



Development of the model for the multi-physics analysis of Molten Salt Reactor Experiment using GeN-Foam code

MASTER THESIS

Author:

Jiadong Bao

Supervisors:

Dr. Prof. Andreas Pautz

Dr. Jiří Křepel

Co-supervisors:

Dr. Carlo Fiorina

Dr. Konstantin Mikityuk

Project FAST

Laboratory for Reactor Physics and System Behavior

Paul Scherrer Institut

CH – 5232 Villigen

August, 2016

Abstract

Worldwide research is ongoing in pursuit of advanced nuclear systems to satisfy the more strict safety requirements. Among the potential proposals, molten salt reactor (MSR) is a promising candidate with its distinguishable advantages concerning safety, economy, sustainability, etc. The analysis of MSR requires tightly coupling between thermal-hydraulic, neutronic and thermal-mechanic solvers, thus raises challenge to the traditional codes. In PSI-FAST group, a modern multi-physics solver GeN-Foam was developed, and it is supposed to be capable of analyzing MSR. This work focuses on the verification of delayed neutron precursor drift model for GeN-Foam. Based on the Molten Salt Reactor Experiment (MSRE), a 1D channel model and a 3D full geometry model to represent the primary circuit were built up. With the models, some tests were carried out to investigate the flow field and the resulting DNP drift. It was observed that with 1D model, the GeN-Foam results match with the analytic solutions. However, with the 3D model, it was hard to obtain a stabilized and physical flow field, thus even harder to achieve a converged solution for DNP drift in the primary circuit. Due to this fact, some suggestions were proposed for the next-step work. And with further modifications, GeN-Foam is foreseen to be used for the analysis of MSR.

Keywords: MSRE, GeN-Foam, multi-physics, DNP drift, primary circuit

Acknowledgement

I would like to express appreciation to my supervisors, Dr. Jiří Křepel and Dr. Carlo Fiorina, for their patience and guidance to support my work and to Dr. Ivor Clifford, who gave me help on how to use SALOME and to understand the basis of GeN-Foam code. I would also like to thank Dr. Konstantin Mikityuk for proposing this work and to thank Pr. Andreas Pautz for hosting me in LRS. I am also grateful for the students in FAST group, Boris, Anne-Laurene, Eirik, Kajetan and Samuel, who built up comfortable and dynamic working environment, and were willing to give help on any kind of problems. Especially I need to thank Eirik, for his kind support to help me overcome stress and depression. Last but not least, I thank my family, my friends back in China, for their support and understanding.

Abstract	1
1. Introduction	4
1.1. History and current development of MSRs	5
1.1.1. Thermal Reactor designs in the history.....	5
1.1.2. Current research towards fast reactors.....	6
1.1.3. Fluoride Salt Cooled High Temperature Reactor.....	7
1.2. MSR features	8
1.3. Objectives of this work	10
2. Applied Tools and theory	11
2.1. Introduction to GeN-Foam	11
2.1.1. Thermal-hydraulic sub-solver.....	12
2.1.2. Neutronic sub-solver.....	15
2.1.3. Coupling scheme.....	16
2.2. Meshing tool SALOME	17
2.3. Monte Carlo code Serpent-2	18
3. Preliminary tests	19
3.1. Cross section generation	19
3.2. Hexahedral mesh V.S. Tetrahedral mesh	20
3.3. DNP drift in 1D channel	21
4. MSRE model development	27
4.1. Geometry Model	27
4.1.1. Reactor vessel and core.....	27
4.1.2. Heat exchanger.....	30
4.1.3. Pump.....	31
4.1.4. Pipes.....	32
4.1.5. Vanes.....	32
4.2. Mesh	34
4.3. Boundary conditions	35
4.3.1. For thermal-hydraulic solver.....	35
4.3.2. For neutronic solver.....	35
4.4. Serpent model	36
4.5. Fuel properties	36
4.6. Other zone-associated properties	37
4.6.1. Thermal-hydraulic zones.....	37
4.6.2. Neutronic zones.....	38

5. Results and discussions	39
5.1. Steady state results within vessel.....	39
5.1.1. The effect of the anti-swirl vanes.....	39
5.1.2. Lateral friction coefficients.....	39
5.1.3. Flow field.....	41
5.1.4. Power distribution.....	42
5.1.5. DNP drift.....	43
5.2. Steady state results for full primary circuit.....	45
6. Conclusions and future work	47
Acknowledgement	1
Bibliography	49
List of Figures	51
List of Tables	51
Appendix A. The salt composition used in Serpent-2 calculation	52
Appendix B. The graphs used for building geometry model	53
Appendix C. The input files for GeN-Foam concerning the porous zones ...	56
Appendix D. The nuclear data adopted by GeN-Foam	58

1. Introduction

The energy resources are the base stone of the modern human civilization. However, nowadays people are facing with more and more severe energy crisis, along with the climate change due to the consumption of fossil fuels. To tackle this issue, nuclear energy and renewable energy are proposed as promising options for replacing fossil fuels. In spite of the prospects which renewable energy advocates always claim, there are some drawbacks, like the environmental impacts, that should not be ignored [1]. As for nuclear energy, the public attitudes towards it are more controversial, especially after the accidents in Chernobyl and recently in Fukushima. Hence, more strict safety requirements are posed for next generation nuclear reactors. Following this trend, the Generation IV International Forum (GIF) was established in 2001, aiming to develop more efficient as well as safer nuclear energy systems¹, and Molten Salt Reactor (MSR) is the focus of this presented work.

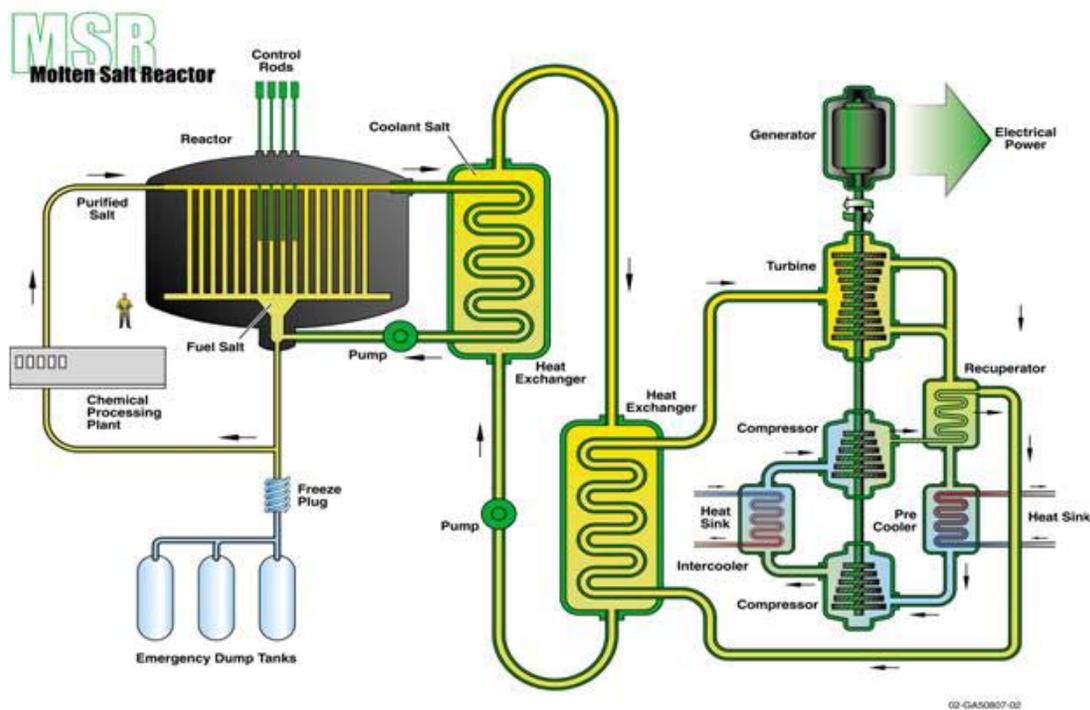


Figure 1: The conceptual scheme of molten salt reactor proposed by GIF

¹ Gas-cooled Fast Reactor (GFR), Lead-cooled Fast Reactor (LFR), Molten Salt Reactor (MSR), Supercritical Water-cooled Reactor (SCWR), Sodium-cooled Fast Reactor (SFR) and Very High Temperature Reactor (VHTR).

1.1. History and current development of MSRs

1.1.1. Thermal Reactor designs in the history

The idea of molten salt reactors was first raised in the U.S. in the 1950s. At that time, the U.S. Air Force aimed at looking for a powerful engine that could supply energy for a long time cruise for bombers. The Aircraft Reactor Experiment (ARE) was established in Oak Ridge National Laboratory (ORNL) in support of the U.S. Aircraft Nuclear Propulsion program. It was a 2.5MW_{th} thermal reactor using molten fluoride salt $\text{NaF} - \text{ZrF}_4 - \text{UF}_4$ as fuel and liquid sodium as secondary coolant, and was moderated by beryllium oxide.

In spite of the unsuccessful attempt of military function, ORNL led another molten salt reactor project for civil purpose, through 1960s. And that is the well-known Molten Salt Reactor Experiment (MSRE). The reactor was constructed in 1964, reached criticality in 1965 and the experiments were finished in 1969. The fuel for the MSRE was $\text{LiF} - \text{BeF}_2 - \text{ZrF}_4 - \text{UF}_4$ (69-29-5-1), the secondary coolant was FLiBe ($2\text{LiF} - \text{BeF}_2$), and the core was filled with graphite matrix acting as a moderator. It managed to reach a temperature as high as 923.15K and operated for about 1.5 years of full power in equivalent.

Numerous valuable data was obtained during the operation. After that, to continue the research, a new project was initiated aiming to develop a molten salt breeder reactor (MSBR). This project was funded from 1968 to 1976, ended up with a MSBR conceptual design, which would use $\text{LiF} - \text{BeF}_2 - \text{ThF}_4 - \text{UF}_4$ (72-16-12-0.4, mol%), to be moderated also by graphite, would use $\text{NaF} - \text{NaBF}_4$ as the secondary coolant. However, the project was closed in 1976 due to the research orientation being shifted towards the liquid metal fast-breeder, and no MSBR was actually constructed.

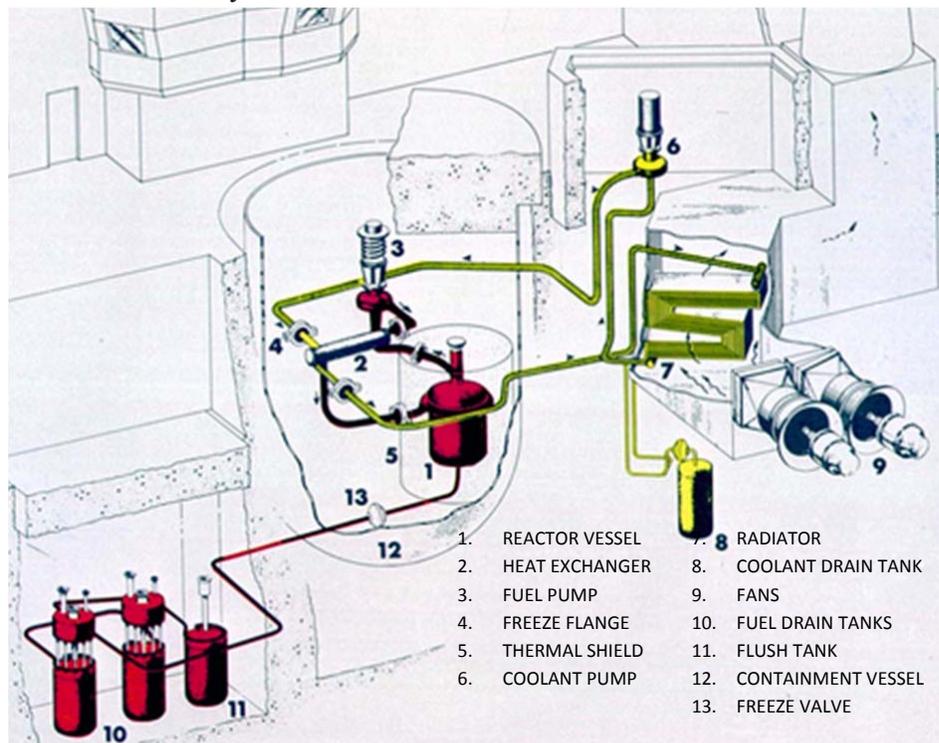


Figure 2: The flow diagram of MSRE

1.1.2. Current research towards fast reactors

After the above mentioned designs, some other design conceptions were proposed, like the denatured molten salt reactor (DMSR). But just like the programs mentioned above, these ideas were also put aside due to lack of politic support, and the tide of MSR research gradually faded out. After decades, when it comes to the new century and people started looking for advanced design for nuclear system, for the sake of safety, economy and sustainability, MSR was re-discovered and quickly gained the focus from the nuclear community. Some countries plan to reproduce the legacy design, like Chinese project Thorium Molten Salt Reactor (TMSR), while others propose new designs, like Russian project Molten Salt Actinide Recycler and Transmuter (MOSART) and the European project Molten Salt Fast Reactor (MSFR).

In the current trend of MSR research, the interest is being shifted towards fast spectrum reactors, either as a burner like MOSART, or as a breeder like MSFR. Those fast MSRs come with a lot of changes to the system design and also to the salt composition.

There is a debate on which salt base to choose, and Fluorine and Chlorine are the two candidates. Chlorine was proposed because it has the harder spectrum due to its higher Z number and low scattering cross-section, which is of interest for fast reactors. However, its drawbacks are just as obvious as its advantages, i.e. the reactor is transparent for neutrons and leakage minimization leads to bigger cores. Also the fast neutron flux on structure material may be higher. The choice of fluorides for most of current reactor designs is justified mainly by the core size and fuel chemistry. [2]

The salt mixture proposed for MSBR is not the best candidate for fast MSR application. Actinides content can be twice higher in the MSFR salt if the Beryllium is removed from the MSBR salt. Beryllium is also a good moderator which is not in favor of fast reactors. The change in salt composition leads to some challenges to the reactor design, one of which is the increased melting point. Taking thorium fuel as an example, the $Li - BeF_2 - ThF_4$ system is changed to $LiF - ThF_4$, and the mole percentage of thorium is increased from 12.3% to 22%. While the $Li - Be$ eutectic melts at $771K$, the Beryllium-free salt has a melting point as high as $841K$ [3]. Moreover, MSFR is a system with quite high thermal power, i.e. 3GW, the designed inlet and outlet temperatures are $923K$ and $1023K$ respectively [4]. This high temperature range along with the high melting point poses challenge to the design as well as the operation of MSFR, i.e. one should ensure the mechanical properties of the structure material is strong enough to face with the high temperatures, and avoid getting temperature below melting point which leads to unfavorable salt freezing. It is also worth noting that in different literatures, the specific heat of the salt are reported with different values by Beneš et al. [3] and by Merle-Lucotte et al. [4] [5]. The latter adopted a correlation leading to higher specific heat, which was employed in the MSFR design. One should note that with a lower specific heat, the challenge from temperature could be even more severe, and the final determination of the specific heat value needs to be done.

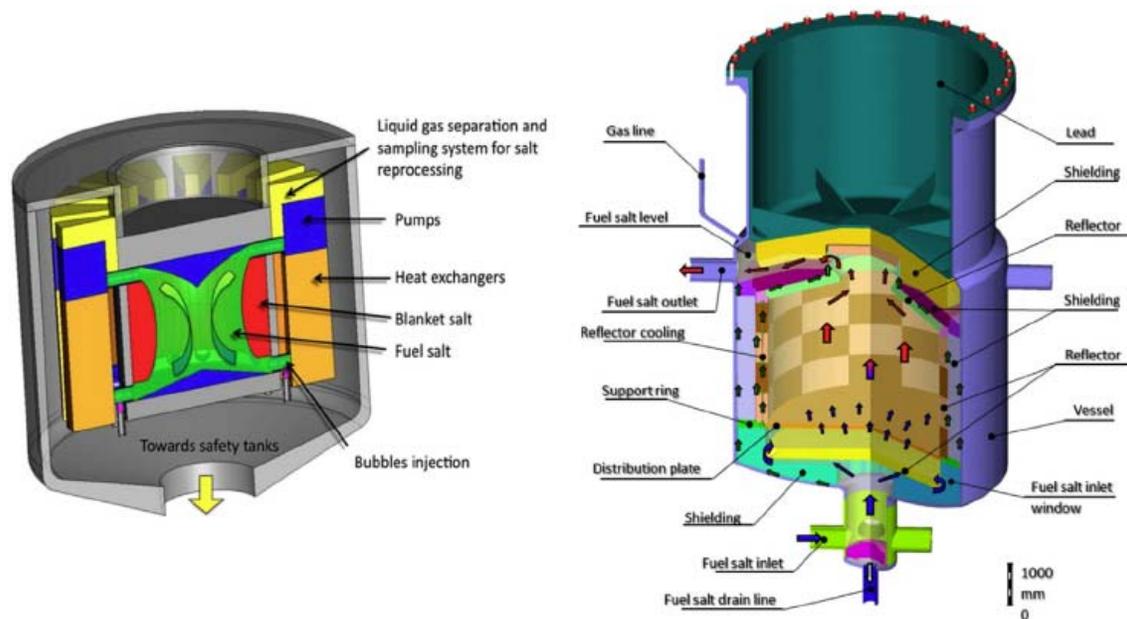


Figure 3: The schematic illustrations MSFR (left) and MOSART (right).

1.1.3. Fluoride Salt Cooled High Temperature Reactor

The Fluoride Salt Cooled High Temperature Reactor (FHR) is another category of nowadays MSR research. Unlike the above mentioned designs, FHR maintains solid fuels, in the form of TRISO² fuel which is used in High Temperature Reactor (HTR) and proved to be reliable in a high temperature environment, and employs molten salt as coolant only. The reactor works in thermal spectrum with graphite being used as the moderator. This type of design gains the advantages of cooling molten salt, such as low pressure environment and high operation temperature [6], but it excludes the attractive features of liquid fuel system concerning sustainability. Furthermore, it preserves the low fuel density of HTR, and the parasitic neutron consumption on the coolant salt is thus higher than in MSR. In spite of the relatively mature technology borrowed from HTR, there are still some common challenges that MSRs are facing with, such as the corrosion problems. Nonetheless, the FHR design might be easier to achieve and can be considered as a transition phase towards the liquid-fuel MSRs.

² Tristructural-isotropic fuel

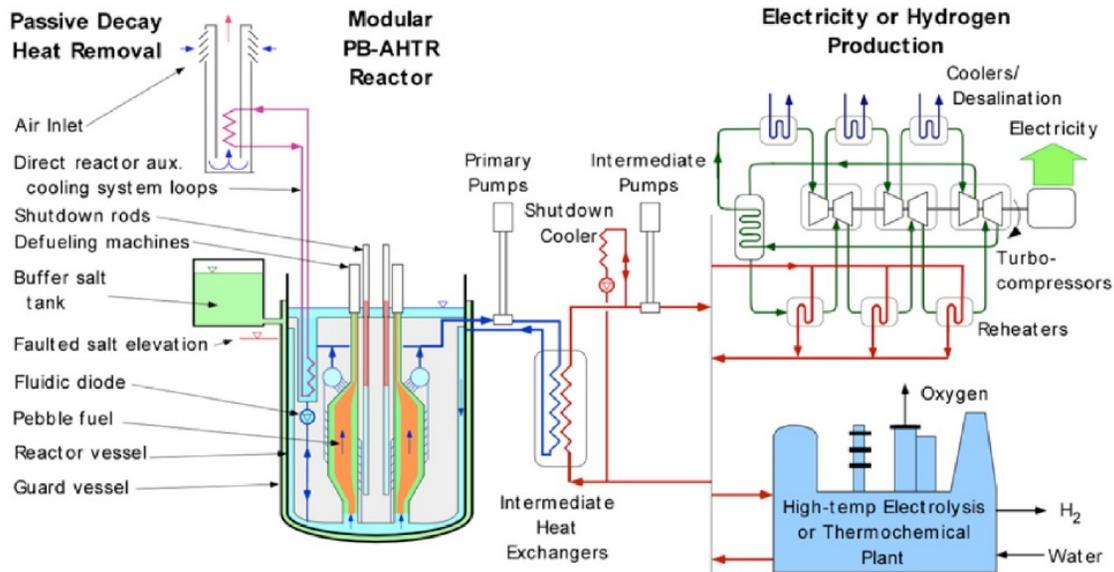


Figure 4: Conceptual drawing of a pebble bed FHR and power generation cycle. [7]

1.2. MSR features

The liquid fueled MSR designs bring some advantages compared to solid-fueled ones, and those advantages are the reasons that distinguish MSR from the other generation IV designs. Here below is a brief summary of MSR's special features over other nuclear systems and also the advantages of liquid fueled over solid fueled systems [8] [2].

- **Safety:**
 - **Negative temperature feedback coefficients.** The salt temperature coefficient is negative due to the negative Doppler coefficient which is in majority of the designs supported by negative salt expansion coefficient. Nonetheless, it is worth noting that MSBR is an exception concerning salt expansion, as the expansion coefficient can be positive for this under-moderated breeder. Besides, the graphite temperature feedback can also be positive [9]. But in general, these slight positive feedback coefficients will not prevail the negative Doppler coefficient in well-designed thermal breeder, and larger safety margin is thus obtained thanks to the overall negative temperature feedbacks.
 - **Easier decay heat removal.** The fuel is dissolved in liquid salt. Therefore it is easy to empty the primary circuit by draining the liquid fuel into storage tanks, and remove the decay heat there while keeping it sub-critical.
 - **Chemical instead of Mechanical barriers.** Majority of fission products (FP) is soluble in the salt and stay there up to very high temperatures [10]. The non-soluble FP in form of gaseous and volatile compounds may be extracted through the off-gas system and chemically immobilized. In general the chemical stability of the FP in MSR acts as a strong barrier. And in case of an

accident, the source term in the core is minimized as the radioactive fission products can be continuously removed during the operation.

- **Low excess reactivity.** Thanks to the recirculation and online access to the fuel salt, there is no need to start the reactor with a high excess reactivity.
- **Low pressure operation conditions.** The low pressure condition imposes less threat to the mechanical integrity of the structure material. Besides, it is less likely to eject radioactive materials outside of the barriers in an accident scenario, thus minimizes the radioactive release.
- **Economy:**
 - **Neutron economics – structural material absence.** Since the geometry can be simpler compared to the conventional reactor design, i.e. no cladding or spacer grid, etc., the loss of neutrons in the structure material can be significantly reduced.
 - **Neutron economics - high availability.** Since the criticality is maintained online, there is no need to shut down the reactor for refueling, thus increase the system's availability.
 - **No fuel fabrication.** The complicated and expensive process of fuel fabrication for common commercial reactors can be avoided with the liquid fuel being employed.
 - **No Fuel enrichment.** Once iso-breeding state will be achieved, the fuel enrichment would be not necessary and the fertile resources ^{238}U or ^{232}Th would represent the only required material.
 - **High temperature.** The relatively high operation temperature enables higher inlet temperature of the working fluid for turbines, thus increase the thermal efficiency concerning energy conversion. It is also beneficial for some chemical, such as hydrogen production.
- **Sustainability:**
 - **Actinides recycling.** The circulating liquid fuel provides convenience concerning fuel reprocessing and actinides recycling, thanks to the online access to the fuel and because it avoids any fabrication issues.
 - **Breeding and burning.** Due to smaller loss, more neutrons participate in the process; combining with fast spectrum, it is favorable to use MSR for thorium and/or uranium breeding and/or actinide burning.

