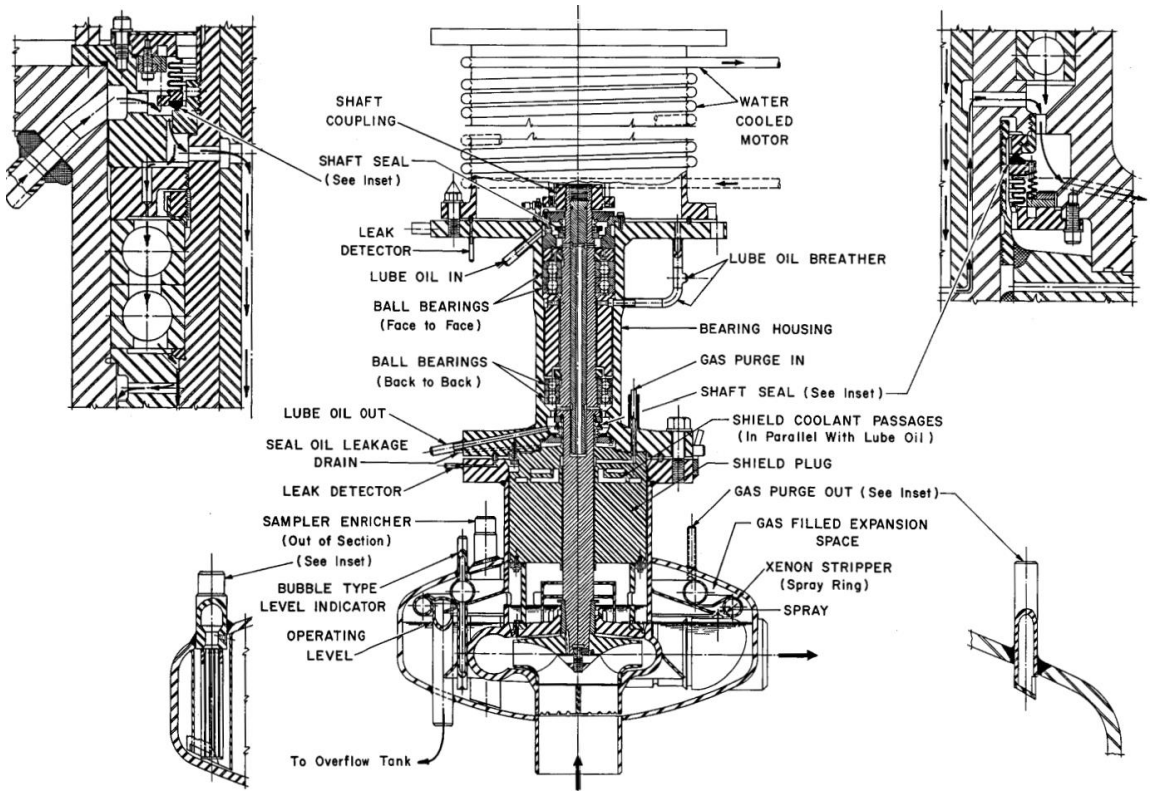


Figure 18: MSRE Fuel Pump "Reprinted from [70]".

MSRE used a unique shaft seal design to prevent the leakage of radioactive material from the reactor core. The shaft seal consisted of a mechanical labyrinth seal and a carbon-graphite packing seal. The labyrinth seal was made up of a series of alternating disks and spacers attached to the reactor vessel and the reactor cover gas plenum. The disks and spacers had grooves that created a tortuous path for the salt to

flow through, which helped to minimize the pressure drop across the seal while preventing the salt from leaking out of the reactor. The carbon-graphite packing seal was used in combination with the labyrinth seal to provide additional leak-tightness. The packing seal consisted of braided carbon-graphite fibers that were compressed against the reactor vessel and the reactor cover gas plenum by a gland nut, creating a tight seal.

A gas purge line was also used for the pump to prevent gas impurities from accumulating in the pump and potentially causing damage. The MSRE had two main types of pumps: the fluoride volatility pumps (FVPs) and the electromagnetic pumps (EMPs). Both types of pumps were designed to operate using an inert gas, such as helium, which was injected into the pump through a gas purge line.

The gas purge line for the pump was designed to prevent gas impurities from accumulating in the pump by continuously flowing the inert gas through the pump while it was in operation. This helped to ensure that any gas impurities were removed from the pump and carried out of the system. The gas purge line for the pump was typically connected to a gas supply system, which supplied the inert gas to the pump at a controlled flow rate.

Heat Exchanger

The heat exchanger serves to transfer thermal energy from the primary salt to the secondary salt. In the heat exchanger, high temperature fluid flows on the shell side of the heat exchanger transfers thermal energy to a cooling salt flowing within a bank of tubes via a combination of convection and conduction. To validate the heat exchanger

model and accurately estimate the required parameters for the simulation of this system component, necessary calculations were conducted for MSRE heat exchanger.

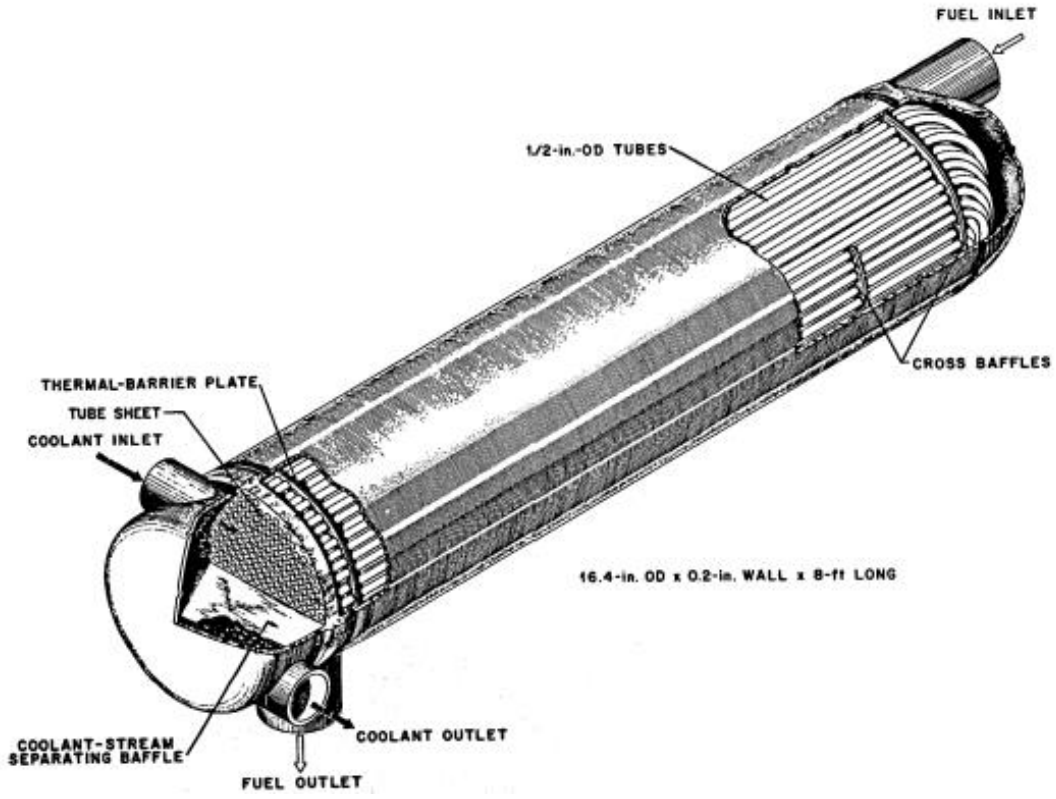


Figure 19: MSRE Primary Heat Exchanger "Reprinted from [70]".

Fuel salt leaves the fuel pump to flow through the shell side of heat exchanger shown in **Figure 19.** Temperature drop across heat exchanger was $\sim 28\ K$, while inlet and outlet temperatures are 926 K and 908 K, respectively. The methods used in the design of heat exchanger are those commonly followed in designing heat exchangers of shell-tube type. The tube side heat transfer coefficient was computed from the Sieder-Tate equation [71]. The fact that Sieder-Tate was used in heat exchanger design

completes the picture and concludes the work done in Chapter II, where the prediction of thermal hydraulic molten salt behavior falls within 10% of the expected value in Sieder-Tate and Dittus-Boelter correlations for all turbulence models.

The heat exchanger is $\sim 0.4\ m$ in diameter and $\sim 2.4\ m$ long, where it contained 163 U-tubes with $\sim 0.01\ m$ diameter, where the heat transfer area is $\sim 24\ m^2$. Inlet and outlet temperature of the cooling salt flowing through the tubes are 825 K and 866 K respectively. GeN-Foam model section covers the details related to dimensional calculations for each component.

Methodology

The concept of Multiphysics refers to simulations using software or computer code as a mean to coupling multiple models together to simulate a scenario with multiple interdependent physical scenarios. In this section, a Multiphysics open-source code is utilized to build and validate a 1D model that studies the transients of thermal MSRs as an approach to enhance the analysis of in-core interactions in the case of a tightly coupled physics due to the flowing salt. The following sections will introduce the adopted Multiphysics code, assumptions, equations, and details incorporated to build the model.

GeN-Foam

GeN-Foam is a Multiphysics solver that is based on OpenFoam, which consists of C++ libraries [72]. It couples together a multi-scale mesh sub-solvers for thermal mechanics, thermal hydraulics, neutronics, and a finite difference model. This work utilizes the thermal hydraulics and neutronics solvers only, where the latter is mainly a

multi group diffusion solver, however, this work employs a point kinetics solver. This section includes the governing equations of thermal hydraulics and neutronics solvers, and the coupling strategy used by GeN-Foam.

Thermal-Hydraulic Sub-solver

Starting from the turbulent single-phase flow equations for a fluid in a porous medium [49], Eqs. 29-31 present the conservation equations adopted in the thermal-hydraulics sub-solver.

$$\frac{\partial \gamma \rho}{\partial t} + \nabla(\gamma u_d) = 0 \quad \text{(Eq. 29)}$$

$$\frac{\partial \rho u_d}{\partial t} + \frac{1}{\gamma}\nabla(\rho u_d \times u_d) = \nabla\big(\mu_T \nabla(\mathrm{u})\big) - \gamma \nabla p + \gamma F_g + \gamma F_{ss} - (\rho u_d \times u_d)\nabla\frac{1}{\gamma} \quad \text{(Eq. 30)}$$

$$\frac{\gamma \rho e}{\partial t} + \nabla\big(u_d(\rho_e + p)\big) = \gamma \nabla(K_T \nabla T) + F_{ss} u_d + \gamma \dot{Q}_{ss} + (K_T \nabla T)\nabla\gamma \quad \text{(Eq. 31)}$$

The γ represents the porosity, where solver considers the volume occupied by fluid and assigns Darcy velocity $u_d = \gamma u$ for it. F_{ss} is the drag force exerted by the solid sub-scale structure on the fluid. Its proportionality to the Darcy velocity is presented as follows:

$$F_{ss} = k(u_d)u_d \quad \text{(Eq. 32)}$$

$\dot{Q}_{ss}$ represents the volumetric heat between the two media:

$$\dot{Q}_{ss} = A_v h(T_{ss} - T) \quad \text{(Eq. 33)}$$

$$\rho_{ss} C_{p,ss} \frac{\partial T_{ss}}{\partial T} = \nabla(\gamma k_{ss} \nabla T) + A_V h(T - T_{ss}) \quad \text{(Eq. 34)}$$

K_{ss} and T_{ss} are the conductivity and temperature of the solid sub-scale structure. In the heat exchanger, T is assumed to be constant for the sake of simplification.

$$\dot{Q}_{ss} = A_V h_{eff}(T_{ext} - T) \quad \text{(Eq. 35)}$$

Eqs 29, 30, 31 are solved by a PIMPLE pressure-based algorithm for compressible flow. GeN-Foam solves $k - \epsilon$ in fluid regions and porous $k - \epsilon$ in porous zones by forcing the values of k and ϵ to converge to the values $k_0$and ϵ_0, which can be defined by a selected convergence rat$\lambda_{\epsilon/k}$:

$$\rho \frac{\partial \epsilon}{\partial t} + \nabla(\rho u_d \epsilon) = \rho \lambda_{\frac{\epsilon}{k}}(\epsilon_0 - \epsilon) \quad \text{(Eq. 36)}$$

$$\rho \frac{\partial k}{\partial t} + \nabla(\rho u_d k) = \rho \lambda_{\frac{\epsilon}{k}}(k_0 - k) \quad \text{(Eq. 37)}$$

The following equation is used to calculate the fuel temperature profile, where axial heat conduction is neglected.

$$\rho_f C_{p,f} \frac{\partial T_f}{\partial t} = k_f \frac{\partial^2 T_f}{\partial r^2} + k_f \frac{1}{r} \frac{\partial T_f}{\partial r} + \dot{Q}_f \quad \text{(Eq. 38)}$$

Neutronics Sub-solver

The neutronics sub-solver uses point kinetics equations in order to study the kinetic behavior of the reactor. It is easier to solve point kinetics than to solve multi-group diffusion equation, since this solver neglects the spatial dependence of neutron flux and precursors. Accordingly, it predicts the time response of a nuclear system in the case of transients. The point kinetics equations for solid fuel are presented in Eq. 39.

$$\begin{cases} \dfrac{dP(t)}{dt} = \dfrac{\rho(t) - \beta_{eff}}{\Lambda} P(t) + \displaystyle\sum_{i=1}^{M} \lambda_i c_i(t) \\ \dfrac{dc_i(t)}{dt} = \dfrac{\beta_{i,eff}}{\Lambda} - \lambda_i c_i(t) \qquad i = 1,2,\dots,M \end{cases} \tag{Eq. 39}$$

Per the point kinetics equations, $\rho(t)$ represents the reactivity, Λ is the mean generation time of the system, $\beta_{i,eff}$ and β_{eff} are the effective delayed neutron fraction for the i-th delayed neutron precursor group and effective total delayed neutron fraction, respectively, λ_i is the decay constant of the i-th precursor group. P is the power of the system and c_i is the amount of delayed neutron of the i-th precursor group.

The derivation of point kinetics model becomes more complex since precursors do transport in the system, and the presence of diffusion term. Hence, precursors equation is modified in order to take into account the precursors motion. Note that τ_c and τ_e represent the effect of the fuel motion on the delayed neutron precursors.

$$\frac{dc_i(t)}{dt} = \frac{\beta_{i,eff}}{\Lambda} P(t) - \lambda_i c_i(t) - \frac{c_i(t)}{\tau_c} + \frac{c_i(t - \tau_e)}{\tau_c} e^{-\lambda_i \tau_e} \qquad i = 1,2,\dots,M \tag{Eq. 40}$$

The previously developed point kinetics models for liquid fuel analysis have derivation complications especially when weighing the multigroup diffusion equation and precursors equation by a weight factor such as the adjoint flux. Also, the models have limitations such as the neglection of precursors radial behavior. The model developed by Mattioli *et al.* [68] and used in this work presents a full spatial distribution of the precursors, which gives more accurate results in the case of 1D model. It also could be utilized in 2D and 3D models.

The derivation of the model started from a diffusion equation coupled with the transport diffusion equation of the precursors. After several steps of derivation, the final equations become as follows,

$$\begin{cases} \dfrac{dP(t)}{dt} = \dfrac{\rho(t) - \beta_{eff}}{\Lambda} P(t) + \sum_{i=1}^{M} \lambda_i c_i^*(t) \\ \dfrac{\partial \widetilde{C_\iota}(r,t)}{\partial t} + u.\nabla \widetilde{C_\iota} = -\lambda_i \widetilde{C_\iota} + \dfrac{\beta_i}{\Lambda} \phi_n(0) P(t) + \nabla . D_{i,f} \nabla \widetilde{C_\iota} \end{cases} \quad \text{(Eq. 41)}$$

$\widetilde{C_\iota}$ is the equation of precursors divided by the integral of the flux along the volume. Implementation of the model in GeN-Foam happens in two steps. First, by initializing the necessary quantities in equation 41. Second step includes solving time spatial equations of precursors with the transient power equation and iterate until convergence is reached. One group flux field is needed, because the model is mono-energetic.

Equilibrium reactivity, shown in **Figure 20**, accounts for the reactivity loss due to precursors drift in liquid fueled reactors. It can be calculated by solving the steady state/spatial version of equation 41, which results in equation 42.

$$\begin{cases} \dfrac{\rho_0 - \beta_{eff}}{\Lambda} P_0 + \sum_{i=1}^{M} \lambda_i c_{i,0}^* = 0 \\ u.\nabla \widetilde{C_\iota} = -\lambda_i \widetilde{C_{\iota,0}} + \dfrac{\beta_i}{\Lambda} \phi_n(0) P_0 + \nabla . D_{i,f} \nabla \widetilde{C_{\iota,0}} \end{cases} \quad \text{(Eq. 42)}$$

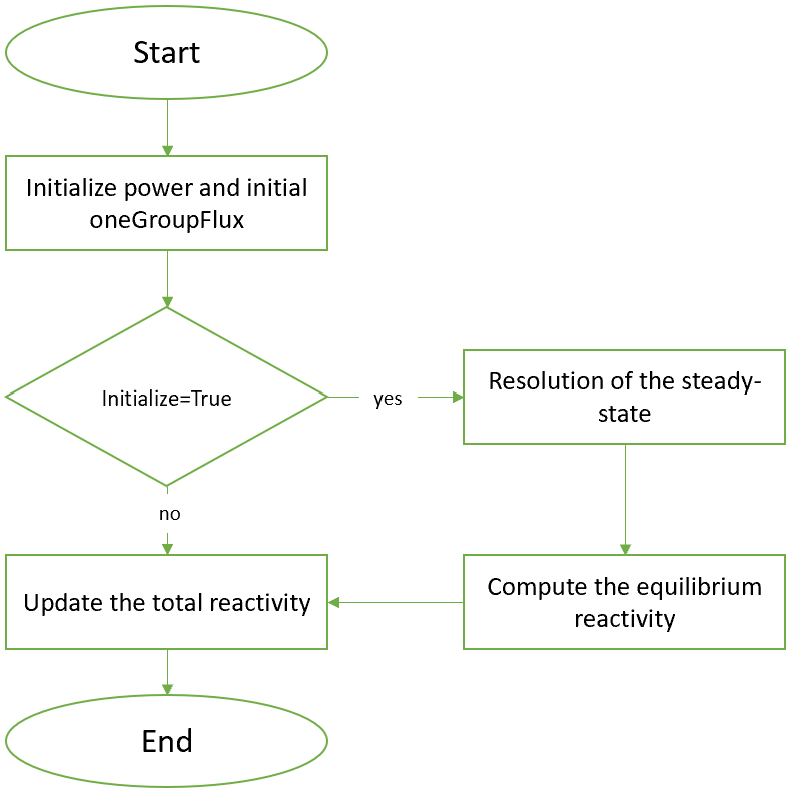


Figure 20: Flow Chart of the first iteration of the point kinetics solver "Reprinted from [68]".

The code solves precursors equation first, because the initial normalized flux is calculated before in addition to all other data such as the initial power and velocity fields of the system. Initial normalized flux was chosen to be the weight function in order to use reasonable function. After that, the weighted precursor of the ith group ($c_{i,0}^{*}$) is calculated as in Eq. 43.

$$c_{i,0}^{*} = \frac{\int W \widetilde{C}_{\iota}(r,t)dV}{\int W \phi_{n}(0)dV} \qquad \text{(Eq. 43)}$$

This step paves the way to calculate equilibrium reactivity as seen in Eq. 16.

$$\rho_0 = \beta_{eff} - \frac{\Lambda \sum_{i=1}^{M} \lambda_i c_{i,0}^*}{P_0} = 0 \qquad \text{(Eq. 44)}$$

After that, the total reactivity is calculated, where the external reactivity and the reactivity due to feedbacks are equal to zero in steady state condition. After solving Eq. 42 and finding the equilibrium reactivity, the full set of equations (Eq. 41) can be solved.

$$\rho(t) = \rho_0 + \rho_{external} + \rho_{feedback} \qquad \text{(Eq. 45)}$$

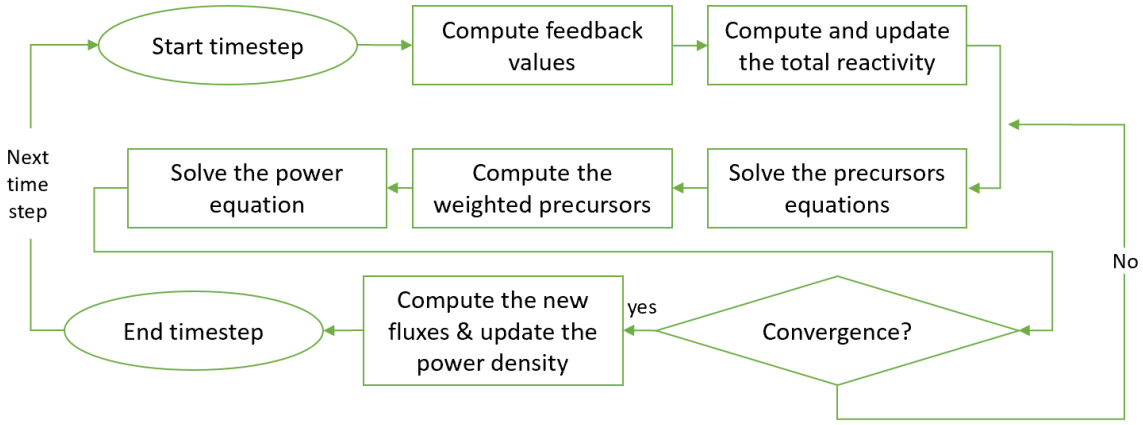


Figure 21: Flow Chart of the time dependent point kinetics solver "Reprinted from [68]".

Figure 21 shows the time dependent slow chart of the point kinetics solver, which shows that it is necessary to calculate the feedback values. These feedbacks are calculated after finding the average values of the main parameters such as density and temperature. Eq. 46 shows the calculation of average temperature by normalizing over the squared adjoint flux. Similarly, the average of other parameters is calculated.

$$T_{fuel,Avg} = \frac{\int \phi^2(r,t) T_{fuel}(r,t) dV}{\int \phi^2(r,t) dV} \quad \text{(Eq. 46)}$$

After finding the average parameters, feedback reactivity was calculated, assuming that the reactivity coefficients are known, as follows

$$\rho_{feedback} = \rho_{fuel}(t) + \rho_{density}(t) \quad \text{(Eq. 47)}$$

$$\rho_{feedback} = \alpha_{fuel}\left(T_{fuel,avg}(t) - T_{fuel,avg}(0)\right) + \alpha_{density}(\rho_{fluid,avg}(t) - \rho_{fluid,avg}(0)) \quad \text{(Eq. 48)}$$

As can be seen in **Figure 21**, after the first timestep calculation, the code calculates feedback values, then it calculates the total reactivity. Afterwards it finds the weighting precursors, subsequently it solves the power equation, and it keeps iterating until it reaches convergence, which allows it to calculate the neutron flux and the power density. Power density is updated through the following scaling factor:

$$S.F. = \frac{P(t)}{P(t - \Delta t)} \quad \text{(Eq. 49)}$$

here Δt is the time step used for the numerical calculation. The previous procedure of solving the point kinetics comes with the advantage of monitoring the time evolution of the system in case of a transient. These parameters are power, total reactivity, and the average temperature.

Coupling Strategy

The source code of GeN-Foam is divided into three sub-solvers, neutronics (neutronics), thermal hydraulics (thermalHydraulics), thermal mechanics (thermoMechanics), and Multiphysics control (multiphysicsControl) to couple the three solvers together. Each case folder has three folders: “fluidRegion”, “neutroRegion”, and “thermomechanicalRegion” and each region folder has multiple subfolders as can be seen in Figure 22. The illustration of the coupling strategy includes thermal-mechanics sub-solver to better demonstrate the interaction between the three distinct physics in GeN-Foam. However, it's important to note that the use of this sub-solver is not within the scope of this work, and it is not needed for the purpose of this research. This approach is used only for the illustration of the interaction between physics, but the results and conclusions of the work will not include the utilization of this sub-solver.

