# Coupled Neutronics and Thermal-Hydraulics Simulations of Molten Salt Nuclear Reactors

Submitted for

PhD Thesis

Ashkhen Nalbandyan

November 30, 2020

Supervisor: Esben Bryndt Klinkby

Co-supervisor: Bent Lauritzen

DTU Physics

Section of Radiation Physics

# Acknowledgements

The energy demand worldwide is growing each year and so are the concerns over the sustainability of the current energy supply models. In our quest for a sustainable future, nuclear power is arguably one of the most optimal solutions we have for providing economically competitive and clean energy for the population of the developing world.
Recently the global nuclear research and industry's efforts target the development of advanced nuclear reactor designs envisaging better economic viability, increased operational safety, and decreased proliferation risks. One of these reactor designs, the Molten Salt Reactor, has significant potential due to its intrinsic safety features, better fuel utilization options, as well as increased economic competitiveness.
On the way to regulatory approval, each reactor design has to be proven feasible and safe to operate under a wide range of operational conditions; normal as well as accident-related. For this purpose modeling and simulation tools are widely deployed in the conceptual and basic design phases. As nuclear reactors are complex systems with physical phenomena occurring at different time and length scales, this implies the application of the multiphysics analysis approach. In the case of MSRs, the development of a new class of modeling techniques and tools is necessary to capture the intrinsic features of the liquid fuel, which introduces a new level of complex interdependence between the reactor processes. This work is dedicated to modeling and simulation of Molten Salt Reactors to further contribute to the efforts of the international nuclear reactor research community towards the development of multiphysics tools for MSR analysis.
This work was conducted at DTU Physics, section of Radiation Physics, under the supervision of Esben B. Klinkby as my main supervisor and Bent Lauritzen as a co-supervisor. Without their guidance and help, this project would have not succeeded. Thanks to Esben for always being supportive and for the productive discussions we had. Thanks to Bent for all the thorough editing of the papers we co-authored and the constructive critique. Esben and Bent helped me to keep this project on track at the same time giving me the freedom of deciding what methods to try and how to proceed with my research.
Additionally, I want to thank all those who helped me with my project and supported me during these years, particularly; Jacob Groth-Jensen for valuable collaboration and all his help, Prof. Antonio Cammi and Stefano Lorenzi for giving me the possibility to do my external research stay at PoliMi.
I want to thank fellow Ph.D. students and colleagues for keeping me company during these years; Amalia, Frederik, Eleine, Grichar, Jacob, Jeppe, Mads, Magdalena, Martin, Raju, Daria, Anastasiia, and Antariksh.
Finally, thanks to my family my sister Eva and my parents for always being there for me despite the distance.
Last but not least, I want to thank my boyfriend Lorenzo for his endless support.

# Abstract

Nuclear power offers sustainable, abundant, and economically competitive energy production with a carbon footprint close to zero. Within the spectrum of Generation IV nuclear power plant designs, the liquid-fuel Molten Salt Reactor (MSR) features several potential benefits in terms of enhanced safety, reduced proliferation risks, and economic competitiveness. Furthermore, high outlet temperatures enable compatibility with process heat applications, and several designs for small modular MSR's show significant promise.

Concerning successful licensing and commercialization of MSRs, Modeling, and Simulation (MS) techniques are required to demonstrate plant behavior under a wide range of operational and accidental scenarios. This is done throughout the design development stages and for the safety assessment of the plant. However, the liquid fuel employed in MSRs introduces several key differences between these and other types of nuclear reactors based on solid fuel, which result in a fundamentally different approach to modeling and simulation.

This Ph.D. thesis focuses on the development of Multiphysics coupling techniques applied to Molten Salt Reactors. Specifically, Monte-Carlo particle transport software capable of modeling heat production from nuclear fission is coupled to high fidelity Computational Fluid Dynamics (CFD) software to accurately capture the impact of the fuel being liquid on the intrinsic operational and safety features of the plant. Alongside this method, a different technique of deterministic neutron transport modeling is also explored. This thesis provides an overview of possible approaches for Multiphysics modeling of MSRs and discusses potential benefits and drawbacks of the techniques applied within this work. The methods developed can be used for the design development and optimization of MSRs. Besides, the work holds merit for other areas of MS applications wherein coupling between different physical phenomena is required.

# List of Publications

The following peer-reviewed publications present the main results and methods discussed in this work:

I. **Nalbandyan A.**, Klinkby E.B., Lauritzen B., "Coupling Techniques for Multiphysics Modeling of Molten Salt Reactors", Proceedings of the 1st International Conference on Small Modular Reactors, G4SR-1, Canadian Nuclear Society, 2018.

II. **Nalbandyan A.**, Klinkby E.B., Lauritzen B.,Groth-Jensen J., Steyn R., "Coupled Neutronics/Thermal-Hydraulics Assessment of Graphite Moderated Molten Salt Reactors", Proceedings of the 18th International Topical Meeting on Nuclear Reactor Thermal Hydraulics, American Nuclear Society, 2019, ISBN: 978-0-89448-767-5.

III. **Nalbandyan A.**, Cammi A., Lorenzi S., Klinkby E.B., Lauritzen B., "Computational Fluid Dynamics Modelling of DYNASTY ", submitted to the journal of Chemical Engineering Science.

IV. Groth-Jensen J., **Nalbandyan A.**, Klinkby E. B., Lauritzen B., Pedersen A.V., "Verification and benchmarking of multiphysics coupling techniques for modeling of molten salt reactors", Submitted to the journal Annals of Nuclear Energy.

Oral Presentations at Conferences related to the work:

1. Canadian Nuclear Society, 1st International Conference on Small Modular Reactors, Ottawa, Canada, 2018

2. American Nuclear Society, 18th International Topical Meeting on Nuclear Reactor Thermal Hydraulics, Portland, Oregon, 2019

3. Thorium Energy Alliance, TEAC 10 Conference, Oak Ridge, Oregon, 2019

# List of Abbreviations

**MS** - Modeling and Simulations

**MSR** - Molten Salt Reactors

**MSRE** - Molten Salt Reactor Experiment

**MSFR** - Molten Salt Fast Reactor

**CFD** - Computational Fluid Dynamics

**MC** - Monte-Carlo

**CSS** - Criticality Source Simulation

**TMS** - Transient Mode Simulation

**DNP** - Delayed Neutron Precursor

**ORNL** -Oak Ridge National Laboratory

**RANS** -Reynolds Averaged Navier-Stokes

# Contents

# List of Figures

# List of Tables

# Part I

# Chapter 1

# Introduction

> *So the ideas of molten fluorides first came into the chain reaction community by 1945. I was a disciple of Eugene Wigner's, and I was bitten by the 'homogeneous' bug, and I've never quite recovered from that.*
>
> *Alvin Weinberg, 1997*

## 1.1 Molten Salt Reactors: A Brief History

After the discovery of the nuclear fission phenomenon by Lise Meitner, Otto Hahn, and Fritz Strassman back in 1938 [1], it did not take long to realize the substantial potential that the fission reaction had in terms of energy generation. The following decades will see rapid growth in the number of nuclear reactors designed to run fission chain reaction in a controlled manner in order to generate heat and electricity. As of today, the nuclear fission reactors are providing around 4.3% of the global primary energy shares and around 10.1% of the total electricity shares [2].
From the very first experimental reactor, the Chicago Pile-1, to the nowadays advanced reactors, the industry of nuclear power generation saw four stages of development. The nuclear power plants are classified into four generations, based on the novelty of design approaches, enhanced economy, and improved safety features.

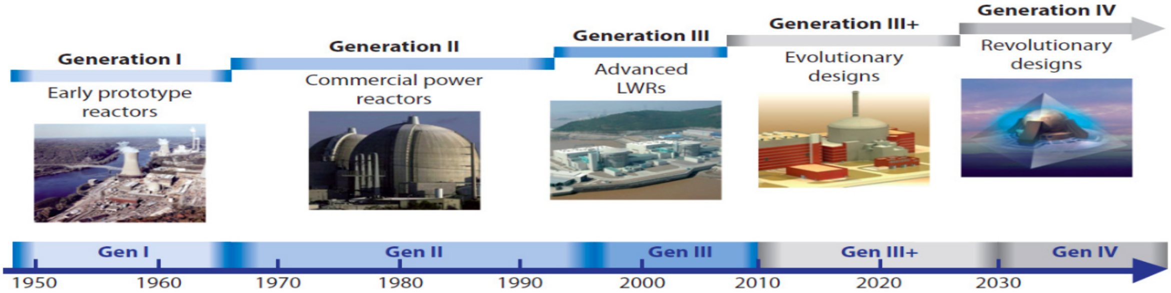


Figure 1.1: The evolution of the nuclear power plant industry (Image courtesy: Technology Roadmap Update for Generation IV Nuclear Energy Systems, 2014)

As Figure 1.1 indicates, we are now operating Generation $III+$ power plants; all the nuclear power plants that are operating currently, have at least one thing in common:

they utilize solid nuclear fuel mostly in form of pellets, packed into fuel rods and assemblies. In the late 1940$s$, however, the US government-funded project known as the Aircraft Reactor Project aimed to deploy a liquid nuclear fuel in form of the fissile material dissolved in fluoride salts [3], [4]. The plan was to come up with a design of a compact reactor that can be placed on an aircraft and thus it had to have high power density and operate at very high temperatures. There was a consensus amongst the project participants that the conventional solid fuel scenario will be outperformed by having a liquid-fueled reactor [5]. The Aircraft Reactor Experiment in 1954 showed the feasibility of the concept with the salt having an operational temperature range of 922 $K$ (inlet) to 1088 $K$ (outlet) at a nominal power of 2.5 $MW$ [4]. Although the idea of having a reactor on an aircraft was eventually abandoned, the project evolved into the civil nuclear power production sector and resulted in the construction and the operation of the Molten Salt Reactor Experiment (MSRE) at the Oak Ridge National Laboratory in the 1960$s$ [6]. The experimental campaign proved the feasibility of having liquid fluoride salt as a reactor fuel carrier and provided substantial insight into the fuel salt chemistry and the radiation-induced material damage.

The concept of having a liquid nuclear fuel came back to the nuclear power community in 2000 when the international nuclear energy community set out to define and develop Generation $IV$ nuclear energy systems. The goals of the Generation $IV$ initiative are to enhance the sustainability and economical competitiveness of the new generation nuclear power plants as well as to ensure safety and reliability by adopting novel, largely passive safety features. Furthermore, the new generation power plants will exhibit increased proliferation resistance [9]. Several advantages of the molten salts that make them especially attractive as fuel carrier, are [6]:

- The fuel is liquid during the reactor normal operation: no solid fuel fabrication and reprocessing costs.

- The fuel burn-up is not limited by the radiation-induced damage to the structural components or reactivity loss, as the fuel can be reprocessed online.

- The liquid fuel can operate at higher temperatures thus increasing the heat to power efficiency: the liquid fuel can be used for high-temperature process heat production.

- No high-pressure vessel and piping is required, as liquid molten salt operates at near atmospheric pressure.

- The liquid molten-salt fuel has strong negative Doppler and density feedback coefficients which result in enhanced stability and safety of the system. The Doppler coefficient reflects the response of the neutron absorption cross-sections in resonance energy range to the increase in fuel temperature; the higher is the temperature, the broader are the resonance absorption peaks leading to more neutrons being absorbed and subsequently less of them being available for fission, which prevents the reactor power from rising. Similarly, the density coefficient reflects the increase in the neutron leakage probability when the fuel density decreases due to the rise of temperature. Both these feedback mechanisms are crucial in ensuring the safe and stable operation of nuclear reactors. Doppler and density coefficients of the fuel are both negative and are stronger for the liquid fuel because the heat is released directly in the fuel salt.

- The salts do not interact vigorously with water and air.

| Reactor and Country | Neutron Spectrum | Salt type | Fuel cycle |
|---|---|---|---|
| MSRE, USA | Thermal | Fluoride | $^{232}$Th –$^{233}$U,$^{235}$U |
| TMSR-LF, China | Thermal | Fluoride | $^{232}$Th –$^{233}$U |
| MSFR, EU | Fast | Fluoride | $^{232}$Th –$^{233}$U, MA |
| CMSR, Denmark | Thermal | Fluoride salts | $^{232}$Th –$^{233}$U,$^{235}$U, SNF |
| CAWB, Denmark | Thermal | Fluoride | $^{232}$Th –$^{233}$U |
| IMSR, Canada | Thermal | Fluoride | $^{235}$U |
| MOSART, Russia | Fast | Fluoride | $^{238}$U –$^{239}$Pu |
| LFTR, USA | Thermal | Fluoride | $^{232}$Th –$^{233}$U |
| TMSR, USA | Thermal | Fluoride | $^{232}$Th –$^{233}$U |
| MCFR, USA | Fast | Cloride | $^{232}$Th –$^{233}$U |

Table 1.1: Overview of several main MSR design concepts based on the neutron energy spectrum and fuel cycle: MA stands for Minor Actinides and SNF stands for Spent Nuclear Fuel.

- MSRs can be built as compact modular units which makes them well suited for the modern electrical grid needs.

Alongside the above-mentioned advantages, the Molten Salt Reactors also exhibit several drawbacks, such as [7], [8]:

- The utilization of molten salts as a fuel carrier demands a thorough understanding of the salt chemistry and interaction mechanisms with the reactor structural materials.
- The absence of fuel cladding and rods implies a diminished number of radionuclide barriers.
- Potentially high radiation loads on the reactor containment can be expected.
- Establishment of appropriate safeguard approaches is required.

In the framework of the Generation $IV$ forum, the liquid fuel molten salt reactor has been chosen as one of the six key technologies with the official concept within the Generation IV framework being the non-moderated Molten Salt Fast Reactor [9]. In parallel to this concept, several other, mainly thermal reactor concepts are being developed and are at different stages of design and deployment. Some of them envisage the implementation of the Th-U fuel cycle, wherein the fertile $Th$ can be used to breed new fuel in an MSR and bring to a closure of the fuel cycle. Other design concepts focus on burning the nuclear waste from conventional nuclear reactors, acting as efficient waste transmuters. Many of these design concepts are small modular reactors, ensuring fast deployment to practically any location and offering a solution to the clean energy demand. A summary of some key MSR designs [8] based on the neutron spectrum and fuel composition is given in Table 1.1 .

All of the reactor designs presented in Table 1.1, except the MSRE, are at various development stages with most of them scheduling prototypes between 2022 and 2030. Such a considerable activity towards the development and deployment of MSRs indicates that

MSRs could potentially become a key player in the nuclear energy market.

## 1.2 Recent developments in MSR modelling

One of the hurdles on the way towards licensing and deploying the MSRs is the development and validation of dedicated simulation tools capable to capture the relevant phenomena of liquid fuel. Such Multiphysics modelling tools should aim to reflect the relevant characteristics of liquid fuel.
Nuclear reactors are complex systems featuring multiphysics processes, e.g conjugate heat transfer, nucleate boiling, neutron kinetics, taking place on a multiscale level ranging from micro to mesoscale phenomena. The different software tools that are developed to model the relevant processes in nuclear reactors are thus designed to run at different time and length scales [11]. Usually, these software tools are coupled in a manner known as an operator-splitting approach, wherein the software tools exchange information only once per time-step. In order to achieve high accuracy with this approach, it is often times necessary to choose the smallest time and length scale for the information exchange between the software and this may lead to a computational inefficiency of the coupling method [10]. The correct and accurate Multiphysics modelling is essential and even more demanding for liquid-fueled MSRs, as the fluid flow characteristics directly impact the core kinetics. The impact of the liquid fuel on the neutronics can be manifested via:

- Transport of delayed neutron precursors with the flow.
  In MSRs as in all other nuclear reactors, the reaction responsible for heat production is nuclear fission. A fission event occurs when a neutron with suitable energy gets absorbed in a fissile nucleus in the fuel. As a result, $\approx 200MeV$ recoverable energy is released, and additional $\approx 2.5$ neutrons are emitted which makes the reaction self-sustainable. A very small amount of fission neutrons (typically less than 1%) is emitted not instantaneously when the nucleus fissions, but long after the initial fission, from the beta-decay of the fission products, called delayed neutron precursors. According to their half-lives, the delayed neutron precursors are categorized into groups ranging from short- (a tenth of a *second*) to long-lived (tens of *seconds*) precursors. Thus, the neutrons available for the fission chain reaction are either prompt or delayed. Even though the fraction of the delayed neutrons is small, they are crucial to the operation of a nuclear power reactor. If a reactor would have been relying on prompt neutrons only, any, even slight change in the effective neutron multiplication factor would have resulted in a drastic increase of reactor power on a very small time scale, making it impossible to control such a reactor [13]. It is due to the impact of the delayed neutrons that control and operation of nuclear power plants are possible, and it is thus very important to be able to accurately estimate the fraction of the delayed neutrons in the reactor. Usually, instead of talking about the average fraction of the delayed neutrons denoted as $\beta$, it is common to use the $\beta_{eff}$ instead, called the effective delayed neutron fraction which is the product of the physical fraction $\beta$ and an importance factor. This importance factor reflects the differences between the prompt and delayed neutron properties and the relative effectiveness of the delayed neutrons depending on their emission spectrum. Calculation of the $\beta_{eff}$ typically involves the introduction of the adjoint neutron transport equation the solution to which gives the neutron importance [14]. Deterministic as well as Monte-Carlo methods are available for the calculation of

the $\beta_{eff}$ [15], [17] and the $\beta_{eff}$ can also be measured in the reactors [18].
If the fuel is liquid, besides the above mentioned difference between $\beta$ and $\beta_{eff}$ due to the differences in energy spectrum between the prompt and the delayed neutrons, some delayed neutron precursors will have sufficient time to be transported with the flow and they will be re-distributed in the core before they decay and give birth to a delayed neutron. These spatial effects will have a direct impact on the $\beta_{eff}$. In MSRs, the motion of the fuel results in a decrease of the $\beta_{eff}$ value. To calculate the $\beta_{eff}$ for the MSRs, the fuel velocity has to be taken into account explicitly. Some previous works achieve this by adopting analytical, deterministic, or Monte-Carlo methods [19], [20].

- In a nuclear fission power plant, most of the heat is produced by the fission fragments interacting with the surrounding atoms as a result of which their kinetic energy is ultimately transformed into thermal energy. On top of that, the kinetic energy of prompt neutrons and the energy of prompt gamma rays are also contributing to heat production. Additionally, the decay of neutron-rich fission fragments results in energy production via $\beta$ and delayed $\gamma$ rays and there is some energy production also from the capture $\gamma$ rays released after a radiative capture of neutrons. Whereas fission energy release ceases after the reactor shutdown, the highly unstable fission fragments continue decaying long after the shutdown, producing decay heat. If not removed properly, the decay heat can cause major safety-related problems after the shutdown and it is thus important to be able to assess the production of the decay heat correctly. Since the fission fragments can be transported with the fuel flow, just as the delayed neutron precursors, this is another aspect of the modelling of a liquid fuel that has to be tackled carefully.

Additionally, the possibility of online reprocessing of the fuel in MSRs as well as the nuances of the salt interaction with structural materials and thermal loads on the structural materials due to the high radiation fields, also imply the necessity of developing integrated modelling platforms that will have the capability to assess e.g. fuel behavior and system thermal-mechanics as well as some specifics of the salt chemistry. Thus, it is of importance to develop methods and tools for multiphysics modelling of MSRs and there has been a significant effort towards the development of dedicated tools for multiphysics modelling of MSRs recently.
Most of the multiphysics modelling tools developed for the MSRs recently, feature neutron kinetics modelling based on approximations, e.g. a multi-group neutron diffusion approximation. Even though the diffusion approach is an approximation to the neutron transport phenomena, it is computationally lightweight and gives reasonably accurate results when applied to reactor analysis and is thus often a desired alternative to the computationally heavy stochastic neutron-kinetics analysis methods, such as the Monte-Carlo method [12]. Moreover, the diffusion approximation-based solvers are widely applied for MSR studies, because they can be integrated well with thermal-hydraulics and thermal-mechanics sub-solvers to form a single software suite. Examples of such solvers are several, e.g. the *Gen-Foam* solver developed by *Fiorina et al.* [16], which adopts multi-group diffusion approximations for solving the neutron transport problem coupled to a thermal-hydraulics and thermal-mechanics sub-solvers. Both the neutronics and the thermal-hydraulics modules are implemented in the OpenFOAM C++ toolbox based on the standard finite volume method.
Another example is the multiphysics solver developed by *Cervi et al.* [21], [22] and featur-

ing not only multi-group diffusion approximation but also a more accurate $SP_3$ approximation to the neutron transport coupled to either one- or two-phase thermal-hydraulics solvers in OpenFOAM.

In yet another similar approach a modelling suite is developed by *Blanco et al.* [23], featuring a neutronics solver (multi-group diffusion or SP1 and SP3 transport) implemented in OpenFOAM and coupled to thermal-hydraulics and thermal mechanics solvers in OpenFOAM.

Other developments include the utilization of conventional software, such as the work by *Cammi et al.* [24] featuring modelling of MSRs using a conventional finite element analysis software COMSOL Multiphysics.

In contrast to the above-presented examples where the main development effort is done for the implementation of the neutronics module and the coupling strategy between the neutronics and the fluid flow, some tools are relying on completely in-house developed solvers for both the neutronics and thermal-hydraulics, such as the multi-physics package developed by *Tiberga et al.* [25] and comprised of two in-house developed software; Phantom-$S_N$ multi-group Boltzmann equation solver for neutronics coupled to the DF-Flows parallel solver for incompressible Navier-Stokes equations. Both the software are based on the Discontinuous Galerkin Finite Method for space discretization [25].

Other efforts in the MSR research community feature extension of codes developed initially for light water or other solid-fueled reactor studies to facilitate liquid fuel characteristics, such as the work presented in *Krepel et al.* [26] in extending the in-house Light Water Reactor(LWR) diffusion code DYN3D which resulted in DYN3D-MSR.

In a similar manner, *Kophazi et al.* [27] coupled in-house codes DALTON and THERM to analyze the MSRE. *SIMMER-III* was also extended to facilitate liquid fuel modelling, as presented in *Wang et al.* [28].

Finally, some high-fidelity simulation environments for MSR modelling are developed as well, e.g. the Moltres featuring deterministic neutronics and thermal-hydraulics solvers in the Multiphysics Object-Oriented Simulation Environment(MOOSE) [29].

Table 1.2 summarizes some of the multiphysics tools developed recently.

## 1.3 Thesis Objective and Outline

In view of the foregoing, the objective of this work is to develop multiphysics modelling techniques focusing on coupling neutronics to thermal-hydraulics for the analysis of MSRs and to demonstrate the feasibility of the developed techniques by either code-to-code validation or, when available, by comparing to experimental data. Three methods are developed and applied to the coupled simulations of the MSRs in steady-state as well as transient scenarios.

The first method presented in the second chapter features a steady-state assessment of the Molten Salt Fast Reactor (MSFR) applying first conventional codes (ANSYS CFX for the thermal-hydraulics and MCNP for the neutronics) and comparing the results to a simplified deterministic neutron transport solver based on diffusion approach, implemented in the open-source library OpenFOAM and coupled to an OpenFOAM thermal-hydraulics solver. Compared to the solvers described before, the solver developed in this work solves only a one-group diffusion equation, which, as previous works show, is suitable for the analysis of the fast spectrum Molten salt Reactors [30]. However, an extension to multi-group diffusion approximation is necessary if the solver is to be applied to the analysis

of thermal spectrum MSRs.
Despite this method having clear advantages in terms of facilitating the coupling, implementing dedicated numerical methods to reduce the coupling nonlinearities, and being computationally lightweight, this approach also implies the necessity to do extensive software development and make sure that all applied approximations yield valid results.

| Developer | NK module | TH module |
|---|---|---|
| CNRS | SP or multi-group diffusion | single-phase incompressible Navier-Stokes |
| PoliMi | SP or multi-group diffusion | single-phase/two-phase, compressible/incompressible Navier-Stokes |
| PSI & EPFL | SP or multi-group diffusion | incompressible, single phase Navier-Stokes |
| TU Delft | $S_N$ solver for multi-group Boltzmann | incompressible Navier-Stokes |
| Univ. of Illinois | Multi-group diffusion | compressible/incompressible Navier-Stokes |

Table 1.2: Several multiphysics modelling tools developed for MSR analysis

In a different approach, it is possible to utilize already validated single-physics codes for reactor neutron kinetics and thermal-hydraulics analysis and to focus on developing a coupling interface and strategy that will allow for accurate coupling. In this work, an external coupling technique is implemented between OpenFOAM and Serpent Monte Carlo neutron transport code based on an existing multiphysics interface between these two software, which facilitates straight-forward data exchange between the two software. Monte-Carlo simulations are the most accurate approach to solving the neutron transport phenomenon, and there have been several approaches developed recently to apply coupled Monte-Carlo/CFD methods to the analysis of MSRs. For example, standalone Monte-Carlo codes have been used together with CFD codes to assess the steady-state behavior and the initial neutronic and thermal-hydraulic design of the MSFR [31], and the Serpent Monte-Carlo code has been extended to allow for the modelling of the fuel flow and integrated with OpenFOAM to account for e.g. fuel compressibility effects [32]. However, the coupled Monte-Carlo/CFD approach is not as widely applied as the previously described solvers relying on approximations to the neutron transport modelling. The reason is firstly the computational time that a Monte-Carlo simulation could potentially take in order to have reliable statistics. Considering that for MSR studies the Monte-Carlo code will have to be coupled to a CFD code which is also often time-consuming, especially for large models and fine grids, the combination of two can potentially cause a bottleneck in terms of computational resources. Another shortcoming of using Monte-Carlo codes in Multiphysics modelling of MSRs can be attributed to the fact that the validated and largely used Monte-Carlo software might not be easily compatible with CFD software and the coupling between the two will not result in a close monolithic piece of software but rather a coupling interface should be developed. This work attempts to develop a reliable way to such coupling, utilizing accurate Monte-Carlo methods and to investigate whether this technique can yield physically adequate results in a computationally reasonable time. Two possible coupling mechanisms are discussed in Chapter 3 and Chapter 4 respectively. The first mechanism described in Chapter 3 is based on coordinate-based delayed neutron precursor tracking in combination with the fuel velocity field obtained from an OpenFOAM CFD solver in order to reflect the fuel movement and its impact on the neutron kinetics. This method is tested by simulating a single channel of the MSRE and comparing fuel and graphite temperatures to the model calculations available from the design phase of the MSRE. Additionally, the capability of the coupling mechanism to model transients is tested by comparing the results to experimental data available from the MSRE operation phase.

The second mechanism described in Chapter 4, is based on developing an additional passive scalar transport solver in OpenFOAM, which, relying on delayed neutron precursor decay constants and fractions as well as the initial delayed neutron precursor source obtained assuming static fuel and extracted from Serpent, solves for the transport of delayed neutron precursors and writes an updated source which reflects the fuel movement and which is used as an input for subsequent neutronics simulation in Serpent. This technique is benchmarked using the results from a multi-physics numerical benchmark for codes dedicated to molten salt fast reactors.

The advantages of this coupling approach are the usage of validated software for individual physics and limited code development required. As a disadvantage, the computational burden of the Monte-Carlo approach can be mentioned as well as the accuracy issues due to the operator-splitting nature of the coupling mechanism, which requires the usage of small time-steps in transient simulation regime and results again in increased computational time.

The coupling methods developed and tested in this work as well as the application cases can be summarized as:

- Neutron Diffusion solver coupled to a CFD solver: 1-group neutron diffusion and delayed neutron transport equations are implemented in OpenFOAM and coupled to a single-phase, incompressible Navier-Stokes equations solver in OpenFOAM. Tested on steady-state MSFR example.

- Monte-Carlo software coupled to CFD, no delayed neutron precursor transport: MCNP and ANSYS CFX are used to carry out a comparison of results to the above-mentioned method results. Tested on steady-state MSFR example.

- Monte-Carlo software coupled to CFD with simplified DNP transport: Serpent is coupled to OpenFOAM with the velocity field from CFD used to shift the DNPs in Serpent. Tested on a single-channel MSRE analysis.

- Monte-Carlo software coupled to CFD with DNP transport solver implemented in OpenFOAM: Serpent is coupled to OpenFOAM CFD solver and an additional solver for DNP transport implemented in OpenFOAM. Tested using the results of a multiphysics benchmark for molten salt reactors.

Additionally, results of a CFD assessment of an experimental natural circulation loop are presented in the last chapter of the thesis, relevant for the passive decay heat removal systems envisaged for molten salt reactors.

# Bibliography

[1] O. R. Frisch, J. A. Wheeler, "The discovery of fission", Physics Today 20, 1967.

[2] "BP Statistical Review of World Energy 2020", 69th edition.

[3] "Review of manned aircraft propulsion program", Atomic Energy Commission and Department of Defense, February 1963.

[4] W. B. Cottrell, H. E. Hungerford, J. K. Leslie, L. Meem, "Operation of the Aircraft Reactor Experiment", September 6, 1956, ORNL.

[5] H. G. MacPherson, "The Molten Salt Reactor Adventure", Nuclear Science and Engineering 90, Pages 374-380, 1985.

[6] R. C. Robertson, "MSRE Design and Operations Report, Part I: Description of Reactor Design", ORNL Internal Reports, 1960.

[7] D. E. Holcomb, "Molten Salt Reactors Today Status & Challenges", Workshop on MSR Technologies – Commemorating the 50th Anniversary of the Startup of the MSRE, 2015.

[8] D. N. Kovacic, L. G. Worrall, A. Worrall, G. F. Flanagan, D. E. Holcomb, R. Bari, L. Cheng ,D. Farley, M. Sternat , "Safeguards Challenges for Molten Salt Reactors", INMM Annual Meeting, 2018.

[9] "Technology Roadmap Updatefor Generation IV Nuclear Energy Systems", NEA OECD, January 2014.

[10] J. C. Ragusa, V. S. Mahadevan, "Consistent and accurate schemes for coupled neutronics thermal-hydraulics reactor analysis", Nuclear Engineering and Design 239, Pages 566–579, 2009.

[11] D. R. Gaston et al., "Physics-based multiscale coupling for full core nuclear reactor", Annals of Nuclear Energy 84, Pages 45–54, 2015. simulation

[12] S. Marguet, "Diffusion Approximation in Neutron Physics", The Physics of Nuclear Reactors, Springer, 2017.

[13] J. R. Lamarsh, A. J. Baratta, "Introduction to Nuclear Engineering", 3rd Edition, Pearson, 2013.

[14] P. Saracco, S. Dulla, P. Ravetto, "The adjoint neutron transport equation and the statistical approach for its solution", European Physical Journal 131(11), 2016.

[15] R. K. Meulekamp, S. C. van der Marck, "Calculating the effective delayed neutron fraction with Monte Carlo", Nuclear Science and Engineering, 152(2), Pages 142–148, 2006.

[16] C. Fiorina et al., "Gen-Foam: a novel OpenFOAM based multi-physics solver for 2D/3D transient analysis of nuclear reactors", Nuclear Engineering and Design 294, Pages 24-37, 2015.

[17] A. Talamo et al., "Deterministic calculation of the effective delayed neutron fraction without using the adjoint neutron flux", International Conference on Advances in Reactor physics to Power the Nuclear Renaissance, Pittsburgh, PA (United States), 2010.

[18] T. Sakurai, S. Okajima, M. Andoh, T. Osugi, "Experimental cores for benchmark experiments of effective delayed neutron fraction $\beta_{eff}$", FCA. Progress in Nuclear Energy 35(2), Pages 131–156, 1999.

[19] M. Aufiero et al., "Calculating the effective delayed neutron fraction in the Molten Salt Fast Reactor: Analytical, deterministic and Monte Carlo approaches", Annals of Nuclear Energy 65, Pages 78-90, 2014.

[20] GF. Zhu, R. Yan, HH. Peng et al. "Application of Monte Carlo method to calculate the effective delayed neutron fraction in molten salt reactor.", Nuclear Science Technology, 2019.

[21] E. Cervi, S. Lorenzi, A. Cammi, L. Luzzi, "Development of a multiphysics model for the study of fuel compressibility effects in the Molten Salt Fast Reactor", Chemical Engineering Science 193, Pages 379–393, 2019

[22] E. Cervi, S. Lorenzi, A. Cammi, L. Luzzi, "An Euler-Euler multi-physics solver for the analysis of the helium bubbling system in the MSFR", 26th International Conference Nuclear Energy for New Europe, Bled, Slovenia, 2017.

[23] J.A. Blanco, P. Rubiolo, E. Dumonteil, "Neutronic modelling strategies for a liquid fuel transient calculation", Physics of Reactors Conference, 2020.

[24] A. Cammi et al., "A multi-physics modelling approach to the dynamics of Molten Salt Reactors.", Annals of Nculear Energy 38 (6), Pages 1356–1372, 2011.

[25] M. Tiberga, D. Lathouwers, J.L. Kloosterman, "A discontinuous Galerkin FEM multi-physics solver for the Molten Salt Fast Reactor", International Conference on Mathematics and Computational Methods applied to Nuclear Science and Engineering, 2019.

[26] J. Křepel, U. Rohde, U. Grundmann, F. P. Weiss, "DYN3d-MSR spatial dynamics code for molten salt reactors.", Annals of Nulcear Energy 6, Pages 449–462, 2007.

[27] J. Kópházi, D. Lathouwers, J.L. Kloosterman, " Development of a three-dimensional time-dependent calculation scheme for molten salt reactors andvalidation of the measurement data of the molten salt reactor experiment.", Nuclear Science Engineering 163 (2), Pages 118–131, 2009.

[28] S. Wang, A. Rineiski, W. Maschek, "Molten salt related extensions of the SIMMER-III code and its application for a burner reactor", Nuclear Engineering and Design 236, Issues 14–16, Pages 1580-1588, 2006.

[29] A. Lindsay et al., "Introduction to Moltres: An application for simulation of Molten SaltReactors", Annals of Nuclear Energy 114, Pages 530–540, 2018.

[30] M. Aufiero et al., "Development of an OpenFOAM model for the Molten Salt Fast Reactor transient analysis", Chemical Engineering Science, 111, Pages 390-401, 2014.

[31] M. Brovchenko et al., "Design-Related Studies for the Preliminary Safety Assessment of the MoltenSalt Fast Reactor", Nuclear Science and Engineering 175, Pages 329-339, 2013.

[32] M. Aufiero, M. Fratoni, and P. Rubiolo, "MonteCarlo/CFD Coupling for Accurate modelling of the Delayed NeutronPrecursors and Compressibility Effects in Molten Salt Reactors", American Nuclear Society Annual Meeting, 2017.

# Chapter 2

# Multiphysics modelling of Molten Salt Reactors

Liquid salt employed as a fuel carrier and primary coolant in Molten Salt Reactors (MSRs) introduces several fundamental differences in the modelling of such reactors. The fluid flow dynamics has a direct impact on the neutron kinetics; most notably, the liquid fuel introduces neutronics effects related to the flow turbulence, stagnation zones, and transport of delayed neutron precursors with the flow. Several largely used and validated codes for neutron transport and nuclear reactor kinetics modelling, e.g. MCNP [10], PARCS [25] or SERPENT [24], are intended mainly for solid fuel modelling and it is in many cases not so straightforward to modify these codes to be applied to liquid fuel modelling. Thus, in recent years efforts have been made towards the development of dedicated modelling tools for MSRs. The main common feature of these tools is the methods and solvers allowing for coupled neutronics and thermal-hydraulics multiphysics modelling of the MSRs with the impact of the liquid fuel on the neutronics being taken into account explicitly.

A common approach in the MSR research community is to implement simplified neutron transport solvers (e.g. based on the neutron diffusion approximation [26], or on the spherical harmonics methods [27]) often using the finite volume method (FVM) [11], e.g. the OpenFOAM C++ based FVM toolbox. As already mentioned in the introduction of this work, the motivation behind efforts in the MSR community towards the development of such customized solvers, is the attractive possibility to have a monolithic, single software suite with dedicated sub-solvers that will allow for a higher fidelity coupling compared to a more conventional approach of coupling two or more distinct software via information exchange interfaces.

In this chapter, following the aforementioned approach, a simplified solver for the neutron transport equation is implemented in OpenFOAM using the neutron diffusion approximation and assuming only one energy group in the energy discretization. This rather simple neutronics solver is coupled to a standard thermal-hydraulics solver available in OpenFOAM, which solves for single-phase, incompressible flow, adopting the Boussinesq approximation for density [6]. In order to gain confidence in the performance and the physical accuracy of the developed solver, two conventional and well-validated software: MCNP [10] for neutron transport and ANSYS CFX [5] for the fluid flow dynamics are used to compare the results obtained with the OpenFOAM solvers.

The modelling results using MCNP and ANSYS CFX can be used for the preliminary system analysis and to obtain an initial steady-state picture of the system, e.g. neutron flux distribution, temperature, and velocity fields [23]. On the other hand, MCNP is designed for solid fuel analysis and despite some attempts to facilitate intrinsic features of liquid fuel within MCNP [19], [20], this requires substantial modifications to the source code and it is not undertaken in this work. This means that when the simplified diffusion solver is compared to the MCNP results, e.g. the delayed neutron precursor transport effects cannot be compared, because MCNP does not account for it. For the same reason, the test cases facilitating the comparison of the two codes are not extended to transients, because the impact of delayed neutron precursor movement is of significant importance especially in fast transients [7].

The Chapter is organized as follows: a brief description of each single-physics solver (MCNP, ANSYS CFX, OpenFOAM thermal-hydraulics, OpenFOAM neutronics) is given. The coupling techniques are presented and explained. A brief system description is provided for the Molten Salt Fast Reactor (MSFR), which is used for the validation case. Finally, the results of the analysis of the MSFR steady-state operation are presented and discussed.

The main outcome of the modelling efforts described in this Chapter is included in the paper "Coupling Techniques for Multiphysics modelling of Molten Salt Reactors" (A. Nalbandyan, E.B.Klinkby, B.Lauritzen) published in the proceedings of the *G4SR-1* International Conference (November 2018, Ottawa, Canada) See Part II, Paper 1.

## 2.1 Multiphysics Coupling Techniques

The multiphysics analysis is not new to reactor modelling and usually implies coupling between different relevant codes for reactor physics, chemistry, and fuel performance, e.g. neutron kinetics, thermal-hydraulics, structural mechanics, and material chemistry. A thorough review of the multiphysics coupling concepts for reactor modelling is presented in [1]. Typically, two approaches to the multiphysics modelling are adopted: coupling of two (or more) distinctive physics codes via a dedicated interface [2] or development of a single framework with specific modules within it for the modelling of each physical phenomenon [3]. The first approach also referred to as a conventional Operator Splitting (OS) technique, is mostly first-order accurate in time and updates each relevant physics once per time step. This approach to the solution of coupled equations leads to either limited accuracy or to the necessity of applying a very small time step at increased computational costs. The second approach adopts implicit time integration schemes to ensure that all single-physics residuals converge within a time-step before moving on to the next time-step [4].

For the MSRs, where the fuel is simultaneously the primary coolant, the accuracy of coupling becomes a necessary prerequisite for the reactor modelling and safety studies.

In this Chapter examples of both using conventional software with an information passing interface as well as of developing a customized solver are presented:

- A solver is developed for the neutron transport equation adopting the one-speed (one energy group) neutron diffusion approximation, making use of the OpenFOAM C++-based library.
  - All neutron-kinetics constants required for the implementation of the solver are generated using Serpent Monte-Carlo software.
- This deterministic neutron transport solver is coupled to an OpenFOAM CFD

solver for one-phase, incompressible Navier-Stokes equations with Boussinesq approximation.

- The customized neutron transport solver and the CFD solver are compared to two validated conventional software: MCNP and ANSYS CFX.

## 2.2 Neutron Transport modelling

For a successful design of a nuclear reactor, it is essential to be able to describe the neutron field distribution as a function of space and time as well as to give the angular and the energy dependencies. The neutron transport through a medium is described by the linear Boltzmann equation known as the neutron transport equation. In its very generic form, this equation is given as follows:

$$\begin{aligned}\frac{1}{v}\frac{\partial\Psi(\mathbf{x},\boldsymbol{\Omega},t,E)}{\partial t}+\boldsymbol{\Omega}\cdot\nabla\Psi(\mathbf{x},\boldsymbol{\Omega},t,E)+\Sigma_t(E)\Psi(\mathbf{x},\boldsymbol{\Omega},t,E) = \\ \int_0^\infty\int_{4\pi}\Sigma_s(\boldsymbol{\Omega}'\cdot\boldsymbol{\Omega},E'\rightarrow E)\Psi(\mathbf{x},\boldsymbol{\Omega}',E',t)d\boldsymbol{\Omega}'dE' \\ +\frac{\chi_p(E)}{4\pi}\int_0^\infty\int_{4\pi}[1-\beta(E')]\nu\Sigma_f(E')\Psi(\mathbf{x},\boldsymbol{\Omega}',t,E')d\boldsymbol{\Omega}'dE'+\frac{1}{4\pi}\sum_{i=1}^{n}\chi_i(E)\lambda_iC_i(\mathbf{x},t)+\frac{1}{4\pi}Q(\mathbf{x},E,t)\end{aligned} \tag{2.1}$$

where $\mathbf{x}$ is the position vector, $\boldsymbol{\Omega}$ is the angular vector, $E$ is the energy, $t$ is time, $\Psi$ is the angular neutron flux, $\Sigma_t$ is the total cross-section, $\chi_p$ is the prompt neutron spectrum, $\beta$ is the delayed neutron fraction, $\Sigma_f$ is the fission cross-section, $\lambda_i$ is the $i^{th}$ delayed neutron precursor decay constant and $C_i$ is the $i^{th}$ neutron precursor group concentration. This first-order integro-differential equation states that the net change in neutron population over time can be represented as a balance between the neutron gain and neutron loss terms. Neutrons can be lost due to collision and leakage. These losses are represented by the second and the third terms on the left-hand side of Eq. 2.1. Neutron gain on the other hand can be either due to the fission, due to the scattering from a different energy group, or due to an external source. The production due to the fission has two contributing terms: direct production of prompt fission neutrons and production of delayed neutrons due to the $\beta$ decay of excited fission products. The gain terms are thus terms 1, 2, 3, and 4 on the right-hand side of Eq. 2.1.

The above equation is solved together with the equation for the delayed neutron precursor balance. For the precursor group $i$ it is as follows:

$$\frac{\partial C_i(\mathbf{x},t)}{\partial t}=-\lambda_iC_i(\mathbf{x},t)+\int_0^\infty\int_{4\pi}\beta_i(E')\nu\Sigma_f(E')\Psi(\mathbf{x},\boldsymbol{\Omega}',t,E')d\boldsymbol{\Omega}'dE' \tag{2.2}$$

The first term on the right-hand side represents the decay of the delayed neutron precursors and the second one is the source term which describes the production of new delayed neutron precursors. We will revisit Eq. 2.2. later in this Chapter, to introduce a fundamental difference between the delayed neutron precursor balance equation for the solid fuel as written above from the equivalent equation for liquid fuel.

Despite Eq. 2.1 describing the neutron field spatial and time distribution very accurately, it is not practical to solve this equation for large geometries, thus for the practical reactor physics problems approximations to this equation are considered. These mathematical approximation models are various, e.g.

- The diffusion approximation which is the simplest approximation to the transport theory and will be introduced and used later on in this work

- Other higher-order approximations such as the spherical harmonic ($P_N$) and simplified spherical harmonic ($SP_N$) approximations.
- The point kinetics approximation which is even simpler than the transport theory and the approximations used for its solution provided above and is applicable to time-dependent reactor problems.

The transport equation can also be solved without significant approximations by adopting a stochastic Monte-Carlo method [28].

In this Chapter two solution methods are presented: the diffusion approximation is used in a customized solver, whereas the Monte-Carlo method is represented by MCNP.

### 2.2.1 Deterministic modelling: The Diffusion Approximation

The diffusion approximation is a widely used technique in reactor analysis due to its simplicity and relative accuracy [18]. As the name suggests, the diffusion approximation treats the neutron transport in the medium similar to the diffusion of solvent in a solution. Depending on the number of energy groups considered, a differentiation is done between one-group, two-group, or multigroup diffusion approaches. If there are $N$ groups considered, for the group $g$ the differential equation to be solved is:

$$\frac{1}{v}\frac{\partial \Phi_g}{\partial t} = \nabla \cdot D_g \nabla \Phi_g - \Sigma_{ag}\Phi_g - \sum_{h=g+1}^{N} \Sigma_{g\to h}\Phi_g + \sum_{h=1}^{g-1} \Sigma_{h\to g}\Phi_h + (1-\beta_{tot})\chi_{pg}\sum_{h=1}^{N} \nu_h \Sigma_{fh}\Phi_h + \chi_{dg}\sum_{i=1}^{n}\lambda_i C_i \tag{2.3}$$

where $D_g$ is the diffusion constant, $\Sigma_{ag}$ and $\Sigma_{fg}$ are the group absorption and fission cross-sections, $\Sigma_{h\to g}$, $h$ is the index used to identify the energy groups other than the group $g$ itself, $\Sigma_{g\to h}$ are the group transfer cross-sections, $\beta_{tot}$ is the physical delayed neutron fraction, $\chi_{pg}$ is the prompt neutron emission spectrum and $\chi_{dg}$ is the delayed neutron emission spectrum. Depending on the cross-section libraries, the number of delayed neutron precursor groups is either six or eight.

Certain boundary conditions have to be defined for the solution of the diffusion equation [29], [30]:

- The diffusion theory does not yield valid results near the boundary surface between the diffusion medium and the outer space. To overcome this issue, a so-called vacuum boundary condition is specified at a distance $d$ from the real boundary of the diffusion medium, where the flux is set to vanish.

  $$d = 0.71 \times \lambda_{tr} = 0.71 \times 3 \times D = 2.13 \times D(cm) \tag{2.4}$$

  where $\lambda_{tr}$ is the neutron transport mean free path in the medium and $D$ is the medium diffusion coefficient [30]. Typically though, $d$ is very small compared to the reactor dimensions and can be neglected.
- Between two diffusion media the neutron flux and the neutron current have to be continuous across the boundary.

### 2.2.2 Stochastic modelling: The Monte-Carlo Method

Equation 2.1 can also be solved without adopting any significant approximation methods by using Monte-Carlo modelling technique. Monte-Carlo method is a set of mathematical algorithms utilizing multiple random samplings to obtain a solution for mathematical and physical problems that could be difficult to solve with other methods. The method has been developed in the late 1940s and used sampling based on pseudo-random numbers to solve statistically both probabilistic and deterministic problems. This numerical simulation method found its applications in physics, chemistry, biology, etc. In particle transport problems Monte-Carlo method found its unique place, being able to simulate the life of a particle from its birth until death, and answering such questions as determining the free path of the particle, the new direction, and energy after a collision, and so on. In contrast to the deterministic methods, the Monte-Carlo method does not solve the particle transport the equation for an average particle behavior, but rather simulates individual particles and detects some aspects of their average behavior. The advantages of the Monte-Carlo method for neutron transport include the possibility to simulate the exact geometry and energy-dependent physics and a very precise calculation of the neutron flux and power distribution in the reactor core. However, it should be noted that the Monte-Carlo method has high demand regarding the computational resources because a large number of particle histories has to be simulated to have acceptable statistical uncertainties in the simulation. Moreover, for the regions with sparse neutron population special techniques have to be applied to reduce the variance of Monte-Carlo calculations and to compensate for under-sampling [34].

## 2.3 OpenFOAM Multiphysics Solver

OpenFOAM is an open-source C++ library based on the Finite Volume Method (FVM) with broad capabilities in modelling complex flow, turbulence and heat transfer, etc [13]. Besides CFD applications it also has dedicated solvers for solid-mechanics and acoustics. The toolkit allows for relatively straightforward discretization and solution of differential equations using high-level operator notation and thus has been widely adapted for customized solver development. In particular for reactor physics analysis, several neutron transport solvers have been recently developed in OpenFOAM, mainly for the analysis of MSRs, but also for other reactors, e.g. the sodium-cooled fast reactor [16], [3]. Below a description of the thermal-hydraulics and neutronics solvers used in this work is given.

### 2.3.1 Thermal-Hydraulics Solver in OpenFOAM

For this work, the *buoyantPimpleFoam* heat transfer solver with Boussinesq approximation is adopted to model the liquid fuel flow in the reactor core. The Boussinesq approximation allows to avoid solving the full compressible Navier-Stokes equations and to treat the density as variable only in the buoyancy term of the momentum equation. The Navier-Stokes equations solved are:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{2.5}$$

where $\rho$ is the fluid density and $\mathbf{u}$ is the velocity.

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) = -\nabla p + \rho \mathbf{g} + \nabla \cdot \boldsymbol{\tau} \tag{2.6}$$

where $\rho$ is the fluid density, $\mathbf{u}$ is the velocity, $\mathbf{g}$ is the gravitational acceleration, $\boldsymbol{\tau}$ is the stress tensor and $p$ is the static pressure field.
Finally, the energy equation in terms of enthalpy is solved :

$$\frac{\partial \rho h}{\partial t} + \nabla \cdot (\rho \mathbf{u} h) + \frac{\partial (\rho K)}{\partial t} + \nabla \cdot (\rho \mathbf{u} K) - \frac{\partial p}{\partial t} = -\nabla \cdot q + \nabla \cdot (\boldsymbol{\tau} \mathbf{u}) + \rho \mathbf{g} \cdot \mathbf{u} + \rho H S \tag{2.7}$$

where $h$ is the enthalpy, $K$ is the kinetic energy, $\mathbf{u}$ is the velocity, $\rho$ is the density, $\mathbf{g}$ is the gravitational acceleration, $p$ is the pressure field and $HS$ is the heat source which is the term that comes from the solution of the neutronics module and can be presented as:

$$HS = E_f \times \Sigma_f \times \Phi (W/cm^3) \tag{2.8}$$

where $E_f$ is the heat produced per fission, $\Sigma_f$ is the fission cross-section and $\Phi$ is the neutron flux.