Nevertheless, MSRs do have their drawbacks and some of the key difficulties are listed below [11].

- **High temperature.** Although it is mentioned above that high temperature brings in some advantages, it also challenges the mechanical properties of structure materials.
- **Corrosion.** The salt is corrosive to the metal structures. Even if some alloys show good resistance to the corrosion, like the INOR-8, the resistance property in a long term, over decades for the operation phase of a reactor, remains to be seen. The redox potential control is the key solution for mitigation of corrosion.

- **Embrittlement.** Under the circumstances with high neutron flux, nickel based alloys suffer from embrittlement. Interaction on Ni with thermal neutrons leads to He production. This tends to damage the mechanical integrity of the pressure vessel, thus limit the life time of a reactor.
- **Low effective delayed neutron fraction.** The motion of liquid fuel takes out the delayed neutron precursors from the core thus reduces the effective delayed neutron fraction.
- **Fuel in entire primary circuit.** The highly radioactive liquid fuel may be present in whole primary circuit, draining tanks, or in the reprocessing facility. Hence the maintenance may be complicated.

1.3. Objectives of this work

Safety evaluation of MSR should include transient analysis of the reactor accounting for all possible accidents. Due to the liquid fuel, the thermal-hydraulic and neutron kinetic analysis of a MSR should be tightly connected by simulating these transients, therefore GeN-Foam³, a modern multi-physics coupling computational tool, has been developed in PSI-FAST group. The thermal-hydraulic solver has been verified with many tests [12], while the recently developed neutronic solver with the drift model for delayed neutron precursors lacks of verification and validation.

This work aims to GeN-Foam application to MSRE simulation, which has the largest available database for MSR experiments. It should be used to the verification and validation of drift model for the delayed neutron precursors (DNPs). It consists of three parts. The first part is an overview of the thermal-hydraulic and neutronic models and their implementations in GeN-Foam, with emphasis on the porous medium approach and DNP drift model. The second part focuses on the model set up of MSRE. And the third part presents the results of the simulations done with GeN-Foam and the comparisons to experimental data as well as the results from other codes, like Serpent-2.

³ Details are introduced in Chapter 2.

2. Applied Tools and theory

2.1. Introduction to GeN-Foam

GeN-Foam stands for Generalized Nuclear Foam. It is a multi-physics solver based on OpenFoam, which is basically a set of C++ libraries [13].

Before this platform was developed, in PSI-FAST group, the transient analysis of a reactor was carried out in a classical way of coupling legacy codes for neutron diffusion and 1-D or sub-channel thermal hydraulics, e.g. TRACE-PARCS. But this classic scheme was born with some limitations, such as the errors from the data exchange interface and the inflexibility of parallel computing. Moreover, besides the coupling between thermal-hydraulics and neutronics, the thermal-mechanical analysis also needs to be included in a transient problem, which TRACE-PARCS was unable to deal with. As modern simulations require more advanced tools for more sophisticated problems with larger size, it is of great necessity to develop new generation solvers. To reduce the development efforts, an open source C++ library, OpenFOAM, was adopted as a base and start point.

This library contains various built-in methods for discretization of partial differential equations and it is capable of parallel computing. It supports unstructured meshes, which provides a high level flexibility for users who work with complex geometries. Based on the said features, GeN-Foam was developed with four kinds of sub-solvers:

- Sub-scale fuel model. A finite-difference method is employed for coarse-mesh simulations, dedicated for computing the temperature profiles in fuel and cladding.
- Thermal-Mechanical sub-solver. There is already a thermal-mechanical solver in OpenFOAM library, which can be modified to be used for coupled simulation.
- Thermal-hydraulic sub-solver. A typical OpenFOAM CFD solver, pimpleFoam was selected as a starting point for developing the named sub-solver. It is based on a standard $k - \epsilon$ model for solving turbulence problems and relies on a pressure-based iteration algorithm. The new feature added to the solver, is the so called porous medium approach⁴, which enables simulating complex geometry with a coarse-mesh and preserves the fine-mesh solution for open spaces.
- Neutronic sub-solver. This solver takes the cross sections generated by a Monte Carlo code, which is Serpent-2 for this work, as an input. And solves the multi-group neutron diffusion equations based on a deformable mesh, thanks to the coupling with the thermal-mechanical sub-solver. Moreover, a drift model for the delayed neutron precursors was included in the solver, which enables the dynamic analysis for a molten salt reactor.

⁴ Details of porous medium approach are explained in Section 2.1.1.

This work focuses on the coupling between the thermal-hydraulic and neutronic solvers, especially on the DNP drift.

In the following sections, more details are introduced to give a more clear explanation of the thermal-hydraulic and neutronic sub-solvers.

2.1.1. Thermal-hydraulic sub-solver

In the field of nuclear engineering, system code, such as RELAP, TRACE, CATHARE, and etc., have been playing a crucial role for decades in the reactor design and researches. Their performance is highly reliable as far as the integrated response is considered. Nonetheless, as the reactor design is facing with increasingly strict safety requirement, some localized phenomena and three dimensional problems need to be taken into account, where system codes fail to give predictions. One solution is to implement a 3D VESSEL component in system codes [14], but this only partially solves the problem in the reactor vessel. The other is to couple system codes with CFD codes. Thanks to the advanced coupling tools, more details can be obtained while the response on system level is still available, e.g. the boron dilution in the core [15], and the pressurized thermal shock on RPVs [16]. However, as mentioned before, most of the coupled codes are based on an explicit data exchange interface, which are prone to errors.

As for GeN-Foam, the problem is resolved on a single computation domain, with fine mesh for the open space but with coarse mesh and porous medium approach for the complex geometry. The errors caused by data exchange through explicit interface are thus eliminated. In this section, the details of the porous medium approach are presented.

It is derived from Navier-Stokes equations, which describe the motion of fluid elements. The typical NS equations and the energy conservation equation for a single-phase flow are listed below:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{u} = 0, \quad (2.1.)$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\nabla P + \nabla \cdot \boldsymbol{\tau} + \mathbf{F}_g, \quad (2.2.)$$

$$\frac{\partial \rho e}{\partial t} + \nabla \cdot (\mathbf{u}(\rho e + p)) + \nabla \cdot \mathbf{q}'' = Q''' , \quad (2.3.)$$

where \mathbf{u} is the velocity vector, ρ is the density, p is the pressure, $\boldsymbol{\tau}$ is the shear stress tensor, \mathbf{g} is the gravity acceleration, C_p is the specific heat, e is the total energy, κ is the thermal conductivity, T is the temperature, and \dot{Q} is the heat source. The energy conservation equation is written in an internal energy formulation.

In a region containing complex geometry, e.g. the core of MSRE where the liquid fuel flows through the channels surrounded by graphite blocks, considering the liquid-solid system as a two-phase flow, a mixture model of liquid and stationary solid phases with inter-phase momentum and heat transfer is applicable. In the work of Clifford [17], a mixture model of

gas-liquid system was derived from a general two-phase model based on the work of Saurel [18]. Following the same idea, i.e. discarding the mass and momentum equations for solid phase as well as the balance equation for phase fractions, referring solid phase as a sub-scale geometry, the conservation equations of a liquid-solid mixture can be written as shown below.

$$\frac{\partial \gamma_l \rho_l}{\partial t} + \nabla \cdot (\gamma_l \rho_l \mathbf{u}_l) = 0, \quad (2.4.)$$

$$\frac{\partial \gamma_l \rho_l \mathbf{u}_l}{\partial t} + \nabla \cdot (\gamma_l \rho_l \mathbf{u} \otimes \mathbf{u}) = -\gamma_l \nabla P_l + P_i \nabla \gamma_l + \nabla \cdot (\gamma_l \boldsymbol{\tau}) + \gamma_l \mathbf{F}_{ss} + \gamma_l \mathbf{F}_g, \quad (2.5.)$$

$$\frac{\partial \gamma_l \rho_l e_l}{\partial t} + \nabla \cdot (\mathbf{u}_l (\gamma_l \rho_l e_l + \gamma_l P_l)) + \nabla \cdot (\gamma_l \mathbf{q}''') = \gamma_l \mathbf{F}_{ss} \mathbf{u}_l + \gamma_l Q_{ss}''', \quad (2.6.)$$

where the subscript l denotes the liquid phase and ss the sub-scale geometry.

A typical CFD approach, like RANS, applies time average on all the quantities, and treats the non-linearity arising from the advection term with Boussinesq eddy viscosity assumption, i.e. proportional to the deformation tensor with the so-called “eddy viscosity” μ_T as a coefficient. This assumption can also be applied here, thus the effect of turbulence is taken into account, unlike in system codes, where the turbulence effect is totally modelled by empirical correlations. In addition to that, the terms \mathbf{F}_{ss} , Q_{ss}''' represent the drag force exerted on the fluid, and the heat transfer from the sub scale geometry, respectively, which require empirical correlations to Reynolds number. The combination of the two models makes up the main idea of porous medium approach on a coarse mesh, i.e. on one hand, takes into account the turbulence effect, on the other hand, using empirical correlations to compensate the lost accuracy compared to a fine mesh solution.

In order to obtain the final format of the equations, some approximations and simplifications need to be introduced.

- Liquid fuel is the only moving phase whose motion is under investigation, the subscript l can thus be abandoned.
- Apply volume and time average to all quantities in the equations [17].
- Use Fourier’s law to model the heat flux \mathbf{q}'' . Therefore, the heat conduction term can be represented as shown below:

$$\nabla \cdot (\gamma \mathbf{q}'') = -\nabla \cdot (\gamma k_T \nabla T), \quad (2.7.)$$

where k_T is the turbulence conductivity to take into account turbulence effect concerning thermal diffusion.

- Adopt equilibrium pressure model, thus the interfacial pressure P_i is equal to P . [17]
- To maintain a continuous velocity profile at the interface between cell zones with different porosities, rewrite the velocity in the form of Darcy velocity. [13]

$$\mathbf{u}_D = \gamma \mathbf{u} \quad (2.8.)$$

With the above approximations and assumptions, equations (2.4.) to (2.6.) can be rewritten as [13]:

$$\frac{\partial \gamma \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}_D) = 0, \quad (2.9.)$$

$$\frac{\partial \rho \mathbf{u}_D}{\partial t} + \frac{1}{\gamma} \cdot (\rho \mathbf{u}_D \otimes \mathbf{u}_D) = \nabla \cdot (\mu_T \nabla \mathbf{u}_D) - \gamma \nabla P + \gamma \mathbf{F}_g + \gamma \mathbf{F}_{ss} - (\rho \mathbf{u}_D \otimes \mathbf{u}_D) \nabla \frac{1}{\gamma}, \quad (2.10.)$$

$$\frac{\partial \gamma \rho e}{\partial t} + \nabla \cdot (\mathbf{u}_D (\rho e + P)) = \gamma \nabla \cdot (k_T \nabla T) + \mathbf{F}_{ss} \cdot \mathbf{u}_D + \gamma Q_{ss}''' + (k_T \nabla T) \cdot \nabla \gamma, \quad (2.11.)$$

and now the spatial derivative of porosity only appears in the last term of Eq.2.10. and Eq. 2.11., which can be simply neglected in most applications in nuclear engineering, since there is no change of porosity within a cell zone, but only at the interface between different zones and typically in the direction of the flow [13].

To fully close the equations, special treatments of \mathbf{F}_{ss} and turbulence are listed below⁵.

- **Drag force modelling.** \mathbf{F}_{ss} represents the drag force exerted on fluid by the sub scale geometry, thus it is proportional to the velocity and can be interpreted as a pressure drop, thus on each direction, there is [13]:

$$F_{ss,i} = (\nabla P)_i, \quad (2.12.)$$

where $(\nabla P)_i$ is the pressure gradient on i_{th} direction. There are various correlations could be applied for modelling the pressure drop, here it can be expressed as [13]:

$$(\nabla P)_i = \frac{f_{D,i} \rho u_{D,i}^2}{2 D_h \gamma^2}, \quad (2.13.)$$

where f_D and D_h are the Darcy friction factor and hydraulic diameter on the concerned direction. It is then assumed that [13]:

$$f_{D,i} = A_{f_{D,i}} Re^{B_{f_{D,i}}}, \quad (2.14.)$$

where Reynolds number is computed based on the local velocity magnitude and the hydraulic diameter given by user. As for $A_{f_{D,i}}$ and $B_{f_{D,i}}$, it is the user's responsibility to select the values based a suitable model.

- **Turbulence modelling.** For clear fluid, the terms μ_T and k_T are modelled with standard $k - \epsilon$ equations for RANS model. As for porous zones, since the interaction between fluid and sub scale structure prevails the turbulence effects, it is not necessary to model these two terms with a high accuracy. Nonetheless, it is crucial to predict reasonable values at the exit of porous zones, as they will affect the turbulence effect in clear fluid regions. Therefore, instead of solving the standard $k - \epsilon$ equations for porous zones, an alternative method is adopted, i.e. forcing values to converge to user-selected values with an convergence rate [13]:

$$\frac{\rho \partial \epsilon}{\partial t} + \nabla \cdot (\rho \mathbf{u}_D \epsilon) = \rho \lambda_{\epsilon/k} (\epsilon_0 - \epsilon), \quad (2.15.)$$

$$\frac{\rho \partial k}{\partial t} + \nabla \cdot (\rho \mathbf{u}_D k) = \rho \lambda_{\epsilon/k} (k_0 - k), \quad (2.16.)$$

⁵ Since heat transfer is not included in this work, the modelling of sub scale heat source is not presented, for details, please refer the reference [13].

where k_0, ϵ_0 are the user selected values and $\lambda_{\epsilon/k}$ is the convergence rate.

If no information is available for the k_0, ϵ_0 at porous zone outlet, the equilibrium values in the porous zone can be taken as a reference by computing the turbulent length scale and the turbulent intensity from available correlations of the form [13]:

$$I = A_i R e^{B_i}, \quad (2.17.)$$

$$l = A_i D_h, \quad (2.18.)$$

and their values can be used to compute k and ϵ and proposed by Launder et al. [19]:

$$I = \frac{1}{|u|} \left(\frac{2}{3} k \right)^{0.5}, \quad (2.19.)$$

$$l = 0.1643 \frac{k^{\frac{3}{2}}}{\epsilon}, \quad (2.20.)$$

2.1.2. Neutronic sub-solver

This sub-solver deals with the standard multi-group neutron diffusion equation:

$$\frac{1}{v_i} \frac{\partial \phi_i}{\partial t} = \nabla \cdot D \nabla \phi_i + \frac{S_{f,i}(1-\beta_{eff,i})\chi_i}{k_{eff}} \phi_i - \Sigma_{d,i} \phi_i + S_d \chi_i + S_{s,i}, \quad (2.21.)$$

where i denotes the group index, ϕ is the neutron flux, v is average neutron velocity, D is diffusion coefficient, Σ_f is fission cross section, β_{eff} is the effective delayed neutron fraction⁶.

And the source terms $S_{f,i}$, S_d , and $S_{s,i}$ are defined respectively as shown below.

$$S_{f,i} = \sum_j v \Sigma_{f,j} \phi_j \quad (2.22.)$$

$$S_d = \sum_k \lambda_k C_k \quad (2.23.)$$

$$S_{s,i} = \sum_{j \neq i} \Sigma_{j \rightarrow i} \phi_j \quad (2.24.)$$

In order to solve problems with circulating liquid fuel, these equations are coupled to a transport equation for solving the concentration of delayed neutron precursors C_k , which takes into account the advection and diffusion caused by fuel movement and it is shown below.

$$\frac{\partial C_k}{\partial t} + \nabla \cdot (\mathbf{u}_D C_k) = \frac{\beta_{eff,k} \sum_j v \Sigma_{f,j} \phi_j}{k_{eff}} + \nabla \cdot (v_T \nabla C_k) - \lambda_k C_k \quad (2.25.)$$

With traditional coupling tool between system code and neutron kinetic codes, e.g. TRACE-PARCS, the turbulence effect is neglected, and the mixing of DNP in open space, e.g. the plena, cannot be well captured due to the 0 or 1 dimension features of system codes. While with GeN-Foam, a more accurate transportation of DNP can be predicted.

⁶ $\beta_{eff,i}$ is used in the equation with the same neutron yield for both prompt and delayed neutrons. Nevertheless, it can be replaced with β_0 , and in that case different neutron yields for prompt and delayed neutrons can be employed.

The sub-solver provides users with high flexibility to select the number and structure of both neutron energy and precursor groups. As mentioned before, the sub-solver takes cross sections generated by Serpent-2 as inputs. Actually, the mesh for neutronic can be divided into several cell zones, similar to the mesh for thermal-hydraulic, and each cell zone is assigned with a homogenized set of cross sections. The conversion is achieved by executing an Octave script. Both time dependent and eigenvalue calculations can be handled with this sub-solver. For eigenvalue problem, with the assigned cross-section sets, a traditional power iteration algorithm [20] is used to compute the neutron flux and the k_{eff} . For a time-dependent case, i.e. reactivity feedbacks taken into account, different cross sections for different perturbed states can be associated in each zone. In each cell, based on the local temperatures of fuel and structure materials as well as the radial and axial expansions, the sub-solver compute the correct cross sections by interpolating between the nominal state and a preselected perturbed state. The interpolation is linear except for fuel temperature in fast reactors, where a logarithmic interpolation is more suitable [21].

It is worth noting that, for a 1D channel case, which is shown in Section 3.3, the turbulence actually needs to be ignored to get reasonable result. And for the 3D case, the turbulence also needs to be removed to avoid blow-up. Therefore, whether to reserve the turbulence term in the equation requires further investigation.

2.1.3. Coupling scheme

The GeN-Foam solver consists of four sub-solvers. The data exchange is achieved by mapping variables from one mesh to another, via a standard OpenFOAM cell-volume-weighted algorithm. The coupling is between all sub-solvers and is a semi-implicit coupling scheme. Since in this work, only the coupling between fluid mechanic equations and neutronic equations is investigated, the full coupling scheme is not presented, but the simplified one for incompressible fluid without thermal expansion.

In a two way coupling, the initial time step can be specified by the user, but the subsequent time step size is computed by the code, based on the more strict criterion between the Courant number condition and a user-selectable maximum allowed power variation, which is set to 2.5% by default.

In the present work, (maybe not true later)only one way loose coupling is considered. Following the typical PIMPLE loop, the mass conservation equation is first solved, then the transport equations for k and ϵ . After that, a velocity prediction is conducted based on the pressure field from the previous iteration. This predicted velocity field is then used for solving the Poisson equation to update the pressure field. Optionally, a pressure correction equation is solved several times to account for the non-orthogonality effects. Then the velocity field is corrected based on the new pressure field. The prediction and correction loop will be repeated several times until predefined convergence criteria are satisfied. Then a new time step begins unless the desired flow field is obtained. After achieving the flow field, GeN-Foam then maps it to the neutronic mesh, which carries out eigenvalue calculation with DNP drift.

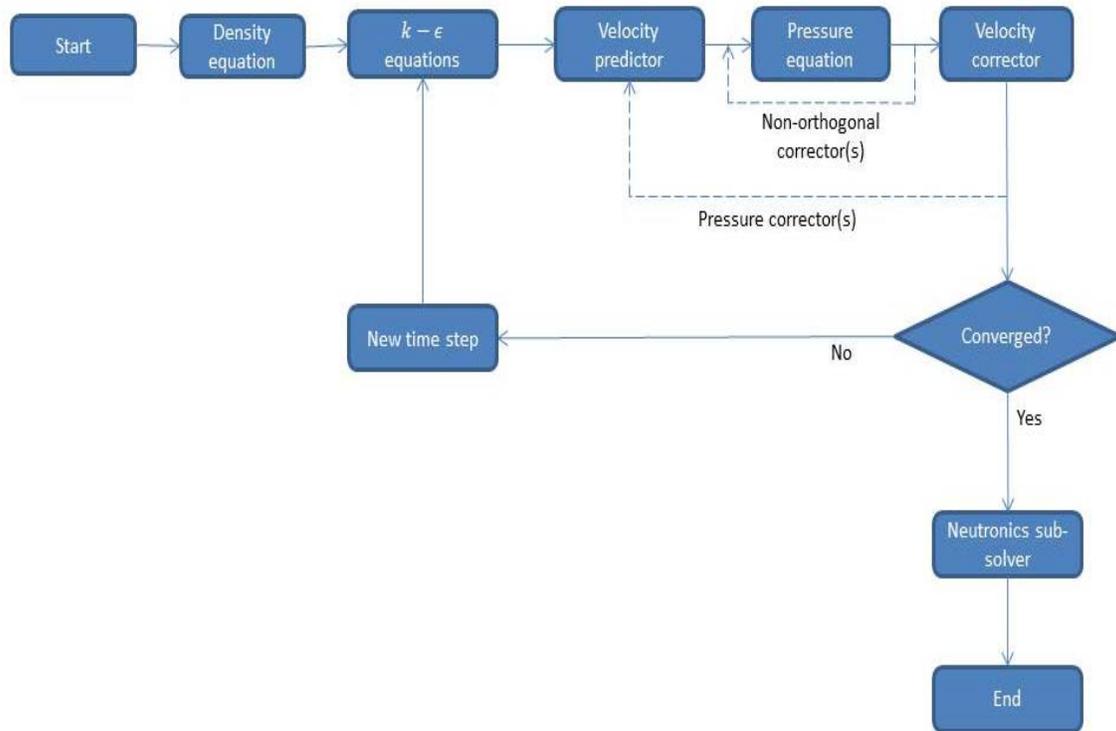


Figure 5: Simplified GeN-Foam coupling scheme for this work

2.2. Meshing tool SALOME

SALOME, developed by CEA, EDF and OPENCASCADE, is an open source integrated CAD/CAE platform that supports pre- and post- processing for numerical simulations [22]. Its ability of modelling and meshing geometries are used in this work.

The geometry module (GEOM) of SALOME allows the user to build complex geometry via graphical user interface (GUI) or text user interface (TUI) in the form of a python script. The meshing module (SMESH) contains various meshing algorithms which are particularly suited for finite-element and finite-volume methods. It allows the user to split a geometry into several regions and mesh them individually, which is a useful feature for this work, since it provides flexible meshing for the core, heat exchanger and pump components of MSRE model, which are treated as porous media.

SALOME can export the mesh into several file formats. For this work, .unv file is selected, and an OpenFOAM utility, ideasUnvToFoam, can convert the mesh into readable files for OpenFOAM solvers.

It is worth noting that when it comes to too complex geometry and a mesh with too many cells, i.e. exceeding 50,000, it is better to use TUI rather than GUI, since the later requires more memory while processing a task and can easily cause a crash.