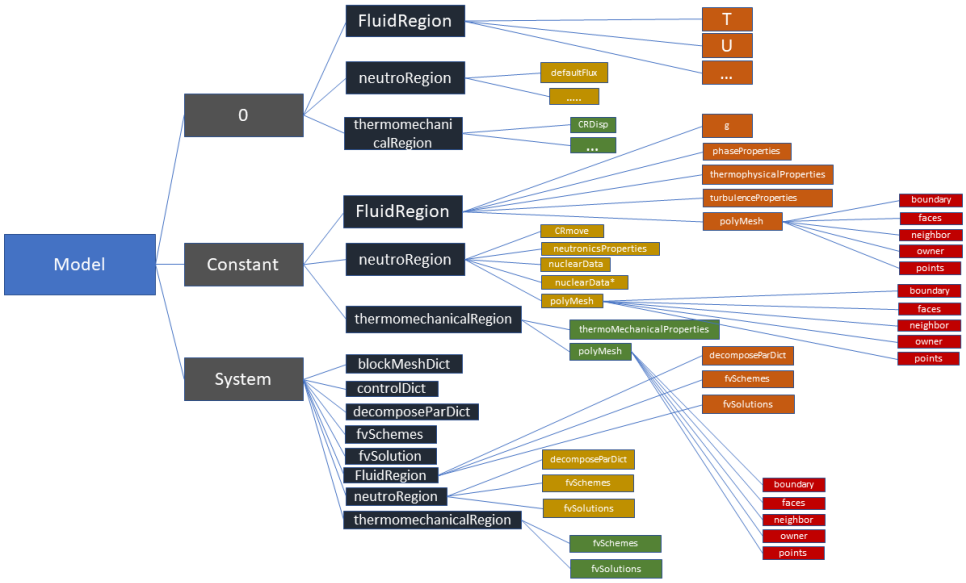


Figure 22: a schematic showing folders and files for an arbitrary simulation in GeN-Foam.

There are three different meshes for the three different physics in GeN-Foam, where each mesh could have a certain refinement as needed. This is a great feature in the code that saves a lot of computational resources by conducting mesh-to-mesh projection of coupling fields. This is accomplished due to the existence of OpenFOAM mesh-to-mesh projection class.

In order to couple between the three different solvers, there are many variables to navigate between the physics as can be seen in **Figure 23**. For instance, starting from thermal mechanics, which needs the temperature of the structures (T_s) that comes from the thermal hydraulics (notice the red arrow), and it forms the only variable needed for thermal mechanics solver. Thermal mechanics has a domain (Ω_{TM}), where physics get solved, then it produces displacement (d), which may be used for parametrizing cross sections, and modified domains if needed for neutronics (Ω_N) and thermal hydraulics (Ω_{TH}).

Neutronics sub-solver needs multiple parameters such as the displacement, modified domain. It also needs temperature, density, and velocities of the coolant, and temperature of the structures. Based on the densities and temperatures, feedback coefficients will be calculated for either point kinetics or spatial kinetics. Neutronics sub-solver solves for fluxes, which produces the power density (q), while thermal hydraulics sub-solver requires power density and the modified domain if available to produce most of the inputs needed by neutronics.

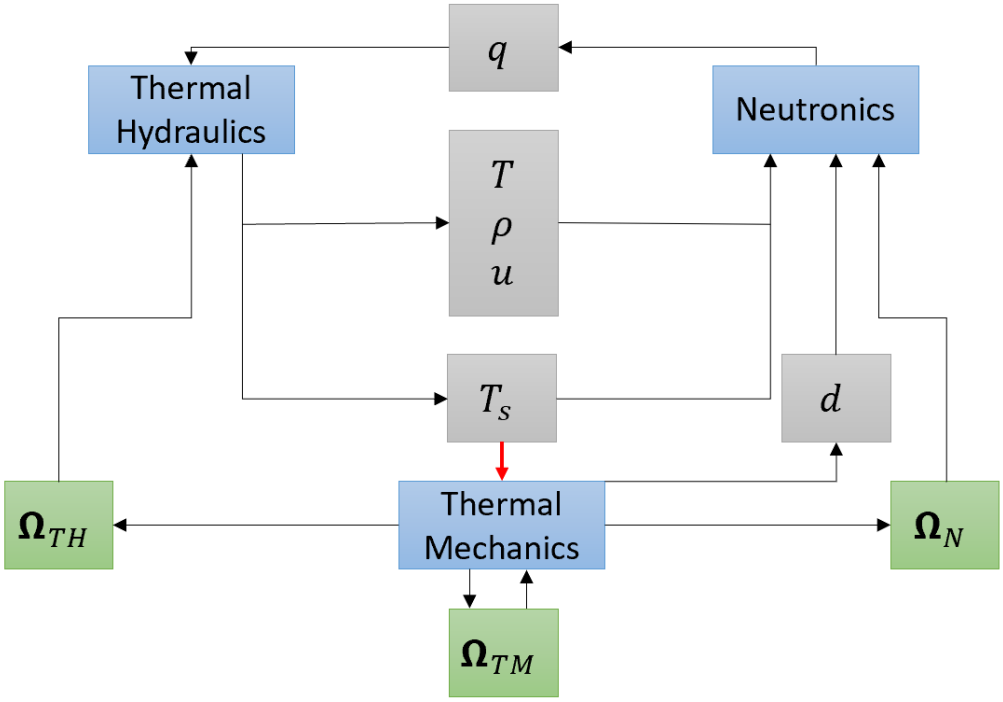


Figure 23: Flow chart showing the coupling strategy in GeN-Foam.

GeN-Foam uses fixed-point iteration for coupling because it is the main approach available in Open-Foam. This approach is very simple since it iterates between the three sub-solvers sequentially and tends to reduce the residuals. It is accurate due to the unlimited number of iterations, which produces high accuracy as in fully coupled solution, and it tends to be stable.

In the case of thermal hydraulics sub-solver, GeN-Foam explicitly uses semi-implicit coupling. OpenFoam mainly carry out thermal hydraulics, which solves Navier-Stokes equations, and what it does to solve pressure-velocity coupling is a semi-implicit loop such as SIMPLE or PIMPLE.

Assumptions

The purpose of building this model is to simplify the investigation of transients for thermal MSRs by running one simulation through a code that couples neutronics and thermal hydraulics in a simple matter as it iterates between the two sub-solvers sequentially aiming to reduce residuals of each of the sub-solvers.

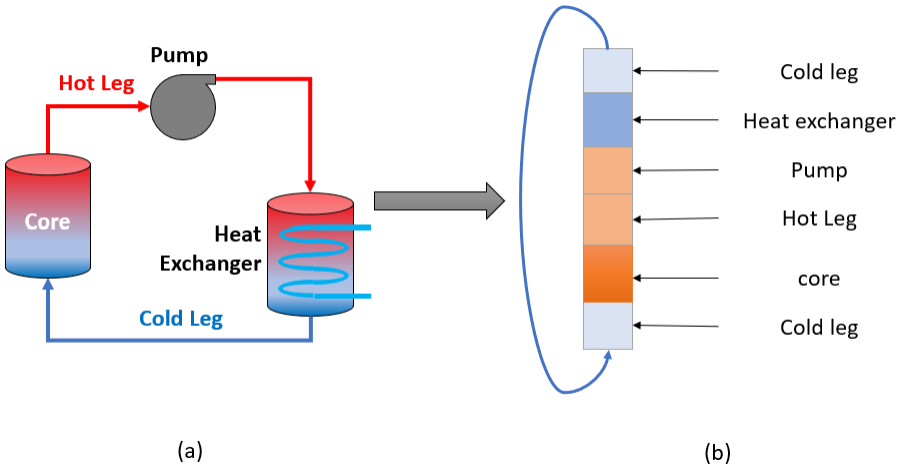


Figure 24: (a) Conceptual scheme of MSR, (b) 1D GeN-Foam model of the primary loop.

The model aims to rescale the primary loop for thermal MSR motivated by the work done by Mattioli [49]. The primary loop of the model was simplified into a 1D vertical section, where each component is defined in a single section. This simplification allows for easier analysis and understanding of the flow through the main components. The flow was allowed through the main components using a topology set dictionary, which eliminates the presence of edges between the sections (components) and ensures that the flow is directed through the intended path. The computational domain consisted of five components as can be seen in **Figure 24 (b).**

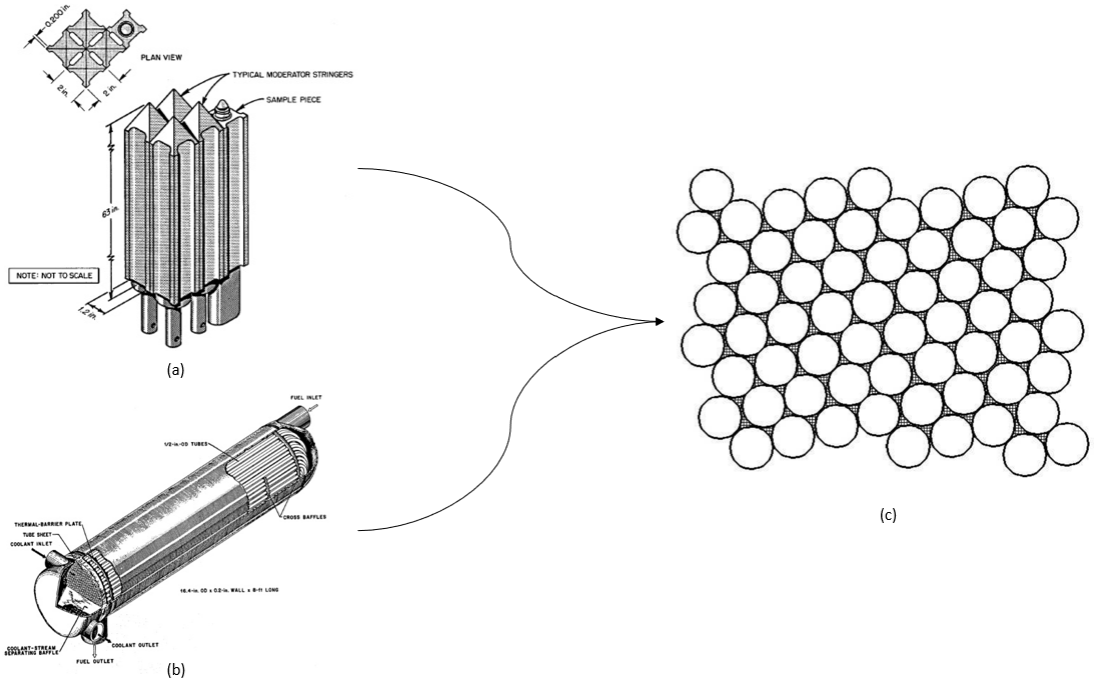


Figure 25:(a) Graphite moderated core "Reprinted from [73]", (b) Heat exchanger "Reprinted from [73]", (c) Assumed Porous medium.

The main assumption tied to the model is porosity in the core to account for the presence of Graphite, where Graphite forms 77.5% of the core. Porosity was assumed in the heat exchanger as well to account for the existence of cooling salt and piping. This assumption required the use of porous $k-\epsilon$ model for thermal hydraulics sub-solver as illustrated in Eqs. 29, 30, and 31.

GEN-Foam Model

This section describes the calculations and procedures employed, utilizing the data and assumptions outlined in previous sections pertaining to the MSRE and the construction of the model, where it delves into the characterization of the dimensions of

the system components, their volumetric fractions, volumetric area and the implementation of physic for both sub-solvers.

The initial step in the analysis was to determine the volume of each component within the system. Fratoni *et al*. [1] quantified these volumes and they are presented in **Table 9** (a modified version of the one in [66]) for the reactor components, which were limited to the five primary components. It is important to mention that the volumes were computed at a temperature of 922K.

Table 9: Salt volume in the components of MSRE primary loop

Component	Volume [m^3]
Core	0.708
Pump	0.62
Hot leg	0.356
Heat Exchanger	0.173
Cold leg	0.139

The model was built based on the volumes provided in **Table 9** with a $0.01m$ diameter and a height of $199.6\ m$, the value of the diameter was chosen to reduce the effect of aspect ratio as a result of refining the mesh. This is a common practice in scientific simulations, as controlling the aspect ratio can improve the accuracy of the results by optimizing the mesh refinement. By choosing a specific diameter and height, the aspect ratio is controlled, and the simulation results can be more accurate and reliable, which can help to improve the understanding of the problem and lead to mesh

independent solutions. **Table 10** shows the updated volume for the 1D model, which becomes height after cutting dimensions to 1D.

Table 10: Heights of primary loop components for the 1D model.

Component	height [m]
Core	70.8
Pump	13.9
Hot leg	35.6
Heat Exchanger	17.3
Cold leg	62

In the model, thermal properties such as thermal conductivity, specific heat capacity, density, and viscosity are used to simulate the heat transfer and fluid flow in the system. These properties are material-specific and are required to accurately represent the thermal and fluid behavior of the system. The thermal conductivity represents the ability of a material to transfer heat, the specific heat capacity represents the amount of heat required to raise the temperature of a unit mass of a material by a certain degree, the density represents the mass per unit volume of the material, and viscosity represents the internal resistance of a fluid to flow. These properties are essential for the simulation of heat transfer and fluid flow and are used in conjunction with the governing equations of heat transfer and fluid dynamics to solve for the temperature distribution and fluid flow in the system.

Table 11 shows the thermal properties of the materials used in the simulation, as well as the values found in the literature for reference. This table provides a

comprehensive overview of the thermal properties that are used in the simulation, where it allows to compare the values used in the simulation with the values found in the literature. By having this information in a table format, it makes it easy to understand the values that were used in the simulation and how they are compared to the literature values.

Table 11: Thermal properties of MSRE fuel salt.

Property	**Design values (1964) [59]**	**ORNL-TM-3039 (1973) [74]**	**1D Model [75]**	**Unit**
Thermal conductivity	4.76	1.44	1	[W/m.K]
Viscosity	0.00739	0.00773	7×10^{-6}	[pa.s]
Density	2471.7	2261.8	2245.2	[kg/m^3]
Specific heat	1927	1984	2386.5	[J/kg.K]

Thermal properties of the molten salt used were determined by utilizing the information presented in a report by Thoma [74], which covers the chemical aspects of MSRE operations. The report is considered to be a reliable source for the determination of these thermal properties. Measurements were performed at an operating temperature of $916\ K$ (approximately $643\ °C$), which is representative of the average temperature in the core of MSRE.

Thermal properties of graphite, a key material in the construction of MSRE core, are crucial for the safe and efficient operation of the reactor. These properties can be found in **Table 12**.

Table 12: Thermal properties of Graphite "Reprinted from [5]".

Property	Unit	Value
Density	[kg/m^3]	1874
Specific heat	[J/kg.K]	1772
Thermal conductivity	[W/m.K]	53

The porous $k-\epsilon$ turbulence model is an extension of the standard $k-\epsilon$ turbulence model, which is widely used in computational fluid dynamics simulations. This model is specifically designed to consider the effect of porous media on the turbulent flow. It solves for the turbulent kinetic energy k and its dissipation rate ϵ as additional transport equations, which are coupled with the conservation equations of mass, momentum, and energy. The turbulence intensity, which is the ratio of the turbulent velocity fluctuations to the mean flow velocity, and the turbulence length scale, which is a measure of the size of the turbulent eddies, are used as input parameters to the model to help define the turbulence in the flow within porous media. **Table 13** shows the values of parameters used in the simulation. The values remained unchanged and were kept at their default settings set by the code developer.

Table 13: Turbulence properties in GeN-Foam

Parameter	Value
Convergence length	0.5
Turbulence intensity coefficient	0.16
Turbulence intensity exponential	-0.125
Turbulence l coefficient	0.07

The hydraulic diameters, volume fractions, and volumetric areas of each component were calculated using the data obtained from the (components) section and the literature, in accordance with established methodologies. The Blasius equation is also used to approximate the velocity profile of the fluid in the boundary layer, and it is used to calculate the friction factor, which is an important parameter in determining pressure drop of the flow.

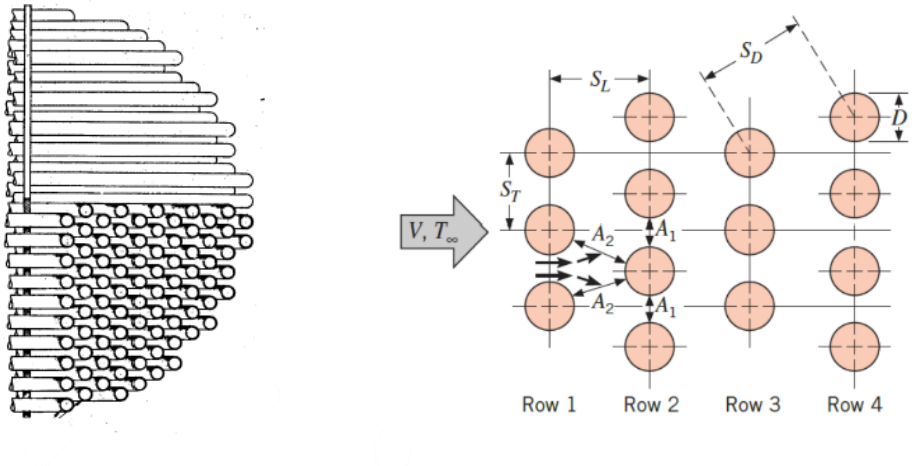
The hydraulic diameter of the shell side in a shell-and-tube heat exchanger is a measure of the effective size of the flow passages on the shell side, and it is used to calculate the fluid flow rate and heat transfer coefficient. It is defined as four times the cross-sectional flow area of the shell side divided by the wetted perimeter of the shell side. For a shell-and-tube heat exchanger, the hydraulic diameter of the shell side is calculated as:

$$D_h = 4 \cdot \frac{A}{P_w} \quad \text{(Eq. 50)}$$

Where A is the flow area, and P_w is the wetted perimeter. In **Figure 26** (a), the heat exchanger tubes of the MSRE are depicted, and **Figure 26** (b) illustrates the lateral and transverse spacing between the tubes to assist in the calculation of the hydraulic diameter.

$$D_h = 4 \cdot \frac{\left(\frac{1}{2} S_T \cdot \frac{\sqrt{3}}{2} \cdot S_L - \frac{\pi D^2}{8}\right)}{\frac{1}{2}\pi D + 2A_2 + A_1} \quad \text{(Eq. 51)}$$

The cross-sectional area A is determined by subtracting the partial area of the tubes in contact with the flow from the total area of the triangle, as described in the numerator of Eq. 51. P_w, on the other hand, represents the length of each side of the flow section and can be found in the denominator of Eq. 51.



(a) (b)

Figure 26: (a) MSRE heat exchanger tube bundle "Reprinted from[76]" and (b) generalized Schematic of Geometric Parameters for Staggered-Pitched Tube Bundles "Reprinted from [77]".

The hydraulic diameter of the shell side is an important parameter in the design and performance analysis of a shell-and-tube heat exchanger, as it is used in the calculation of the heat transfer coefficient, pressure drop and fluid flow rate on the shell side. Same calculations were done for the core taking into consideration the difference in dimensions and geometry. Dittus-Boelter correlation is utilized in the modeling of the core and heat exchanger due to its demonstrated accuracy in approximating heat transfer phenomena. The literature was used to determine the proportion of fuel within the core

and heat exchanger, as well as the volumetric area, which represents the ratio of heat transfer surface area to volume.

The point kinetic equation adopted in this work is a commonly employed mathematical model in the study of nuclear reactors. It describes the behavior of the reactor's neutron population, including the rates of neutron production and loss. Two key parameters in the point kinetic equation are lambda (λ) and beta (β). Lambda, also referred to as the delayed neutron precursor decay constant, which quantifies the rate of delayed neutron production. Beta, also known as the effective delayed neutron fraction, which represents the proportion of total neutron production that originates from delayed neutrons.

To provide a clear understanding of these parameters in the context of MSRE, **Table 14** presents lambda and beta values for U233 fuel used in MSRE. The values of lambda and beta in this table are specific to MSRE and are reported in units of reciprocal seconds. The data from **Table 14** was employed along with the files depicted in **Figure 22** to construct the model within the GeN-Foam framework.

Table 14: Kinetic parameters used in the model "Reprinted from [78]".

Beta [pcm]	Lambda [1/s]
22.8	0.0126
78.8	0.0337
66.4	0.1394
73.6	0.325
13.6	1.13
8.8	2.5

GeN-Foam framework was utilized to construct a one-dimensional (1D) model of MSRE. The data presented in this chapter was used to build the model. The process involved conducting multiple calculations and employing boundary conditions to accurately capture the behavior of the reactor's neutron population. The values obtained from the model were then compared to those found in literature, considering rescaling the model from 3D to 1D.

Boundary conditions for the 1D model were divided into two main categories: thermal hydraulic and neutronics. The neutronics boundary conditions were specified in the “defaultFlux” file, while multiple files were used to define the thermal hydraulic boundary conditions.

For all cases, the bottom and top of the domain were set as cyclic boundary conditions. Additionally, wall functions were defined for each turbulent property, and the velocity boundary condition was set to “slip” due to the one-dimensional nature of the model. On the other hand, the temperature boundary condition on the walls was set to zero gradient.

Table 15 presents a summary of all thermal hydraulic boundary conditions used in the model. Feedback coefficients used in the model for fuel salt and moderator are -11.034 pcm/K and -5.814 pcm/K, respectively [78]. Power density of the model was determined by using the power measurements from the MSRE The primary loop operation parameters are listed in **Table 16.**

Table 15: Thermal hydraulic boundary conditions.

	Top	Bottom	Wall
U	Cyclic	Cyclic	slip
T	Cyclic	Cyclic	zeroGradient
P_rgh	Cyclic	Cyclic	fixedFluxPressure
k	Cyclic	Cyclic	kqRWallFunction
nut	Cyclic	Cyclic	nutkWallFunction
epsilon	Cyclic	Cyclic	epsilonWallFunction
alphat	Cyclic	Cyclic	zeroGradient
P	Cyclic	Cyclic	Calculated

Table 16: MSRE operation parameters used in the model "Reprinted from [59]".

Parameter	Initial condition
Rated Power [MW]	8
Flow rate [m^3/s]	0.0757
Inlet temperature [K]	905.4
Outlet temperature [K]	927.6

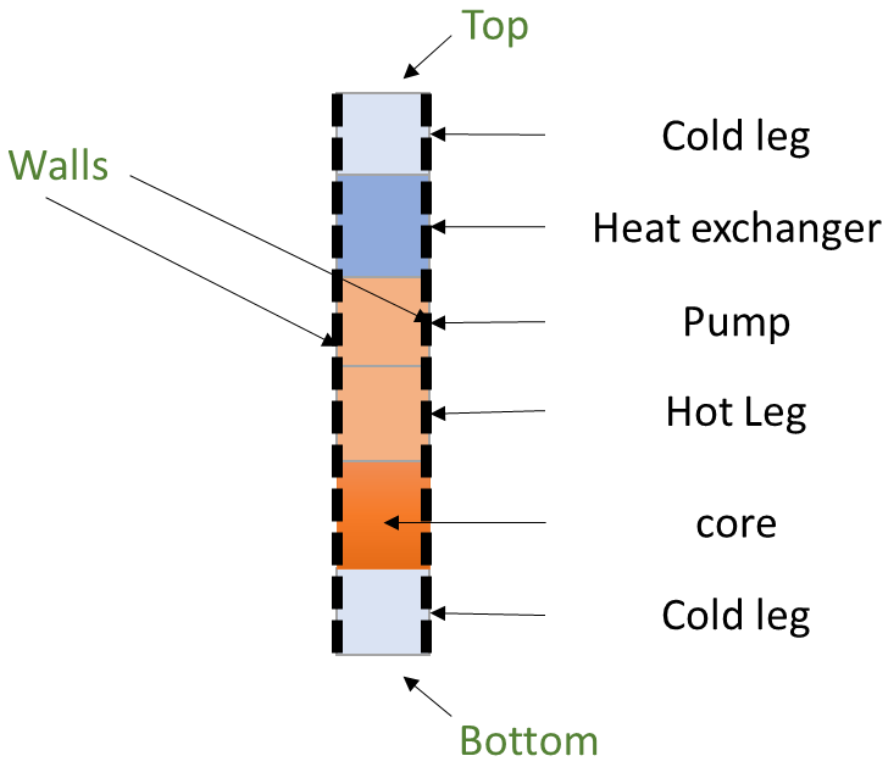


Figure 27: GeN-Foam 1D model.

Figure 27 depicts the 1D model developed for MSRE primary loop, which has been rescaled and simplified to better understand the behavior of the system. The model is developed using GeN-Foam software [68] and includes the main five components of the primary loop: the core, hot leg, heat exchanger, pump, and cold leg. Boundary conditions for the model are listed in **Table 15** based on the developed model as in **Figure 27**. These boundary conditions specify the conditions at the edges of the system, such as the temperature and velocity on the walls and flow rate at the inlet and outlet. By incorporating these conditions into the model, the behavior of the primary loop can be

correctly simulated and analyzed. The simulation's input files are provided in Appendix A, where all data utilized for the simulation is included.