The solution algorithm adopted for thermal-hydraulics is the PIMPLE which is a hybrid approach composed of the *SIMPLE* combined with *PISO* algorithms [12]. This algorithm applies a predictor-corrector approach to the solution of the momentum and pressure equations. Within each time step, the momentum equation is solved first. Then the mass fluxes are calculated and the pressure equation is solved. After the pressure field is obtained, the mass flows are updated and the momentum equation is solved again. The procedure is repeated until convergence is reached. After all residuals drop below the user defined criteria, *PIMPLE* moves to the next time step [13].

### 2.3.2 Neutronics Solver in OpenFOAM

The neutronics solver discretizes and solves the one-group neutron diffusion equation together with the transport equation for the delayed neutron precursors:

$$\frac{1}{v}\frac{\partial \Phi}{\partial t} = \nabla \cdot D \nabla \Phi - \Sigma_a \Phi + \frac{1}{k_{eff}}(1-\beta)\nu \Sigma_f \Phi + \sum_{i=1}^{8} \lambda_i C_i \tag{2.9}$$

$$\frac{\partial C_i}{\partial t} = -\nabla(\mathbf{u} c_i) + \nabla \cdot D' \nabla C_i - \lambda_i C_i + \frac{1}{k_{eff}} \beta_i \cdot \nu \Sigma_f \Phi \tag{2.10}$$

where $v$ is the average neutron velocity, $\Phi$ is the scalar neutron flux, $D$ is the medium diffusion coefficient, $\Sigma_a$ is the one-group neutron absorption cross-section, $k_{eff}$ is the effective multiplication factor, $\beta$ is the total fraction of the delayed neutrons, $\Sigma_f$ is the one-group neutron fission cross-section, $\lambda_i$, and $C_i$ are the delayed neutron group-wise fractions and group-wise concentrations respectively, $D'$ is the molecular diffusion coefficient responsible for the level of turbulent convection of the delayed neutron precursors. As compared to the Equation 2.2, the new equation for delayed neutron precursors has two new terms: delayed neutron precursor transport with the velocity u field and turbulent convection.

The one-group cross-sections and the delayed neutron precursor data are generated in *Serpent 2.1.31* adopting *the ENDF-BVII.1* cross-section data library. Eight delayed neutron precursor groups are considered.

The one-group cross-sections are assumed to have a linear dependency from the fuel density and logarithmic dependency on the fuel temperature [21]:

$$\Sigma(T,\rho) = \frac{\rho}{\rho_0}(\alpha_T(log\frac{T}{T_0}) + \Sigma_0) \tag{2.11}$$

where $T$ is the actual temperature, $T_0$ is the reference temperature, $\Sigma_0$ is the cross-section at the reference temperature and $\alpha_T$ is the Doppler coefficient.

The $k_{eff}$ in the simulations in this Chapter is set to 1 , i.e. the system is critical at steady-state. More details about the neutronics solver can be found in Appendix A.

### 2.3.3 Coupling scheme and solution algorithm

The coupling scheme is shown in Figure 2.1 - the coupling procedure is as follows:

- The thermal-hydraulics solver is called and solves for the flow velocity, pressure, and temperature.
- The temperature and density fields are used to update the one-group cross-sections.
- The velocity field is passed to the neutronics solver for the transport of delayed neutron precursors.
- The neutronics solver is called and solves for the flux and the delayed neutron transport with the updated cross-sections and velocity field.
- A heat source is obtained.
- The heat source is updated in the thermal-hydraulics module.
- The cross-sections and the velocity field are updated and the neutronics problem is solved again.

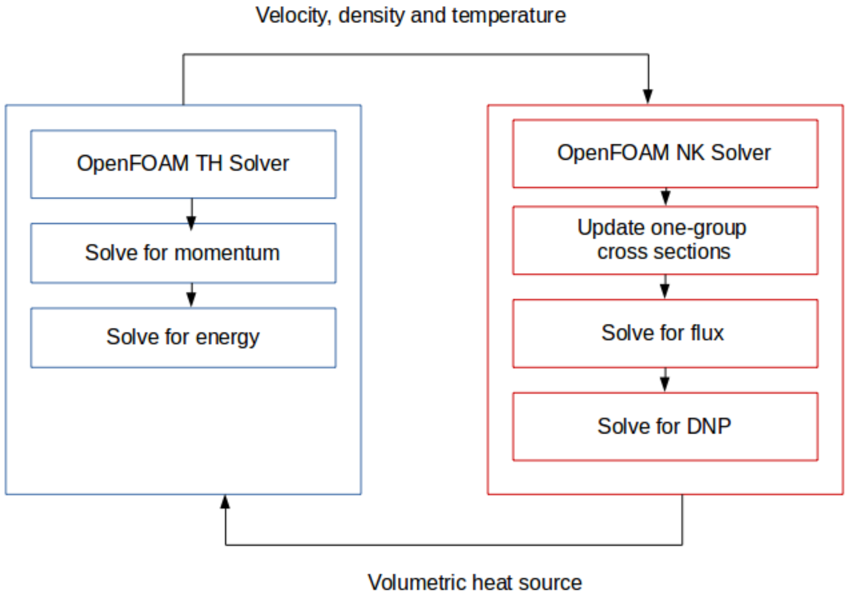


Figure 2.1: The coupling scheme between the thermal-hydraulics and neutronics solvers in OpenFOAM: the thermal-hydraulics problem is solved first and fixed point iterations are performed between the solvers within each time step to ensure the convergence of the temperature and neutron flux fields.

Similar to the iterative method used to converge the solution to the Navier-Stokes equations within each timestep, to fully resolve the non-linearities due to the coupling of different physics phenomena, a standard fixed point iteration method is adopted [15]. The iterations between the thermal-hydraulics and neutron kinetics solvers are repeated until convergence is reached.

The solvers used for differential equations are reported in Table 2.1.

| Equation | Solver |
|---|---|
| Energy | Preconditioned bi-conjugate gradient stabilized |
| Pressure | Preconditioned conjugate gradient |
| Velocity | Preconditioned bi-conjugate gradient stabilized |
| DNPs | Preconditioned bi-conjugate gradient stabilized |
| Neutron flux | Preconditioned conjugate gradient |

Table 2.1: Solvers used for the differential equations

As for the time scheme, the 2nd order implicit Crank-Nicolson scheme is used for time derivatives. For the Crank-Nicolson scheme, an off-centering coefficient $\phi$ is introduced, such that the integration of an ordinary differential equation $\dot{y} = f(y, t)$ from timestep $n$ to $n + 1$ can be represented as [14]:

$$y_{n+1} - y_n = \gamma h f_{n+1} + (1 - \gamma) h f_n \tag{2.12}$$

where $\gamma$ is a weighing coefficient equal to 0.5 for the standard Crank-Nicolson scheme and

$$\phi = \frac{1 - \gamma}{\gamma} = 1 \tag{2.13}$$

In OpenFOAM, the scheme is defined as:

```
    ddtSchemes
{
    default          CrankNicolson <coeff>
    ddt(phi)         CrankNicolson <coeff>;
}
```

where the coefficient defines the blending between Euler and Crank-Nicholson schemes.

## 2.4 ANSYS CFX and MCNP Solvers

### 2.4.1 ANSYS CFX CFD software

**ANSYS CFX** is a high fidelity, high accuracy CFD software used over a wide range of fluid flow modelling applications [5]. The software is based on the finite volume approach, wherein the governing equations of the respective physics, e.g. fluid flow, are integrated and stored in the finite volumes constructed by discretization (meshing) of the spatial domain. ANSYS CFX implores vertex-centered method for the discretization, where the variables are stored on grid points (Figure 2.2 ). This method is difficult to implement numerically and requires more storage memory as the variable storage grid is large, but it offers better performance than the more popular cell-centered FVM when it comes to distorted grids [11].

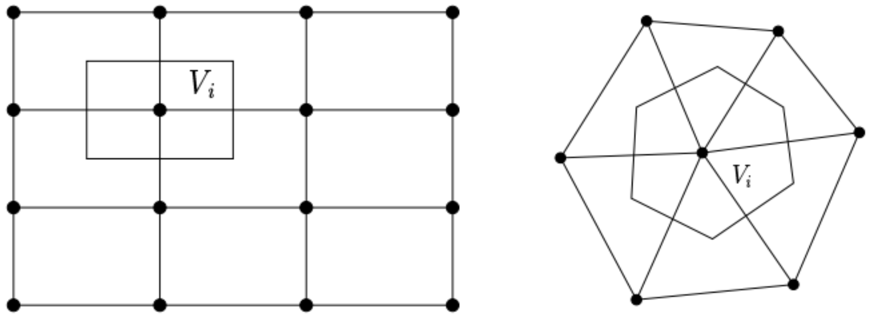


Figure 2.2: Vertex centered discretization on a quadrilateral (left) and a triangular (right) grids.

The code also has strong application in the safety analysis of nuclear reactors [8]. The solver is based on the solution of the unsteady Navier-Stokes equations in their conservation form: continuity, momentum, and energy equations are solved simultaneously together with two state equations for density and enthalpy used to close the system. The incompressible flow equations are:

$$\nabla \cdot (\mathbf{u}) = 0 \tag{2.14}$$

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) + \nabla p - \nabla \cdot \boldsymbol{\tau} - \rho \mathbf{g} = 0 \tag{2.15}$$

$$\rho \frac{\partial h}{\partial t} - \frac{\partial p}{\partial t} + \rho \nabla \cdot (\mathbf{u}h) - \nabla \cdot (\lambda \nabla T) - \nabla \cdot (\mathbf{u}\boldsymbol{\tau}) = 0 \tag{2.16}$$

$$dh = c_p dT + \frac{dp}{\rho} \tag{2.17}$$

$$c_p = c_p(T) \tag{2.18}$$

The equations (2.14)-(2.16) represent mass, momentum, and energy conservation, and the equations (2.17)-(2.18) are the state equations for incompressible flow. In the equations (2.14)-(2.18) $\rho$ is the fluid density, $\mathbf{u}$ is the flow velocity vector, $h$ is the total enthalpy, $\boldsymbol{\tau}$ is the stress tensor, $p$ is the pressure and $T$ is the temperature. After applying FVM on this set of differential equations, they are transformed into a set of algebraic equations which can be represented in a generic form as:

$$A[T] = b \tag{2.19}$$

where A denotes the solution matrix coefficients, T is the vector of unknowns and b is the equation right-hand side. ANSYS CFX uses a coupled solver, meaning that fully implicit methods are applied to solve the equations as a single system. The solver performs outer (time) iterations and inner (linearization) iterations. The solver applies a false time step for steady-state solutions, which acts as a guiding parameter helping the solution to converge faster to steady-state conditions [9]. The set of algebraic equations obtained after linearization of the differential equations is solved using the Algebraic Multigrid method, whereby the convergence process is accelerated by carrying out the iterations on a progressively coarser virtual mesh and then transferring the results back to a finer mesh [9].

The software also has an in-built CAD creation (Design Modeler or Space Claim) and meshing tools. Both structured and unstructured meshes are supported. The geometry can be exported directly into MCNP format.

### 2.4.2 MCNP Neutron Transport Code

The MCNP software is a general-purpose particle transport software widely used in reactor physics [10]. It is based on the Monte Carlo method; the solution to the integral form of the neutron transport equation (the Boltzmann equation) is obtained by simulating a large number of particle histories and recording user-requested aspects of their average behavior. MCNP6 is applied in the present work, with the $ENDF - BVII.1$ libraries for the neutron interaction cross-sections [31]. A $KCODE$ routine is applied to calculate the eigenvalues for the critical system and to obtain the neutron spectrum and flux. Additionally, for this work, the heat distribution in the problem geometry is obtained using a superimposed mesh tally of type A, available in MCNP for visualizing flux, dose, heat deposition, and other quantities on a cylindrical, spherical, or rectangular grid [33]. A tally multiplier is used to convert the flux output written by the tally to heat deposition and to normalize the results to an appropriate total power level.

### 2.4.3 Coupling mechanism

The coupling scheme is presented in Figure 2.3; first, the ANSYS CFX solver is called and solves for the fluid flow, e.g. the velocity, density, and temperature fields are obtained. In this simple steady-state coupling the velocity impact on the neutron transport is not considered; the temperature and the density fields are the only coupling parameters transferred from the CFD solution to the neutronics input. Subsequently, an MCNP standard *KCODE* calculation is performed to obtain the in-core heat distribution. The heat source is then mapped back into the CFD solver and the cycle is repeated. Several iterations are done until the convergence of the temperature field residuals drops below the user-specified criteria.

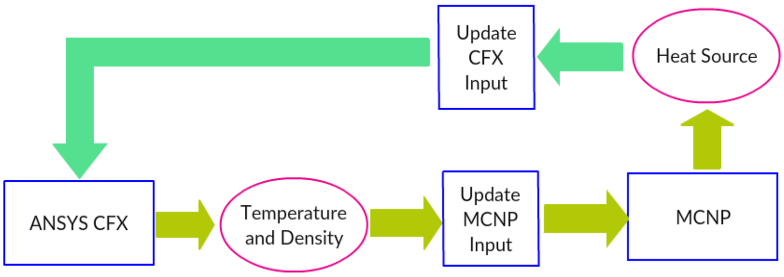


Figure 2.3: ANSYS CFX-MCNP Coupling Mechanism

## 2.5 Modelling of the MSFR Steady-State behavior

In this section, the two above-described coupling techniques are applied to investigate the steady-state behavior of the MSFR. It has to be underlined, that MCNP-ANSYS CFX coupling is used as a reference framework to assess the capability of the one group diffusion solver and the OpenFOAM thermal-hydraulics solver to describe the system correctly in terms of the main physical phenomena, as both MCNP and ANSYS CFX are validated and accepted commercial codes. This coupling technique however does not take into account any specifics related to the liquid fuel due to the limitations of the software packages.

### 2.5.1 The MSFR design description

The Molten Salt Fast Reactor (MSFR) concept has been chosen as a Generation IV system and is in active development since 2004 [22]. The outline of the reactor conceptual design is shown in Figure 2.4.

The MSFR is a fast spectrum reactor with fluoride salts adopted as a fuel carrier. The fissile inventory can be comprised of either $^{233}$U, $^{235}$U, natural U or trans-uranic (TRU) elements (Pu, Np, Am, Cm). The perspective of TRU usage makes the MSFR an attractive option as a nuclear waste burner. Moreover, both fuel scenarios include $^{232}$Th as a fertile element, allowing for the realization of the closed fuel cycle.

The main system components for the MSFR are briefly outlined below.

- **Reactor Core**

  Based on generic reactor design glossaries, the active core of the MSFR is the vol-

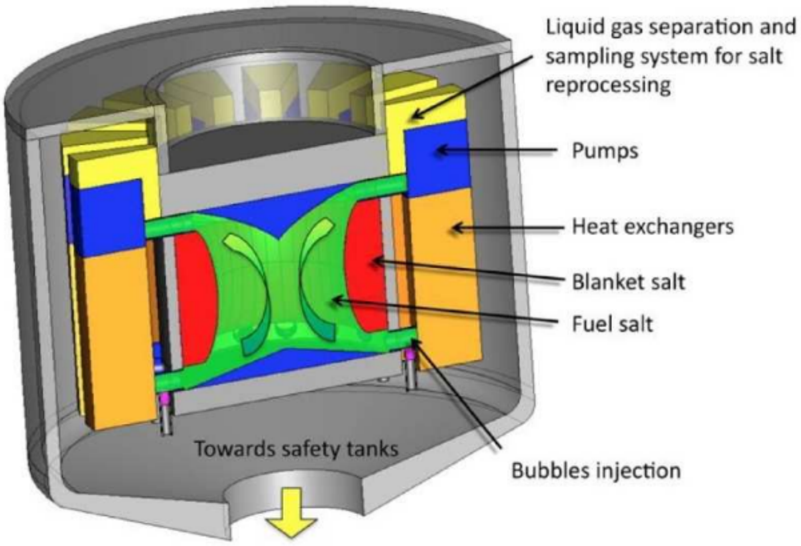


Figure 2.4: Conceptual design of the MSFR

ume of the fuel salt where most of the fission events occur and the bulk of the energy is released. The fuel salt is injected from the bottom of the core and is extracted at the top. The inlet temperature is 650 °C, the outlet temperature depends on the core power level, and the nominal power of 3 $GW_{th}$ is 750 °C. This is also close to the upper limit of the estimated range of the material structural integrity (800 °C).

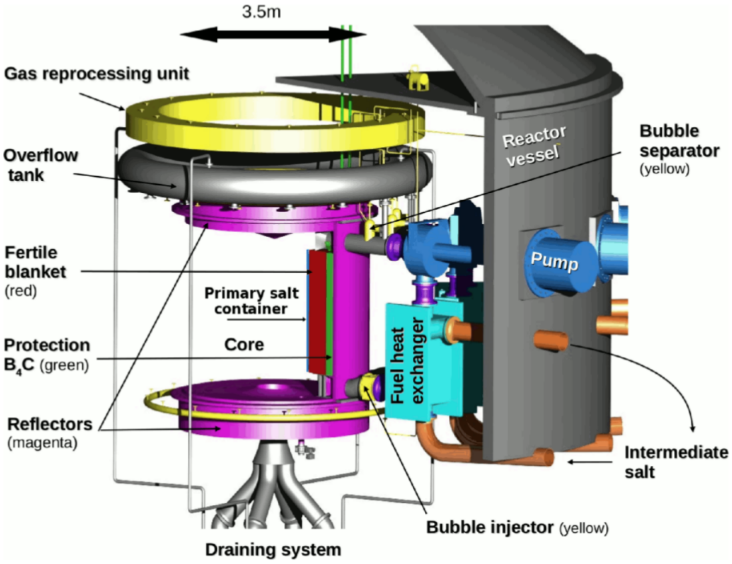


Figure 2.5: The preliminary MSFR design with cylindrical core

The initial parametric studies resulted in a cylindrical core shape, and in this chapter the modelling considers a cylindrical core shape depicted in Figure 2.5. However, as it will be shown, such a shape results in large zones of flow stagnation, which then leads to excess temperature and high thermal loads on the reactor vessel. Due to this the reactor core shape was optimized and a new, re-circulation-free shape has been proposed [17]. Chapter 3 of this work presents modelling results with the new core design and it is demonstrated that the stagnation and the resulting salt high-temperature issues are mostly resolved.

- **Heat Exchangers, Pumps and Other Structural Components**
  The reactor design envisages 16 fuel loops, each comprised of a pump and a heat exchanger. Each fuel loop removes roughly 187 $MW_{th}$ energy.
  Several heat exchanger designs were proposed and studied, the current candidate is the plate type heat exchanger [23]. When choosing an optimal heat exchanger design several factors, e.g. the pressure drop, the material corrosion/erosion considerations, the compactness of the design, etc. have to be considered. A final definition of the heat exchanger design for the MSFR will thus require substantial modelling and parametric studies.
  The pumps ensure sufficient flow through the core to have appropriate temperature rise and they affect the fuel circulation time and the core power level. No specific requirements are present for the pumps.
  Other components of the reactor are the upper and the lower core reflectors composed of Ni-based alloy, the fertile blanket with$^{232}$Th$F_4$ which also serves as a radial reflector for the neutrons, and finally the reactor vessel housing the core components, the piping, etc. The material composition for the Ni-based alloy is specified in Table 2.2 and is used throughout this chapter [23].

| Ni | W | Cr | Mo | Fe | Ti | C | Mn | Si | Al | B | P | S |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.79 | 0.01 | 0.08 | 0.0074 | 0.0063 | 0.0029 | 0.0029 | 0.0026 | 0.0025 | 0.0005 | 0.0003 | 0.0002 | 0.00004 |

Table 2.2: Alloy composition for the structural components (at. fraction)

- **Fuel Salt**
  Typically, fluoride or chloride-based salts are considered as fuel carriers for the molten salt reactors, with the fluoride-based salts having wider application than the chloride salts. This is mainly because fluorine chemistry has been more extensively studied and there are examples of fluorine salt behavior in the reactor core from the e.g. MSRE. Additionally, for the thermal spectrum reactors, fluoride salts are a logical choice, as chlorine hardens the neutron spectrum. A favorable scenario for using chloride salts is a fast spectrum reactor intended for burning actinides; actinides have a higher solubility in chloride salts and form lower melting point solutions as with fluoride salts. In the case of the MSFR, the fluoride salt is chosen as a fuel carrier based on several considerations such as stability under irradiation, the possibility of reprocessing, good compatibility with Th, and no production of radioisotopes that are difficult to handle, such as the $^{36}$Cl in case of the chloride salts.

  As mentioned before, both TRUs and U isotopes can be used in the initial fissile composition of the MSFR fuel. The generic fuel vector composition is 77.5% LiF and 22.5% ThF$_4$-UF$_4$, or 77.5% LiF and 22.5% ThF$_4$-TRUF$_3$. In this chapter, the $^{233}$U based fuel is considered. The fuel thermophysical properties are reported in Table 2.3. The initial fuel inventory is shown in Table 2.4.

| Quantity | Value at 973.5 $K$ |
|---|---|
| Density[g $cm^{-3}$] | 4125 |
| Dynamic viscosity $Pas$ | 0.01 |
| Thermal conductivity [ $Wm^{-1}K^{-1}$] | 1.009 |
| Calorific capacity $C_p$ [ $Jkg^{-1}K^{-1}$] | 1596 |

Table 2.3: Fuel salt (77.5% LiF and 22.5% $ThF_4$-$UF_4$) parameters calculated based on the formulas provided in [32].

| Fuel Salt | Li | F | $^{232}$Th | $^{233}$U |
|---|---|---|---|---|
| | 29 | 62.6 | 7.4 | 1 |
| Blanket Salt | Li | F | $^{232}$Th | |
| | 29 | 62.6 | 8.4 | |

Table 2.4: Fuel and blanket salt initial composition (mol %)

A constant online fuel reprocessing is envisaged for the MSFR. Two main systems are designed for this: the gaseous bubbling system for the removal of insoluble fission products, e.g. noble gases, and the batch reprocessing system, wherein a constant amount of the fuel salt (circa 40 l/day) is extracted from the core, cleaned, and the fissile elements are re-injected into the core. This chapter does not deal with the modelling of fuel reprocessing. For further details on the reprocessing systems of the MSFR, the reader is referred to e.g. [16].

The main parameters of the MSFR for the nominal operation considered in this chapter are summarized in Table 2.5.

| Parameter | Value | Unit |
|---|---|---|
| Thermal Power | 3 | GW |
| Inlet temperature | 650 | $°C$ |
| Outlet temperature | 750 | $°C$ |
| Fuel Salt volume | 18 | $m^3$ |
| Blanket Salt volume | 7.3 | $m^3$ |
| Core dimensions | Height-2.255, Radius-1.1275 | $m$ |
| Fuel recirculation time | 4 | $s$ |

Table 2.5: Main parameters of the reference MSFR design

### 2.5.2 Model geometry and setup

A $3D$ geometry is adopted in this work, with a cylindrical core geometry ($D = h = 2.25m$). The geometry is depicted in Figure 2.6: $1/16th$ of the entire reactor model is considered, thus one loop is modelled with the in-core fuel and the heat exchanger. The total salt volume is  $1.2m^3$, with an in-core to total salt volume ratio of approx. 0.5.

The unstructured tetrahedral mesh is used. The mesh is constructed using ANSYS Workbench, and then translated into OpenFOAM format and used for both the CFD and the diffusion neutronics solver.

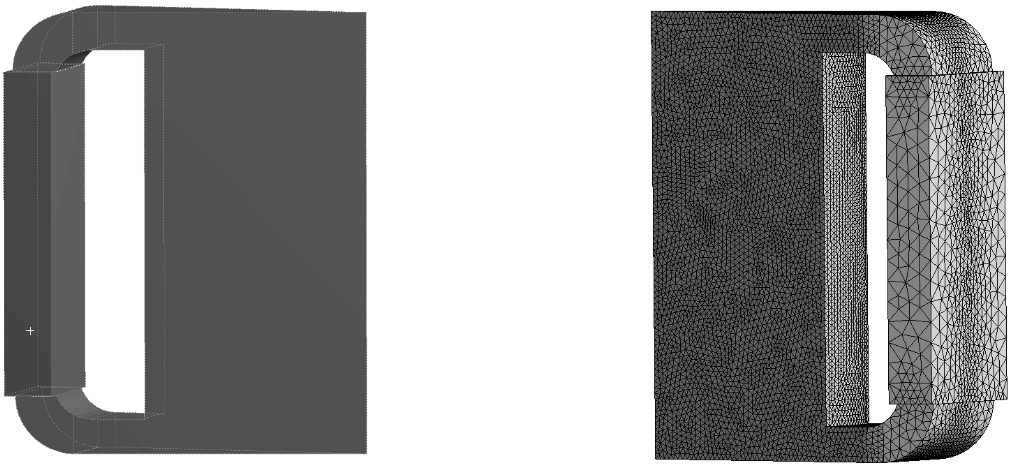


Figure 2.6: MSFR single loop geometry (left) and an example CFD mesh (right)

The MCNP neutronics model adopts the same geometry as the ANSYS CFX and OpenFOAM; an in-built capability of ANSYS CFX allows for an easy geometry translation into MCNP supported format.
To calculate the volumetric heat source in MCNP, a coarser mesh is superimposed over the geometry (Figure 2.7).

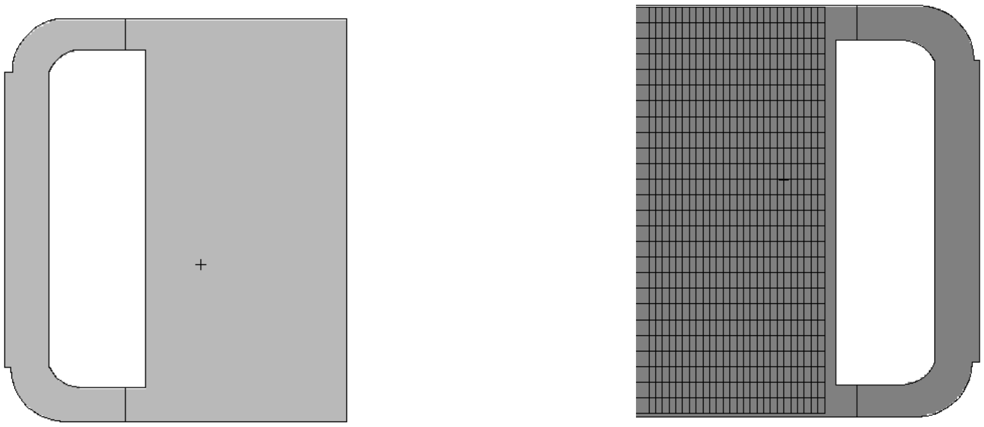


Figure 2.7: MCNP geometry (left) and the in-core superimposed mesh tally (right)

### 2.5.3 Model assumptions and boundary conditions

The assumptions made for the modelling and the boundary conditions are outlined below.

- In ANSYS CFX, the pump is modelled as a general momentum source added via the user defined functions option to the continuity equation;

$$S = -C \times (\mathbf{u} - \mathbf{u_{ref}}) \tag{2.20}$$

  where $C$ is the momentum source coefficient, $\mathbf{u}$ is the velocity vector and $\mathbf{u_{ref}}$ is the reference velocity value one wants to achieve in the domain.

- The heat exchanger is modelled as a heat sink to achieve 650 °C inlet temperature. This is again achieved by using the user defined functions option.
- The standard $k-\epsilon$ turbulence model is used.
- The heat source for the fuel is coming from neutronics module, in this case from the MCNP superimposed mesh tally results and is added as an external source to the energy equation via a user defined function .

As for the boundary conditions:

- A symmetry boundary conditions are applied on the symmetry cut planes for both the velocity and the temperature.
- A no-slip boundary condition is applied to the velocity on the core walls and an adiabatic wall boundary condition is applied to the temperature to achieve a thermal insulation of the domain.
- In OpenFOAM, the pump is modelled as a porous source using the Darcy-Forchheimer model [5].

$$\nabla P = (\mu D + \frac{1}{2}\rho F \mathbf{u})\mathbf{u} = \mu D \mathbf{u} + \frac{1}{2}\rho F \mathbf{u}^2 \tag{2.21}$$

  where $D$ and $F$ are the viscous and inertial loss coefficients and the $d$ and $f$ coefficients for OpenFOAM can be calculated as:

$$d = D\mu \tag{2.22}$$

$$f = \frac{\rho F}{2} \tag{2.23}$$

  where $\mu$ is the dynamic viscosity and $\rho$ is the density.
- The heat exchanger is modelled as a heat sink:

$$Sink = \gamma \times (T - T_0) \tag{2.24}$$

  where $\gamma$ is a volumetric heat transfer coefficient, $T$ is the actual fuel temperature and $T_0$ is the inlet temperature.
- The realizable $k-\epsilon$ turbulence model is used.
- The heat source is obtained from the neutronics solver and is added to the energy equation.

The boundary conditions are chosen as follows:

- Symmetry boundary conditions on the symmetry cut planes are applied, just as in the ANSYS CFX.
- For the remaining walls a no-slip boundary condition is applied to the velocity, and an adiabatic boundary condition is applied to the temperature, again identical to the ANSYS CFX boundary conditions.

Thus, for the thermal-hydraulics part, the ANSYS CFX and the OpenFOAM models are adopting identical approaches regarding the implementation of the heat and momentum sources and the boundary conditions. The only difference is the adoption of the realizable $k-\epsilon$ turbulence model in OpenFOAM as compared to the standard $k-\epsilon$ model in ANSYS CFX. The realizable $k-\epsilon$ model has a superior behavior compared to the standard model when applied to the modelling of complex flow patterns with vortexes and stagnation zones. It was thus of interest to apply this model for the analysis of the MSFR, to see whether the flow distribution is more uniform with this model. Unfortunately, the model is not available for the ANSYS CFX version used in this work.

MCNP criticality calculation KCODE with the parameters specified in Table 2.6 is used for flux calculations. $2 \times 10^6$ neutron histories are simulated in MCNP resulting in 105 pcm uncertainty on the $k_{eff}$. A cylindrical mesh tally of type A is superimposed over the core in MCNP to score the heat deposition in the core. For the neutron flux, vacuum boundary conditions are used in the axial direction and reflective boundary conditions for the symmetry planes.
All constants (delayed neutron precursor group decay constants, half-lives, cross-section and diffusion coefficient values) used in the OpenFOAM solver are generated in Serpent using the same $ENDF - BVII.1$ data library as in MCNP calculations. The flux calculated by the one-group neutronics solver is used to calculate the power density and to pass it to the temperature equation as a heat source.

| Parameter | Value | Default value |
|---|---|---|
| nsrck (number of source histories per cycle) | 10000 | 1000 |
| rkk (initial guess for $k_eff$ | 1 | 1 |
| ikz (number of histories to skip) | 100 | 30 |
| kct (total number of cycles) | 200 | 200 |

Table 2.6: MCNP KCODE Parameters

### 2.5.4 Results and discussion

In Figure 2.8 the velocity field distribution and the velocity plots for the fuel salt flow are depicted, as predicted by ANSYS CFX CFD solver (Figure 2.8$a$) and the OpenFOAM CFD solver (Figure 2.8$b$). Both solvers predict a large flow stagnation zone near the core wall, where the flow stream is trapped and re-circulation occurs. As it is shown later, this causes an undesirable temperature hot spot to be formed, which could cause thermal overloading on the core structural materials and in general, is outside the range of the operational temperatures desired for the MSFR. This characteristic flow pattern which is a direct result of the core design became one of the main driving factors for the MSFR core design to be modified from a cylindrical to a toroidal shape [17].

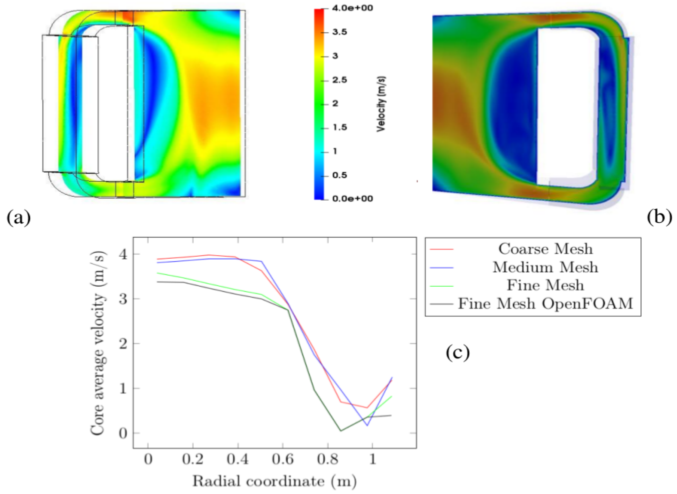


Figure 2.8: (a) Velocity field ANSYS CFX (b) Velocity field OpenFOAM (c) Mesh impact on the average in-core velocity distribution

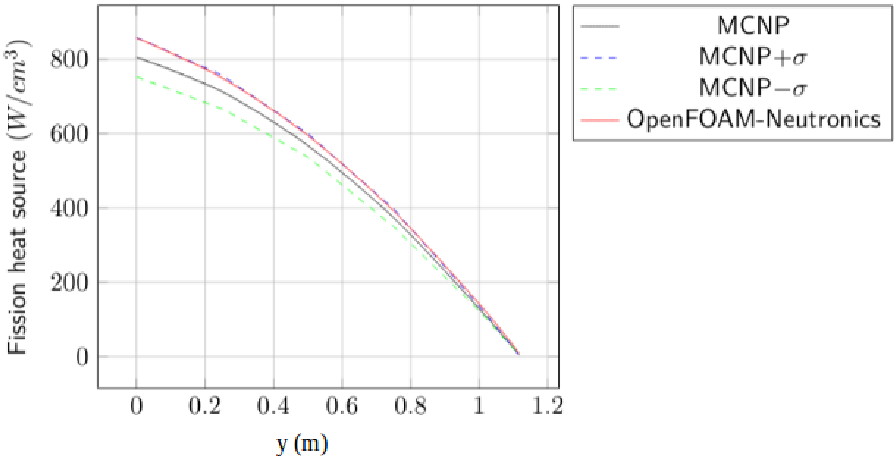


Figure 2.9: Heat source radial distribution: MCNP results with 1$\sigma$ confidence interval and the OpenFOAM 1 group diffusion solver results

The mesh impact on the in-core velocity distribution is shown in Figure 2.8c. The results are provided for coarse, medium, and fine meshes in ANSYS CFX and for the fine mesh only in OpenFOAM, because the mesh independence study was performed in ANSYS CFX only, and then the converged grid was translated to and used in OpenFOAM. It is evident, that the grid coarseness has a significant impact on the results; the mesh converged velocity field (542246 elements) is on average 20% lower than the one obtained with the coarse mesh.
When comparing the results obtained for the same fine mesh in ANSYS CFX and OpenFOAM, the profiles show a relative difference of around 5 % in the central part of the core and the recirculation zone. The difference attributed to the turbulence model; the realizable $k$−$\epsilon$, which is not available for ANSYS CFX, is more stable and better adapted to resolve flows with strong gradients and recirculation.

In Figure 2.9 the heat distribution in the core is shown obtained from MCNP and the OpenFOAM solver.
As a reminder, the volumetric heat source can be written as $HS = \Phi \times \Sigma_f \times E_f$, where $\Phi$ is the neutron flux, $\Sigma_f$ is the one-group fission cross-section and finally, $E_f$ is the energy released per fission.

Acceptably good agreement between the rather simple one-group diffusion model and the MCNP calculation can be appreciated. The one-group diffusion solver is overestimating the MCNP results by an average of 6%. This can be attributed to the fact that a one-group cross-section value is used in the OpenFOAM, generated by Serpent and the fission heat source is calculated after the neutronics solver solves for the flux with a very simple one-group approach, whereas MCNP directly tallies the fission heat source on a superimposed mesh and is, in general, relying on very accurate calculation methods.

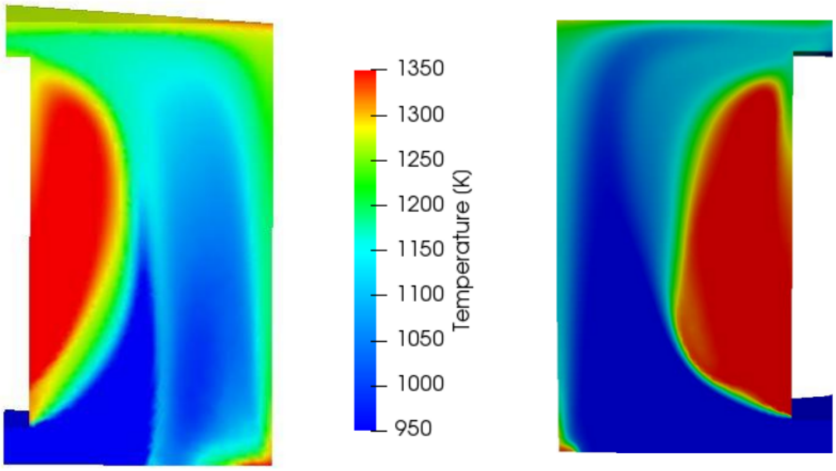


Figure 2.10: In-core temperature distribution obtained by ANSYS CFX and OpenFOAM(right).

Finally, the temperature distribution in the core is shown in Figure 2.10. Due to the large stagnation zone, the temperature rises above the expected range for the MSFR steady-state operation and reaches $1350K$. As already discussed for the velocity, the cylindrical core shape results in a specific flow pattern leading to the flow stagnation and a consequent formation of an overheated fuel region which can cause unnecessary thermal loads on the reactor vessel structural materials as well as have undesirable effects on the neutronic performance of the fuel.

Some differences in the temperature profile between ANSYS CFX and OpenFOAM are noticeable; even though the average in-core temperatures and the temperature differences between the inlet and the outlet are close, there is a larger hot-zone predicted by OpenFOAM. This is attributed to the fact that the fission heat source is slightly larger for OpenFOAM calculations, and to the fact that OpenFOAM predicted lower velocity in the recirculation zone.

## 2.6 Conclusion

In this Chapter, a preliminary coupled thermal-hydraulics neutronics analysis of the MSFR at steady-state condition was performed using a simple one-group neutron diffusion solver implemented in OpenFOAM and coupled to an OpenFOAM standard CFD solver.To validate the results, it was decided to compare the OpenFOAM CFD results (velocity and temperature distributions) to simulation results obtained with a largely validated commercial CFD code; ANSYS CFX. Additionally, the fission heat source distribution in the core obtained from the one-group diffusion solver in OpenFOAM has been compared to the results in MCNP, a widely used and validated neutron transport software based on the Monte-Carlo method. To facilitate the comparison, the heat source obtained from MCNP was given to ANSYS CFX and the average fuel temperature and density from ANSYS CFX were used to update the MCNP input.

The comparison of the results between the respective software is satisfactory, taking into account that different approaches to the solution of the neutronics problem were adopted, one very precise and the other one very simple.

This coupling technique does not take into account any impact of the fuel flow on the neutronics, e.g. delayed neutron precursor transport or possible retention of the delayed neutron precursors inside the core due to the flow stagnation near the radial wall.

The main conclusion for the modelling efforts described in Chapter 2 is:

- Despite the simpleness of the neutron transport solver developed in OpenFOAM, it seems to agree reasonably with MCNP results.
- The velocity and temperature fields from OpenFOAM agree with the ANSYS CFX results, thus the coupling is realized correctly.

As future work, several improvements could be implemented to further develop the solvers described in this Chapter:

- Develop a multi-group diffusion solver to replace the one-speed approximation used here.
- Test the neutronics solver developed in OpenFOAM for transient simulation regime.
- Verify the solver by comparing it to other multiphysics software suits that have the same capabilities and use the same modelling approach for the neutronics, i.e. the diffusion approximation.

# Bibliography

[1] M. N. Avramova, K. N. Ivanov. “Multiphysics modelling of Nuclear Reactors”, McGraw-Hill Education, 2017.

[2] D. Lee, T.J. Downar, A. Ulses, “Analysis of the OECD/NRC BWR turbine trip transient benchmark with the coupled thermal-hydraulics and neutronics code TRAC-M/PARC”, Nuclear Science and Engineering 148, No. 2, Pages 291-305

[3] C. Fiorina, K.Mikityuk, "Application of the new Gen-Foam multi-physics solver to the European Sodium Fast Reactor and verification against available codes", Proceedings of ICAPP 2015.

[4] J. C. Ragusa, V. S. Mahadevan, “Consistent and accurate schemes for coupled neutronics thermal-hydraulics analysis”, Nuclear Engineering and Design 239, Pages 566-579

[5] https://www.ansys.com/products/fluids/ansys-cfx

[6] OpenFOAM User Guide, v2006, https://www.openfoam.com/documentation/guides/latest/doc/index.html

[7] D. Wooten, J.J. Powers, "A Review of Molten Salt Reactor Kinetics Models", UC Berkley ORNL, 2018.

[8] M. Aghzaryan, T. Malakyan, A. Nalbandyan, A. Amirjanyan, "CFD modelling of mixing phenomena for pressurized thermal shock analysis of the downcomer of WWER-440. ", Proceedings of Nureth-16, 2015.

[9] ANSYS CFX Solver Theory Guide, ANSYS, Inc., 2006

[10] MCNP, https://mcnp.lanl.gov/

[11] F. Moukalled, L.Mangani, M.Darwish "The Finite Volume Method in Computational Fluid Dynamics", Springer, 2016. "Calculation of effective point kinetics parameters in the Serpent 2 Monte Carlo code", Annals of Nuclear Energy 65:272, 2014.

[12] J. H. Ferziger, M. Perić, "Computational Methods for Fluid Dynamics.", Springer-Verlag 1996.

[13] T. Holzman, "Mathematics, Numerics, Derivations and OpenFOAM", Holzmann CFD, Release 7.0.

[14] S. Seng, C. Monroy, S. Malenica, "On the use of Euler and Crank-Nicolson time-stepping schemes for seakeeping simulations in OpenFOAM, VII International Conference on Computational Methods in Marine Engineering", MARINE Conference Proceedings, 2017.

[15] P. B. Bailey, L. F. Shampine, P. E. Waltman, "Nonlinear Two Point Boundary Value Problems", Mathematics in Science and Engineering 44, Pages 21-49, 1968.

[16] M. Aufiero, M.Brovchenko, A. Cammi, I. Clifford, O.Geoffroy, D. Heuer, A. Laureau, M. Losa, L. Luzzi, E. Merle-Lucotte, M.E. Ricotti, H. Rouch, "Calculating the effective delayed neutron fraction in the Molten Salt FastReactor: Analytical, deterministic and Monte Carlo approaches", Annals of Nuclear Energy 65, Pages 78-90, 2014.

[17] H. Rouch, O. Geoffroy, P. Rubiolo, A. Laureau, M. Brovchenko, D. Heuer, E. Merle, "Preliminary thermal-hydraulic core design of a Molten Salt Fast Reactor (MSFR)", Annals of Nuclear Energy 64, Pages 449-456, 2014.

[18] J. C. Lee, "Nuclear Reactor Physics and Engineering", Pages 77-98, John Wiley & Sons, Ltd, 2020.

[19] J. Kophazi et al., "Monte Carlo calculation of the effects of delayed neutron precursor transport in molten salt reactors", PHYSOR Conference proceedings, 2004.

[20] J. Kophazi et al., "MCNP based calculation of Reactivity Loss due to o Fuel Circulation in Molten Salt Reactors", JAERI Conference proceedings, 2013.

[21] A. E. Waltar et al., "Fast Spectrum Reactors", Springer, 2012.

[22] H. Boussier et al., “The Molten Salt Reactor in Generation IV: Overview and Perspectives”, Proceedings of the Generation4 International Forum Symposium, 2012.

[23] M. Brovechnko et.al., "Design-Related Studies for the Preliminary Safety Assessment of the Molten Salt Fast Reactor", Nuclear Science and Engineering, 2013.

[24] J.Leppänen et al., "The Serpent Monte Carlo code: Status, development, and applications in 2013." Annals of Nuclear Energy 82, Pages 142-150, 2015.

[25] T. Downar, Y. Xu, V. Seker, "PARCS v3.0 U.S. NRC Core Neutronics Simulator", User Manual, March 2010.

[26] S. Marguet, "The Physics of Nuclear Reactors", Springer 2017.

[27] W. Kofink, "Studies of the spherical harmonics method in neutron transport theory", ORNL, May 1957.

[28] D. G. Cacuci (ed.), "Handbook of Nuclear Engineering", Springer, 2010.

[29] W. Stacey, "Nuclear Reactor Physics", 2nd Edition, ISBN 978-3-527-40679-1, 2007.

[30] J. R. Lamarsh, A. J. Baratta "Introduction to Nuclear Engineering", 3rd Edition, ISBN 0-201-82498-1, 2001.

[31] Evaluated Nuclear Data Library File, URL https://t2.lanl.gov/nis/data/endf/

[32] V. Ignatiev, O. Feynberg, A. Merzlyakov, "Progress in Development of MOSART Concept with Th Support", Proceedings of ICAPP 2012, 2012.

[33] D.B. Perlowitz, "MCNP 6 User Manual", 2011.

[34] J. Spanier, E.M. Gelbard, "Monte Carlo Principles and Neutron Transport", ISBN-13: 978-0486462936, 1969.

# Chapter 3

# Coupled Monte Carlo - CFD modelling of Molten Salt Reactors

In this chapter, a coupling technique implemented between the Serpent Monte Carlo code v2.30 [1] and an OpenFOAM C++ [10] based toolbox v18.06 is presented. The coupling interface is implemented using Python routines and it is based on exchanging relevant coupling terms between the Serpent neutron transport modelling software and the OpenFOAM CFD solver. The impact of the liquid fuel on the delayed neutron precursor (DNP) field is taken into account by using the velocity field from the CFD solver to shift the DNP production positions in the Serpent neutronics calculations. The coupling method is validated by comparison to the experimental data and the model calculations for the Molten Salt Reactor Experiment.

This chapter builds on the fact that in chapter 2 the applicability of the OpenFOAM CFD solvers to the analysis of the liquid fuel in MSRs has been demonstrated; OpenFOAM allows for high fidelity CFD modelling of fluid flow dynamics and it is well suited for implementing user-tailored features. Moreover, a dedicated multiphysics interface is already available between Serpent and OpenFOAM allowing for the exchange of non-uniform temperature, density, and fission power distributions between the two software [4] which is well suited to be used in this work. Compared to the diffusion solver presented in chapter 2, the differences of this coupling approach are:

- Serpent relies on Monte Carlo techniques for the solution of the neutron transport equation and thus, it presents the most accurate way of solving the neutron transport equation. No approximations and assumptions are used in the method applied to the solution of the neutron kinetics problem, in contrast with the diffusion approximation used in chapter 2.
- Each of the respective software codes for CFD and neutronics can be used as they are, without modification of the source codes. This limits the development and debugging process to the coupling interface only.

The main issue of concern when using Monte-Carlo/CFD coupling is the computational cost of the simulations which is high due to the statistics requirements of the Monte-Carlo methods and can also be substantially high for fine mesh CFD analysis. Moreover, if a mesh grid is also used in the neutronics calculations, this makes the neutron tracking process more time-consuming. However, there are some remedies for the issue of the computational time; the Monte-Carlo computational cost can be reduced by using pre-calculated neutron propagation in the reactor, e.g., the Transient Fission Matrix approach which has been developed recently [3]. As for the CFD part, usually modern

CFD software have very good parallel computing capabilities which can help run large calculations in a reasonable time.

On the other hand, the method described in this chapter and featuring two standalone software coupled via some information transfer routines can be numerically outperformed by more monolithic coupling strategies featuring solution of both the neutronics and the CFD problems by the same software suite. The method described in chapter 2 is one example of such an approach; although simple in implementation regarding the neutronics part, it featured two solvers implemented in the same software frame and utilizing the same spatial and temporal discretization methods and solution algorithms with a possibility to easily iterate between the modules to reach convergence. chapter 2 also gave an overview of other high-fidelity software packages developed for MSR studies that allow for very good numerical stability and coupling accuracy due to their implementation strategy. Recently some work has been done also to allow such a high-accuracy internal coupling between Monte-Carlo and CFD codes; in [25] the authors present a technique of coupling some OpenFOAM routines to Serpent at source code level allowing for efficient tracking and shifting of DNPs.

However, the examples of coupling Monte-Carlo codes to CFD for MSR analysis are not as many as the implementation of other, more computationally light-weight approaches to neutron kinetics modelling.

The method presented in this chapter aims to realize an external coupling between Serpent and OpenFOAM and to investigate whether this approach demonstrates good performance when applied to the analysis of liquid fuel MSRs, specifically the MSRE.

The structure of the chapter is as follows: first, the neutronics module (Serpent Monte-Carlo code) and the CFD module (conjugate heat transfer solver in OpenFOAM) used in this work are described followed by a detailed description of the coupling approach implemented between OpenFOAM and Serpent. Then a description is given for the Molten Salt Reactor Experiment (MSRE) which has been chosen as a test case in this chapter. This is followed by test model description, result presentation, and discussion. The coupling is validated against the data available from the MSRE. Both steady-state and transient scenarios are modelled.