2.3. Monte Carlo code Serpent-2

Monte Carlo method is a probabilistic method, which apply random sampling to obtain numerical results for problems which are difficult or impossible to solve by deterministic methods. In terms of its application in nuclear reactor physics, a Monte Carlo code tracks the path of a single neutron from its initial birth to the final death by absorption or escaping. The possible interactions during a neutron's life time are randomly sampled. After sampling large number of neutrons, a detailed result with statistic uncertainty can be obtained. It has the advantage of simulating very complex geometry with high fidelity and apparently the sacrifice here is the computational efforts.

Serpent-2 is a Monte Carlo code, specializing on burnup calculation. And it was developed at the VTT Technical Research Centre of Finland. It adopts the continuous-energy interaction data from ACE format cross section libraries [23]. The user can choose between various libraries. In this work, the library JEFF-3.1.1u is selected. With the homogenization methods, it is possible to use Serpent-2 for generating the desired zone-average cross section sets for GeN-Foam's neutronic solver.

3. Preliminary tests

3.1. Cross section generation

As mentioned before, GeN-Foam relies on other codes to provide the cross section sets. In the present work, Serpent-2 is employed for this purpose. With the conventional method, to generate the cross sections, one needs to divide the whole domain into several zones, and then the average cross sections of each zone will be computed by Serpent. With the help of an Octave script, those data can be converted to readable file for GeN-Foam and each set of cross sections is assigned to its corresponding zone. Nonetheless, due to the heterogeneity of geometry, the cross sections in the regions at the center and those close to the periphery strongly differ from the average values, thus significant discrepancy may arise in criticality calculation, when the averaged cross sections are employed. Another method, in order to minimize this discrepancy, is to generate the cross sections based on an infinite lattice, since neutrons at the center see repetitive geometries, similar to an infinite scenario.

In this section, simple geometries were created for the investigation on the two methods. In Serpent input file, a lattice composed of graphite bar and fuel salt was filled into a $142.324\text{cm} \times 142.324\text{cm} \times 170\text{cm}$ cuboid. Two types of cross sections were generated, with black boundary⁷ and reflective one, respectively. For GeN-Foam, the same geometry was adopted and one simulation was carried out for each set of cross section, both with a 0 flux boundary condition. It was observed that the one with cross sections generated from the infinite lattice shows better agreement with the Serpent result. The obtained k_{eff} values are listed in Table 1, noting that the Serpent k_{eff} was obtained when it was set with a black boundary condition.

	Serpent-2	GeN-Foam	
		Infinite lattice XS	Zone-averaged XS
k_{eff}	1.14689	1.143717	1.120623

Table 1: Comparison of k_{eff} values, XS means cross sections

These tests were carried out for the core zone, where neutrons at the center actually see an environment similar to infinite lattice. However, for some zones, this method may not necessarily give better agreements, e.g. in the downcomer. To keep consistency with previous study, the conventional zone-averaged method was adopted to generate the cross sections in this work.

⁷ Neutrons are killed at the boundary.

3.2. Hexahedral mesh V.S. Tetrahedral mesh

In general, with a given number of cells, the structured mesh provides more accurate solution and requires less computational efforts [24], however it takes longer time, sometimes unaffordable, to mesh a complex geometry, while the unstructured mesh can be easily obtained if it is acceptable for the problem under investigation. But unstructured mesh comes with issues leading to errors, e.g. non-orthogonality, etc. To obtain accurate results, the user needs to select appropriate numerical schemes to amend the errors arising from the bad mesh quality.

As illustrated in Figure 6, the cell centers of two adjacent cells are denoted with point P and N , respectively. The center of their interface is represented by point f and vector \mathbf{S} is the face normal vector. The non-orthogonality is then the angular deviation between the vectors \mathbf{S} and \mathbf{PN} . This deviation affects the gradient of the face center f , and thus affects the diffusive terms [24].

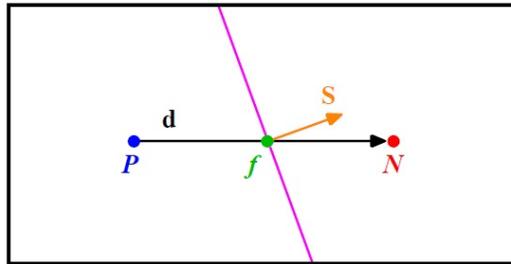


Figure 6: Schematic drawing of mesh non-orthogonality. [24]

The GeN-Foam user can select between corrected schemes, i.e. take into account the non-orthogonality for entire mesh which potentially leads to unstable solutions, or uncorrected scheme, meaning totally discarding this issue with the risk of ending up with severely inaccurate results. A tradeoff is required between stability and accuracy. The user can then use the limited ϕ scheme to find the needed balance. The value of ϕ can be selected in the range from 0 to 1, with 0 meaning uncorrected and 1 meaning corrected [25].

In this section, a test of sensitivity to mesh types and numerical schemes concerning neutron solver is presented. The geometry is the same as the one described in Section 3.1. Tetrahedral and hexahedral meshes were created for the geometry, as illustrated in Figure 7.

In terms of mesh sensitivity, hexahedral mesh converges to a mesh-independent much faster than the tetrahedral one, which agrees with general rule that hexahedral mesh is more efficient with a given cell number. It is also clear that with the scheme *faceCorrected* for surface normal gradient and *leastSquares* for gradient, the result from tetrahedral mesh shows good agreement with the one from a hexahedral mesh. The comparison is shown in Figure 8.

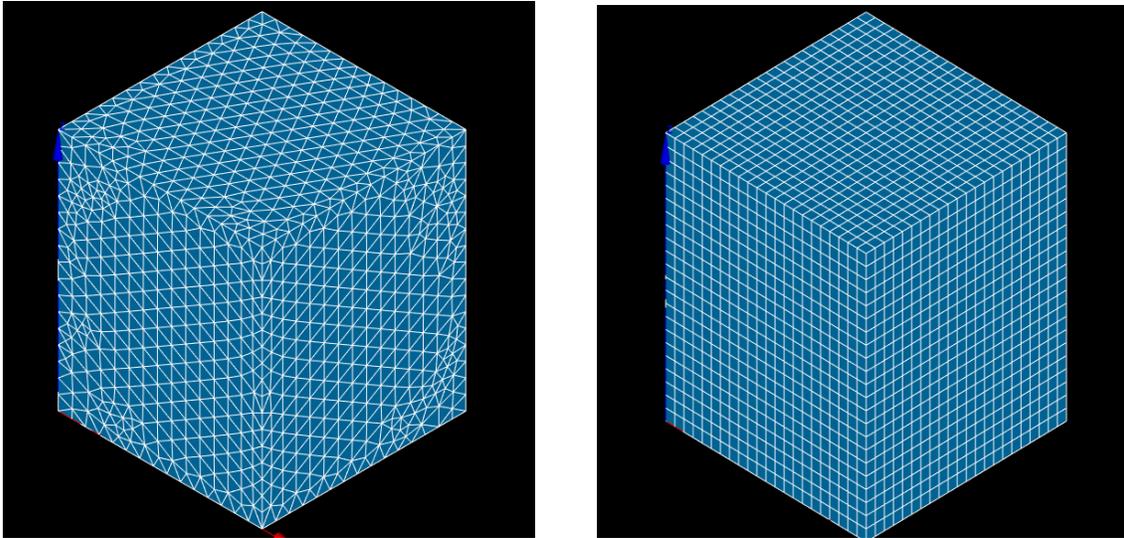


Figure 7: The layout of tetrahedral mesh (left) and hexahedral mesh (right) for a cuboid geometry.

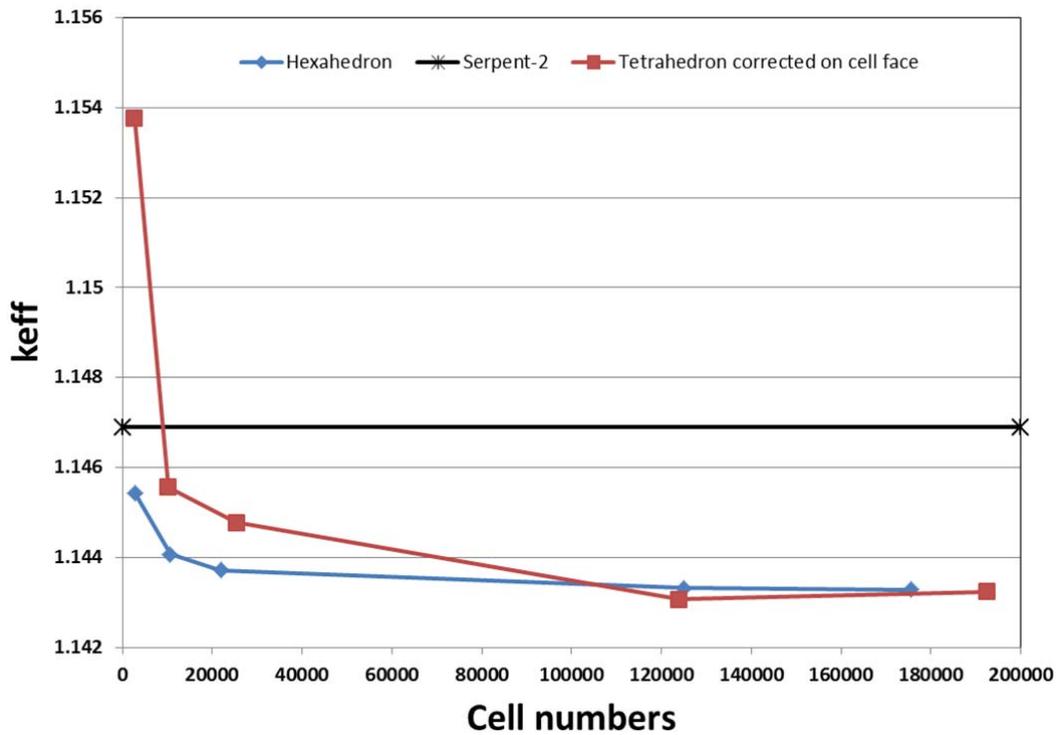


Figure 8: The comparison of sensitivity to mesh types and numerical schemes.

3.3. DNP drift in 1D channel

Before the investigation on the whole primary circuit, a preliminary test was carried out on a simplified geometry and the results were compared to 1D analytic solution to verify the DNP drift model for the neutronic solver [26].

This simplified geometry model is basically a $5\text{cm} \times 5\text{cm}$ rectangular pipe, consisting of 1.7m long pipe as the core region and 0.640058m section at the downstream representing the out-of-core primary circuit. The porosity in the core region is set to 0.224, while 1 in the other. And the cross sections set are taken from the core and upper plenum respectively. The mesh is 1 dimensional on x direction. The applied boundary conditions are listed in Table 3. The simplified geometry model with associated porosity is shown in Figure 9. It is worth noting that the neutron source for analytic solution is extracted from GeN-Foam solution using the variable “neutroSource”. And the units of neutron source and DNP concentrations are different, i.e. the former is per volume and the latter is per fuel volume. To maintain the consistency, one should always remember to consider the porosity at the regions where it has a non-zero value.

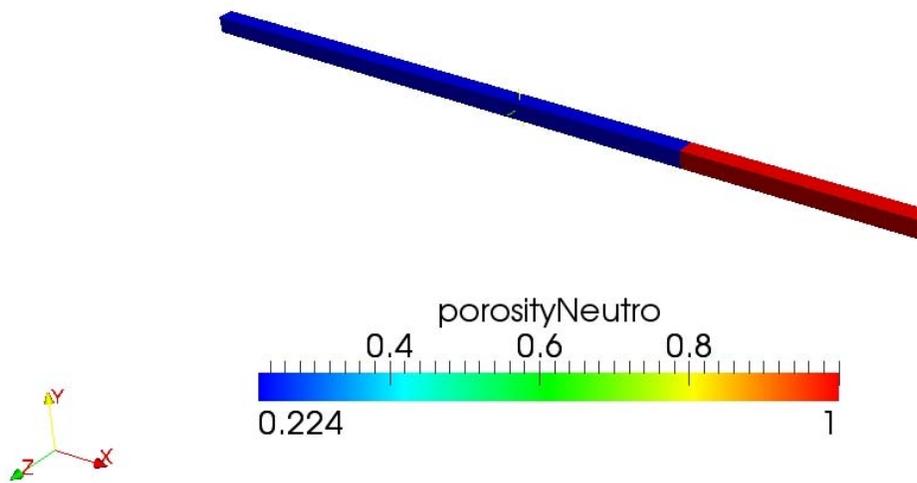


Figure 9: The rectangular pipe to represent the 1D channel

	Thermal-hydraulic		Neutronic	
	Velocity	Pressure	Precursors	Flux
Inlet	fixedValue: 0.04051m/s	zeroGradient	cyclic	0
Outlet	zeroGradient	fixedValue: 0	cyclic	0
Core walls	slip	zeroGradient	symmetry	symmetry
Other walls	slip	zeroGradient	zeroGradient	0

Table 2: Boundary conditions for the 1D channel

The first investigation was carried out on a static situation, where the velocity is zero and the decayed DNPs are supposed to be equal to the DNPs produced by fission. As one can see from the Figure 10, under such a circumstance, the concentrations of DNPs are inversely proportional to their decay constants, i.e. the faster the decay, the smaller the concentration. The comparison between the results and the analytic solutions is listed in Figure 11.

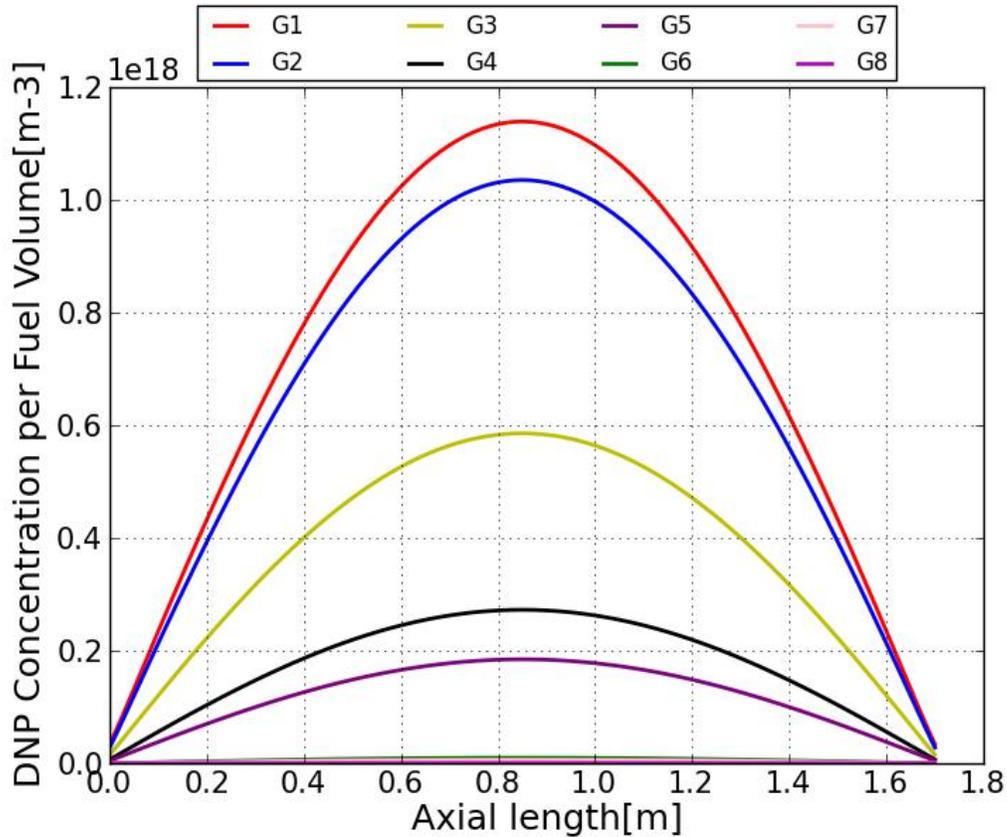


Figure 10: The DNP concentration per fuel volume at the core region when the salt is at static state for all the 8 groups

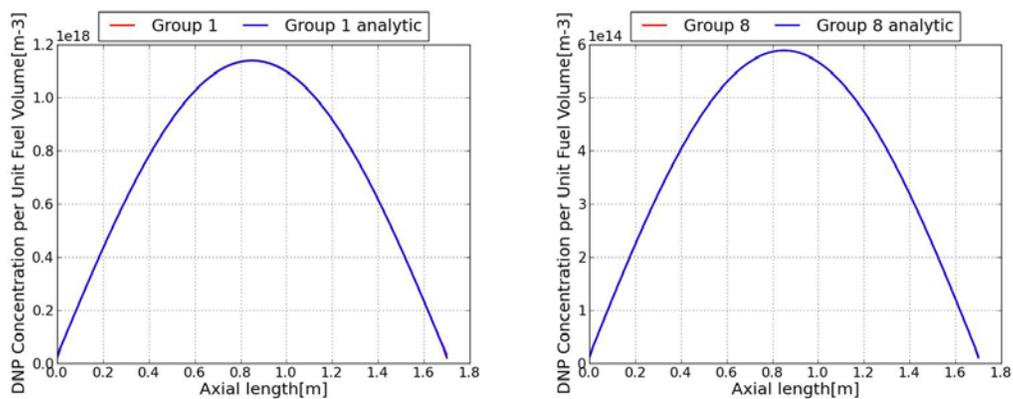


Figure 11: Comparison of DNP concentrations with analytic solutions when salt is static, illustrated by group 1 and group 8.

The results showed very good agreements with the analytic solutions. When the salt starts to move, the DNPs will be transported by moving fluid via both advection and turbulence diffusion, if turbulence exists and have significant effects. Here in this case, since the Reynolds number is below 2000, the flow is laminar, turbulence diffusion will be neglected. The DNP concentrations of each group over the whole system when salt is circulating are shown in Figure 12.

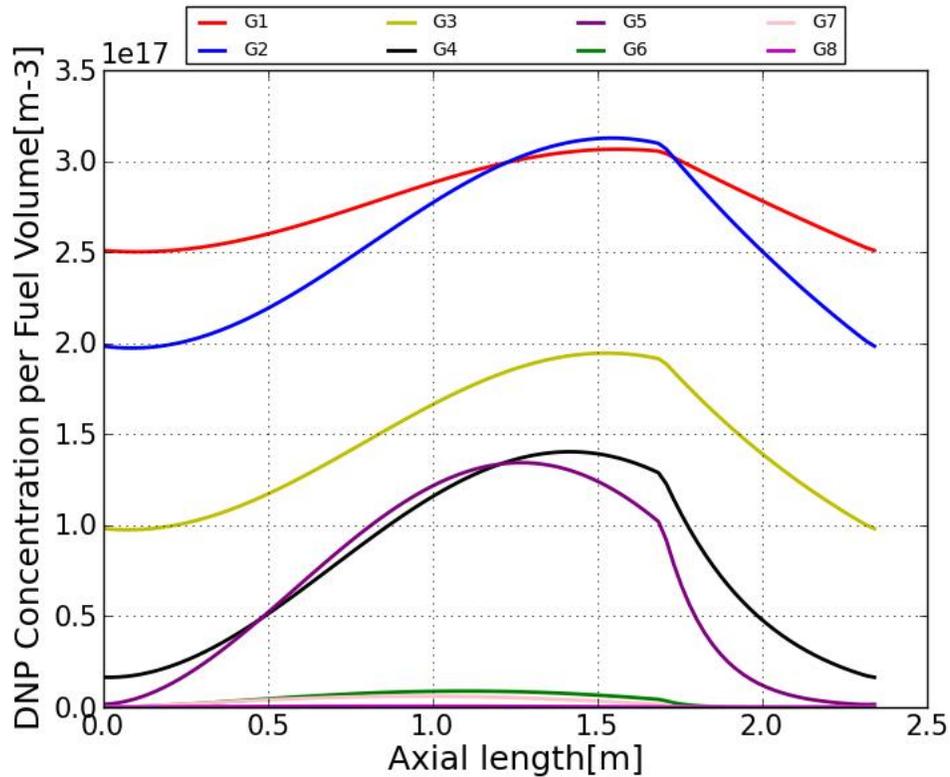


Figure 12: The DNP concentrations when salt is recirculating for all 8 groups

From the plots, it is clear that at the core region, the peaks of DNP concentrations are shifted towards the downstream part, which represents the upper plenum. A more clear illustration is shown in Figure 14, where group 4 is selected to show the drift of DNP. In the out-of-core primary circuit, starting from the core exit, DNP concentrations decrease, following an exponential law. For those fast decay groups, like group 7 and 8 in this case, their concentrations completely vanish during the transportation, however, the other groups with smaller decay constants will re-enter the core and contribute to the neutron generation there.

Again, the results are compared with analytic solutions and good agreements are observed. The plots are listed in Figure 13, to show the comparison of simulation results and analytic solutions when the salt is recirculating.