Results

The proposed 1D model undergoes validation against literature values from MSRE, where the validation process is separated into two primary components, thermal hydraulics and Multiphysics. The model's input files are populated with accurate data in order to increase the performance of the model. The results obtained by the model are then compared to literature values from MSRE, exhibiting a good level of agreement. These results demonstrate the proposed model's capability to predict the performance of thermal MSR with adequate accuracy. It's worth to mention that implementation of the 1D model using GeN-Foam required a significant amount of modeling effort and time, however the results obtained from the model provide a good match to the literature values. The simulation progresses through three stages. In the first stage, only the neutronics sub-solver is activated and the thermal hydraulics sub-solver is not. In the next stage, only the thermal hydraulics sub-solver is activated. Finally, in the last stage, both sub-solvers are activated, and the neutronics sub-solver switches from solving diffusion to solving point kinetics.

Validation

The initial step in conducting simulations involved utilizing the neutronics sub-solver to generate a power density profile by applying the diffusion equation. This is crucial as failure to do so may result in the power density being represented as a step

function, leading to potential inaccuracies as precursor behavior is dependent on spatial location. It is important to note that while the utilization of the diffusion equation to produce the power density shape is not directly related to the primary simulation, it is a crucial step in creating a cosine shaped power profile in the core (see **Figure 28**).

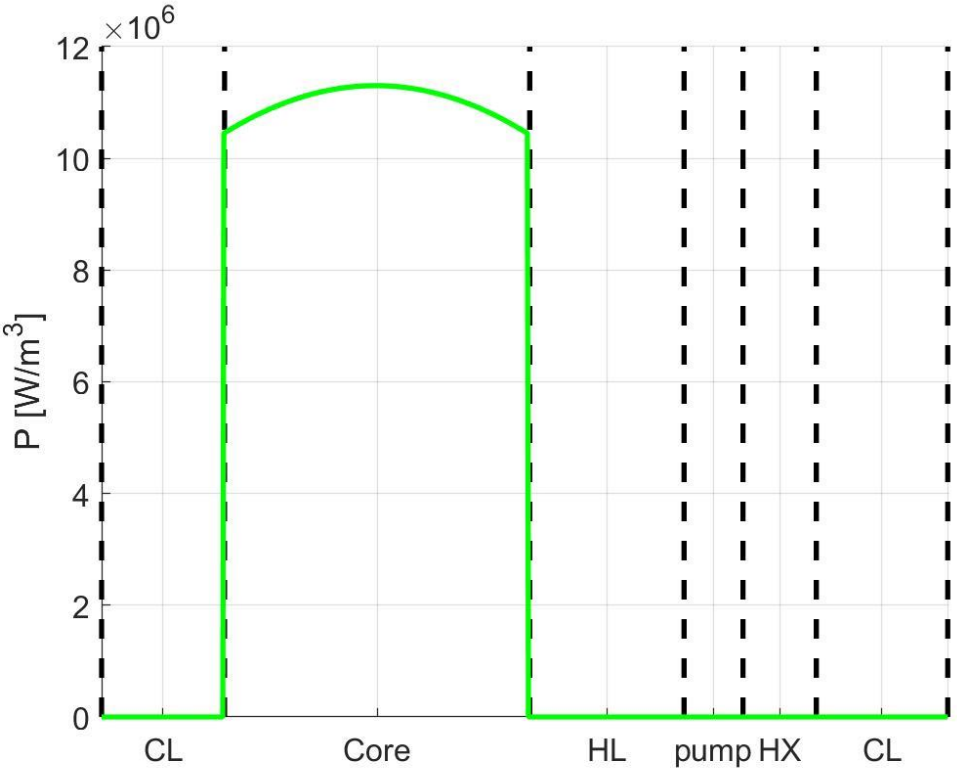


Figure 28: Power density in 1D primary loop of MSRE.

The neutronics sub-solver was specifically employed at this stage to accurately generate the power density profile, which is necessary to ensure the overall accuracy of the simulation as precursor behavior is dependent on spatial location. To calculate power density, the rated power (**Table 16**) is divided by the core volume (**Table 9**) resulting in $11.299 * 10^6\ W/m^3$ and then simulated using the diffusion equation for the first 100

iterations, where the simulation depicted in **Figure 28** confirms that the core is the sole source of power. This process provides the necessary power density required for the simulation.

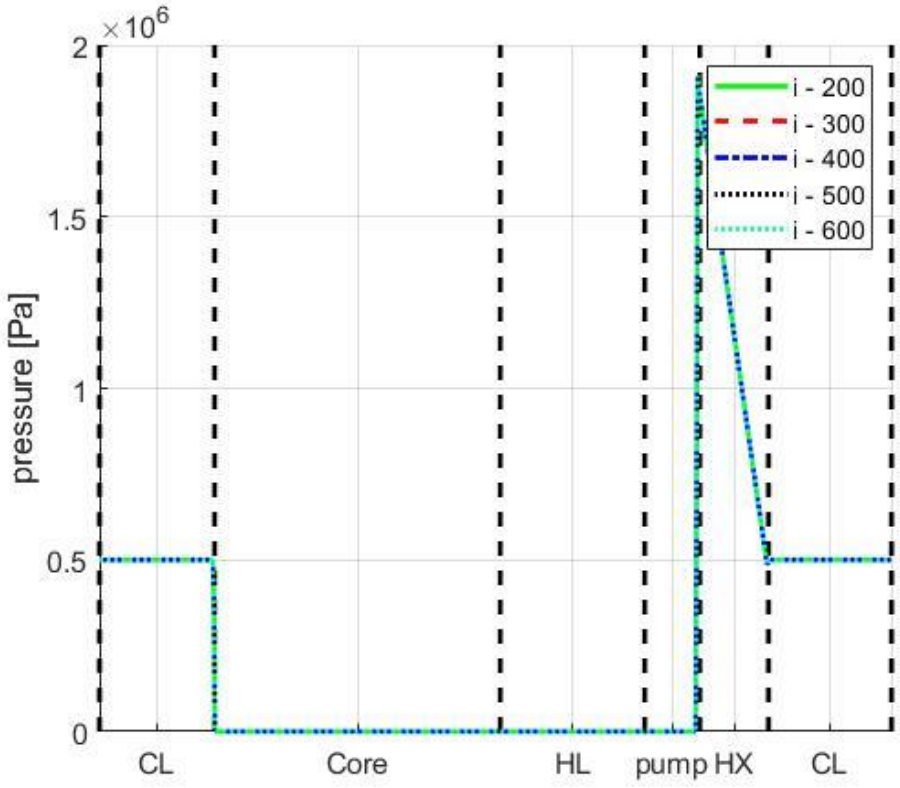


Figure 29: Pressure drop for 1D primary loop in MSRE.

After this, a steady state simulation was performed to validate the thermal hydraulics sub-solver by examining key metrics such as velocity, pressure, and temperature, which are commonly used in thermal hydraulic simulations.

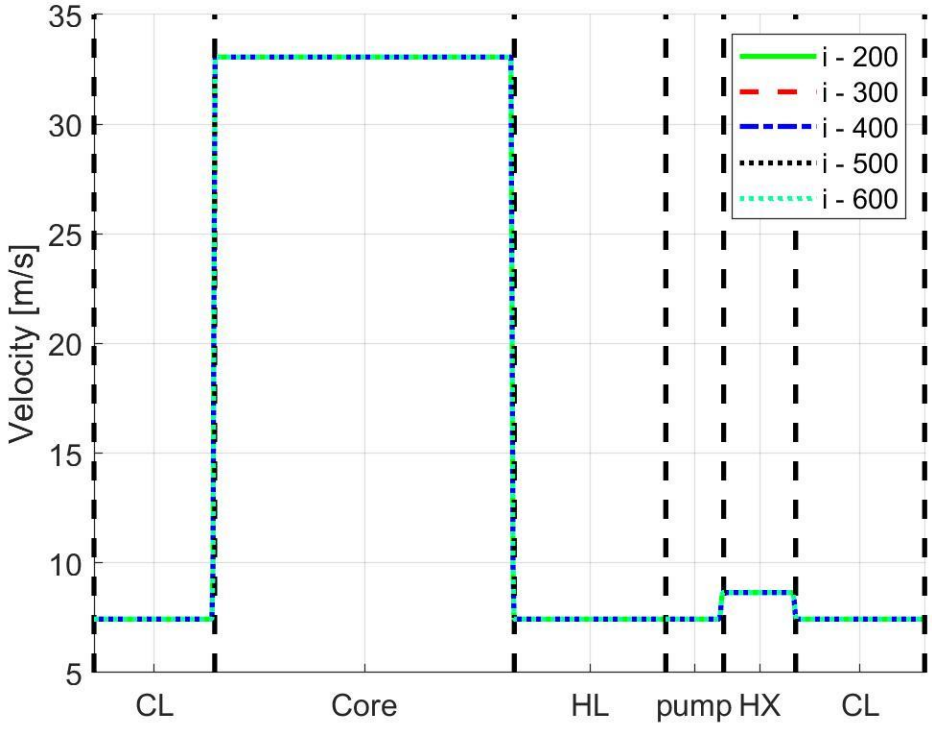


Figure 30: Velocity in 1D primary loop of MSRE.

pressure drop is an important concept in fluid dynamics because it affects the flow rate of a fluid, and as a result, the overall performance of any system that involves fluid flow. The pressure drop in porous medium, such as the core and heat exchanger in **Figure 29**, can be explained by the resistance offered by the porous medium to the fluid flow. In porous medium, the pressure drop is affected by factors such as the porosity, permeability and the fluid velocity. These factors all contribute to the resistance offered by the porous medium to the fluid flow. The pressure at the inlet of the core decreases as a result of the porous medium present, then the pressure increases due to the operation of the pump. After that, it decreases again as it enters the heat exchanger, caused by the

presence of porosity. The decrease in pressure in the heat exchanger is less than that in the core because it has a lower proportion of porous medium.

Velocity of the flowing salt is calculated by taking the volumetric flow rate and dividing it by the cross-sectional area of the domain, as per the value given in **Table 16** of the literature. Here, the area of the domain is the width of the domain. In **Figure 30**, it can be observed that the velocity of the fluid in all reactor components is constant at approximately $7.43\ m/s$, except for the core and heat exchanger. In these two components, the presence of a porous medium leads to a phenomenon known as Darcy velocity. As the porosity of the medium increases, the velocity of the fluid also increases. This can be seen in the case of the core, where a porosity of 0.775 results in a velocity of $33\ m/s$, and in the case of the heat exchanger, where a porosity of 0.14 results in a velocity of $8.7\ m/s$. Darcy velocity is defined as follows:

$$U_d = \frac{U}{\gamma} \qquad \text{(Eq. 52)}$$

It is important to use the correct velocity value when analyzing fluid flow in porous media, as this can have a significant impact on the flow rate and, subsequently, the power rate. This relationship is illustrated in the following Eqs. that relate velocity, flow rate, and power rate.

$$P = \dot{m} C_p \Delta T \qquad \text{(Eq. 53)}$$

$$\dot{m} = \rho U A \qquad \text{(Eq. 54)}$$

The last piece of the puzzle that is required to wrap up Eq. 53 is the temperature difference. Temperature difference, or ΔT, between the inlet and outlet of the core is a key factor in determining the amount of heat that can be transferred and subsequently the power rate. **Figure 31** represents the convergence of the temperature distribution in the primary loop of the MSRE as a function of iteration number. The velocity in the primary loop (**Figure 30**) is held constant as an input parameter, while the temperatures in each component are solved for iteratively, until the temperature distribution reaches a steady state. As the iteration number increases, the temperature values in the loop converge towards the steady state temperature distribution.

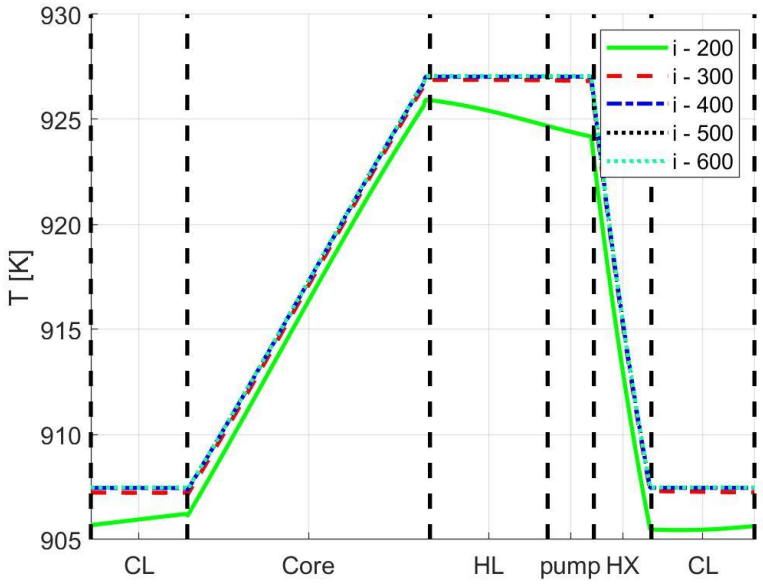


Figure 31: Temperature in 1D primary loop of MSRE at multiple iterations.

Figure 31 illustrates the temperature distribution in the primary loop of the model, as the molten salt coolant circulates through the loop. The figure shows how the

temperature increases as the salt enters the core, reaching a peak temperature before decreasing as it passes through the heat exchanger and reaches the cold leg, where the lowest temperature in the primary loop is observed. The inlet temperature is $907.48K$ and the outlet temperature is $927.06K$.

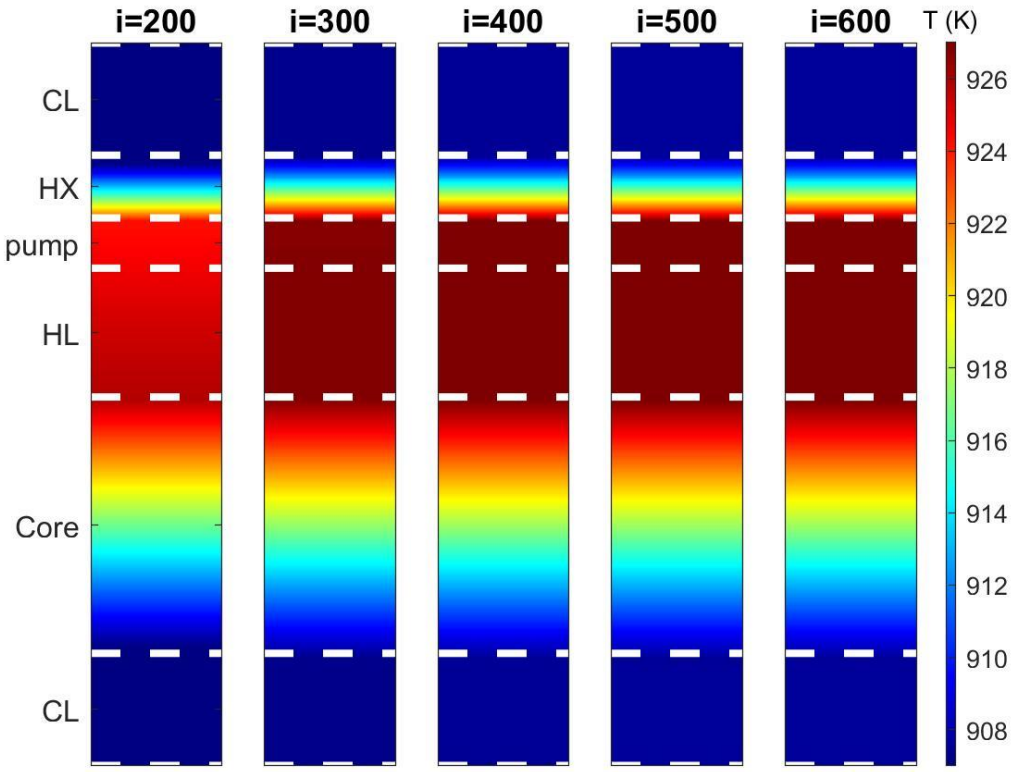


Figure 32 :Temperature contour in 1D primary loop of MSRE.

The temperature contour is another way to represent thermal behavior of the molten salt coolant as it circulates through the primary loop, and in order to understand the heat transfer dynamics of the system. Another representation of the temperature distribution can be found in **Figure 32.**

After all, the step of validating the thermal hydraulic sub-solver has been considered accomplished as the results of the simulation using GeN-Foam model matched perfectly with the experimental data obtained from MSRE. As illustrated before, calculations were performed using a set of defined inputs and the model was constructed accordingly.

Figure 33 illustrates the comparison of core outlet and inlet temperatures between the results obtained from GeN-Foam 1D model and the experimental data from MSRE. The discrepancy between the two data sets is observed to be less than 0.06% for the core outlet temperature and less than 0.23% at the inlet. The model and the experiment have small discrepancies around the heat exchanger due to the effect of the ambient temperature in the experiment that are not considered in the model.

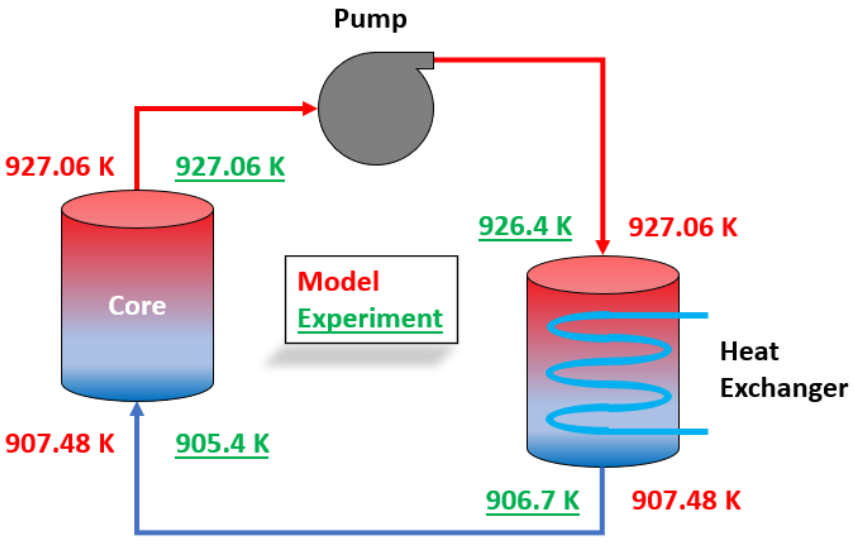


Figure 33: Comparison of inlet/outlet temperatures between GeN-Foam 1D model and MSRE data for core and heat exchanger.

The results, as depicted in **Figure 33**, demonstrate a high degree of accuracy between the outcomes of MSRE experiment and GeN-Foam 1D model, which further strengthens the validity and reliability of the simulation process utilized. This high degree of agreement between the model and experimental results instills confidence in the validity of the 1D model and provides justification for proceeding to the next step of validating the neutronics sub-solver. This validation process involves the application of point kinetics equation to analyze a transient at 8MW power by inserting $13pcm$ reactivity and comparing the results of the simulation to the experimental data as depicted in **Figure 34**.

A reactivity insertion transient is a change in the reactivity of a nuclear reactor, caused by the insertion or withdrawal of control rods or other types of reactivity-moderating devices. This change in reactivity can cause the power level of the reactor to change quickly, leading to a transient event. **Figure 34** shows a reactivity insertion transient simulated using 1D GeN-Foam model, and in order to be consistent with the experiment, the simulation is validated against MSRE test for U233 fuel to model the behavior of the reactor during the transient.

The simulation results were then compared to experimental data from the reactor as can be seen in **Figure 34** (a). The comparison showed a good match between the experiment and the simulation. This indicates that the model exhibits good correspondence with the experimental data in capturing the dynamics of the reactor during the transient. The maximum discrepancy between the model and the experiment

during the first 55 seconds of the transient was 1.5% at 10 seconds. However, after 55 seconds, the discrepancy rose to 4% at 85 seconds.

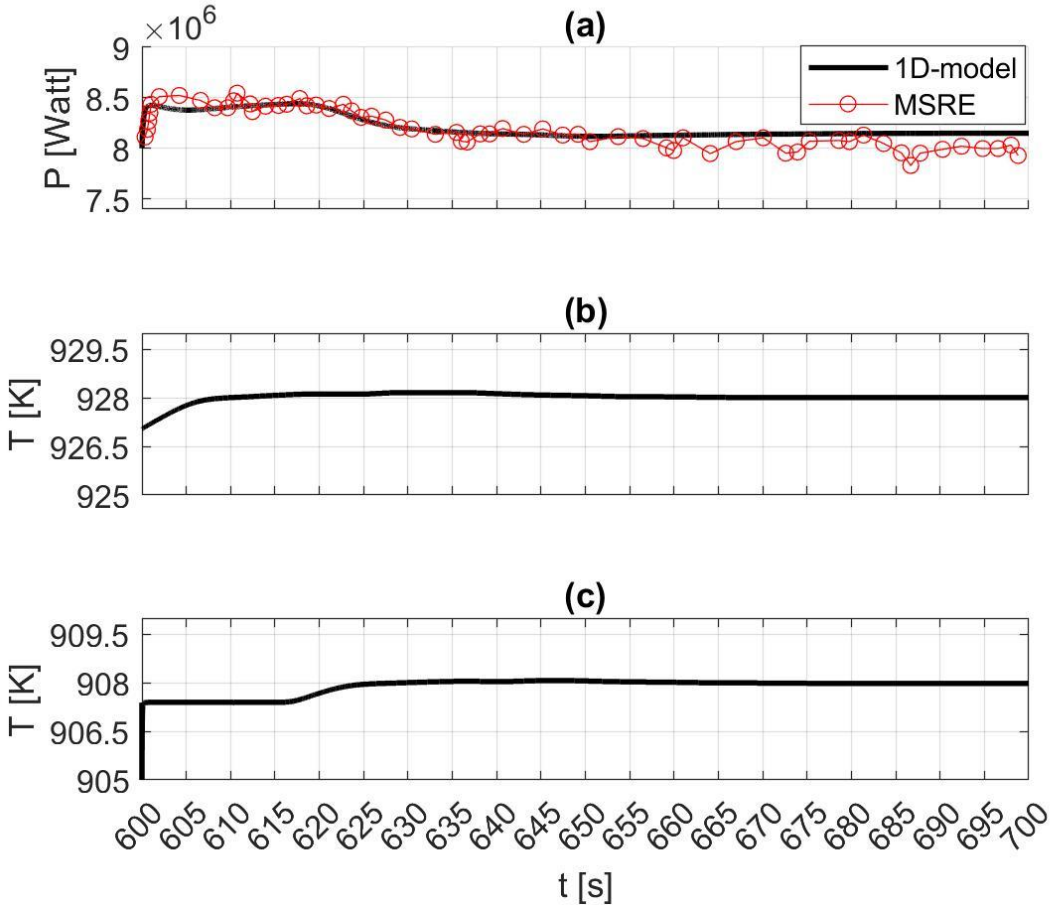


Figure 34: Prediction of GeN-Foam 1D model performance for the reactivity insertion event at 8MW "Adopted from [78]", (a) power, (b) core outlet temperature, (c) core inlet temperature.

Despite the limited availability of temperature data during the transient, the increase in inlet and outlet temperature **Figure 34 (b), (c)** during the transient was in

good agreement with Mochizuki [61], where temperature increase $(0.75K < T < 1.5K)$ was within the expected range.

Figure 34 presents a broad view of the data, while **Figure 35** zooms in for a closer look. The comparison between experiment and model shows a discrepancy after 55 seconds of the transient, as the model's calculated power falls outside the range of experimental data. This difference can be attributed to the lack of consideration of the secondary loop in the 1D model, emphasizing the need to include all relevant factors for accurate results.