The main results of this chapter are presented in the "Coupled Neutronics/Thermal Hydraulics Assessment of Graphite Moderated Molten Salt Reactors", (A. Nalbandyan, E.B. Klinkby, B. Lauritzen, J. Groth-Jensen, R. Steyn, Proceedings of the 18th International Topical Meeting on Nuclear Reactor Thermal Hydraulics. American Nuclear Society, p. 5342-5355. See Part II, Paper 2 of this thesis).

## 3.1 Neutron transport modelling in Serpent

Serpent is a continuous energy Monte Carlo code intended for particle transport modelling. It is a three-dimensional software, operates in steady-state as well as the transient mode, and can be used for a variety of reactor physics applications, such as criticality and burnup calculations, transient analysis, and validation of deterministic codes [1]. The neutron tracking in Serpent is a combination of delta-tracking and surface-tracking approaches, with the former being used by default and the latter only in presence of strong absorbers in the geometry [2]. Serpent estimates a reaction rate by scoring reaction rate estimates over the problem geometry and energy domain during the simulation:

$$R = \frac{1}{V}\int_V \int_E f(r,E)\Phi(r,E)d^3rdE \tag{3.1}$$

$$R_{implicit} = \sum_{i=1}^{N} f_i\Phi_i \tag{3.2}$$

where $R$ is the reaction rate, $V$ is the volume of the domain where the simulation is carried out, $E$ is the energy domain of the problem, $f(r,E)$ is the response function which can be e.g. a cross-section of interest, such as total macroscopic fission or absorption cross-section and $\Phi(r,E)$ is the neutron flux and $R_{implicit}$ is the implicit reaction rate estimate which is scored based on the flux estimate and the response function and can be interpreted as an expected rate of occurrence for a specific reaction. There are also analog estimators available in Serpent which directly count the number of events of the particular type. As in any Monte-Carlo-based code, results obtained in Serpent have statistical uncertainties and several points have to be kept in mind when setting up simulations to obtain statistically trustworthy results. In particular, in traditional k-eigenvalue calculations for a steady-state fission chain reaction, a sufficient number of neutrons per generation has to be simulated and a sufficient number of cycles has to be skipped to obtain converged source distribution before starting to score reaction estimators [5].

One of the features available in Serpent and used in this work is the inbuilt multiphysics interface which enables temperature and density field exchange with external solvers [4]. This interface allows for separation of the density and temperature information from the main Serpent geometry information [6].In this work, the multiphysics interface definition is based on the unstructured OpenFOAM mesh. As finite volume meshes can in general have a large number of elements, Serpent applies an adaptive search mesh that is superimposed over the imported OpenFOAM mesh which speeds up the process of retrieving temperature and density data at interaction points [7]. The unstructured mesh-based geometry and interface are defined as follows:

```
1) 9 1 2 % interface card 9
2) 1 ./volpower % volumetric power distribution as output
3) -1 900 % nominal density to be used and temperature
4) 5 3 2 2 2 % adaptive search mesh parameters
5) poly mesh/points % adaptive search mesh folder
6) poly mesh/faces
7) polyMesh/owner
8) polyMesh/neighbour
9) materials.txt
10) rhos.txt 1 % non-uniform density file
11) Ts.txt 1 % non-uniform temperature file
12) -1 % mapping each cell to a unique power tally bin
```

Some recent applications of Serpent multiphysics capabilities include coupling to fuel performance codes, depletion calculations, coupling to thermal-hydraulics, and solid mechanics [8], [9].

Another feature of Serpent that is extensively utilized in this work is the generation of prompt and delayed neutron sources. To retrieve these sources, the following line has to be included in the input:

```
set savesrc FILEPATH [PN PP NX NY NZ]
```

The PN and PP values are used to adjust the probability of storing an individual neutron or a DNP to the source, NX, NY, and NZ specify the Cartesian mesh size used to tally the DNPs. There are two ways in Serpent to track the DNPs: point-wise and mesh-based. If the first one is used, there is no need to specify a grid size different from the default 1 × 1 × 1. If, however, one wants to use the concentration of DNPs per grid cell rather than the absolute number of the DNPs produced in the domain, then customized Cartesian mesh can be defined.
The source generation routine writes four files:

- *source.main* - An ASCII list with general information on the saved neutron and DNP sources.
- *source.live* - A binary list of neutrons saved randomly at different interaction points.
- *source.prec* - An ASCII list of delayed neutron precursors at the beginning and at the end of simulation, based on mesh: e.g., if only one cell was specified (default option), this file will look as follows:

```
0 0 0 0.000000E+00 0.000000E+00
0 1 0 0.000000E+00 0.000000E+00
0 2 0 0.000000E+00 0.000000E+00
0 3 0 0.000000E+00 0.000000E+00
0 4 0 0.000000E+00 0.000000E+00
0 5 0 0.000000E+00 0.000000E+00
0 6 0 0.000000E+00 0.000000E+00
0 7 0 0.000000E+00 0.000000E+00
1 0 0 1.366033E+18 1.535576E-06
1 1 0 2.654902E+18 2.601758E-06
1 2 0 1.167101E+18 2.049288E-06
1 3 0 7.528236E+17 2.247747E-06
1 4 0 5.342619E+17 2.012693E-06
1 5 0 6.995189E+16 2.345596E-06
1 6 0 2.668216E+16 2.147624E-06
1 7 0 3.382217E+15 1.257258E-06
```

  where the first column indicates the beginning (0) and the end (1) of the simulation interval, the second column is the number of the DNP groups (0-7), the third column is the total population of the DNPs, and the last column is the estimated relative error.
- *source.precpoints* - A binary list of the DNPs according to their birth coordinates. There are nine columns in this file; the $x, y, z$ coordinates where the DNP was scored, four parameters that are not used with default values of $1, 0, 0, 1$, the weight of the precursor, and the precursor group. As an example, the first three lines of a generic *source.precpoints* file after decoding from binary format is shown, where the birth coordinates of the DNP are given for three DNPs from group 5, 4 and 3:

```
55.47,89.48,66.56,1.0,0.0,0.0,5.634e-05,1.51,5.0
58.47,80.51,56.93,1.0,0.0,0.0,0.00147,3.44,4.0
51.27,100.40,47.02,1.0,0.0,0.0,0.005,7.56,3.0
```

The sources generated during a criticality source simulation are a prerequisite to set-up and run a transient simulation in Serpent. Moreover, when continuing the transient from the previous time-step, the sources generated for the previous time-step have to be provided.
In the current work Serpent, 2 version 2.1.31 is used.

## 3.2 Conjugate heat transfer modelling in OpenFOAM

For modelling an MSRE like system with the fuel salt and the solid graphite moderator, a conjugate heat transfer solver *chtMultiRegionFoam* [11] in OpenFOAM [10] is suitable. The Navier-Stokes equations for the mass, momentum, and energy for the liquid domain and the energy conservation equation for the solid domain are discretized and solved with dedicated temperature coupling boundary conditions applied on the solid-liquid interface. The main equations for the fluid domain are;
The continuity equation:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{3.3}$$

where $\rho$ is the fluid density and $\mathbf{u}$ is the fluid velocity field.
The momentum conservation equation is:

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) = -\nabla p + \rho \mathbf{g} + \nabla \cdot \boldsymbol{\tau} \tag{3.4}$$

where $\mathbf{u}$ is the fluid velocity field, $\rho$ is the fluid density, $g$ is the gravitational acceleration, $\boldsymbol{\tau}$ is the stress tensor.
Finally, the energy conservation equation written in terms of enthalpy, is as follows:

$$\frac{\partial \rho h}{\partial t} + \nabla \cdot (\rho \mathbf{u} h) + \frac{\partial \rho K}{\partial t} + \nabla \cdot (\rho \mathbf{u} K) - \frac{\partial p}{\partial t} = -\nabla \cdot \boldsymbol{q} + \nabla \cdot (\boldsymbol{\tau} \mathbf{u}) + \rho \mathbf{g}\ \cdot \mathbf{u} + \rho HS \tag{3.5}$$

where $h$ is the fluid enthalpy, $K$ is the kinetic energy, $\mathbf{u}$ is the velocity, $\rho$ is the density. $HS$ is the heat source term.
For the solid body only the energy conservation equation is discretized and solved:

$$\frac{\partial \rho h}{\partial t} = \nabla (\alpha \nabla h) \tag{3.6}$$

where $h$ is the specific enthalpy and $\alpha = k/C_p$ is the thermal diffusivity defined as the ratio between the thermal conductivity and the specific heat capacity.
The coupling between the solid and fluid bodies is defined using two additional conditions. At the fluid to solid interface the temperatures are assumed equal:

$$T_f = T_w \tag{3.7}$$

The heat flux entering the domain in one region is equal to the heat flux leaving the domain in the other region:

$$k_f \frac{\partial T_f}{\partial \mathbf{n}} = -k_s \frac{\partial T_w}{\partial \mathbf{n}} \tag{3.8}$$

where $k_f$ and $k_s$ are the fluid and solid thermal conductivity respectively, $T_f$ is the fluid temperature next to the wall, $T_w$ is the solid wall temperature, and $\mathbf{n}$ is the directional normal to the wall.
The coupled system of momentum-pressure equations and the energy equation are solved using the *PIMPLE* algorithm [12]. It is a combined version of the *SIMPLE* [13] and *PISO* [14] algorithms. One of the benefits of the *PIMPLE* algorithm is that larger time steps can be used because Courant ($Co$) numbers larger than 1 are acceptable:

$$Co = \Delta t \sum_{1}^{n} \frac{u_{xi}}{x_i} \leq Co_{max} \tag{3.9}$$

for a generic n-dimensional problem, the Courant-Friedrichs-Lewy condition is given by the Equation 3.9; a $Co$ number larger than 1 means that the information from a given mesh cell during the given time-step can propagate further than to the immediate neighbor cell. This can understandably, result in instability and divergence of the solution. On the other hand, when the $Co$ number is limited to one, this usually implies very small time steps and a drastic increase in computational time. However, the *PIMPLE* algorithm, as mentioned above, combines the steady-state *SIMPLE* and the transient *PISO* algorithms to solve this restriction on $Co$ number by searching for a steady-state solution via *SIMPLE* algorithm withing each time-step of the transient *PISO*, before moving on to the next time-step. This means that only the converged steady-state solutions are propagated from one time-step to another, ensuring the stability of the transient solution.
The general structure of the *PIMPLE* algorithm is as follows:
within a single time step

- Solve for momentum
- Compute the mass fluxes
- Solve for pressure
- Correct the mass fluxes
- Re-calculate the velocity field
- Re-calculate the pressure based on the corrected velocity

The correction for pressure and velocity fields is repeated according to the user-specified number of *outer correctors.*

## 3.3 The coupling approach

In Figure 3.1 the coupling scheme is presented as implemented for the steady-state simulations. The routine starts with calling Serpent, which then performs a standard criticality source simulation with uniform starting temperature and density to obtain volumetric power distribution in the system for a given power level. The power distribution is then passed to the OpenFOAM which then solves for the mass, momentum, and energy using the volumetric heat source written by Serpent. The non-uniform temperature and density distributions are used to update the neutronics cross-sections and the k-eigenvalue calculation is repeated. As the nuclear reaction cross-sections depend on the temperature and density via the Doppler and density feedback mechanisms, it is important to update the temperature and density distributions to obtain the correct neutronics picture of the system. The iterations between the thermal-hydraulics and neutronics modules are repeated until temperature and volumetric heat source convergence.

The steady-state coupling scheme is thus quite closely resembling the traditional operator splitting approach, as e.g. the ANSYS CFX and MCNP coupling discussed in chapter 2. However, the dedicated multiphysics interface implemented between Serpent and OpenFOAM makes the coupling more tailored and the iteration routine automatized via a Python script, ensures that both modules are converged before the results are accepted. The user can either set a fixed number of iterations and check whether the temperature and the heat source reach a steady-state distribution and do not change any longer, or convergence criteria could be defined and monitored for the temperature and the heat source residuals.

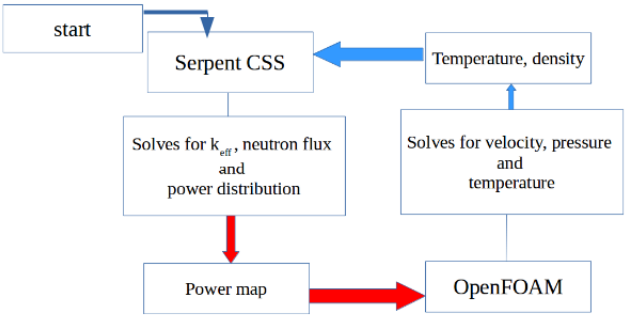

start
Serpent CSS
Temperature, density
Solves for $k_{eff}$, neutron flux and power distribution
Solves for velocity, pressure and temperature
Power map
OpenFOAM

Figure 3.1: Steady-state coupling scheme. CSS stands for Criticality Source Simulation.

For time-dependent simulations, the coupling scheme is shown in Figure 3.2 and the procedure is as follows:

- Using the same mesh grid as for the CFD, perform a Serpent Criticality Source Simulation (CSS) to obtain the power map in the fuel as well as the prompt neutron and delayed neutron precursor sources (see Section 3.1 for a detailed description of source files). Note, that using the same mesh grid here is necessary because we are aiming to utilize the *volpower* file written directly by Serpent and already in a compatible format for OpenFOAM, and since the *volpower* writes the heat deposition per cell, we need identical mesh for both the neutronics and the CFD modules.
- Solve for mass, momentum, and energy in the CFD module using the power map from the Serpent CSS. This will provide us with the velocity field necessary for the DNP transport and with temperature and density fields to update the neutronics input.
- Using the prompt neutron and delayed neutron precursor sources CSS, run a transient mode simulation (TMS) in Serpent, on a coarse mesh grid. The transient mode simulations in Serpent also called dynamics mode simulations, allows the assessment of the time-dependent evolution of the system, e.g. the power evolution with time after a reactivity insertion in the system. This is achieved by having a time binning definition in the problem setup, on top of the and similar to the energy binning used for k-eigenvalue calculations. The time binning is defined as follows:

```
tme Name 2 NB Tmin Tmax
```

where Name is a user-chosen name for the time discretization, 2 indicates uniform

binning, NB is the number of bins, Tmin and Tmax are the lower and upper time limits respectively, e.g. $0a$ to $1s$.
Since the transient simulations usually take more time than a CSS in Serpent, to save some computational time we do not use the same fine CFD grid for the TMS. As in this coupling approach, point-wise DNP tracking is used as opposed to mesh-based tracking, it is not needed to have identical meshes for the neutronics and the CFD modules.

- Using the velocity field obtained from the CFD module and the delayed neutron precursor (DNP) distribution source from the TMS; perform precursor transport and write a new DNP source. There are several ways to perform the transport: the most exact one would be to loop over the DNP birth coordinates, match those with the same coordinates on the CFD grid and read the corresponding velocity value. Then, knowing the neutronics time-step and the velocity, one can calculate the corresponding coordinate shift for the birth positions of the DNPs:

$$\mathbf{A}(x,y,z) = \mathbf{A}(x_0,y_0,z_0) + \Delta\mathbf{A}(x,y,z) = \mathbf{A}(x_0,y_0,z_0) + \mathbf{u} \times \Delta t_n$$

  where $\mathbf{A}(x,y,z)$ is the new generic DNP position vector, $\mathbf{A}(x_0,y_0,z_0)$ is the original position vector from the Serpent DNP source corresponding to static fuel, $\Delta\mathbf{A}(x,y,z)$ is the changes in the position considering that the fuel is not static and is simply the product of the velocity ($\mathbf{u}$) and the neutronics time-step $\Delta t_n$.
  However, considering that there is a substantial number of coordinates to loop through, for some cases, when the velocity field is uniform, or when only the axial transport of the DNPs is significant, the above-described routine can be somewhat simplified, e.g. by dividing the problem geometry into several regions and assuming uniform average velocity within each of the regions. Then one only needs to check to which region the certain DNP coordinates belong, and shift them by the velocity field of that region. In this work, the axial transport of the DNPs is of importance, and the uniformity of the axial velocity profile allows to use the average domain velocity to shift the DNPs rather than adopting the more time-consuming approach of looping through the DNP birth coordinates and finding the point on the CFD mesh grid that matches the given DNP birth position to extract the corresponding velocity value.
- Per one CFD time-step ($\Delta t_{CFD} = 1s$), there are 10 corresponding Serpent TMS calculations ($\Delta t_n = 0.1s$), as the neutronics response time scale is smaller than that of the CFD. At the end of the neutronics iterations, the newly established power level is used to derive the power adjustment factor to set the normalization power level for the next time-step step.
- Proceed to the next time step: use the power adjustment factor to set the new normalization power level for the Serpent CSS, use the temperature and density profiles from CFD as well to update the CSS input, and perform a CSS calculation to obtain the new volumetric power for CFD and the new neutron and DNP source files for the TMS.

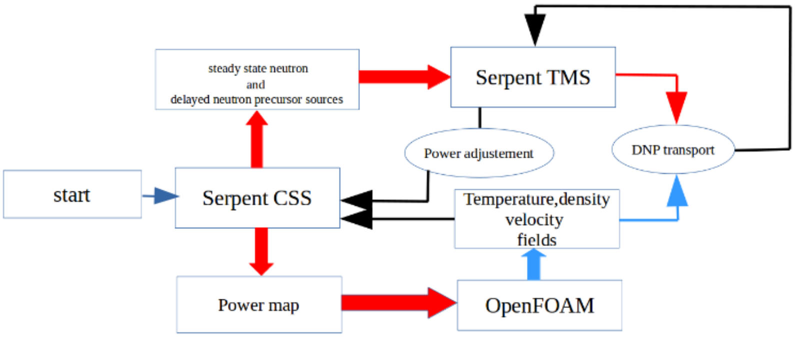


Figure 3.2: Transient coupling scheme. TMS stands for Transient Mode Simulation.

Some Python code snippets showing the routines to shift DNPs and to control the iterations between the CFD and neutronics modules as well as the script for decoding DNPs binary source file written by Serpent can be found in Appendix B.

## 3.4 Application of the Coupling Methodology

In the following sections of the chapter, the application of the above-presented coupling methodology to the steady-state and transient analysis of the Molten Salt Reactor Experiment [15](Figure 3.3) is presented. MSRE is of particular interest as the only molten salt reactor that has been in operation and the only source of experimental data for the molten salt reactors. In this work, the analysis is confined to a simplified model of only one fuel channel with an adjacent moderator region. The main goal is to assess the ability of the above-described coupling model to reproduce acceptable results when compared to the experimental data both qualitatively and quantitatively.

### 3.4.1 MSRE system description

The Molten Salt Reactor Experiment operated at the Oak Ridge National Laboratory from 1964 to 1967 and aimed at showcasing several attractive factors of molten salts as fuel and coolant, i.e.

- The liquid fuel offers enhanced burnup capabilities, not limited by radiation damage to the structural materials and reactivity loss, thanks to the online reprocessing capabilities.
- High-temperature process heat production and thermal efficiencies comparable to fossil-fuel plants.
- No need for high-pressure vessel and piping.
- Good neutron economy due to the low parasitic neutron absorption cross-section.
- The fluoride salts used in the MSRE do not react rapidly with the water or air.

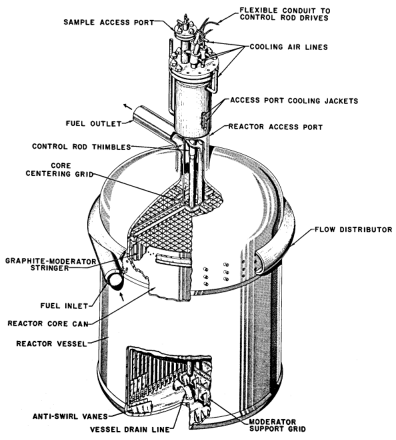


Figure 3.3: MSRE vessel outline [19]

Initially, the plant was designed to be operated at 10 $MW$ thermal power for the experimental findings being reasonably scalable for a full-scale industrial plant. However, due to the technical limitations of the heat exchanger, this was reduced to 7 $MW$. The general outline of the reactor is presented in Figure 3.3.
The reactor vessel is 1.524 $m$ in diameter and 2.438 $m$ in height and contains a graphite core structure; 1.370 $m$ diameter and 1.626 $m$ in height. The fuel enters the core at the top of the vessel, it is evenly distributed in the flow distributor section and flows downwards to be collected in the lower plenum. Here the salt passes through flow straightening vanes and flows upwards through the graphite moderating channels. The fuel salt exiting the core is collected in the fuel pump and pumped into a shell and tube type heat exchanger where the heat is removed by the secondary coolant salt (Figure 3.4 ). This salt dissipates the heat in air-cooled radiators. Helium is used as a cover gas as well as to remove volatile fission gaseous products from the fuel salt in the fuel pump. There are four drain tanks to keep the fuel salt when it is not in use as well as several mechanisms to prevent the salt from freezing at any point during the operation, e.g. there are heaters in the radiator and special shut-off doors to close the airflow in case the reactor power drops.
To control the reactor power during power operation airflow adjustment in the radiator is used, whereas at low power operation the control rods are used to keep the flux constant and the airflow is used to regulate the temperature.

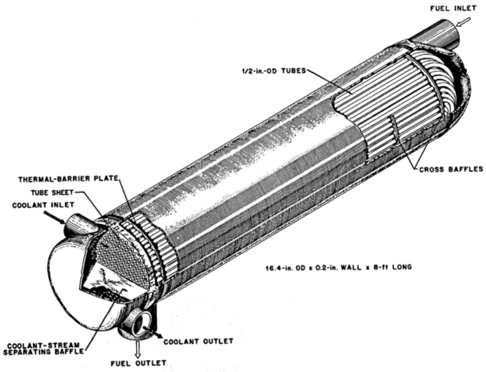


Figure 3.4: MSRE Primary Heat Exchanger [19]

The reactor core is composed of graphite blocks called stringers with half-channels being machined into the four faces of the stringers to form a passage for the fuel salt. A typical stringer drawing is shown in Figure 3.5.

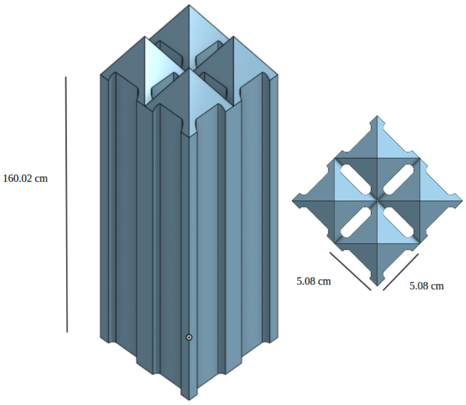


Figure 3.5: The MSRE graphite stringer

It can be seen that the half-channels machined into the stringer faces form a passage for the fuel which has roughly an ellipsoidal shape. There are 1160 fuel channels in the MSRE core, counting the annulus between the core can and the graphite block and the annuli around the control rod thimbles.
There are three control rods in the MSRE design, located in the center of the core, as depicted in Figure 3.6. The main purpose of these rods is to provide excess reactivity to e.g.compensate for xenon poisoning or loss of delayed neutrons due to the fuel flow. The principal factor in the reactor shutdown mechanism is the negative temperature feedback coefficient of the fuel salt, together with the three control rods being dropped into the core by gravity when the reactor power level reaches 15 $MW$. However, the shutdown mechanism is not strongly dependent on the fast-acting rods. As an absorber material, gadolinium oxide is used in the rods. There is also graphite sampling equipment in the center of the core to investigate the graphite moderator behavior under irradiation and its interaction with the fuel salt.

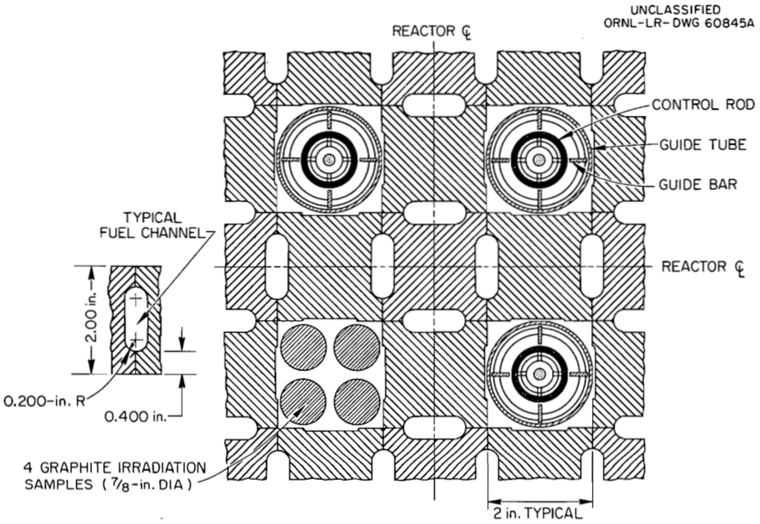


Figure 3.6: MSRE control rod and graphite sample arrangement [19]

The main operational parameters of MSRE are summarized in Table 3.1.

| Parameter | Value | Unit |
|---|---|---|
| Design thermal power | 10.00 | $MW$ |
| Fuel inlet T | 908.15 | $K$ |
| Fuel inlet P | 1.38 | $bar$ |
| Fuel outlet T | 935.93 | $K$ |
| Fuel outlet P | 0.48 | $bar$ |
| Average power density | 14.00 | $kW/l$ |
| Total flow rate | 272.55 | $m^3/h$ |

Table 3.1: MSRE Fuel Operational Parameters [19]

Several fuel options were considered for the MSRE; highly enriched fuel with $^{235}$U, partially enriched fuel with $^{235}$U, fuel with $^{232}$Th-$^{235}$U and finally fuel with $^{232}$Th-$^{233}$U which was used for dynamic analysis. The last fuel was tested to investigate the scenario of the reactor being operated with $^{235}$U bred from $^{232}$Th. The fuel salts composition is reported in Table 3.2.

| | Salt 1 | Salt 2 | Salt 3 | Salt 4 |
|---|---|---|---|---|
| LiF | 70.000 | 66.800 | 65.000 | 58.900 |
| $BeF_2$ | 23.600 | 29.000 | 29.100 | 36.500 |
| $ZrF_4$ | 5.000 | 4.000 | 5.000 | 4.500 |
| $UF_4$ | 0.400 | 0.200 | 0.900 | 0.115 |
| $ThF_4$ | 1.000 | 0.000 | 0.000 | 0.000 |
| $^{235}$U enrichment | 93.140 | 93.140 | 34.400 | 0.000 |
| $^{233}$U enrichment | 0.000 | 0.000 | 0.000 | 91.490 |

Table 3.2: Fuel salts composition in $mol\%$ and the fissile nuclide fraction (enrichment) in $atom\%$

The main properties of the graphite are summarized in Table 3.3. The results of the analysis of the MSRE graphite were considered to be satisfactory for a research reactor; it was compatible with the fuel salt with an acceptable percentage of noble metal fission products being plated on the graphite and acceptable dimension changes due to the irradiation [16].

| Property | Value | Unit |
|---|---|---|
| Density | 1.83-1.89 | $g/cm^3$ |
| Total porosity | 17.70 | % |
| Thermal conductivity (without grain) | 658.68 | $W/m^2/K$ |
| Specific heat at 922.00 K | 1758.46 | $J/kg/K$ |

Table 3.3: MSRE graphite properties [16]

### 3.4.2 MSRE modelling and validation of fuel circulation-induced effects

As it was already mentioned, the fuel circulation in Molten Salt Reactors induces some specific effects related to the reactor kinetic behavior, mainly due to the transport of delayed neutron precursors with the fuel flow. In this section, a theoretical model developed at ORNL and accounting for the reactor period dependence on the fuel circulation time is briefly described. This theoretical model was validated during the zero-power kinetic experiments and these experiments are also described very briefly [17].
It is convenient to analyze the fuel circulation related effects at very low reactor power because at negligible power levels these effects reduce to essentially two contributors:

1. Reactivity loss due to the delayed neutron precursors decaying in the external loops of the reactor and not in the core.
2. Effects due to the shift of the delayed neutron precursors in the direction of fuel circulation.

The ORNL theoretical model related the static reactivity normally calculated by reactor analysis tools available at the time, to the reactor asymptotic period. Thus, the effect of the fuel circulation on the reactor period can be evaluated. The time-dependent reactor equations used in the ORNL model for the neutron flux and for the delayed neutron precursors, accounting for the fuel transport in the axial direction, are :

$$\frac{1}{v}\frac{\partial\Phi}{\partial t} = L\Phi + (1-\beta_T)f_P P\Phi + \sum_{i=1}^{6}\lambda_i f_{di} C_i \tag{3.10}$$

$$\frac{\partial C_i}{\partial t} = \beta_i P\Phi - \lambda_i C_i - \frac{\partial(\mathbf{u_z} c_i)}{\partial z} \tag{3.11}$$

here $\Phi$ and $c_i$ are the neutron flux and the delayed neutron precursor concentration, u_z is the fuel velocity in axial direction, $\lambda_i$ is the decay constant, $\beta_T$ is the total delayed neutron fraction (the sum of individual $\beta_i$ group fractions), $L$ is the total neutron loss operator (leakage, absorption and inelastic scattering) and $P$ is the production by fission operator. The prompt and delayed neutron energy spectra are denoted as $f_P$ and $f_{di}$ respectively. As it can be seen from Eq. 3.11, the delayed neutron precursor concentration depends on the transport of the precursors in the axial direction with the fuel flow.
If one assumes that the operators $L$ and $P$ are time-independent, for a fixed rod position, then we can look for a condition under which the flux and precursor concentrations vary in time in form of $e^{wt}$, where $w^{-1}$ will be the reactor stable period and the static reactivity would then be the largest eigenvalue of the flux equation. Through some meticulous mathematical derivations described in details in [17], the general in-hour equation which relates the stable period to the static reactivity is obtained:

$$\rho_s = \omega[\Lambda + \sum_{i=1}^{6}\frac{(\beta_i' - \gamma_i')}{\omega+\lambda_i}] + \sum_{i=1}^{6}\gamma_i' \tag{3.12}$$

where $\rho_s$ is the static reactivity, $\omega$ is the reactor period and $\Lambda$ is the prompt neutron generation time. Notice, that this equation resembles the in-hour equation for static fuel, but differs from it by $\beta_i$ effective production fractions for the delayed neutron precursors which are now reduced by a $\gamma_i$ term which depends on the reactor period and the rate of the fuel circulation (fuel velocity).
In the ORNL model, to calculate the static reactivity $\rho_s$, the reactor geometry is subdivided into several sub-regions: reactor core, upper and lower plenums, and external piping (Figure 3.7).

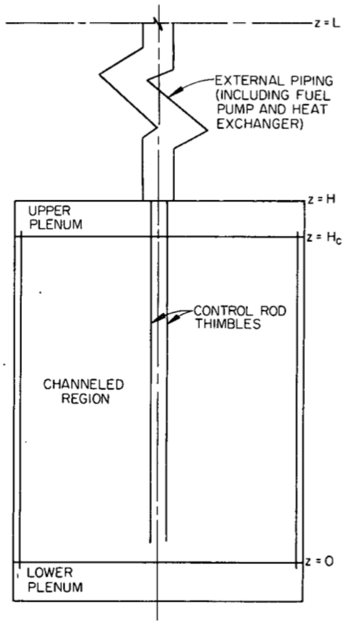


Figure 3.7: The MSRE core model used in ORNL neutronics calculations [17]

The general procedure to compute the static reactivity used in this approach can be described as follows:

- Calculation of static and adjoint neutron fluxes using standard reactor physics analysis tools.
- From the reactor inverse period and the static flux distribution corresponding to the same state, calculate the static precursor concentration in the fuel, $c_i^0(z)$.
- Once the concentration is known, calculate the precursor axial distribution:

$$c_i(z) = \alpha_k c_i^0(z) \tag{3.13}$$

- Finally, calculate $\rho_s$

The developed calculation model gives the relation between the reactor period and the reactivity and explicitly takes into account the effect of delayed neutron precursor movement. In Figure 3.8the the delayed neutron emission axial density in the fuel salt is shown calculated according the above described model, for stationary fuel case (no fuel movement), for reactor stable period of 10 s (the period is constant in time), and for 0 s period, i.e. critical reactor with circulating fuel, where the period is defined as:

$$\tau = \frac{l_d}{k-1} \tag{3.14}$$

where $l_d$ is the mean generation time with delayed neutrons and $k$ is the multiplication factor.

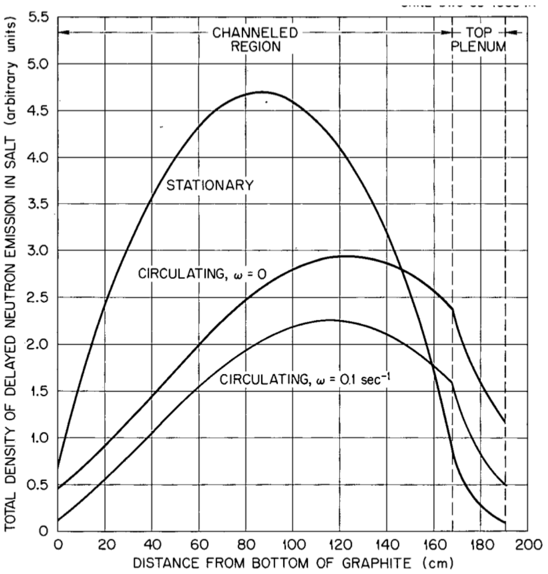


Figure 3.8: Axial distribution of delayed neutron precursor concentration for different reactor periods (calculation results) [17]

Alongside the theoretical model, several zero power experiments were carried out on the MSRE to investigate the effects of fuel circulation on reactivity. These experiments attempted to measure the control rod's worth at different conditions. Two baseline experiments were performed; establishment of the critical $^{235}$U concentration for stationary fuel with all control rods withdrawn, and the extra amount of $^{235}$U needed to reach criticality with circulating fuel and all rods withdrawn, as the circulating fuel caused reactivity loss.

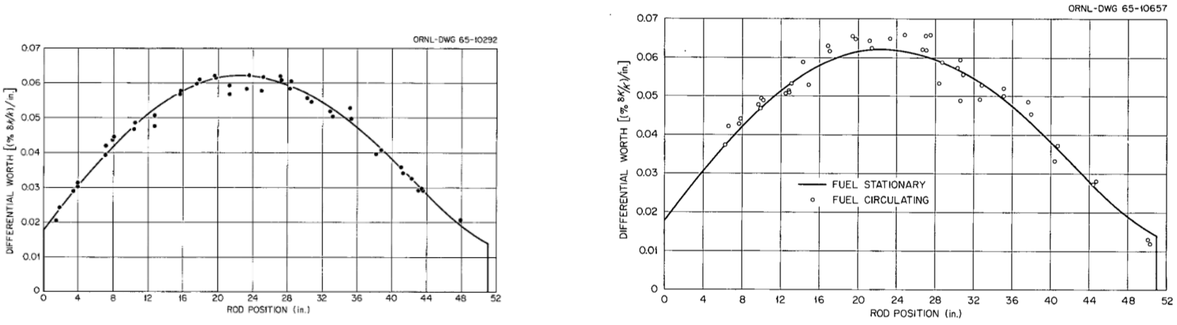


Figure 3.9: Control rod worth for stationary (left) and circulating fuel(right), model and experiment [17]. The dots correspond to measurement points, the continuous line is the fitted curve for stationary fuel.

After the baseline experiments, rod worth measurements were done with the fuel stationary and then circulating and the reactor period was measured for each critical rod position. This was done by first adjusting the rod to make the reactor critical then withdrawing the rod to a fixed position and keeping it there until the reactor power increased by a certain prescribed amount and then the rod was lowered to bring the power back to the initial level. The differential worth of one of the rods measured for stationary and circulating conditions and compared to the numerical calculation results is presented in Figure 3.9 [17]. Fairly good agreement between the model and the measurements can be noted, with the deviations stemming from the experimental uncertainties as well as several assumptions made in the calculation model.

The general conclusion regarding the effects of fuel motion on reactivity is that there is a net loss of reactivity due to the motion of delayed neutron precursors and distortion of the emission towards the upper core and plenum. In the Results sections of this chapter, a comparison between the results obtained with the coupling method and the ones discussed here is presented. Note, that similar to the method developed at ORNL to encounter for fuel motion by considering DNP transport in the axial direction of the fuel motion, the method used in this work to shift DNPs and described in the sub-section 3.3, similarly facilitates the DNP transport, considering transport in the axial direction, with the flow.

## 3.5 Model setup

This section describes the geometry models, materials, boundary conditions, and other model setup parameters for the MSRE simulations carried out in this chapter.

As Monte-Carlo simulations are already computationally heavy, and in this work, the objective is to establish the feasibility of coupled CFD Monte-Carlo modelling protocols on a test case of the MSRE, in which case an entire-core CFD mesh would be in the order of several million cells, it was deemed suitable to start with a single-channel analysis. This is to limit the computational burden but also to make it easier for the debugging of coupling routines. The main limitation of using a single-channel model is that information on the radial distribution of parameters such as the neutron flux or the fuel temperature can not be obtained.

Before starting the coupled analysis, a purely neutronics calculation using Serpent was run on the entire core model to obtain the power distribution in the core as well as the axial and radial fluxes. No thermal-hydraulics coupling effects are considered at this stage. The full-core model is presented in Figure 3.10 [18].

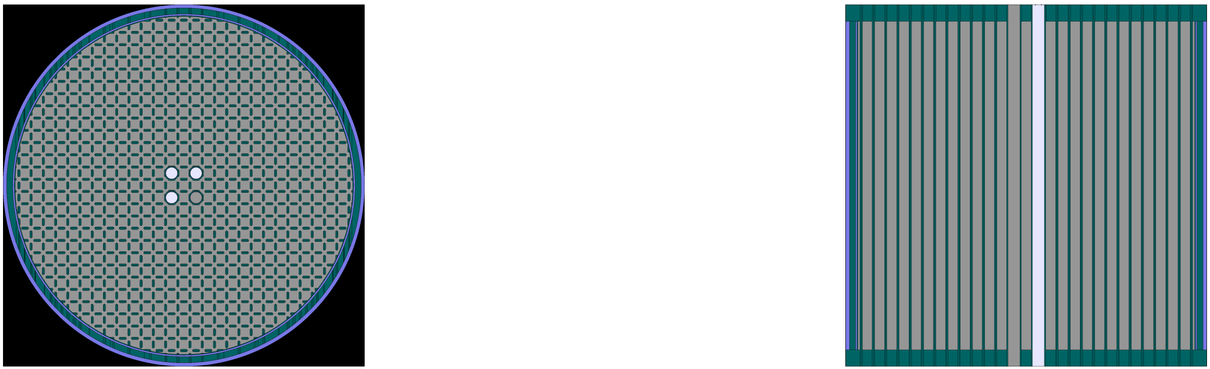


Figure 3.10: MSRE full core model in Serpent: top view (left) and side view (right)

The lattice temperature for the calculations with the full-core model is considered uniform $922K$, the fuel is fresh (no fission product poisons) and the control rods, as well as the experimental sample position, are estimated as 70% gadolinium oxide - 30% Al oxide and pure graphite rods respectively.

As for the single-channel model, two-channel shapes are considered; a simplified cylindrical fuel channel is considered in one approach, as opposed to the actual nearly-ellipsoidal shape of the fuel passage formed by half-channels machined into the graphite faces in the MSRE. The fuel to moderator ratio is kept constant for both models. The main purpose is to find out to which extent the shape simplification affects the results of the analysis. Note, that the ORNL calculations model considered cylindrical channel shape as well. The two models are depicted in Figure 3.11. The main parameters for the geometry are summarized in Table 3.4.

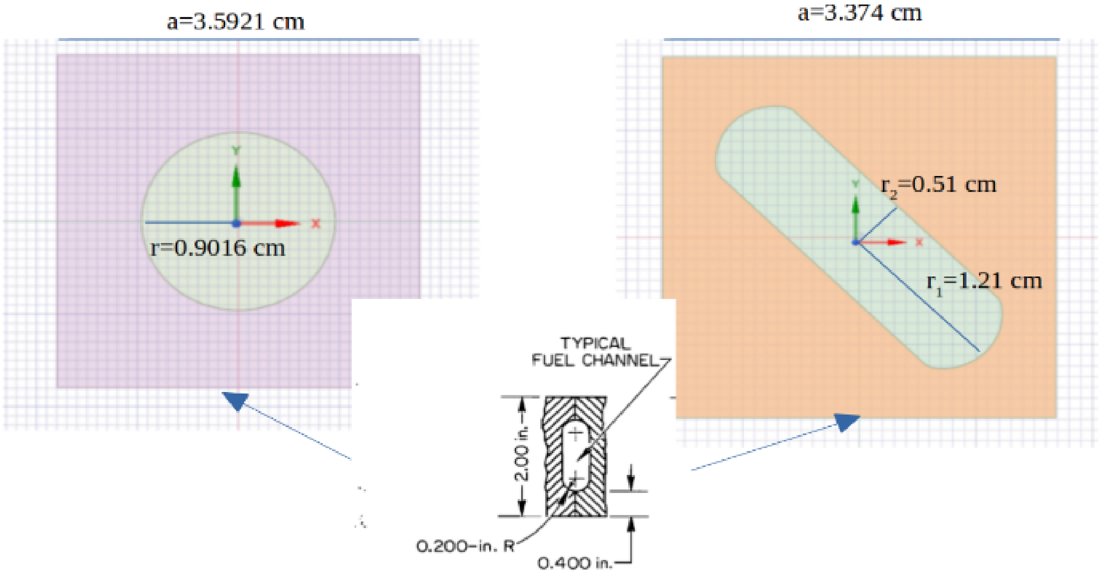


Figure 3.11: Single-channel models: cylindrical and stadium-shaped channels. The drawing in the lower-center is the original ORNL drawing for the channel shape and dimensions.

| Channel shape | Radius [mm] | Height[mm] | Graphite lattice side length [mm] | Fuel vol. fraction |
|---|---|---|---|---|
| Cylindrical | 9.016 | 1600 | 36.560 | 0.244 |
| Stadium-shaped | 24.200 ×10.200 | 1600 | 33.740 | 0.244 |

Table 3.4: Main geometrical parameter values for both channel types

A structured hexagonal mesh is used to model the two geometries, presented in Figure 3.12. The mesh grid independence study shows that for the cylindrical model $1.8 \times 10^5$ mesh elements are sufficient to ensure independence, meanwhile for the stadium-shaped channel $2.3 \times 10^5$ mesh cells are required. The mesh orthogonal quality is reported in Table 3.5. As a reminder, the mesh orthogonal quality is an important characteristic showing how close the mesh elements are to the optimal size and shape on a scale of 1 being an ideal match and 0 being the worst possible quality. Usually, for models where structured meshing is possible, an orthogonal quality above 0.8 is achievable.

| Model | Number of elements | Orthogonal quality |
|---|---|---|
| Cylindrical | $5.0 \times 10^3$ | 0.87 |
| | $4.0 \times 10^4$ | 0.90 |
| | $3.0 \times 10^4$ | 0.92 |
| | $9.0 \times 10^4$ | 0.94 |
| | $1.8 \times 10^5$ | 0.98 |
| Elliptical | $5.0 \times 10^3$ | 0.87 |
| | $1.0 \times 10^4$ | 0.90 |
| | $5.0 \times 10^4$ | 0.92 |
| | $1.1 \times 10^5$ | 0.94 |
| | $2.3 \times 10^5$ | 0.98 |

Table 3.5: Mesh element number and orthogonal quality for considered model

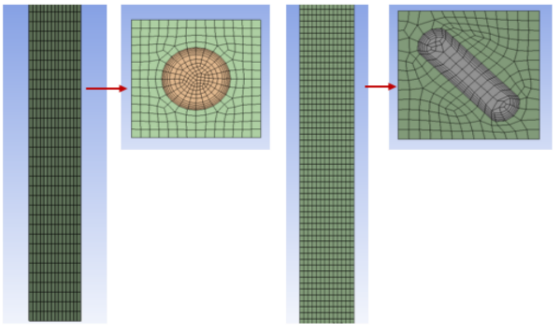


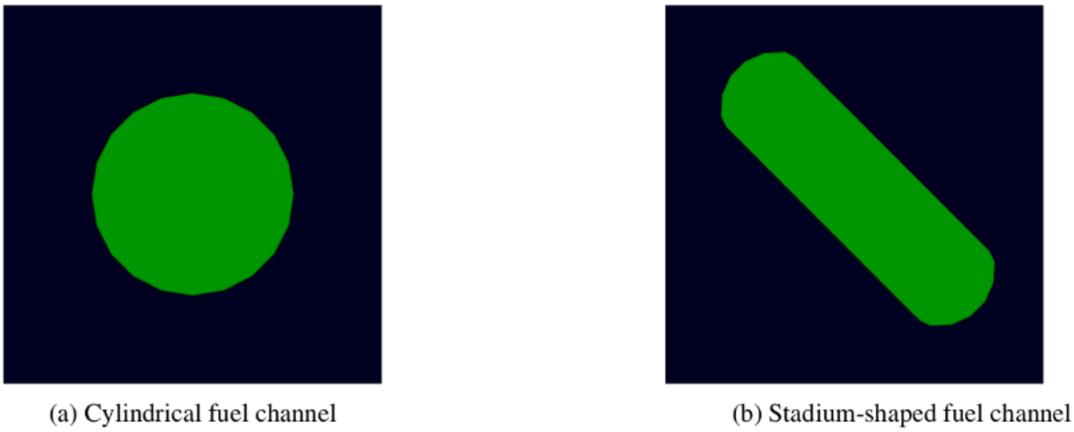
Figure 3.12: Mesh used for the thermal-hydraulics simulations

Thanks to the multiphysics interface between Serpent and OpenFOAM, the same geometry used for thermal-hydraulics studies can be transferred to Serpent for the neutronics calculations (Figure 3.13).

| Material | $\rho[kg/s]$ | $\mu[Pas]$ | $k[Wm^{-1}K^{-1}]$ | $C_p[Jkg^{-1}K^{-1}]$ |
|---|---|---|---|---|
| Fuel salt [at 922 K] | 2080 | 0.007 | 5.530 | 1970 |
| Graphite [at 922 K] | 1860 | - | 58.800 | 1760 |

Table 3.6: Fuel and graphite main properties [20]

The boundary conditions for both neutronics and thermal-hydraulics modules are summarized in Table 3.7; radially infinite geometry is considered through the reflective boundary condition applied on the graphite stringer periphery, whereas axially the geometry is finite and vacuum boundary condition is applied on both the fuel and the graphite upper and lower boundaries. For steady-state and transient analysis different fuel compositions are used, reported in Table 3.9 for the steady-state and in Table 3.8 for the transient. This is done to be able to carry out comparisons with the available ORNL data in a consistent manner. The fissile nuclide is $^{235}$U at 93% enrichment for the steady-state part and $^{233}$U at 91.49% enrichment for the transient simulation to be consistent with the experiments. The main thermal-hydraulics properties for the fuel are presented in Table 3.6.



(a) Cylindrical fuel channel

(b) Stadium-shaped fuel channel

Figure 3.13: Top view: geometry used for neutronics calculations in Serpent

| Quantity | BC/IC | Value | Unit |
|---|---|---|---|
| Temperature | Neutronics IC, fuel | 923.5 | $K$ |
| Neutrons | Neutronics BC, graphite and fuel axial dir | vacuum | - |
| Neutrons | Neutronics BC, graphite radial dir | reflective | - |
| Temperature | CFD BC, fuel inlet | 923.5 | $K$ |
| Velocity | CFD BC, fuel inlet | 0.183 | $m/s$ |

Table 3.7: Initial and boundary conditions (IC and BC) for neutronics and CFD modules

The inlet velocity for the fuel channel is chosen based on the assumption made in the ORNL thermal-hydraulics model of MSRE, wherein the entire core is divided into 5 regions with characteristic velocities (Table 3.10). The particular channel modelled here is assumed to be positioned in the section of the core with the most full-sized channels in it (region 2) and thus the velocity of the fuel is 0.183 $m/s$.

| Component | mol % |
|---|---|
| LiF | 70.00 |
| $BeF_2$ | 24.85 |
| $ZrF_4$ | 5.00 |
| $UF_4$ | 0.15 |

Table 3.8: Fuel vector composition for the steady-state calculations [19]

| Component | mol % |
|---|---|
| LiF | 58.900 |
| $BeF_2$ | 36.500 |
| $ZrF_4$ | 4.500 |
| $UF_4$ | 0.115 |

Table 3.9: Fuel vector composition for the transient calculations [19].

| Region | Number of full sized channels | Fuel volume fraction | Average fuel velocity [m/s] |
|---|---|---|---|
| 1 | 12 | 0.256 | 0.600 |
| 2 | 940 | 0.224 | 0.183 |
| 3 | 108 | 0.224 | 0.454 |
| 4 | 78 | 0.142 | 0.249 |
| 5 | 0 | 1.000 | 0.079 |

Table 3.10: Core regions used in the ORNL thermal hydraulics model. Region 5 is the annulus between the graphite and the core shell [19].

## 3.6 Results and Discussion

In this section first the axial and radial flux distributions for the entire MSRE core as well as the power distribution in the core are shown, obtained by performing a purely neutronics simulation in Serpent. The goal is to compare the obtained values to the ORNL available data to check the correct set-up of the neutronics model.
After that results on steady-state and transient single-channel analysis are shown and compared to the ORNL results. For the steady-state, fuel and graphite axial temperatures are compared to model calculations performed at ORNL. For the transient part, the power evolution after a step reactivity insertion is compared to ORNL experiments.