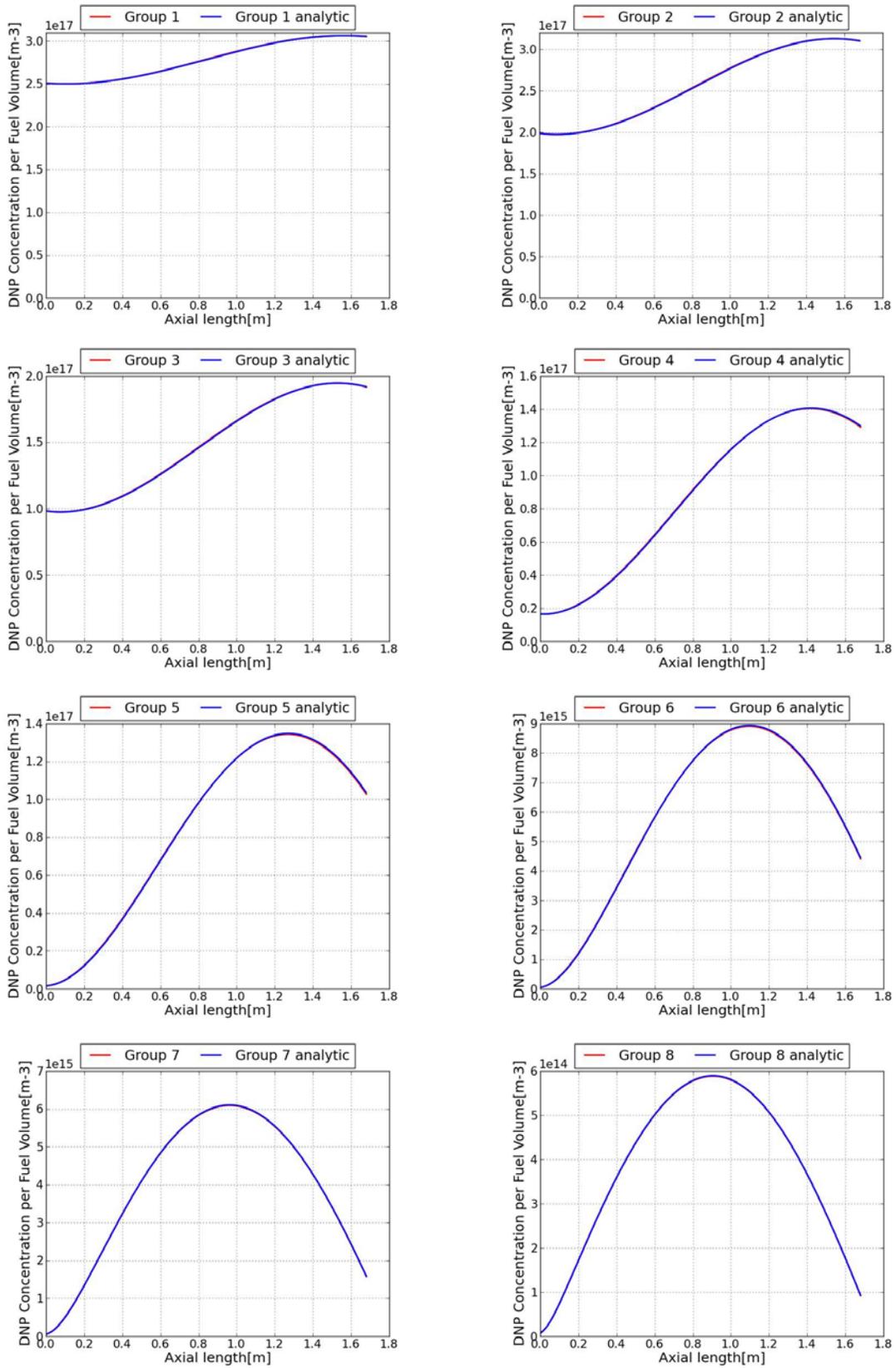


Figure 13: Comparison of DNP concentration with analytic solution when salt is recirculating, from group 1 to group 8

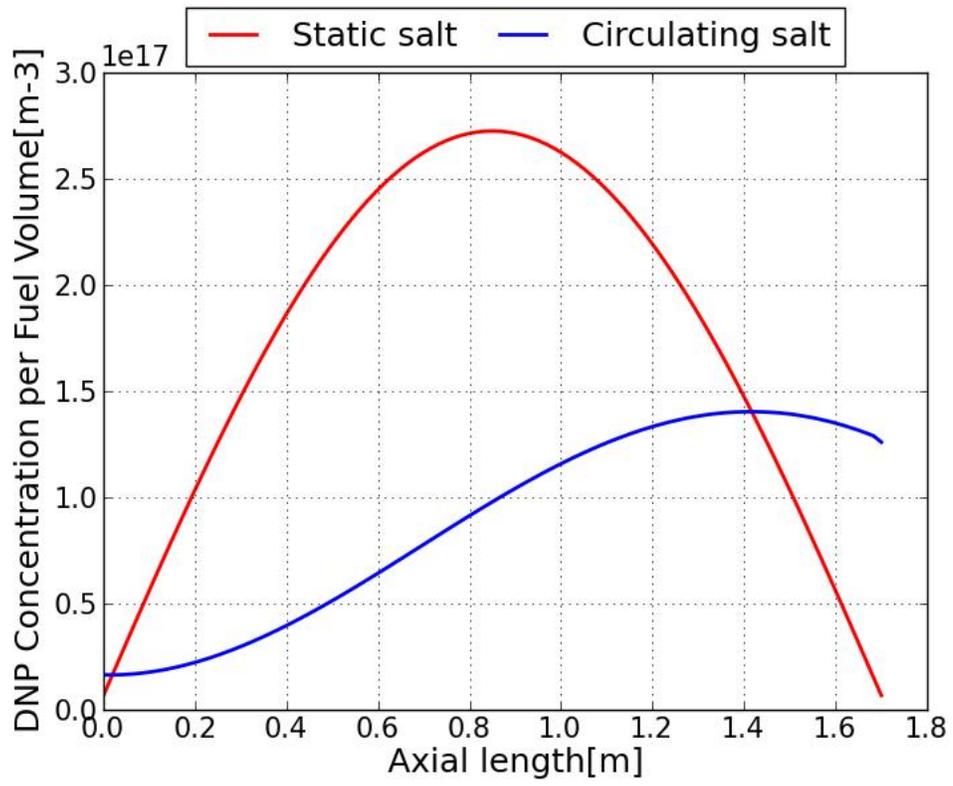


Figure 14: Comparison of DNP concentration of group 5 when the salt is static and recirculating

4. MSRE model development

4.1. Geometry Model

The geometry model of MSRE is mostly based on the dimension data explicitly provided in the report ORNL-TM-728 [27], however, the description of the crucial components is not completed, thus some data are measured from schematic graphs from the said report, and they are listed in Appendix B. To build the geometry model, the GOEM module in SALOME was used.

4.1.1. Reactor vessel and core

The reactor vessel is a tank made out of INOR-8, which is called Hastelloy-N nowadays. The volume encapsulated by the vessel wall can be divided into several zones, the core, and the lattice block below the core, the upper plenum, the lower plenum, the container wall, the distributor volute and the downcomer. The cutaway view of the reactor vessel is presented in Figure 4. And the data used to build the geometry is listed in Table 1.

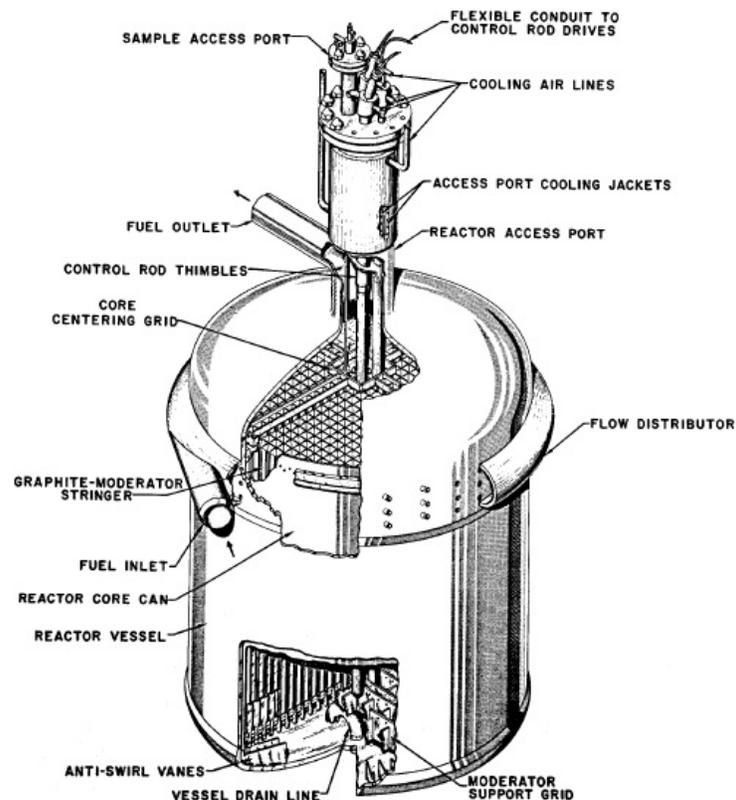


Figure 15: MSRE Reactor vessel [27]

Item	Dimension [cm]
Core	
Diameter	141
Height	171
Core Container Wall	
Inner Diameter	141
Outer Diameter	143
Height	174
Others	
Full Vessel Height	266
Distributor Inner Diameter	20

Table 3: Data used to build the reactor vessel

The salt enters the vessel from the fuel inlet with a temperature of 635°C, and will be evenly distributed in the volute thanks to the perforated wall which contains non-uniformly distributed holes to adjust the mass flow. However, the perforated wall is not included in the geometry model, and the evenly distributed flow can't be reproduced. Then the fuel follows a spiral path, flowing downwards in the downcomer, which provides extra cooling to the core container wall. Right before it enters the lower plenum, the flow will be guided by the anti-swirl vanes, thus loses its tangential momentum and flows towards the plenum center. Afterwards, the fuel flows upwards through the fuel channels surrounded by the graphite matrix. Finally, it exits the vessel from the fuel outlet with a temperature of 663°C.

It is worth noting that the complex geometry of the graphite matrix with the fuel channels in between, which is shown in Figure 16, is completely abandoned in the geometry model. This part will be modelled with a coarse mesh and porous medium approach, thus enormously reduce the level of problem's complexity.

Instead of curves, the surfaces of plena are modelled with segments for the sake of simplicity. And the core container wall is not explicitly involved in the model, but an empty space at its place. The view of the vessel geometry model is shown in Figure 17.

Last but not least, the control rods are not included in the geometry model for the sake of simplicity. The model can't fully reproduce the reality; however, this tradeoff has to be made due to the given time frame.

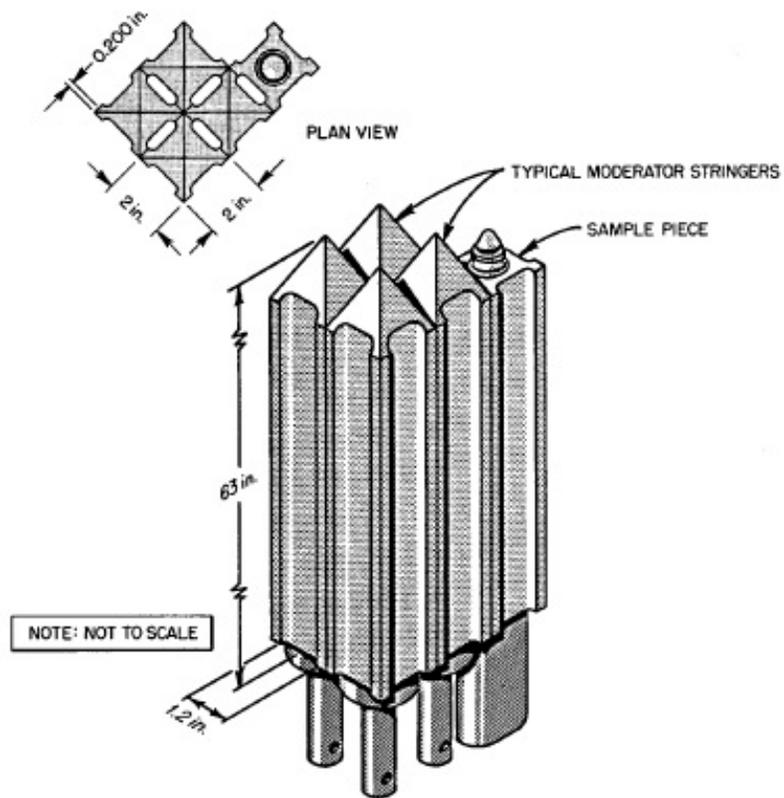


Figure 16: The graphite matrix and the fuel channels in between

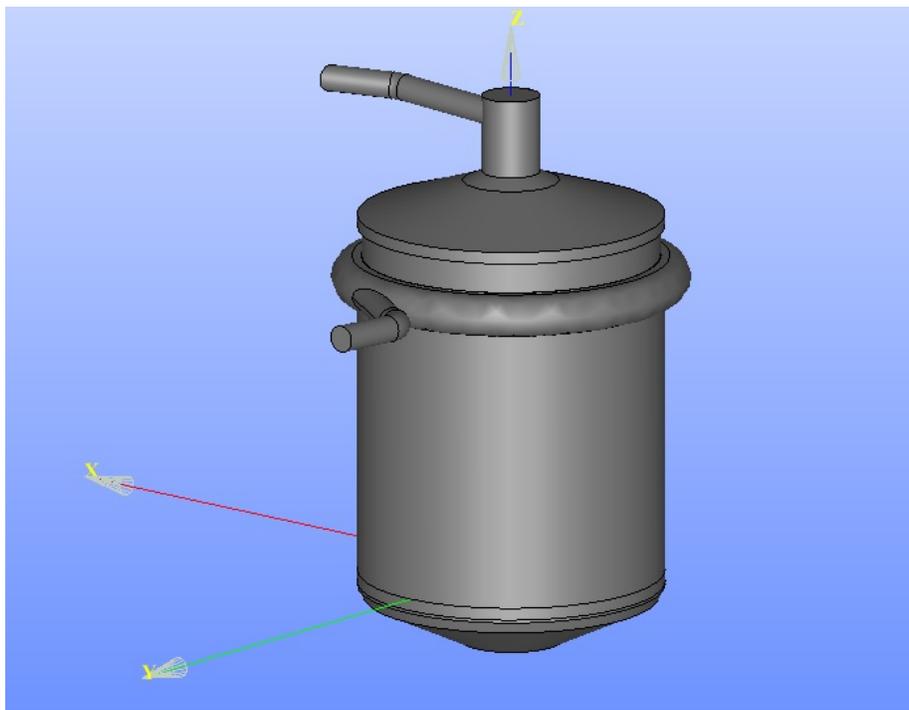


Figure 17: The geometry model for vessel generated by SALOME

4.1.2. Heat exchanger

The heat exchanger is a crucial component in the piping system. It consists of a shell and tubes in the shape of bended bundles. Primary fuel flows in the shell side while secondary in the tubes. In addition to that, baffles are placed at certain positions to achieve a cross flow, thus enhance the heat transfer. A cutaway view of the heat exchanger is shown in Figure 18.

In this work, since no energy transfer is taken into account, only the hydraulic properties have been involved. The whole component is modelled with a simple cylinder with a downward inclination of 3 degree from the primary fuel inlet to the outlet, and only the shell is involved in the model while the tubes are abandoned thanks to the porous medium approach. The data used to build the model is listed in Table 5. And the geometry model build in SALOME is shown in Figure 19.

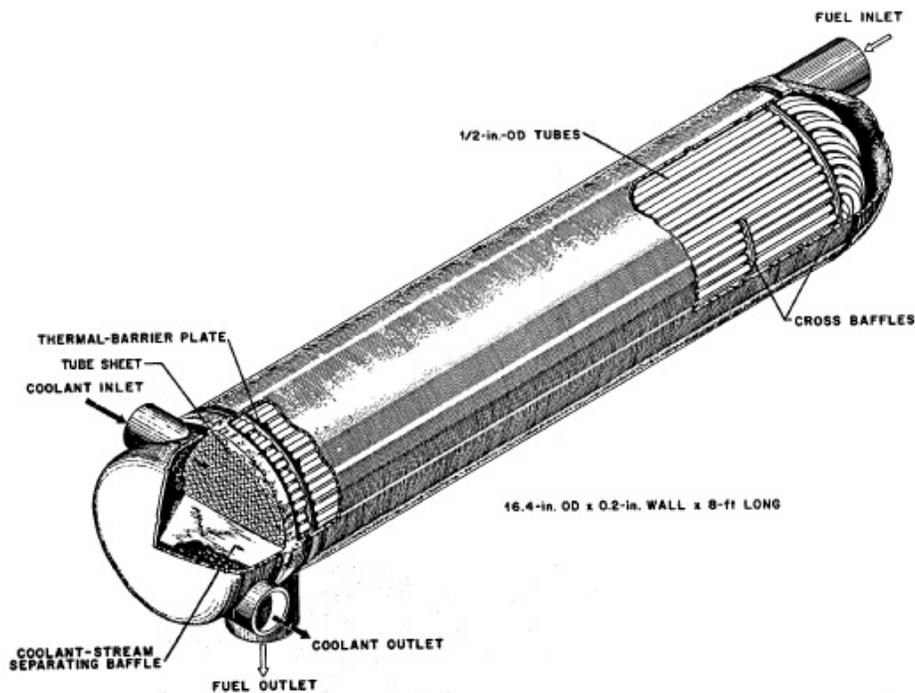


Figure 18: The cutaway view of the primary heat exchanger. [27]

	Value	Unit
Shell Diameter	40.64	cm
Shell Length	199.27	cm
Inclination Angle	3	degree

Table 4: The data used to build the heat exchanger geometry model.

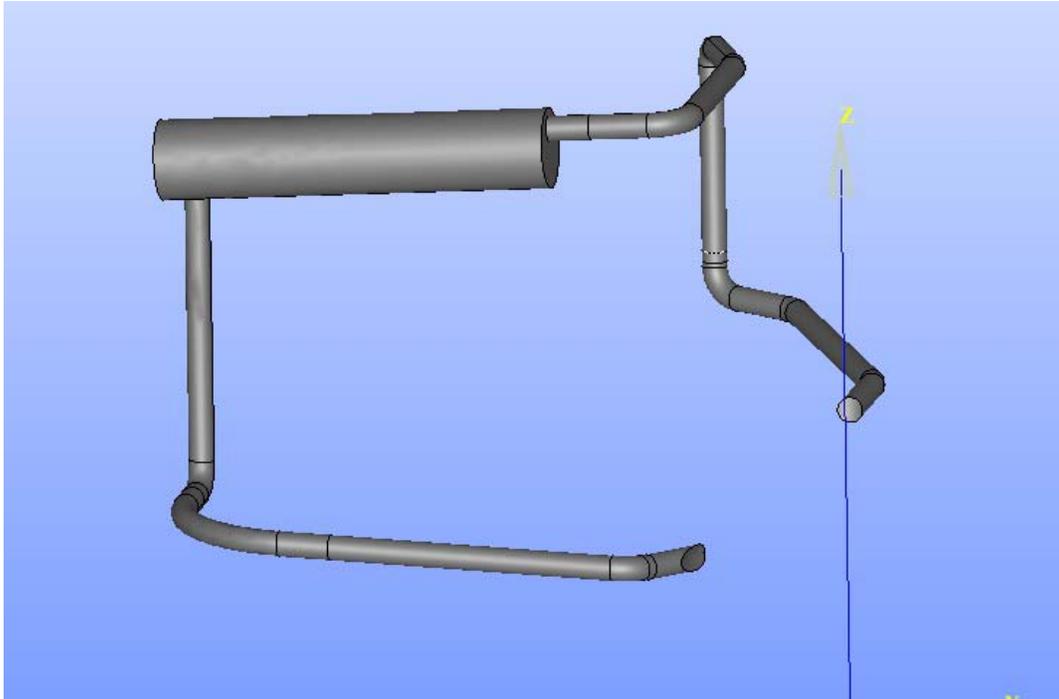


Figure 19: The geometry model for the piping system generated by SALOME

4.1.3. Pump

The pump is a quite complicated component in the real geometry. Again, thanks to the porous medium approach, it is not necessary to model the pump in details. A part of vertical pipe is selected to represent the pump, with correctly assigned momentum source, the mock-up pump is supposed to provide the required volume flow rate, which is $0.0757m^3 \cdot s^{-1}$ according to the design data. [27]

The dimension data used to build the geometry model for pump can be found in Table 6. And the pipe representing the pump is illustrated in Figure 20, with grey color and a label indicating its name “pump”.

	Value	Unit
Diameter	12.7	cm
Length	80	cm

Table 5: The data used to build the geometry model for the pump

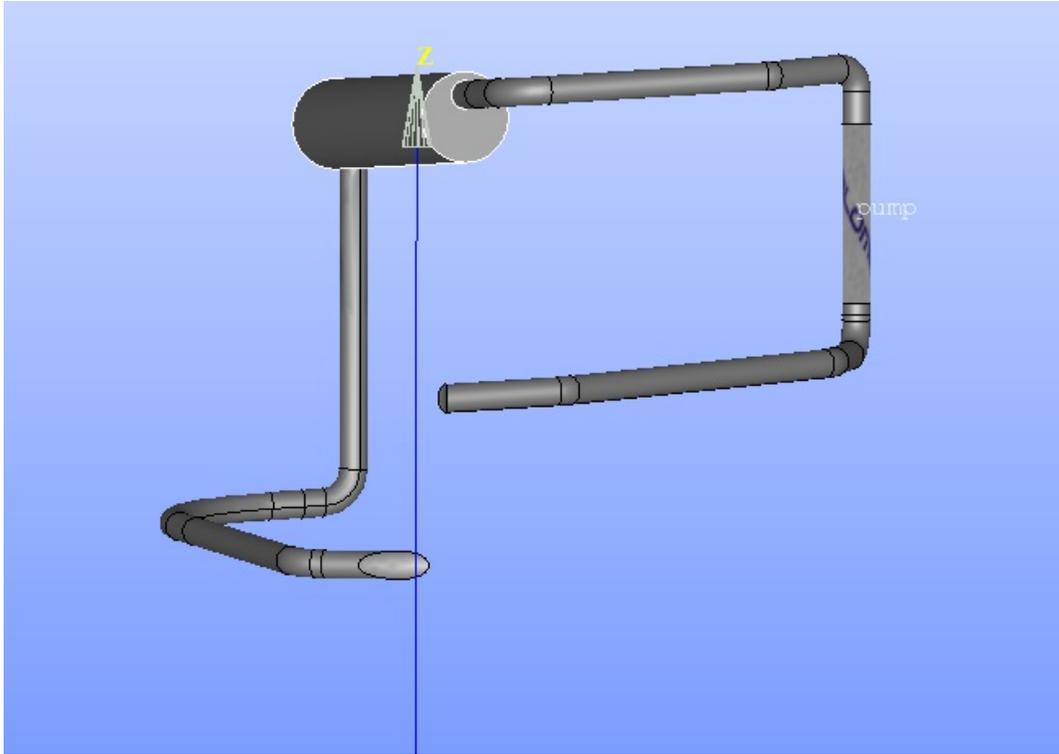


Figure 20: The location of the pump in the geometry model built by SALOME

4.1.4. Pipes

The pipes, i.e. the hot and cold legs, basically use the data measured from the graphs mentioned at the beginning of this chapter, as the description in the report is highly incomplete, except for the pipe diameter, which is 12.7cm [27]. The shape of the geometry model can be seen in either Figure 19, or Figure 20.

It is worth noting that all the pipes are downward inclined with 3 degree towards the core. This design is to ensure a complete draining when an accident happens.

4.1.5. Vanes

The 48 anti-swirl vanes in the lower plenum play an important role, i.e. to stop the tangential momentum of the fuel and guide it towards the plenum center, so that the flow is free of vortex. In the geometry model, these vanes are represented by zero thickness baffles. The view of the vanes in the lower plenum is shown in Figure 21. And the geometry model made with SALOME is shown in Figure 22.



Figure 21: The anti-swirl vanes in the lower plenum. [28]

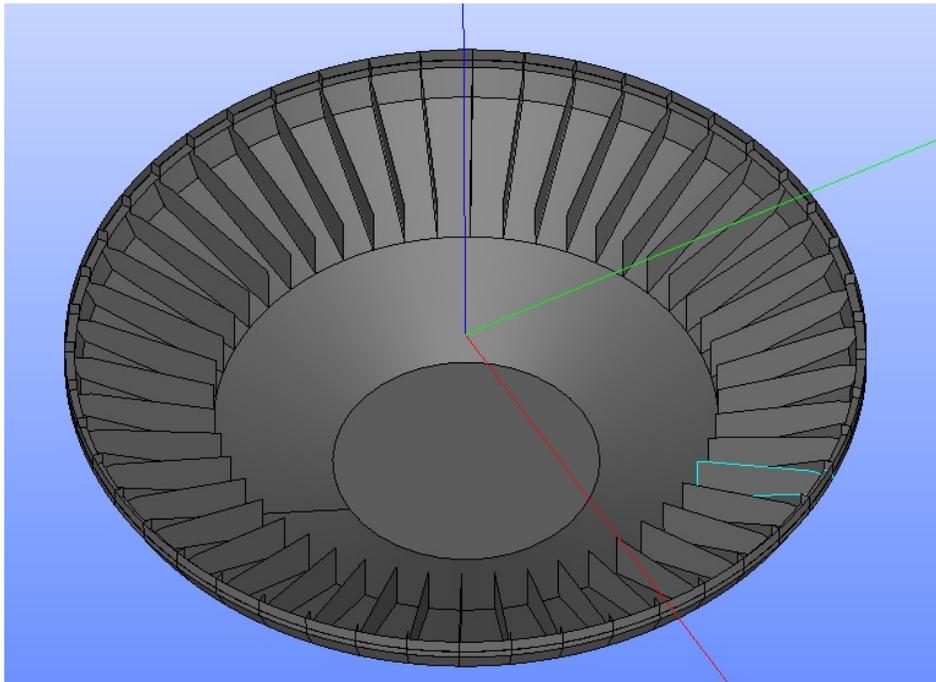


Figure 22: The geometry model of the vanes in the lower plenum

4.2. Mesh

Considering the complexity of the geometry model, tetrahedral mesh is preferred over a hexahedral one. With SALOME's SMESH module, a mesh containing 160,279 cells were automatically generated. A view of the mesh is shown in Figure 23.