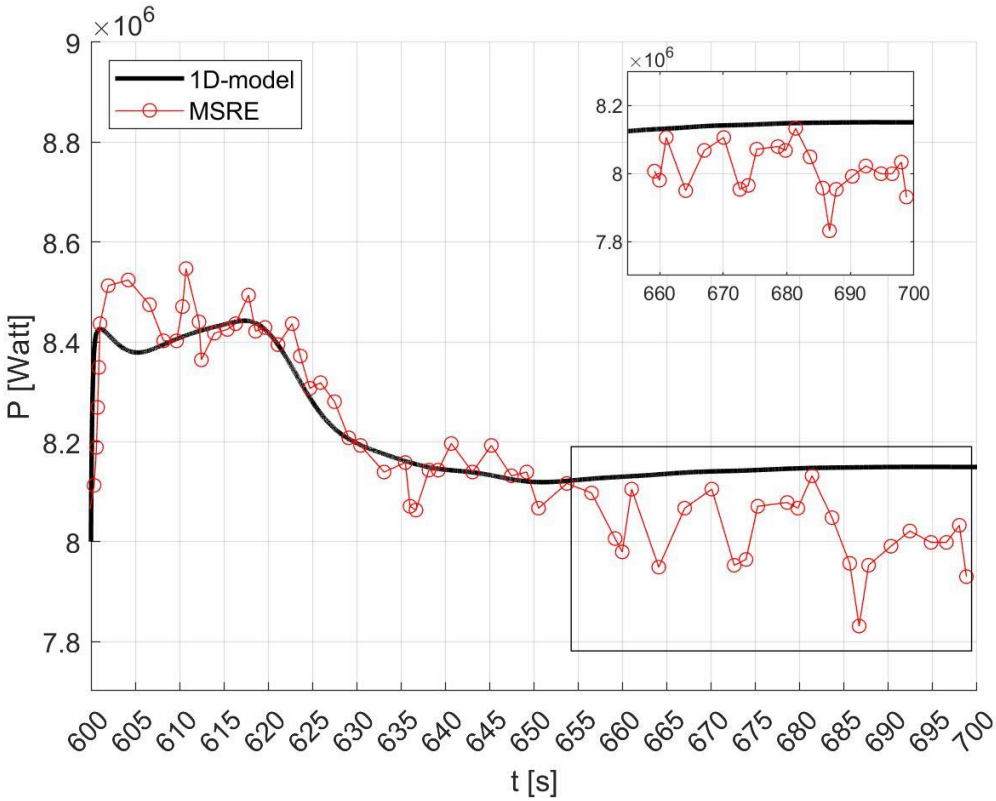


Figure 35: Power prediction of GeN-Foam 1D model performance for the reactivity insertion event at 8MW.

Figure 35 provides further insights into the differences between the model and experiment. At the power peak of the transient, the model and experiment exhibited a 1.03% discrepancy. This discrepancy increased to 4.06% after 85 seconds of the transient. Nevertheless, the power behavior observed during the transient exhibited a trend consistent with the experiment.

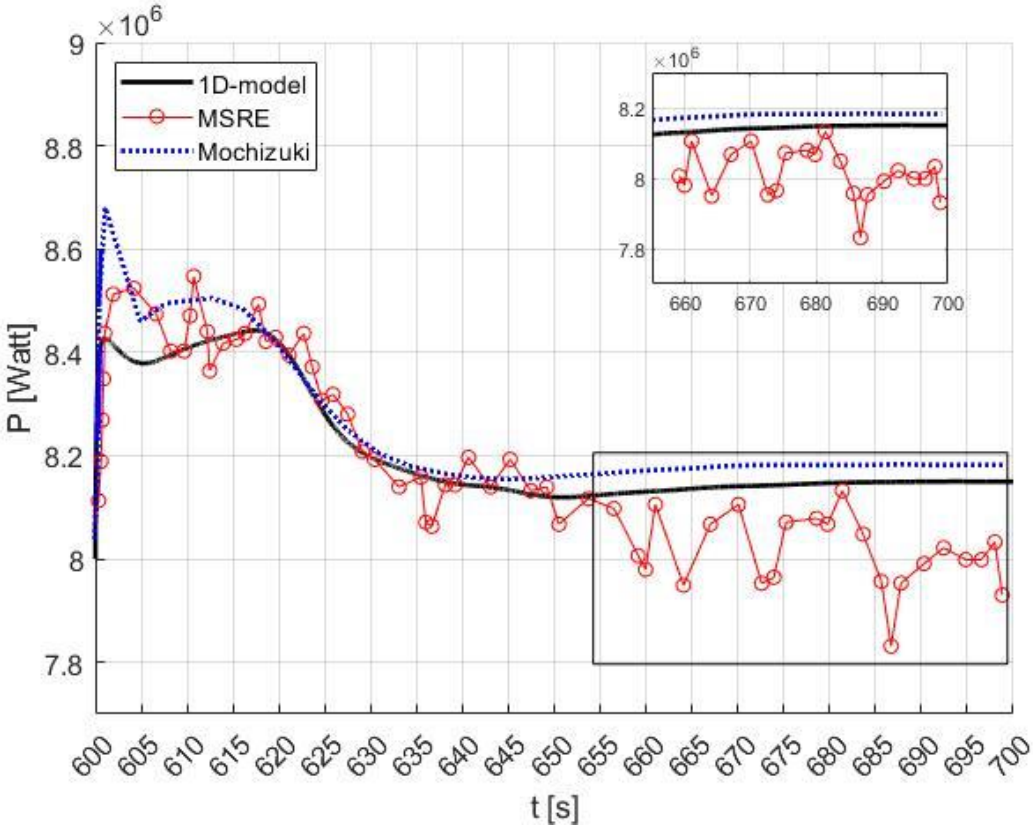


Figure 36: Power behavior comparison of GeN-Foam 1D and RELAP5-3D models against MSRE for the reactivity insertion event at 8MW "Adopted from [61]".

Figure 36 compares the results with those from literature. The 1D model and the model from literature [61] have one thing in common - the absence of the secondary loop and heat exchanger in their design. This means that the GeN-Foam model's success

in modeling the transient is noteworthy, and the deviation seen in **Figure 35** and **Figure 36** at 55 seconds can be attributed to the lack of consideration of the secondary loop in the model.

GeN-Foam assumes a constant average temperature in its heat exchanger model, which results in the temperature in the heat exchanger remaining unchanged during the transient. As a result, the model does not account for any increase in temperature in the secondary loop. This assumption affects the accuracy of the fuel feedback in the core, as a lower temperature leads to higher density and higher fission cross-section, causing an overestimation of power after 55 seconds of the transient.

Overall, the good match between the experiment and simulation results suggests that the model used in the simulation is a reliable and accurate representation of the behavior of the reactor during a reactivity insertion transient. This can provide valuable information on the dynamics of the reactor during transients and can be used to ensure whether the reactor's design and operation are safe and reliable.

Delayed neutron precursors (DNP), play a crucial role in the control of nuclear fission reactions. They are isotopes that emit low-energy neutrons some time after undergoing fission. These neutrons are used to regulate the reaction rate in a nuclear reactor, helping to maintain a steady and controlled power output.

In Molten Salt Reactors (MSRs), precursors can also be used to achieve improved reactivity control and enhanced safety features. Because MSRs operate at high temperatures, they can achieve higher fuel burnup compared to traditional water-cooled reactors. However, this requires precise control of the fission reaction rate to prevent

overheating. Precursors play a role in enabling this control. **Figure 37** presents a graphical representation of the DNP distribution within the model, highlighting that the first three groups have the highest concentrations and are almost evenly spread throughout the entire loop. On the other hand, the last three groups demonstrate a higher concentration in the core, with their concentration diminishing as it moves away from the core. It is important to note that comprehending the distribution of these precursors is crucial for the reactor to operate safely and efficiently.

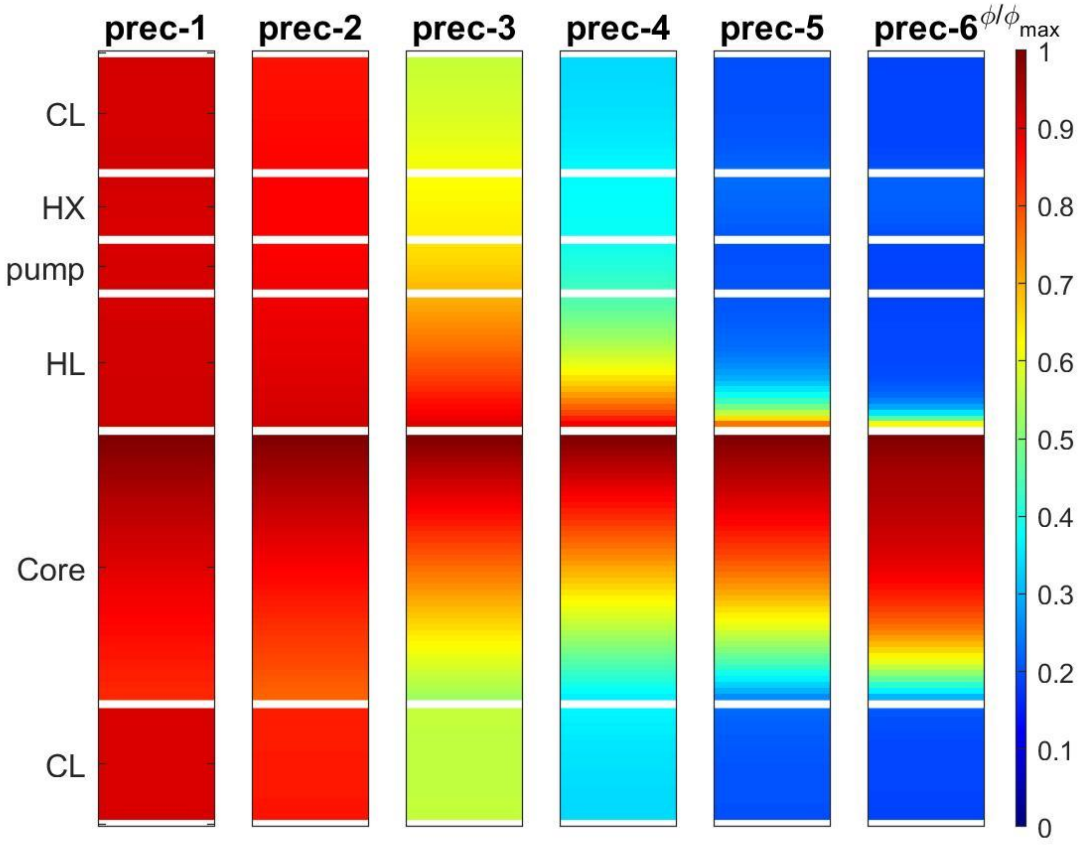


Figure 37: Contour of DNP concentrations in the primary loop as computed by 1D GeN-Foam model.

Figure 38 displays the spatial distribution of the precursors, characterized by a generally uniform, increasing pattern of concentration across the core, with peak values at the outlet of the core. However, the concentration decreases beyond the core, with a more pronounced drop observed for precursors 2-5. This distribution is influenced by the decay constant of each precursor and the mass flow rate or residence time of the salt in each component.

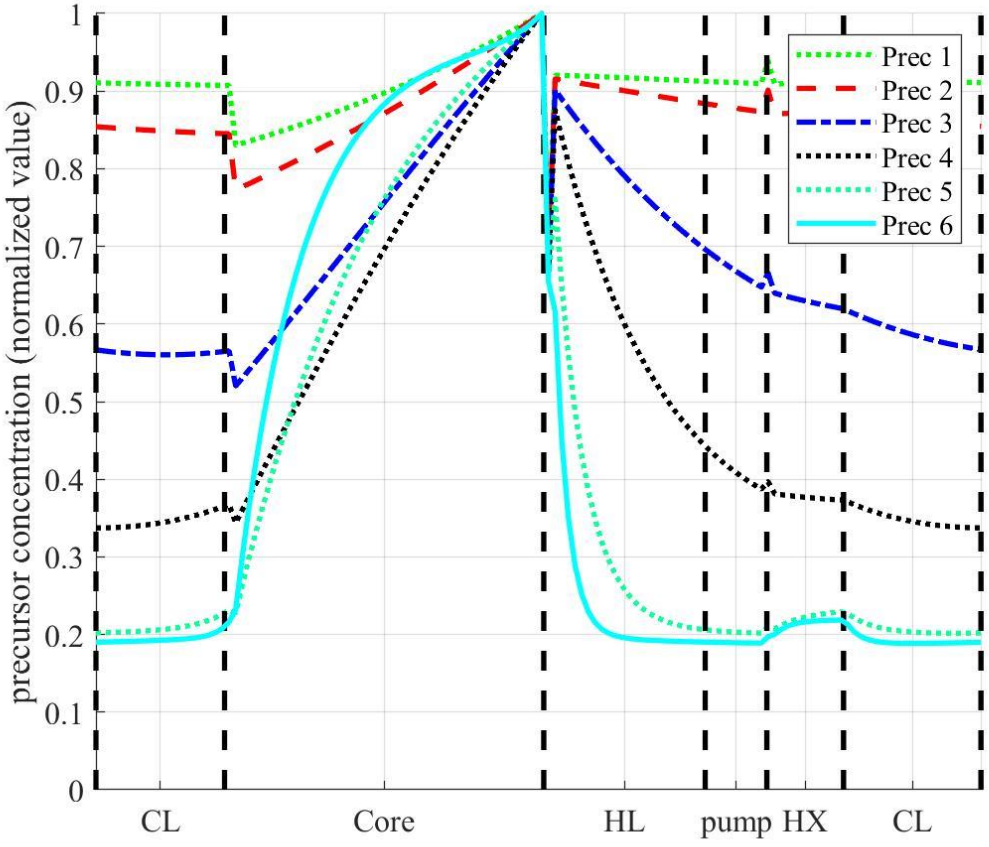


Figure 38: DNP concentrations in the primary loop as computed by 1D GeN-Foam model.

Figure 39 displays a transient of 50 pcm reactivity insertion and highlights the contrast between GeN-Foam 1D model and another previously created 1D model. By comparing the power and temperature, it becomes evident that GeN-Foam model aligns more closely with the experimental data than the other Multiphysics model. However, there is a disparity in the initial power peak prediction, which is due to the lack of implementation of the power portion produced in the moderator in this version of GeN-Foam solver. The temperature misprediction is a result of the aforementioned misprediction in **Figure 33**, which shows a difference of 2 degrees. Nonetheless, when compared to the experiment, the model performed well and surpassed the previously developed 1D models.

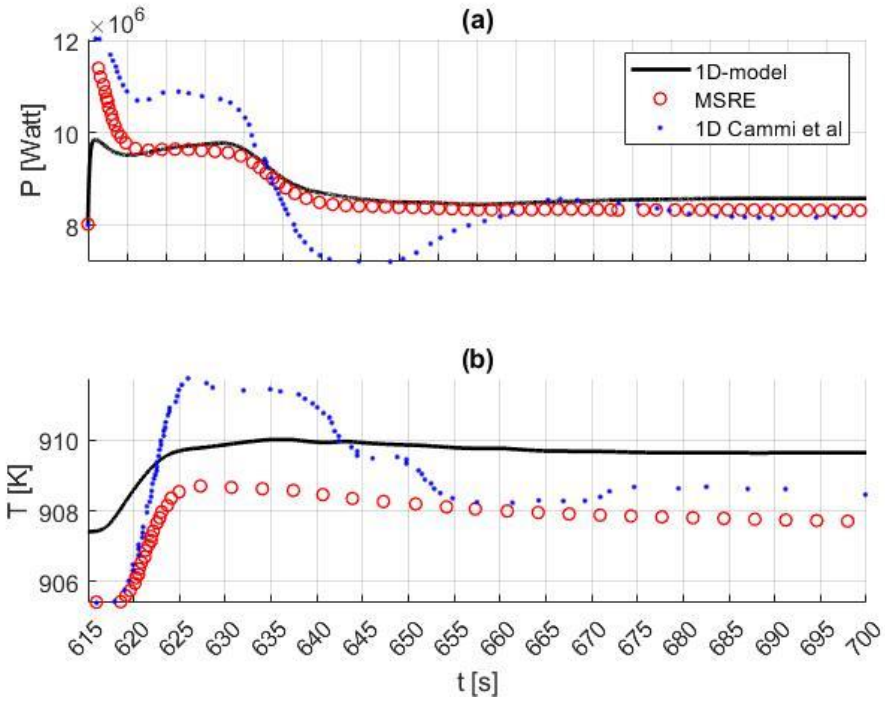


Figure 39: (a) Power and (b) Temperature prediction of GeN-Foam 1D model performance for the reactivity insertion (50pcm) event at 8MW "Adopted from [49]".

Conclusions

GeN-Foam model for the simulation of MSRE primary loop has undergone a successful validation of its sub-solvers for thermal-hydraulic and neutronics analysis. The results displayed a substantial level of conformity with experimental data, demonstrating the model's efficacy. However, there remains a degree of uncertainty between the experimental results and the 1D model.

This uncertainty is attributed to various sources, including discrepancies in the thermophysical properties and numerical inaccuracies due to the grid refinement level utilized in GeN-Foam model. To better comprehend this uncertainty, further analysis through uncertainty quantification methods is necessary.

Additionally, the limitations of the GeN-Foam model have been noted, including the need for modifications to accurately predict power production in graphite and the inability to simulate multiple reactivity transients simultaneously.

In the current study, it is anticipated that the temporal discretization uncertainty will be of lesser significance in comparison to the spatial discretization uncertainty and input uncertainty. This is due to the implementation of a small time step size to meet the CFL criterion of less than one and the utilization of stringent convergence criteria.

This validation represents a crucial milestone in the advancement of GeN-Foam as a trustworthy tool for thermal molten salt reactor simulations and paves the way for future progress in the field.

CHAPTER IV

UNCERTAINTY ANALYSIS

It is crucial to determine the reliability of the turbulence and Multiphysics models utilized in this study by evaluating the effects of mathematical assumptions and numerical methods on the results. All models, regardless of whether they are mathematical or not, have some degree of uncertainty. The uncertainty in numerical models usually arises from two factors: numerical uncertainty and input uncertainty. The sensitivity study was conducted in two phases: numerical and input uncertainty analysis.

Numerical phase of the sensitivity analysis employed the Grid Convergence Index (GCI) [80] as a metric for evaluating the accuracy of the numerical solutions. GCI is utilized to ensure that the results of the sensitivity analysis are robust and reliable by determining the convergence of the numerical solutions as it becomes independent of mesh size.

Input uncertainty phase of the sensitivity analysis aims to determine the influence of uncertain input parameters on the outcome of the model. For Chapter II, the sensitivity analysis is performed using one-factor-at-a-time (OFAT) approach [81], in which individual input parameters are varied systematically while keeping all other parameters constant. For Chapter III, Latin Hypercube Sampling (LHS) method is utilized to perform sensitivity analysis, providing a more randomized sampling of the input space, which was followed by linear regression analysis to predict the correlation between the input parameters of the simulation.

The results of the sensitivity analysis in both chapters contribute to a comprehensive understanding of the system under investigation and how it responds to changes in input parameters. This information is valuable for improving the accuracy of simulations and understanding the effects of varying input parameters on the results.

Numerical Uncertainty

The first step towards a numerical representation of a model is the discretization of the partial differential equations governing the system. The level of uncertainty related to this discretization process is significantly influenced by the refinement and structure of the computational grid and the numerical approach implemented. Ideally, the solution would tend towards greater accuracy as the computational grid is refined, however, in some circumstances, the solution may manifest non-convergent or oscillatory behavior, hindering the evaluation of the uncertainty [24].

In order to validate the meshing process, a grid independence study was conducted, where the numerical error (δ_{num}) was estimated by using a grid convergence index (GCI) parameter to quantify the error [80].

GCI is calculated via a process that begins by acknowledging that the difference between the numerical solution $f(N)$ at a mesh size N (number of elements) and the exact solution f_{exact} can be expressed according to:

$$E = f(N) - f_{exact} = A \cdot (N)^p + HOT \quad \text{(Eq. 55)}$$

where p is the formal order of accuracy and HOT represents higher order terms. By neglecting the HOT and considering results from three meshes (1, 2, and 3) with N1

> N2 > N3, and a constant refinement factor ($r = N_2/N_3 \ = N_1/N_2$), the observed order of accuracy (p_{obs}) can be found from:

$$\frac{f_3 - f_2}{f_2 - f_1} = r^{p_{obs}} \quad \text{(Eq. 56)}$$

where f represents the validation metric, or measurement of interest. The GCI represents the numerical error (δ_{num}), and can be quantified according to:

$$\delta_{num} \equiv GCI = \frac{F_s}{r^p - 1}|f_2 - f_1| \quad \text{(Eq. 57)}$$

where F_s is a factor of safety (assume F_s = 1.25 such that $p_{obs} \sim p$). This analysis assumes that results of interest are found within the asymptotic range for all three meshes and that the HOT can indeed be negligible.

GCI is a key component in ensuring the reliability of the mesh prior to conducting any simulations. Although GCI is presented in the current chapter as part of the uncertainty analysis, its significance has been emphasized throughout the preceding chapters as a crucial step in the pre-simulation process.

In this study, the temporal discretization uncertainty is expected to be less significant compared to the spatial discretization and input uncertainty, as a result of using a small/adjustable time step size to ensure that Courant number is less than one In addition, a strict convergence criteria were applied for all CFD and Multiphysics simulations.

GCI-RANS

For the 2D pipe model, the size of the element is reduced until independence is reached in the simulation output (velocity, and Nusselt number in this case) in order to find the best grid size for the present study.

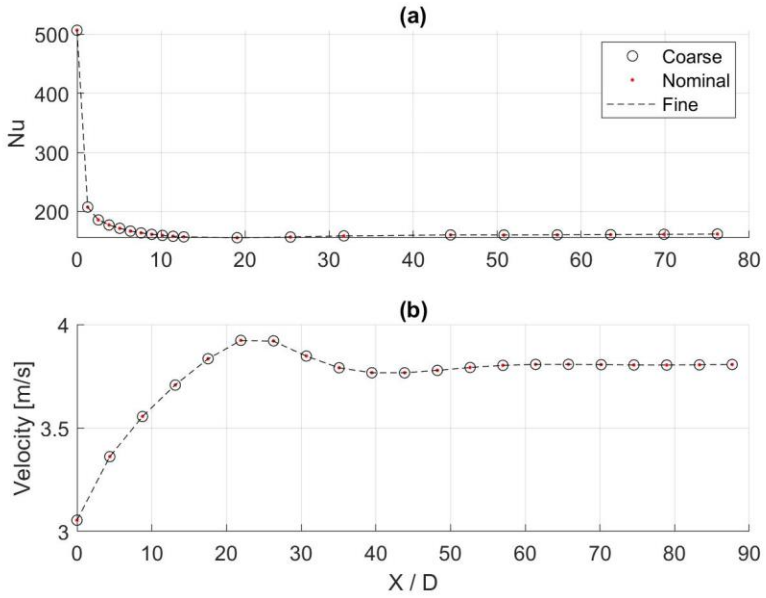


Figure 40: Grid refinement results for (a) Nusselt number, and (b) the centerline streamwise velocity.

Three different grid sizes (N1 = 402,369, N2 = 205,506 and N3 = 104,720) are tested to find out the effect on the previously mentioned parameters of interest. It has been found that there is no significant change in centerline velocity, and Nusselt number (GCI = 1.7096e-04 m/s and 0.0226 respectively). Streamwise velocity and Nusselt number for the three grids shown in **Figure 40**. revealed that the solutions are grid independent to acceptable levels. Therefore, the medium mesh (N= 205,506) has been chosen to perform all simulations for the present study.

GCI-GeN-Foam

GCI analysis is performed to investigate the effect of mesh size reduction on the simulation results. The number of cells is increased from 400 to 1600 in increments of a refinement factor (r) that equals to 2. **Figure 41** (a) depicts the temperature profile for the steady state case, while Figure 1 (b) shows the power profile for a transient.

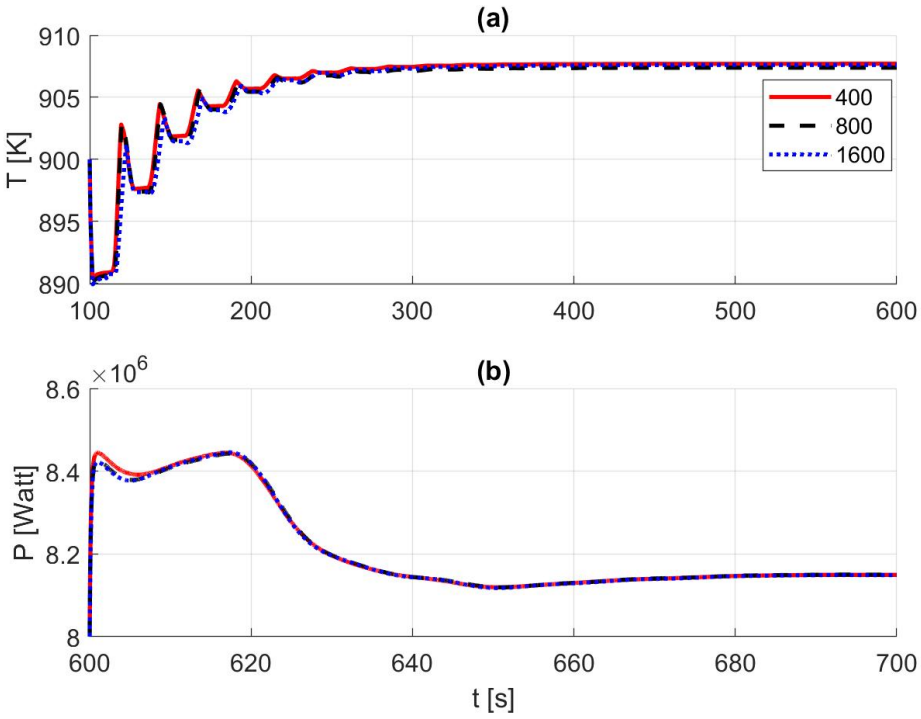


Figure 41: Grid refinement results for (a) steady state temperature, and (b) power.