### 3.6.1 Full-core flux and power distribution

The full-core flux and power density distributions are presented in this sub-section, for the reactor power level of $10MW$. In Figure 3.14 the radial and axial flux distributions in the MSRE core are shown and compared to the flux distribution calculated by ORNL using *Equipoise* multi-group diffusion code [21]. The overall agreement in the shape and the numerical values between the ORNL calculations and the Serpent results are satisfactory and the relation of the slow flux to the fast flux is captured correctly. Moreover, a flux depression caused by the control-rod timble around 5 cm is visible for the Serpent model as well as it is for the ORNL calculations. It can be noticed, that the discrepancies when comparing the axial fluxes are lower than for the radial fluxes: the average discrepancy between the axial fast fluxes is 3%, whereas for the axial thermal fluxes it is 7%. The discrepancies are larger for the radial fluxes: the fast radial fluxes differ by an average of 5%, and the radial thermal fluxes differ by an average of 15%. Note, that all reported average differences are calculated excluding the boundary values; there is no information available in the [19] report on the boundary conditions used for the ORNL multi-group diffusion neutronics module calculations and the flux values at the boundaries differ with respect to the Monte-Carlo exact solution results (for the radial flux the difference is the largest, ≈ 45%). The overall differences between the Monte-Carlo simulation results and the Equipoise calculation results are attributed to the neutronics model implementation in Equipoise:

- The code is limited to 2D problem geometries.
- The entire MSRE geometry is represented as 19 different cylindrical regions with different proportions of basic materials (fuel, moderator, and INOR).
- The materials within each separate region are treated as a homogeneous mixture, which means that in the regions where non-homogeneity (more than one material) is present, only the overall flux shape can be calculated.
- Apart from these geometry and material-related simplifications, the code solves for two-group diffusion, which lacks precision when compared to the continuous-energy Monte-Carlo methods.

The statistical deviation for Serpent calculations is within 2% and a $2\sigma$ confidence interval is included in Figure 3.14 for the Serpent results.

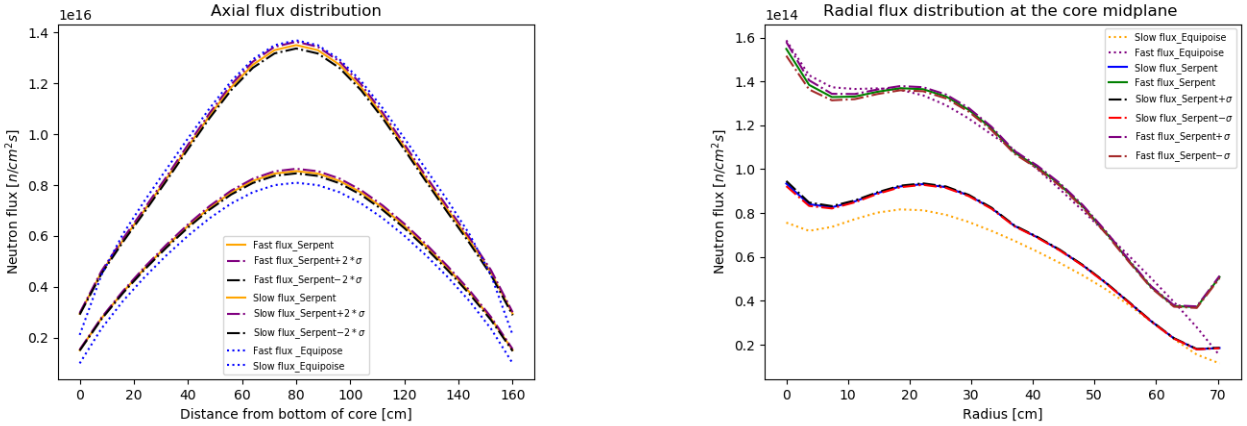


Figure 3.14: Radial and axial flux distributions. The radial flux is sampled at the core mid-plane, the axial flux is sampled at $18cm$ from the core center line.

In Figure 3.15 the normalized fission density distribution in the fuel is provided. Again, the agreement between the Serpent and *Equipoise* is good.

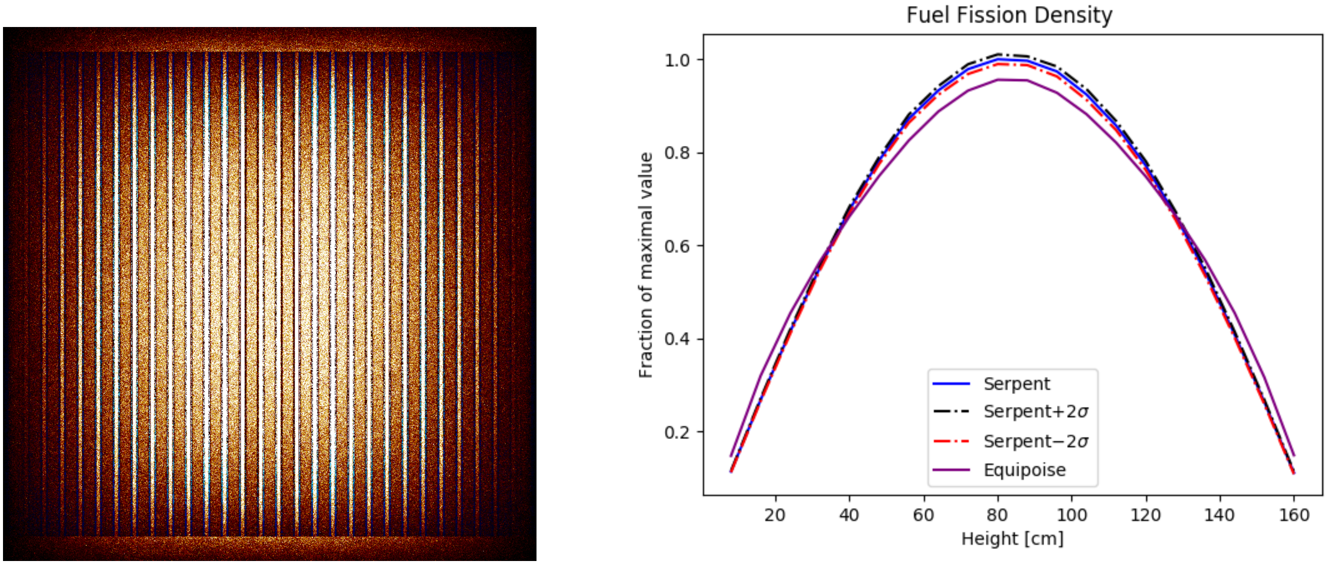


Figure 3.15: Left: fission rate in the fuel: brighter color means higher rate of fission. Right: Normalized axial fission density distribution

### 3.6.2 MSRE steady-state analysis

In this section, the main results for the reactor single-channel analysis under steady-state operating conditions are reported and compared to the results available in the MSRE design and operation report [19]. Figure 3.16 shows the heat deposition in the fuel for the cylindrical and the elliptical channel types, as calculated by Serpent. The axial power distribution profile follows closely the earlier shown axial flux distribution in the reactor core. The channel shape does not affect the power spatial distribution and both the cylindrical as well as stadium-shaped channels predict almost identical power distribution in the channel.
This volumetric power distribution is used as a heat source for the CFD module as it is discussed below in relation to the temperature distribution in the channel.

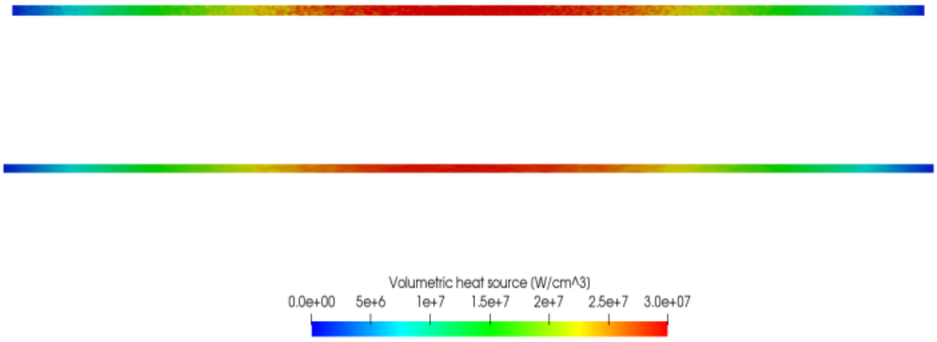


Figure 3.16: Fuel heat deposition for the stadium-shaped channel (top) and the cylindrical channel (bottom)

The fuel radial and axial velocity profiles are shown in Figure 3.17 and Figure 3.18. It is clear from the radial distribution, that the fuel has a larger velocity in the central part of the channel and the velocity decreases next to the wall, due to the laminar nature of the flow. This has an important effect on the temperature distribution in the fuel and

the adjacent graphite stringer, as shown later.
When comparing the stadium-shaped channel and the cylindrical channel, the shape affects slightly the axial velocity, resulting in $0.005 m/s$ higher axial velocity than for the cylindrical channel. As the stadium-shaped channel forms a more narrow and elongated passage for the flow, a slightly higher flow velocity in the stadium-shaped channel is normal.

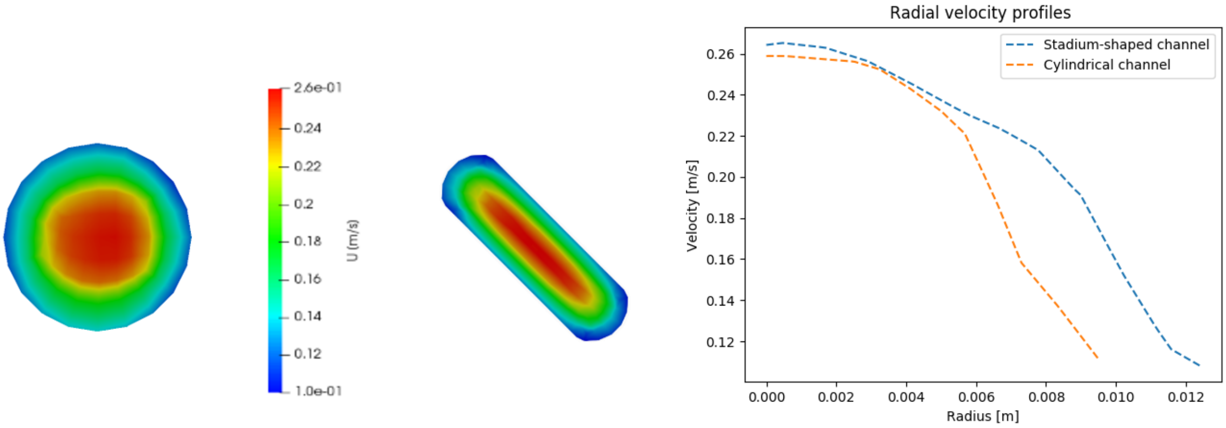


Figure 3.17: Fuel velocity radial profiles. In case of the elliptical (stadium-shaped) channel the major axis is used.

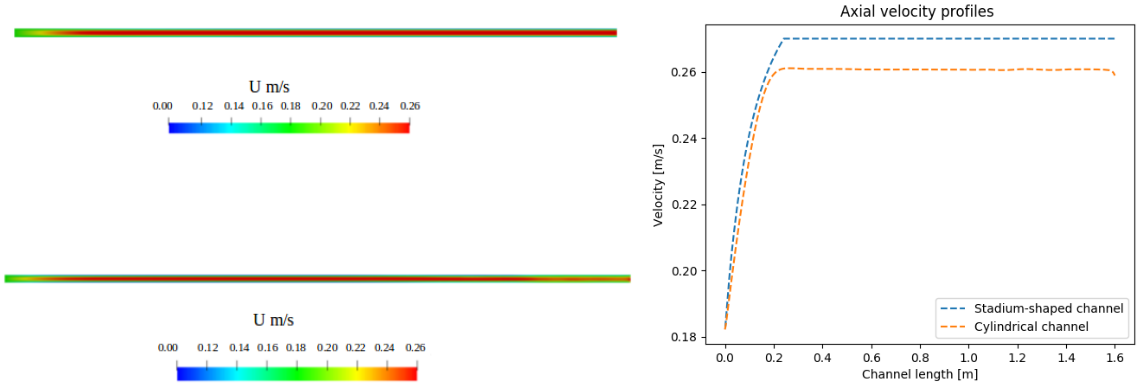


Figure 3.18: Fuel velocity axial profiles: top-left for the elliptical channel (stadium-shaped), bottom-left for the cylindrical channel

The $Re$ number calculation justifies that the flow is laminar and there is no need to consider a turbulence model, as the $Re$ number for both channel models is in the range of 1200 to 1800.
Finally, Figure 3.19 compares the axial temperature in the fuel and the graphite. Both the simulation and the ORNL model calculations agree, that in the graphite stringer the temperature is higher than in the adjacent fuel channel during steady-state operation:

$$T_g = T_f + \Delta T \tag{3.15}$$

The temperature difference $\Delta T$ is attributed to:

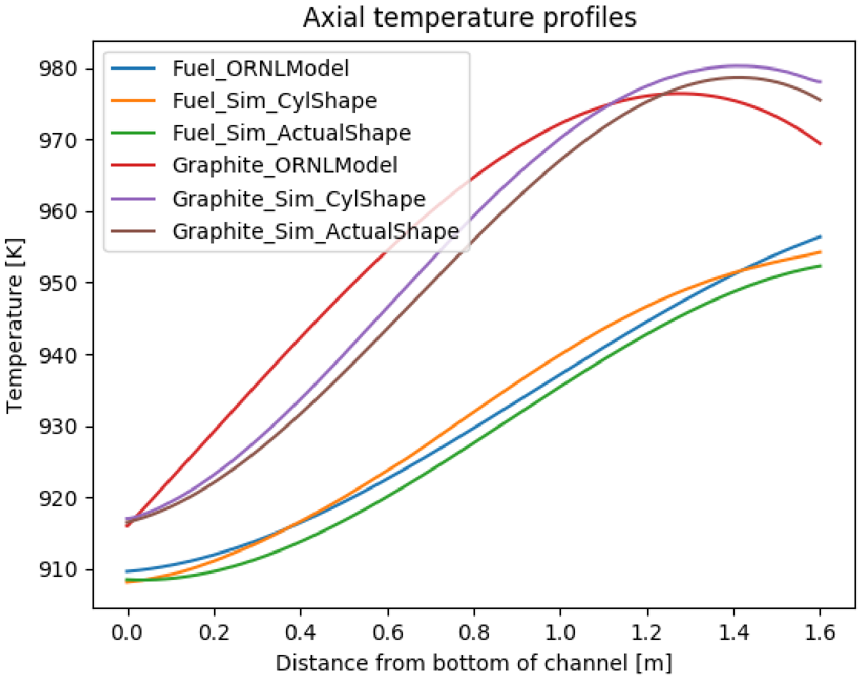


Figure 3.19: Temperature distribution in the fuel channel and the adjacent graphite stringer. The simulation is compared to the MSRE model calculations [19]

a. The Poppendiek effect: the laminar flow results in the near-wall fuel velocity being lower than the velocity in the channel central part. Combined with the internal heat generation in fuel, this causes the fuel temperature adjacent to the wall to be higher than the average across the channel. The ORNL model considers coefficients for cylindrical channel order to evaluate the Poppendiek effect [22]. This assumption might slightly overestimate the Poppendiek effect, as shown in Figure 3.19.

b. The heat flow from graphite causes temperature drop across the graphite-fuel interface. The difference between this temperature and the mean graphite temperature is necessary to conduct heat from the graphite.

When comparing the agreement between the simulations and the ORNL model calculations, it can be noticed, that:

- For the fuel:
  - The ORNL model calculations assume an inlet temperature of $909.61K$, which is $1.46K$ higher than the inlet temperature used in the simulations. This is related to the fact, that in the ORNL model calculations the reactor is considered to be consisting of the main core and the peripheral regions and the fuel is assumed to heat up after entering the reactor at $908.15K$ and reaching the main core. The simulations, however, consider the average reactor inlet temperature of $908.15K$ as a channel inlet temperature in the simulations.
  - The fuel temperature calculated by the ORNL model has a relative difference of $0.75K$ on average and a maximum relative difference of $2.57K$ compared to the simulation results with the actual channel shape, a $2.25K$ average relative difference, and a $4.26K$ maximum relative difference compared to the simulation results considering the cylindrical fuel channel shape.

    The differences between the model calculations and the CFD simulations can be attributed to the fact that the ORNL model is a strictly analytical approach to the calculation of the temperatures in the fuel channel and the adjacent graphite stringer, it only accounts for the fuel radial velocity impact and the temperature distribution shape is derived from the shape of the fuel power density by further considering that the radial and the axial power-density distributions are separable. The following analytical expression is then used in the model for calculating the fuel temperatures:

$$T_f(r,z) = T_f(z=0) + \xi \frac{A_r}{u(r)} cos\alpha - cos[\frac{\pi}{77.7}(z+4.36)] \tag{3.16}$$

where $\xi$ is a constant defined as $\frac{(Q_f)_m}{(\rho C_p)_f}$ ($(Q_f)_m$ is the specific power which includes the heat added from the moderator and $(\rho C_p)_f$ is the specific heat) and $\alpha$ is defined as $\frac{\pi}{77.7}(0+4.36)$. $A_r$ is the radial variation in the power density distribution and $u(r)$ is the radial velocity field.

The simulations, on the contrary, represent a full 3D model of the channel and the graphite stringer. The velocity field (both radial and axial components) are taken into account when solving for temperature and the volumetric heat source is directly coming from a neutronics calculation rather than using an analytical expression for the power density distribution.

Additionally, the ORNL model shows the individual fuel channel and the adjacent graphite stringer axial temperatures for the hottest radial position in the core, whereas the simulations consider one arbitrary fuel channel.

Considering the above mentioned, the discrepancies between the analytical ORNL model and the CFD simulations are acceptable.

The differences between the two CFD simulations with the actual channel shape and the cylindrical channel shape are attributed to the shape of the heat transfer area and the flow distribution.

- For the moderator:
  - The ORNL model calculations have an average relative difference of $2.52K$ compared to the CFD simulations with the cylindrical channel shape and a difference of $4.72K$ in the case of the actual channel shape. In both cases, the CFD simulations are underestimating the temperature in the graphite stringer compared to the model calculations. The ORNL model calculates the temperature in the graphite by evaluating the temperature difference between the graphite and the fuel and adding it to the fuel temperature (Eq. 3.22). As explained above, this difference is attributed to the Poppendiek effect and the temperature across the channel-stringer wall. The Poppendiek effect is evaluated in the ORNL model by assuming a cylindrical channel shape which is expected to overestimate the real temperature distribution. The temperature at the channel-stringer wall as well cannot be calculated analytically for the actual channel shape, thus an approximate method is adopted, wherein two graphite stringer geometries are considered; one represents the stringer as a cylinder with the same cross-sectional area as the actual graphite stringer in the MSRE has and the other one as a slab cooled on both sides. The first approximation is underestimating the temperature and is considered as a lower limit value, whereas the second one overestimates the temperature and is considered as an upper limit value. Subsequently, the temperature at the channel-stringer wall is assigned a value in between these two. It is thus not surprising that the model calculations with approximations that tend to

overestimate the temperature distributions predict higher temperatures in the graphite stringer compared to the CFD simulations where no approximations are used.

– The temperature rise in the graphite stringer is not only due to the heat conduction from the hot fuel but also due to the production of heat within the graphite moderator itself due to the neutron and gamma heating. The ORNL calculations assumed 6% of the total reactor power to be generated in the graphite due to neutron and gamma heating and this is the value used to assign uniform heat generation in the graphite. There are no details provided in the report [19] as to how exactly the calculations resulted in the reported 6 % value. The Serpent simulations presented in this work account for neutron and prompt gamma heating, as Serpent does not account for delayed gammas. The Serpent simulations result in 4.3% of the power being produced in graphite and this is the value used as a heat source for the graphite.

The fuel mean temperature, the graphite bulk mean temperature, as well as the average temperature difference between the fuel and the graphite, are reported in Table 3.11. Note, that no graphite permeation by fuel is considered for the presented results neither in the simulations nor in the ORNL model calculations.

| Model | Fuel T $K$ | Graphite T $K$ | $\Delta$ T $K$ |
|---|---|---|---|
| ORNL | 930.7 | 957 | 26.3 |
| Cylindrical channel | 931.5 | 955 | 23.5 |
| Stadium-shaped | 929 | 950 | 21 |

Table 3.11: Fuel and graphite mean temperatures and the temperature difference for steady-state operation

## 3.7 Reactivity insertion transient

In this section results of a step reactivity insertion transient for $^{233}$U, based fuel are presented and compared to the MSRE data [23] and MSRE theoretical predictions based on the model described in [24]. During the initial approach to power operation with $^{233}$U fuel, measurements of the flux response to step reactivity perturbation were taken for three power levels: 1, 5, and 8 $MW$. OpenFOAM-Serpent simulation is performed for 8 MW power level only.

The reactor initial power level is 8 $MW$ and the reactivity insertion is 0.0248 $\%\delta k/k$. The delayed neutron precursor movement is explicitly taken into account via coupling between the flow field and the neutronics module as described in Section 3.3. Simple re-entry of the delayed neutron precursors into the channel is considered based on the characteristic circulation time; the DNPs that end up outside the core are re-injected back with a delay corresponding to the out-of-channel time based on the fuel re-circulation. As shown in Figure 3.20 the power increases rapidly during the first second after the reactivity insertion. The simulation results predict a slightly higher peak at around 8.74 $MW$, whereas the MSRE data shows a peak power of 8.6 $MW$. The power increase is accompanied by the fuel temperature increase. The power starts decreasing until about 10 $s$ due to the negative temperature feedback coefficient. Between 10 and 17$s$ it can be noticed that the power level is reaching a plateau since that the temperature of the fuel

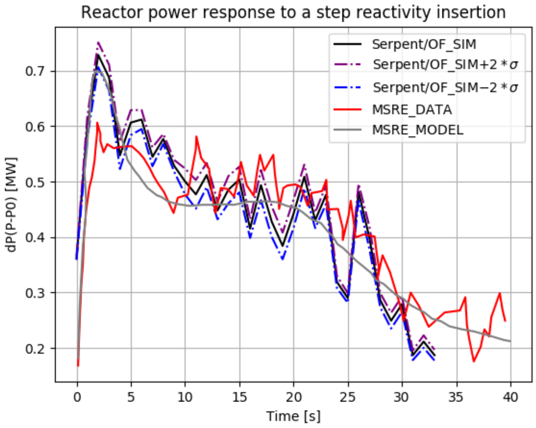


Figure 3.20: Power response to the reactivity insertion

in the channel does not change in this interval and the feedback mechanism has balanced the effects of the step reactivity insertion. When the hot fuel which exited the channel after the initial power increase re-enters the channel together with the DNPs, the power goes down again, and at around 40 $s$ after the reactivity insertion reaches 8.2 $MW$. The experimental limitations on data processing related to the noise level [23] allow to make an overall reactor behavior comparison only; however, this comparison shows that the main temperature feedback effects are captured. It should be mentioned, however, that the out-of-channel treatment of the DNPs in the coupling approach is very simple and should be enhanced in the future to take into account the decay of the DNPs and the realistic out-of-core time which in this case is not realizable as only one single channel is modelled.

## 3.8 Conclusion

This chapter presented a coupling approach between Serpent Monte-Carlo code and OpenFOAM CFD library. The coupling features an exchange of the relevant coupled terms between the neutron-kinetics and the thermal-hydraulics. The convergence of individual physics is ensured via iterating the single-physics solvers within a time-step. The effects of the fuel motion are taken into account by introducing the fuel velocity as an operator that acts on the coordinate-based delayed neutron precursor positions. The information on delayed neutron precursors is available from the Serpent calculations.
The coupling approach is tested for the MSRE single fuel channel modelling with an adjacent graphite stringer. Both steady-state and transient scenarios are modelled. The steady-state temperature profiles in the fuel and the graphite are in good agreement with the ORNL model calculations. The transient scenario of reactivity insertion is compared to both model calculations and the experimental values. It can be appreciated that despite a somewhat simplified treatment of the delayed neutron precursor transport, the simulation results capture the overall transient behavior and the reactor thermal feedback mechanism. The advantages of this coupling technique are:

- The most accurate modelling approach for the neutronics module is used.
- No extensive modification of the software used is necessary.

The main drawbacks of the approach are:

- The CFD model requires mesh which is in the order of magnitude of $10^5$ for a single fuel channel and an adjacent graphite stringer; scaling this up to an entire reactor would require substantial computational resource for the CFD.
- In the transient mode the simulation time is dictated by the time required for the

Monte-Carlo simulations. Considering that high statistics have to be ensured and that the neutronics time-step is an order of magnitude smaller than the thermal-hydraulics time-step.

- The coupling follows the mechanism of the traditional operator splitting approach. Thus, the order of accuracy is expected to be lower than for the more integrated coupled solvers which solve for different physics simultaneously.

The clock-times for the CFD and the neutronics modules for both the steady-state and the transient simulation regimes are reported in Table 3.12. Since the neutronics time-step is 10 times smaller than that used for the CFD module, it is easy to see that e.g. in the time-dependent simulation mode for the cylindrical channel, it will take ≈ 10 hours to simulate 1 $s$ in Serpent as compared to only ≈ 0.003 hours to simulate 1 $s$ in OpenFOAM. This shows that the main bottle-neck for coupled Monte-Carlo CFD simulations is the computational time taken up specifically by the Monte-Carlo neutronics solvers.

| Channel shape | Mesh cell number | Simulation regime | N of processors | Module | Clock-time ($min$) |
|---|---|---|---|---|---|
| Cylindrical | $1.8 \times 10^5$ | Steady-state | 1 | TH | 11.33 |
| Elliptical | $2.3 \times 10^5$ | Steady-state | 1 | TH | 22.30 |
| Cylindrical | $1.8 \times 10^5$ | Steady-state | 30 | NK | 51.00 |
| Elliptical | $2.3 \times 10^5$ | Steady-state | 20 | NK | 252.00 |
| Elliptical | $2.3 \times 10^5$ | Time-dependent | 1 | TH | 0.466 |
| Elliptical | 16 | Time-dependent | 20 | NK | 115.00 |

Table 3.12: Simulation times reported for the cylindrical and elliptical channel shapes, for the CFD module (TH) and the neutronics module (NK). N processors indicates the number of processors used. Note, that for the steady-state regime the total simulation time is reported. For the time-dependent regime, the time it takes to complete one time-step is reported.

# Bibliography

[1] J. Leppänen et al., "The Serpent Monte Carlo code: Status, development, and applications in 2013.", Ann. Nucl. Energy 82 (2015) 142-150.

[2] T. Kaltiaisenaho, "Statistical Tests and the Underestimation of Variance in Serpent 2", VTT, 2014.

[3] A. Laureau et al., "Transient coupled calculations of the Molten Salt Fast Reactor using the Transient Fission Matrix approach", Nuclear Engineering and Design 316 (2017) 112-124.

[4] R. Tuominen, V. Valtavirta, J. Peltola, and J. Leppänen, "Coupling Serpent and OpenFOAM for neutronics - CFD multi-physics calculations", Proceedings of PHYSOR 2016.

[5] F. B. Brown, "K-effective and Other Concerns for Monte Carlo Eigenvalue Calculations," Progress in Nuclear Science and Technology 2, 738–742, 2001.

[6] V. Valtavirta, J. Leppänen, T. Viitanen, "Coupled neutronics–fuel behavior calculations in steady-state using the serpent 2 Monte Carlo code", Annals of Nuclear Energy 100, Pages 50–64, 2017.

[7] J. Leppänen et al," Unstructured Mesh-Based Multi-Physics Interface for CFD Code Coupling in the Serpent 2 Monte Carlo Code", Proceedings of PHYSOR, 2014.

[8] M. Aufiero, C. Fiorina, A. Laureau, P. Rubiolo, V. Valtavirta, "Serpent-OpenFOAM coupling in transient mode: simulation of a Godiva prompt critical burst." Proceedings of M&C Conference, 2017.

[9] M. Daeubler, A. Ivanov, B. L. Sjenitzer, V. Sanchez, R. Stieglitz, R. Macian-Juan, "High-fidelity coupled monte Carlo neutron transport and thermal-hydraulic simulations using Serpent 2/SUBCHANFLOW". Annals of Nuclear Energy 83, Pages 352–375.

[10] OpenFOAM API Guide v2006,
*https://www.openfoam.com/documentation/guides/latest/doc/index.html*

[11] I. Gallego-Marcos, "Thermal Mixing CHT Simulations with OpenFOAM:URANS and LES.", Master's Thesis, KTH, 2013.

[12] T. Holzmann, "Mathematics, Numerics, Derivations and OpenFOAM", 2017.

[13] S. V. Patankar, "A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows", Int. J. of Heat and Mass Transfer 15, Issue 10, Pages 1787-1806, 1972.

[14] R. Issa, "Solution of the implicitly discretized fluid flow equations by operator-splitting", Journal of Computational Physics 62, Issue 1,Pages 40-65, 1968.

[15] P.N. Haubenreich, J. R. Engel, "Experience with the Molten-Salt Reactor Experiment", Nuclear Applications and Technology 8, 1970.

[16] W. R. Grimes, "Molten Salt Reactor Chemistry", Oak Ridge National Laboratory, Oak Ridge, Tennessee 37830, Nuclear Applications and Technology 8, 1970.

[17] B. E. Prince, "Period Measurements on the molten salt reactor experiment during fuel circulation: theory and experiment", ORNL Internal Report, 1966.

[18] https://github.com/gridley/msr-neutronics/tree/master/MSRE/detailedMSRE

[19] J.R. Engel, P.N. Haubenreich, "Temperatures in the MSRE core during steady-state operation", ORNL Internal Report, 1962.

[20] S. Bell, S.Haubenreich, R. Lindauer, J. Tallackson
"MSRE Design and Operation Report, Part 5: Reactor Safety Analysis Report", ORNL, Internal Report, 1964.

[21] M.L. Tobias, T. B. Fowler, "Equipoise - An IBM-704 code for the solution of two-group, two-dimensional, neutron diffusion equations in cylindrical geometry", ORNL, Internal Report, 1960.

[22] H.F.Poppendiek, L.D.Palmer "Forced Convection Heat Transfer between Parallel Plates and in Annui with Volume Heat Source Within the Fluids", ORNL, Internal Report, 1954.

[23] R.C. Steffy, "Experimental Dynamic Analysis of the MSRE with U-233 fuel ",ORNL, Internal Report, 1970.

[24] R.C. Steffy, and P. J . Wood, "Theoretical Dynamic Analysis of the MSRE with U-233 Fuel", ORNL, Internal Report, 1968.

[25] M. Aufiero, P. Rubiolo, M. Fratoni, "Monte Carlo/CFD coupling for accurate modelling of the delayed neutron precursors and compressibility effects in molten salt reactors", Recent Advancements in liquid and solid molten salt reactors, Conference Proceedings, 2017.

# Chapter 4

# The Multiphysics Benchmark for Molten Salt Fast Reactor and the Verification of Serpent-OpenFOAM Coupling Approach

As it has already been discussed in this work, in recent years several multiphysics coupling methods and coupled solvers dedicated to the analysis of the MSRs have been developed. Most of these solvers feature in-house neutronics solvers adopting various approximations, e.g. neutron diffusion or spherical harmonics methods, and they are coupled to commercial or customized, in-house developed CFD solvers. Most of these coupling approaches have been developed and tested in application to the Molten Salt Fast Reactor (MSFR), as it is the official design concept accepted by the Generation IV forum [1]. As there is not much experimental and operational data available on the MSRs, and those available are limited to the graphite-moderated MSRs, there was a need to develop a baseline benchmark to be used to validate the various coupling approaches developed for the MSFR. Recently results on a multiphysics benchmark for codes developed for the Molten Salt Fast Reactor were published [2]. In this benchmark, several participants compare their results for the steady-state as well as transient operational regimes of the MSFR, aiming to spot and quantify any possible sources of discrepancies stemming from the coupling approaches. DTU was not an initial participant in the benchmark, however, considering the value of the benchmark for the verification of coupling techniques developed for the MSRs, it was decided to test the coupled Serpent-OpenFOAM approach based on this benchmark. In this chapter selected observable from the benchmark are compared to the results obtained with the coupled Serpent-OpenFOAM approach. The method used to perform the benchmark is largely identical to the one described in chapter 3 of the thesis, with the exception of the handling of the delayed neutron precursor transport modeling. Instead of having the flow velocity as an external parameter acting on the delayed neutron precursor field, a dedicated passive scalar transport solver is developed in OpenFOAM capable to handle explicit mesh-based transport of the DNPs.

On top of comparing Serpent-OpenFOAM results to those obtained by the benchmark initial participants, a comparison to an in-house multiphysics solver SEALION developed by Seaborg Technologies is performed as well [12].

The results of this verification benchmark are presented in the paper "Verification and benchmarking of multiphysics coupling techniques for modelling of molten salt reactors",

submitted for publishing to the Annals of Nuclear Energy journal (see Part II, Paper 4). The raw data used to generate the plots shown in this chapter can be downloaded here: https://github.com/AshNlb/MSFR_Numericalbenchmark_Data.git
Additionally, Appendix C presents the OpenFOAM source code files used to implement the DNP transport solver.

## 4.1 Brief description of the codes used by the participants

A very brief description of the codes used by four initial benchmark participants for thermal-hydraulics as well as neutronics modules is presented below. On top of that, a short description is given also for the SEALION code [12] and the DTU approach is explained in details.

- TU Delft
  - Phantom $S_N$ Multi-group Boltzmann equation solver coupled to the delayed neutron precursor transport equation is used for the neutron transport.
  - DGFlows parallel solver for incompressible Navier-Stokes equations is used for the fluid dynamics part.

  Both solvers are based on a Discontinuous Galerkin Finite Element Method (DG-FEM) for space discretization whereas for time discretization implicit Backward Differentiation Formulae (BDF)are used [4].
- CNRS
  - Multi-group diffusion or SP1 and SP3 spherical harmonics approximations, implemented using $C++$ libraries in OpenFOAM is used for the neutron transport.
  - OpenFOAM solver for a single-phase incompressible flow with Boussinesq approximation and PIMPLE algorithm is used for the fluid dynamics. For the detailed description refer to [5].
- PoliMi
  - Multi-group diffusion method is used for the neutron transport modelling, implemented using $C++$ libraries in OpenFOAM.
  - OpenFOAM solver for single-phase incompressible flow with Boussinesq approximation is used for the fluid flow dynamics.
    For detailed code description, see [6], [7], [8], [9], [10].
- PSI
  - Gen-Foam multiphysics solver with possibilities to use multi-group neutron diffusion or spherical harmonics approximation, implemented using $C++$ libraries in OpenFOAM is used for the neutron transport modelling.
  - For the fluid flow modelling an OpenFOAM based solver for incompressible flow is used.
    For detailed code description, see [11].

This concludes the list of the four initial participants involved in the development and performance of the benchmark. The DTU and the SEALION approaches are described below.

- SEALION (Seaborg Technologies)
  - Modified point-kinetics equations with pre-computed temperature feedback for the neutron transport part is used.

  - For the fluid flow modelling, a modified incompressible, single-phase Navier-Stokes CFD solver with Boussinesq approximation in OpenFOAM.
    For details, see [12].

The DTU approach used in this work is implemented as follows:

- DTU
  - Neutron transport is realized in Serpent Monte-Carlo code.
  - For the thermal-hydraulics part an OpenFOAM solver for incompressible flow with PIMPLE algorithm and Boussinesq approximation.
  - The delayed neutron precursor transport is modelled with an additional scalar transport solver implemented in OpenFOAM with the group constants being imported from Serpent.

The coupling scheme is presented in Figure 4.1.

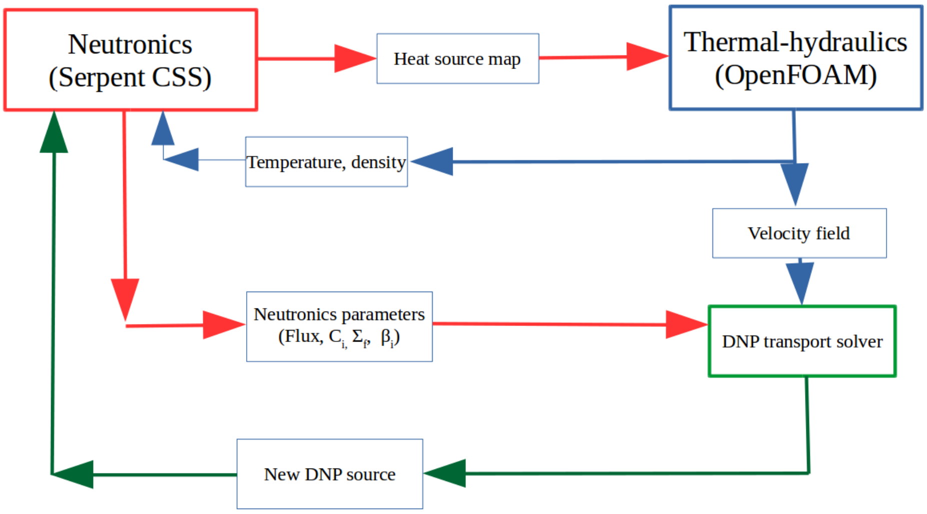


Figure 4.1: The Coupling Scheme

The coupling is implemented as follows:The cycle is started by a Serpent criticality source simulation. The volumetric heat source, as well as the DNP fractions and decay constants, the average number of neutrons emitted per fission, and the energy released per fission are obtained.

Additionally, the initial distribution of the DNP source corresponding to a static fuel is extracted. In Chapter 3 it is described, that Serpent allows to track DNPs either on a coordinate basis or a mesh basis. While in Chapter 3 the coordinate-based tracking is used, it is convenient for the DNP transport solver developed in this chapter to use the mesh-based tracking. The mesh specified in Serpent for DNP tracking has the same cell number as the one used in the thermal-hydraulics module. Thus, we obtain a concentration of DNPs of each group in each mesh cell. An example of how a typical DNP source written on a mesh basis is shown below:

```
– 1 2 3269 438393000000000000 0
  1 2 3270 413797000000000000 0
  1 2 3271 384222000000000000 0
  1 2 3272 349589000000000000 0
  1 2 3273 310259000000000000 0
  1 2 3274 266984000000000000 0
```

```
1 2 3275 220752000000000000 0
1 2 3276 172601000000000000 0
1 2 3277 123439000000000000 0
1 2 3278 73888400000000000 0
1 2 3279 24463000000000000 0
1 2 3280 23594200000000000 0
1 2 3281 71077100000000000 0
1 2 3282 118229000000000000 0
1 2 3283 164336000000000000 0
1 2 3284 208638000000000000 0
1 2 3285 250231000000000000 0
1 2 3286 288262000000000000 0
1 2 3287 322103000000000000 0
```

The firs column indicates the end of the simulation (1), the second column is the DNP group (in this case 2), the third column is the mesh cell number (in this case cells 3269-3287 are shown), the fourth column is the number of DNPs of the given group in the given cell and the last column is the absolute error.

- The volumetric heat source is passed on to the OpenFOAM CFD solver which solves for the mass, momentum, and energy.
- The converged velocity field from the CFD solver together with the initial DNP source distribution is used then in the DNP scalar transport solver. The DNP scalar transport solver solves then the following equation for each DNP group:

$$\frac{\partial C_i}{\partial t} = -\nabla(\mathbf{u}c_i) + \nabla \cdot D\nabla C_i - \lambda_i C_i + \beta_i \nu \Sigma_f \Phi \tag{4.1}$$

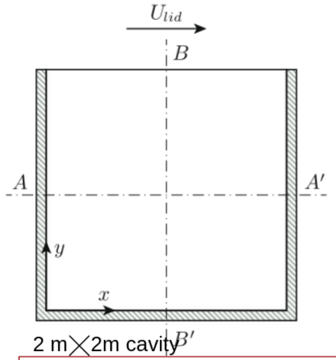
The fuel velocity field is thus taken into account explicitly. All the relevant terms for the DNP production and decay are extracted from Serpent.

- Once the converged DNP distribution is obtained, together with the non-uniform temperature and density fields from the thermal-hydraulics solver it is passed to the neutronics module, and the Serpent calculation is repeated.
- For transient scenarios, it is important to understand on which time-scale the neutronics and the thermal-hydraulics response is occurring and how often the coupled quantities have to be exchanged between the modules to yield valid results. From the neutronics point of view, the non-uniform temperature and density are used as input from the CFD solver together with the DNP fields obtained from the DNP transport solver. The CFD solver needs the volumetric power from neutronics and the neutron transport solver needs the DNP constants and the DNP distributions from the neutronics module. Provided that the selected time-step in the thermal-hydraulics module is small enough, we obtain the temperature and density fields and pass them on to the neutronics module where a criticality eigenvalue simulation is performed to obtain the new power source and the constants needed for the delayed neutron precursor transport solver. This input data is circulated back to the thermal-hydraulics module and the thermal-hydraulics problem is solved again with the updated input. The argument behind running neutronics in a steady-state regime is that when obtaining the instantaneous picture of the system neutronics, the thermal-hydraulics input can be assumed to stay fixed. As

the thermal-hydraulics time-scale is larger than that of the neutronics and the thermal-hydraulics picture of the system changes due to the changes in the power profile coming from the neutronics, this assumption can be made without the risk of missing relevant feedback mechanisms. In the same way, it is assumed that when performing a steady-state neutronics simulation, the thermal-hydraulics input can be assumed to be fixed.
To ensure a convergence of the results, the solvers are iterated in this manner within a time-step.

## 4.2 Benchmark geometry

As a benchmark geometry, the lid-driven cavity problem is used, which is a standard CFD problem used to test new solvers and solution methods due to its simplicity [3]. The problem geometry is a simple 2D square domain filled with liquid, with Dirichlet boundary conditions on all sides; three static sides and one moving side on which velocity is imposed. The geometry and the boundary conditions are shown in Figure 4.2.



2 m×2m cavity

Note: The cavity is 2D: it has 1 m depth in pseudo z axis, but all parameters are dependent only on x and y.

- ❑ The cavity is filled with molten salt at T=900 K
- ❑ Vacuum BC for neutronics to each boundary
- ❑ No slip BC (U=0) for thermal hydraulics everywhere, except the top wall
- ❑ The top wall is moving with Ux velocity
- ❑ All walls are adiabatic (∂T/∂n=0), cooling through a volumetric heat sink
- ❑ Uniform volumetric heat transfer coefficient
- ❑ Neumann BC for the DNPs

Figure 4.2: Benchmark geometry and boundary conditions [2]

The cavity is filled with liquid molten salt: LiF-BeF$_2$UF$_4$. The material properties for the salt are given in Table 4.1. The fuel salt composition is specified in Table 4.2.

| Property | Units | Value |
|---|---|---|
| Density | $kgm^{-3}$ | $2.0 \times 10^3$ |
| Kinematic viscosity | $m^2s^{-1}$ | $2.5 \times 10^{-2}$ |
| Volumetric heat capacity | $Jm^{-3}K^{-1}$ | $6.15\times10^6$ |
| Thermal expansion coefficient | $K^{-1}$ | $2.0\times10^{-4}$ |
| Prandtl number | - | $3.075\times10^5$ |
| Schmidt number | - | $2.0 \times10^8$ |

Table 4.1: Salt properties [2]

| Isotope | $^{6}$Li | $^{7}$Li | $^{9}$Be | $^{19}$F | $^{235}$U |
|---|---|---|---|---|---|
| Atomic fraction (%) | 2.11488 | 26.0836 | 14.0992 | 56.3969 | 1.30545 |

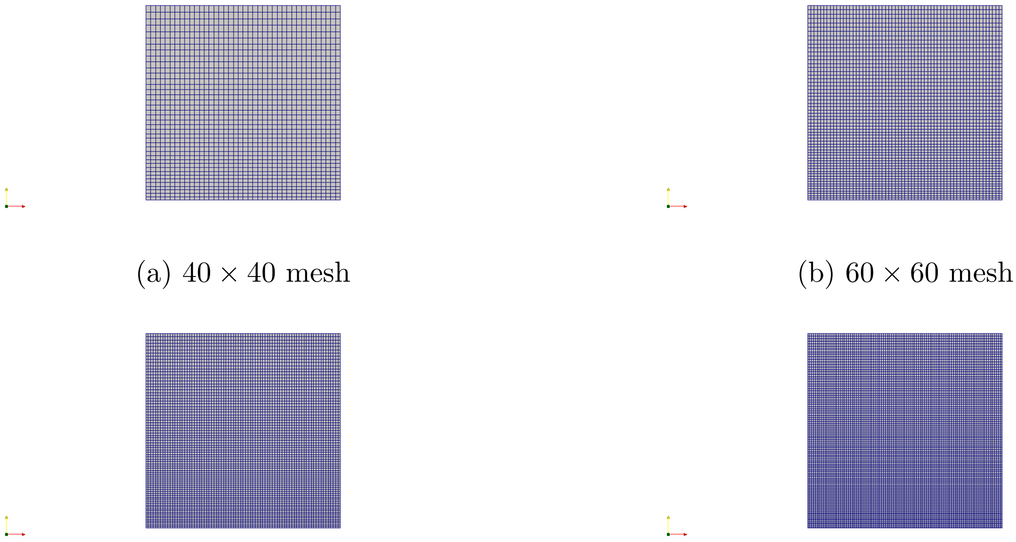
Table 4.2: Salt composition [2]

The benchmark is carried out in three phases:

1. Phase 1 - Participants present results on single-physics testing, without any coupling.
2. Phase 2 - The coupling between thermal-hydraulics and neutronics is gradually introduced and the effect of it is accounted for under steady-state operational conditions.
3. Phase 3 - The transient behavior of the system in presence of the coupling is investigated.

The main benchmark results will be compared with the published results in this work. For a detailed explanation of how the other participants performed the work, the readers are referred to [2].

In OpenFOAM, the standard *blockMesh* utility is used to generate the model mesh. This utility generates structured hexahedral mesh by decomposing the geometry domain into a set of hexahedral blocks. For this work, meshes of different coarseness are considered, with a uniform expansion in *x, y, z-* directions. The meshes are presented in Figure 4.3. The model is pseudo-2D: $z$ direction has a depth of 1 m, but only one mesh cell is specified along $z$ and all the boundary conditions in this direction (front and back faces) are specified as *empty*, which forces the solver to not solve conservation equations for these boundaries.



(a) 40 × 40 mesh

(b) 60 × 60 mesh

(c) 80 × 80 mesh

(d) 100 × 100 mesh

Figure 4.3: Grids used for the mesh independence sTU Delfty

A mesh independence study performed on the grids shown in Figure 4.3 resulted in the 60 × 60 mesh being selected as a baseline for the simulations. As a criterion for checking the mesh independence, the average velocity in the domain is chosen and the results are reported in Figure 4.4.

All thermal-hydraulics and neutronics simulations presented in this work are performed on the 60 × 60 mesh grid. Using the same grid for both thermal-hydraulics and neutronics allows for direct communication between the two modules, e.g. the volumetric heat source and the delayed neutron precursor distributions can be generated and used without any further modifications related to the grid translations. The OpenFOAM generated mesh is linked to Serpent used for the neutronics simulations by a dedicated multiphysics interface.

The mesh grid specifications are given in Table 4.3.

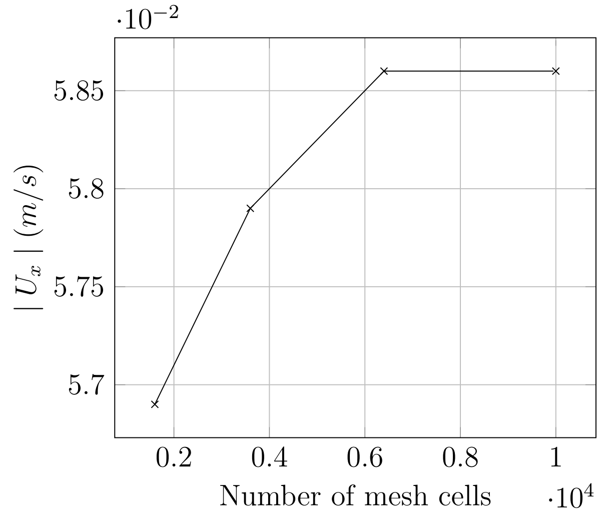


Figure 4.4: Average flow velocity as a function of mesh cell number

| Participant | Mesh grid |
|---|---|
| CNRS | nonuniform 200 × 200 |
| PoliMi | uniform 400 × 400 |
| PSI | nonuniform 200 × 200 |
| TU Delft | uniform 50 × 50 |
| DTU | nonuniform 60 × 60 |

Table 4.3: Mesh grids used by benchmark participants. Non-uniform refers to a mesh refined towards the cavity walls.

All the numerical schemes used by the participants for the integration of time derivatives, diffusion and convection terms, are reported in Table 4.4.

| Term | Participant | Numerical Scheme |
|---|---|---|
| Diffusive | CNRS | Gauss Linear |
| | PoliMi | Second order schemes |
| | PSI | Upwind centered |
| | TU Delft | DG-FEM |
| | DTU | Gauss linear corrected |
| Convective | CNRS | Gauss upwind |
| | PoliMi | Second order schemes |
| | PSI | Upwind biased |
| | TU Delft | DG-FEM |
| | DTU | Gauss upwind |
| Time derivative | CNRS | Implicit Euler |
| | PoliMi | Implicit Euler |
| | PSI | Implicit backward Euler |
| | TU Delft | BDF |
| | DTU | Implicit Euler |

Table 4.4: Numerical Schemes used by the benchmark participants.