The quality of the mesh is quite poor. With the OpenFOAM utility *checkMesh*, it was found that the maximum mesh non-orthogonality is 82.81292, and 313 faces are severely non-orthogonal, i.e. the deviation between the face vector and the center vector is over 70 degrees. However, no effort was put on refinement or looking for a better-quality mesh, as the idea was to use coarse mesh and get preliminary results for evaluations. And to get a fast convergence for flow field, non-orthogonal correctors were disabled, thus the obtained results might not be reliable.

The same mesh was used for both thermal-hydraulic and neutron solver, to eliminate possible errors when mapping fields between different meshes. Therefore, the results from neutronic solver are questionable, as it might not be a mesh-independent solution. In the further work, the above said issues should be taken into account, but not in the time frame given for this work.

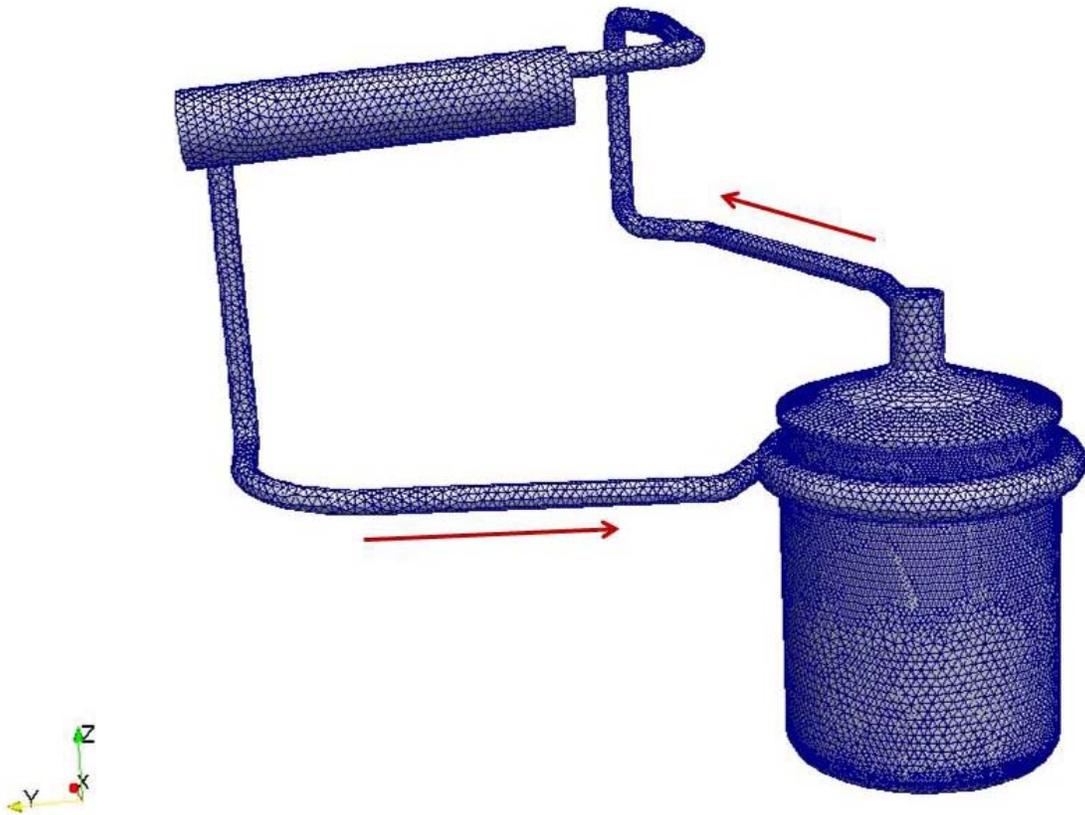


Figure 23: The mesh generated for MSRE geometry model with red arrows denoting the flow direction.

4.3. Boundary conditions

4.3.1. For thermal-hydraulic solver

All of the components are wrapped in a surface, which can be considered as a wall. However, the model is not assigned with a wall boundary condition concerning velocity, but the one named “slip”. In general, it basically imposes zeroGradient for a scalar, and fixedValue zero and zeroGradient for normal and tangential components respectively for a vector. For the variable velocity, it means there is a velocity jump between the near-wall cells and the wall. The reason for not using a proper wall boundary condition is that the model is built with coarse mesh and the near wall region is not properly refined, thus the wall boundary condition with such large cells doesn’t guarantee accurate result, and actually the wall friction effects should be represented by the pressure drop modelling for a porous zone.

Additionally, since there is no explicit constrains for the pressure in boundary conditions, it is necessary to provide the solver with a pressure reference. This is done via specifying the pressure reference point and reference value in the file `/system/neutroRegion/fvSolution`, under the case directory. Here the point right below the pump section is selected and the value is assigned with 149.588KPa , which is the recorded pressure at the pump suction for MSRE. [27]

4.3.2. For neutronic solver

With the new version of GeN-Foam, which is based on OpenFOAM 3.0.1, the user can select between Albedo and Dirichlet boundary conditions to prescribe neutron flux. And with an Albedo boundary condition, the user will be able to adjust the ratio between outgoing and incoming neutron flux, thus simulate a reflective boundary. However, the parallelization of this version is not compatible with the available Linux clusters, thus the calculation can be highly time consuming. Hence, in this work, the OpenFOAM 2.3.1 based GeN-Foam is employed, and one can only use a fixed value for the fluxes.

The fixed value boundary condition is selected for neutron flux on every face, and the value is set to 0, except for the core container can. As the core container can is not explicitly involved in the geometry model, a 0 flux boundary condition will make the wall impenetrable for neutrons, and prevent them from travelling from the core to the downcomer. To resolve this problem, the boundary condition can be set as *cyclicAMI*, which stands for cyclic Arbitrary Mesh Interface. It maps a patch to another, e.g. the inner container wall to the outer, no matter whether the patches are conformal or not [29].

4.4. Serpent model

In this work, the Serpent model is based on H. Kim's Master thesis [11], with some modification accounting for the simplifications and the facts that some dimensions were obtained from measurements. A vertical cutaway view of the model is shown in Figure 24.

As mentioned before, the control rods and graphite samples are removed, thus the core is completely filled with graphite matrix and fuel channels. The support plate beneath the core, which is 100% Hastelloy-N is also removed. Furthermore, the shielding and the gas in the chamber are not included either. The model is highly simplified, thus one can't rely on it to compare with experiment data. Nevertheless, it provides a reference for GeN-Foam's neutronic calculation.

Another significant modification is the outlet tube added at the top of the vessel. The surface of upper plenum is also adjusted from a sphere face to two cone faces while the lower plenum reserves its shape, as in upper plenum the flux is higher and the wall shape plays a more important role, it is therefore necessary to keep the same geometries there for both models. The difference concerning the lower plenum surface shape seems to be insignificant, as up to now, the check of distribution, e.g. axial power, shows good agreement there⁸.

It is worth noting that the salt composition in the lower plenum differs from the one used for core and the upper plenum, as the Hastelloy-N support grid occupies some volume. According to the ORNL report, 90.8% of fuel salt and 9.2% of Hastelloy-N was filled in the lower plenum [11]. The details of salt composition can be found in Appendix A.

4.5. Fuel properties

The thermal-hydraulic solver requires thermal-physical properties of the fluid to carry out calculation. In this work, since no thermal expansion is considered, the thermal-physical properties of the fuel are kept constant. Their values are taken from the ORNL report at 648.9°C, and are listed in Table 7.

	Value	Unit
Density ρ	2146	$kg \cdot m^{-3}$
Dynamic viscosity μ	0.008267877	$Pa \cdot s$
Specific heat c_p	1967.796	$J \cdot kg^{-1} \cdot K$
Heat conductivity k	5.54	$W \cdot m^{-1} \cdot K^{-1}$

Table 6: The thermal-physical properties adopted for this work.

⁸ See details in Section 5.2.

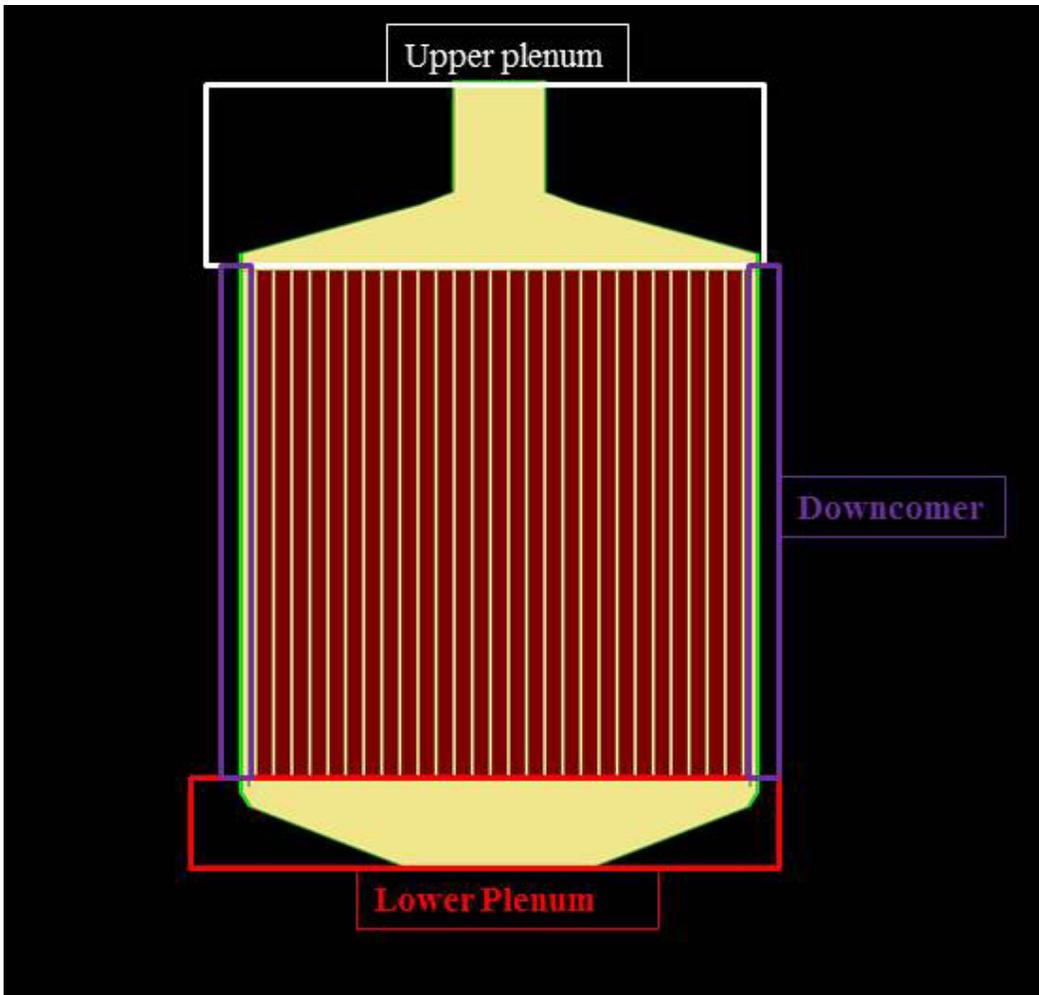


Figure 24: A vertical cutaway view of MSRE model in Serpent-2 and the arrangement of the zones

4.6. Other zone-associated properties

4.6.1. Thermal-hydraulic zones

As mentioned before, the mesh can be divided into several zones. Some, or all, of the zones can be considered as porous media and these porous media require extra parameters to feed into the empirical correlations, accounting for either momentum or heat source/sink. Since energy transfer is not involved, the heat-related parameters are not explained here. Here below, a brief introduction is given for the fluid-mechanic-related parameters.

The parameters can be classified into 3 categories, i.e. general properties (voidFraction, Reynolds number limits to distinguish laminar and turbulent flows, the hydraulic diameter), coefficients of empirical correlations for Darcy friction, and coefficients of empirical correlations for computing k and ϵ . The voidFraction is just another name of porosity. The Reynolds number limits for the end of laminar flow and start of turbulent flow are set to 1,000 and 2,300 respectively. As for coefficients for the $k - \epsilon$'s correlation, a standard

model [13] is adopted and the same values are assigned to all the porous zones. As for the Darcy friction's coefficients, Blasius correlation is used for this work. However, for the core region, the coefficients need to be adjusted in order to obtain the one-directional flow.

In the MSRE core, there are 1140 fuel channels surrounded by the graphite matrix, while in the presented model, it is represented by a cylinder without any inner structures and the flow can be any direction. To force a one-directional flow, normally two options are available, i.e. using longitudinal baffles and setting higher lateral friction coefficients. In this work, the latter one is selected, for it required fewer cells and makes the simulation lighter, nonetheless it might be a less stable method and one needs to be careful about the selected values. Several simulations were carried out to find the proper coefficients for the lateral friction, and the results are presented in Section 5.1.

To see the parameters set for each porous zone, please refer to Appendix C.

4.6.2. Neutronic zones

The mesh for neutronic solver is also split into several zones, which are not necessarily consistent with the thermal-hydraulic zones. Then each zone is assigned with a set of cross sections and coefficients for the multi-group diffusion equations, e.g. the diffusion coefficient, the effective fission cross sections, the delayed neutron fractions, etc. Figure 24. shows the zone arrangement for the in-vessel components. As for the out-of-core primary circuit, the parameters for the delayed neutrons are more important than the reaction cross sections, thus an approximation can be made that the hot leg and the heat exchanger take zone-averaged values from the upper plenum, while the cold leg takes them from the downcomer. A detailed list of those zone-averaged values can be found in Appendix D.

5. Results and discussions

5.1. Steady state results for within vessel

Using the above in-vessel geometry model, some calculations were carried out to investigate the behavior of core itself at a steady state.

5.1.1. The effect of the anti-swirl vanes

Before the flow enters the lower plenum, it encounters the anti-swirl vanes which are supposed to stop the tangential momentum moment of the flow and guide it towards the center of the plenum. It was found that the flow can be extremely unstable when the vanes are missing. The comparison of the flow field with and without the vanes is shown in Figure 25.

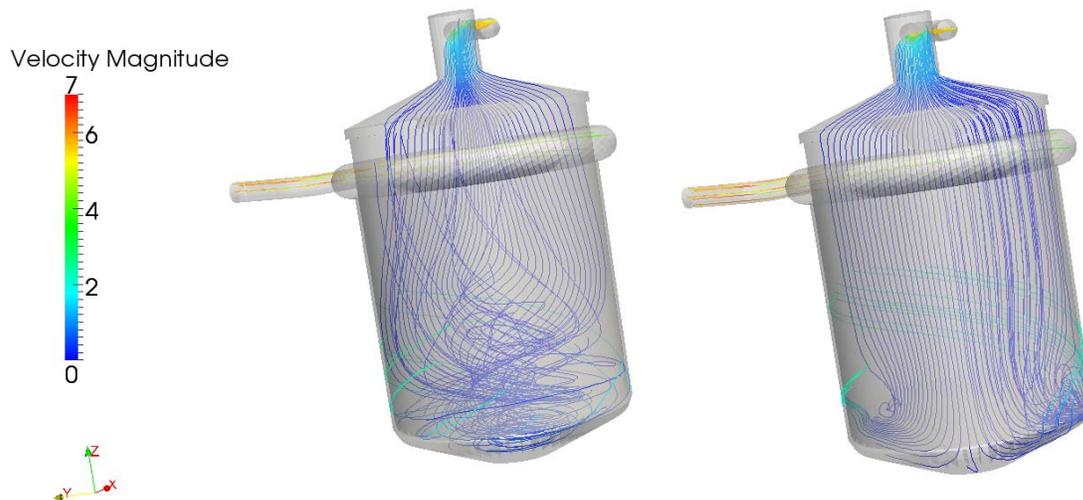


Figure 25: The comparison of the flow field in vessel with (left) and without (right) vanes.

5.1.2. Lateral friction coefficients

The theory of porous medium approach was introduced in Section 2.1.1, and the importance of setting appropriate parameters for the Darcy friction was emphasized in section 3.6.1, here in this section, a brief procedure of looking for the suitable parameters at core region is presented.

The involved geometry model is cut from the full primary circuit, and only the in-vessel components as well as short parts of cold and hot legs are taken into account. A constant velocity with a magnitude of $5.9758m \cdot s^{-1}$, which is computed from the rated volume flow rate and the cross sectional area of the pipe, is prescribed at the inlet.

To start with, the same friction coefficients were set to all three directions. The obtained flow field is shown in Figure 26. The geometry model is visually half transparent to enable a view inside the core. The spiral streamlines indicate the flow in the downcomer, while the straight ones are in the core. In this case, some vortex could be observed in the lower part of the core.

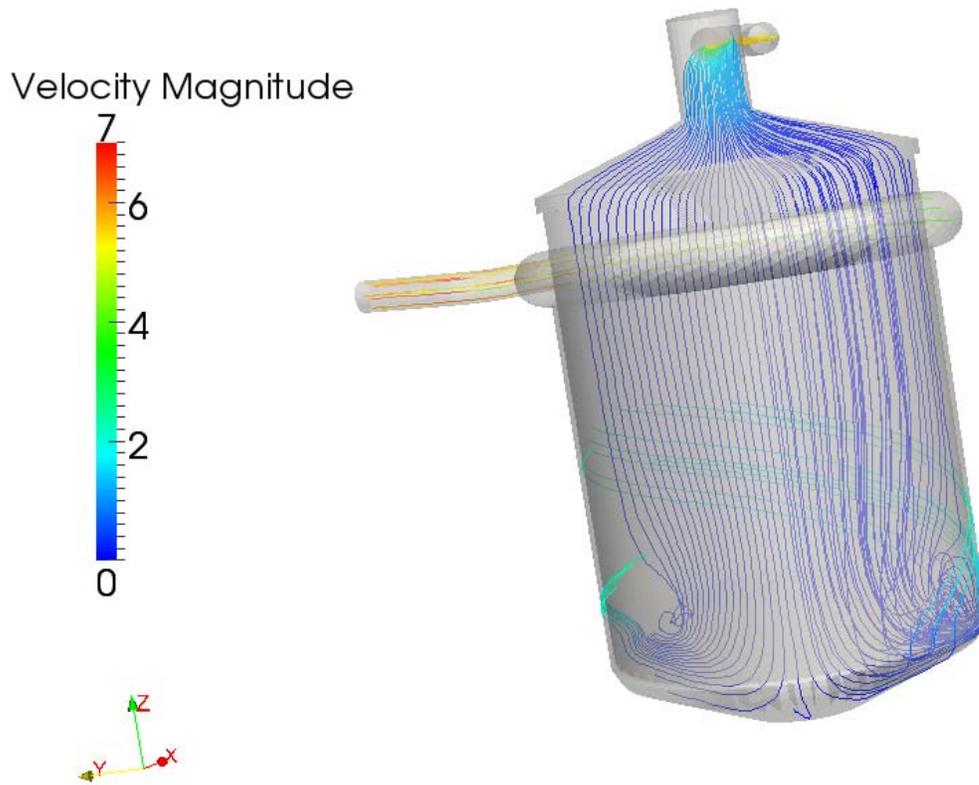


Figure 26: Flow field in the core with the same friction on all directions.

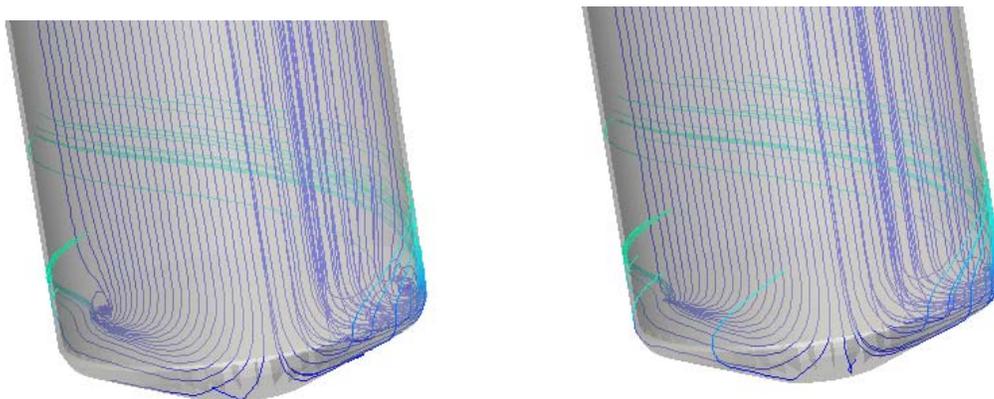


Figure 27: Flow field with lateral friction coefficients being 100 times higher (left) and 10,000 times higher (right).

Then the lateral⁹ friction coefficients were set to 100 times higher than the main stream direction. The streamlines in core were straightened but the vortex doesn't completely vanish. After increasing the lateral friction coefficients to 10,000 times higher, the vortex in core disappeared, the flow in core was confined to vertical direction only. The comparison of the two cases is shown in Figure 27.

5.1.3. Flow field

In this case, the porous medium treatment for turbulence was not activated, thus concerning k and ϵ the only reference information is the values prescribed at the inlet, which were computed based on the same correlation for the porous zone involving via turbulence intensity and turbulence length scale. The obtained turbulent viscosity is pretty high, which is unphysical, but it helps to stabilize the flow field, which allows a preliminary investigation. The streamlines while represents the salt flowing through the core center is shown in Figure 28.