The results showed no significant difference between the mesh with 800 cells and the mesh with 1600 cells. As a result, 800 cells are selected as the nominal mesh size to run all subsequent simulations because it has been found that there is no significant change in temperature, and power (GCI = 0.9512 K and 7.84×10^3 Watt respectively).

This step is taken to ensure that the mesh size would not have any further impact on the accuracy of the GCI analysis.

Input uncertainty

Uncertainty-RANS

Sensitivity Analysis of a numerical model studies the variations in the output due to variations of its inputs. The terminology adopted in this work is one at a time (OAT) [82], where input factors of the simulation vary from their nominal one at a time, whereas the other factors are kept fixed. This approach is common in thermal-fluids computational works [83] and it is consistent with recommendations from the ASME V&V-20 standards [81]. The model can be expressed in a vector form as follows:

$$y = g(x), \qquad \begin{cases} x = [x_1, x_2, \ldots, x_n] \\ y = [y_1, y_2, \ldots, y_n]^T \end{cases} \qquad \text{(Eq. 58)}$$

The left-hand side (LHS) variable of Eq. 58 represents validation metrics, Nu and friction factor were chosen to be the validation metrics in this work. The right-hand side (RHS) variables represent multiple parameters, such as: dynamic viscosity, density, thermal conductivity, pipe length, pipe diameter, and turbulent Prandtl number Pr_t.
In this work, the scalar output variable is defined by running the simulation using the nominal value of input parameters. The next step is to run multiple simulations with changing (perturbing) the input variables (parameters), where perturbation is constrained by the value of the estimated uncertainty designated for each input parameter. After that, the output sensitivity to the ith parameter is measured using the so-called sensitivity coefficient (S_i), which is defined as follows:

$$S_i(\bar{x}) = \frac{\partial g}{\partial x_n} \quad \text{(Eq. 59)}$$

Where $\bar{x}$ is the nominal value for the parameters of interest. The sensitivity coefficient of a certain parameter (input, x_n) reveals its effect on the output variable g. Now, the uncertainty of function f can be calculated using the following series as follows:

$$u_f = \sqrt{\sum_{n=1}^{\infty} (\frac{\partial g}{\partial x_n} \cdot \delta x_n)^2} \quad \text{(Eq. 60)}$$

Where $\frac{\partial g}{\partial x_n}$ is the partial derivative of the relation for g with respect to the nth independent variable x_n, and $\delta x_n$is the estimated uncertainty in the independent variable x_n. Eq. 58 is very useful for simple cases, however, for complex numerical simulations like our case, it is more efficient to linearize the terms in Eq. 59, so it becomes:

$$S_i(\bar{x}) = \frac{\Delta g}{\Delta x_n} \quad \text{(Eq. 61)}$$

For this assumption to be valid, the input parameters shouldn't be significantly correlated, and Gaussian distribution is assumed between inputs and outputs [8]. In the analysis of this work, the previously mentioned six parameters were perturbed with upper and lower limits depending on their uncertainties, to account for the effect of these perturbations on Nu and friction factor f [33]. This was done after applying fitting analysis for each property in the simulation of the nominal values. The perturbation was applied without changing the shape of the obtained behavior in the nominal simulation. It is worth noting that Re was fixed to be $3 \times 10^4$and heat flux at the value 637.9437

KW for all runs so that the variations in the output would be due to input parameters perturbations only.

Table 17: Nominal simulation inputs and their assigned uncertainties

Variable (x_n)	**Viscosity (μ)**	**Conductivity (k)**	**Diameter (D)**	**Density (ρ)**	**Length (L)**	**Specific heat (Cp)**
Nominal Value	3.2×10^{-3}	0.918	0.007874	2000	0.6906	1905
Uncertainty %	2	2.5	0.1	2	0.1	10

Post processing as a final step of the analysis included two terminologies; Quantitative Sensitivity Analysis, where each input factor is associated with a quantitative evaluation of its relative effect, and Qualitative Sensitivity Analysis, where sensitivity is estimated qualitatively by visual inspection of model outputs variables. It was noticed that due to perturbations, thermal properties such as heat transfer coefficient h and Nu have changed and velocity profile varied, thus the fully developed length showed a slight change. **Table 17** shows the values of uncertainty for each input parameter.

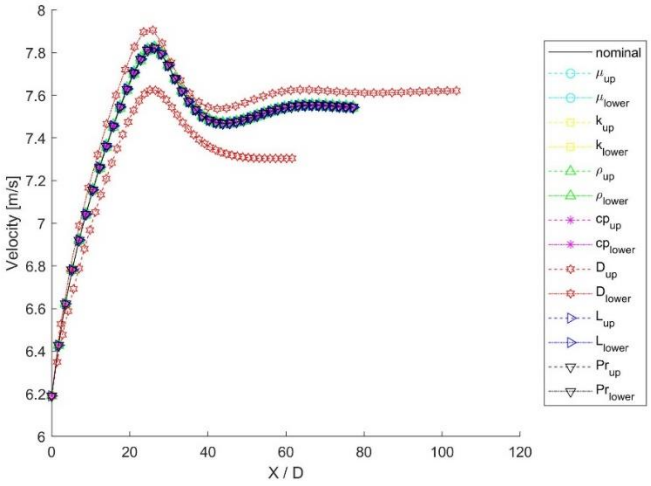


Figure 42: Velocity profile variation along streamline upon changing input parameters for $k-\epsilon$ model.

Figure 42 shows velocity profiles as a function of input parameters. It is noticeable that velocity profile changes depending on the effect of the input parameter on the model, and it is not necessary that they all have the same effect. For instance, diameter perturbations recorded the largest effects on velocity, while viscosity variation had minor effects. Such effect of diameter perturbation is anticipated because most of the thermal and hydrodynamic parameters are either directly or indirectly related to the tube diameter. The largest value of uncertainty in the velocity of 3.2% resulted from diameter perturbation, while the other parameters effect was less than 0.07%. It should be noted that Pr is computed based on results based on viscosity, density and diameter. Nusselt number is one of the most important variables in heated pipe flows, thus, it is convenient to use it as a validation metric. **Figure 43** shows Nusselt number along the pipe for the five turbulent models.

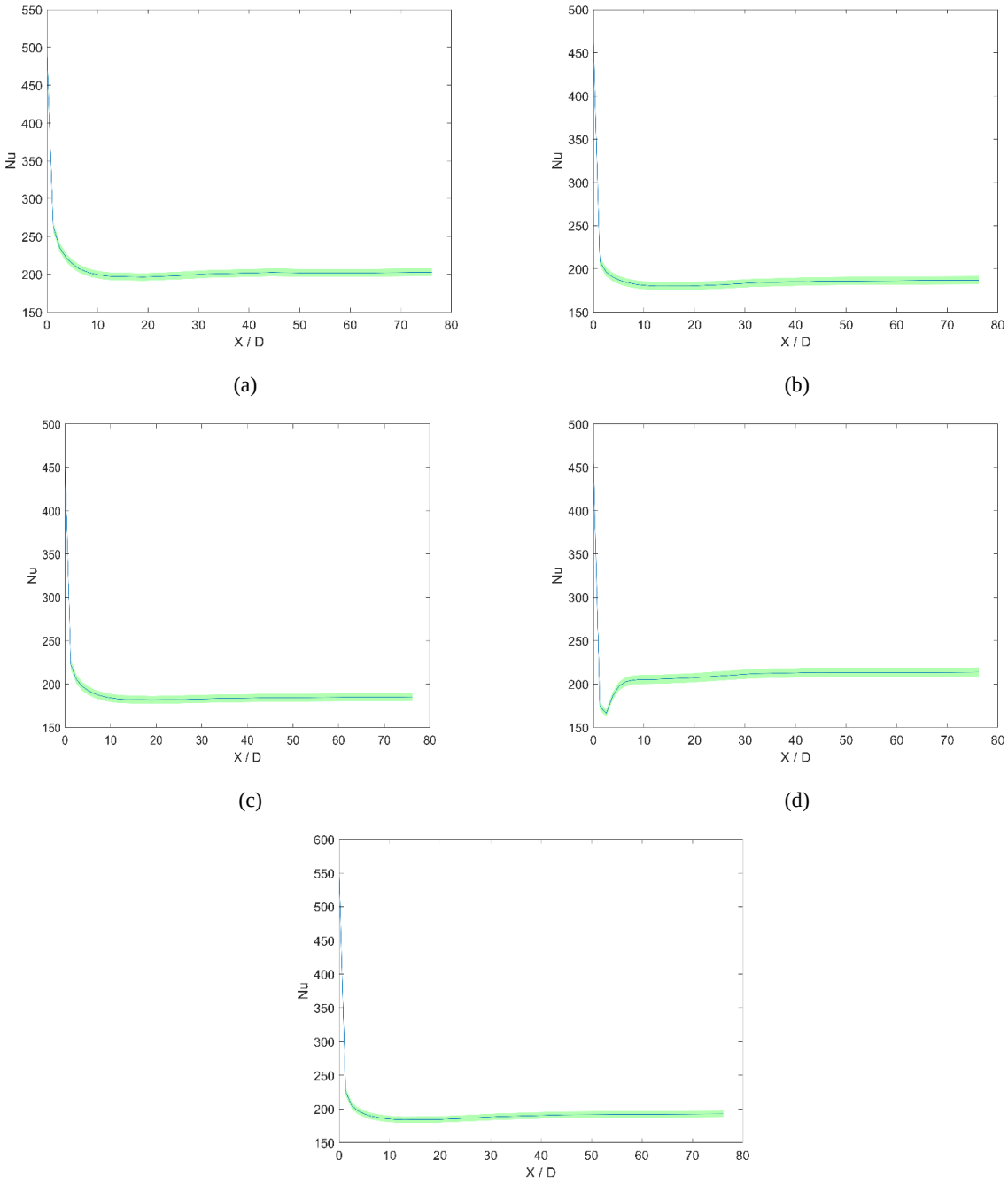
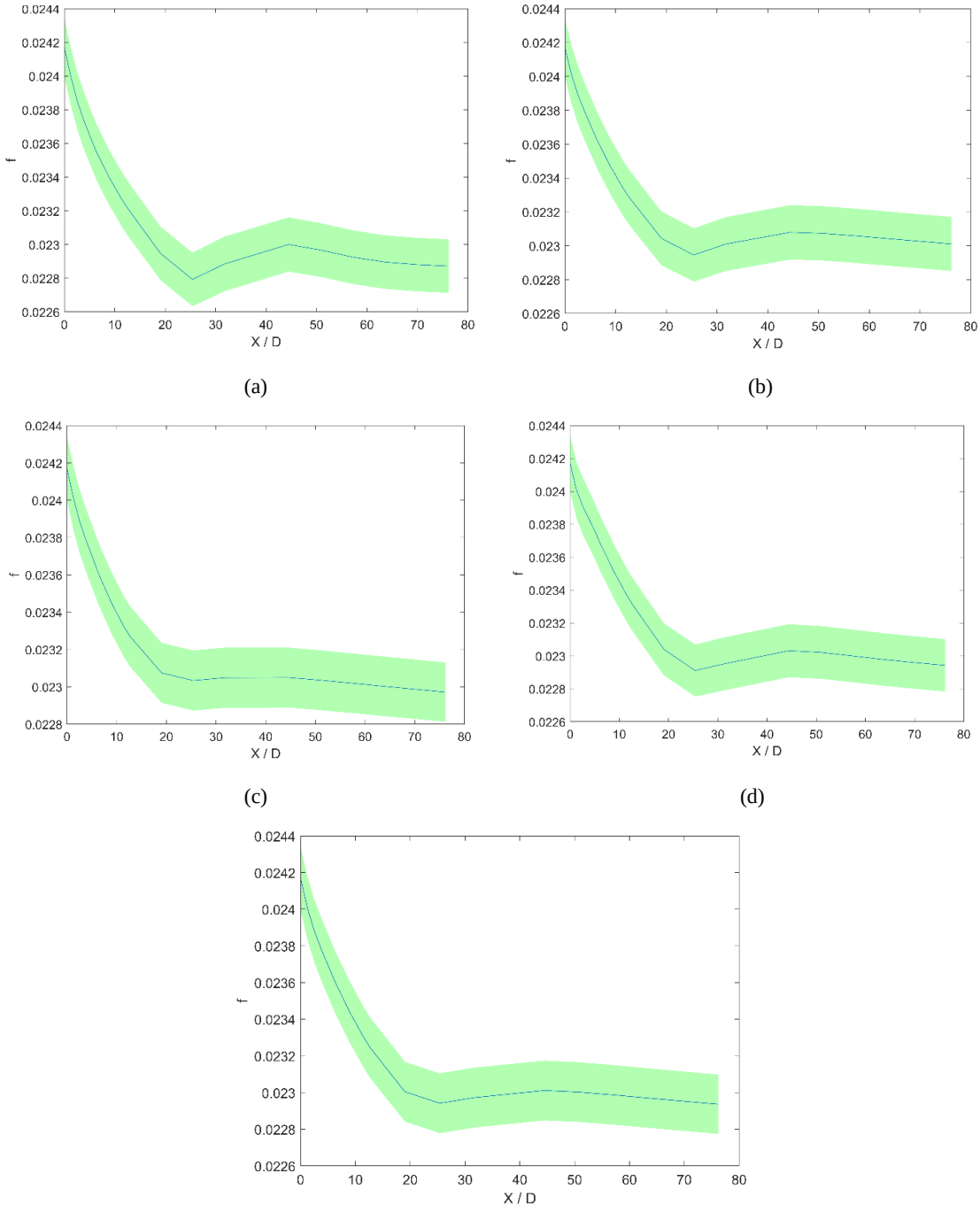


Figure 43: Nusselt number along the axial direction for different models, a. $k-\epsilon$, b. $k-\omega$ SST, c. $k-\omega$, d.V2F, e. SA

The uncertainty roughly ranged from ~2.6% at the inlet down to ~2.5% at the outlet, where no significant variation was noticed between models. This value of uncertainty was the total value of the uncertainties resulted due to perturbations in the input parameters.

From hydrodynamic perspective, f was chosen to be a second validation metric. Its behavior varies from one model to another (**Figure 44**), where the most abnormal behavior can be seen in $k-\epsilon$ and V2F models. However, this didn't affect the uncertainties. Overall, the uncertainty in all cases varied around 0.7%.



(a)

(b)

(c)

(d)

(e)

Figure 44: friction coefficient along the axial direction for different models, a. $k-\epsilon$, b. $k-\omega$ SST, c. $k-\omega$, d.V2F, e. SA

Another hydrodynamic characteristic that was included in the analysis is l_h. **Figure 44** shows l_h of nominal values for the five turbulent models. The input uncertainty was investigated on 13 different points along the pipe and error bars are presented to better illustrate the uncertainty of each model. Input uncertainties for all models ranged between $(5.3 * 10^{-3}\% - 6.9 * 10^{-3}\%)$

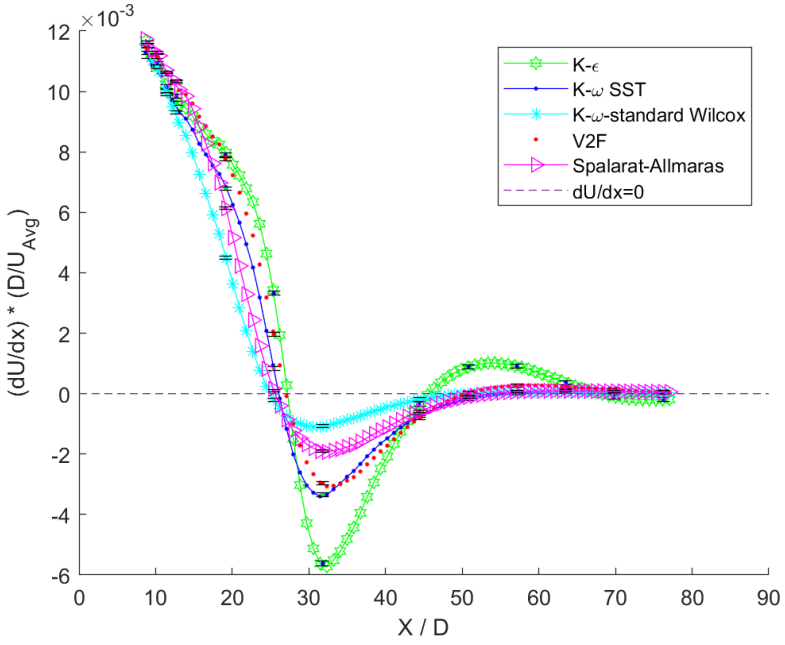


Figure 45: Entry length with the uncertainty for all models

$k - \omega$ and Spalarat-Allmaras were the most stable among other models in terms of converging to the horizontal line ($dU/dx = 0$). k-ω SST hasn't converged as smooth as the two other models, but it converged earlier than all other models. From Figure 45, qualitatively the most deficient model to converge was $k - \epsilon$. V2F faced some difficulty converging as well.

In some cases that entry length exceeds 60 diameters to converge when the normalized definition ($dU/dx = 0$) is used for fully developed length. **Table 18** includes the entry lengths for the models and their related uncertainties with the highest Re ($Re = 3 \times 10^4$). Accordingly, it isn't possible to adapt the terminology that was used to define entry length earlier, because the velocity gradient crosses the horizontal line two times without reaching exact value of ($dU/dx = 0$), thus, the entry length was defined on the second time the velocity gradient intersected the horizontal line of $dU/dx = 0$, except for the case of $k - \epsilon$ where it crosses the line three times.

Table 18: Hydrodynamic entry length.

Model	Entry length	Uncertainty
$\boldsymbol{k-\epsilon}$	67	$\pm$5.8
$\boldsymbol{k-\omega\ SST}$	56	$\pm$5.0
$\boldsymbol{k-\omega}$	51	$\pm$4.3
V2F	51	$\pm$4.6
SA	54	$\pm$4.6

The entry length ranged from 51 up to 67 as can be seen in **Table 18**. The lower bound of entry length has notably increased compared to **Figure 10**, which is because of the difference between the two criteria in defining the entry length, where the new criterion is stricter in terms of defining the l_h.

Uncertainty-GeN-Foam

The input error is computed using Latin Hypercube Sampling (LHS) [84], which is a statistical sampling technique that is commonly used in uncertainty analysis, computer experiments and data simulation. It aims to evenly distribute sample points across the domain of a function, avoiding clustering of points and ensuring that each parameter has an equal representation in the sample.

The technique of dividing the parameter domain into Y intervals defined by function F, with Y being the number of samples required, results in a sample set where each parameter is equally represented. This can be achieved by selecting one sample randomly from each interval. By creating a matrix of ranks R with $n \times k$ elements, and denoting the jkth element as r_{jk}, it is possible to obtain a latin hypercube sample Z as in the following Eq:

$$Z_{jk} = F_k^{-1}(n^{-1}(r_{jk} + \gamma_{jk} - 1)) \quad \text{(Eq. 62)}$$

LHS is adopted in this work due to its advantages over other uncertainty analysis methods, including:

1. Improved distribution of samples: LHS evenly distributes samples across the domain, ensuring that all regions are well represented and reducing the risk of missing important areas.
2. Efficient use of resources: LHS is a relatively efficient method, requiring fewer samples than other methods, such as Monte Carlo simulation, to achieve a given level of precision.

3. Improved model prediction: LHS provides a more accurate representation of the model than other methods, reducing the risk of biased results and leading to better model predictions.
4. Easier to implement: LHS is a simple and straightforward method to implement, making it an accessible option for users with limited computational resources or experience.

Table 19: Nominal simulation inputs and their assigned uncertainties "Adopted from [33] [61] [75] [78] [85]".

Variable	Nominal Value	Min	Max	Uncertainty %	Unit
Heat exchanger temperature (T_{hx})	876	864.97	887	1	K
Heat transfer coefficient (h)	6.1539×10^3	6.04×10^3	$6.295.4 \times 10^3$	2	W/m^2.K
Density (ρ)	2245	2222	2271.9	1	Kg/m^3
Specific heat (c_p)	2386.5	2300	2486.8	3	j/kg.K
Viscosity (μ)	0.007	0.000153	0.016309	43	Pa.s
Prandtl number (Pr)	16.66	16.5	16.802	0.1	[-]
Fuel feedback coefficient (α_f)	-11.034×10^{-5}	-15.4×10^{-5}	-7.6386×10^{-5}	27	Pcm/K
Moderator feedback coefficient (α_m)	-5.814×10^{-5}	-10.2×10^{-5}	-2.5482×10^{-5}	52	Pcm/K
Decay constant (λ_1)	0.0126	0.0104	0.014825	16	1/s
Decay constant (λ_2)	0.0337	0.0294	0.03769	9	1/s
Decay constant (λ_3)	0.139	0.102	0.17984	20	1/s
Decay constant (λ_4)	0.325	0.284	0.35255	7	1/s
Decay constant (λ_5)	1.13	1.12	1.1427	1	1/s
Decay constant (λ_6)	2.5	1.91	3.0564	20	1/s
Precursor (β_1)	22.8×10^{-5}	17.3×10^{-5}	30×10^{-5}	22	pcm
Precursor (β_2)	78.8×10^{-5}	73×10^{-5}	84.8×10^{-5}	6	pcm
Precursor (β_3)	66.4×10^{-5}	61×10^{-5}	72.9×10^{-5}	7	pcm
Precursor (β_4)	73.6×10^{-5}	67.7×10^{-5}	79×10^{-5}	7	pcm
Precursor (β_5)	13.6×10^{-5}	7.61×10^{-5}	19.1×10^{-5}	36	pcm
Precursor (β_6)	8.84×10^{-5}	3.1×10^{-5}	14.2×10^{-5}	56	pcm
Reactor period (T_r)	4×10^{-4}	1.18×10^{-4}	5.91×10^{-4}	40	s

The heat transfer equation is used to analyze the effect of 21 input parameters that have been randomly sampled using LHS method, and the outputs generated through this process are then used to determine the corresponding System Response Quantities (SRQs). **Table 19** presents the parameters that were altered and input into the LHS algorithm, accompanied by their respective uncertainties. It is important to note that the uncertainties vary significantly, ranging from 1% to 56%, due to conflicting information from various sources regarding parameters such as μ, α_f, α_m, and Tr. The constants and precursors were perturbed based on the differences between U-233 and U-235 fuels, in order to take into account the impact of changing the fuel on transients.

Figure 46 shows a histogram that links the distribution of normalized power samples obtained through LHS to their frequency. The red dotted lines mark the average and the average plus/minus two standard deviations ($2\sigma = 0.0311$), which form a confidence interval that covers 95% of the normal distribution and display the spread of the input samples.

In a LHS sample, the normal distribution is often used as a model for the underlying population distribution, so the results from the LHS sample will also follow a normal distribution, and based on the distribution of normalized power displayed in **Figure 46**, the average Normalized power across the 60 input uncertainty calculations is observed to be $8.08 \times 10\text{^}6$ Watt, which is 1% higher than the value predicted using the analytical solution.

The data produced by Latin Hypercube Sampling (LHS) being normally distributed with 95% confidence interval indicates that the distribution of the data is

close to a normal distribution. The 95% confidence interval in this case represents the range of values within which we can be 95% confident that the true population mean lies.

Having data that follows a normal distribution with a tight confidence interval has several advantages in statistical analysis:

1. Normality assumptions: Many statistical techniques, such as linear regression and ANOVA, make assumptions about the normality of the data. If the data produced by LHS follows a normal distribution with a tight confidence interval, it suggests that these assumptions are likely to be met, making it easier to use these statistical techniques on the data.
2. Robustness: Normally distributed data is considered to be more robust than data that is not normally distributed. This means that the results obtained from statistical analysis of normally distributed data are less sensitive to outliers and extreme values.
3. Simplicity: Normally distributed data is easier to work with and interpret than data that is not normally distributed. This is because there are many well-established statistical methods for analyzing normally distributed data, and the results of these methods are often easier to understand and interpret than the results of methods designed for non-normally distributed data.