To quantify the relative differences between different participants in absence of a reference solution, a metric is employed, defined as follows:

$$\epsilon_C = \sqrt{\frac{\sum_{i=1}^{N_p}\left(Q_c(\boldsymbol{r}_i) - Q_{avg}(\boldsymbol{r}_i)\right)^2}{\sum_{i=1}^{N_p} Q_{avg}^2(\boldsymbol{r}_i)}} \tag{4.2}$$

here $\boldsymbol{r}_i$ denotes a point on the sample line, $N_p = 201$ is the number of points used to sample the quantity $Q_C$ for a given code $C$. From definition of $\epsilon_C$ it can be seen, that the expected result is the average value of $Q_{avg}(\boldsymbol{r}_i) = \frac{1}{N_C}\sum_{c=1}^{N_C} Q_c(\boldsymbol{r}_i)$ excluding the code under scrutiny. A collected measure for the full set of codes is taken to be the averaged value of $\epsilon_C$ : $\epsilon = \frac{1}{N_C}\sum_{C=1}^{N_C}\epsilon_C$.

## 4.3 Single physics verification

In this section, the thermal-hydraulics results obtained by OpenFOAM v6.0 and the neutronics results obtained in Serpent 2.31 are compared to the benchmark results. No coupling whatsoever is present at this stage, the independent single-physics solvers are being tested.

1. A thermal-hydraulics standalone test is performed and the resulting velocity field is compared to the results of other participants.
   As an input, uniform temperature of 900 $K$ and lid velocity of 0.5 $m/s$ is assumed. Velocity horizontal components along the cavity vertical centerline are compared.
   In Figure 4.5 the velocity horizontal component along the vertical centerline is plotted and compared to the benchmark results and the velocity magnitude distribution is shown.
   The agreement between all participants is excellent, which is not surprising because the cavity problem is just a standard CFD problem and should be well resolved by all CFD solvers in use. The average discrepancy between DTU results and the results of the other participants is ≈ 0.55% for the vertical centerline. The other participants report 0.8% relative discrepancy between their results for the vertical centerline.

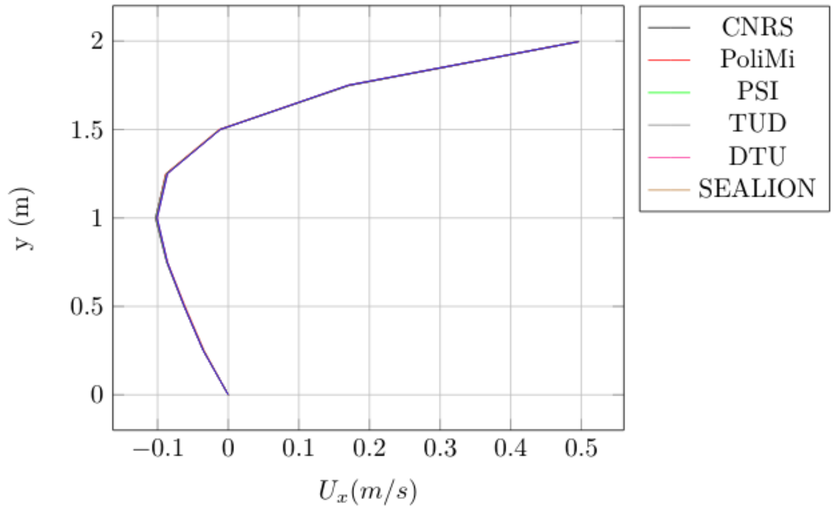


Figure 4.5: Velocity horizontal component along the vertical centerline of the cavity

   The slight difference between the DTU results and the other participants is attributed to the differences in the mesh grid (different coarseness and different handling of mesh refinement).

2. A neutronics standalone test is performed and the fission rate density distributions are compared.
   As an input, a uniform fuel temperature of 900 $K$ and a power level of 1 $GW$ are

taken.

The fission rate density along the vertical centerline is extracted for comparison (Figure 4.6) as well as the initial excess reactivity of the system (Table 4.5)is reported. To achieve high statistics, $10^6$ neutrons per cycle were simulated in Serpent, with 300 active and 50 inactive cycles respectively. The simulation took approximately 4 hours to complete running in parallel on 30 $Intel(R)Xeon(R)CPUE5-2680v2@2.80GHz$ processors.

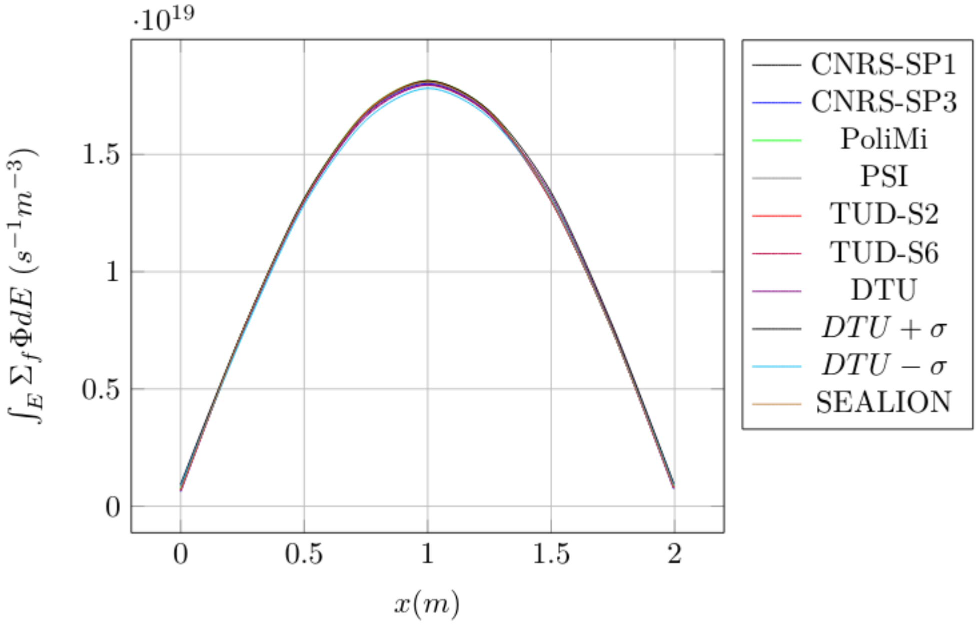


Figure 4.6: Fission rate density along the horizontal centerline

| Participant | $\rho(pcm)$ |
|---|---|
| CNRS-SP1 | 411.3 |
| CNRS-SP3 | 353.7 |
| PoliMi | 421.2 |
| PSI | 411.7 |
| TU Delft-S2 | 482.6 |
| TU Delft-S6 | 578.1 |
| .DTU | 739 ±7.3 |
| SEALION | 741±2.5 |

Table 4.5: The excess reactivity of the system

The agreement between the DTU results and the other participants for the fission rate density is good, with an average relative difference of 1.6%. The values at boundaries are higher in the DTU case than for the other participants and it is attributed to the different treatment of the vacuum boundary conditions for diffusion and higher-order transport codes.

Regarding the system initial reactivity, it can be noticed, that the participants have a relative difference of up to 224 pcm (e.g. TU Delft-S6 vs CNRS SP-3) which is considered acceptable, and as the later results show, it does not seem to have an impact on the e.g. temperature distributions. The DTU approach which uses a highly accurate neutronics modelling approach, predicts a 253 pcm higher reactivity value as compared to the average of all the other results. The difference for the DTU results is mainly attributed to two reasons; DTU method adopts highly accurate Monte-Carlo code for the neutronics part, which can be expected to give some differences when compared to other, more simplified transport codes. Apart from this, it has to be noted as well, that differently from other participants, DTU and SEALION approaches initially used JEFF-3.2 nuclear data libraries instead of JEFF-3.1. The reason for this is the incompatibility of the Serpent-2 release used in this work with JEFF-3.1 libraries for some nuclides (interested readers can check more about this issue here: (`https://ttuki.vtt.fi/serpent/viewtopic.php?f=24&t=2328&p=6444&hilit=ENDF+Law+67+not+supported#p6444`).It was possible later on to obtain a JEFF-3.1 data library version compatible with Serpent-2 and to quantify the relative difference between the two libraries. The difference in excess reactivity when evaluated with Serpent2 using JEFF-3.1 vs JEFF-3.2 is 386 pcm which is consistent with the difference between the average of the other participant and the DTU and SEALION codes. For the fission rate densities, the relative difference was found to be $\approx$ 0.2%, whereas for the temperature distributions the relative difference when using the heat source generated with the two libraries went down to $\approx$ 0.02% (see Additional Plots in this chapter). It was thus decided to not repeat the benchmark calculations with the JEFF-3.1, as the systematic uncertainties coming from the data libraries are relatively very small.

3. Temperature testing is performed considering a one-way coupling in form of the fission heat source being imported from the neutronics solver.
   Fixed velocity field from the thermal-hydraulics standalone test and a volumetric heat source from the neutronics standalone test are taken as an input.
   Non-uniform temperature distribution is obtained and shown in Figure 4.7. The agreement with the benchmark is very good, with the relative discrepancy between the DTU and other participants being $0.2 - 0.4\%$ and these discrepancies are attributed to the accumulation and superposition of the discrepancies coming from the standalone thermal-hydraulics and neutronics modules.

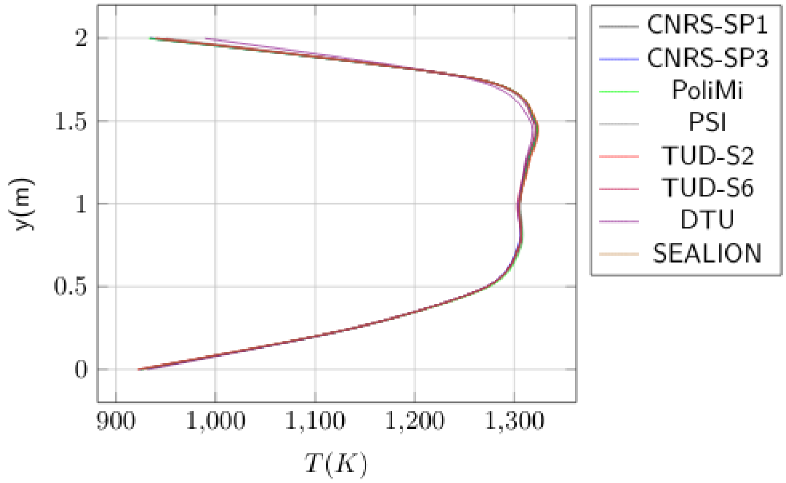


Figure 4.7: Temperature distribution along the vertical centerline

This concludes the single-physics verification stage. The CFD solver used in this work agrees very well with the other participants in predicting the velocity and temperature fields. The neutronics module in Serpent is also in good agreement with the other neutronics models and the noticed discrepancies in the excess reactivity are due to the usage of a different transport method and a different nuclides library.

## 4.4 Steady State verification

1. An investigation of the impact of liquid fuel on the delayed neutron precursors is undertaken at this step.
   The neutron transport problem is solved in the presence of fuel motion, with fixed fuel velocity from thermal-hydraulics standalone test, with T=900 $K$ and P=1 $GW$. The delayed neutron precursor source is sampled along the vertical centerline, shown in Figure 4.8. Additionally, reactivity loss concerning the stationary fuel due to the movement of delayed neutron precursors is reported in Table 4.6.

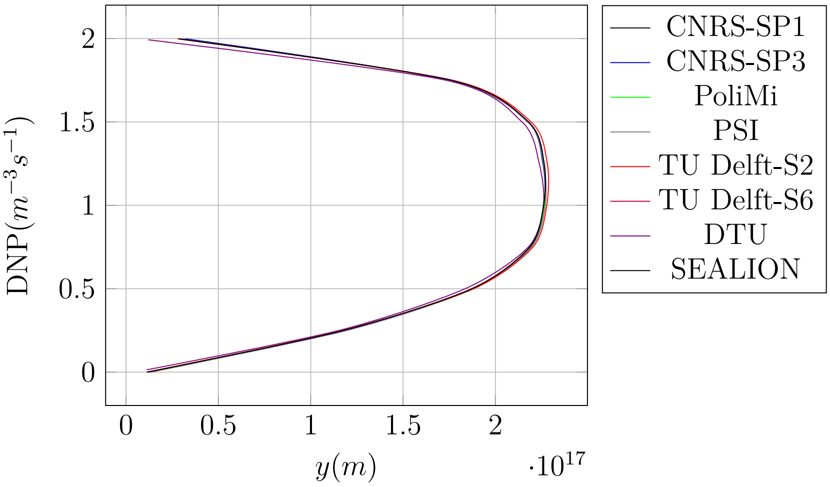


Figure 4.8: DNP source along the vertical centerline

The DTU results for the delayed neutron precursor source have a relative discrepancy of 2.8% compared to the other participants. This is attributed to the systematic discrepancies accumulating from the previous steps.

| Participant | $\rho - \rho_0(pcm)$ |
|---|---|
| CNRS-SP1 | -62.5 |
| CNRS-SP3 | -62.6 |
| PoliMi | -62 |
| PSI | -63 |
| TU Delft-S2 | -62 |
| TU Delft-S6 | - 60.6 |
| DTU | - 60±15 |
| SEALION | -55.2 |

Table 4.6: Reactivity change compared to the stationary fuel due to the movement of delayed neutron precursors

It can be appreciated, that all participants report very close reactivity losses, with a difference of only a few $pcm$.
The distribution of the 1st and the 5th groups of the delayed neutron precursors is given in Figure 4.9. The 1st group of the DNPs, with a half-life of 55 $s$, has enough time to be transported with the flow and to diffuse. Consequently, the DNPs from group 1 will have a vertically shifted horizontally expanded spatial distribution. Group 5 with a half-life of only 2.37 $s$, decays closer to the original production

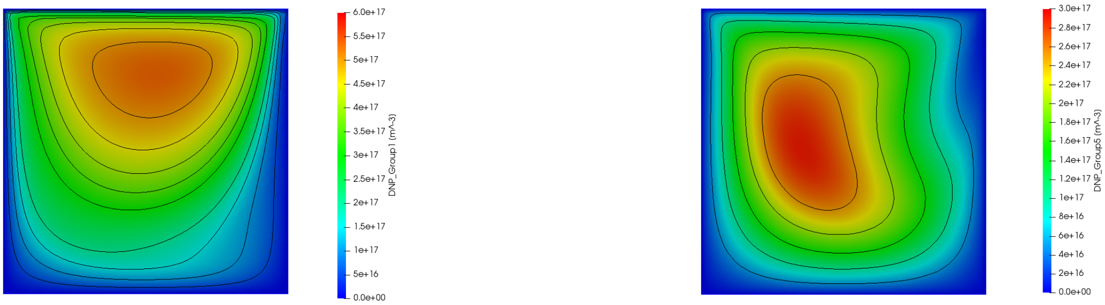


Figure 4.9: Distribution of the long-lived (Group 1) and short-lived (Group 5) DNP groups.

location as there is not enough time for the DNPs to be transported further before they decay.
Figure 4.9 illustrates the important impact of the fuel velocity field on the DNPs and the relation of the velocity magnitude and characteristic re-circulation time to the DNP half-lives.

In this case, the geometry does not have an inlet or an outlet, but it is easy to imagine, that if there would have been an external loop to the main fuel domain, e.g. a heat exchanger or other piping, some of the DNPs especially from the long-living groups could have ended up outside the core before decaying. The transport of delayed neutron precursors results in reactivity loss reported in Table 4.6.

2. In this step the two-way coupling between the non-uniform temperature distribution on the flux shape and the impact of the flux shape on the temperature distribution is investigated.
Fixed velocity field from thermal-hydraulics standalone test, P= $1GW$ and the volumetric heat transfer coefficient of $= 10^6 W m^{-3} K^{-1}$ are taken as an input.
The density feedback of the fuel is evaluated by showing the fission rate density change compared to the stationary fuel case presented before. Additionally, the temperature distribution with the new volumetric heat source is obtained. Finally, the reactivity change is also calculated compared to the previous step with circulating fuel. Table 4.7 shows the reactivity change from the previous step. There is a satisfactory agreement between all results. The maximum difference between the original benchmark participants is 39 pcm, the maximum difference between DTU results compared to the other participants is 67 pcm.

| Participant | $\rho - \rho_1 (pcm)$ |
|---|---|
| CNRS-SP1 | -1152.0 |
| CNRS-SP3 | -1152.7 |
| PoliMi | -1161.0 |
| PSI | -1154.8 |
| TU Delft-S2 | -1145.2 |
| TU Delft-S6 | - 1122.0 |
| DTU | - 1094.0± 103 |
| SEALION | -1217.5±30 |

Table 4.7: Reactivity change from Step 1.1

Figure 4.10 shows the the fission rate density change compared to Step 0.2 as a result of flux shape-temperature coupling manifested via fuel density feedback. The discrepancy between the results of all participants as well as between the DTU and other results is larger at this step and is attributed to:

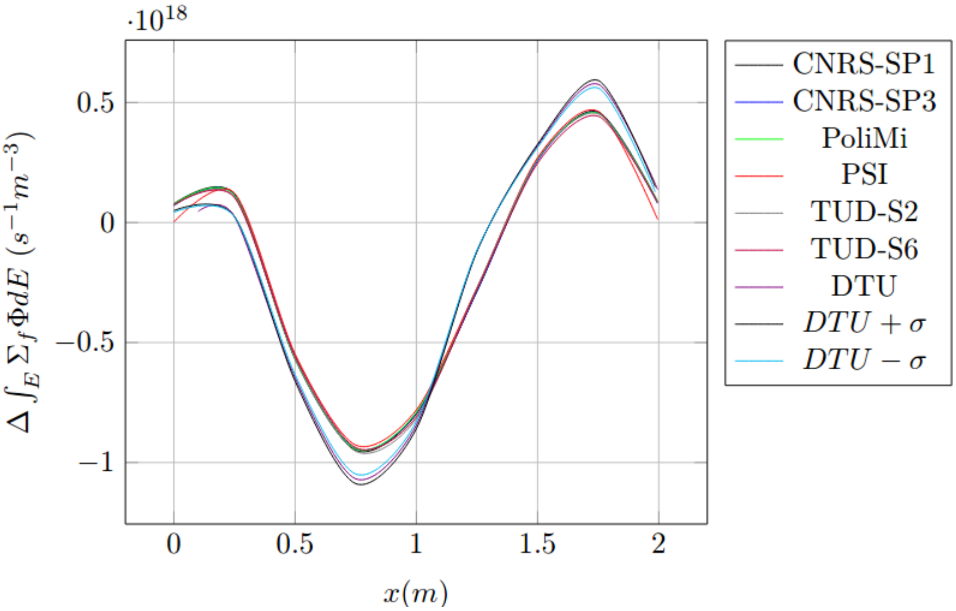


Figure 4.10: Fission Rate Density Change from Step 0.2

- Accumulated discrepancies from previous steps
- Differences in temperature dependent cross-section treatment for some participants; TU Delft, SEALION and DTU adopt element-wise cross-section treatment, the other participants use element-wise treatment method, which causes some difference between the results of the other participants and subsequently, also between the DTU and other results.
- The method of the coupling: fission rate density difference probed at this stage is a second-order effect, which means that the coupling methods and solvers are probed at a deeper level to analyze their degree of predicting identical results. Considering the different approach adopted by DTU for solving the neutron transport problem, as well as the different approach of the DNP movement effects being accounted for, it is not unexpected to have larger discrepancies at this level of comparison between the DTU results and the other participant results.

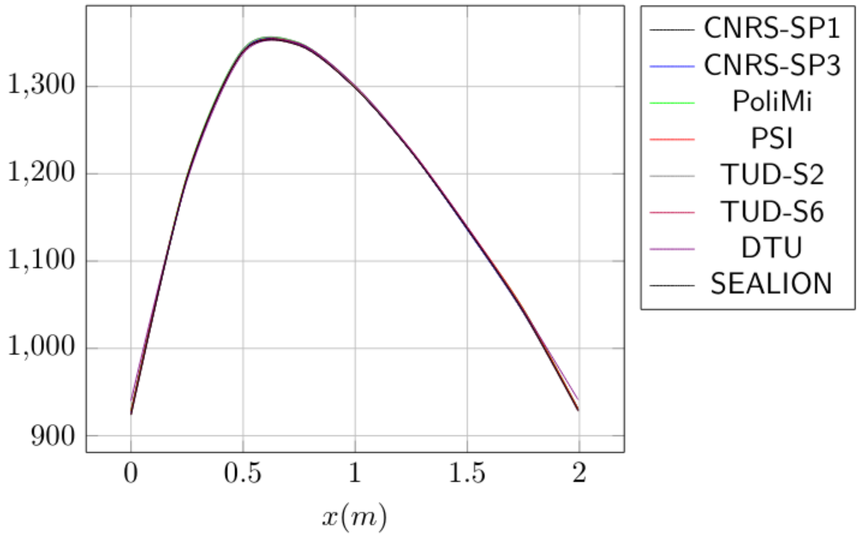


Figure 4.11: Temperature profile along the horizontal centerline

However, as for the temperature profile the volumetric power normalized by the total power level is used, we do not see any significant difference in the temperature profiles compared to other participants, as shown in Figure 4.11.

3. The impact of buoyancy on the velocity, temperature and delayed neutron precursor source is investigated at this stage.
P= $1GW$ power, U=0 $m/s$lid velocity and a heat transfer coefficient = $10^6 Wm^{-3}K^{-1}$ are considered as an input.
In Figure,4.12 the velocity distribution along the horizontal centerline is depicted. A fair agreement between DTU results and the benchmark can be observed, with an average difference of 3.85%

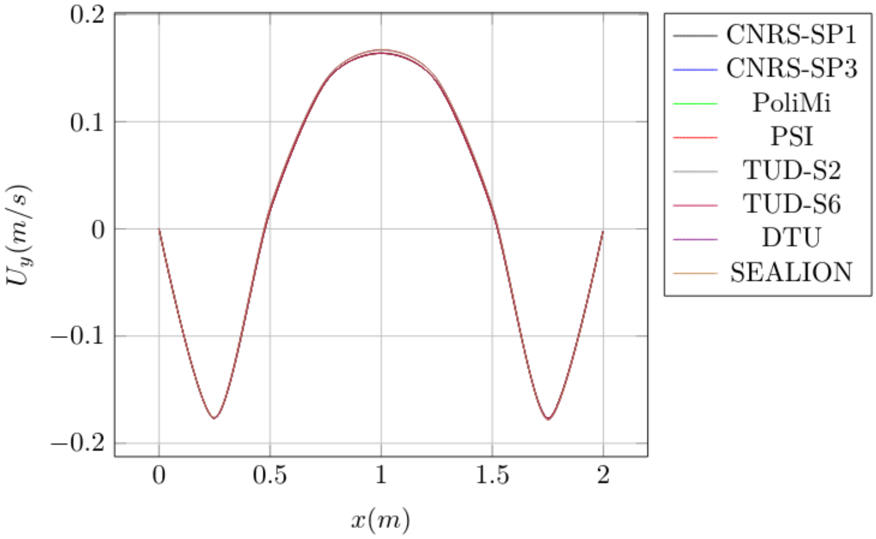


Figure 4.12: Velocity profile along the horizontal centerline

This holds true also for the temperature distribution shown in Figure 4.13. The difference between our results and the benchmark results is less than 2%.

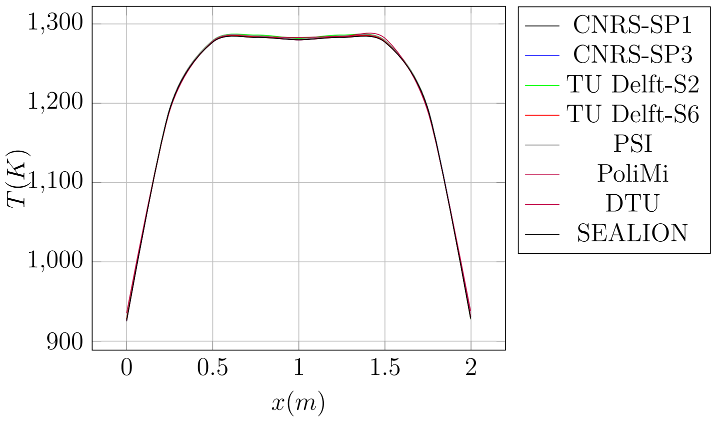


Figure 4.13: Temperature distribution along the horizontal centerline

In Figure 4.14 the delayed neutron precursor source distribution along the horizontal centerline is depicted. The agreement with all the other results is very good.

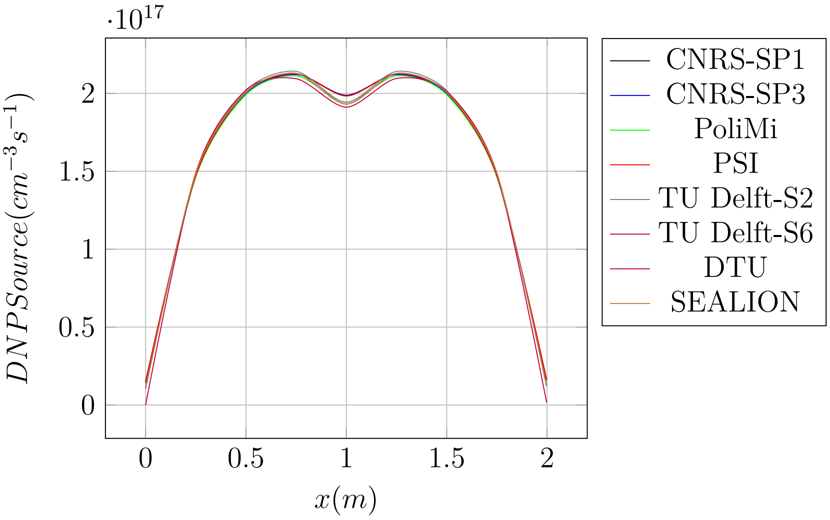
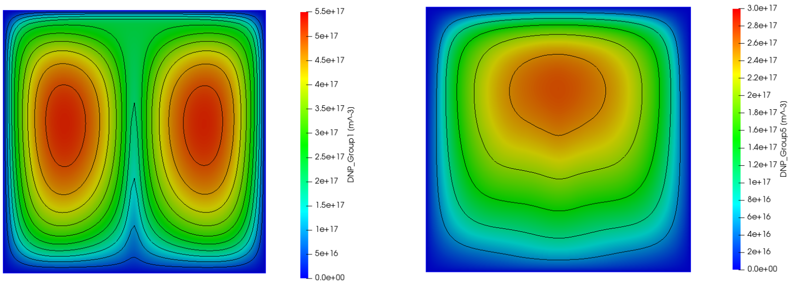


Figure 4.14: DNP Source along the horizontal centerline

The Figure 4.15 shows the distribution of the long and short-lived DNPs in presence of buoyancy.



(a) Group 1 (b) Group 5

Figure 4.15: DNP group distribution for Step 3

| Participant | $\rho_{1.3} - \rho_{0.2}(pcm)$ |
|---|---|
| CNRS-SP1 | -1220.5 |
| CNRS-SP3 | -1220.7 |
| PoliMi | -1227 |
| PSI | -1219.6 |
| TU Delft-S2 | -1208.5 |
| TU Delft-S6 | - 1184.4 |
| DTU | -1249± 25 |
| SEALION | -1218.2 ±22 |

Table 4.8: Reactivity change compared to Step 0.2

Again, Group 1 lives long enough to assemble the shape of the velocity profile, whereas Group 5 is less relocated from the production positions before decaying. The impact of the DNP transport on the reactivity is quantified in Table 4.8. The maximum difference between the initial participants is 43 pcm, the difference between DTU and other results is 65 pcm.

The agreement between the DTU results and the benchmark is considered fair and statistically compatible with the results of the other benchmark participants.

4. Everything tested in the previous tests is combined together in this step where a full coupling scenario is tested.

   The power and the velocity are changed in steps of 0.1 $m/s$ and 0.2 $GW$ between (0, 0.5) $m/s$ and (0, 1) $GW$ respectively.

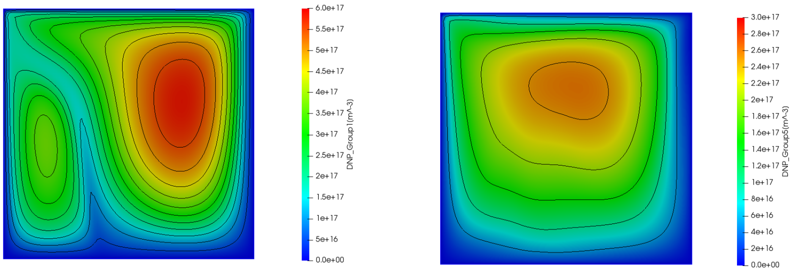
   The reactivity as a function of velocity and power is analysed. In Table 4.9 the results for two ultimate cases of 0 $m/s$ and 0.5 $m/s$ velocities are shown.

| Code | U($m/s$) | P=0.2 ($GW$) | P=0.4 ($GW$) | P=0.6 ($GW$) | P=0.8 ($GW$) | P=1 ($GW$) |
|---|---|---|---|---|---|---|
| CNRS-SP1 | 0 | -264.5 | -503.4 | -737.8 | -974.2 | -1220.5 |
| CNRS-SP3 | | -265.8 | -503.4 | -738.5 | -976.2 | -1220.7 |
| PoliMi | | -266 | -498 | -734 | -975 | -1227 |
| PSI | | -268 | -504.8 | -739 | -975 | -1215.1 |
| TU Delft-S2 | | -236.7 | -498.1 | -731.1 | -967.2 | -1208.5 |
| TU Delft-S6 | | -258 | -487.8 | -716.3 | -947.9 | -1184.4 |
| DTU | | -206±28 | -440 ± 30 | - 728 ± 42 | -913± 27 | -1118± 29 |
| CNRS-SP1 | 0.5 | -276.5 | -503.5 | -732.9 | -966.3 | -1204.8 |
| CNRS-SP3 | | -276.8 | -503.5 | -733 | -966.5 | -1205.2 |
| PoliMi | | -284 | -508 | -737 | -972 | -1214 |
| PSI | | -278.1 | -504.6 | -733.1 | -964.3 | -1198.8 |
| TU Delft-S2 | | -273.1 | -497.8 | -725.2 | -956.4 | -1193 |
| TU Delft-S6 | | -267.5 | -487.8 | -710.8 | -937.6 | -1169.7 |
| DTU | | -289±30 | -468±27 | -655±28 | -836±31 | -1093± 33 |

Table 4.9: Reactivity change from Step0.2

A satisfactory agreement can be noticed. The amount of the negative reactivity inserted is determined by one of the two key phenomena prevailing: delayed neu-

tron precursor re-distribution due to the velocity field or the negative temperature feedback mechanism due to the increasing power. It can be seen, that for low power levels increasing the velocity leads to slightly higher negative reactivity insertion, however, from P=0.6 $GW$ the effect reverses due to the positive reactivity insertion related to the shift of the temperature pick.
The delayed neutron precursor group-wise distributions are shown in Figure 4.16.



(a) Group 1

(b) Group 5

Figure 4.16: DNP Group distribution for Step 4

## 4.5 Transient verification

Started from the steady-state reached at the previous Step, with $\mathbf{u} = 0.5 m/s$ and $P = 1GW$, the volumetric heat transfer coefficient is perturbed according to a sin wave with several frequencies. The power gain is analyzed as a function of frequency. In this work, only one frequency is tested: f=0.025 $Hz$. The results are presented in Table4.10. For this frequency, the gain computed by coupled Serpent2-OpenFOAM approach is 0.713 and the phase shift is 27 ° as compared to the average of 0.835 and 18.6 ° obtained by the benchmark participants. This relatively large discrepancies are mainly attributed to the sampling time-step used in Serpent2-OpenFOAM coupling approach. The time-step chosen in coupled Serpent-OpenFOAM approach leads to $4 \times 10^3$ data points being sampled for the f=0.025 $Hz$, meanwhile the time-step chosen by other participants results in $10^4$ data points being sampled.

$$G = \frac{(P_{max} - P_{ave})/P_{ave}}{(\gamma_{max} - \gamma_{avg})/\gamma_{avg}} \tag{4.3}$$

| Code | Gain | Phase Shift (in $^o$) |
|---|---|---|
| DTU | 7.130e-01 | -2.700e+01 |
| CNRS-SP1 | 8.399e-01 | -1.944e+01 |
| CNRS-SP3 | 8.392e-01 | -1.933e+01 |
| PoliMi | 8.357e-01 | -1.945e+01 |
| PSI | 8.280e-01 | -1.800e+01 |
| TU Delft-S2 | 8.349e-01 | -1.620e+01 |
| TU Delft-S6 | 8.332e-01 | -1.620e+01 |

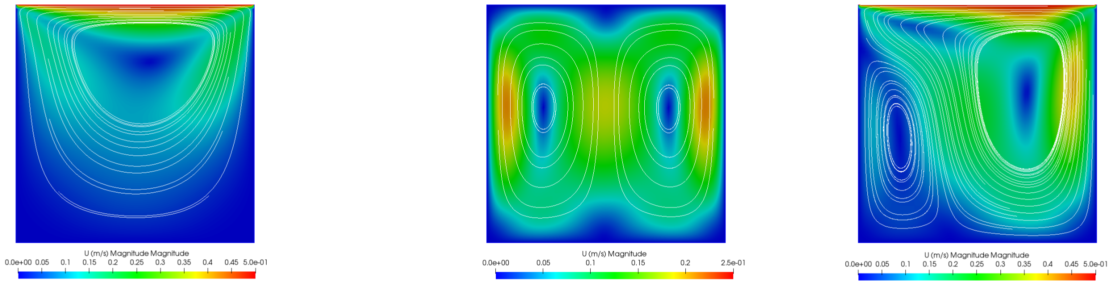
Table 4.10: Power gain and phase shift for f=0.025 $Hz$ perturbation frequency

## 4.6 Conclusion

In this Chapter, the external coupling between Serpent and OpenFOAM already described in Chapter 3 is modified to allow for delayed neutron precursor mesh-based tracking and transport which is realized in OpenFOAM. The method is tested based on the multiphysics benchmark for the molten salt fast reactor published recently [2]. The agreement is sufficiently good; the coupled approach performs very well for the steady-state regime and captures correctly the impact of the fuel movement on the DNP distribution and the reactivity. The fuel density feedback mechanism is also well represented via this coupling approach. For the specific transient modelled in the benchmark very small time-steps are imposed to capture the periodic trend of the power. This could be realizable without additional computational burden for the simplified neutronics codes, but to do the same in Serpent and additionally to ensure high statistics is computationally expensive. Considering that the coupled solvers have to be iterated within the time-step to ensure convergence, the method hits a bottleneck for the transient regime and is tested only for one parameter set of the transient test case. In order to have a confidence that the method is verified for transient scenarios, a more extensive testing is required. However, as the physical behavior of the system is captured correctly and it is well shown in the steady-state scenario, the method can be applied for transients with the use of sufficiently small time-steps to capture the transient response of the system. However, with the current implementation of the coupling approach this can be achieved at a cost of computational time and resources.

## Additional plots

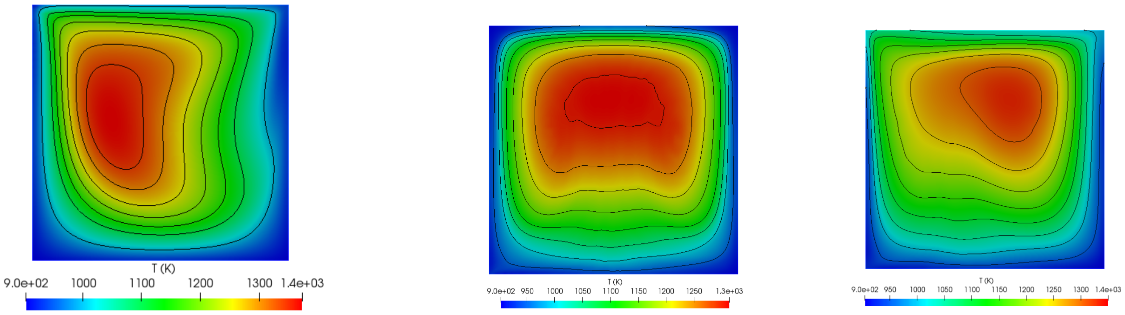
Several additional plots for the velocity and the temperature distributions from different benchmark steps are presented below. In Figure 4.17 velocity profiles are plotted for the benchmark step 1 (standalone thermal-hydraulics module testing), step 2 (buoyancy effects on the flow distribution), and step 3 (full coupling in steady-state).



(a) Step 1 (b) Step 2 (c) Step 3

Figure 4.17: Velocity profiles for benchmark steps 1, 2 and 3

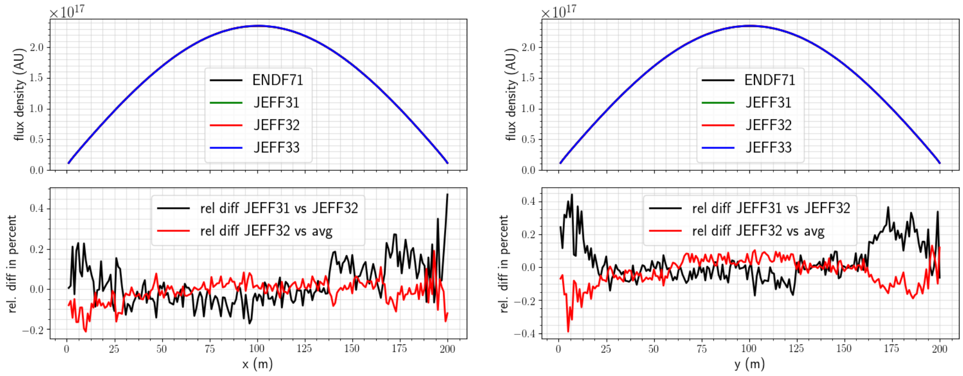
For each of these velocity profiles obtained at the different steps of the benchmark, the temperature profiles are shown in Figure 4.18.



(a) Step 1 (b) Step 2 (c) Step 3

Figure 4.18: Temperature profiles for for benchmark steps 1, 2 and 3

Finally, a comparison has been carried out between the JEFF-3.1 vs JEFF-3.2 libraries to quantify the relative difference. In Figure 4.19 the flux density distributions are compared, and the relative difference is plotted as well. The relative difference between JEFF-3.2 and JEFF-3.1 are around 0.2% and are comparable to the relative difference between JEFF-3.2 results vs the average results of the other benchmark participants.

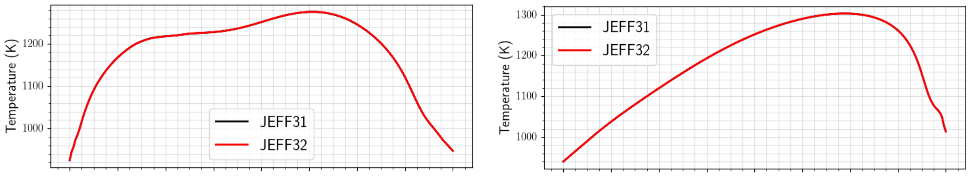


(a) Flux density distribution for the horizontal centerline

(b) Flux density distribution for the vertical centerline

Figure 4.19: Flux density profiles obtained with JEFF-3.1 and JEFF-3.2

Similarly, in Figure 4.20 the temperature profiles are shown obtained with the heat sources calculated by the two libraries and a relative difference of only 0.02% is observed.



(a) Temperature distribution for the horizontal centerline

(b) Temperature distribution for the vertical centerline

Figure 4.20: Temperature profiles obtained with JEFF-3.1 and JEFF-3.2

# Bibliography

[1] H. Boussier et al., "The Molten Salt Reactor in Generation IV: Overview and Perspectives", Proceedings of the Generation4 International Forum Symposium, San Diego, USA, 2012

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

[3] `https://www.cfd-online.com/Wiki/Lid-driven_cavity_problem`

[4] M. Tiberga, D. Lathouwers, J.L. Kloosterman, "A discontinuous Galerkin FEM multiphysics solver for the Molten Salt Fast Reactor", International Conference on Mathematics and Computational Methods applied to Nuclear Science and Engineering (M&C 2019). Portland, OR, USA..

[5] J. A. Blanco,P. Rubiolo, E. Dumonteil, "Neutronic modelling strategies for a liquid fuel transient calculation", Physics of Reactors Conference (PHYSOR) 2020, Cambridge, UK.

[6] E. Cervi, S. Lorenzi, A. Cammi, L. Luzzi, "An Euler-Euler multi-physics solver for the analysis of the helium bubbling system in the MSFR.", 26th International Conference Nuclear Energy for New Europe, 2017, Bled, Slovenia.

[7] E. Cervi, S. Lorenzi, A. Cammi, L. Luzzi, "Development of a multiphysics model for the sTU Delfty of fuel compressibility effects in the Molten Salt Fast Reactor.", Chem. Eng. Sci. 193, 379–393.

[8] E. Cervi, S. Lorenzi, L. Luzzi, A. Cammi, "An OpenFOAM solver for criticality safety assessment in dynamic compression events.", Transactions of the American Nuclear Society, Orlando, USA, vol. 119, pp. 855–858..

[9] E. Cervi,S. Lorenzi, A. Cammi, L. Luzzi, "Development of an SP 3 neutron transport solver for the analysis of the Molten Salt Fast Reactor.", Nucl. Eng. Des. 346, 209–219.

[10] E. Cervi, S. Lorenzi, L. Luzzi, A. Cammi, " Multiphysics analysis of the MSFR helium bubbling system: a comparison between neutron diffusion, SP 3 neutron transport, and Monte Carlo approaches.", Ann. Nucl. Energy 132, 227–235.

[11] C. Fiorina et al., "GeN-Foam: a novel OpenFOAM based multi-physics solver for 2D/3D transient analysis of nuclear reactors.", Nucl. Eng. Des. 294, 24–37.

[12] A.V. Pedersen et al., "Seaborg External multiphysics Architecture for Licensing and Ip development Of Nuclear reactors.", EUROSTARS refnr 11837 project, 2018.

# Chapter 5

# CFD Simulation of DYNASTY natural circulation loop: Experimental Validation

For some advanced Generation III+ and Generation IV reactor design concepts passive safety systems intended for decay heat removal are one of the focal points. The importance of passive cooling for the increased plant reliability was highlighted in e.g. the analysis of the Fukushima accident [1]. In general, nuclear reactors where reactivity control, core cooling, and prevention of any sort of radioactive substance release is to be prevented under a wide range of normal operating conditions as well as in case of an accident, will benefit significantly from adopting passive cooling systems. However, in contrast to conventional reactors where the passive heat removal can be realized utilizing localized heat sinks and sources and is quite well studied and understood as a thermo-dynamical phenomenon, in Molten Salt Reactors the heat removal through natural circulation means dealing with an internal heat generation, i.e. the heat source is distributed. Internal heat generation might alter the equilibrium stability of the system. In this Chapter, a complete CFD analysis of such a natural circulation loop (NCL): DYNASTY, is performed. The goal of the study is to assess the CFD model against experimental results to establish whether the model predicts the same equilibrium state for given operational conditions. Additionally, the study tested several turbulence models to establish one that agreed the most with the experimental results. This is important, because the transition of a natural circulation flow from laminar to the turbulent regime is not well studied analytically, and no direct conclusions can be made whether the flow will behave as laminar or turbulent based solemnly on the characteristic parameters, e.g. the *Re* number.
The main results of the analysis are submitted to the Chemical Engineering Science Journal: "Computational Fluid Dynamics Modelling of the DYNASTY Loop", A.Nalbandyan, A.Cammi, S.Lorenzi, E.B. Klinkby, B. Lauritzen.

## 5.1 Description of the DYNASTY loop

DYNASTY is an NCL built to operate with molten salts as well as with water or glycol. The schematic of the facility is shown in Figure 5.1.

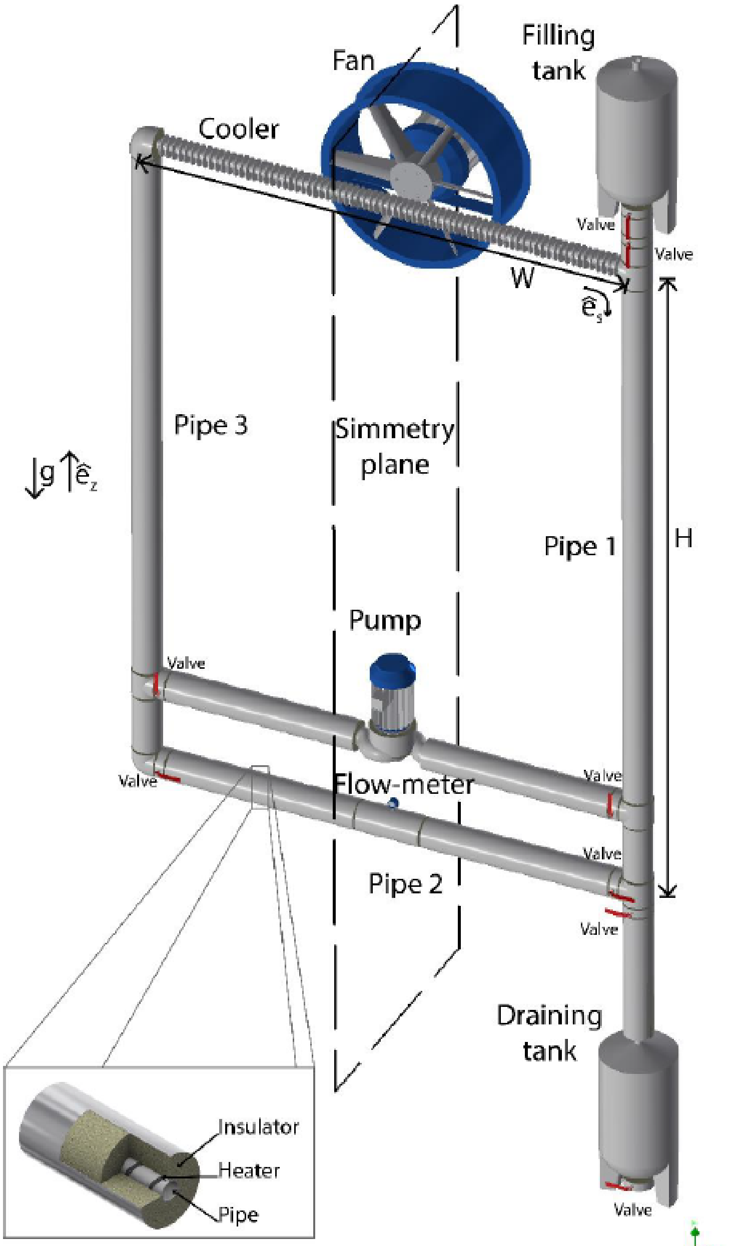


Figure 5.1: Schematic view of DYNASTY

The facility is divided into five sections: the two vertical legs (Pipe 1 and Pipe 3), the horizontal cooler at the top, and the two horizontal legs at the bottom. Depending on the operational regime, only one of the bottom horizontal legs is connected to the rest of the system; in the case of natural circulation, the leg with flow meter is connected (Pipe 2) and for the forced circulation the leg with the pump is connected. There are also filling and draining tanks connected to one of the vertical legs. For the experiment considered in this work, the loop operated in a natural circulation regime, with water as a heat carrier. As internal heat generation is not an easily feasible engineering process and could potentially pose control challenges, all-external heating via electrical stripes wrapped around the pipes is applied. The schematic is highlighted in Figure 5.1. During an experiment, all sections except the cooler can be heated. The cooler is finned to increase the heat transfer efficiency and is coupled to a fan for heat removal. The heating

mechanism can be either distributed (all three legs heated) or localized with either the horizontal or the vertical legs being heated.
The main working and geometrical parameters for the loop are summarized in Table 5.1.

| **Parameter** | **Dimension** | **Unit** |
|---|---|---|
| Height | 3.09 | $m$ |
| Width | 3.10 | $m$ |
| Pipe inner diameter | 0.03816 | $m$ |
| Pipe wall thickness | 0.002 | $m$ |
| Pressure | 101325 | $Pa$ |
| Temperature range | 299.15-353.15 | $K$ |

Table 5.1: Main geometrical and operational data for DYNASTY (water as working fluid)

In addition to the Coriolis-effect mass flow rate meter, thermocouple sensors are also installed on the loop both for the pipes and for the water temperature measurements. The water temperature is measured on the cooler inlet, cooler outlet, Pipe 1 outlet, and Pipe 3 inlet. In the case of distributed heating, the temperature difference across the cooler is the driving force for the flow in natural circulation.
The facility is made from AISI-316 stainless steel and the thermal insulation is made out of Rockwool.
For a more detailed description of the facility, the reader is referred to e.g. [2].

## 5.2 The CFD model

In Figure 5.2 the CAD model adapted for the simulations is depicted. The model is built in ANSYS Spaceclaim. Several simplifications are made for the CFD model; the mass flow rate meter and the draining tank are not modelled, however, as it will be discussed later on, the mass flow rate meter impact is encountered through porous media. The highlighted section shows the water filling tank, modelled as a porous medium as well to encounter pressure losses.
The model meshing is done in ANSYS Workbench. The pipes are meshed using hexahedral structured mesh except for the filling tank region. The liquid part is meshed as a single part using unstructured tetrahedral mesh (Figure 5.3). Meshes of multiple coarseness are tested. The main mesh parameters are reported in Table 5.2 and are complying with standard meshing guidelines for ANSYS Workbench suggesting an average orthogonal quality above 0.2 and a maximal skewness less than 0.95. These characteristics show how close the mesh elements are to the optimal size, e.g.

$$Skewness = \frac{l_0 - l_a}{l_0} \tag{5.1}$$

where $l_0$ is the optimal cell size and $l_a$ is the actual size.