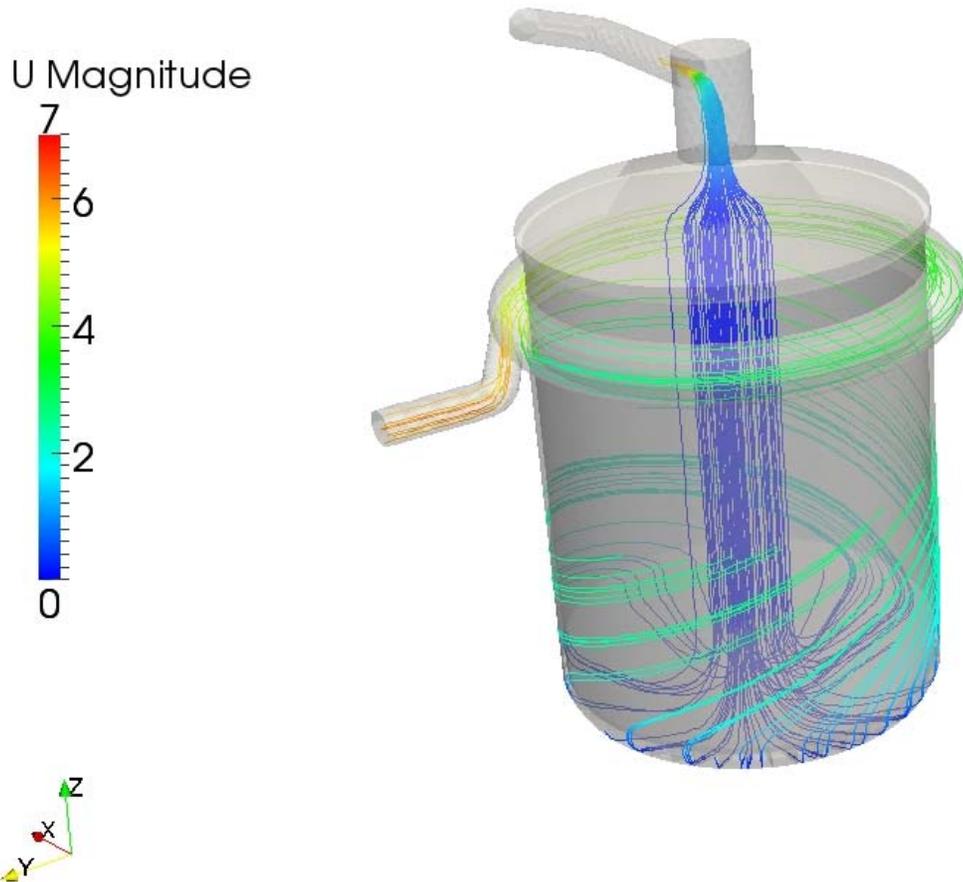


Figure 28: Selected central stream lines of the flow through core center and its extension in the downcomer.

⁹ When considering the z direction as main-stream direction, the lateral directions, which are perpendicular to the main-stream one, are actually the x and y directions.

It was observed that the salt entered the vessel from the inlet pipe and then circulated in the volute or entered the downcomer. Unlike in reality, where there is perforated wall to uniformly distribute the flow, the downward mass flow rate was not guaranteed to be the same everywhere. The flow followed a spiral path in the downcomer while its tangential momentum moment was gradually decreased. When it touched the anti-swirl vanes, the tangential momentum should be totally stopped, and the flow is then supposed to move towards center of lower plenum, guided by the vanes. After that, the flow went vertically upwards through the core, following a straight pattern thanks to the enlarged lateral friction coefficients which acts like a wall. When it exited the core, the flow was concentrated in the upper plenum and then finally left the vessel through the outlet.

The flow in the core is one directional and vertical. A plot of the profile is shown in Figure 29. As mentioned before, in GeN-Foam, there are two types of velocities available. One is the Darcy velocity solved in the equation, which takes into account the porosity, and keeps consistency across the porous zone interfaces, while the other is the real velocity and it is related to the quantities that have their units in the form of per fuel volume. In the plot below, it is the real velocity that is presented.

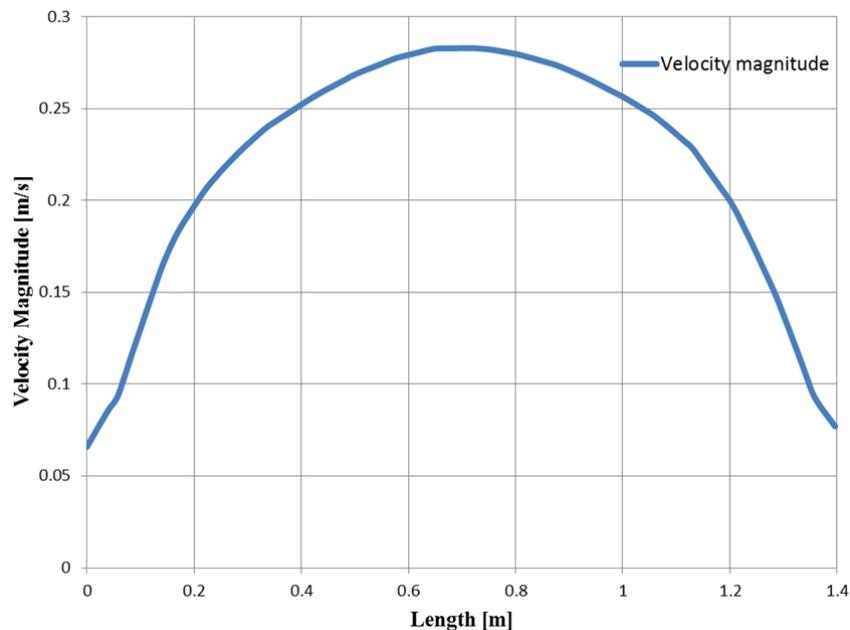


Figure 29: The velocity magnitude profile in the core.

The typical velocity through a fuel channel is about $0.22m/s$, with flow rate in the center being 20% higher than that at the periphery [27]. From the above plot, the average velocity in the core is $0.221652m/s$. In spite of the agreement concerning the average velocity, the profile is not symmetric due to the lack of perforated wall in the distributor, thus the flow is non-uniformly distributed.

5.1.4. Power distribution

The power distributions in the core at a steady state, when the salt is stagnant, were compared between Serpent-2 and GeN-Foam, to make sure the generated cross section sets are suitable for GeN-Foam model. The plots are shown in Figure 30. The axial power distribution shows

good agreement between the two results, although in the lower part, the power in Serpent is slightly higher than GeN-Foam. One should note that the mesh is still quite coarse, and a mesh-independent result was not reached.

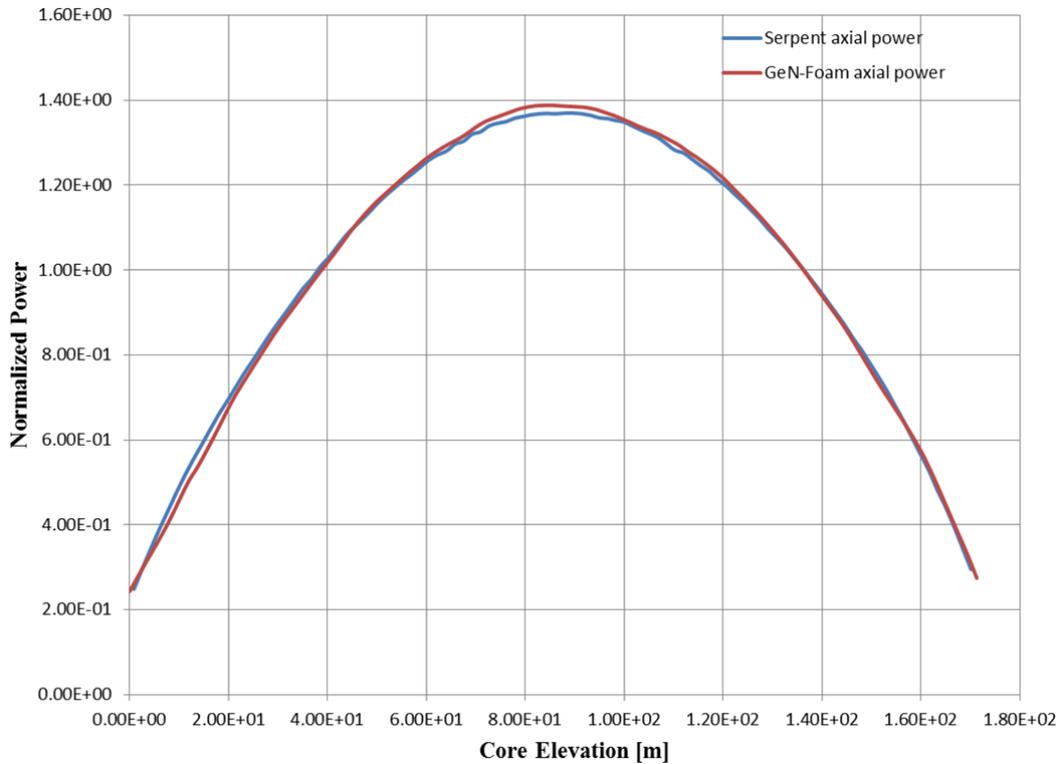


Figure 30: The axial power distribution.

	Serpent-2	GeN-Foam
k_{eff}	1.15799	1.11446

Table 7: The comparison of keff of Serpent-2 and GeN-Foam

The comparison of the k_{eff} values is shown in Table 8. For GeN-Foam, the above said *faceCorrected* and *leastSquares* schemes were applied. The discrepancy between the two results might arise from the method of generating cross sections, i.e. zone-averaged, and could also be due to the fact that mesh-independent result has not been reached.

5.1.5. DNP drift

When the salt is at static state, the DNPs follow the distribution of neutron sources, and decay locally to produce the delayed neutrons; while when the salt is moving, the DNPs will be transported through the primary circuit and decay everywhere, which affect the reactor's behavior when it comes to transient, thus it is good to know how the DNPs are transported at steady state which provides a reference for the further analysis. The comparison of DNP concentrations is shown in Figure 31. for static and moving salt cases. Group 1, group 5 and group 8 are selected as examples. Their decay constants vary from small to large. It was observed that the precursors were drifted towards the upper plenum for the slower decay groups, while for the fast decay, the DNP distribution was hardly influenced by the flow.

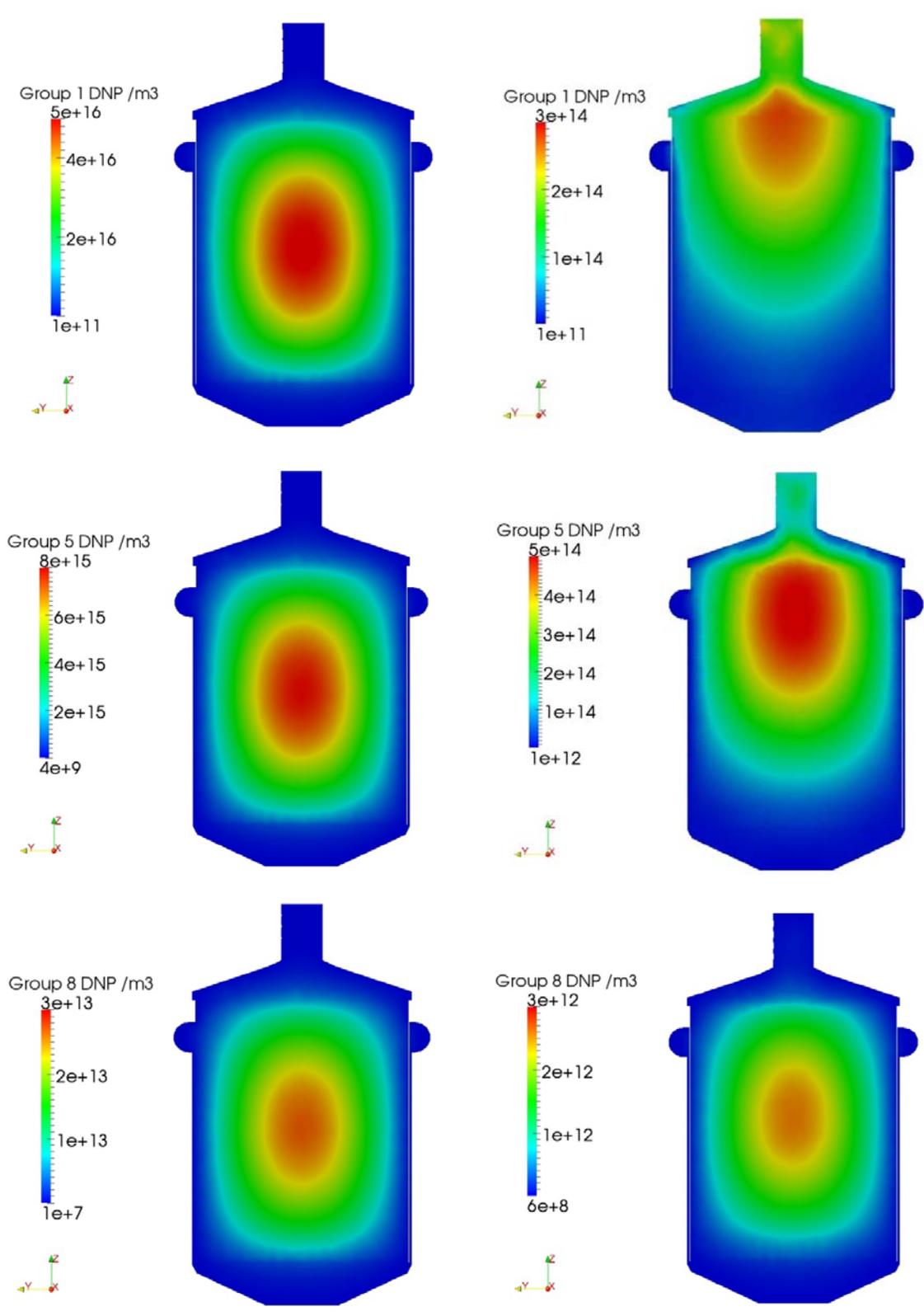


Figure 31: The concentration distribution of the delayed neutron precursor with static salt (left) and with moving salt (right)

5.2. Steady state results for full primary circuit

In this case, the pump was used to drive the flow. And to reach a stabilized flow field, the simulation requires over 15 hours on a 10 core Linux clusters, which is quite time consuming and makes the searching process not very effective. Furthermore, the *porouskEpsilon*¹⁰ method was activated in order to obtain a realistic flow field. However, with the default settings¹¹, some large vortex was observed in the core and lower plenum, even if the later friction coefficients were set to 10,000 higher, and the flow was severely unstable, leading to a blow-up for the simulation.

To stabilize the flow, it was decided to enhance the turbulence by artificially increasing the turbulent intensity constants. The default value was 0.16, and it was observed that if the value is too high, like 25.6, the simulation is easily to diverge. The value 12.8 was selected for this case, after several rounds of testing. It is worth noting that the solution obtained in this way became quite sensitive. The change of momentum source or turbulence intensity constants made on-the-fly or even at restart will lead to an unstable solution which will then diverge. One needs to start with 0 velocity initial condition after changing the said parameters. Taking into account all these limitations, a flow field was obtained when the momentum source was equivalent to $8e^5 Pa$, which is much higher than the expected value. But the compensation needs to be made as the maximum turbulent viscosity is over 100, and the flow requires large momentum source to push it. The streamlines of the obtained flow field in the lower plenum and the core is shown in Figure 32.

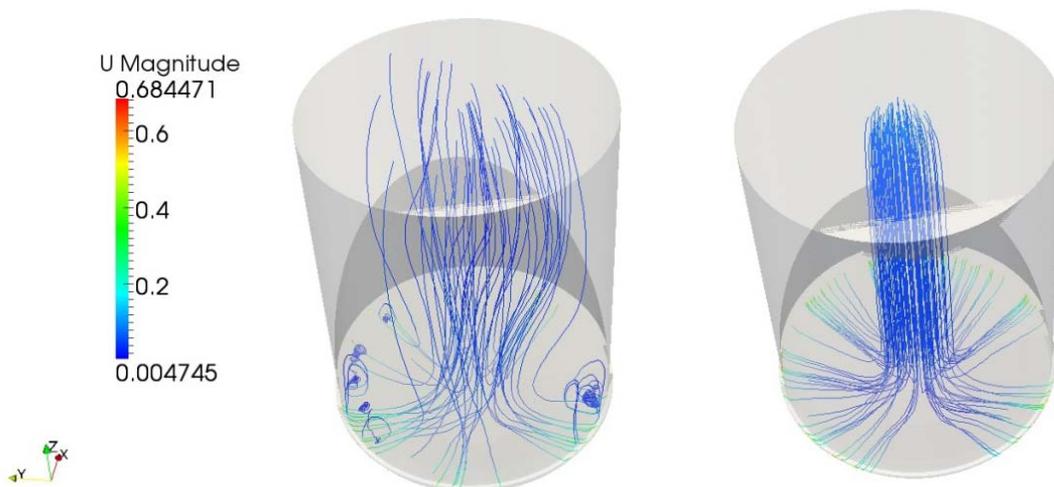


Figure 322: The comparison with the flow at core center and its extension in the lower plenum for the cases with k, ϵ being computed by porous media approach (left), and with only prescribed values at inlet.

Some vortex can be observed at the periphery of the interface of the core and the lower plenum, where the flow just entered the lower plenum and passed by the core container wall.

¹⁰ The GeN-Foam keyword for the special treatment of turbulence in porous zone described in Section 2.1.1.

¹¹ See Section 2.1.1.

Then another model was applied to get a stabilized flow. The pipe was cut on the cold leg. A velocity was prescribed at the cut face with the same value for the in-vessel tests, and the pump was disabled. In this case, the porousEpsilon was not used and the only core and heat exchanger were considered as porous media. The only reference of k, ϵ is the value specified at the inlet. In this way, an unphysical but stabilized flow field was achieved. The streamlines are shown in Figure 32. And a comparison of the streamlines at core center to the case where k, ϵ were computed with porous medium approach is shown in Figure 33.



Figure 333: The streamlines along the primary circuit.

No vortex was observed as the turbulent viscosity is as high as 20,000, which makes the flow field extremely unphysical. And the flow was diverted to two directions after it entered the distributor, which differs from reality. However, the attempt to calculate DNP drift based on this flow field also failed. It is likely that DNP drift solver doesn't strongly rely on a stable flow field, but a reasonable and physical one. Moreover, the schemes for neutronic solver should probably also be updated. Both corrected and uncorrected schemes were tested for this case, but none of them avoids the divergence. The reason is probably not in the gradient but in other terms. However, there was no time to carry out further tests.

6. Summary and future work

Molten salt reactor is an advanced reactor design which has many advantages concerning safety, economy and sustainability. One special feature that distinguishes it most from the other designs is the liquid fuel and the resulting drift of delayed neutron precursors. The analysis of the DNP drift requires tightly coupled multi-physics simulations, thus a modern coupling tool GeN-Foam was developed in PSI-FAST group and will be used to analyze the MSR transients. This work aimed at developing a model for MSRE and validating the DNP drift model for GeN-Foam, nonetheless it is just a start point which presents some abilities of GeN-Foam and points out potential directions for the further works.

Some preliminary tests were carried out to understand the capabilities of GeN-Foam and its user environment. The first was a comparison between the different methods of generating cross section sets for GeN-Foam using Serpent-2. For a simple cuboid geometry, the cross sections were generated on infinite lattice and as zone-averaged over the core. The GeN-Foam's results have better agreement with Serpent's results for the infinite lattice cross sections. However, a fully prove was not conducted, and to keep consistency with previous study in FAST group, the conventional method of using zone-averaged cross sections was adopted for this work.

The second test was related to the neutronics solver meshing and showed that with a tetrahedral mesh and proper non-orthogonality correctors, i.e. *faceCorrected* for surface normal gradient and *leastSquares* for the gradient, the result is comparable with the one obtained from a hexahedral mesh. Therefore a tetrahedral mesh was created for this work in order to spend least time on meshing.

Another preliminary test was the DNP drift in a 1D channel. The distribution of precursors computed by GeN-Foam was compared to an analytic result, and a good agreement between the solutions was observed.

A geometry model was built up using the open source CAD tool SALOME. Due to the fact that the ORNL reports don't provide descriptions for the geometry with full details, some dimension data was measured from the design graphs, which may involve human errors. Furthermore, the Serpent model for this work was adopted from H. Kim's Master thesis, and some modifications were necessary to account for the measurements as well as some other simplifications. It was also found that SALOME 7.7.1 has stability issue. The GUI tends to crash when the geometry is complicated and it requires a lot of memory to process, while TUI behaves more stable. However, it is more convenient to do an on-the-fly change via TUI, as the dimensions of a component can be directly modified in the script while for the GUI, the needs to delete the component and then build it from scratch.

The mesh was also generated by SALOME with a built-in method to automatically create a tetrahedral mesh. The quality of the mesh was quite poor. However, the refinement of mesh to look for a mesh-independent solution was not considered since the idea of this work was to test the coarse mesh approach of GeN-Foam. And actually the mesh was generated as coarse as possible to accelerate the computation speed. However, the speed with the available mesh is not very promising. It takes over 15 hours to reach a stabilized solution for thermal-hydraulic solver, running on 10 cores Linux cluster. The long running time is due to the small time step which is constrained by the courant number limitation on a few anomalous cells, as the mesh quality is poor.

Some parameters were investigated to ensure that GeN-Foam is doing proper work and the cross sections fed to it are reasonable values, e.g. the comparison of axial power distributions. And the results showed good agreement.

In order to get a stabilized and physical solution for the flow field, all the components of the primary circuit were considered as porous media, except for the upper and lower plena, and the *porouskEpsilon* method with default settings was activated. However, big vortex was observed in the core and caused simulation to finally diverge. Then the intention was shifted to get a stabilized flow field. This was done by increasing the turbulence viscosity, no matter whether it is physical or not. In this way, the flow field can be much more stabilized. However, the neutronic solver for DNP drift still did not converge on this flow field.

Accordingly, several improvements for both thermal-hydraulic and neutron models may be considered as future steps:

- **Geometry Model.** The geometry model requires further improvements to involve more details, like the perforated wall in the flow distributor. It may be responsible for the difficulty to get a reasonable flow field. In reality, the flow was uniformly distributed after exiting the distributor, in this work, however, the flow was non uniform, thus it may cause vortex due to the mixing in the lower plenum. It requires further analysis. Moreover, the method adopted in this work to confine a one-directional flow, i.e. artificially increasing the lateral friction coefficients, is not a stable approach. A conventional and more numerical stable method is to use longitudinal baffles.
- **Mesh.** A single bad cell is enough to violate the simulation, thus the mesh plays a crucial role. The application of porous medium approach on a coarse mesh frees the user from refining the mesh around the complex geometry. However, it is likely that the current mesh is too coarse to account for the large scale flow movements. Mesh refinement needs to be taken into account for the future work, at least in the upper and lower plena where it is not trivial to properly apply porous medium approach.
- **Discretization Schemes.** The reason is for the crash of neutronic solver when DNP drift is activated is not clear. It might be due to the unphysical flow field, but it could also be related to the discretization schemes applied for the neutronic solver as well as the mesh. As a matter of fact, when a scheme is chosen to take into account the non-orthogonality correctors, the solution can be unstable which leads to crash. However,

if the scheme is uncorrected, the results, especially concerning neutronic solver, can be completely unreliable. Therefore, it is probably necessary in the future work to find a suitable and efficient scheme.

Even though, the simulation of DNP drift through the whole primary circuit was not converging, the GeN-Foam's potential for MSR application was approved. And after the necessary tuning and validation, it will be an important instrument for MSR safety analysis.