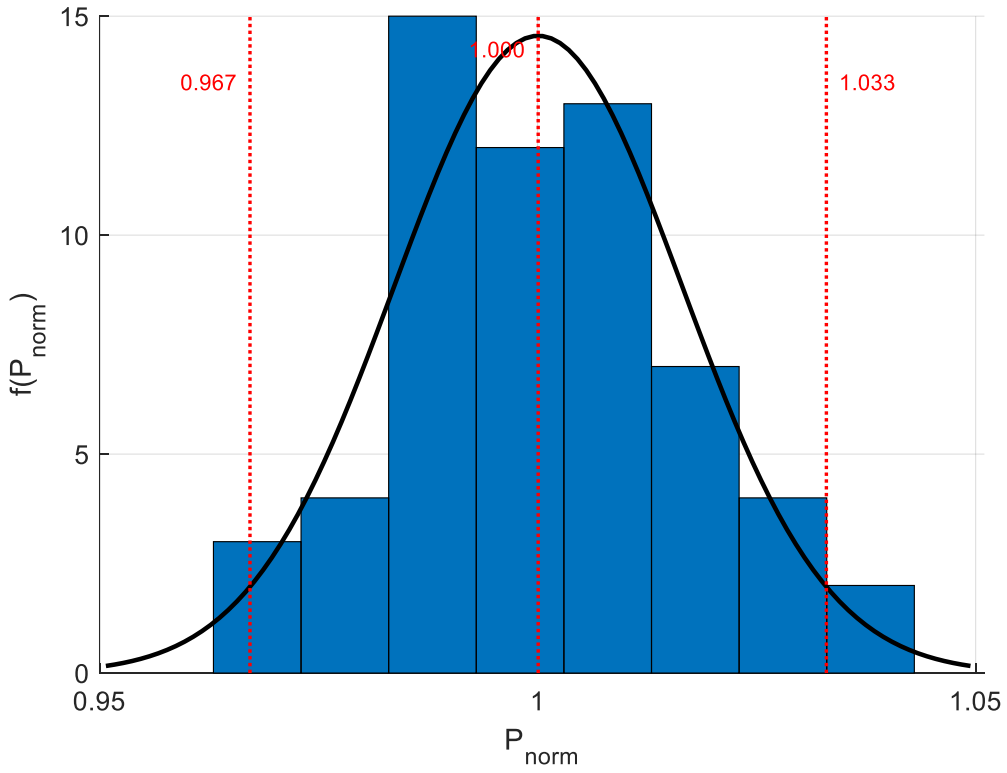


Figure 46: probability distribution function of normalized power from LHS input uncertainty analysis.

The input uncertainty for the initial power, as shown in **Figure 46**, has been calculated to be 4.83×10^4Watt for the initial power peak (0.57% relative uncertainty). However, the discretization uncertainty for the same quantity is 7.84×10^3Watt (0.16% relative uncertainty), which is less than the input uncertainty. Given that both the discretization and input uncertainties quantified, the total uncertainty in a computational system response quantity is computed by Eq.63 with a total uncertainty $\delta_{tot} = 4.89 \times 10^4$Watt.

$$\delta_{tot} = \sqrt{(GCI)^2 + (LHS)^2} \quad \text{(Eq. 63)}$$

The variation displayed in **Figure 47** reflects the spread of the calculated power values resulting from the Latin Hypercube Sampling applied to the inputs of the simulations. This uncertainty appears to be acceptable as it encompasses the difference between the model and the experimental results for most of the transient period. The mismatch with experimental values after 55 seconds of the transient, as explained in **Figure 36**, has been considered justified. It is possible that incorporating a secondary loop in the model might decrease the power values and bring them within the range of the experimental values.

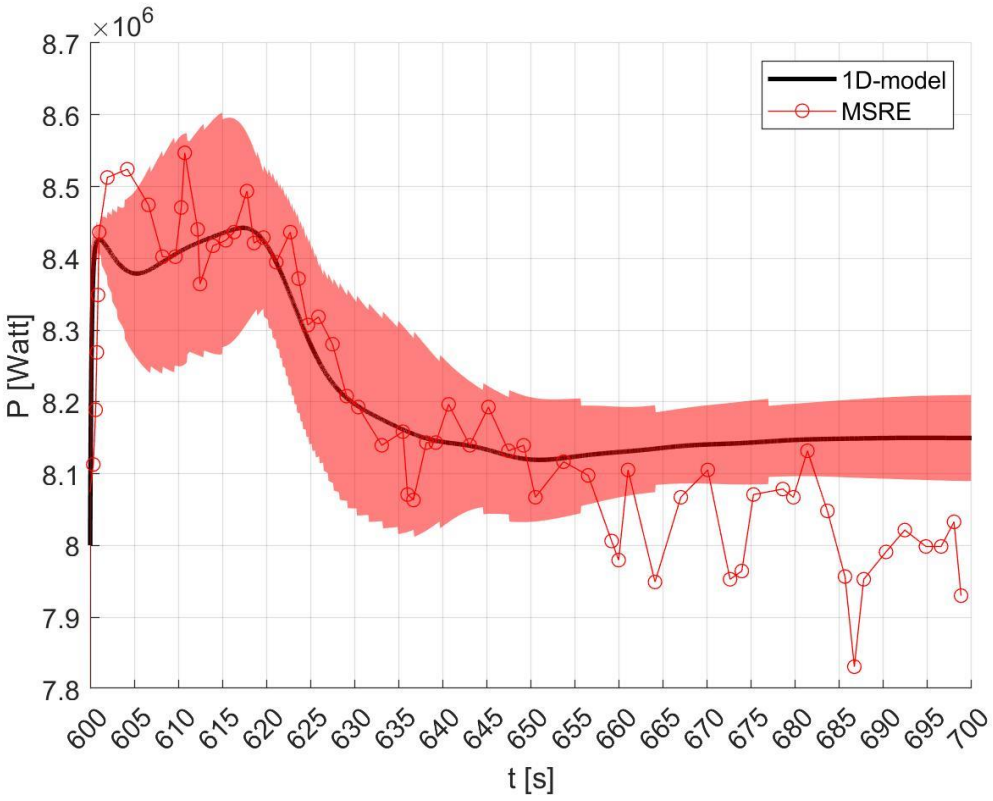


Figure 47: Power Input uncertainty obtained by LHS.

The current study incorporates linear regression as a statistical analysis tool to assess the relationships between input variables and output parameters in a molten salt reactor. The model is trained using a dataset derived from simulations performed in GeN-Foam. The trained linear regression model can then be used to predict the values of key parameters, such as temperature, density, heat capacity, and power, as a function of other inputs. This approach enables a more thorough comprehension of the reactor system behavior. The following section outlines the utilization of linear regression as a means of streamlining the process of understanding and optimizing the performance of the molten salt reactor.

Correlation between Inputs

In LHS, correlations between the inputs can play an important role in ensuring that the resulting samples are representative of the underlying distribution of the inputs. The goal is to spread the samples evenly across the range of each input variable, while preserving any correlations between variables, which helps to reducing sampling bias and increasing the accuracy of the results. In this study, it is anticipated to find an interaction between ρ, Cp, and ΔT as they are all part of the heat transfer rate equation. To further understand the relationship between variables, linear regression model is adopted to investigate correlations and interactions between other inputs.

Linear regression is a simple, yet widely used, statistical model that can be used to make predictions about the relationship between a dependent variable and one or more independent variables. The results of the regression analysis, such as the regression equation and coefficients, can be used to better understand the relationship between the

independent and dependent variables and to make accurate predictions for future data points.

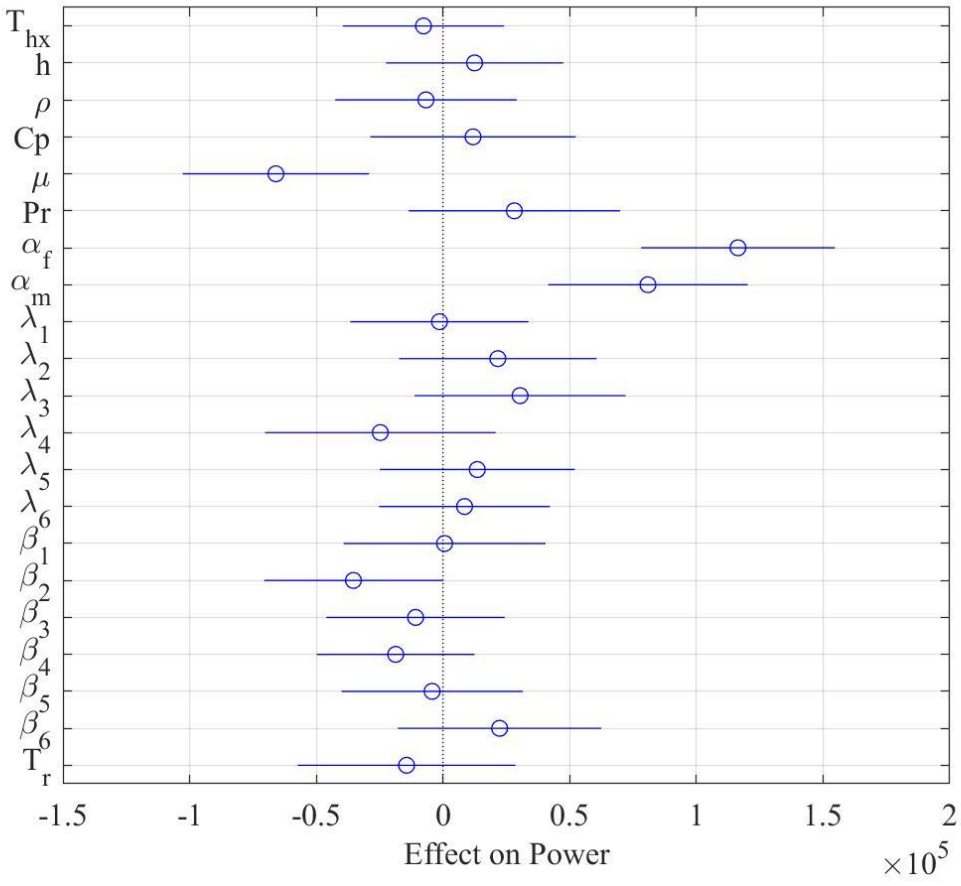


Figure 48: Effect of predictors (simulation inputs) on response (power)

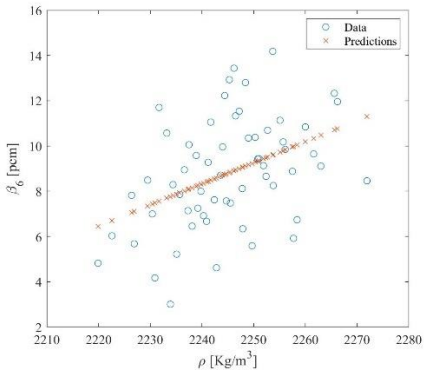
The linear regression workflow in MATLAB typically involves three main steps: defining the model, training the model, and making predictions. In this work, the command **mdl = fitlm(data, modelspec) is used** to fit a linear regression model to the data. By using the **fitlm** function utilizing a linear regression model of the form $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$, where y is the dependent variable, and $x_1, x_2, \ldots, x_n$

are the independent variables (predictors), and $\beta_0,\ \beta_1,\ \beta_2,\ \ldots,\ \beta_n$ are the coefficients to be determined by the optimization algorithm.

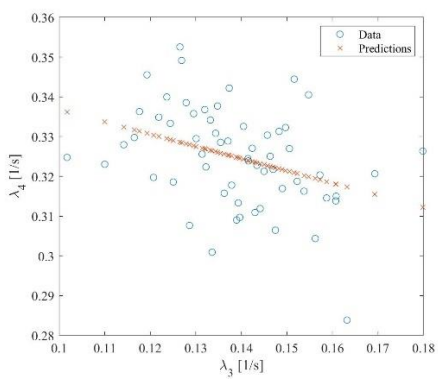
In the first step, the independent variables (predictors) are specified and the dependent variable by providing the **data** argument to the **fitlm** function (see **Table 19**). In the second step, the optimization algorithm was used to fit the model to the data and determine the optimal coefficients that minimize the difference between the predicted and actual values of the dependent variable. The **modelspec** argument to the **fitlm** function has been used to specify the names of predictors. Finally, trained model is used to make predictions for new data points by providing the values of the independent variables and receiving a predicted value for the dependent variable.

The confidence interval for the effect on the response is indicated by the length of each horizontal line in **Figure 48**. Each line represents a 95% confidence interval for the impact of a change in the corresponding predictor on the response. For instance, the estimated impact of varying the viscosity from its minimum to maximum (see **Table 19**) value is an increase of 0.015 with a 95% confidence interval ranging from $-1.02e5$ Watt to $-0.3e5$ Watt.The negative sign means that the effect of viscosity on the power is negative

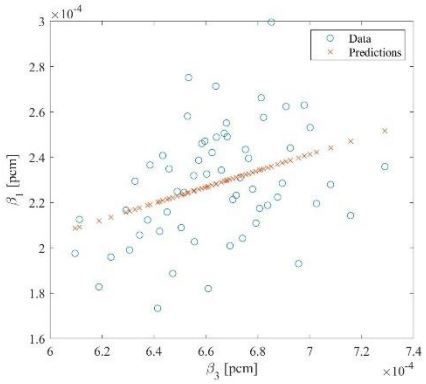
In a linear regression model, the correlation between the predictors plays a crucial role in determining the model's strength and accuracy. The correlations between each pair of inputs are presented in Appendix B, where the figures reveal the highest correlated variables, which range from 0.28 to 0.43.



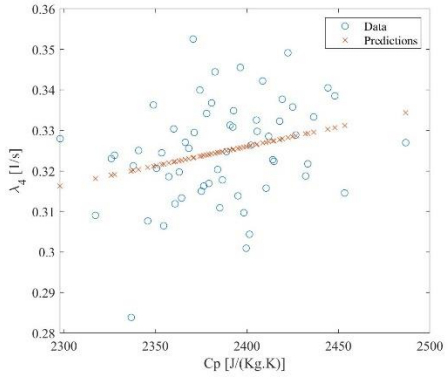
(a)



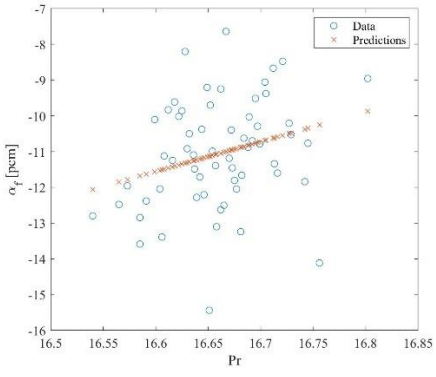
(b)



(c)



(d)



(e)

Figure 49: Highest fitted correlations between simulation inputs, (a) $\rho\ Vs\ \beta_6$, (b) $\lambda_3 Vs\ \lambda_4$, (c) $\beta_1 Vs\ \beta_3$, (d) $C_P\ Vs\ \lambda_4$, (e) $Pr\ Vs\ \alpha_f$.

The highest correlation was found between ρ and β_6, with a value of 0.43 (see **Figure 49** (a)), while **Figure 49** (d) and (e) have the lowest correlation of 0.28. Despite the highest correlation between ρ and β_6, it still falls below the commonly accepted limit for considering a correlation. The low correlation between the predictors can be recognized by the high P-value, which is used as an indicator of the absence of correlation.

Additionally, the R-squared values for all inputs are low, with a maximum value of 0.18. R-squared, also known as the coefficient of determination, or a measure of the proportion of variation in the target variable that is explained by the predictors in a regression model. Low R-squared values indicate a high variance or scattered data.
It's worth noting that in a linear regression model, high correlation between predictors is desirable as it suggests a strong relationship between the predictors and the target variable. However, low correlation between predictors can lead to unreliable and weak predictions.

It is important to differentiate between correlation and interaction among predictors in linear regression models. Correlation refers to the relationship between two variables and is a measure of the strength and direction of the linear association between two continuous variables. This relationship is quantified using a correlation coefficient, such as Pearson's or Spearman's rho.

Correlation analysis is a bivariate technique and does not account for the influence of other variables. Interaction, on the other hand, refers to the phenomenon where the effect of one predictor on the response variable is dependent on the value of

another predictor. This relationship is represented by an interaction term in the regression equation and captures the deviation from the expected relationship between the two predictors if they were considered individually.

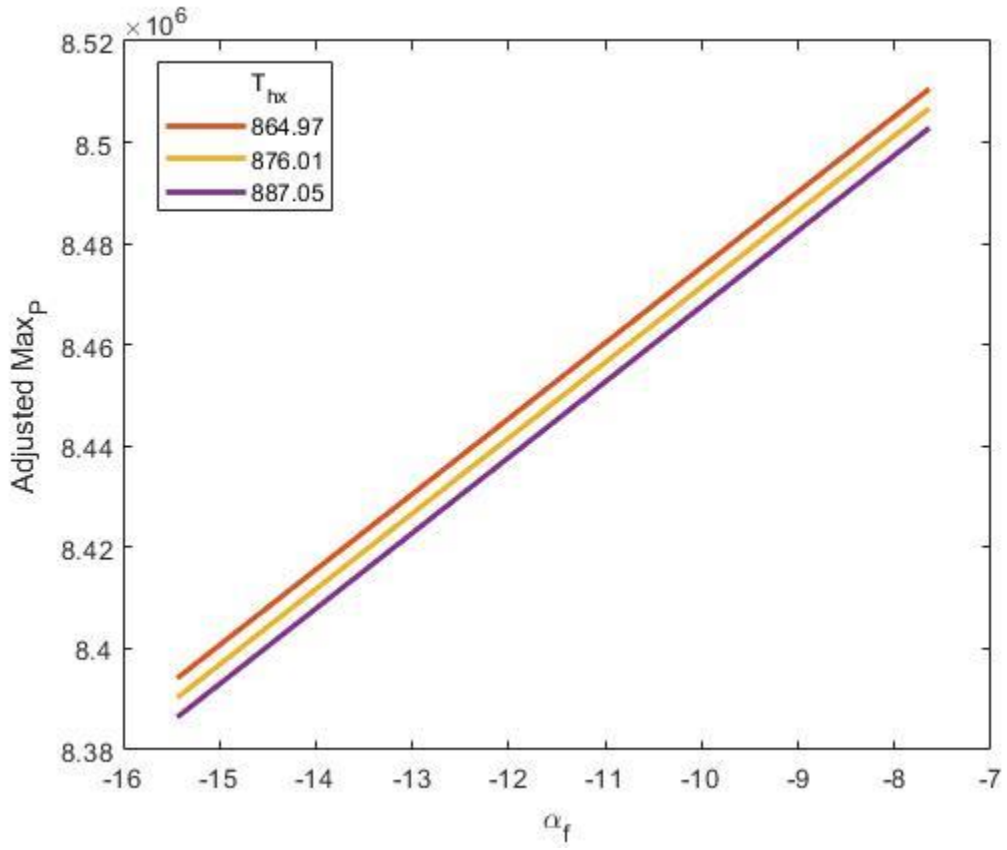


Figure 50: Interaction of the temperature with the fuel feedback coefficient.

Figure 50 shows the interaction between the temperature and α_f, where the correlation between these two variables wasn't high enough to be included in **Figure 49**, which means that correlation quantifies the relationship between two predictors, while interaction quantifies the deviation from the expected relationship between two predictors when they are considered jointly in the regression model.

Conclusions

This chapter conducted uncertainty analysis for the previous two chapters To ensure the accuracy of their results, they used the OFAT method for the turbulence modeling study and the LHS method for the Multiphysics study. The use of different methods was for different studies aimed at ensuring the validity of the resultscfor each study.

The uncertainty quantification in Chapter II improved the confidence in the mesh selection as the GCI values for velocity and Nusselt number were small. It also confirmed that the input uncertainty for the friction factor and Nusselt number remained within reasonable limits, with a maximum deviation of 8.6% for the entry length and 2.6% for the friction factor and Nusselt number.

The purpose of the uncertainty analysis in Chapter III was to determine the extent of error and to better understand the differences between the model created in GeN-Foam and the experiment. The results indicated that the uncertainty in the input values was greater than the numerical uncertainty, suggesting that the mesh used in the model was reliable. The analysis also highlighted differences in the thermophysical properties cited in the literature, such as viscosity and feedback coefficients.

The uncertainty analysis conducted in this section increases confidence in the model and demonstrates that the agreement between the experiment and the model falls within the range of total uncertainty.

Although the uncertainty error covered the range of discrepancy between the simulation and the MSRE for the transient. The uncertainty range did not include the

discrepancy in the experimental power data after 55 seconds, as it did not include the secondary loop of the reactor in the model.

Having data generated by LHS that follows a normal distribution with a strong 95% confidence interval is desirable as it makes statistical analysis easier to interpret and the results are likely to be more robust and accurate.

CHAPTER V

CONCLUSIONS

This study has made a valuable contribution to the field of thermal molten salt reactors by examining the flow of molten salt through a heated pipe using RANS and analyzing how different turbulence models impact thermal hydraulic metrics. The goal was to simulate a case similar to the piping in MSR primary loops, which contain heaters to prevent the molten salt from solidifying. This was an important preliminary step to a more extensive study that used a 1D model to validate the point kinetics solver of Multiphysics tool (GeN-Foam), enabling efficient analysis of thermal MSR transients. This validation opens the door for further research on a range of transient scenarios. The validated 1D model was then used to conduct a comprehensive sensitivity study that quantified the effects of input and numerical uncertainties on both the neutronics and thermal hydraulics designs of the reactor core and loop. These findings provide a greater understanding of molten salt flow and MSR transients, offering valuable insights that can be applied to improving the design and operation of MSR.

The study consisted of three phases. Firstly, the behavior of molten salt in a circular heated pipe was investigated to simulate the flow of molten salt in molten salt reactors, where heaters surround the pipes to prevent solidification. The results of this study using CFD confirmed that different turbulence models provide relatively similar results and match the Dittus-Boelter and Sieder-Tate correlations for the fully developed region of the pipe.

The second phase involved using an open-source CFD tool to analyze the behavior of molten salt fuel in a reactor, taking into account the Multiphysics nature of the system, including the coupling of neutronics and thermal hydraulics. A 1D simple model was developed to investigate a transient in an MSRE reactor, and the results showed a good match with the experiment, with a maximum discrepancy of 4%.

Finally, a comprehensive uncertainty analysis was performed in the third phase, utilizing both OFAT for turbulence modeling and LHS for the GeN-Foam 1D model. The analysis indicated that the input uncertainty was greater than the numerical uncertainty for both studies, and the agreement between the experiment and the model falls within the total uncertainty range for the GeN-Foam model.

Future Work

The results of the study on the behavior of molten salt in a circular heated pipe showed that the turbulence models investigated agreed well with the Dittus-Boelter and Sieder-Tate correlations in the fully developed region. However, in the developing region, the predictions of Nu^* showed significant variations among the models, presenting a challenge in selecting an appropriate turbulence model for more complex reactor flow configurations. The simplicity of the GeN-Foam model used in this study was a contributing factor to the results, and it is important to note that the results might differ in 2D or 3D models and with the presence of heterogeneity. Hence, further investigation of other turbulence models in GeN-Foam is necessary.

When considering more complex reactor geometries and incorporating the added complexities of neutronics and the changing thermophysical properties of the salt due to the presence of nuclear fuel, it is essential to exercise caution and carefully consider the limitations of the turbulence models used.

The limitations of GeN-Foam model have been noted, including the need for modifications to accurately predict the fraction of power produced in graphite, which will mainly affect the value of the initial peak. In addition, the inability to simulate multiple reactivity transients simultaneously, prevents the researchers from investigating more complex transients for MSRs.

The limitations of the 1D model developed were identified as being unsuitable for validating natural circulation experiments in MSREs, due to its vertical orientation that only accounts for downward gravity effects on all components. It is suggested to create a more comprehensive 2D model for the primary loop, taking into account the selection of turbulence models for improved accuracy.

It is recommended to continue with the study and enhance the model by incorporating the secondary loop to increase the accuracy of the results. For instance, using a 2D model, the heat exchanger connecting the primary and secondary loops could be modeled in more detail, instead of using an average temperature as in this study, to produce results that are closer to the experimental findings.

The fact that GeN-Foam only offers the porous k-epsilon model for simulating fluid flow through porous media provides an opportunity to investigate and compare the performance of different turbulence models in this context. By modifying the existing

solvers or developing new ones, it is possible to explore the behavior of alternative turbulence models and evaluate their accuracy in simulating fluid flow through porous media, which can help to advance our understanding of these complex systems.

REFERENCES

[1] M. W. Rosenthal, P. R. Kasten, and R. B. Briggs, “Molten-salt reactors—history, status, and potential,” *Nuclear Applications and Technology*, vol. 8, no. 2, pp. 107–117, 1970.

[2] J. Serp *et al.*, “The molten salt reactor (MSR) in generation IV: overview and perspectives,” *Progress in Nuclear Energy*, vol. 77, pp. 308–319, 2014.

[3] R. C. Robertson, “MSRE Design and Operation Report, Part 1, Description of Reactor Design, Oak Ridge National Laboratory, ORNL-TM-0728.”