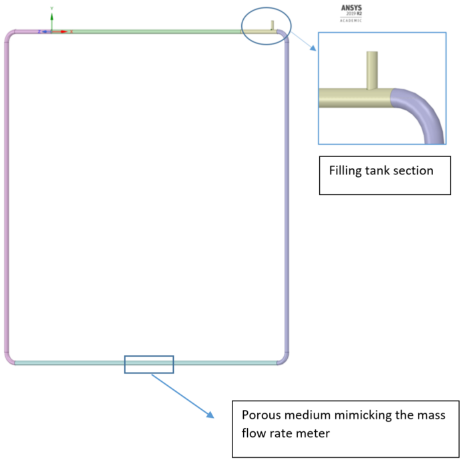


Figure 5.2: DYNASTY CAD model

| Mesh | Coarsest | Coarse | Fine | Finest |
|---|---|---|---|---|
| Element Number | $\approx 1\times10^6$ | $\approx 4\times10^6$ | $\approx 6\times10^6$ | $\approx 8\times10^6$ |
| Skewness | 0.68 | 0.87 | 0.84 | 0.81 |
| Orthogonal quality | 0.96 | 0.95 | 0.950 | 0.93 |

Table 5.2: Mesh Data

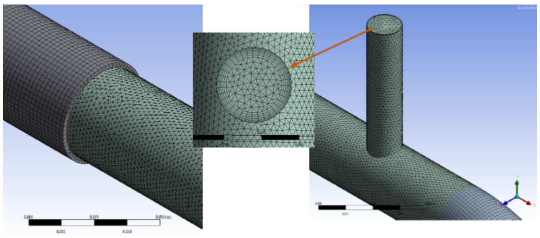


Figure 5.3: DYNASTY mesh: Closeup at inflation layers applied on water body

Together with the mesh orthogonal quality and skewness, the dimensionless wall distance otherwise known as the wall $y^+$ value is an important parameter that is crucial to construct a correct mesh that resolves the boundary layer flow. The $y^+$ is defined as

$$y^+ = \frac{\rho \mathbf{u}_\tau \Delta y}{\mu} \tag{5.2}$$

where $\Delta y$ is the first cell height of the mesh boundary layer, $\rho$ is the density, $\mu$ is the dynamic viscosity, and $\mathbf{u}_\tau$ is the frictional velocity calculated as:

$$\mathbf{u}_\tau = \sqrt{\frac{C_f}{2}\mathbf{u}} \tag{5.3}$$

here u is the free stream velocity and $C_f$ is the skin friction factor; an empirical coefficient determined for internal flows as [14]:

$$C_f = 0.079 Re^{-0.25} \tag{5.4}$$

The dimensionless wall distance helps to describe the near-wall flow, which is subject to numerical and modelling challenges due to the viscosity-induced effects. Usually, the near-wall flow can either be modelled by resolving the viscous sub-layer, or by adopting wall functions to approximate the flow behavior across it. If the viscous sub-layer is to be resolved, very fine mesh is usually required, with a $y^+ \approx 1$. This is frequently not viable for a large industrial model and thus wall functions are generally adopted for near-wall flow modelling. The validity of applying wall functions depends on the flow regime ($Re$ number) and the turbulence model used.

For the CFD analysis, the OpenFOAM customized CFD toolbox is used, specifically, the chtMultiRegionFoam solver for conjugate heat transfer [3], [4]. *PIMPLE* algorithm is used for the coupled momentum-pressure equations system. As mentioned earlier, the filling tank and the mass flow rate meter are included as porous media [5]. The localized pressure loss is represented as

$$\Delta P = \zeta \frac{\rho \mathbf{u}^2}{2} \tag{5.5}$$

where $\zeta$ is the pressure loss coefficient. In OpenFOAM the velocity-dependent pressure gradient can be further broken down to viscous and inertial contributions using the Darcy-Forcheimer model:

$$\nabla P = (\mu D + \frac{1}{2}\rho F \mathbf{u})\mathbf{u} = \mu D \mathbf{u} + \frac{1}{2}\rho F \mathbf{u}^2 \tag{5.6}$$

The pressure loss in e.g. y direction will then be:

$$\Delta P = \Delta y(\mu D \mathbf{u} + \frac{1}{2}\rho F \mathbf{u}^2) = \mu d \mathbf{u} + \frac{1}{2}\rho f \mathbf{u}^2 \tag{5.7}$$

The $D$ and $F$ coefficients can be calculated if the velocity-dependent pressure loss is known, as follows:

$$D = \frac{d}{\mu} \tag{5.8}$$

$$F = \frac{2f}{\rho} \tag{5.9}$$

In this work for the pressure loss evaluation on the filling tank, empirical correlations available in hydraulics handbooks, e.g. in [6] were used. As for the mass flow rate meter, the information provided by the vendor was used.

For the pipe flow, the transition from laminar to the turbulent regime is well studied and $Re$ number values are tabulated. For loops, however, this is not the case. Some results report $Re$ number values as low as 340 for a flow transition from laminar to turbulent in an NCL [7]. The transition criteria still being unclear, it was deemed necessary to test several turbulence modelling approaches in this work to establish one that agrees with the experiment the most. All the methods are based on the $RANS$ approach, wherein the Reynolds stress equation is split into a steady-state component and a fluctuating part. This part is represented through Reynolds stress tensor and to relate the stress tensor to the steady-state mean flow values, closure models are required, known as turbulence models. The parameter that relates the Reynolds stress to the mean flow is the eddy viscosity which represents internal momentum transfer via eddies that are formed in a turbulent flow. Several eddy-viscosity turbulence models are depending on the number of closure transport equations they solve. In this work the following three $RANS$ models are considered:

- $k-\omega$ Shear Stress Transport ($k-\omega$ *SST*) is a two-equation model solving for the turbulent kinetic energy $k$ and the specific turbulence dissipation rate $\omega$.

$$k = \frac{3}{2}\mathbf{u}I^2 \tag{5.10}$$

$$\omega = \frac{\sqrt{k}}{C_\mu l} \tag{5.11}$$

  - Uses k-$\omega$ approach in the boundary layer, thus allowing to apply it to low Re flow without dumping functions.
  - Switches to k-$\epsilon$ model in free stream region
  - Better performance for adverse pressure gradient and
  - Suggested for low Re number flows.

- Langtry-Menter $k-\omega$ Shear Stress Trasnport turbulence model ($k-\omega$ *SST LM*) This four-equation model is also known as $\gamma Re_\theta$ model.
  - It aims specifically to capture flow transition from the laminar to the turbulent regime [8] and is particularly interesting to investigate for the natural circulation regime with internal heat generation in the fluid.
  - This model performs well at low *Re* numbers and is also successful at predicting transition scenarios [9].

  The model introduces the transition *Re* number as:

$$Re_\theta = \begin{cases} 1173.51 - 589.428Tu + \frac{0.2196}{Tu^2} & Tu \leq 1.3 \\ \frac{331.5}{(Tu-0.5658)^{0.671}} & Tu > 1.3 \end{cases} \tag{5.12}$$

  where the *Tu* factor depends on the free stream velocity and the turbulent kinetic energy and is defined as :

$$Tu = \frac{\sqrt{(2k/3)}}{|\mathbf{u}_\infty|} \tag{5.13}$$

  For this work, based on the experimental data and the main geometrical parameters of the loop, the transition *Re* number is calculated to be 584, further showcasing that in the case of natural circulation with distributed heating the transition *Re* number can be much lower than e.g. for an infinitely long straight pipe, for which the transition *Re* is around 2300.

- realizable $k-\epsilon$ turbulence model. These two-equation model solving for the turbulence kinetic energy $k$ and the energy dissipation rate $\epsilon$. It differs from the standard $k-\epsilon$ model by adopting a new, more exact transport equation for the dissipation rate and by changing the turbulent viscosity from being constant to being a function of the mean flow properties. As a result, the *realizable* $k-\epsilon$ model is more accurate and reliable for many applications than the standard $k-\epsilon$ model. It is especially success full in describing complex flows with rotation, vortexes, and stagnation zones.

The experimental data used in this work for validation of the CFD model had 450 *Watt* total power provided to the loop and distributed 2/3 to the hot leg and 1/3 to the cold leg. In absence of more complicated heat loss models, the heat losses are taken into account in the CFD simulations by decreasing the net power provided to the legs. The cooler is modelled by providing the convective heat transfer coefficient calculated based on semi-empirical correlations [10].

The solver accounts for the conjugate heat transfer between solid and fluid regions as well as the buoyancy and the turbulence effects adopting a segregated solution strategy; first, the equations for the fluid region are solved followed by the solution for a solid region. For the fluid region, the PIMPLE algorithm with three correctors is adopted for pressure,

velocity, and energy equations, meaning that the pressure is corrected three times within the PIMPLE loop. The outer correctors are two, meaning that within a time step the PIMPLE loop is performed two times for the whole set of the equations before moving to the next time step. As for the numerical schemes, the first-order implicit Euler scheme is used for the time derivative terms, second-order unbounded Gauss linear and first-order bounded Gauss upwind schemes are used for the divergence terms.
A special temperature coupling boundary condition available in the OpenFOAM, *TemperatureCoupledBaffleMixed* [3], is used for the temperature on all contact zones between the fluid and the solid. This boundary condition represents the coupling condition for the temperature given by Eq.6. On the tank outlet an *inlet-outlet* boundary condition is used. This boundary condition usually behaves as a zero gradient Neumann boundary condition, except when there is backflow into the domain; then the *inletOutlet* boundary condition changes to a fixed value to prevent a non-physical flow re-entry situation. The boundary condition for the velocity in the liquid zone is set to uniform zero fixed value on all boundaries and to *pressureInletOutletVelocity* on the tank outlet boundary; this boundary condition applies a zero gradient condition on the outflow and for the inflow, a velocity derived from internal cell values is applied. For the cooler external surface, a coefficient boundary condition mode is used: the overall heat transfer coefficient and the ambient temperature are specified.
Both the fluid and the solid zones are initially set to $333.15K$ temperature, and the fluid velocity is set to 0.

## 5.3 Results and discussion

We compare the CFD simulation results to the experimental data, as well as two additional model results:

- An analytical method wherein the stability of the NCL is represented in two-dimensional space of two parameters, usually dimensionless numbers (e.g. Pr, Re). More about this method applied specifically to the DYNASTY loop can be found in [11], [12].
- 1D object-oriented modelling in Modelica carried out at PoliMi and described in [11].

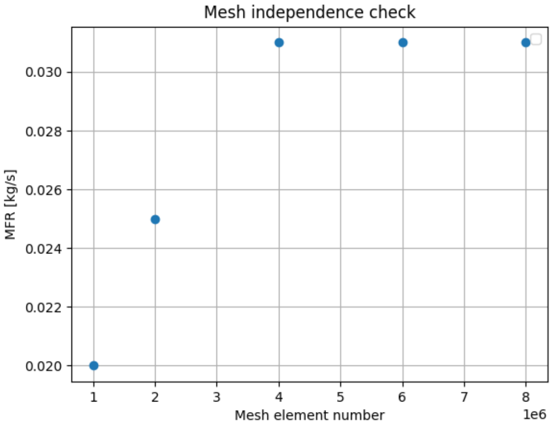


Figure 5.4: Mesh Independence check

The main aim of the comparison is to establish whether the CFD simulations agree with the experiment and the two above-mentioned methods in predicting the same state of stability for the NCL with the input parameters described in the previous section. To achieve this, two crucial output parameters are being compared:

- The mass flow rate dynamic behavior of the water,
- The temperature difference across the cooler defines the driving water temperature.
- The adiabatic mixing temperature in the loop.
- Additionally, the pressure losses are compared.

First, in Figure 5.4, the mesh independence check is shown based on the mesh elements number vs the average mass flow rate in the loop. Based on this, the fine mesh model with $6 \times 10^6$ elements is chosen for the simulations.

In Figure 5.5 the mass flow rate dynamic behavior is shown for all three turbulence models considered compared to the experiment and the object-oriented modelling results. In Figure 5.6 a close-up view of the last 1000 s of the simulation is shown to better visualize the degree of agreement between the simulation and the experiment. The CFD results are compared to the experimental data and Modelica results. Both CFD and Modelica simulations reflect the initial oscillation of the mass flow rate due to the typical initial transient of the natural circulation where hot and cold fluid plug start circulating in the circuit.

As for the turbulence modelling comparison, the realizable $k-\epsilon$ turbulence model overestimates the peak mass flow rate value by almost two times both at the initial largest peak around 500 s and for the second one occurring around 900 s. On the other hand, the two $k$–$\omega$ SST models seem to better reproduce these initial transients also in terms of characteristic frequency. However, it should be pointed out that the exact values of the maximum and minimum peak strongly depend on the initial conditions of the system, especially in terms of the turbulence parameters. In this light, the more appropriate figures of merit for the validation of the CFD model are the steady-state values.

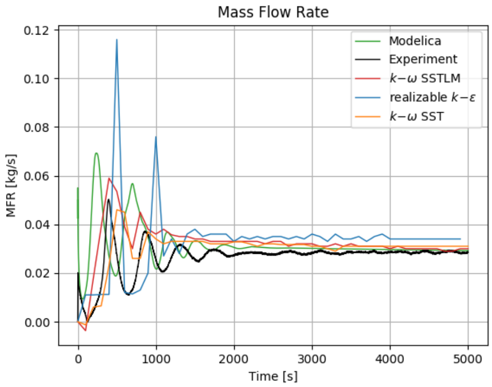


Figure 5.5: Mass flow rate

The mass flow rate stabilizes starting from t=2000 s, with the realizable $k-\epsilon$ turbulence model further exhibiting some small oscillations and finally stabilizing after 4000 s with the average stable mass flow rate being 0.034 kg/s. This is ≈ 16 % higher than the experimental average of 0.029 kg/s. The $k-\omega$ models are in better agreement with the experimental results, overestimating the mass flow rate less than 10 %. The relative error calculation is done based on the results from 2000 s to 5000 s interval, where the mass flow rate does not exhibit large oscillations (see Table 5.3 for a result overview). The evaluation of the stabilized mass flow rate is strongly dependent on the estimation of the temperature and the pressure drop.

The relative difference between the models and the experiment are reported in Table 5.3.

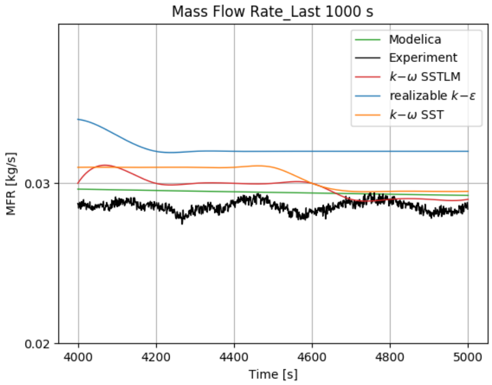


Figure 5.6: Mass flow rate for the last 1000 s of the simulation

| Case | Mass flow rate (kg/s) | Relative error |
|---|---|---|
| Experiment | 0.029 | - |
| CFD-$k-\omega$ SST | 0.0311 | 7.2% |
| CFD-$k-\omega$ SSTLM | 0.0310 | 6.9% |
| CFD-realizable $k-\epsilon$ | 0.0337 | 16.2% |
| Modelica | 0.030 | 1% |

Table 5.3: Average stabilized mass flow rate compared to the experiment

In Figure 5.7 the temperature difference across the cooler is shown for the CFD models, the experiment, and the 1D object-oriented model. Again, the overall behavior is the same for all the models, indicating large temperature differences across the cooler at the beginning of the transient when the hot and cold fluids start flowing in the loop. The oscillations decrease in amplitude after 2000 $s$.

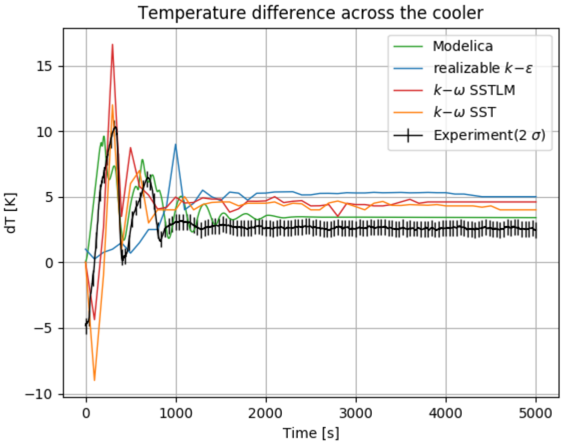


Figure 5.7: Temperature difference across the cooler

Figure 5.8 shows that 1D object-oriented model results are in closer agreement with the experiments than the CFD model results which is attributed to the heat loss modelling. As mentioned previously, the CFD model only takes the heat losses into account by decreasing the net power supply. The 1D object-oriented model, however, has a more sophisticated heat loss model implemented based on the pipe material properties and flow conditions. Table 5.4 summarizes relative differences between the models.

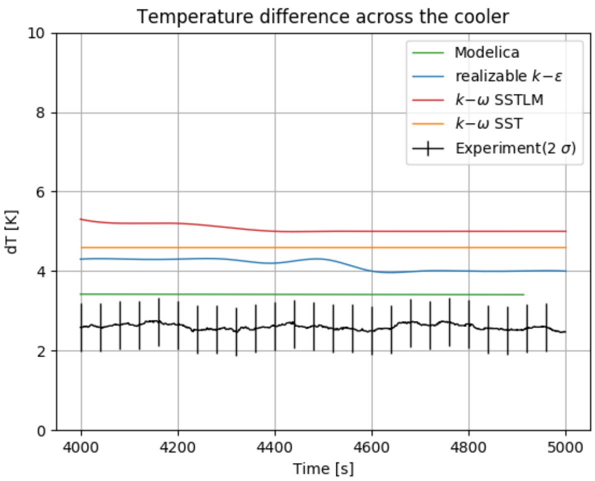


Figure 5.8: Temperature difference across the cooler for the last 1000 s

| Case | Average $\Delta T$ (K) | Difference (K) |
|---|---|---|
| Experiment | 2.7±0.7 | - |
| CFD-$k-\omega$ SST | 4.5 | 1.8 |
| CFD-$k-\omega$ SSTLM | 5 | 2.3 |
| CFD-realizable $k-\epsilon$ | 4.3 | 1.6 |
| Modelica | 3.5 | 0.8 |

Table 5.4: The average stabilized temperature difference

In Figure 5.9, the adiabatic mixing temperature in the loop is shown.

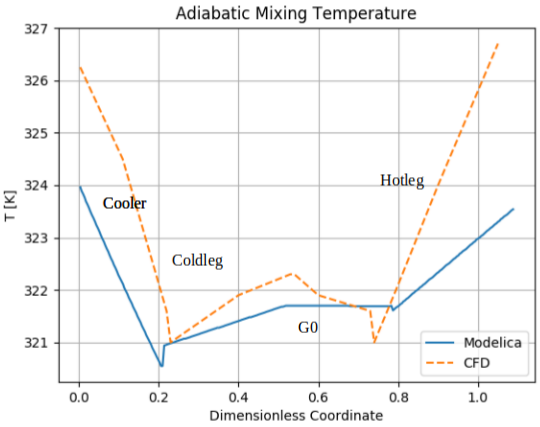


Figure 5.9: Adiabatic mixing temperature for stabilized flow

The CFD model agrees with the experiment in establishing energy balance, however, it predicts around $2K$ higher temperatures in the hot and the cold legs which can be contributed to the initial conditions, the heat loss modelling, and the turbulence models used.

The Re and Pr numbers calculated from CFD results for the stabilized flow are 1145 and 345 respectively, and these results place the considered transient in the stable region on the stability map shown in Figure 5.10 which further shows the validity of the CFD model.

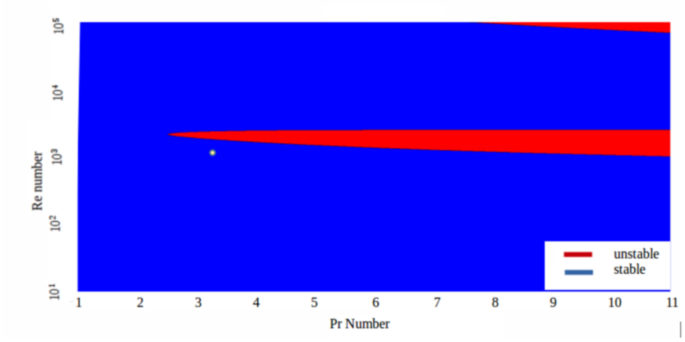


Figure 5.10: Stability Map

Finally, in Figure 5.11 the pressure loss in the loop due to the tank and the mass flow rate meter is shown for the CFD model and the 1D object-oriented model. The same trend of the $k - \omega SST$ model exhibiting the best agreement with the experiment can be noticed.

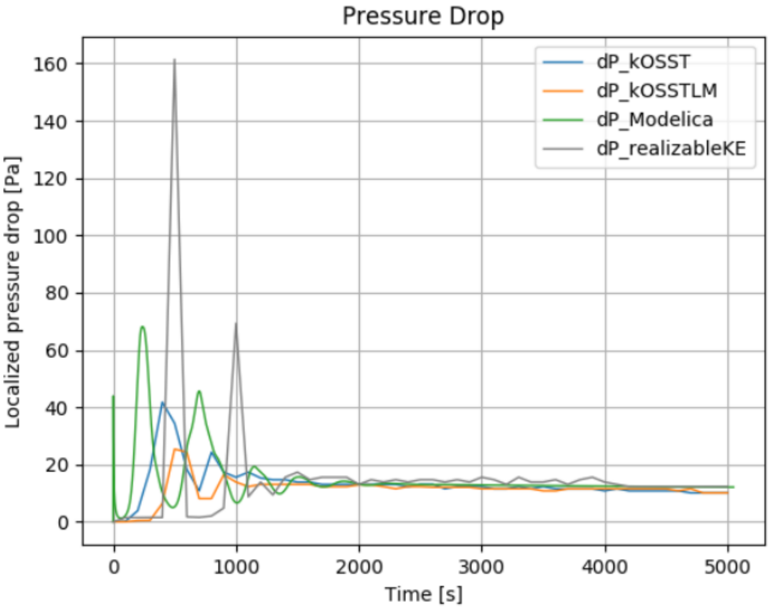


Figure 5.11: Pressure drop in the loop

## 5.4 Conclusion

In this Chapter, the main outcomes of a CFD model validation for a natural circulation loop with internal heat generation were presented. Understanding the behavior of such a natural circulation loop under various operational conditions is of direct importance for the Molten Salt Reactor Systems, as they seek to implement natural circulation in passive safety features, e.g. as a decay heat removal system. The main questions of interest are:

- Does the CFD model predict the same equilibrium stability state as the experiment (qualitative validation)?
- How good is the numerical agreement between the CFD model and the experiment concerning the main flow parameters (quantitative validation)?
- Which turbulence model can be recommended in modelling natural circulation flow in a loop?

It was established, that the CFD model shows good agreement with both the experiment and the two additional model results in predicting a stable equilibrium state. The relative differences in terms of mass flow rate between the CFD model and the experiment are below 17 %, however, there is a large difference between different turbulence models: the realizable $k-\epsilon$ model which predicts 17 % higher mass flow rate is probably not suited for natural circulation modelling. The other two turbulence models, the $k-\omega SST$ and the $k-\omega SSTLM$, both have only a 6.87 % difference in terms of the mass flow rate compared to the experiment. The same trend is true when analyzing the temperature difference across the cooler. Here, however, the CFD model shows slightly higher discrepancies with the experiment, most likely to be attributed to the heat loss modelling and the particular effects of the turbulence models. In general, a final recommendation for better performance of the CFD model would be:

- Implement a more sophisticated heat loss evaluation method for the CFD model.
- Further investigate the $k-\omega SST$ and $k-\omega SSTLM$ models; currently, the former one seems to perform better when it comes to the prediction of the temperature difference. However, taking into account that $k-\omega SSTLM$ is specifically developed to predict critical Re numbers and flow transition, with better heat loss modelling this turbulence model could perform better.

# Bibliography

[1] S. Hee Kang, "Performance analysis of the passive safety features of iPOWER under Fukushima-like accident conditions.", Nuclear Engineering and Technology Volume 51, Issue 3, June 2019, Pages 676-682

[2] A. Cammi, L.Luzzi, M. Cauzzi, A.Pini, "DYNASTY: An Experimental Loop for the Study of Natural Circulation with Internally Heated Fluids", 12th International Conference on Heat Transfer, Fluid Mechanics and Thermodynamics (HEFAT 2016), Malaga, Spain

[3] OpenFOAM API Guide v2006,
*https://www.openfoam.com/documentation/guides/latest/doc/index.html*

[4] M. el Abbassia, D.J.P. Lahaye, C. Vuik, "modelling turbulent combustion coupled with conjugate heat transfer in OpenFOAM", Naples, Italy, September 17-21, 2017

[5] H.E.Hafsteinsson, "Porous Media in OpenFOAM, Proceedings of CFD with Open-Source Software", 2008, Edited by Nilsson H., *http://dx.doi.org/10.17196/OS CFDYEAR 2008*

[6] C. Hirsch, "Numerical Computation of Internal and External Flows", 2nd Edition, 2007

[7] Halligan, K.P., Viskanta, R., 1986. "Heat transfer from a rod bundle under natural circulation conditions.", Division of Accidental Evaluation, Office of NuclearRegulatory Research, U.S. Nuclear Regulatory Commission.

[8] R.B. Langtry and F.R. Menter, "Correlation-Based Transition modelling for Unstructured Parallelized Computational Fluid Dynamics Codes.", AIAA Journal, 47(12):2894–2906, 2009.

[9] P. Ranjan, W. J. Warton, K. A. James, "A Comparison of Physics- and Correlation-Based Turbulence Models at Low Reynolds Numbers", Journal of Aircraft,2020, Vol. 57:1, pp. 1–2

[10] T. Bergman, A. Lavine, F. Incropera, D. DeWitt, "Fundamentals of Heat and Mass Transfer.",John Wiley  Sons, Inc., Hoboken, NJ, United States

[11] A. Pini,A. Cammi, L. Luzzi,"Analytical and numerical investigation of the heat exchange effect on the dynamic behavior of natural circulation with internally heated fluids", Chemical Engineering Science, Vol. 145, 2016, pp. 108-125

[12] D. Ruiz,A. Cammi, L. Luzzi, "Dynamic stability of natural circulation loops for single-phase fluids with internal heat generation.", Chemical Engineering Science 126, 573-583

# Chapter 6

# Conclusion and Outlook

Molten Salt Reactors are a highly promising design concept for the next generation of nuclear power plants due to their increased economical competitiveness, good breed and burn capabilities, intrinsic safety features related to the fuel being liquid, and reduced proliferation risks amongst several other attractive features.

However, on the pathway of a successful design and licensing of the MSRs, there are several hurdles to be handled. Some of the focal points are related to the better understanding of the molten salt chemistry and the interaction of the fuel carried with the reactor structural materials whereas another major *R&D* field for the MSRs is the development of high fidelity modelling and simulation tools that are capable to reflect the specific features of the liquid fuel. As it was discussed in this work, the fuel being liquid imposes a new level of coupling between the fuel thermal-hydraulics and neutron-kinetics. The impact of the fuel being liquid manifests itself, inter alia, through the reduction of the effective fraction of the delayed neutrons. Additionally, fuel recirculation and stagnation zones can also have an undesired impact on the neutron-kinetics. It is thus important to use specifically developed tools and methods to capture the correct coupling between the thermal-hydraulics and neutron-kinetics in the MSRs. In this work, three coupling approaches were implemented and tested on the available simulations and experimental data for the MSRE and the MSFR:

- A simplified neutron-kinetics solver based on the one-speed diffusion approximation was implemented in OpenFOAM and coupled to an OpenFOAM CFD solver for a single-phase incompressible flow with Boussinesq approximation. To test the solvers, a code-to-code comparison with conventional software is performed. Conventional software for neutron transport MCNP-6.1 and ANSYS-CFX 19.1 for thermal-hydraulics were used for this purpose. As a test case, the MSFR steady-state operation scenario was chosen. Even though the neutronics solver is simple and lightweight, it predicted a power distribution very close to the one computed by a precise Monte-Carlo approach in MCNP with an average difference of 6%.

  As for the coupling approaches, MCNP and ANSYS CFX exchanged relevant coupling terms after each of the software finished its respective run. This is a more traditional approach to coupling which could, in general, lead to a decreased accuracy and coupling non-linearities. The development of a customized solver such as the one-speed diffusion neutronics solver in this work allows implementing better coupling controls, with the coupling terms being exchanged multiple times during a time-step and ensuring convergence of individual physics modules before moving to the next time-step.

However, the development of such customized neutronics solvers usually involves a higher degree of approximations being used and it also implies substantial code development and debugging efforts as well as subsequent verification and validation efforts.

- In another approach, the Serpent neutron transport software was coupled externally to OpenFOAM using a Python interface for exchanging the coupling information and swapping the necessary files. The transport of delayed neutron precursors was taken into account by introducing fuel velocity as an external parameter and modifying the delayed neutron precursor source generated initially for a static fuel according to the fuel velocity and the time-step. The technique was tested on a single channel model of the Molten Salt Reactor Experiment, both in steady-state and transient regimes and yielded sufficiently good results. For the steady-state regime, the fuel and the graphite moderator temperature distributions were compared to the model calculation performed at ORNL. For the transient scenario, the reactor power response to a step reactivity insertion was analyzed and the results were compared to the experimental data available in the ORNL reports. Despite the rather simple treatment of the delayed neutron precursor transport, the coupled approach was able to reproduce the main trends of the power response curve.
  The main advantages of this technique are the possibility to use a very high accuracy neutron transport modelling and the minimal modification requirements in the respective standalone software. The drawback is the computational time required by the Serpent Monte Carlo code in transient simulations. As shown in Chapter 3, the time required to simulate $1s$ of power evolution after a reactivity insertion transient in Serpent, is around 20 hours, as compared to a tenth of a minute required by CFD for the same $1s$ interval. Additionally, this approach could suffer in accuracy when compared to a performance of more tightly integrated software suits for MSR modelling, especially for the modelling of reactor fast transients.

- Finally, a third approach was developed, featuring again Serpent for neutronics calculations and an OpenFOAM CFD solver for thermal-hydraulics, but with a dedicated DNP transport solver implemented in OpenFOAM. The main difference compared to the previous technique is the possibility to realize the DNP transport with high accuracy, taking into account realistic 3D velocity field distributions, and easily communicate this information to the neutronics module. The method is tested by comparison to a molten salt fast reactor benchmark results. The agreement for the steady-state scenario is very good, the discrepancies are in order of a few %, and reach ≈ 10-15 % only when probing the second-order effects, like the relative change between the fission rate densities. The effect of the DNP movement on the reactivity loss is captured very precisely, with only a 3.4% difference compared to the average value obtained by the other participants, demonstrating that the coupling technique is implemented correctly and is capable to capture the relevant phenomena of the liquid fuel thermal-hydraulics and neutronics coupling. For the transient regime, the test case is limited to only one parameter set and it is thus difficult to make a unilateral statement on the accuracy. However, the bottleneck of the transient simulations is the very long time required to analyze as quite fine time-steps are required to catch the periodic behavior of the power as a function of the sinusoidal trend in the fuel salt cooling.

Additionally, this works included also a complete CFD assessment of the DYNASTY natural circulation loop designed to investigate the stability of internally heat generation liquids in natural circulation. This is relevant for the passive heat removal systems of MSRs. The simulations concluded that the transition between laminar and turbulent flow regimes in NCLs with internal heat generation can be correctly captured by the $k{-}\omega$ turbulence models. The $k{-}SST$ model showed the highest accuracy for the scenario simulated in this work. The $k-SSTLM$ model which is meant for transition flows performed well, but requires further tuning of the relevant parameters (e.g. the transition $Re$ number).
The methods described in this work can be further developed and optimized:

- The diffusion neutronics solver is very simple and can be optimized by including more energy-groups. Also, a more extensive verification including transient test cases is required.
- The Serpent-OpenFOAM coupling method can be optimized with a focus on reducing the computational time and implementing coupling controls to have better control on the coupling nonlinearities of such an approach where two (or more) standalone software/solvers are coupled.
- Some of the methods presented in this work have limited showcasing of transient scenarios. This could be improved to show more wide applicability of the methods for the transient modelling of MSRs.

Additionally, the coupled modelling of MSRs undertaken in the scope of this work focused on the modelling of liquid fuel neutron kinetics, thermal-hydraulics, and heat transfer. Some aspects of MSR modelling, e.g. thermal-mechanical modelling or fuel burnup and salt chemistry are not included in the scope of this work and present potential further development and extension opportunities for the coupling techniques described in this work.
This work provides interesting insight especially for those in the MSR community who would like to apply Monte-Carlo techniques for the neutronics analysis of the MSRs and to couple them with CFD tools.

# Appendix A

# One-group neutron diffusion solver in OpenFOAM

In this Appendix the OpenFoam header (.H) files for the one-group neutron diffusion equation and the delayed neutron precursor transport equation are included.

A. The neutronFluxEqn.H

```
    {
     G.storePrevIter();

    fvScalarMatrix neutronFluxEqn
    (
        fvm::ddt(v,G) ==
        fvm::laplacian(D, G)
      - fvm::Sp(Sigma_a,G)
      + nu & Sigma_f & G
      + DNPS

    );

    neutronFluxEqn.relax();

    fvOptions.constrain(neutronFluxEqn);

    neutronFluxEqn.solve();
    resPicard = GMax((mag(G-G.prevIter()))->internalField());
    Info << "Picard iteration :
    " << iterPicard+1 << " residual = " << resPicard << endl;

    radiation->correct();

    fvOptions.correct(G);

}
```

B. The neutronPrec<i>Eqn.H (there are either 8 or 6 .H files for each DNP group).

```
    {
     dnp1.storePrevIter();
    fvScalarMatrix neutronPrec1Eqn
    (
        fvm::ddt(dnp1)
      + fvm::div(phi, dnp1)
      - fvm::laplacian(d, dnp1)
      + fvm::Sp (lambda_1,dnp1)==
      beta_1*((eta&Sigma_f)&G)
);


    neutronPrec1Eqn.relax();

    fvOptions.constrain(neutronPrec1Eqn);

    neutronPrec1Eqn.solve();

resPicard = dnp1Max((mag(dnp1-dnp1.prevIter()))->internalField());
    Info << "Picard iteration :
    " << iterPicard+1 << " residual = " << resPicard << endl;

    radiation->correct();

    fvOptions.correct(dnp1);

}
```

The delayed neutron precursor data extracted from Serpent.

| Group | $\lambda_i$ [1/s] | Fraction |
|---|---|---|
| 1 | 1.29110E-02 | 2.21190E-04 |
| 2 | 3.47380E-02 | 5.84487E-04 |
| 3 | 1.19286E-01 | 4.39045E-04 |
| 4 | 2.86213E-01 | 9.04313E-04 |
| 5 | 7.88340E-01 | 2.68769E-04 |
| 6 | 2.44170E+00 | 1.00460E-04 |

Table A.1: Delayed neutron precursor data for the MSFR fuel composition used in this work

In the Table A.2 the average number of neutron emitted per fission and the energy release per fission extracted from Serpent are provided.

| Parameter | Value | Unit |
|---|---|---|
| $\nu$ | 2.53005 | - |
| $E_f$ | 2.00006E+02 | $MeV$ |

Table A.2: One-group neutronics constants

# Appendix B

# Coupling Serpent to Open-FOAM: Python routines

The Python script presented below is used to decode the binary file written by Serpent and containing point-wise delayed neutron precursor production information (source.precpoints).

```
import shlex
import logging
import subprocess
import struct
import csv
import os
import sys
file_in = open("inputfiledir")
file_out = open("outputfiledir","wb")
class decodeDNP:

    """ A class that decoded Serpent binary precpoints file"""
    def decode_binary_DNP_info(self, file_in,
    write_output_to_file = True):
        with open(file_in, ’rb’) as f:
            rawData = f.read(file_in)
        # each chunk is 72 bytes long
        chunks = [ rawData[x:x + 72]
        for x in range(0, len(rawData), 72)]
        # each chunk is comprised of 9 doubles
        ## DNP_info -> decoded data.
        DNP_info = [struct.unpack(’d’ * 9, c) for c in chunks]
        if write_output_to_file == True:
            with open(file_out + ’.txt’, ’w’) as csvF:
                writer = csv.writer(csvF)
                writer.writerows(DNP_info)
        return DNP_info
```

The following script shows how the DNPs are shifted by the mean velocity field of the fuel.

```
#Author J. Groth-Jensen
  def shift_DNP_position(self) -> None:
        """
Shifts the DNP position based on velocity information retrieved from
        the last OpenFOAM run
        :return: None
        """
        self.log.info("Shifting the DNP positions")
        # Retrives the mean of the velocity field along the
        z-axis
        Uzmean = self.calculate_Umean_in_fuel_regions()
        # loops over the list of DNPs and shift the position
        in the z-axis direction.
        for i, dnp in enumerate(self._DNP_field):
            # Checks if the DNP is in the fuel (just based on
            the z-coordinate ) and
            shifts the DNP position
            if self.is_DNP_in_fuel(dnp) == True:
                self._DNP_field[i, 2] = dnp[2] + Uzmean * 100 *
                self.runconfig.deltaT_CFD
                if self._DNP_field[i, 2] > 160 : self._DNP_field[i, 2]
                = self._DNP_field[i, 2] - 160
```

# Appendix C

# Passive scalar transport solver for the DNP transport in OpenFOAM

In this Appendix the OpenFOAM header file used to provide the delayed neutron precursor transport equation is given together with the solver .C file.
The solver files can also be obtained here:
https://github.com/AshNlb/Passive-Scalar-Transport-Solver-for-DNPs-in-OpenFOAM.git.
The transport equation solved for each DNP group is:

$$\frac{\partial C_i}{\partial t} = -\nabla(\mathbf{u}c_i) + \nabla \cdot D\nabla C_i - \lambda_i C_i + \beta_i \nu \Sigma_f \Phi \tag{C.1}$$

```
fvScalarMatrix neutronPreciEqn
(
    fvm::ddt(rhok,dnp1)
  + fvm::div(phi, dnp1)
  - rhok*fvm::laplacian(DT, dnp1)
  + fvm::Sp(lambda1,dnp1)
 == rhok*dnp1_s
);
```

Where $dnpi_s$ is the DNP production term written per mesh cell, so it has the same format as the dnp1 field itself. The production term is:

$$dnpi_s = \beta_i \Phi \times \nu \times \Sigma_f \tag{C.2}$$

where $\beta_i$ is the DNP fraction, $\Phi$ is the scalar neutron flux, $\nu$ is the number of neutrons emitted per fission, $\Sigma_f$ is the fission cross section.
On the other hand, we have the specific power written on the same mesh in Serpent:

$$\frac{P}{V} = \Phi \times \Sigma_f \times E_r \tag{C.3}$$

Where $P/V$ is the specific power per unit volume, $V$ is the volume, $E_r$ is the heat produced per fission in Joules.
By expressing $\Phi$ in terms of specific power from the previous equation and using it to

calculate the DNP production term, we get:

$$dnpi_s = \frac{\beta_i \times \nu \times P}{E_r V} \tag{C.4}$$

The DNPTransportinOF.C file is included below.

```
int main(int argc, char *argv[])
{
    #include "postProcess.H"
    #include "setRootCaseLists.H"
    #include "createTime.H"
    #include "createMesh.H"
    #include "createControl.H"
    #include "createFields.H"
    #include "createTimeControls.H"
    #include "CourantNo.H"
    #include "setInitialDeltaT.H"
    #include "initContinuityErrs.H"
    turbulence->validate();
    Info<< "\nStarting time loop\n" << endl;
    while (runTime.run())
    {
        #include "readTimeControls.H"
        #include "CourantNo.H"
        #include "setDeltaT.H"
        runTime++;
        Info<< "Time = " << runTime.timeName() << nl << endl;
        // --- Pressure-velocity PIMPLE corrector loop
        while (pimple.loop())
        {
            #include "TEqn.H"
            // --- Pressure corrector loop
            while (pimple.correct())
            {
            }
            if (pimple.turbCorr())
            {
                laminarTransport.correct();
                turbulence->correct();
            }
        }
        #include "neutronPrec1Eqn.H"
        #include "neutronPrec2Eqn.H"
        #include "neutronPrec3Eqn.H"
        #include "neutronPrec4Eqn.H"
        #include "neutronPrec5Eqn.H"
        #include "neutronPrec6Eqn.H"
        #include "neutronPrec7Eqn.H"
        #include "neutronPrec8Eqn.H"
        runTime.write();
        Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
            << "  ClockTime = " << runTime.elapsedClockTime() << " s"
            << nl << endl;
    }
```

# Part II

# Publications

# Coupling Techniques for Multiphysics Modeling of Molten Salt Reactors

A. Nalbandyan[1], E. B. Klinkby[1], and B. Lauritzen[1]

[1] *Center for Nuclear Technologies, Technical University of Denmark, Roskilde, Denmark*

**Abstract**

Molten Salt Reactors (MSRs) are receiving increased attention arising from their potential advantages compared to conventional reactors; these include inherent safety features and reduced costs. The circulating fuel in an MSR induces a strong coupling between neutronics and thermal hydraulics, in part due to the delayed neutron field being affected by the fuel velocity fields, thus requiring new modelling approaches. In this paper a conventional operator-splitting and a multiphysics coupling technique between neutronics and thermal hydraulics, applied to a simple test geometry, are compared. Commercial software, ANSYS CFX for thermal hydraulics and MCNP for neutronics, are applied in the conventional operator-splitting technique, whereas multiphysics coupling is investigated by means of open source software OpenFOAM. In particular, fuel temperature, velocity and neutron fluxes obtained by the two approaches are presented and compared.

## 1 Introduction

Modelling nuclear fission reactors requires coupling different physics phenomena; neutron kinetics affects coolant flow and is in turn affected by the thermal-hydraulics properties of the flow. Although these multiphysics aspects have been previously applied to the reactor analysis of conventional nuclear power plants [1], the need to develop enhanced coupling methods has increased with the increased interest in MSRs [2], where the fuel itself is liquid (during operation). The fuel circulation in such systems leads to the decrease in the effective fraction of delayed neutrons contributing to the fission process and since delayed neutrons are crucial for reactor control, flow circulation must be modelled correctly. Moreover, the flow pattern can cause local overheated fuel regions which might affect the reactor performance. In turn, the fuel power distribution determines the flow temperature fields. This implies a tight coupling between the thermal-hydraulics and the neutronics.

Two different approaches to couple neutronics and thermal-hydraulics codes are discussed in this paper. In the conventional operator-splitting (OS) technique, different software codes are used for the distinct physics and the coupling data is exchanged between the codes. The method is relatively easy to implement but usually only provides low-order accuracy [3].

A second approach is based on using a single software, adapting it to deal with both the neutronics and thermal hydraulics modules. This allows for improving the numerical accuracy by applying a dedicated time discretization and convergence control methods for the non-linear equations. However, these methods are in general more difficult to implement than the black-box coupling scheme.

In this paper the two different coupling techniques are applied to a simple geometry representing circulating molten salt flow. The black-box coupling approach is applied with the ANSYS CFX [4] thermal hydraulics and Monte Carlo N-Particle [5] neutronics codes. In the multiphysics coupling approach, the open source software OpenFOAM [6] is used with the same geometry to solve the deterministic neutron transport equation and the incompressible Navier- Stokes equations for the fuel flow. The method is implemented based on techniques described in [6], [7].

The fuel temperature and velocity field and power distributions are compared in the two modelling approaches. It is shown, that both techniques yield similar results for reactor steady state behaviour, with the direct coupling approach being more flexible in providing control over numerical coupling and integration schemes.

# 2 Coupling schemes

## 2.1 Conventional Operator-Splitting technique

The OS technique is a widely used coupling method in which the multiphysics equations are decomposed and subsequently the different physics are solved individually, typically using dedicated software for each physics module. Most of the known OS techniques applied in reactor analysis are only first-order accurate in time due to the specific linearization of the non-linear coupling terms in the equations [3]. When applying this method to stiff problems (i.e., with rapidly varying terms present in some solution modes) [17], numerical stability and convergence may be improved by choosing a very small time step, however, at increased computational cost.

As is common practice in reactor analysis, we present a method to couple two such monophysics codes; ANSYS-CFX (thermal hydraulics) and MCNP (neutronics). In Figure 1 the coupling scheme is presented; temperature and density of the fuel are extracted at the end of thermal hydraulics calculation and used to update the neutronics input. The power deposition in the system calculated by MCNP is then used to update the heat source for the next thermal hydraulics calculation. Coupling terms are updated via the User FORTRAN Interface of ANSYS CFX [16].

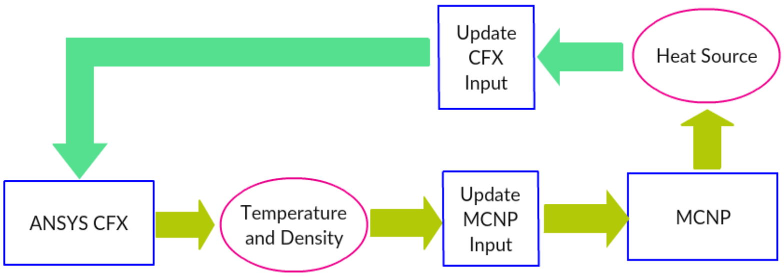


Figure 1: ANSYS CFX-MCNP Coupling Mechanism

The ANSYS CFX solves the unsteady Navier-Stokes equations in their conservation form: continuity, momentum and energy equations are solved simultaneously together with two state equations for density and enthalpy used to close the system. As an example, the equations for incompressible flow are:

$$\nabla \mathbf{u} = 0 \tag{1}$$

$$\rho \frac{\partial u}{\partial t} + \nabla(\mathbf{u} \times \mathbf{u}) + \nabla p - \nabla \tau - S_M = 0 \tag{2}$$

$$\rho \frac{\partial h}{\partial t} - \frac{\partial p}{\partial t} + \rho \nabla(\mathbf{u} h) - \nabla(\lambda \nabla T) - \nabla(\mathbf{u} \tau) = 0 \tag{3}$$

$$dh = c_p dT + \frac{dp}{\rho} \tag{4}$$

$$c_p = c_p(T) \tag{5}$$

The equations (1)-(3) represent mass, momentum and energy conservation and the equations (4)-(5) are the state equations for incompressible flow. In the equations (1)-(6) $\rho$ is the fluid density, $\mathbf{u}$ is the flow velocity vector, $h$ is the total enthalpy, $\tau$ is the stress tensor, $p$ is the pressure and $T$ is the temperature. ANSYS CFX uses a coupled solver, meaning that fully implicit methods are applied to solve the equations as a single system. The solver performs outer (time) iterations and inner (linearization) iterations. The solver applies a false time step for steady-state solutions, which acts as a guiding parameter helping the solution to converge faster to steady-state conditions [16]. The set of algebraic equations obtained after linearization of the differential equations is solved using the Algebraic Multigrid method, whereby the convergence process is accelerated by carrying out the iterations on a progressively coarser virtual mesh and then transferring the results back to a finer mesh [4].

The MCNP software solves the integral form of the neutron transport equation (the Boltzmann equation) by Monte-Carlo methods. It simulates a large number of histories from which reactor kinetics output is generated. MCNP6 is applied in the present work, with the ENDF-BVII.1 libraries for neutron interaction cross-sections [7].

## 2.2 Multiphysics Coupling

In a different approach, multiphysics software coupling allows for implementation and solution of the coupled partial differential equations using a single software code. In the present work the OpenFOAM software based on conventional finite volume method is used to implement and to solve both the neutronics and the thermal hydraulics equations. The software is proven to be suitable for Multiphysics modelling of molten salt reactors [8]. Along with the thermal-hydraulics and energy balance equations providing the fuel velocity and temperature fields, the one group neutron diffusion equation and the delayed neutron precursor concentration equation are embedded in OpenFOAM, to obtain the neutronics picture of the core. The OpenFOAM neutronics solver is thus based on a deterministic approach. The neutronics equations are:

$$\frac{1}{v} \frac{\partial \Phi}{\partial t} = D \Delta \Phi - \Sigma_a \Phi + (1 - \beta_{tot}) v \Sigma_f \Phi + \sum_{i=1}^{n} \lambda_i l_i \tag{6}$$

$$\frac{\partial l_i}{\partial t} = -\lambda_i l_i - \nabla(u l_i) + \Delta \frac{\nu_t}{Sc_T} l_i + \beta_i v \Sigma_f \Phi \tag{7}$$

Here $\Phi$ is the scalar neutron flux, $\nu$ is the average neutron velocity, D is the diffusion coefficient, $\Sigma_a$ and $\Sigma_f$ are the neutron absorption and fission cross-sections, $\beta_i$ is the delayed neutron fraction for $i_{th}$ delayed neutron group, $\beta_{tot}$ is the total delayed neutron fraction, and $\lambda_i$ and $l_i$ are the decay constant and the concentration of the $i_{th}$ delayed neutron precursor group.
A six group precursor representation is adopted, with delayed neutron fractions and decay constants being extracted from the ENDF-BVII.1 evaluated nuclear data library [8].