Bibliography

- [1] R. Ali, "Environmental impacts of renewable energy," in *Electric Renewable Energy Systems*, London, Joe Hayton, 2016, pp. 519-542.
- [2] B. Hombourger, "Parametric Lattice Study for Conception of a Molten Salt Reactor in Closed Thorium Fuel Cycle," 2013.
- [3] O. Beneš and R. Konings, "Thermodynamic properties and phase diagrams of fluoride salts for nuclear applications," *Journal of Fluorine Chemistry*, vol. 130, no. 1, pp. 22-29, 2009.
- [4] C. Fiorina, D. Lathouwers, M. Aufiero, A. Cammi, C. Guerrieri, J. L. Koosterman, L. Luzzi and M. E. Ricotti, "Modelling and analysis of the MSFR transient behaviour," *Annals of Nuclear Energy*, vol. 64, pp. 485-498, 2014.
- [5] E. Merle-Lucotte, D. Heuer, M. Allibert, M. Brovchenko, N. Capellan and V. Ghetta, "Launching the thorium fuel cycle with the Molten Salt Fast Reactor," in *In: Proc.Int. Conf. ICAPP*, Nice, France, 2011.
- [6] N. Zweibaum, G. Cao, A. T. Cisneros, B. Kelleher, M. R. Laufer, R. O. Scarlat, J. E. Seifried, M. H. Anderson, C. W. Forsberg, E. Greenspan, L.-W. Hu, P. F. Peterson and K. Sridharan, "Phenomenology, methods and experimental program for fluoride-salt-cooled, high-temperature reactors (FHRs)," *Progress in Nuclear Energy*, vol. 77, pp. 390-405, 2014.
- [7] D. E. Holcomb, S. M. Cetiner, G. F. Flanagan, F. J. Peretz and J. Graydon L. Yoder, "An Analysis of Testing Requirements for Fluoride Salt-Cooled High Temperature Reactor Components," the U.S. Department of Energy(DOE), 2009.
- [8] J. Serp, M. Allibert, O. Beneš, S. Delpech, O. Feynberg, V. Ghetta, D. Heuer, D. Holcomb, V. Ignatiev, J. L. Koosterman, L. Luzzi, E. Merle-Lucotte, J. Uhlíř, R. Yoshioka and D. Zhimin, "The molten salt reactor (MSR) in generation IV: Overview and perspectives," *Progress in Nuclear Energy*, vol. 77, pp. 308-319, 2014.
- [9] J. Krepel, B. Hombourger, C. Fiorina, K. Mikityuk, U. Rohde, S. Kliem and A. Pautz, "Fuel cycle advantages and dynamics features of liquid fueled MSR," *Annals of Nuclear Energy*, vol. 64, pp. 380-397, 2014.

- [10] C. W. Forsberg, "Molten Salt Reactors," in *The Americas Nuclear Energy Symposium*, Miami, 2002.
- [11] H. Kim, "Static and transient analysis of Molten Salt Reactor Experiment using SERPENT-2/TRACE/PARCS codes," 2015.
- [12] M. Bonet, "Thermal-hydraulics verification of a coarse-mesh OpenFOAM-based solver for a Sodium Fast Reactor".
- [13] 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, 2015.
- [14] G. A. Roth and F. Aydogan, "Theory and implementation of nuclear safety system codes - Part II: System code closure relations, validation, and limitations," *Progress in Nuclear Energy*, vol. 76, pp. 55-72, 2014.
- [15] D. Bertolotto, A. Manera, S. Frey, H.-M. Prasser and R. Chawla, "Single-phase mixing studies by means of a directly coupled CFD/system-code tool," *Annals of Nuclear Energy*, pp. Volume 36, Issue 3, Pages 310-316, 2009.
- [16] V. González-Albuixech, G. Qian, M. Sharabi, M. Niffenegger, B. Niceno and N. Lafferty, "Coupled RELAP5, 3D CFD and FEM analysis of postulated cracks in RPVs subjected to PTS loading," *Nuclear Engineering and Design*, vol. 297, pp. 111-122, 2016.
- [17] I. D. Clifford, "A Hybrid Coarse and Fine Mesh Solution Method for Prismatic high Temperature Gas-cooled Reactor Thermal-fluid Analysis," 2013.
- [18] R. Saurel and R. Abgrall, "A Multiphase Model for Compressible Flows with Interfaces, Shocks, Detonation Waves and Cavitation," *Journal of Fluid Mechanics*, vol. 150, no. 2, pp. 425-467, 1999.
- [19] B. Launder and B.I.Sharma, "Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc," *Letters in Heat and Mass Transfer*, vol. 1, no. 2, pp. 131-138, 1974.
- [20] W. Stacey, *Nuclear Reactor Physics*, WILEY-VCH Verlag GmbH & Co.KGAA., 2007.
- [21] A. Waltar, D. Todd and P. Tsvetkov, *Fast Spectrum Reactors*, Springer, 2012.
- [22] "SALOME," OPEN CASCADE, [Online]. Available: <http://www.salome-platform.org/user-section/about>. [Accessed 15 July 2016].
- [23] J. Leppänen, *Serpent - a Continuous-energy Monte Carlo Reactor Physics Burnup Calculation Code - User's Manual*, 2015.
- [24] J. Guerrero, "OpenFOAM Introductory Course Slides," Wolf Dynamics, [Online]. Available: <http://www.wolfdynamics.com/images/pdf/module3.pdf>. [Accessed 5 August 2016].

[25] B. Soubelet, "Time efficient fluid dynamics analysis of sodium fast reactor wire wrapped rod bundles," Paul Sherrer Institut, Villigen, 2015.

[26] J. Krepel, "Dynamics of Molten Salt Reactors," Dresden, 2006.

[27] R.C.Robertson, "MSRE DESIGN AND OPERATIONS REPORT PART I DESCRIPTION OF REACTOR DESIGN," OAK RIDGE NATIONAL LABORATORY, 1965.

[28] "MOLTEN-SALT REACTOR PROGRAM SEMIANNUAL PROGRESS REPORT FOR PERIOD ENDING JULY 31, 1964, ORNL-3768," Oak Ridge National Laboratory, 1964.

[29] C. Greenshields, "OpenFOAM User Guide: 5.2 Boundaries," CFD Direct Ltd, 2 March 2015. [Online]. Available: <http://cfd.direct/openfoam/user-guide/boundaries/>. [Accessed 9 August 2016].

List of Tables

TABLE 1: COMPARISON OF K _{EFF} VALUES, XS MEANS CROSS SECTIONS	19
TABLE 3: BOUNDARY CONDITIONS FOR THE 1D CHANNEL	22
TABLE 4: DATA USED TO BUILD THE REACTOR VESSEL.....	28
TABLE 5: THE DATA USED TO BUILD THE HEAT EXCHANGER GEOMETRY MODEL.....	30
TABLE 6: THE DATA USED TO BUILD THE GEOMETRY MODEL FOR THE PUMP.....	31
TABLE 7: THE THERMAL-PYSICAL PROPERTIES ADOPTED FOR THIS WORK.....	36
TABLE 8: THE COMPARISON OF KEFF OF SERPENT-2 AND GEN-FOAM	43

List of Figures

FIGURE 1: THE CONCEPTUAL SCHEME OF MOLTEN SALT REACTOR PROPOSED BY GIF.....	4
FIGURE 2: THE FLOW DIAGRAM OF MSRE	5
FIGURE 3: THE SCHEMATIC ILLUSTRATIONS MSFR (LEFT) AND MOSART (RIGHT).....	7
FIGURE 4: CONCEPTUAL DRAWING OF A PEBBLE BED FHR AND POWER GENERATION CYCLE. [7].....	8
FIGURE 5: SIMPLIFIED GEN-FOAM COUPLING SCHEME FOR THIS WORK.....	17
FIGURE 6: SCHEMATIC DRAWING OF MESH NON-ORTHOGONALITY. [23]	20
FIGURE 7: THE LAYOUT OF TETRAHEDRAL MESH (LEFT) AND HEXAHEDRAL MESH (RIGHT) FOR A CUBOID GEOMETRY.	21
FIGURE 8: THE COMPARISON OF SENSITIVITY TO MESH TYPES AND NUMERICAL SCHEMES. ...	21
FIGURE 9: THE RECTANGULAR PIPE TO REPRESENT THE 1D CHANNEL.....	22
FIGURE 10: THE DNP CONCENTRATION PER FUEL VOLUME AT THE CORE REGION WHEN THE SALT IS AT STATIC STATE FOR ALL THE 8 GROUPS.....	23
FIGURE 11: COMPARISON OF DNP CONCENTRATIONS WITH ANALYTIC SOLUTIONS WHEN SALT IS STATIC, FROM GROUP 1 TO GROUP 8.....	23
FIGURE 12: THE DNP CONCENTRATIONS WHEN SALT IS RECIRCULATING FOR ALL 8 GROUPS	24
FIGURE 13: COMPARISON OF DNP CONCENTRATION OF GROUP 5 WHEN THE SALT IS STATIC AND RECIRCULATING	26
FIGURE 14: COMPARISON OF DNP CONCENTRATION WITH ANALYTIC SOLUTION WHEN SALT IS RECIRCULATING, FROM GROUP 1 TO GROUP 8.....	25

FIGURE 15: MSRE REACTOR VESSEL[15] 27

FIGURE 16: THE GRAPHITE MATRIX AND THE FUEL CHANNELS IN BETWEEN [26]..... 29

FIGURE 17: THE GEOMETRY MODEL FOR VESSEL GENEARTED BY SALOME..... 29

FIGURE 18: THE CUTAWAY VIEW OF THE PRIMARY HEAT EXCHANGER. [26]..... 30

FIGURE 19: THE GEOMETRY MODEL FOR THE PIPEING SYSTEM GENERATED BY SALOME 31

FIGURE 20: THE LOCATION OF THE PUMP IN THE GEOMETRY MODEL BUILT BY SALOME 32

FIGURE 21: THE ANTI-SWIRL VANES IN THE LOWER PLENUM. [27]..... 33

FIGURE 22: THE GEOMETRY MODEL OF THE VANES IN THE LOWER PLENUM 33

FIGURE 23: THE MESH GENERATED FOR MSRE GEOMETRY MODEL WITH RED ARROWS
DENOTING THE FLOW DIRECTION..... 34

FIGURE 24: A VERTICAL CUTAWAY VIEW OF MSRE MODEL IN SERPENT-2 AND THE
ARRANGEMENT OF THE ZONES 37

FIGURE 25: THE COMPARISON OF THE FLOW FIELD IN VESSEL WITH (LEFT) AND WITHOUT
(RIGHT) VANES..... 39

FIGURE 26: FLOW FIELD IN THE CORE WITH THE SAME FRICTION ON ALL DIRECTIONS..... 40

FIGURE 27: FLOW FIELD WITH LATERAL FRICTION COEFFICIENTS BEING 100 TIMES HIGHER
(LEFT) AND 10,000 TIMES HIGHER (RIGHT)..... 40

FIGURE 28: SELECTED CENTRAL STREAM LINES OF THE FLOW THROUGH CORE CENTER AND
ITS EXTENSION IN THE DOWNCOMER. 41

FIGURE 29: THE VELOCITY MAGNITUDE PROFILE IN THE CORE 42

FIGURE 30: THE AXIAL POWER DISTRIBUTION..... 43

FIGURE 31: THE CONCENTRATION DISTRIBUTION OF THE DELAYED NEUTRON PRECURSOR
WITH STATIC SALT (LEFT) AND WITH MOVING SALT (RIGHT)..... 44

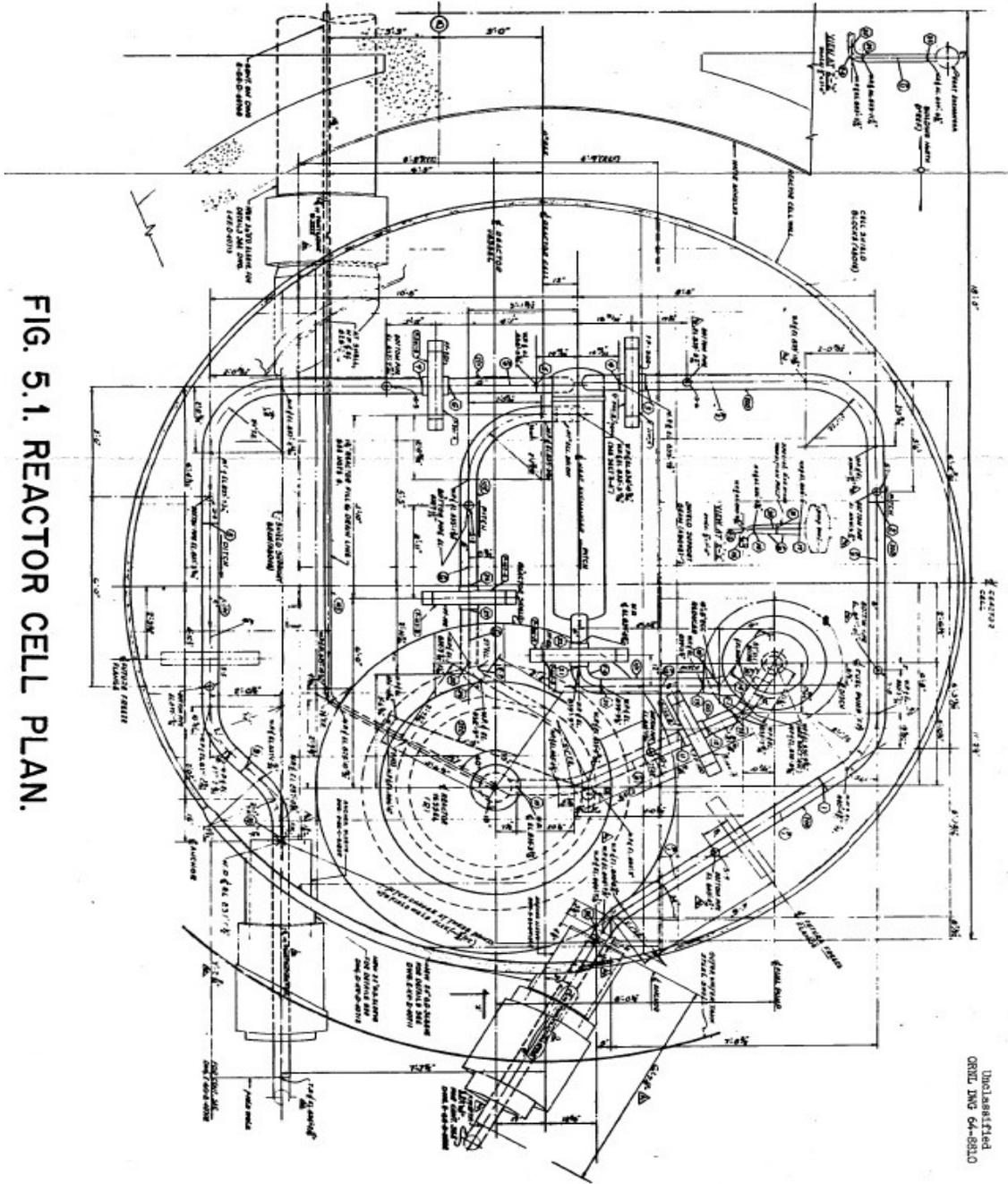
FIGURE 32: THE COMPARISON WITH THE FLOW AT CORE CENTER AND ITS EXTENSION IN THE
LOWER PLENUM FOR THE CASES WITH k, ϵ BEING COMPUTED BY POROUS MEDIA
APPROACH (LEFT), AND WITH ONLY PRESCRIBED VALUES AT INLET. 45

FIGURE 33: THE STREAMLINES ALONG THE PRIMARY CIRCUIT. 46

Appendix A. The salt composition used in Serpent-2 calculation

Fuel salt (wt%)		Salt in lower plenum (wt%)	
Li-7	1.10E-01	Li-7	7.88E-02
Be-9	8.80E-02	Be-9	6.30E-02
F-19	6.84E-01	F-19	4.90E-01
Zr	1.10E-01	Zr	7.88E-02
U-232	1.61E-06	U-232	1.20E-06
U-233	6.68E-03	U-233	4.78E-03
U-234	5.55E-04	U-234	4.00E-04
U-235	5.11E-05	U-235	3.70E-05
U-236	3.65E-06	U-236	2.60E-06
U-238	1.02E-05	U-238	7.30E-06
		Ni	1.86E-01
		Cr	1.61E-02
		Mo	6.81E-02
		Fe	1.04E-02
		Si	1.24E-03
		Mn	1.95E-03

Appendix B. The graphs used for building geometry model



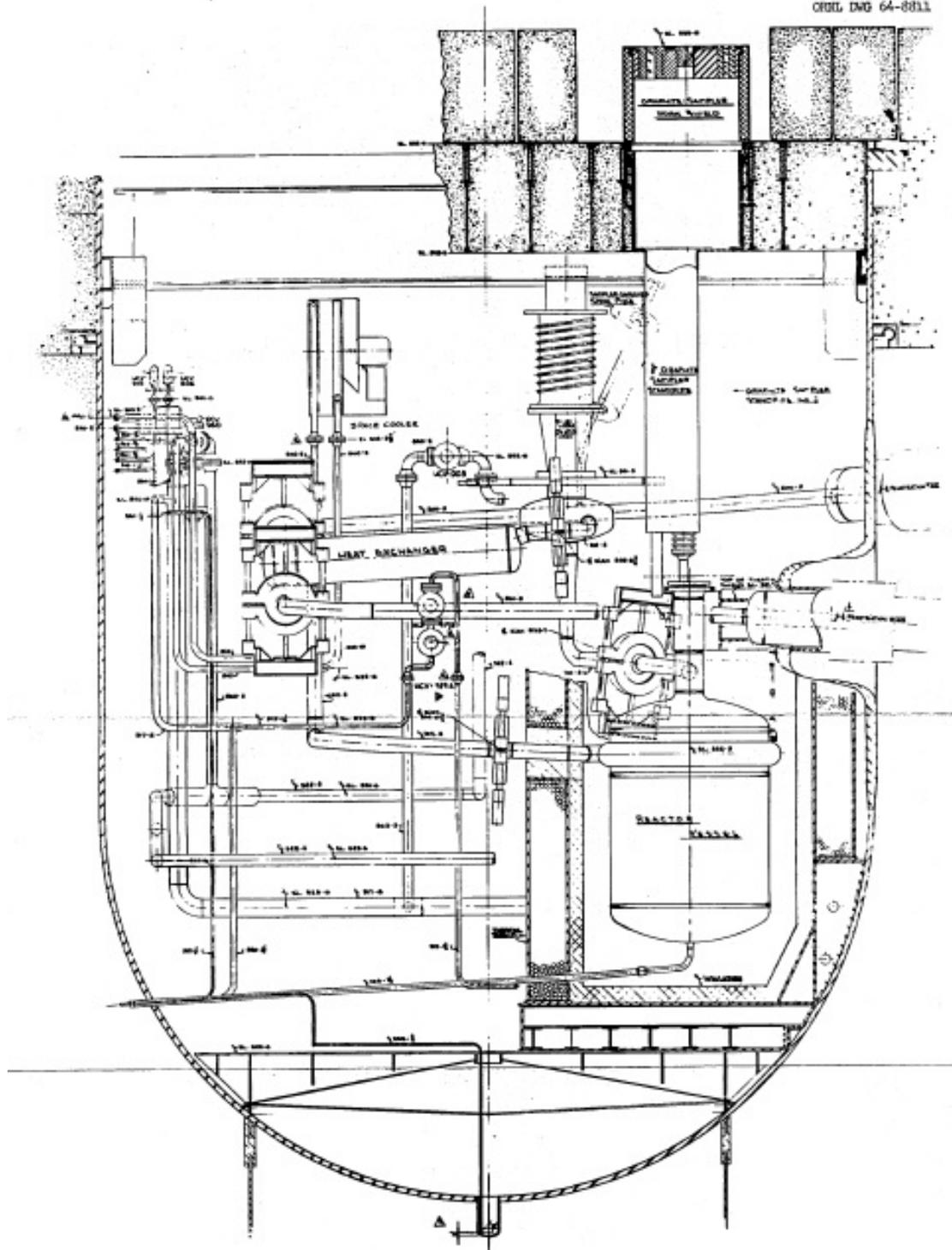


FIG. 5.2. REACTOR CELL ELEVATION.