[4] A. van Wijk and D. Lathouwers, “Computational Modeling of the Flow Field in a Molten Salt Reactor Core,” 2008.

[5] Rosenthal MW, KastenPR, and Briggs RB, “Molten Salt Reactors. History, Status, and Potential,” *Nucl Appl Technol*, vol. 8, no. 2, pp. 107–117, 1970, doi: 10.13182/nt70-a28619.

[6] “Advances in Small Modular Reactor Technology Developments,” 2016.

[7] H. Rouch *et al.*, “Preliminary thermal-hydraulic core design of the Molten Salt Fast Reactor (MSFR),” *Ann Nucl Energy*, vol. 64, pp. 449–456, Feb. 2014, doi: 10.1016/j.anucene.2013.09.012.

[8] R. Freile and M. Kimber, “Influence of molten salt-(FLiNaK) thermophysical properties on a heated tube using CFD RANS turbulence modeling of an experimental testbed.,” 2019.

[9] Paul N. Haubenreich & J. R. Engel, “Experience with the Molten- Salt Reactor Experiment, Nuclear Applications and Technology,” *Nucl Appl Technol*, vol. 8, no. 2, pp. 118–136, 1970, doi: 10.13182/nt8-2-118.

[10] M. D. Grele and L. G. Gedeon, “Forced-Convection Heat-Transfer Charachtaristichs of Molten FLINAK Flowing in an lnconel X System, National Advisory Committee for Aeronautics,” 1954.

[11] H. W. Hoffman and J. Lones, “Fused salt heat transfer Part II: Forced ConvectionHeat Transfer in Circular Tubes Containing NaF-KF-LiF Eutectic,” 1955. doi: 10.1017/CBO9781107415324.004.

[12] B. Vriesma, “Aspects of Molten Fluorides As Heat Transfer Agents for Power Generation, Nucl Technol, vol. 165, no. 2, pp. 166–173,” 1979. doi: 10.13182/NT165-166.

[13] J. Ambrosek, M. Anderson, K. Sridharan, and T. Allen, “Current status of knowledge of the fluoride salt (FLiNaK) heat transfer,” *Nucl Technol*, vol. 165, no. 2, pp. 166–173, 2009, doi: 10.13182/NT165-166.

[14] M. v Smirnov, V. A. Khokhlov, and E. S. Filatov, “Thermal Conductivity of Molten Alkali Halides and Their Mixtures,” 1987.

[15] Y. M. Ferng, K. Y. Lin, and C. W. Chi, “CFD investigating thermal-hydraulic characteristics of FLiNaK salt as a heat exchange fluid,” *Appl Therm Eng*, vol. 37, pp. 235–240, 2012, doi: 10.1016/j.applthermaleng.2011.11.021.

[16] A. K. Srivastava, A. M. Vaidya, N. K. Maheshwari, and P. K. Vijayan, "Heat transfer and pressure drop characteristics of molten fluoride salt in circular pipe," *Appl Therm Eng*, vol. 61, no. 2, pp. 198–205, 2013, doi: 10.1016/j.applthermaleng.2013.07.051.

[17] S. Zhang, X. Sun, and E. E. Dominguez-Ontiveros, "Numerical study on convective heat transfer and friction characteristics of molten salts in circular tubes," *Ann Nucl Energy*, vol. 142, p. 107375, 2020, doi: 10.1016/j.anucene.2020.107375.

[18] S. Zhang, X. Sun, and E. E. Dominguez-Ontiveros, "Numerical study on convective heat transfer and friction characteristics of molten salts in circular tubes," *Ann Nucl Energy*, vol. 142, Jul. 2020, doi: 10.1016/j.anucene.2020.107375.

[19] Y. M. Ferng, K. Y. Lin, and C. W. Chi, "CFD investigating thermal-hydraulic characteristics of FLiNaK salt as a heat exchange fluid," *Appl Therm Eng*, vol. 37, pp. 235–240, May 2012, doi: 10.1016/j.applthermaleng.2011.11.021.

[20] M. Nallasamy, "Turbulence models and their applications to the prediction of internal flows: A review," *Comput Fluids*, vol. 15, no. 2, pp. 151–194, 1987, doi: 10.1016/S0045-7930(87)80003-8.

[21] S.-C. CD-adapco, "11.0 User Guide, CD-adapco." Inc, 2016.

[22] P. R. Spalart and S. R. Allmaras, “One-equation turbulence model for aerodynamic flows,” *Recherche aerospatiale*, no. 1, pp. 5–21, 1994, doi: 10.2514/6.1992-439.

[23] T.-H. Shih, W. W. Liou, A. Shabbir, Z. Yang, and J. Zhu, “A new k-epsilon eddy viscosity model for high Reynolds number turbulent flows: Model development and validation,” 1994.

[24] A. D. Fradeneck, K. V. Kirkland, D. Banerjee, J. Alvarado, and A. A. Polycarpou, “Towards the Improvement of Common RANS Models in the Prediction of Buoyancy Influenced Heated Gas Flows: A Dissertation.,” 2020.

[25] V. Shukla and A. K. Kaviti, “Performance evaluation of profile modifications on straight-bladed vertical axis wind turbine by energy and Spalart Allmaras models,” *Energy*, vol. 126, pp. 766–795, 2017.

[26] F.-F. Ning and L.-P. Xu, “Application of one-equation spalart-allmaras turbulence model in the numerical simulation of internal flows,” *Journal of Engineering Thermophysics*, vol. 22, no. 3, pp. 304–306, 2001.

[27] B. A. Sen, U. G. Yuksel, and K. Kirkkopru, “Comparison of turbulence models for an internal flow with side wall mass injection,” ISTANBUL UNIV (TURKEY) DEPT OF MECHANICAL ENGINEERING, 2004.

[28] C. E. Clifford, A. Hassan, M. King, and K. Vierow-Kirkland, “Experimentally validated computational fluid dynamics of buoyancy-influenced flows: A dissertation.,” 2020.

[29] D. C. Wilcox, “Reassessment of the scale-determining equation for advanced turbulence models,” *AIAA journal*, vol. 26, no. 11, pp. 1299–1310, 1988.

[30] F. R. Menter, “Zonal two equation κ-ω turbulence models for aerodynamic flows,” in *AIAA 23rd Fluid Dynamics, Plasmadynamics, and Lasers Conference, 1993*, 1993. doi: 10.2514/6.1993-2906.

[31] F. R. . Menter, “Zonal Two Equation k-w , Turbulence Models for Aerodynamic Flows .,” *24th Fluid Dynamics Conference*, 1993, doi: 10.2514/6.1993-2906.

[32] P. A. Durbin, “Separated flow computations with the k-epsilon-v-squared model,” *AIAA journal*, vol. 33, no. 4, pp. 659–664, 1995.

[33] M. S. Sohal, M. A. Ebner, P. Sabharwall, and P. Sharpe, “Engineering Database of Liquid Salt Thermophysical and Thermochemical Properties,” 2010. [Online]. Available: http://www.inl.gov

[34] R. Freile and M. Kimber, “Influence of molten salt-(FLiNaK) thermophysical properties on a heated tube using CFD RANS turbulence modeling of an experimental testbed,” *EPJ Nuclear Sciences & Technologies*, vol. 5, p. 16, 2019, doi: 10.1051/epjn/2019027.

[35] A. Klein, “Turbulent Developing Pipe Flow,” vol. 103, no. June, pp. 243–249, 1981.

[36] M. D. Grele and L. Gedeon, “Forced-Convection Heat-Transfer Charachtaristichs of Molten FLINAK Flowing in an lnconel X System,” *National Advisory Committee for Aeronautics*, no. February 1954, 1958.

[37] S. v Patankar and D. B. Spalding, “A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows,” in *Numerical prediction of flow, heat transfer, turbulence and combustion*, Elsevier, 1983, pp. 54–73.

[38] M. S. Sohal, M. a Ebner, P. Sabharwall, and P. Sharpe, “Engineering database of liquid salt thermophysical and thermochemical properties,” *Idaho National Laboratory, Idaho Falls CrossRef*, no. March, pp. 1–70, 2010, doi: ext-10-18297.

[39] I. B. Vriesema, “Aspects of Molten Fluorides As Heat Transfer Agents for Power Generation.,” *Tech Hogesch Delft Afd Werktuigbouwkd (Rep) WTHD*, no. 112, 1979.

[40] M. V. Smirnov, V. A. Khokhlov, and E. S. Filatov, “Thermal conductivity of molten alkali halides and their mixtures,” *Electrochim Acta*, vol. 32, no. 7, pp. 1019–1026, 1987, doi: 10.1016/0013-4686(87)90027-2.

[41] J. Ambrosek, M. Anderson, K. Sridharan, and T. Allen, “Current status of knowledge of the fluoride salt (FLiNaK) heat transfer,” *Nucl Technol*, vol. 165, no. 2, pp. 166–173, 2009, doi: 10.13182/NT165-166.

[42] D. B. Bryant, E. M. Sparrow, and J. M. Gorman, “Turbulent pipe flow in the presence of centerline velocity overshoot and wall-shear undershoot,”

International Journal of Thermal Sciences, vol. 125, pp. 218–230, Mar. 2018, doi: 10.1016/j.ijthermalsci.2017.11.028.

[43] "Evaluation of the K-Epsilon-VV-F Turbulence Model for Natural Convection in a Rectangular Cavity," *Seok-Ki Choi, Seong-O Kim, Eui-Kwang Kim, Hoon-Ki Choi*, 2012.

[44] S. M. Ghiaasiaan, *Convective heat and mass transfer*, vol. 9781107003. 2011. doi: 10.1017/CBO9780511800603.

[45] C. A. Sleicher and M. W. Rouse, "A convenient correlation for heat transfer to constant and variable property fluids in turbulent pipe flow," *Int J Heat Mass Transf*, vol. 18, no. 5, pp. 677–683, 1975, doi: 10.1016/0017-9310(75)90279-3.

[46] E. N. Sieder and G. E. Tate, "Heat transfer and pressure drop of liquids in tubes," *Ind Eng Chem*, vol. 28, no. 12, pp. 1429–1435, 1936.

[47] F. W. Dittus, "Heat transfer in automobile radiators of the tubular type," *Univ. of California Pub., Eng.*, vol. 2, no. 13, pp. 443–461, 1930.

[48] "Generation IV International Forum. A technology roadmap for Generation IV nuclear energy systems. GIF-002-00, US DOE Nuclear Energy Research Advisory Committee and the Generation IV International Forum, 2002".

[49] S. Lorenzi, C. Fiorina, and A. Cammi, "Derivation and implementation of a point kinetics model for Molten Salt Reactors in GeN-Foam Supervisor Co-Supervisors."

[50] N. R. Brown, D. J. Diamond, S. Bajorek, and R. Denning, “Thermal-Hydraulic and Neutronic Phenomena Important in Modeling and Simulation of Liquid-Fuel Molten Salt Reactors,” *Nucl Technol*, vol. 206, no. 2, pp. 322–338, Feb. 2020, doi: 10.1080/00295450.2019.1590077.

[51] P. R. Rubiolo, M. T. Retamales, V. Ghetta, and J. Giraud, “High temperature thermal hydraulics modeling of a molten salt: application to a molten salt fast reactor (MSFR),” *ESAIM Proc Surv*, vol. 58, pp. 98–117, 2017, doi: 10.1051/proc/201758098.

[52] J. Krepel, U. Grundmann, U. Rohde, and F. P. Weiss, “DYN1D-MSR dynamics code for molten salt reactors,” *Ann Nucl Energy*, vol. 32, no. 17, pp. 1799–1824, Nov. 2005, doi: 10.1016/j.anucene.2005.07.007.

[53] J. Křepel, U. Rohde, U. Grundmann, and F. P. Weiss, “DYN3D-MSR spatial dynamics code for molten salt reactors,” *Ann Nucl Energy*, vol. 34, no. 6, pp. 449–462, Jun. 2007, doi: 10.1016/j.anucene.2006.12.011.

[54] A. Cammi, C. Fiorina, C. Guerrieri, and L. Luzzi, “Dimensional effects in the modelling of MSR dynamics: Moving on from simplified schemes of analysis to a multi-physics modelling approach,” in *Nuclear Engineering and Design*, May 2012, vol. 246, pp. 12–26. doi: 10.1016/j.nucengdes.2011.08.002.

[55] M. Zanetti, A. Cammi, C. Fiorina, and L. Luzzi, “A Geometric Multiscale modelling approach to the analysis of MSR plant dynamics,” *Progress in*

Nuclear Energy, vol. 83, pp. 82–98, Aug. 2015, doi: 10.1016/j.pnucene.2015.02.014.

[56] J. Bao, C. Fiorina, and K. Mikityuk, “Development of the model for the multi-physics analysis of Molten Salt Reactor Experiment using GeN-Foam code MASTER THESIS,” 2016.

[57] J. Kópházi, D. Lathouwers, and J. L. Kloosterman, “Development of a three-dimensional time-dependent calculation scheme for molten salt reactors and validation of the measurement data of the molten salt reactor experiment,” *Nuclear Science and Engineering*, vol. 163, no. 2, pp. 118–131, 2009, doi: 10.13182/NSE163-118.

[58] C. Shi, M. Cheng, and G. Liu, “Development and application of a system analysis code for liquid fueled molten salt reactors based on RELAP5 code,” *Nuclear Engineering and Design*, vol. 305, pp. 378–388, Aug. 2016, doi: 10.1016/j.nucengdes.2016.05.034.

[59] D. de Wet and M. S. Greenwood, “Status Report on the MSRE TRANSFORM Model for Thermal-Hydraulic Benchmarking,” 2019. [Online]. Available: www.osti.gov

[60] T. Fei, T. Hua, B. Feng, F. Heidet, and R. Hu, “MSRE Transient benchmarks using SAM,” in *International Conference on Physics of Reactors: Transition to a Scalable Nuclear Future, PHYSOR 2020*, 2020, vol. 2020-March, pp. 1464–1471. doi: 10.1051/epjconf/202124707008.

[61] H. Mochizuki, “Validation of neutronics and thermal-hydraulics coupling model of the RELAP5-3D code using the MSRE reactivity insertion tests,” *Nuclear Engineering and Design*, vol. 389, Apr. 2022, doi: 10.1016/j.nucengdes.2022.111669.

[62] D. Lecarpentier and V. Carpentier, “A neutronic program for critical and nonequilibrium study of mobile fuel reactors: The Cinsf1D code,” *Nuclear Science and Engineering*, vol. 143, no. 1, pp. 33–46, 2003, doi: 10.13182/NSE03-A2316.

[63] T. Yamamoto, K. Mitachi, K. Ikeuchi, and T. Suzuki, “Transient characteristics of small molten salt reactor during blockage accident,” *Heat Transfer - Asian Research*, vol. 35, no. 6, pp. 434–450, Sep. 2006, doi: 10.1002/htj.20123.

[64] C. Nicolino, G. Lapenta, S. Dulla, and P. Ravetto, “Coupled dynamics in the physics of molten salt reactors,” *Ann Nucl Energy*, vol. 35, no. 2, pp. 314–322, Feb. 2008, doi: 10.1016/j.anucene.2007.06.015.

[65] D. L. Zhang, S. Z. Qiu, G. H. Su, C. L. Liu, and L. B. Qian, “Analysis on the neutron kinetics for a molten salt reactor,” *Progress in Nuclear Energy*, vol. 51, no. 4–5, pp. 624–636, May 2009, doi: 10.1016/j.pnucene.2008.11.008.

[66] D. Zhang, S. Qiu, and G. Su, “Development of a safety analysis code for molten salt reactors,” *Nuclear Engineering and Design*, vol. 239, no. 12, pp. 2778–2785, Dec. 2009, doi: 10.1016/j.nucengdes.2009.08.020.

[67] D. L. Zhang, S. Z. Qiu, G. H. Su, and C. L. Liu, “Development of a steady state analysis code for a molten salt reactor,” *Ann Nucl Energy*, vol. 36, no. 5, pp. 590–603, May 2009, doi: 10.1016/j.anucene.2009.01.004.

[68] A. S. Mattioli, “Derivation and implementation of a point kinetics model for molten salt reactors in GeN-foam,” 2021.

[69] S. E. Beall, P. N. Haubenreich, R. B. Lindauer, and J. R. Tallackson, “Msre design and operations report. part v. reactor safety analysis report,” Oak Ridge National Lab., Tenn., 1964.

[70] P. N. Haubenreich, J. R. Engel, B. E. Prince, and H. C. Claiborne, “MSRE design and operations report. Part III. Nuclear analysis,” Oak Ridge National Lab., Tenn., 1964.

[71] C. H. Gabbard, “Reactor Power Measurement and Heat Transfer Performance in the Molten Salt Reactor Experiment,” Oak Ridge National Lab., Tenn., 1970.

[72] C. Fiorina, I. Clifford, M. Aufiero, and K. Mikityuk, “GeN-Foam: A novel OpenFOAM® based multi-physics solver for 2D/3D transient analysis of nuclear reactors,” *Nuclear Engineering and Design*, vol. 294, pp. 24–37, Dec. 2015, doi: 10.1016/j.nucengdes.2015.05.035.

[73] M. Fratoni, D. Shen, G. Ilas, and J. Powers, “Molten Salt Reactor Experiment Benchmark Evaluation (Project 16-10240).”

[74] R. H. Guymon, “MSRE systems and components performance,” ed. and comp.; Oak Ridge National Lab., Tenn.(USA), 1973.

[75] R. E. Thoma, “Chemical Aspects of MSRE Operations,” Oak Ridge National Lab., Tenn., 1971.

[76] R. J. Kedl and C. K. McGlothlan, “Tube Vibration in MSRE Primary Heat Exchanger,” Oak Ridge National Lab., Tenn., 1968.

[77] S. M. Ghiaasiaan, *Convective heat and mass transfer*. Cambridge University Press, 2011.

[78] Steffy Jr. and R.C., “Experimental dynamic analysis of the MSRE with 233U fuel, ORNL-TM-2997. Oak Ridge National Laboratory. https://www.osti.gov/servlets/pur l/4132458.,” 1970.

[79] H. K. Versteeg and W. Malalasekera, *An introduction to computational fluid dynamics: the finite volume method*. Pearson education, 2007.

[80] A. S. of M. Engineers, *Standard for Verification and Validation in Computational Fluid Dynamics and Heat Transfer: An American National Standard*. American Society of Mechanical Engineers, 2009.

[81] F. Pianosi *et al.*, “Sensitivity analysis of environmental models: A systematic review with practical workflow,” *Environmental Modelling & Software*, vol. 79, pp. 214–232, 2016.

[82] A. G. Weiss *et al.*, “Characterization of the Finite Element Computational Fluid Dynamics Capabilities in the Multiphysics Object Oriented Simulation Environment,” *Journal of Nuclear Engineering and Radiation Science*, vol. 9, no. 2, p. 021402, 2022.

[83] M. D. McKay, R. J. Beckman, and W. J. Conover, “A comparison of three methods for selecting values of input variables in the analysis of output from a computer code,” *Technometrics*, vol. 42, no. 1, pp. 55–61, 2000.

[84] R. C. Steffy and P. J. Wood, “Theoretical dynamic analysis of the MSRE with 233U fuel,” *ORNL-TM-2571*, 1969.

APPENDIX A

GEN-FOAM INPUT FILES

The appendix contains a comprehensive list of input files utilized in building the 1D model using GeN-FOAM. **Figure 22** provides an overview of the input files and their corresponding folders.

Appendix A.1: 0 Folder

fluidRegion

U

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.2.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volVectorField;
    location    "0/fluid";
    object      U;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [ 0 1 -1 0 0 0 0 ];

internalField   uniform (0 0 0);

boundaryField
{
    bottom
    {
        type            cyclic;
    }

    top
    {
    type            cyclic;
    }

    walls
    {
        type            slip;
    }



}
```

T

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.2.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    location    "0/fluid";
    object      T;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 0 0 1 0 0 0];

internalField   uniform 900;

boundaryField
{
    bottom
    {
        type            cyclic;
    }

    top
    {
        type            cyclic;
    }

    walls
    {
        type            zeroGradient;
    }

}


// ************************************************************************* //
```

P

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.3.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    location    "0/fluid";
    object      p;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [1 -1 -2 0 0 0 0];

internalField   uniform 100000;

boundaryField
{
    bottom
    {
        type            cyclic;
    }

    top
    {
        type            cyclic;
    }

    walls
    {
        type            calculated;
        value           $internalField;
    }

}


// ************************************************************************* //
```

alphat

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  4.x                                   |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    location    "0/fluid";
    object      alphat;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [1 -1 -1 0 0 0 0];

internalField   uniform 4.297e-04;

boundaryField
{
    bottom
    {
        type            cyclic;
    }

    top
    {
        type            cyclic;
    }

    walls
    {
        type            zeroGradient;//alphatJayatillekeWallFunction;
        value           $internalField;
        Prt             0.075;

    }


}


// ************************************************************************* //
```

nut

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  4.x                                   |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    location    "0";
    object      nut;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 2 -1 0 0 0 0];

internalField   uniform 5.714e-05;

boundaryField
{
    bottom
    {
        type            cyclic;
    }

    top
    {
        type            cyclic;
    }

    walls
    {
        type            nutkWallFunction;
        value           $internalField;

    }

}


// ************************************************************************* //
```

k

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  4.x                                   |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    location    "0/fluid";
    object      k;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 2 -2 0 0 0 0];

internalField   uniform 5e-03;

boundaryField
{
    bottom
    {
        type            cyclic;
    }

    top
    {
        type            cyclic;
    }

    walls
    {
        type            kqRWallFunction;
        value           $internalField;

    }

}


// ************************************************************************* //
```

epsilon

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  4.x                                   |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    location    "0/fluid";
    object      epsilon;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 2 -3 0 0 0 0];

internalField   uniform 0.1175;

boundaryField
{
    bottom
    {
        type            cyclic;
    }

    top
    {
        type            cyclic;
    }

    walls
    {
       type            epsilonWallFunction;
        value           $internalField;
    }

}


// ************************************************************************* //
```

neutroRegion

defaultFlux

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.2.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    location    "0";
    object      defaultFlux;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [ 0 -2 -1 0 0 0 0 ];

internalField   uniform 1;

boundaryField
{
    bottom
    {
        type            cyclic;
    }

    top
    {
        type            cyclic;
    }

    walls
    {
        type            zeroGradient;
    }

    front
    {
        type           zeroGradient;
    }
    back
    {
        type           zeroGradient;
    }

}

// ************************************************************************* //
```

Appendix A.2: constant Folder

fluidRegion

phaseProperties

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  6
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      phaseProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

// ------------------------------------------------------------------------- //
// --- THERMAL-HYDRAULIC TYPE ---------------------------------------------- //
// ------------------------------------------------------------------------- //

thermalHydraulicsType    "onePhase";

fluidProperties
{

}

// ------------------------------------------------------------------------- //
// --- STRUCTURES PROPERTIES ----------------------------------------------- //
// ------------------------------------------------------------------------- //

structureProperties
{
    "core"
    {
        volumeFraction      0.775;
        Dh                  0.0123;

        /*
        powerModel
        {
            type            fixedPower;
            Cp              2386.5;
            rho             2245;
            T               916;
        }
        */
        passiveProperties
        {
            volumetricArea  124; //
            rho             1874;
            Cp              1772;
            T               916; // initial temp. better if the same as for fluid
        }

    }
```

```
    }

    "hx"
    {
        volumeFraction      0.14;
        Dh                  0.0209; //https://mooseframework.inl.gov/virtual_test_bed/msr/msre/msre_sam_model.html

        powerModel
        {
            type            constantTemperature;
            volumetricArea  180;
            T               876;
        }
    }

    "pump"
    {
        volumeFraction      0;
        Dh                  0.127;
        momentumSource      (0 0 8720000);




    }
}

// ************************************************************************* //
// --- REGIME MAP MODELS ------------------------------------------------- //
// ************************************************************************* //

regimeMapModels
{
}

// ************************************************************************* //
// --- REGIME PHYSICS FOR EACH REGIME ------------------------------------- //
// ************************************************************************* //

physicsModels
{
    dragModels
    {
       "core"
        {
           type     ReynoldsPower; // check this is Blasius
            coeff   0.316;
            exp     -0.25;
        }

       "hx"
        {
           type     ReynoldsPower; // check this is Blasius
            coeff   0.316; // coeff*Re^exp
            exp     -0.25;


    }

    heatTransferModels
    {

        "core"
        {
            type    NusseltReynoldsPrandtlPower; // const + coeff*Re^expRe*Pr^expPr
            const   0;
            coeff   0.023;
            expRe   0.8;
            expPr   0.4;
        }
        "hx"
        {
            type    constant;
            value   6.1539e+3; // W/m2K
        }