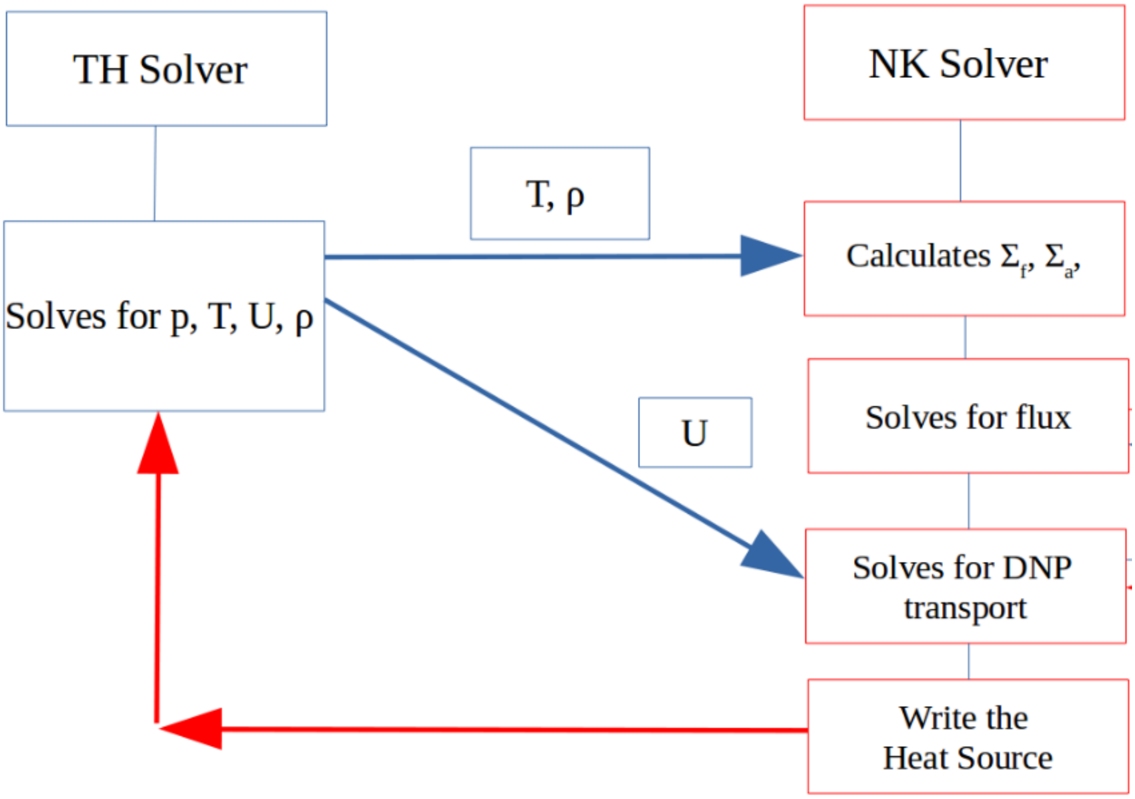


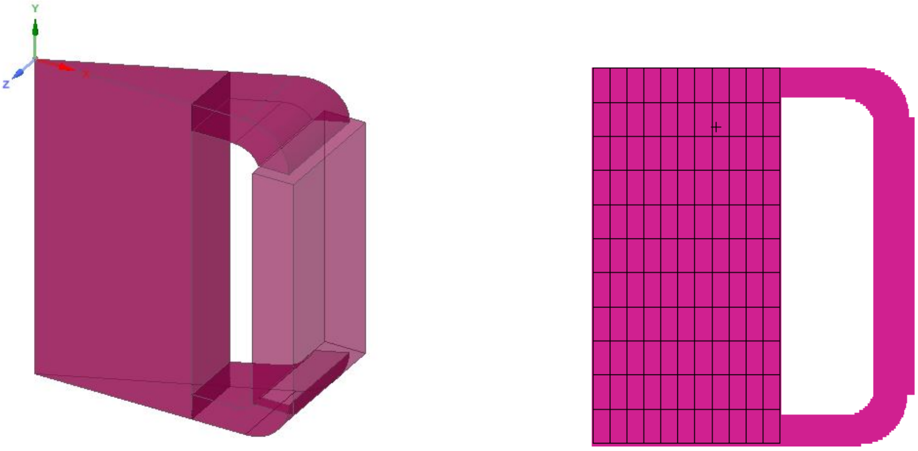
Figure 2: OpenFOAM coupling scheme; steps presented are per each timestep ($n$ timesteps in total)

.

The block coupling strategy is adopted based on works described in [9] and [10]. The general outline of the scheme is presented in Figure 2. The coupling terms are updated and the equations are iteratively solved within each time step up to the convergence of each individual physics block. Thus, this coupling approach allows resolving non-linearities due to the coupling within a time step.

## 3 Model and setup

Simulation results available in literature consider two representations of MSR cores: a 2D simplified axis-symmetrical model with a cylindrical core and an optimized 3D toroidal core shape model [10]. For the purpose of this work the exact same geometry for both neutronics and thermal hydraulics is adopted; the modelled 3D salt loop represents 1/16th of a cylindrical core ($D = h = 2.25m$) connected to a heat exchanger (Figure 3). The total salt volume is  1.2 $m^3$, with an in-core to total salt volume ratio of approx. 0.5. The MCNP neutronics model adopts the same geometry. Albedo boundary conditions are applied to represent the reflectors which are not included in the model (axial as well as radial). The fission heat deposition is calculated in the core only, by applying a superimposed mesh over the geometry (Figure 3(b)), and is subsequently translated into a heat source for thermal hydraulics calculations.



(a) Geometry (ANSYS CFX)

(b) In-core superimposed mesh tally (MCNP)

Figure 3: Fuel salt model

As a fuel carrier salt $Li - F$ is chosen, the fertile fuel is $^{232}ThF_4$, with an initial inventory of 2.5 % fissile $^{233}UF_4$. The salt parameters at 923.5 K are provided in Table 1. Previous studies on MSR [11] are used to set the nominal operating conditions for the loop; the full core nominal power is 3 GWth and the total flow rate is 18932 kg/s.

| Quantity | Value at 973.5 [K] |
|---|---|
| Density[g $cm^{-3}$] | 4125 |
| Dynamic viscosity [Pa s] | 0.01 |
| Thermal conductivity [W $m^{-1}K^{-1}$] | 1.009 |
| Calorific capacity $C_p$ [J $kg^{-1}K^{-1}$] | 1596 |

Table 1: Fuel salt (Fuel salt Li-F (77.5 %)-ThF4-UF4 (22.5 %) parameters (values calculated based on the formulas provided in [15]).

A tetrahedral mesh technique is applied in ANSYS CFX to avoid unnceccesary geometry decomposition required for sweep (hexahedral) meshing. The tetrahedral mesh increases the total mesh cell number by approximately a factor of six, but is often a faster way to mesh complex geometries with curvatures [18]. To ensure that the solution is mesh converged, several meshes of different coarseness are applied. As can be inferred from Figure 4, the solution becomes mesh independent when the number of mesh cells is half a million or more. As a tracking variable the average core velocity is chosen. As soon as the variable does not change with further mesh refinement, the solution is considered to be mesh-converged.

The converged ANSYS CFX mesh is subsequently translated into an OpenFOAM supported format (Figure 5).

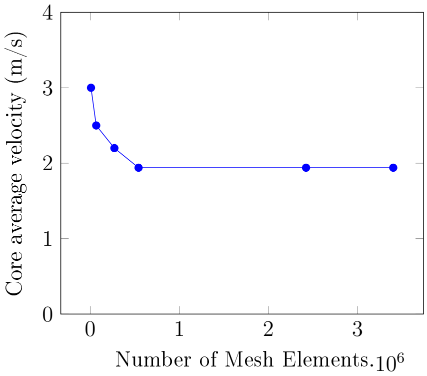
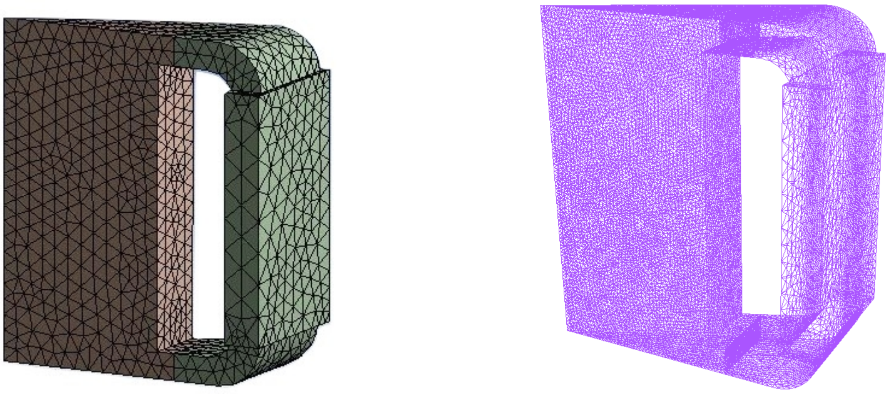


Figure 4: ANSYS CFX mesh-dependent core average velocities. The blue curve is to guide the eye.

Adiabatic wall boundary conditions are applied to the outer walls and symmetry boundary conditions on the symmetry planes. The heat exchanger is modelled as a simple volume with proper momentum loss and heat sink models. To resolve the turbulent flow, Reynolds Averaged Navier-Stokes (RANS) two-equation standard k-$\epsilon$ turbulence model is used [4] in ANSYS CFX. The choice of turbulence model is based on its availability for the majority of thermal hydraulics codes and its general popularity for solving standard engineering problems [17]. OpenFOAM calculations adopt realizable k-$\epsilon$ turbulence model; contrary to the standard k-$\epsilon$ model it is complaiant with certain mathematical constrainst on Reynolds stresses and is believed to describe recirculating flows better [6].

## 4 Results and analysis

In Figure 6 the velocity distribution fields of the fuel flow for fine mesh case obtained by two methods are depicted. The flow has the maximum velocity in the core central

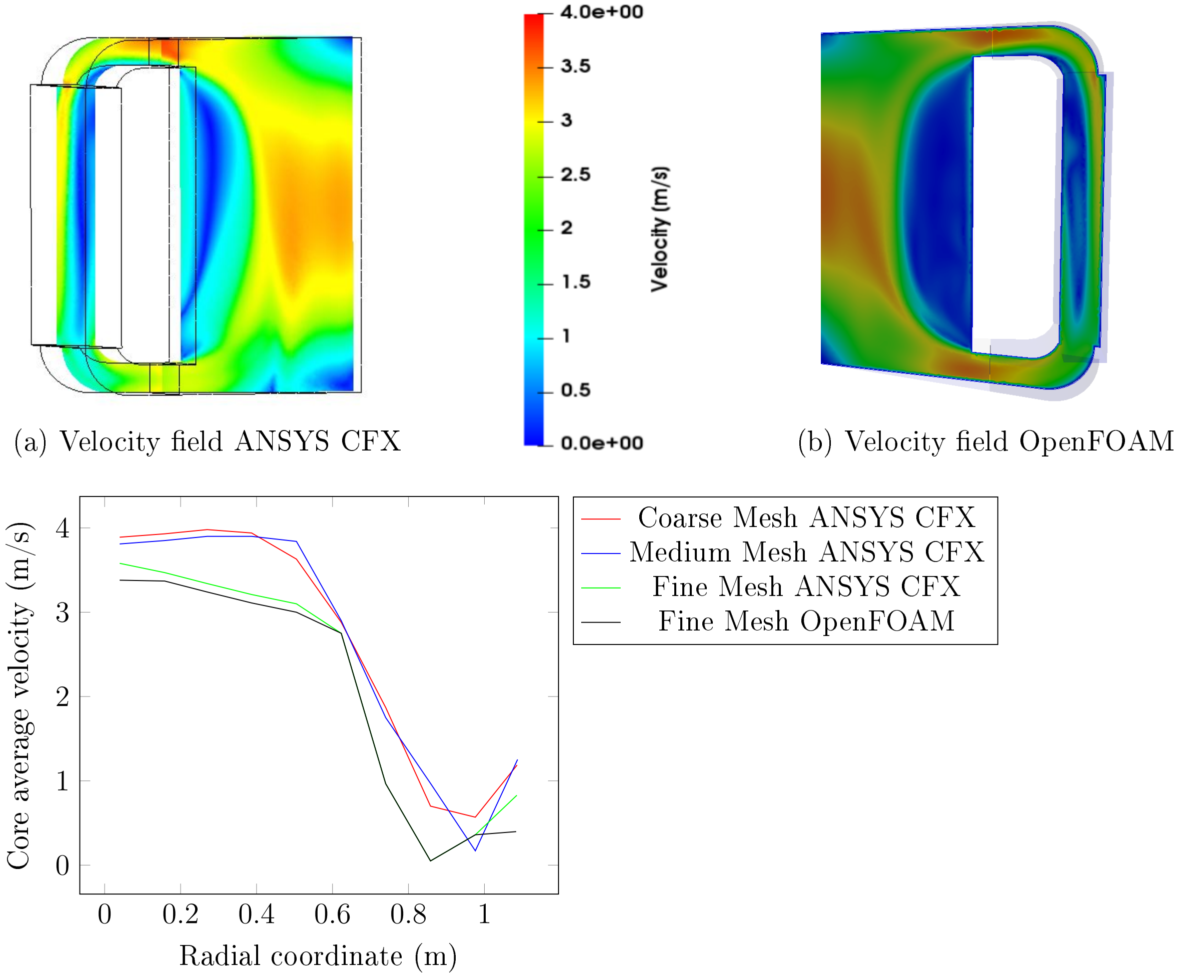


(a) ANSYS CFX sample mesh (coarse)

(b) OpenFOAM mesh (fine)

Figure 5: Volume discretization (meshing)

part; whereas the salt appears to be stagnant near the walls due to the large recirculation vortexes. These results are supported by data available in the literature [8] and are an effect of the specific flow pattern due to the core shape. The OpenFOAM results are in very good agreement with the ANSYS CFX results, showing slight differences only in the recirculation zone; the velocity drops smoothly in ANSYS CFX from the core center to the perifery, however slightly rising again near the core wall, whereas for the OpenFOAM the velocity distribution is more flat. The average in-core velocity is thus slightly smaller in the OpenFOAM than in ANSYS-CFX. These differences are likely to be attributed to the realizable k-$\epsilon$ turbulence model used in OpenFOAM.
The mesh impact on the in-core velocity distribution is shown in Figure 6(c). The results obtained with the fine mesh are on average 20 % less than those obtained for the coarse mesh.



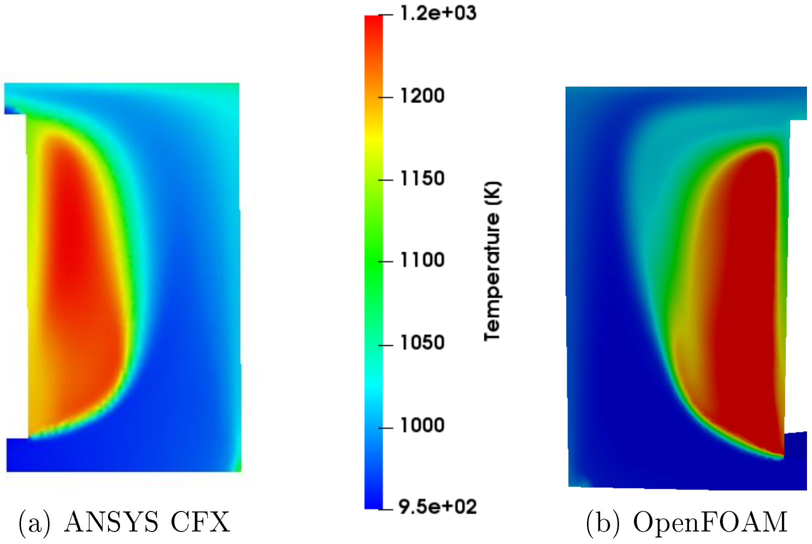
(a) Velocity field ANSYS CFX

(b) Velocity field OpenFOAM

(c) Mesh impact on the average in-core velocity distribution. The results are presented for coarse ($\leq 2 \times 10^5$), medium ($< 5 \times 10^5$), fine ($\geq 5 \times 10^5$)

Figure 6: Velocity fields and the mesh impact analysis

The in-core temperature distribution is shown in Figure 7. The maximum temperature difference is 300 K, due to the hot spot appearing near the core wall where the large recirculation vortexes cause local stagnation of the flow. The temperature difference between the core inlet and outlet is around 100 K.



(a) ANSYS CFX

(b) OpenFOAM

Figure 7: In-core temperature fields

# 5 Conclusions

This paper compares two approaches to solve coupled neutronics-thermal hydraulics equations with reference to molten salt reactors, applied on a simple 3D case geometry. A method of black-box coupling between two conventional software codes, ANSYS CFX and MCNP, is developed and presented. This method is compared to the multiphysics coupling technique developed in OpenFOAM [8], and the reactor steady-state conditions are examined. The conventional OS technique employing two distinct mono-physics software codes (MCNP and ANSYS CFX) shows aceptable capability to model reactor steady-state behavior, and the results are in good agreement with previous studies on an equivalent axial-symmetric (2D) geometry [13]. However, taking into account that MCNP is designed to deal with static fuels only, and allows inclusion of fuel flow effects at high computational and programming costs, as well as its limited capabilities of modelling transient scenarios, the OS coupling approach using MCNP is not optimal for time-dependent analysis of liquid fuel reactors. Moreover, being a standard OS technique, the ANSYS CFX to MCNP coupling lacks flexible control over convergence of the distinct physics modules. The multiphysics coupling using OpenFOAM allows better control over numerical stability and can potentially save computational costs, making it the preferable choice for tight coupling of neutronics and thermal hydraulics for Molten Salt Reactors. It is able to directly take into account the fuel flow effects on reactor kinetics, most imprtant, on the delayed neutron precursor distribution, and thus provides an adequate coupled neutron kinetics-thermal hydraulics picture.

# References

[1] D. Lee, T.J. Downar, A.Ulses, “Analysis of the OECD/NRC BWR turbine trip transient benchmark with the coupled thermal-hydraulics and neutronics code TRAC-M/PARC”, Nuclear Science and Engineering, Vol. 148, No. 2, pp. 291-305

[2] M. Hron, J. Uhlir, C. Renault, “Molten Salt Reactor: current status and future prospects within European activities”, Proceedings of FISA 2006, Luxembourg 2006, pp. 270-280

[3] Jean C. Ragusa, Vijay S. Mahadevan, “Consistent and accurate schemes for coupled neutronics thermal-hydraulics analysis”, Nuclear Engineering and Design, Vol. 239, pp. 566-579

[4] ANSYS CFX Solver Theory Guide, ANSYS, Inc., 2006

[5] MCNP, https://mcnp.lanl.gov/

[6] OpenFOAM User’s Guide Version 5.0, 2017

[7] ENDF, http://www.nndc.bnl.gov/endf/b7.1/

[8] M. Aufiero, M.Brovchenko, A.Cammi, I.Clifford, O.Geoffroy, D. Heuer, A. Laureau, M. Losa, L. Luzzi, E. Merle-Lucotte, M.E.Ricotti, H. Rouch, "Calculating the effective delayed neutron fraction in the Molten Salt FastReactor: Analytical, deterministic and Monte Carlo approaches", Annals of Nuclear Energy 65(2014):78-90

[9] I.Clifford, H. Jasak, "The application of a multi-physics toolkit to spatial reactor dynamics", International Conference on Mathematics, Computational Methods and Reactor Physics. Saratoga Springs, NY, USA, May 3-7, 2009.

[10] I. Clifford, K. Ivanov, "PBMR 400MW benchmark calculations using the simplified P3 approach", Proceedings of HTR. Prague, Czech Republic, October 18-20, 2010

[11] H. Rouch, O. Geoffroy, P. Rubiolo, A. Laureau, M. Brovchenko, D. Heuer, E. Merle-Lucotte, “Preliminary thermal-hydraulic core design of the Molten Salt Fast Reactor MSFR”, Annals of Nuclear Energy , Vol. 64, 2014, pp. 449–456.

[12] M. Brovchenko, D. Heuer, E. Merle-Lucotte, M. Allibert, V. Ghetta, A. Laureau, P. Rubiolo, “Design Related Studies for the Preliminary Safety Assessment of the Molten Salt Fast Reactor”, Nuclear Science and Engineering, Vol. 175, 2013, pp. 329-333

[13] C. J. Greenshields, “OpenFOAM User Guide”, 2017

[14] C. Fiorina, D. Lathouwers, M. Aufiero, A. Cammi, C. Guerrieri, J.L. Kloosterman, L. Luzzi, M.E. Ricotti, 2014. “Modelling and analysis of the MSFR transient behaviour”, Annals of Nuclear Energy Vol. 64, 2014, pp. 485–498

[15] V. Ignatiev, O. Feynberg, A. Merzlyakov, “Progress in Development of MOSART Concept with Th Support”, Proceedings of ICAPP 2012, Chicago, USA, 2012

[16] ANSYS CFX-Pre User’s Guide, ANSYS, Inc., November 2013

[17] M.N. Spijker, “Modelling and analysis of the MSFR transient behaviour”, Journal of computational and applied mathematics Vol. 72, 1996, pp. 393–406

[18] ANSYS Meshing User’s Guide, ANSYS Inc., 2010

# COUPLED NEUTRONICS/THERMAL HYDRAULICS ASSESSMENT OF GRAPHITE MODERATED MOLTEN SALT REACTORS

A. Nalbandyan[1], E. B. Klinkby[1], B. Lauritzen[1], J. Groth-Jensen[2], and R. Steyn[2]

[1]Center for Nuclear Technologies, Technical University of Denmark, Roskilde, Denmark,
*asnalb@dtu.dk,esbe@dtu.dk,blau@dtu.dk*,

[2]Seaborg Technologies,Copenhagen, Denmark,
*jgj@seaborg.co,rwn@seaborg.co*

**Abstract**

To model a Molten Salt Reactor (MSR) core, we apply a Multiphysics coupling scheme between the finite volume Computational Fluid Dynamics (CFD) code OpenFOAM and the Monte Carlo based neutronics code Serpent. The scheme employs the Serpent Multiphysics interface, which allows for high fidelity coupling to OpenFOAM by directly passing variable fields between the two codes. We simulate a graphite-moderated channel type MSR and compare the simulation results to data available on the Molten Salt Reactor Experiment (MSRE). Specifically, fuel and graphite temperature profiles and fuel velocity fields are derived for steady state operation and compared to the results of model calculations performed at the Oak Ridge National Laboratory (ORNL). A simple transient scenario of a step reactivity insertion is also modeled and the feedback of the system is evaluated and compared to experimental results.

## 1 Introduction

The Molten Salt Reactor (MSR) is considered a promising design concept with potential for meeting the safety, economy and sustainability objectives of Generation IV reactors, and in 2000 the fluid-fueled MSR became one of the six technologies selected by the Generation IV International Forum (GIF) as candidates for the next generation nuclear energy system. The MSR employs a high-temperature circulating fuel salt, moderated by graphite or yielding a fast neutron spectrum [2].

In the late 1960's the Molten Salt Reactor Experiment (MSRE) was constructed at the Oak Ridge National laboratory (ORNL); it comprised a circulating fuel, single-region, graphite moderated reactor designed for heat generation. A power output of 10 MW thermal was initially decided upon [3], which was later reduced to 7.5 MW due to heat transfer limitations [4]. The experiment aimed to demonstrate the practicality and efficiency of a molten salt fueled reactor. During the project extensive calculations were performed and experimental data were collected on the reactor steady state operation as well as some transient scenarios. This data were published in series of ORNL reports. Recently large efforts have been made towards research and development of MSRs [5]. Although the focus has been on the Molten Salt Fast Reactor concept (MSFR) featuring a non-moderated reactor core, reactor operation data is still only available for the graphite moderated concept.

To demonstrate compliance with the design and safety criteria for a next generation MSR, detailed modeling of the neutronics and thermal hydraulic properties of the reactor is required. Simulation of liquid fuel systems, however, calls for a new level of closely coupled neutronics and thermal-hydraulics models, *inter alia* to account for the transport of delayed neutron precursors with the flow. The fuel circulation time is typically of the same order of magnitude as the half-lives of delayed neutron precursors; thus the effective delayed neutron fraction present in the core is affected by the flow velocity. In turn, the neutron flux distribution determines the fuel temperature fields; hence, a tight coupling between the thermal-hydraulics and the neutronics modules is required [6].

In this article we couple the 3D neutronics code Serpent [9] and the open source CFD tool OpenFOAM [10] and apply this Multiphysics coupling technique to a graphite-moderated channel type MSR mimicking the MSRE. Graphite and fuel temperature distributions as well as the power distribution are calculated and compared to data available on MSRE steady state operation [11]. A simple transient scenario of a step reactivity insertion is also modeled and the reactivity feedback of the system is evaluated and compared to the experimental results [12].

## 2 Multiphysics Coupling Technique

Nuclear reactor modeling is conventionally carried out using coupled monophysics codes: Monte Carlo codes are used for reactor neutron kinetics simulation and system codes are used for thermal hydraulics modelling. Such an approach allows taking into account the reactor feedback mechanisms related to the temperature and density variation of different materials. For detailed calculations of the coolant flow in the reactor core during normal as well as under accident scenarios Computational Fluid Dynamics (CFD) codes are applied. However, taking into account the heavy computational costs of Monte Carlo methods as well as CFD, it is only recently that attempts have been made to couple these codes [1]. To have Monte Carlo/CFD coupled calculations is particularly important for applications to liquid fuel reactors and recently considerable development is recorded in this area, with conventional as well as in-house developed software being applied. The methods applied to studies of Molten Salt Reactors mostly feature coupling between two mono-physics codes, or adaptation of a CFD module to solve multiphysics equations involving delayed neutron precursor transport. A detailed overview of techniques available for coupling of MSRs can be found in reference [7]. In particular, with regard to studies on MSRE, COMSOL Multiphysics solver has been adapted to solve for delayed neutron precursor convection and diffusion [8].

In this article an external coupling of the Serpent Monte Carlo code (version 2.1.30) [9] to the OpenFOAM open source C++ tool-box (version 1806) [10] is presented and applied to an MSRE core channel. Serpent code is used to perform criticality calculation in order to obtain the fission power distribution and OpenFOAM solver *chtMultiRegionFoam* is used to solve for fluid flow. The method utilizes Serpent multi-physics coupling interface [1], which enables exchange of field files between OpenFOAM and Serpent without having to modify the files. Hence, the coupling allows for efficiently passing power distribution, temperature and density fields between the neutronics and thermal-hydraulics modules.

Temperature and density output files from OpenFOAM are used as input for the Serpent and a volumetric power distribution file written by Serpent is used as input for OpenFOAM. The general coupling scheme is presented in Figure 1.

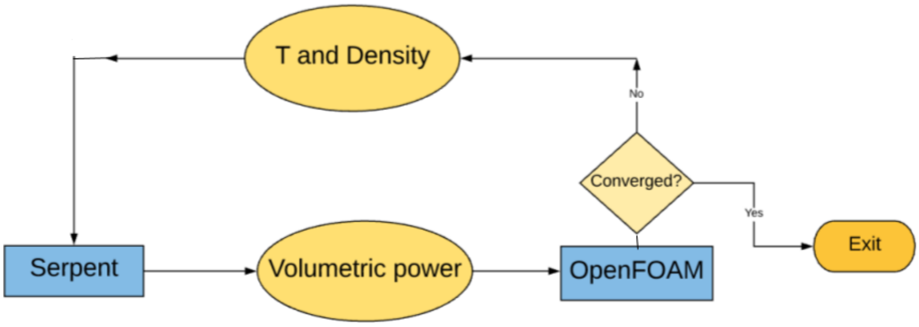


Figure 1: Serpent to OpenFOAM general coupling scheme

The calculation is initiated with the fuel having a uniform temperature and density and iteration cycles of the neutronics and thermal hydraulics modules are carried out until convergence is reached. As a convergence criterion the difference between the temperature fields of two consecutive OpenFOAM calculations is used.

# 3 MSRE Model

The MSRE core represents a lattice of graphite stringers with half-channels imprinted in each graphite stringer face for fuel passage [3]. A typical core graphite stringer arrangement is shown in Figure 2.

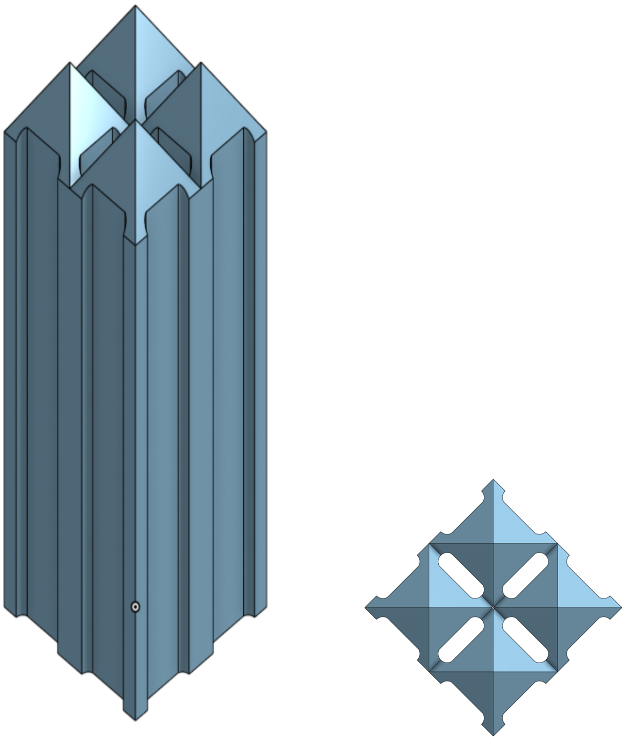


Figure 2: MSRE typical graphite stringer structure

## 3.1 The Thermal Hydraulic Model

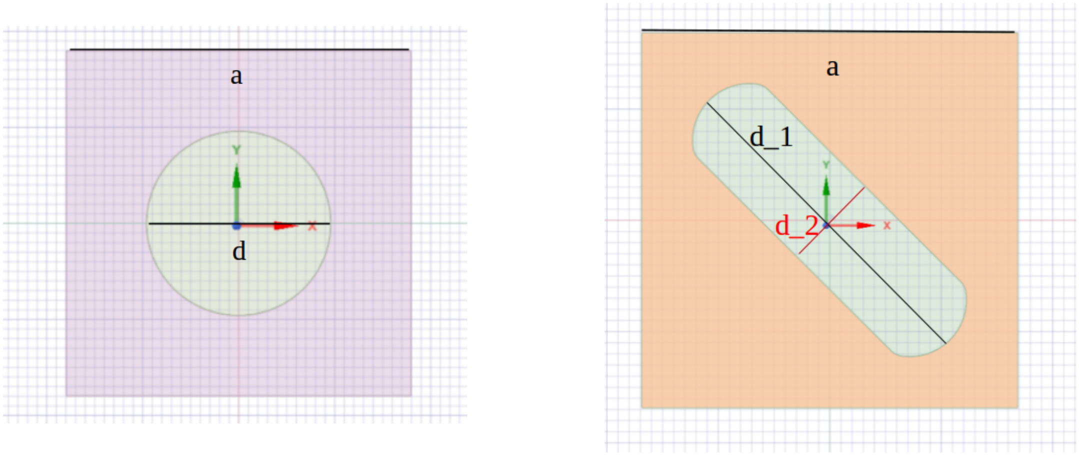
In the thermal hydraulic model of the MSRE the core is divided into five concentric annular regions [11]. A summary of the main properties of these regions is provided in Table 1.

| Region | Number of full sized fuel channels | Fuel volume fraction | Average fuel velocity[m/s] |
|---|---|---|---|
| 1 | 12 | 0.256 | 0.60 |
| 2 | 940 | 0.224 | 0.183 |
| 3 | 108 | 0.224 | 0.454 |
| 4 | 78 | 0.142 | 0.249 |
| 5 | 0 | 1 | 0.079 |

Table 1: Main properties of five core regions of the MSRE. Region 5 is the annulus between the graphite and the core shell. [11]

For this work the core Region 2 having the largest number of full-sized fuel channels is considered. The ORNL analytical model assumes a circular fuel channel shape to ease the calculation procedure and to be able to use thermal hydraulic coefficients available for circular channel shapes [13]. In this work both the simplified cylindrical channel shape as well as the stadium-shaped fuel channel are analyzed. The results obtained with the cylindrical fuel channel are used to benchmark against ORNL calculations; subsequently we evaluate the effect of having stadium-shaped fuel channels.

Both geometrical models are constructed keeping the fuel to moderator volume ratio equal to that of the MSRE core region 2. Since the Monte Carlo calculations and the coupling is computationally heavy, the model considered here is comprised of a single basic cell. The basic cell consists of a fuel channel surrounded by graphite. In Figure 3 both the cylindrical and stadium-shaped channels are represented for the thermal hydraulic module.



(a) Cylindrical fuel channel

(b) Stadium shaped channel

Figure 3: Thermal Hydraulics Geometry Model: a is the graphite lattice side, d is the cylindrical channel diameter, d _1 and d_2 are the stadium shaped channel large and small diameters respectively.

The main geometrical parameters are given in Table 2.

| Channel shape | Diameter [mm] | Channel length [mm] | a [mm] | Fuel volume fraction |
|---|---|---|---|---|
| Cylindrical | 18.032 | 1600 | 35.921 | 0.224 |
| Stadium | 24.2 ×10.2 | 1600 | 33.74 | 0.224 |

Table 2: Main geometrical dimensions of the two models

The main thermodynamical properties of the fuel and the graphite used as moderator are summarized in the Table 3. All the properties are extracted from ORNL reports on the MSRE operation [11], and temperature dependent representation is used for thermal hydraulic simulations.

| Material | $\rho [gcm^{-3}]$ | $\mu [Pas]$ | $\kappa$ [ $Wm^{-1}K^{-1}$] | $C_p$ [ $Jkg^{-1}K^{-1}$] |
|---|---|---|---|---|
| Fuel [at 922 K] | 2080 | 0.007 | 5.53 | 1970 |
| Graphite [at 922 K] | 1860 | - | 58.8 | 1760 |

Table 3: Material properties; $\rho$ is the density, $\mu$ is the dynamic viscosity, $\kappa$ is the thermal conductivity and $C_p$ is the heat capacity.

All meshes are generated using the ANSYS workbench meshing [14] and are translated to the OpenFOAM format. In Figure 4 meshes for both models are shown.

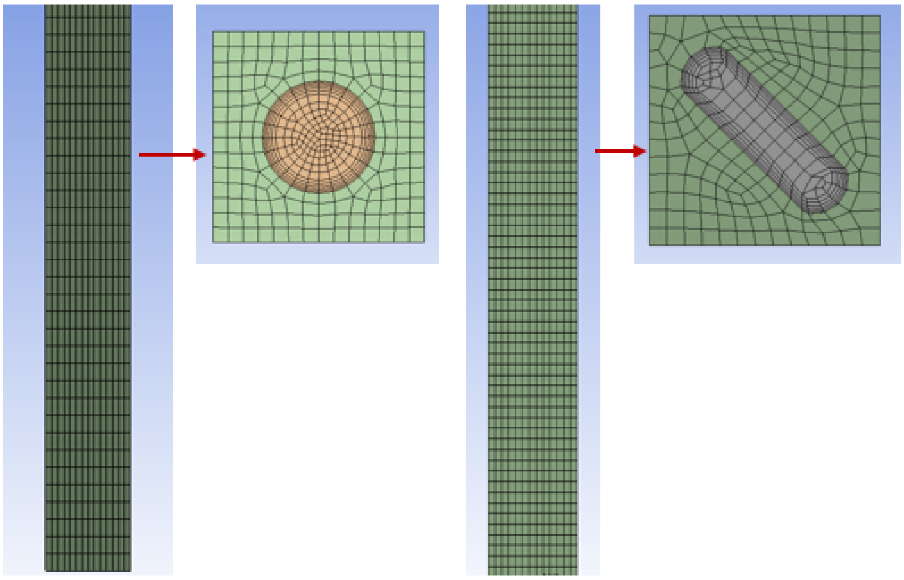


Figure 4: Mesh: side and top views

Both meshes are hexahedral; the cylindrical channel model has average orthogonal quality of 0.98 and the stadium-shaped channel model has an average orthogonal quality of 0.9. A mesh sensitivity analysis is also performed. As tracking variable for mesh independence the average fuel velocity in the fuel channel is chosen. Figure 5 shows that the solution becomes mesh independent with a element number of $\sim$ 180000 for the cylindrical model and $\sim$ 230000 elements for the stadium-shaped model.

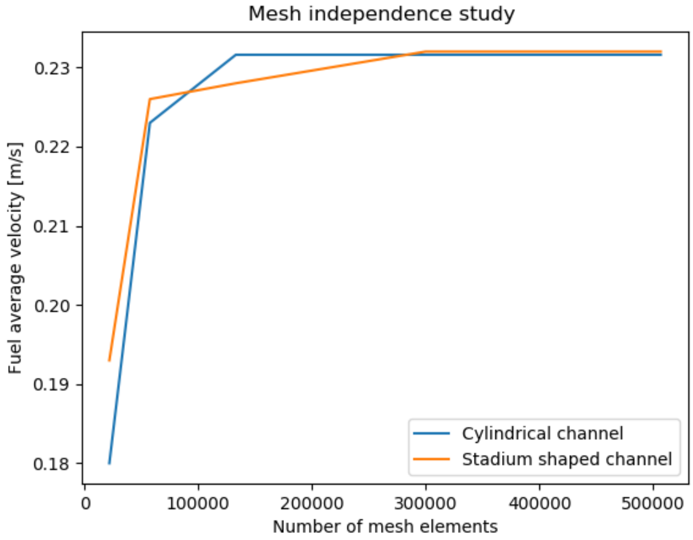
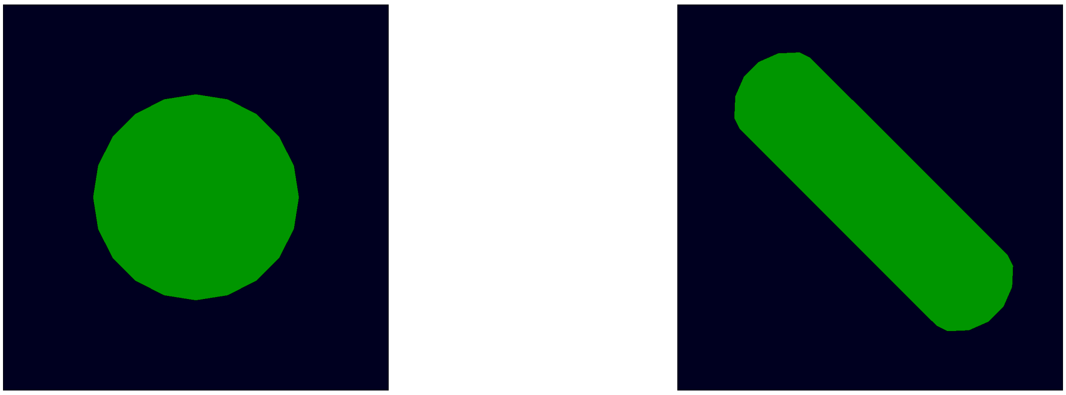


Figure 5: Fuel average velocity as a function of mesh element number

Using the hydraulic diameter ($D_h$) of a single channel, the Reynolds number (Re) was calculated to be approximately 1008. The flow regime is considered laminar although there are buoyancy effects to take into account. The Richardson number (Ri) for flow over a vertical plate was calculated to be 1.8, showing mixed convection, for this reason the thermo-hydraulics model takes density changes into account. No turbulence model is included in the thermal hydraulics model for the steady state condition because of the channel flow limiting transition from the buoyancy effects and also to keep consistency between the MSRE calculations and the thermo-hydraulic model.

## 3.2 Neutronics Model

The neutronics model geometry is constructed using OpenFOAM unstructured mesh. The model geometry is shown in Figure 6. The model is finite in axial direction, whereas reflective boundary conditions are applied in radial direction.



(a) Cylindrical fuel channel

(b) Stadium-shaped fuel channel

Figure 6: Neutronics geometry model, top view

The fuel composition is given in Table 4. According to the ORNL data, criticality is established at 0.15 mol % of $UF_4$[3]. The isotopic composition of the Uranium is as follows: 1 % $^{234}$U, 93% $^{235}$U, 1% $^{236}$U and 5%$^{238}$U. As moderator CGB grade graphite developed at the ORNL is used; this has the attractive feature of being both a good moderator and having low permeability for the fuel [11].

| Component | *mol%* |
|---|---|
| LiF | 70 |
| $BeF_2$ | 24.85 |
| $ZrF_4$ | 5 |
| $UF_4$ | 0.15 |

Table 4: MSRE fuel composition for Steady State analysis

The simulation starts by running the Serpent Monte Carlo criticality calculation code, which gives the initial power distribution. The Serpent 2.1.30 solver is used in this work. A total of 100 neutron generations are simulated, with 10000 neutrons per generation and 25 generations are inactive; they are used to ensure fission source convergence. The calculation is performed in coupled neutron gamma transport mode. For the neutron cross-section data the JEFF 3.1 cross-section libraries are used, adopting 8 group delayed neutron precursor representation and with thermal scattering libraries for the graphite moderator [12]. The initial guess for the $k_{eff}$ is one. The thermal scattering data is interpolated by Serpent and adjusted to the graphite temperatures. The power distribution is then passed to OpenFOAM and a detailed temperature and density map is generated, which is subsequently used as input for Serpent.

### 3.3 Transient Simulation Procedure

The MSRE dynamic analysis is comprised of series of reactivity insertion tests performed to evaluate the time response of the system following a rapid change in reactivity [5]. The test signals were of three types: Pseudo Random Binary Tests (PRBT) perfomed by jogging the control rod in order to generate a sequence of positive and negative reactivity insertions, pulse tests featuring single pulse like disturbance to the system and finally step reactivity insertion tests. The tests were performed for $^{233}$U fueled MSRE, thus the fuel salt used in the steady-state analysis is changed for this part of the work. The new fuel composition is provided in Table 5.

In this work step reactivity insertion is simulated and the system behavior is compared to the results of the MSRE experimental dynamic analysis.

Conventional criticality calculations assume the modeled systems to be at steady state condition. In recent releases of Serpent a functionality to model the time evolution of a critical systems is facilitated. It is a prerequisite for the implementation, that the systems should be exactly critical, and the code developers stress, that the expected modelling accuracy depends strongly on this assumption. Because of the reflective boundary conditions used in the steady state simulations described above, the system is slightly supercritical. In order to bring the system down to critical, an enrichment optimization procedure based on the Newton-Raphson method has been employed. Since the obtained value of the criticality has a statistical uncertainty, special care must be taken to ensure convergence. Thus a tolerance parameter is introduced and the statistics of the underlying Serpent criticality search is increased accordingly when approaching the criticality.

In order to simulate a transient scenario in Serpent, first a steady state neutron source distribution and the delayed neutron precursor distribution fields are generated. The steady state (SS) source distribution output is linked to a standard Serpent criticality calculation, similar to the one used for the steady state calculation. Serpent tracks the neutrons and the delayed neutron precursors in a continuous manner

through time. At the beginning of each time interval a source for the prompt neutrons and precursor data is be provided.
For the transient Serpent simulation an additional time binning is specified which is finer than the one used for the thermal hydraulics calculations. As such, the approach presented in this work deals with two types of time steps; one for the thermal hydraulics and one for the neutronics calculations. Since the time scale governing the neuton dymamics is shorter than that of the thermal hydraulics, the first set of time steps is chosen as a subset of the latter. From the transient Serpent simulation the positions of the delayed neutron precursors is obtained and is then combined with the fluid velocity fields obtained from OpenFoam in order to perform the drift of the delayed neutron precursors. Additionally the power profile from the transient Serpent simulation is used to adjust the power level for the next time step iteration. The coupling scheme used in the transient simulations is presented in Figure 7.

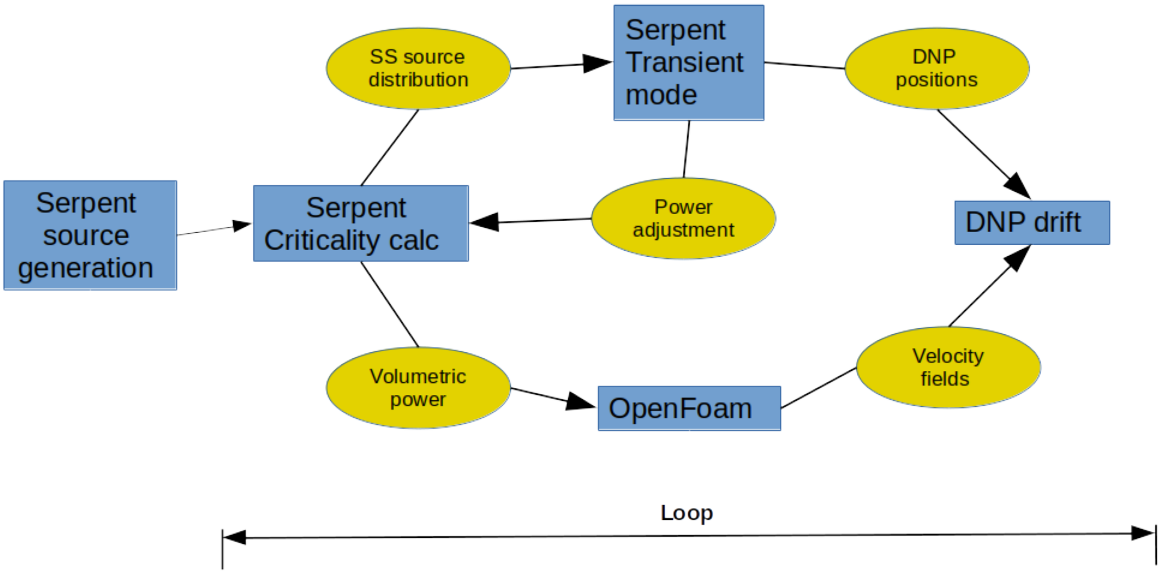


Figure 7: Serpent to OpenFOAM coupling scheme used for the transient simulations

Similar to the steady state simulations described in the previous section, the temperature and density distributions are linked from OpenFoam to Serpent in the initialization of the next time-step iteration (not shown in the figure).

| Component | *mol%* |
|---|---|
| $LiF_2$ | 58.9 |
| $BeF_2$ | 36.5 |
| $ZrF_4$ | 4.5 |
| $UF_4$ | 0.115 |

Table 5: MSRE fuel composition for transient tests. The fuel is enriched to 91.49 % $^{233}$U

## 4 Results

In Figure 8 the normalized axial neutron flux distributions for the simulated models and the ORNL calculations are shown. The curves show the thermal neutron flux, with thermal neutron energies between 1 eV and 0.1 MeV are taken. The fast neutron flux covers the energy range of 0.1 MeV to 1 MeV. Simulation results are in good agreement with the MSRE results, based on calculations with Equipose: a multiregion, two-group diffusion program [4].

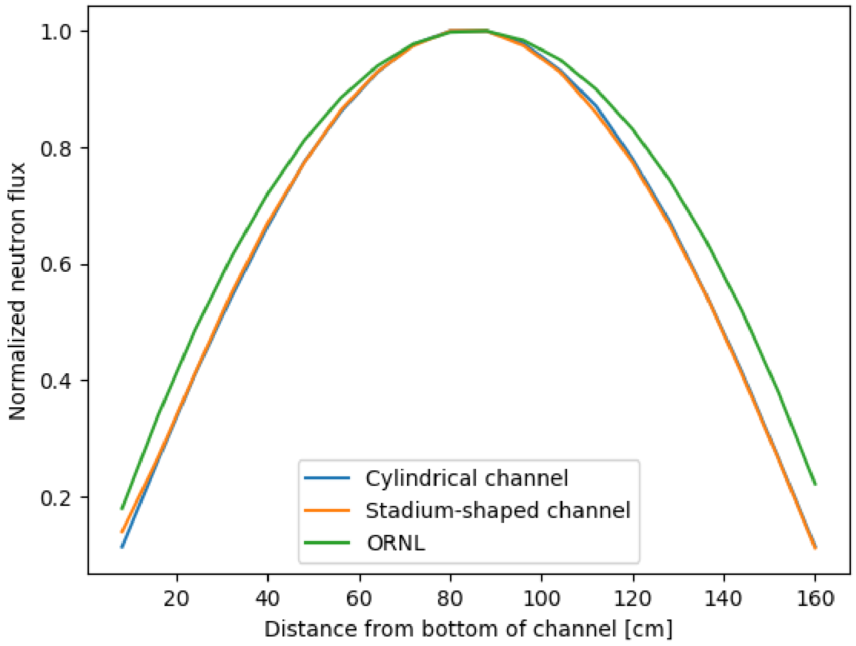


Figure 8: Axial distribution of slow flux for the channel at 17.78 cm from the core centerline

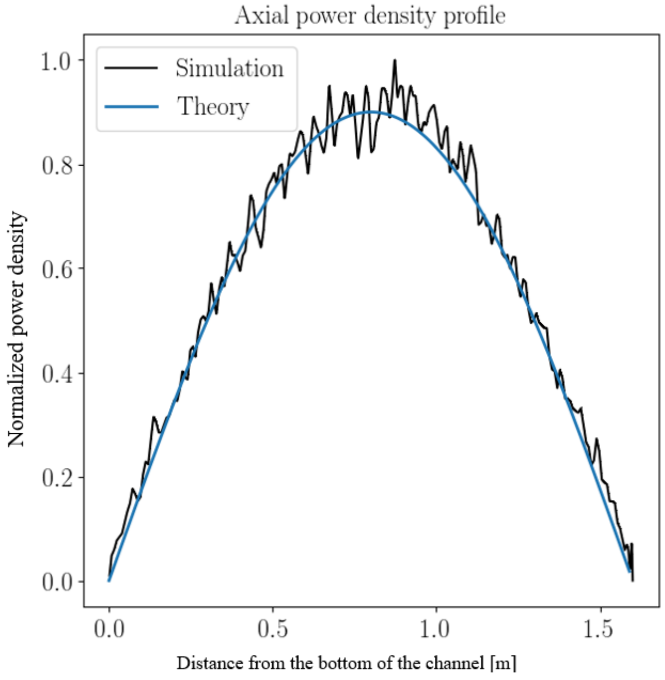


Figure 9: The normalized power distribution

More interesting in terms of the effect on the reactor temperature variations is the power density distribution provided in Figure 9. The axial distribution of the power density along the centerline of the fuel channel is shown. The power density is proportional to the fission reaction rate or to the neutron flux. In Figure 10 the mean fuel and graphite axial temperature distributions for the fuel channel are shown for three cases: ORNL calculations for cylindrical channel shape, coupled Serpent/OpenFOAM simulation for cylindrical channel shape and finally coupled Serpent/OpenFOAM simulation for the MSRE stadium-shaped channel. The simulated data for a cylindrical channel shape is in good agreement with the model calculations; the maximum difference between the graphite temperatures for simulated case and ORNL calculations is 8.64 K, and for the fuel the curves are almost identical with the average difference of 1.49 K. In the simulation the graphite heats up slightly more towards the channel exit, whith the maximum temperature difference being between the channel outlet temperatures for the simulation and the ORNL calculation. This could be attributed to the possible differences in the deposited volumetric heat values used in the ORNL calculations and in the simulation, as well as the temperature sampling location for both cases. The comparison between the simulation and the ORNL model calculations for the cylindrical shaped channel shows, that the coupled simulation technique is capturing the system behaviour correctly.