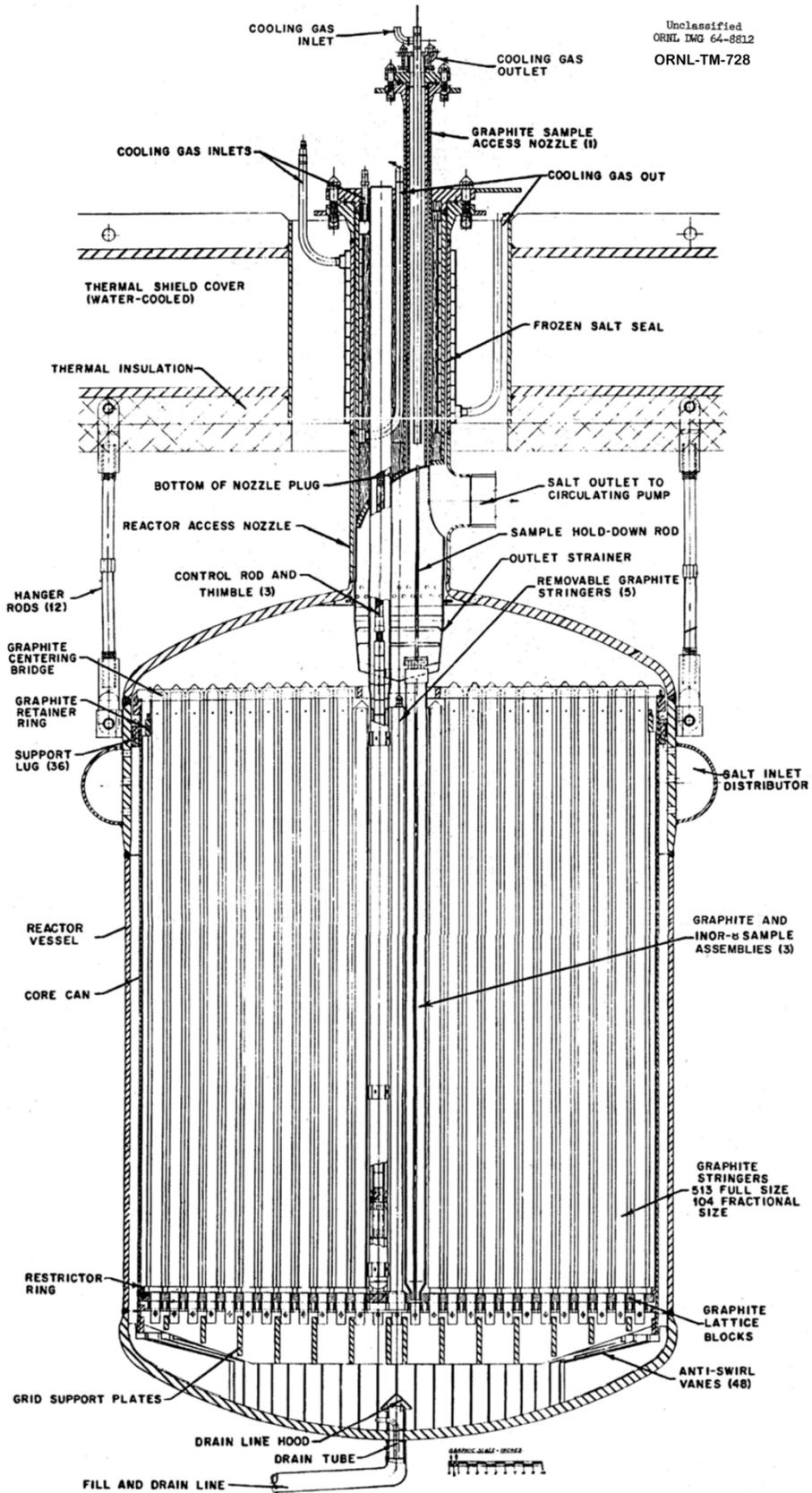


Fig. 5.4.
CROSS SECTION
MSRE REACTOR VESSEL AND ACCESS NOZZLE

Appendix C. The input files for GeN-Foam concerning the porous zones

```

/*----- C++ -----*/
|=====|
| \ / | F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| \ / | O p e r a t i o n | Version: 2.2.1 |
| \ / | A n d | Web: www.OpenFOAM.org |
| \ / | M a n i p u l a t i o n |
|=====|

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

// ***** //

// constants needed to calculate fluid flow in regions treated as porous medium
// all cellZones must be included
// by not defining one of the properties, a default value will be used corresponding to a region of clear fluid

model          byZoneCorrelation;

zones
(
  coreUni
  {
    voidFraction      0.224;
    volumetricAreaFuel      689.5076844;
    volumetricAreaSS      199.8572999;
    reynoldsTurb      (2.3e3 2.3e3 2.3e3); // transition Reynolds to turbulent
    reynoldsLam      (1e3 1e3 1e3); // transition Reynolds to laminar
    darcyConstTurb      (3160 3160 0.316); // A in A*Re^b for Darcy friction factor (in case of turbulent flow)
    darcyConstLam      (64e4 64e4 64); // A in A*Re^b for Darcy friction factor (in case of laminar flow)
    darcyExpTurb      (-0.25 -0.25 -0.25); // b in A*Re^b for Darcy friction factor (in case of turbulent flow)
    darcyExpLam      (-1 -1 -1); // b in A*Re^b for Darcy friction factor (in case of laminar flow)
    hydraulicDiameter      0.01524; // characteristic length
    nusseltConstTurb1      (0.0185 0.0185 0.0185); // A in A*Re^b*Pr^c + D for Nusslet (turbulent case)
    nusseltConstLam1      (0 0 0); // A in A*Re^b*Pr^c + D for Nusslet (laminar case)
    nusseltConstTurb2      (4.82 4.82 4.82); // D in A*Re^b*Pr^c + D for Nusslet (turbulent case)
    nusseltConstLam2      (4.0 4.0 4.0); // D in A*Re^b*Pr^c + D for Nusslet (laminar case)
    nusseltReExpTurb      (0.827 0.827 0.827); // b in A*Re^b*Pr^c + D for Nusslet (turbulent case)
    nusseltReExpLam      (0 0 0); // b in A*Re^b*Pr^c + D for Nusslet (laminar case)
    nusseltPrExpTurb      (0.827 0.827 0.827); // c in A*Re^b*Pr^c + D for Nusslet (turbulent case)
    nusseltPrExpLam      (0 0 0); // c in A*Re^b*Pr^c + D for Nusslet (laminar case)

    externalRhoCp      4.8e6; // rho*cp for the subscale structure

    turbulenceIntensityConst      (0.16 0.16 0.16); // A in A*Re^b for turbulent intensity
    turbulenceIntensityExp      (-0.125 -0.125 -0.125); // b in A*Re^b for turbulent intensity
    turbulenceLengthScaleConst      (0.07 0.07 0.07); // A in A*Dh for turbulent length scale
    kepsilonConvergenceRate      (5 5 5); // rate of convergence of k and epsilon to their equilibrium value in the porous medium
  }
)

```

```

lattice_block
{
    voidFraction          0.224;
    volumetricAreaFuel    689.5076844;
    volumetricAreaSS      199.8572999;
    reynoldsTurb          (2.3e3 2.3e3 2.3e3); // transition Reynolds to turbulent
    reynoldsLam          (1e3 1e3 1e3); // transition Reynolds to laminar
    darcyConstTurb       (3160 3160 0.316); // A in A*Re^b for Darcy friction factor (in case of turbulent flow)
    darcyConstLam       (64e4 64e4 64); // A in A*Re^b for Darcy friction factor (in case of laminar flow)
    darcyExpTurb        (-0.25 -0.25 -0.25); // b in A*Re^b for Darcy friction factor (in case of turbulent flow)
    darcyExpLam        (-1 -1 -1); // b in A*Re^b for Darcy friction factor (in case of laminar flow)
    hydraulicDiameter    0.01524; // characteristic length
    nusseltConstTurb1    (0.0185 0.0185 0.0185); // A in A*Re^b*Pr^c + D for Nusslet (turbulent case)
    nusseltConstLam1     (0 0 0); // A in A*Re^b*Pr^c + D for Nusslet (laminar case)
    nusseltConstTurb2    (4.82 4.82 4.82); // D in A*Re^b*Pr^c + D for Nusslet (turbulent case)
    nusseltConstLam2     (4.0 4.0 4.0); // D in A*Re^b*Pr^c + D for Nusslet (laminar case)
    nusseltReExpTurb     (0.827 0.827 0.827); // b in A*Re^b*Pr^c + D for Nusslet (turbulent case)
    nusseltReExpLam     (0 0 0); // b in A*Re^b*Pr^c + D for Nusslet (laminar case)
    nusseltPrExpTurb     (0.827 0.827 0.827); // c in A*Re^b*Pr^c + D for Nusslet (turbulent case)
    nusseltPrExpLam     (0 0 0); // c in A*Re^b*Pr^c + D for Nusslet (laminar case)

    externalRhoCp        4.8e6; // rho*cp for the subscale structure

    turbulenceIntensityConst (0.16 0.16 0.16); // A in A*Re^b for turbulent intensity
    turbulenceIntensityExp   (-0.125 -0.125 -0.125); // b in A*Re^b for turbulent intensity
    turbulenceLengthScaleConst (0.07 0.07 0.07); // A in A*Dh for turbulent length scale
    kepsilonConvergenceRate (5 5 5); // rate of convergence of k and epsilon to their equilibrium value in the por
}

cold_leg
{
    voidFraction          1;
    reynoldsTurb          (2.3e3 2.3e3 2.3e3); // transition Reynolds to turbulent
    reynoldsLam          (1e3 1e3 1e3); // transition Reynolds to laminar
    darcyConstTurb       (3160 3160 0.316); // A in A*Re^b for Darcy friction factor (in case of turbulent flow)
    darcyConstLam       (64e4 64e4 64); // A in A*Re^b for Darcy friction factor (in case of laminar flow)
    darcyExpTurb        (-0.25 -0.25 -0.25); // b in A*Re^b for Darcy friction factor (in case of turbulent flow)
    darcyExpLam        (-1 -1 -1); // b in A*Re^b for Darcy friction factor (in case of laminar flow)
    hydraulicDiameter    0.127; // characteristic length

    turbulenceIntensityConst (0.16 0.16 0.16); // A in A*Re^b for turbulent intensity
    turbulenceIntensityExp   (-0.125 -0.125 -0.125); // b in A*Re^b for turbulent intensity
    turbulenceLengthScaleConst (0.07 0.07 0.07); // A in A*Dh for turbulent length scale
    kepsilonConvergenceRate (5 5 5); // rate of convergence of k and epsilon to their equilibrium value in the por
}

hot_leg
{
    voidFraction          1;
    reynoldsTurb          (2.3e3 2.3e3 2.3e3); // transition Reynolds to turbulent
    reynoldsLam          (1e3 1e3 1e3); // transition Reynolds to laminar
    darcyConstTurb       (3160 3160 0.316); // A in A*Re^b for Darcy friction factor (in case of turbulent flow)
    darcyConstLam       (64e4 64e4 64); // A in A*Re^b for Darcy friction factor (in case of laminar flow)
    darcyExpTurb        (-0.25 -0.25 -0.25); // b in A*Re^b for Darcy friction factor (in case of turbulent flow)

    darcyExpLam          (-1 -1 -1); // b in A*Re^b for Darcy friction factor (in case of laminar flow)
    hydraulicDiameter    0.127; // characteristic length

    turbulenceIntensityConst (0.16 0.16 0.16); // A in A*Re^b for turbulent intensity
    turbulenceIntensityExp   (-0.125 -0.125 -0.125); // b in A*Re^b for turbulent intensity
    turbulenceLengthScaleConst (0.07 0.07 0.07); // A in A*Dh for turbulent length scale
    kepsilonConvergenceRate (5 5 5); // rate of convergence of k and epsilon to their equilibrium value in the poro
}

distri_downcomer
{
    voidFraction          1;
    reynoldsTurb          (2.3e3 2.3e3 2.3e3); // transition Reynolds to turbulent
    reynoldsLam          (1e3 1e3 1e3); // transition Reynolds to laminar
    darcyConstTurb       (3160 3160 0.316); // A in A*Re^b for Darcy friction factor (in case of turbulent flow)
    darcyConstLam       (64e4 64e4 64); // A in A*Re^b for Darcy friction factor (in case of laminar flow)
    darcyExpTurb        (-0.25 -0.25 -0.25); // b in A*Re^b for Darcy friction factor (in case of turbulent flow)
    darcyExpLam        (-1 -1 -1); // b in A*Re^b for Darcy friction factor (in case of laminar flow)
    hydraulicDiameter    0.0254; // characteristic length

    turbulenceIntensityConst (0.16 0.16 0.16); // A in A*Re^b for turbulent intensity
    turbulenceIntensityExp   (-0.125 -0.125 -0.125); // b in A*Re^b for turbulent intensity
    turbulenceLengthScaleConst (0.07 0.07 0.07); // A in A*Dh for turbulent length scale
    kepsilonConvergenceRate (5 5 5); // rate of convergence of k and epsilon to their equilibrium value in the poro
}

distri_downcomer
{
    voidFraction          0.98;
    reynoldsTurb          (2.3e3 2.3e3 2.3e3); // transition Reynolds to turbulent
    reynoldsLam          (1e3 1e3 1e3); // transition Reynolds to laminar
    darcyConstTurb       (3160 3160 0.316); // A in A*Re^b for Darcy friction factor (in case of turbulent flow)
    darcyConstLam       (64e4 64e4 64); // A in A*Re^b for Darcy friction factor (in case of laminar flow)
    darcyExpTurb        (-0.25 -0.25 -0.25); // b in A*Re^b for Darcy friction factor (in case of turbulent flow)
    darcyExpLam        (-1 -1 -1); // b in A*Re^b for Darcy friction factor (in case of laminar flow)
    hydraulicDiameter    0.05; // characteristic length

    turbulenceIntensityConst (0.16 0.16 0.16); // A in A*Re^b for turbulent intensity
    turbulenceIntensityExp   (-0.125 -0.125 -0.125); // b in A*Re^b for turbulent intensity
    turbulenceLengthScaleConst (0.07 0.07 0.07); // A in A*Dh for turbulent length scale
    kepsilonConvergenceRate (5 5 5); // rate of convergence of k and epsilon to their equilibrium value in the poro
}
}

```

Appendix D. The nuclear data adopted by GeN-Foam

```
/*
crossSection dictionary
Generated by serpentToFoamXS
26-Jul-2016
From SERPENT results file: MSRE5GCRallout_res
*/

/*
physical delayed neutron fraction and spectrum
*/

FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     constant;
    object       MSRE5GCRallout_res;
}

energyGroups 5 ;

precGroups 8 ;

pTarget 1.000000e+007 ;

keff 1.000000e+000 ;

zones
(
    coreUni
    {
        fuelFraction 2.240000e-001 ;
        IV nonuniform List<scalar> 5 (5.345290e-008 3.283210e-006 2.707870e-005 5.458380e-005 2.025370e-004 );
        D nonuniform List<scalar> 5 (2.026250e-002 9.078740e-003 8.668260e-003 8.627980e-003 8.279800e-003 );
        nuSigmaEff nonuniform List<scalar> 5 (5.949690e-003 4.074810e-002 2.760790e-001 6.173140e-001 6.110760e-001 );
        sigmaPow nonuniform List<scalar> 5 (6.945916e-014 5.225291e-013 3.543396e-012 7.923525e-012 7.844402e-012 );
        scatteringMatrix 5 5 (
            ( 1.623840e+001 3.418240e+000 0.000000e+000 0.000000e+000 0.000000e+000 )
            ( 0.000000e+000 3.915970e+001 5.037830e-001 0.000000e+000 0.000000e+000 )
            ( 0.000000e+000 3.985860e-003 3.564090e+001 4.861510e+000 0.000000e+000 )
            ( 0.000000e+000 0.000000e+000 4.005340e-002 3.631240e+001 4.167330e+000 )
            ( 0.000000e+000 0.000000e+000 0.000000e+000 8.191650e-002 4.148560e+001 )
        );
        sigmaDisapp nonuniform List<scalar> 5 (3.419700e+000 5.276000e-001 5.034100e+000 4.519500e+000 3.899000e-001 );
        chiPrompt nonuniform List<scalar> 5 (7.092970e-001 2.907030e-001 0.000000e+000 0.000000e+000 0.000000e+000 );
        chiDelayed nonuniform List<scalar> 5 (8.564780e-002 9.143520e-001 0.000000e+000 0.000000e+000 0.000000e+000 );
        Beta nonuniform List<scalar> 8 (1.892290e-004 3.923750e-004 3.387520e-004 4.861850e-004 7.170260e-004 9.768410e-005 1.331830e-004 );
        lambda nonuniform List<scalar> 8 (1.246670e-002 2.829170e-002 4.252440e-002 1.330420e-001 2.924670e-001 6.664880e-001 1.634780e+000 );
        discFactor nonuniform List<scalar> 5 (1 1 1 1 1 );
        integralFlux nonuniform List<scalar> 5 (1.000000e+000 1.000000e+000 1.000000e+000 1.000000e+000 1.000000e+000 );
    }
)

.attice_block

fuelFraction 2.240000e-001 ;
IV nonuniform List<scalar> 5 (5.345290e-008 3.283210e-006 2.707870e-005 5.458380e-005 2.025370e-004 );
D nonuniform List<scalar> 5 (2.026250e-002 9.078740e-003 8.668260e-003 8.627980e-003 8.279800e-003 );
nuSigmaEff nonuniform List<scalar> 5 (5.949690e-003 4.074810e-002 2.760790e-001 6.173140e-001 6.110760e-001 );
sigmaPow nonuniform List<scalar> 5 (6.945916e-014 5.225291e-013 3.543396e-012 7.923525e-012 7.844402e-012 );
scatteringMatrix 5 5 (
    ( 1.623840e+001 3.418240e+000 0.000000e+000 0.000000e+000 0.000000e+000 )
    ( 0.000000e+000 3.915970e+001 5.037830e-001 0.000000e+000 0.000000e+000 )
    ( 0.000000e+000 3.985860e-003 3.564090e+001 4.861510e+000 0.000000e+000 )
    ( 0.000000e+000 0.000000e+000 4.005340e-002 3.631240e+001 4.167330e+000 )
    ( 0.000000e+000 0.000000e+000 0.000000e+000 8.191650e-002 4.148560e+001 )
);
sigmaDisapp nonuniform List<scalar> 5 (3.419700e+000 5.276000e-001 5.034100e+000 4.519500e+000 3.899000e-001 );
chiPrompt nonuniform List<scalar> 5 (7.092970e-001 2.907030e-001 0.000000e+000 0.000000e+000 0.000000e+000 );
chiDelayed nonuniform List<scalar> 5 (8.564780e-002 9.143520e-001 0.000000e+000 0.000000e+000 0.000000e+000 );
Beta nonuniform List<scalar> 8 (1.892290e-004 3.923750e-004 3.387520e-004 4.861850e-004 7.170260e-004 9.768410e-005 1.331830e-004 2.72459 );
lambda nonuniform List<scalar> 8 (1.246670e-002 2.829170e-002 4.252440e-002 1.330420e-001 2.924670e-001 6.664880e-001 1.634780e+000 3.412 );
discFactor nonuniform List<scalar> 5 (1 1 1 1 1 );
integralFlux nonuniform List<scalar> 5 (1.000000e+000 1.000000e+000 1.000000e+000 1.000000e+000 1.000000e+000 );
}
```

```

UP
{
fuelFraction 1.000000e+000 ;
IV nonuniform List<scalar> 5 (5.315110e-008 2.717590e-006 2.689270e-005 5.352980e-005 1.970230e-004 ) ;
D nonuniform List<scalar> 5 (1.958540e-002 1.116880e-002 1.176980e-002 1.156350e-002 1.130200e-002 ) ;
nuSigmaEff nonuniform List<scalar> 5 (2.138730e-002 1.288390e-001 1.050980e+000 2.294360e+000 2.284080e+000 ) ;
sigmaPow nonuniform List<scalar> 5 (2.495378e-013 1.651905e-012 1.348900e-011 2.944921e-011 2.932085e-011 ) ;
scatteringMatrix 5 5 (
( 1.798980e+001 4.374500e+000 0.000000e+000 0.000000e+000 0.000000e+000 )
( 0.000000e+000 3.203530e+001 2.331450e-001 0.000000e+000 0.000000e+000 )
( 0.000000e+000 4.298740e-003 2.616500e+001 3.004410e+000 0.000000e+000 )
( 0.000000e+000 0.000000e+000 3.845400e-002 2.657280e+001 2.547570e+000 )
( 0.000000e+000 0.000000e+000 0.000000e+000 7.468330e-002 2.983020e+001 )
);
sigmaDisapp nonuniform List<scalar> 5 (4.348900e+000 3.084000e-001 3.636400e+000 3.756100e+000 1.174100e+000 ) ;
chiPrompt nonuniform List<scalar> 5 (7.074550e-001 2.925450e-001 0.000000e+000 0.000000e+000 0.000000e+000 ) ;
chiDelayed nonuniform List<scalar> 5 (9.443040e-002 9.055700e-001 0.000000e+000 0.000000e+000 0.000000e+000 ) ;
Beta nonuniform List<scalar> 8 (1.892290e-004 3.923750e-004 3.387520e-004 4.861850e-004 7.170260e-004 9.768410e-005 1.331830e-004
lambda nonuniform List<scalar> 8 (1.246670e-002 2.829170e-002 4.252440e-002 1.330420e-001 2.924670e-001 6.664880e-001 1.634780e+0
discFactor nonuniform List<scalar> 5 (1 1 1 1 1 ) ;
integralFlux nonuniform List<scalar> 5 (3.350683e-002 3.428113e-002 2.266222e-002 1.845032e-002 1.171146e-002 ) ;
}

```

```

LP
{
fuelFraction 9.080000e-001 ;
IV nonuniform List<scalar> 5 (5.360800e-008 2.778630e-006 2.695400e-005 5.343540e-005 1.866740e-004 ) ;
D nonuniform List<scalar> 5 (1.900220e-002 8.918400e-003 9.063690e-003 8.924720e-003 8.609020e-003 ) ;
nuSigmaEff nonuniform List<scalar> 5 (1.937870e-002 1.193010e-001 9.563400e-001 2.085610e+000 1.974250e+000 ) ;
sigmaPow nonuniform List<scalar> 5 (2.265490e-013 1.529670e-012 1.227431e-011 2.676975e-011 2.534352e-011 ) ;
scatteringMatrix 5 5 (
( 1.863940e+001 4.445200e+000 0.000000e+000 0.000000e+000 0.000000e+000 )
( 0.000000e+000 3.948780e+001 2.272990e-001 0.000000e+000 0.000000e+000 )
( 0.000000e+000 7.666710e-003 3.430480e+001 3.070670e+000 0.000000e+000 )
( 0.000000e+000 0.000000e+000 6.432930e-002 3.472250e+001 2.599370e+000 )
( 0.000000e+000 0.000000e+000 0.000000e+000 1.294760e-001 3.790740e+001 )
);
sigmaDisapp nonuniform List<scalar> 5 (4.466700e+000 4.644000e-001 3.906900e+000 4.077600e+000 2.330900e+000 ) ;
chiPrompt nonuniform List<scalar> 5 (7.088460e-001 2.911540e-001 0.000000e+000 0.000000e+000 0.000000e+000 ) ;
chiDelayed nonuniform List<scalar> 5 (8.241230e-002 9.175880e-001 0.000000e+000 0.000000e+000 0.000000e+000 ) ;
Beta nonuniform List<scalar> 8 (1.892290e-004 3.923750e-004 3.387520e-004 4.861850e-004 7.170260e-004 9.768410e-005 1.331830e-004
lambda nonuniform List<scalar> 8 (1.246670e-002 2.829170e-002 4.252440e-002 1.330420e-001 2.924670e-001 6.664880e-001 1.634780e+0
discFactor nonuniform List<scalar> 5 (1 1 1 1 1 ) ;
integralFlux nonuniform List<scalar> 5 (2.709748e-002 3.411760e-002 2.213409e-002 1.820924e-002 8.143650e-003 ) ;
}

```

```

distrib_downcomer
{
fuelFraction 1.000000e+000 ;
IV nonuniform List<scalar> 5 (5.334550e-008 3.070450e-006 2.712190e-005 5.422100e-005 1.885770e-004 ) ;
D nonuniform List<scalar> 5 (1.904910e-002 9.025440e-003 9.070200e-003 8.934240e-003 8.610400e-003 ) ;
nuSigmaEff nonuniform List<scalar> 5 (1.940330e-002 1.331200e-001 9.581510e-001 2.118790e+000 1.989110e+000 ) ;
sigmaPow nonuniform List<scalar> 5 (2.264714e-013 1.706908e-012 1.229755e-011 2.719564e-011 2.553424e-011 ) ;
scatteringMatrix 5 5 (
( 1.870020e+001 4.341110e+000 0.000000e+000 0.000000e+000 0.000000e+000 )
( 0.000000e+000 3.912490e+001 3.065850e-001 0.000000e+000 0.000000e+000 )
( 0.000000e+000 7.018040e-003 3.416550e+001 3.210420e+000 0.000000e+000 )
( 0.000000e+000 0.000000e+000 5.854300e-002 3.454200e+001 2.799380e+000 )
( 0.000000e+000 0.000000e+000 0.000000e+000 9.617970e-002 3.793350e+001 )
);
sigmaDisapp nonuniform List<scalar> 5 (4.378900e+000 5.424000e-001 4.051000e+000 4.277300e+000 2.326400e+000 ) ;
chiPrompt nonuniform List<scalar> 5 (7.080100e-001 2.919900e-001 0.000000e+000 0.000000e+000 0.000000e+000 ) ;
chiDelayed nonuniform List<scalar> 5 (7.332730e-002 9.266730e-001 0.000000e+000 0.000000e+000 0.000000e+000 ) ;
Beta nonuniform List<scalar> 8 (1.892290e-004 3.923750e-004 3.387520e-004 4.861850e-004 7.170260e-004 9.768410e-005 1.331830e-004
lambda nonuniform List<scalar> 8 (1.246670e-002 2.829170e-002 4.252440e-002 1.330420e-001 2.924670e-001 6.664880e-001 1.634780e+0
discFactor nonuniform List<scalar> 5 (1 1 1 1 1 ) ;
integralFlux nonuniform List<scalar> 5 (1.209665e-002 1.044655e-002 9.637960e-003 9.083593e-003 5.979112e-003 ) ;
}
}

```