    }
}

// ************************************************************************* //
// --- MISCELLANEA ------------------------------------------------------- //
// ************************************************************************* //
pMin                    0;
pRefCell                0;
pRefValue               5e5;


// ************************************************************************* //
```

thermophysicalProperties

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  6
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      thermophysicalProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

thermoType
{
    type            heRhoThermo;
    mixture         pureMixture;
    transport       const;
    thermo          hConst;
    equationOfState Boussinesq;
    specie          specie;
    energy          sensibleEnthalpy;
}

mixture
{
    specie
    {
        nMoles          1;
        molWeight       554.179;
    }
    equationOfState
    {
       rho0             2245;
       T0               916;
       beta             2e-4;
    }
    thermodynamics
    {
        Cp              2386.5;
        Hf              0;
        Sf              0;
    }
    transport
    {
        mu              0.007;
        Pr              16.66;
    }
}



// ************************************************************************* //
```

turbulenceProperties

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.2.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      turbulenceProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

simulationType RAS;     // laminar;//

RAS
{
    RASModel        porousKEpsilon;//laminar;//kEpsilon;//;

    turbulence      true;

    printCoeffs     true;
}

porousKEpsilonProperties
{
        "core"
        {
            convergenceLength           0.5;
        turbulenceIntensityCoeff    0.16;
        turbulenceIntensityExp      -0.125;
        turbulenceLengthScaleCoeff  0.07;
        }
        "hx"
        {
            convergenceLength           0.5;
        turbulenceIntensityCoeff    0.16;
        turbulenceIntensityExp      -0.125;
        turbulenceLengthScaleCoeff  0.07;
        }

}


// ************************************************************************* //
```

neutroRegion

neutronicsProperties

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.2.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      neutronicsProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

model           pointKinetics;

eigenvalueNeutronics    false;

// ************************************************************************* //
```

reactorState

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2106                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    arch        "LSB;label=32;scalar=64";
    class       dictionary;
    location    "constant/neutroRegion";
    object      reactorState;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

keff            0.1;

pTarget         8e+06;

externalReactivity 0.00013;


// ************************************************************************* //
```

nuclearData

```
/*
 crossSection dictionary
 Generatd by serpentToFoamXS
 04-Aug-2014
 From SERPENT results file: esfr_core__res
*/

/*
physical delayed neutron fraction and spectrum
*/

FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    constant;
    object      msfr_n_res;
}

fastNeutrons            true; // if true interpolates cross section with logarithmic temp for fuel (otherwise, square root)

adjustDiscFactors       false;

useGivenDiscFactors     false;

energyGroups 1 ;

precGroups 6 ;

zones
(
    core
    {
        fuelFraction 0.225;
    IV nonuniform List<scalar> 1 (3.887390e-07  );
    D nonuniform List<scalar> 1 (1.4e-02  );
    nuSigmaEff nonuniform List<scalar> 1 (7.106120e-01  );
    sigmaPow nonuniform List<scalar> 1 (2.10989E-11  );
    scatteringMatrixP0  1  1 (
    ( 3.071420e+01 )
    );
    sigmaDisapp nonuniform List<scalar> 1 (3.262000e-01  );
    chiPrompt nonuniform List<scalar> 1 (0.01000e+00  );
    chiDelayed nonuniform List<scalar> 1 (0.240000e+00  );
        lambda nonuniform List<scalar> 6
        (0.0126 0.0337 0.139 0.325 1.13 2.5);
        Beta nonuniform List<scalar> 6
        (22.8e-05 78.8e-05 66.4e-05 73.6e-05 13.6e-05 8.8e-05);
    discFactor nonuniform List<scalar> 1 (1.0  );
    integralFlux nonuniform List<scalar> 1 (1.0  );

    }
```

```
}

HotLeg
{
fuelFraction 1.000000e+00 ;
IV nonuniform List<scalar> 1 (8.766510e-07  );
D nonuniform List<scalar> 1 (4.333110e-02  );
nuSigmaEff nonuniform List<scalar> 1 (0.000000e+00  );
sigmaPow nonuniform List<scalar> 1 (0.000000e+00  );
scatteringMatrixP0  1  1 (
( 1.141000e+01 )
);
sigmaDisapp nonuniform List<scalar> 1 (0.01  );
chiPrompt nonuniform List<scalar> 1 (0.000000e+00  );
chiDelayed nonuniform List<scalar> 1 (0.000000e+00  );
    lambda nonuniform List<scalar> 6
    (0.0126 0.0337 0.139 0.325 1.13 2.5);
    Beta nonuniform List<scalar> 6
    (22.8e-05 78.8e-05 66.4e-05 73.6e-05 13.6e-05 8.8e-05);
discFactor nonuniform List<scalar> 1 (1.0  );
integralFlux nonuniform List<scalar> 1 (1.0  );

}

pump
{
fuelFraction 1.000000e+00 ;
IV nonuniform List<scalar> 1 (8.766510e-07  );
D nonuniform List<scalar> 1 (4.333110e-02  );
nuSigmaEff nonuniform List<scalar> 1 (0.000000e+00  );
sigmaPow nonuniform List<scalar> 1 (0.000000e+00  );
scatteringMatrixP0  1  1 (
( 1.141000e+01 )
);
sigmaDisapp nonuniform List<scalar> 1 (0.01 );
chiPrompt nonuniform List<scalar> 1 (0.000000e+00  );
chiDelayed nonuniform List<scalar> 1 (0.000000e+00  );
    lambda nonuniform List<scalar> 6
    (0.0126 0.0337 0.139 0.325 1.13 2.5);
    Beta nonuniform List<scalar> 6
    (22.8e-05 78.8e-05 66.4e-05 73.6e-05 13.6e-05 8.8e-05);
discFactor nonuniform List<scalar> 1 (1.0  );
integralFlux nonuniform List<scalar> 1 (1.0  );

}

hx
{
fuelFraction 1.000000e+00 ;
IV nonuniform List<scalar> 1 (8.766510e-07  );
D nonuniform List<scalar> 1 (4.333110e-02  );
nuSigmaEff nonuniform List<scalar> 1 (0.000000e+00  );
sigmaPow nonuniform List<scalar> 1 (0.000000e+00  );
scatteringMatrixP0  1  1 (
( 1.141000e+01 )
```

```
    );
    sigmaDisapp nonuniform List<scalar> 1 (0.01  );
    chiPrompt nonuniform List<scalar> 1 (0.000000e+00  );
    chiDelayed nonuniform List<scalar> 1 (0.000000e+00  );
        lambda nonuniform List<scalar> 6
        (0.0126 0.0337 0.139 0.325 1.13 2.5);
        Beta nonuniform List<scalar> 6
        (22.8e-05 78.8e-05 66.4e-05 73.6e-05 13.6e-05 8.8e-05);
    discFactor nonuniform List<scalar> 1 (1.0  );
    integralFlux nonuniform List<scalar> 1 (1.0  );

    }

    ColdLeg
    {
    fuelFraction 1.000000e+00 ;
    IV nonuniform List<scalar> 1 (8.766510e-07  );
    D nonuniform List<scalar> 1 (4.333110e-02  );
    nuSigmaEff nonuniform List<scalar> 1 (0.000000e+00  );
    sigmaPow nonuniform List<scalar> 1 (0.000000e+00  );
    scatteringMatrixP0  1  1 (
    ( 1.141000e+01 )
    );
    sigmaDisapp nonuniform List<scalar> 1 (0.01  );
    chiPrompt nonuniform List<scalar> 1 (0.000000e+00  );
    chiDelayed nonuniform List<scalar> 1 (0.000000e+00  );
        lambda nonuniform List<scalar> 6
        (0.0126 0.0337 0.139 0.325 1.13 2.5);
        Beta nonuniform List<scalar> 6
        (22.8e-05 78.8e-05 66.4e-05 73.6e-05 13.6e-05 8.8e-05);
    discFactor nonuniform List<scalar> 1 (1.0  );
    integralFlux nonuniform List<scalar> 1 (1.0  );

    }
);

//- Relevant only for pointKinetics

promptGenerationTime  4e-04;//1.25859e-05;


Beta  (22.8e-05 78.8e-05 66.4e-05 73.6e-05 13.6e-05 8.8e-05  );  //Beta static

lambda  (0.0126 0.0337 0.139 0.325 1.13 2.5  );

feedbackCoeffFastDoppler 0;

feedbackCoeffTFuel -11.034e-05;

feedbackCoeffTClad 0;

feedbackCoeffTCool 0;

feedbackCoeffRhoCool 0;

feedbackCoeffTStruct -5.814e-05;



absoluteDrivelineExpansionCoeff 0;

controlRodReactivityMap ( ( 0.1 -0.01 ) ( 0 0 ) ( -0.1 0.01 ) );

initPrecursorsLiquidFuel true;


fuelFeedbackZones
(
    "core"
);
coolantFeedbackZones
(
    "core"
);
structFeedbackZones
(
    "core"
);
```

Appendix A.3: system Folder

blockMeshDict

```
/*--------------------------------*- C++ -*----------------------------------*\
| =======                 |                                                 |
| \      /  F ield        | OpenFOAM: The Open Source CFD Toolbox           |
|  \    /   O peration    | Version:  5.0                                   |
|   \  /    A nd          | Web:      www.OpenFOAM.org                      |
|    \/     M anipulation |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     5.0;
    format      ascii;
    class       dictionary;
    object      blockMeshDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

convertToMeters 1;

vertices
(
    (0 0 0)  //0
    (0.01 0 0)  //1
    (0.01 0.01 0)  //2
    (0 0.01 0)  //3
    (0 0 199.6) //4
    (0.01 0 199.6)  //5
    (0.01 0.01 199.6)   //6
    (0 0.01 199.6)   //7


);

blocks
(
    hex (0 1 2 3 4 5 6 7) (1 1 800) simpleGrading (1 1 1)
);

edges
(
);

boundary
(



top
{
type cyclic;
neighbourPatch bottom;
faces ((4 5 6 7));
}

bottom
{
type cyclic;
neighbourPatch top;
faces ((0 1 2 3));
}
        walls
        {
            type wall;
            faces
            (
                (7 6 2 3)
            (4 5 1 0)
            );
        }


);
```

controlDict

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  6
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      controlDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

application     GeN-Foam;

startFrom       startTime;

startTime       300;

stopAt          endTime;

endTime         400;

deltaT          0.0001;

writeControl    adjustableRunTime;

writeInterval  100;

purgeWrite      0;

writeFormat     ascii;

writePrecision  8;

writeCompression off;

timeFormat      general;

timePrecision   7;

runTimeModifiable true;

//- Physics to Solve

solveFluidMechanics true;

solveEnergy     true;

solveNeutronics true;

solveThermalMechanics false;

//- Global options

liquidFuel true;

//- Time step control options

adjustTimeStep      true;

maxDeltaT           1;

maxCo               1;

maxPowerVariation   0.025;


// ************************************************************************* //
```

topoSetDict

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.2.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      topoSetDict;
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

actions
(
    {
        name    core;
        type    cellZoneSet;
        action  new;
        source  boxToCell;
        sourceInfo
        {
            box (0 0 31) (0.01 0.01 101.8);
        }
    }
    {
        name    HotLeg;
        type    cellZoneSet;
        action  new;
        source  boxToCell;
        sourceInfo
        {
            box (0 0 101.8) (0.01 0.01 137.4);
        }
    }
    {
        name    pump;
        type    cellZoneSet;
        action  new;
        source  boxToCell;
        sourceInfo
        {
            box (0 0 137.4) (0.01 0.01  151.3);
        }
    }
    {
        name    hx;
        type    cellZoneSet;
        action  new;
        source  boxToCell;
        sourceInfo
        {
            box (0 0 151.3) (0.01 0.01  168.6);
        }
    }
    {
        name    ColdLeg;
        type    cellZoneSet;
        action  new;
        source  boxToCell;
        sourceInfo
        {
            boxes
            (
                (0 0 0) (0.01 0.01  31)
                (0 0 168.6) (0.01 0.01 199.6)
            );
        }
    }

);
```

fvSolution

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  6
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSolution;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

solvers
{
    p_rgh
    {
        solver          GAMG;
        smoother        DIC;
        tolerance       1e-8;
        relTol          0;
    }

    p_rghFinal
    {
        $p_rgh;
        relTol          0;
    }

    "e.*"
    {
        solver          smoothSolver;
        smoother        symGaussSeidel;
        tolerance       1e-8;
        relTol          0;
        minIter         1;
    }

    "h.*"
    {
        solver          smoothSolver;
        smoother        symGaussSeidel;
        tolerance       1e-8;
        relTol          0;
        minIter         1;
    }

    ".*"
    {
        solver          PBiCGStab;
        preconditioner  diagonal;
        tolerance       1e-08;
        relTol          0.001;


    }


    alpha
    {

    }
}

PIMPLE
{
    nCorrectors     1;
    nNonOrthogonalCorrectors 0;
    partialEliminationMode      implicit;

    momentumMode                faceCentered;
    oscillationLimiterFraction 0;
}

relaxationFactors
{
    equations
    {
        ".*"            1;
    }

    fields
    {

        "massTransfer"              0.25;
        "interfacialTemperature"    0.25;
    }
}


// ************************************************************************* //
```

fvSchemes

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  6
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSchemes;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

ddtSchemes
{
    default         Euler;
}

gradSchemes
{
    default         Gauss linear;
}

divSchemes
{
    default                                 none;

    div(phi,alpha)                          Gauss vanLeer;
    div(phir,alpha)                         Gauss vanLeer;

    div(phi,alpha.liquid)                   Gauss vanLeer;
    div(phir,alpha.vapour,alpha.liquid)     Gauss vanLeer;
    div(phir,alpha.structure,alpha.liquid)  Gauss vanLeer;
    div(phi,alpha.vapour)                   Gauss vanLeer;
    div(phir,alpha.liquid,alpha.vapour)     Gauss vanLeer;
    div(phir,alpha.structure,alpha.vapour)  Gauss vanLeer;

    "div\(phi.*,U.*\)"                      Gauss upwind;
    "div(alphaRhoPhi,U)"                    Gauss upwind;
    "div(alphaRhoPhiNu,U)"                  Gauss linear;
    "div(alphaRhoPhi,K)"                    Gauss upwind;

    "div\(alphaRhoPhi.*,k.*\)"              Gauss upwind;
    "div\(alphaRhoPhi.*,epsilon.*\)"        Gauss upwind;

    "div\(alphaRhoPhi.*,(h|e).*\)"          Gauss upwind;
}

laplacianSchemes
{
    default         Gauss linear uncorrected;
}

interpolationSchemes
{
    default         linear;
}

snGradSchemes
{
    default         uncorrected;
}


// ************************************************************************* //
```

neutroRegion

fvSolution

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.2.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      fvSolution;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

solvers
{
    "prec.*|precStar.*|adjoint_prec.*"
    {
        solver          PBiCG;//Stab;
        preconditioner  DILU;
        tolerance       1e-8;
        relTol          1e-3;

        /*solver          smoothSolver;
        smoother        GaussSeidel;
        tolerance       1e-8;
        relTol          0.01;*/
    }
    "flux.*|adjoint_flux.*"
    {

        solver          PCG;
        preconditioner  DIC;
        tolerance       1e-8;
        relTol          1e-3;
   /*
        solver          GAMG;
        tolerance       1e-8;
        relTol          0.01;  //0.1;
        smoother        DICGaussSeidel;//;
        nPostSweeps     1;
        nPreSweeps  1;
        nFinestSweeps   1;

        cacheAgglomeration  true;
        nCellsInCoarsestLevel 20;
        agglomerator     faceAreaPair;
        mergeLevels      1;
        processorAgglomerator masterCoarsest; */
    }
    "angularFlux.*"
    {


        solver          PBiCGStab;//GAMG;//
//  smoother        GaussSeidel;//;
        preconditioner  DILU;
        tolerance       1e-8;
        relTol          1e-1;

    }
}

neutronTransport
{
    integralPredictor           true;// integral neutron balance made at each time step to predict fluxes at next step (can be unstable)
    implicitPredictor           false;
    aitkenAcceleration          true;
    neutronIterationResidual    0.000001; // required accuracy for the coupling of different energy groups
    maxNeutronIterations        50; // up to 3-400 if no acceleration techniques
}

// ************************************************************************* //
```

fvSchemes

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.2.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      fvSchemes;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

ddtSchemes
{
    default         Euler;
}

gradSchemes
{
    default         Gauss linear;
}

divSchemes
{
    default         Gauss linear;
    "div(facePhi_,angularFlux_)"   Gauss upwind;
}

laplacianSchemes
{
   default         Gauss linear corrected;
}

interpolationSchemes
{
    default         linear;
}

snGradSchemes
{
    default         corrected;
}

fluxRequired
{
    default         false;
}

// ************************************************************************* //
```

APPENDIX B

PREDICTORS CORELLATION FIGURES

The following figures depict the correlation between the predictors used in the GeN-Foam simulation and their impact on the response variable (Power peak). **Figure 51** presents a heat map of the correlations, where the colors indicate the strength and direction of the correlations. **Figure 52**, on the other hand, presents the correlations in a more interpretable way. It shows that all correlations follow a normal distribution with 95% confidence, and it also displays the correlation index values, like in **Figure 51**. Additionally, it plots a fitting line for each correlated variable. The two figures provides are produced in R Studio, one of the best software for graphical representation for statistical analyses.

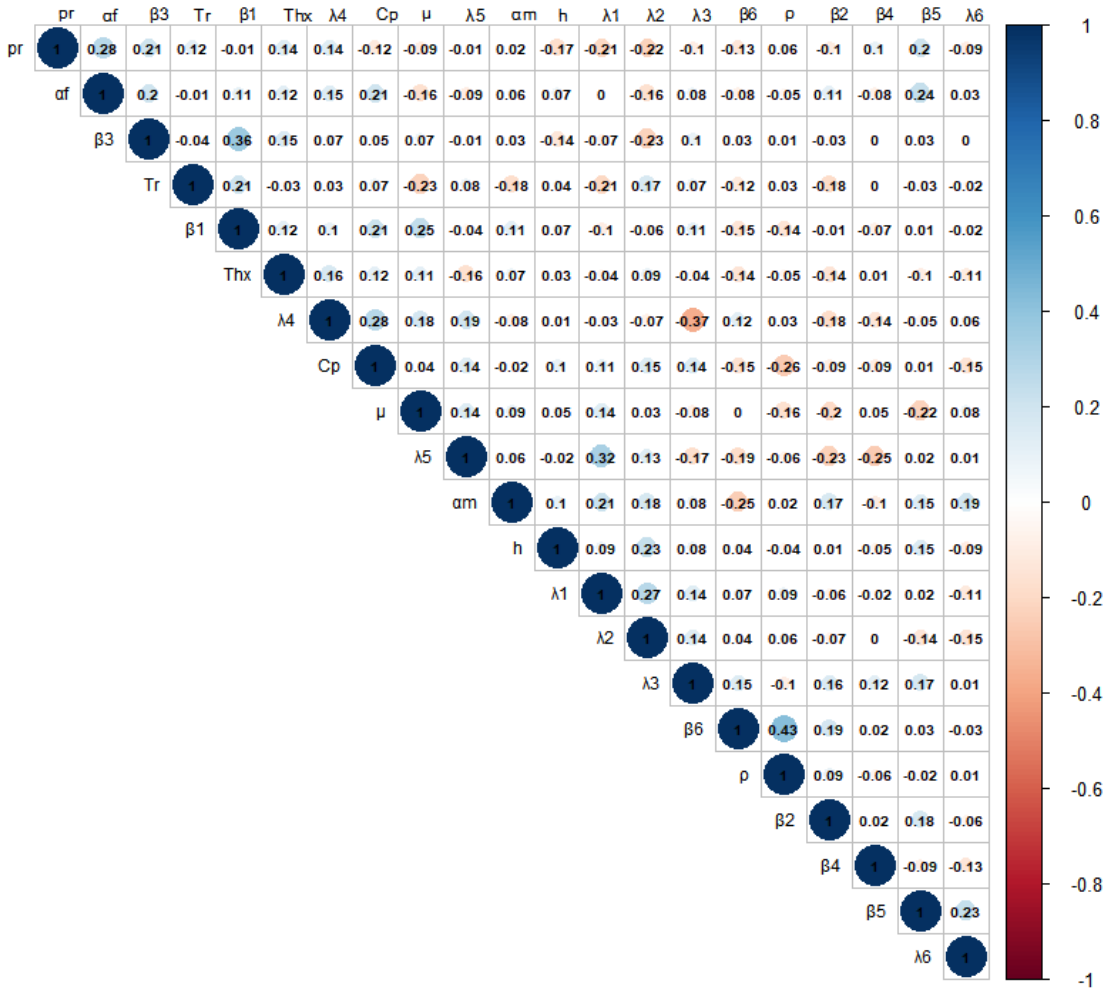


Figure 51: Diagonal heat map for the correlations of all predictors in the simulations

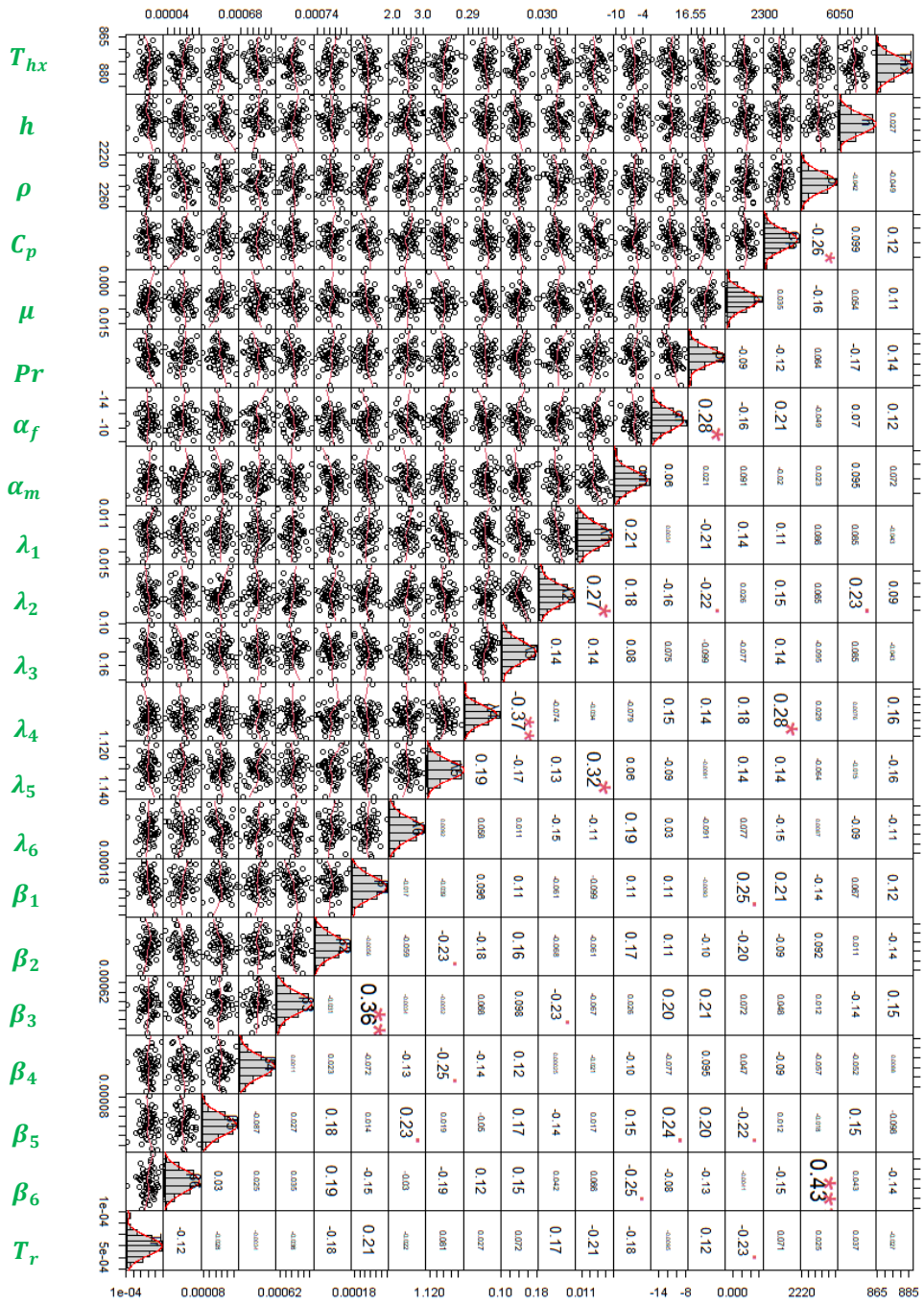


Figure 52: A full matrix of correlation prediction, normal distribution and curve fitting for simulation inputs.