The simulation case with a realistic stadium-shaped channel geometry has a graphite mean temperature which is on the average 10 K lower than for the cylindrical shape for the largest part of the curve, except the last 0.1 m of the channel where it is larger than the ORNL calculations. Hence, for the realistic channel shape the graphite temperature is lower than for the cylindrical case [4]. The fuel temperature is on average 1.6 K lower for the stadium-shaped channel than for the the cylindrical shape.

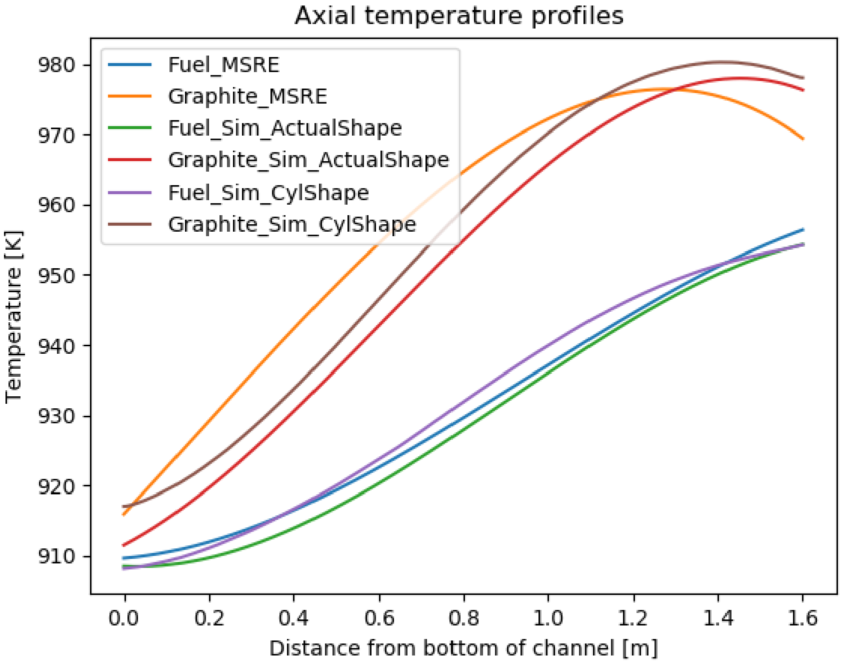


Figure 10: Simulation vs ORNL model calculation results

In both the simulation and in the ORNL model fuel soak-up in graphite is not considered.

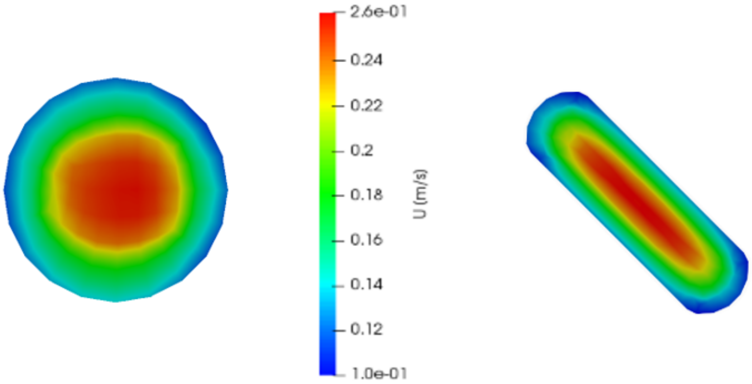


Figure 11: Velocity distributions

At a steady state power operation the mean graphite stringer temperature is higher than the mean fuel temperature in the adjacent channel:

$$T_g = T_f + \Delta T \tag{1}$$

The temperature difference $\Delta T$ can be attributed to:

a. The near-wall fuel velocity is lower than the velocity in the channel central part (Figure 11) and there is internal heat generation in fuel, which causes the fuel temperature adjacent to the wall to be higher than the average across the channel. The effect is known as Poppendiek phenomenon [11]. The ORNL model considers coefficients for cylindrical channel shape in order to evaluate the Poppendiek effect [13]. As shown in Figure 9 this assumption slightly overestimeates the Poppendiek effect.

b. Heat is generated in the graphite itself and the temperature difference at the interface is needed to conduct the heat to the surface [11]. The ORNL calculations assumed 6 % of the total heat to be generated in the graphite due to neutron and gamma heating. The Serpent simulation presented in this work accounts for only prompt gamma heating, as Serpent does not account for delayed gammas. Thus, the total heat deposited in graphite according to the simulation is 4.3 %.

The fuel mean temperature, the graphite bulk mean temperature as well as the average temperature difference between graphite and the fuel for all three cases for the fuel channel of the core are shown in Table 6.

| Model | Fuel[K] | Graphite T [K] | Δ T [K] |
|---|---|---|---|
| ORNL | 930.7 | 957 | 26.3 |
| Cylindrical channel | 931.5 | 955 | 23.5 |
| Stadium shaped | 929 | 950 | 21 |

Table 6: Fuel and graphite mean temperatures and the temperature difference for steady state operation

### 4.1 Reactivity Insertion Transient Case

In this section the results from the trainsient simulation are presented and compared to MSRE data taken from [12] and [18].

In the same way that the steady state simulations could be benchmarked by looking at the spatial distributions of the neutron flux, so can be the transient simulations. Among other things, this can be used to test the coupling between neutronics and fluid dynamics.

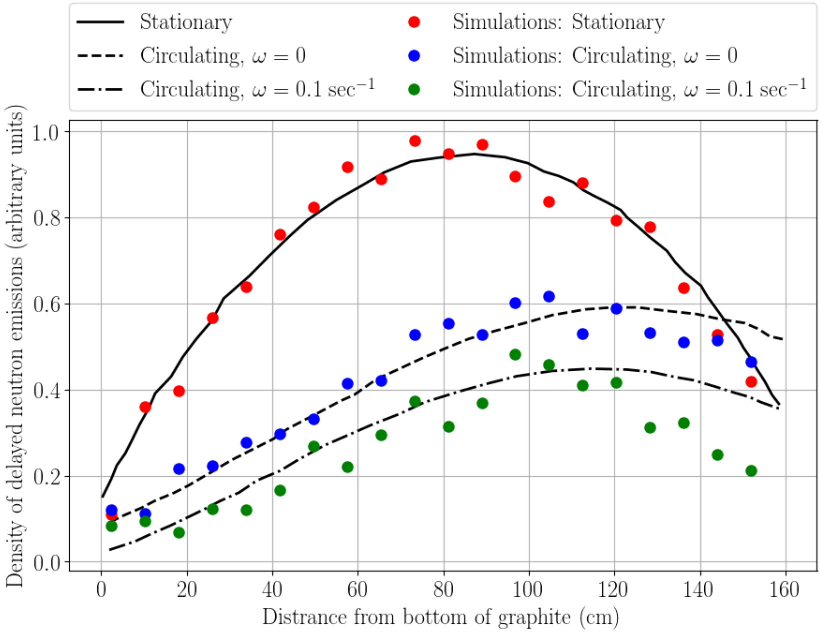


Figure 12: Axial distribution of the net source delayed neutrons with the reactor initially at 10 MW

Figure 12 shows the axial distribution of the net delayed neutron source for the particular cases of the inverse reactor period $\omega = 0$ (circulating salt in a critical reactor) and $\omega = 0.1$ sec$^{-1}$ (10 sec stable period), and also for the case when the fuel is not circulating.The simulation data shown in Fig 12 is obtained by multiplying the individual precursor concentrations (groupwise) by their corresponding decay constant. The stable reactor period for a given reactivity insertion is found by solving a modified version of the inhour equation [18]. As seen in this figure the results from the present simulation are compatible with the ORNL model, though there are deviations close to the end of the fuel channel.

Another and perhaps more important benchmark for the transient simulation, is the power response to a reactivity insertion. For a reactor operating at 8 MW, the power response to a reactivity insertion of 0.0248 % $\delta k/k$ is shown in figure 13.

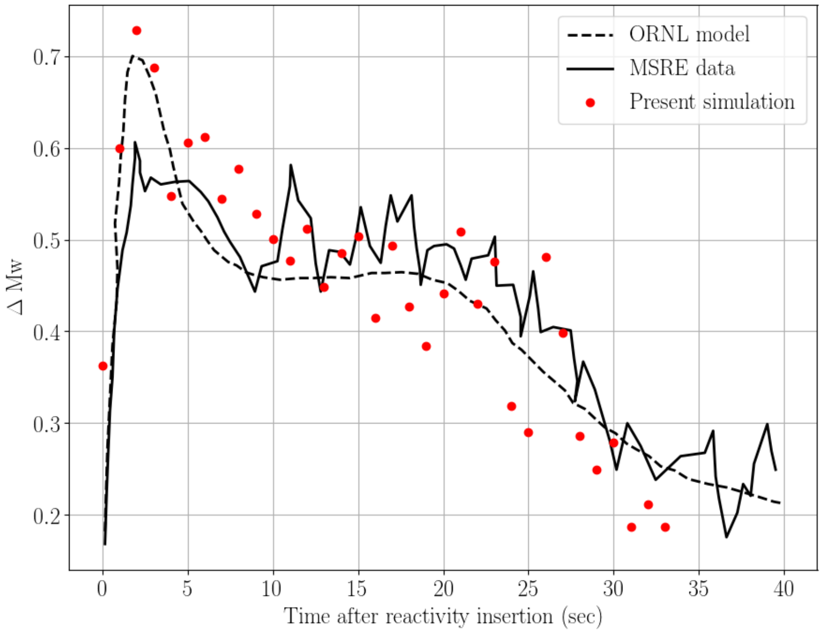


Figure 13: Reponse of the power to a step change in reactivity

The maximum power level is reached within seconds after the reactivity insertion. As a consequence, the reactivity insertion is followed by a rapid increase in fuel temperature which, due to the negative temperature coefficient of reactivity $\alpha_T$, causes the power level to drop. The drop in power coupled with the negative $\alpha_T$ causes the reactor to re-estiablish criticality and the power level to reach a plateau in a few seconds. Once the fuel re-enters the core at higher temperature, the reactor power decreases once more and a new plateau is established and so forth. In general all of the effects discussed above are well described by the present simulation. However a noticable feature is that the decrease in power level tends to occur faster than in data. This effect which can be explained by overestimation of $\alpha_T$ in the modeling. However due to the limited statistics of the transient simulations, fluctuations are present and thus makes it difficult to draw a clear cut conclusion.

# 5 Conclusion

In this paper Multiphysics coupling of the Serpent 3D Monte Carlo neutronics code to the OpenFOAM CFD code is described and applied to steady-state as well as transient analysis of the MSRE core. The purpose of this work is to demonstrate the efficiency of such a Multiphysics coupling technique in modeling liquid fueled reactors. Additionaly, a detailed CFD assessment of the actual MSRE fuel channel shape is done.

A fuel channel of the core is modeled and temperature profiles for fuel and graphite are derived. We compare the stadium-shaped fuel channel to the simplified cylindrical channel shape, as considered in the ORNL model. It is shown that the average graphite temperature is slightly lower (7 K on the average). The fuel temperature curves with both approaches are almost identical.

The Multiphysics coupling technique proves to be well adapted to capture the characterstics of the investigated liquid fuel reactor channel. Serpent 3D multiphysics coupling interface allows convenient coupling to OpenFOAM; the volumetric power files produced by Serpent are supported by OpenFOAM and in return Serpent is able to take as input cell-wise temperature and density profiles from OpenFOAM. Automatized coupling procedure using a Python script allows to set a user specified convergence criterion and handles efficiently the file transfer between the software.

# References

[1] R. Tuominen, V. Valtavirta, J. Peltola, and J. Leppänen, "Coupling Serpent and OpenFOAM for neutronics-CFD multiphysics calculations", Proceedings of PHYSOR 2016, May 1-5, 2016.

[2] David LeBlanc, "Molten Salt Reactors:A new beginning for an old idea", Nuclear Engineering and Design 240 (2010) 1644-1656.

[3] R.C. Robertson, "MSRE Design And Operations Report, Part I, Description of Reactor Design", ORNL, January 1965.

[4] T.W. Kerlin and S.J. Ball, "Experimental Dynamic Analysis of the Molten Salt Reactor Experiment", ORNL, 1966

[5] J. Serp et al., "The Molten Salt Reactor in Generation IV: Overview and Perspectives", Progress in Nuclear Energy, volume 77, pages 309-319, November 2014.

[6] M. Aufiero, P. Rubiolo, M. Fratoni, "Monte Carlo/CFD coupling for accurate modeling of the delayed neutron precursors and compressibility effects in molten salt reactors", Transactions of the American Nuclear SocietyAt: San FranciscoVolume: 116, 2017

[7] A. Cammi, V. Di Marcello, L. Luzzi, M. E. Ricotti, "A multi-physics modelling approach to the dynamics of Molten Salt Reactors", Annals of Nuclear Energy 38(6):1356-1372

[8] M. Zanetti,L. Luzzi,A. Cammi, C. Fiorina, "An Innovative Approach to Dynamics Modeling and Simulation of the Molten Salt Reactor Experiment, PHYSOR 2014 - The Role of Reactor Physics Toward a Sustainable Future, Kyoto, Japan, September 2014."

[9] J. Leppänen, et al. "The Serpent Monte Carlo code: Status, development and applications in 2013." Ann. Nucl. Energy, 82 (2015) 142-150.

[10] `https://www.openfoam.com/releases/openfoam-v1806/`

[11] J.R. Engel, P.N. Haubenreich, "Temperatures in the MSRE core during Steady State Power Operation", ORNL, November 1962

[12] R.C. Steffy, Jr., "Experimental Dynamic Analysis of the Molten Salt Reactor Experiment", April 1970

[13] H.F.Poppendiek, L.D.Palmer "Forced Convection Heat Transfer between Parallel Plates and in Annui with Volume Heat Source Within the Fluids", ORNL-1701 (May 11, 1954)

[14] `https://www.ansys.com/products/fluids/ansys-cfx/`

[15] S. Bell, et al., "MSRE Design and Operations Report, part iv, Reactor Safety Analysis Report", ORNL, August 1964.

[16] `https://www.oecd-nea.org/dbdata/jeff/`

[17] Sai Chaitanya Tadepalli, Anurag Gupta, K. Umasankari, “Neutronic analysis of MSRE and its study for validation of ARCH code”, Nuclear Engineering and Design 320 (2017) 1–8

[18] B.E.Prince, "Periode measurements on the molten salt reactor experiment during fuel circulation: Theory and experiment", October 1966

ELSEVIER

Annals of Nuclear Energy 00 (2021) 1–22

# Verification of multiphysics coupling techniques for modeling of molten salt reactors

J. Groth-Jensen[a], A. Nalbandyan[a], E. B. Klinkby[a], B. Lauritzen[a], P. Sabbagh[b], A.V. Pedersen[c]

[a]*DTU Physics, Frederiksborgvej 399, 4000 Roskilde, Denmark*
[b]*DAES SA, Avenue des Grandes-Communes, 8, 1213 Petit Lancy, Switzerland*
[c]*Seaborg Technologies, Titangade 11, 2200 Copenhagen, Denmark*

## Abstract

Crucial to the development of Molten Salt Reactor (MSR) designs is the application of multiphysics codes to model the tightly coupled neutronics and thermal-hydraulics behaviour of the liquid fuel. However, the verification and validation of such codes is not a trivial task, in particular for fast reactor designs, where no experimental data are available. In absence of experimental data, a benchmark was developed by LPSC/CNRS-Grenoble for multiphysics codes dedicated to MSR studies. In this study we present two independent multiphysics approaches and apply them to this benchmark. The first approach utilizes the Serpent2 multiphysics interface, allowing for high fidelity coupling of the finite volume computational fluid dynamics code OpenFOAM and Serpent2. In this approach, Serpent2 serves as the neutronics solver and is coupled to an OpenFOAM based thermal-hydraulics solver and supplemented by a delayed neutron precursors transport solver implemented in OpenFOAM. The main advantage of this coupling approach is that it allows for using a high accuracy Monte-Carlo approach to solve the neutron transport equations. The second approach is a novel approach that utilizes the SEALION framework. The SEALION code employs a specialized thermal hydraulics solver based on OpenFOAM, coupled with a custom-made modified point kinetics neutronics solver, that explicitly accounts for the altered neutron importance due to the transport of delayed neutron precursors. The main advantage of this approach is that it allows for a pre-determination of the temperature feedback effects using Monte-Carlo codes, such as Serpent2. Both approaches are verified against results from the benchmark and the overall agreement between the results demonstrates the validity of both approaches.

*Keywords:*
Multiphysics Modeling, Molten Salt Reactors, Verification and Benchmarking, Serpent, OpenFOAM

## 1. Introduction

Molten Salt Reactors (MSRs) have recently gained renewed interest due to potential advantages compared to conventional reactors based on solid fuels; these include inherent safety features, low pressure operation, large negative feedback coefficients and the possibility for online fuel reprocessing. MSR designs, however, are typically characterized by a low technology readiness level and several challenges including the corrosion of the structure materials, developing a fuel cycle and protection against proliferation.

Among several conceptual MSR designs, the Molten Salt Fast Reactor (MSFR) has been selected as a Generation IV Reactor candidate (Boussier et al. (2012)). The MSFR design concept is outlined in Figure 1. The design has evolved over the years from having a cylindrical core to a segmented core geometry in the framework of the European SAMOFAR project (SAMOFAR (2019)). In the segmented design approach the core is toroidal, surrounded by a fertile salt blanket and has sixteen fuel inlets and outlets connected to sixteen heat exchangers. In case of an accident the fuel is supposed to drain into an emergency draining tank which is sealed with a freeze plug during normal operation. The molten fluoride salt enters the toroidal core at a nominal temperature of 650 °C, and the design temperature raise within the core is 100 °C at a nominal flow rate of 4.5 $m^3$/s.

The liquid fuel in the core introduces several fundamental differences compared to a solid fueled reactor. In particular, the delayed neutron precursors (DNP) are no longer stationary, but move with the flow, resulting in a decreased effective delayed neutron fraction. Since the liquid fuel is also the primary coolant, turbulence effects have direct impact on the neutronics performance of the reactor. In addition to this and the application of gaseous fission product removal systems, the

*Email Addresses:* jacgro@dtu.dk (J. Groth-Jensen), asnalb@dtu.dk (A. Nalbandyan), esbe@dtu.dk (E. B. Klinkby), blau@dtu.dk (B. Lauritzen), pascal.sabbagh@daes.pro (P. Sabbagh), avp@seaborg.co (A.V. Pedersen)

| Developer | Neutronics code package | Thermal hydraulics code package | Reference |
|---|---|---|---|
| CNRS | Multi-group diffusion and $SP_N$ | OpenFOAM | (Blanco et al. (2020)) |
| PoliMi | Multi-group diffusion and $SP_3$ | OpenFOAM | (Cervi et al. (2019a), Cervi et al. (2019b)) |
| PSI/EPFL[1] | Multi-group diffusion and $SP_3$ | GenFOAM | (Fiorina et al. (2015)) |
| TUD | Phantom $S_N$ | DGFlows | (Tiberga et al. (2019)) |

Table 1: multiphysics code used in MSFR benchmark (Tiberga et al. (2020)).

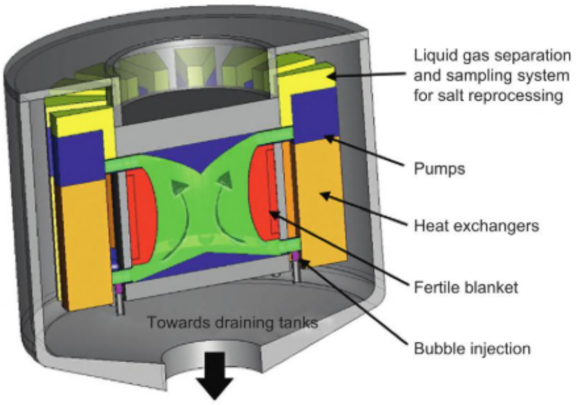


Figure 1: The outline of the MSFR design.

compressibility of the fuel contribute to the strong coupling between neutron kinetics and thermal hydraulics; thus requiring multiphysics modeling tools to study such reactors. Regulatory approved modelling tools have been developed specifically for solid fuel reactors and even though some of them can be extended or modified to address liquid fuels (Fletcher y Schultz (1995), Zhang et al. (2018)), no multiphysics analysis framework has been approved specifically for liquid fuel reactors. A short summary of some multiphysics tools developed and applied for the analysis of MSRs, in particular the MSFR, is presented in Table 1. A brief description of the codes and further references can be found in (Tiberga et al. (2020)).

All codes in Table 1 are based on the solution of the neutron transport equation using either a diffusion or an $SP_X$ approximation (Cervi et al. (2019a), Fiorina et al. (2012), Fiorina et al. (2014))). A new contribution to the original fast MSR benchmark developed by LPSC/CNRS-Grenoble (Aufiero (2015), Laureau (2015), Aufiero y Rubiolo (2018)) has recently been published (Tiberga et al. (2020)). The original benchmark and the recent extension by Tiberga et al. can be used by multiphysics code developers to verify their modeling approaches. The structure of the benchmark, wherein coupling terms are introduced gradually, allows for easy debugging of the codes and identification of the sources for discrepancy. Unless otherwise stated the study here will refer only to the results from the recent publication by Tiberga et al.

For the present benchmark the *CNRS* code employs either a $SP_1$ or a $SP_3$ model for the neutronics combined with a thermal-hydraulics solver based on the finite volume C++ computational fluid dynamics (CFD) toolbox OpenFOAM (Weller et al. (1998)). The *PoliMi* code uses a multi-group diffusion model for the neutronics and OpenFOAM on the thermal-hydraulics side. The *PSI* code implements a multi-group diffusion sub-solver in the OpenFOAM code. Finally the *TUD* code employs an $S_N$ multi-group model for the neutronics and a parallel solver for the incompressible Navier-Stokes equations.

In this paper, we present two multiphysics modeling approaches and verify them against the benchmark. The first approach (Nalbandyan et al. (2019)) utilizes the Serpent2 (Leppänen et al. (2015)) multiphysics interface (Valtavirta (2016)), allowing for high fidelity coupling of OpenFOAM and Serpent2. In this approach, Serpent2 serves as the main neutronics solver and is coupled to an OpenFOAM based thermal-hydraulics solver and a DNP transport solver implemented in OpenFOAM as well. The main advantage of this coupling approach is that it allows for using a high accuracy Monte-Carlo method to solve the neutron transport equations. In the following this approach will be referred to as the DTU approach. The second approach utilizes the SEALION framework. The SEALION framework, in its current iteration, employs a specialized thermal hydraulics solver based on OpenFOAM coupled with a custom-made modified point kinetics neutronics solver explicitly taking the altered neutron importance from DNP transport into account. The main advantage of this approach is that it allows for pre-determining the temperature feedback using Monte-Carlo codes such as Serpent2, leading to a significant reduction in computational requirements. In general point, kinetics solvers are expected to capture the physics of reactor systems, and provide a good approximation for a given reactor transient, as long as the transient is (mainly) driven by the fundamental form. It is therefore of interest to expand and demonstrate point kinetics methods for liquid fueled reactor systems such as MSRs. Despite their reduced order, point kinetics solvers are still being used extensively in nuclear reactor transient analysis codes such as RELAP5/7 (Fletcher y Schultz (1995),Zhang et al. (2018)).

Both modelling approaches are explained in greater details in Section 4.

In Section 2, a detailed description of the benchmark is provided. Section 3 describes the various phases and steps that constitutes the benchmark. The assumptions for every step of the benchmark as well as the output obsevables are described in this section. The two code packages used in paper will be presented in Section 4. The results obtained from applying these codes to the benchmark are presented and analyzed in

Section 5 and 6.

## 2. Description of the benchmark

One of the traditional benchmark cases used to verify thermal-hydraulics solvers for incompressible flow is the lid driven cavity model (Aufiero (2015), Laureau (2015)). The geometry is fairly simple, it captures the main characteristics of the flow and provides a clear identification of relevant parameters for the simulation. The benchmark geometry has been adapted in (Tiberga et al. (2020)) as well as this work for verification of the neutronics-thermal hydraulics coupling techniques for MSRs; the cavity model also represents a simplified Molten Salt Fast Reactor design.

### *2.0.1. Geometrical layout of the benchmark*

The lid driven cavity consists of a 2m by 2m domain filled with a molten salt. The benchmark geometry is shown in Figure 2. As OpenFOAM is unable to deal with purely 2D geometry the cavity is extended along a pseudo $z$-axis by 1m. However, no segmenting of the geometry along the z-axis has been used, in order to preserve the 2D aspects of benchmark, reduce complexity and avoid inherent 3D effects like turbulence. The boundary conditions for the walls perpendicular to the pseudo $z$-axis (front and back faces) are specified as *empty*, which forces the solver to not solve conservation equations for these boundaries. As indicated in Figure 2, two centerlines $AA'$ and $BB'$ have been introduced to facilitate a point-wise comparison of the observables for the various steps of the benchmark.

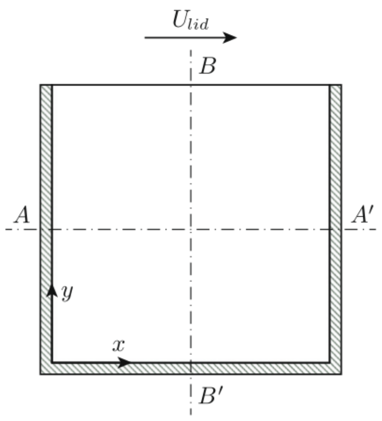


Figure 2: Benchmark geometry: a simple $2m \times 2m \times 1m$ cavity treated as a 2D problem.

### *2.0.2. Input data and boundary conditions: Neutronics*

While the specific choice of nuclear library for the benchmark is JEFF-3.1(A. Koning (2006)), the nuclear data used in this paper is taken from JEFF-3.2(A.J.M. Plompen (2020)). The systematic uncertainties introduced by using a different version of the nuclear data library is addressed in Section 5. The benchmark further prescribes that nuclear data is evaluated at the reference temperature $T_{ref} = 900K$ only. The motivation for this is that the Doppler broadening effect on microscopic cross sections is negligible; this avoids biases arising from the differences in treatment of this complex effect by each code. As a consequence hereof, the neutronic group constants are only affected by the change in density due to expansion of the fuel. The change in fuel density with temperature is implemented according to:

| Isotope | Atomic fraction (%) |
|---|---|
| $^{6}$Li | 2.11488 |
| $^{7}$Li | 26.0836 |
| $^{9}$Be | 14.0992 |
| $^{19}$F | 56.3969 |
| $^{235}$U | 1.30545 |

Table 2: Fuel salt composition.

$$\rho(T) = \rho(T_{ref})[1 - \alpha(T - T_{ref})] \tag{1}$$

where $\alpha$ is thermal expansion coefficient.

The DNPs are binned into eighth groups with the decay constants and fractions shown in Table 3. The parameters are extracted using Serpent2, based on the adjoint Meulekamp's method (Zhong et al. (2011)).

| Group | $\lambda_i$ (s$^{-1}$) | $\beta_i$ |
|---|---|---|
| 1 | $1.24667 \times 10^{-2}$ | $2.42173 \times 10^{-4}$ |
| 2 | $2.82917 \times 10^{-2}$ | $1.20029 \times 10^{-3}$ |
| 3 | $4.25244 \times 10^{-2}$ | $4.97887 \times 10^{-4}$ |
| 4 | $1.33042 \times 10^{-1}$ | $1.36834 \times 10^{-3}$ |
| 5 | $2.92467 \times 10^{-1}$ | $1.85916 \times 10^{-3}$ |
| 6 | $6.66488 \times 10^{-1}$ | $6.19812 \times 10^{-4}$ |
| 7 | 1.63478 | $5.79049 \times 10^{-4}$ |
| 8 | 3.55460 | $1.71090 \times 10^{-4}$ |

Table 3: DNP group-wise data.

The fuel-filled cavity is intended to constitutes a homogeneous, bare reactor and standard vacuum conditions are applied for the neutron flux at each boundary.

Throughout the benchmark the neutron flux is normalized to a fixed reference power $P$. An expectation is final step of the benchmark where the power transfer function is benchmarked.

### *2.0.3. Input data and boundary conditions: Thermal hydraulics*

The notion of a homogeneous bare reactor is reflected in the thermal hydraulics layout of the cavity as well. The thermo-physical data used in the benchmark on the CFD side is shown in Table 4.

The relative high value chosen for the Schmidt number is motivated by the need to limit mixing of DNPs while still keeping the benchmark sensitive to the individual choice of the discretization scheme. For the sake of simplicity the fuel is considered to be incompressible with a laminar flow, and

| Property | Units | Value |
|---|---|---|
| Density | $\text{kgm}^{-3}$ | $2.0 \times 10^3$ |
| Kinematic viscosity | $\text{m}^2\text{s}^{-1}$ | $2.5 \times 10^{-2}$ |
| Volumetric heat capacity | $\text{Jm}^{-3}\text{K}^{-1}$ | $6.15 \times 10^6$ |
| Thermal expansion coefficient | $\text{K}^{-1}$ | $2.0 \times 10^{-4}$ |
| Prandtl number | - | $3.075 \times 10^5$ |
| Schmidt number | - | $2.0 \times 10^8$ |

Table 4: Salt thermodynamic and transport properties.

changes to the density is modeled assuming the Boussinesq approximation - see Equation (1). As such the density variations only affects the buoyancy term. In both the DTU and the SEALION approaches the thermal-hydraulics model assumes an incompressible single-phase flow with the Boussinesq approximation for the mass and momentum balance as well as a conservation of energy. Decay heat is neglected as well. In order to obtain a steady state solution in the presence of a fixed power normalization, cooling is simulated with a volumetric heat sink:

$$q'''(\boldsymbol{r}) = \gamma(T_{ref} - T(\boldsymbol{r})) \qquad (2)$$

where $\gamma$ denotes the volumetric heat transfer coefficient and $T(\boldsymbol{r})$ denotes the temperature at point $\boldsymbol{r}$. $\gamma$ is considered to have a uniform value in the cavity but can vary from one benchmark step to another and is also allowed to vary in time. The benchmark further seeks to mimic the bare reactor by imposing a zero-velocity (i.e no-slip) boundary condition for the fuel at all walls of the cavity expect for the lid, which moves at velocity $U_{lid}$. The boundary condition for the DNP concentration is a homogeneous Neumann. Finally, the cavity is considered to be insulated so adiabatic boundary conditions are applied to all walls.

## 3. Phases and steps of the benchmark

The multiphysics benchmark is performed in three main stages: first the individual solvers for the neutronics and the thermal-hydraulics are tested, then a fully coupled steady state analysis of the model is carried out, and finally the coupling is tested for a forced convection transient scenario.

Hence, the benchmark stages are structured as follows:

- Single physics testing: *Phase 0*
- Steady-state simulation: *Phase 1*
- Time dependent simulation: *Phase 2*

Tables 5 summarizes the input and output observables in each phase of the benchmark.

### *3.1. Phase 0: Single physics benchmark*

Phase 0 of the benchmark is dedicated to testing of the single physics solvers. This means that the input to the given solver is fixed and the single physics solver is run until convergence is obtained. Table 5 summarizes the input and output observables in each step of phase 0.

### *3.1.1. Step 0.1: Velocity field*

In this step the steady-state flow distribution in the cavity is evaluated assuming a fixed velocity of the lid. The aim of this step is to establish a CFD baseline flow which can be used in the next phase of the benchmark where the multi-physics coupling is introduced. The fuel is kept at the reference temperature, reducing the number of equations to be solved.

### *3.1.2. Step 0.2: Neutronics*

Once the flow baseline has been established by the single physics CFD solver the next step is to establish a baseline for the heating source from fission events in the cavity. As such the aim of this step is to verify and provide the fission rate density under the assumptions of a static fuel with a uniform and fixed temperature. The idea of this step is to gauge systematic effects arising from assumptions made in neutronics models as well from the choice of input data. The normalized neutron flux and the eigenvalue of the fundamental mode , namely the multiplication factor $k_{eff}$ are verified in this step.

### *3.1.3. Step 0.3: Temperature*

Once the steady-fluid flow and the heating source has been established, the ability of the thermal-hydraulics solver to find a steady-state solution to the temperature distribution in the cavity is assessed. Conforming with the idea of keeping the tests in the first phase of the benchmark as simple as possible this step keeps both the velocity field and the heating source fixed, thereby reducing the complexity of Navier-Stokes equation significantly and thereby allowing the benchmark to focus on the assessment of the passive scalar transport capability of the solver.

### *3.2. Phase 1: Steady-state coupling*

Based on the steady-state solution obtained phase 0, this phase introduces the coupling gradually though a series of steps. First, the neutronics response to a fixed flow is invistigated. Next, the changes to the temperature field arising from a fixed power profile is studied. The buoyancy effects will be benchmarked and finally the full coupling seek to benchmark the changes to the reactivity as a function of different power levels and cavity lid velocities.

Table 5 summarizes the input and output observables in each step of phase 1.

### *3.2.1. Step 1.1: Circulating fuel*

The objective of this step is to assess the changes to the neutronics caused by a movement of the fuel. For the sake of simplcity and the notion of a gradually introduced coupling both the flow and the temperature distribution are kept fixed. In these settings the main effect on the neutroncs should be a loss of reactivity caused by transportation of the DNPs. As such both the delayed neutron source and the reactivity change is benchmarked in this step.

| Step | Input | Observables |
|---|---|---|
| Phase 0: Single physics benchmark | | |
| 0.1 | • $U_{lid} = 0.5\text{ms}^{-1}$ | • Velocity components along centerlines $AA'$ and $BB'$. |
| 0.2 | • $T = 900\text{K}$<br>• $P = 1\text{GW}$ | • Fission rate density, $\int_E \Sigma_f \Phi dE$ along $AA'$<br>• Reactivity $\rho$ |
| 0.3 | • Fixed flow field from Step 0.1<br>• Fixed heat source distribution from Step 0.2<br>• $\gamma = 1 \times 10^{-6}\text{Wm}^{-3}\text{K}^{-1}$ | • Temperature distribution along centerlines $AA'$ and $BB'$. |
| Phase 1: Steady-state coupling | | |
| 1.1 | • Fixed flow field from Step 0.1<br>• $T = 900\text{K}$<br>• $P = 1\text{GW}$ | • Delayed neutron source, $\sum_i \lambda_i C_i$ along $AA'$ and $BB'$<br>• Reactivity change from Step 0.2, $\rho - \rho_{s_{0.2}}$ |
| 1.2 | • Fixed flow field from Step 0.1<br>• $P = 1\text{GW}$<br>• $\gamma = 1 \times 10^{-6}\text{Wm}^{-3}\text{K}^{-1}$ | • Temperature distribution along $AA'$ and $BB'$<br>• Reactivity change from Step 1.1, $\rho - \rho_{s_{1.1}}$<br>• Change of fission rate density along $AA'$ and $BB'$ with respect to the solution obtained at Step 0.2, $\int_E \Sigma_f \Phi dE - \int_E \Sigma_{f,s_{0.2}} \Phi_{f,s_{0.2}} dE$ |
| 1.3 | • $P = 1\text{GW}$<br>• $U_{lid} = 0\text{ms}^{-1}$<br>• $\gamma = 1 \times 10^{-6}\text{Wm}^{-3}\text{K}^{-1}$ | • Velocity components and temperature distribution along $AA'$ and $BB'$<br>• Delayed neutron source along $AA'$ and $BB'$<br>• Reactivity change from Step 0.2, $\rho - \rho_{s_{0.2}}$ |
| 1.4 | • $\gamma = 1 \times 10^{-6}\text{Wm}^{-3}\text{K}^{-1}$<br>• $P$ variable in the range [0,1]GW with a step of 0.2GW<br>• $U_{lid}$ variable in the range [0,0.5] $\text{ms}^{-1}$, with a step of 0.1 $\text{ms}^{-1}$ | • Reactivity change from Step 0.2, $\rho - \rho_{s_{0.2}}$, as a function of $P$ and $U_{lid}$ |
| Phase 2: Time dependent coupling | | |
| 2.1 | • Steady-state solution from Step 1.4 with $U_{lid} = 0.5$ $\text{ms}^{-1}$ and $P = 1.0$ GW<br>• $\gamma = 1 \times 10^{-6}\text{Wm}^{-3}\text{K}^{-1}$ | • Power gain and phase-shift as a function of the perturbation frequency. |

Table 5: Input and observables of phase 0-2.

### *3.2.2. Step 1.2: Power coupling*

The objective of this step is to the assess reactivity feedback mechanism caused by the changes to the fuel temperature distribution. As the temperature field changes so will the neutron flux distribution in the cavity. By fixing the velocity field and introducing a heat sink a steady-states solution can be obtained under the assumption of the fixed power normalization. Thus by assessing the steady-state temperature distribution and changes to the fission rate density the density feedback effects can be isolated and studied.

### *3.2.3. Step 1.3: Buoyancy*

The objective of this step is to assess the capabilities of the codes to correctly model buoyancy effects caused by the temperature gradients. A natural step before benchmarking the codes in fully coupled scenario is thus to apply the codes to a simplified setting without forced convection. Apart from the steady-state velocity field also the delayed neutron source and resulting change to the reactivity is are investigated.

### 3.2.4. Step 1.4: Full coupling

The objective of this step is to assess the capabilities of the codes to correctly predict the loss of reactivity in a setting that constitutes a realistic reactor simulation. In this step all the effects studied in the previous steps are included, namely the changes to the neutronics caused by the fuel flow from Step 1.1, the reactivity feedback mechanism caused by the changes to the fuel temperature distribution from Step 1.2 and the buoyancy effects from Step 1.3. To mimic the simulation of a reactor under different operational condition, the simulation is repeated for different power normalization levels and cavity lid velocities.

### 3.3. Phase 2: Time dependent coupling

In phase 2 the full coupling developed above is studied in a transient scenario.

### 3.3.1. Step 2.1: Forced convection transient

The benchmark does not include a simple reactivity insertion scenario like a loss of heat sink. Its is argued in (Tiberga et al. (2020)) that such a comparison would be limited to the specific transient and its characteristic time constant. Instead a global perturbation in the frequency domain is applied by introducing a sine wave perturbation of the volumetric heat transfer coeffcient $\gamma$. The response of a system to a sinusoidal input will in general consist of the sum of the steady-state and a transient response.

The sinusoidal perturbation to the heat transfer coefficient is implemented as a 10 % variation around the reference value of $\gamma = 1 \times 10^{-6}\mathrm{Wm}^{-3}\mathrm{K}^{-1}$ with frequencies $f \in [0.0125, 0.025, 0.05, 0.1, 0.2, 0.4, 0.8]$Hz. As a consequence of the negative fuel density feedback coefficient a sinusoidal perturbation to $\gamma$ will lead to a sinusoidal shape of the reactor power, from which the power gain is defined as:

$$G(f) = \frac{\big(P_{max}(f) - P_{avg}(f)\big)/P_{avg}(f)}{\big(\gamma_{max} - \gamma_{avg}\big)/\gamma_{avg}} \tag{3}$$

and the phase-shift can be determined.Here $P_{max,avg}(f)$ denotes the maximum and time-averaged reactor power. $\gamma_{max,avg}$ denotes the frequency independent maximum and time-averaged value of the volumetric heat transfer coeffcient. Table 5 summarizes the input and output observables in each step of phase 2.

## 4. Code packages

### 4.1. Serpent2 to OpenFOAM coupling technique

The coupling scheme is represented in Figure 3. Two distinct software and three distinct solvers are involved: Serpent2 software is used for neutronics part, whereas the DNP transport and the thermal-hydraulics are calculated by the OpenFOAM software. The procedure is as follows (here explained for steady-state mode):

- Serpent2 Criticality Source Simulation (Serpent CSS) is run on the same mesh as used for thermal-hydraulics, using the Serpent2 multiphysics interface capabilities. The heat source map in the domain as well as the total neutron flux, mesh-based DNP distribution, fission cross section and the delayed neutron fractions and decay constants are derived.

- The heat source map is passed on to the thermal-hydraulics solver; in this work the standard OpenFOAM *buoyant-PimpleFoam* solver with *Boussinesq* approximation is used. Fuel velocity and temperature fields are obtained.

- The converged velocity field is passed on to the DNP transport solver; this is a scalar transport solver implemented in OpenFOAM, which, using the neutronics parameters obtained from Serpent2 and the velocity field obtained from OpenFOAM, solves the scalar transport equation for each DNP group:

  $$\frac{\partial C_i}{\partial t} = -\nabla(uC_i) + \nabla(\frac{\nu_t}{Sc'})\nabla C_i - \lambda_i C_i + \beta_i \Phi \nu \Sigma_f \tag{4}$$

  where $C_i$ is the DNP concentration per group, $u$ is the flow velocity, $\nu_t$ is the turbulent kinematic viscosity, *Sc*' is the turbulent Schmidt number, $\lambda_i$ is the decay constant per group, $\beta_i$ is the fractional yield of delayed neutrons to fission neutrons per group, $\Phi$ is the neutron flux and $\Sigma_f$ is the fission cross section.
  The converged DNP distributions are written out for each group.

- The temperature distribution from the thermal-hydraulics solver and the DNP sources from the DNP transport solver are fed back into Serpent and the criticality source simulation routine is repeated.

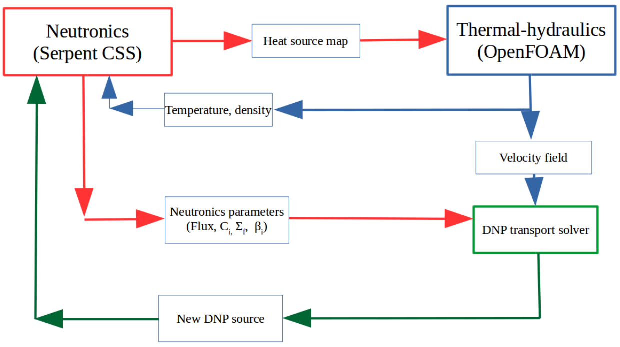


Figure 3: The multiphysics coupling scheme.

This routine is repeated as many times as necessary to ensure that the results have all converged. For the transient scheme, the exchange of coupling terms is performed within each timestep. An advantage of this multiphysics scheme is that it is simple to implement and it also simplifies the debugging process. Another advantage is the flexibility of the concept, e.g. one can choose to use either mesh-wise DNP distributions or, for more complicated geometries, the total group-wise concentration can be used, with proper spatial weighting of DNPs, which will eliminate the necessity to use a mesh with the same number of cells in the neutronics module. Moreover, the fuel velocity field impacts the neutronics module as an external operator, via the updated DNPs source which means that no modifications as such are required in the neutronics solvers. The main disadvantages of the method are the computational time needed for the Monte-Carlo calculations and the need to use small time-steps in the time dependent simulations to limit the errors assigned from the coupling terms.

The approach in the transient phase of the benchmark is essentially the same as described above. This approach is justifiable provided that the thermal-hydraulic time-steps are small enough, during a time step.

*Specifications for benchmark calculations*

A mesh dependence resulted in the 80 × 80 mesh being selected as a baseline for the simulations. As a criteria for checking the mesh independence, the average velocity in the domain is chosen and the results are reported in Figure 4.

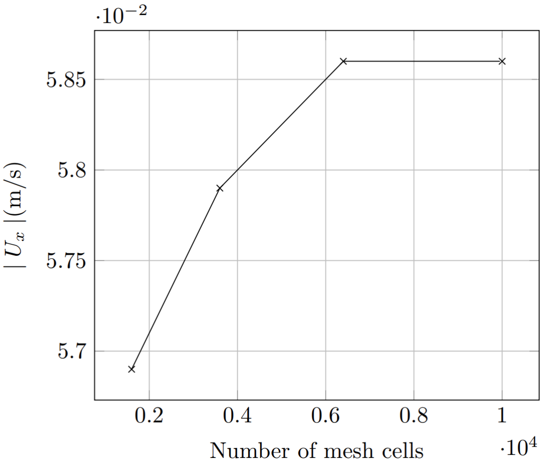


Figure 4: Results from a mesh dependence study

Using the same mesh for both thermal-hydraulics and neutronics simulations allows for direct exchange of variables between the two modules, e.g. the volumetric heat source and the DNP distributions can be generated and used without any modifications related to mesh translations. The OpenFOAM generated mesh is linked to Serpent2, used for the neutronics simulations, by a dedicated multiphysics interface (Tuominen et al. (2016)).

The discretization schemes for the integration of the temporal, diffusive and convective terms are a CrankNicolson, linear Gaussian and a linear limited Gaussian, respectively. The convergence criteria for the coupled solver is influenced by the convergence criteria for the mono-physics solvers individually. For the neutronics solver the number of active and inactive cycles are chosen to be high enough to ensure convergence in the Serpent CSS run. On the thermal-hydraulics a tolerance of $10^{-8}$ to $10^{-4}$ is used in OpenFOAM depending on the quantity is being solved for. For the DNPs the tolerance was set to $10^{-8}$ whereas for all other quantities, the tolerance was set to $10^{-4}$. For single quantities a relative tolerance was set. In steady-state simulations, the codes are iterated until convergence.

### *4.2. SEALION framework description*

In general, point kinetics solvers are expected to capture the physics of a reactor system to a good approximation for a given reactor transient, as long as the transient is mainly driven by the fundamental form. If the fundamental form, and thus the fission power shape, does not change appreciably during the transient, the results obtained from a point kinetics model can be directly convoluted with the shape function of the fission power, thus allowing for extracting the time dependent local power density.

In the case of coupled simulations, this procedure can be carried out iteratively in a suitable coupling scheme. In this sense, for coupled analysis, the point kinetics approximation is expected to be less suited for transients where the fundamental form changes significantly (e.g. simulation of rod ejection) and where higher order methods, e.g., diffusion models, are required. Nevertheless, due to the extensive application of point kinetics solvers in transient reactor analysis, as well as the extensive regulatory experience and confidence with such solvers, it is attractive to examine to what an extend the reduced order physics point kinetics framework can be applied to liquid fueled reactor systems (Valocchi et al. (2020)).

The main challenge in modifying point kinetics to accommodate liquid fueled reactor systems is the change in neutron importance DNPs. In general, in a static fueled system, i.e for most conventional reactors, a neutron resulting from decay of a DNP created at some location will have the same importance as a prompt neutron appearing in the same location, disregarding neutron emission spectra effects. In effect, the equations governing the dynamics can be collapsed into the point kinetics equations for the reactor and DNPs amplitudes appearing as a set of ODEs in time with no reference to spatial variables (see e,g, Diniz et al. (2020)) with DNPs only distinguished by the fact that they appear in a ”delayed” manner and with emission spectrum importance effects captured by the effective delayed neutron fractions. The point kinetics equations can therefore be seen as an effective theory in the sense that information about the spatial distribution of the fundamental form/shape function has been lost, but appears indirectly in terms of the effective kinetics parameters appearing in the resulting equations.

In the case of liquid fueled reactors such an approach is bound to fail. Indeed, suppose that a DNP is created near the center of the reactor core and subsequently moves to the edge of the core before decaying. Such a DNP will clearly appear with decreased importance than had it appeared statically. This effect needs to be accounted for in a satisfactory point kinetics model and clearly spatial information about the DNP concentrations will be needed to compute the importance. In this sense, we have no real hope of constructing a true point kinetics model for a liquid fueled system as spatial information about the DNPs will be needed to predict dynamics. However, given that the delayed neutron fraction is small the bulk part of neutron multiplicity is driven by prompt neutrons; the introduction of moving DNPs is not expected to change the fundamental and adjoint form of the system significantly. This expectation is also supported by theoretical arguments presented in (Ash (1979)). It is therefore reasonable to expect that a modified point kinetics approximation can still be employed in the sense that the equations retain the usual form, but the DNP contribution acquires the equation for the reactor amplitude from a non-trivial spatial weighting over the DNP concentrations - see below for details.

In particular, the reactivity appearing in the point kinetics equations is related to the *k*-eigenvalue and can, for a given temperature distribution, be extracted using Monte-Carlo methods. As it is undesirable to perform a time consuming full *k*-code calculation in each time step of a coupled calculation, the SEALION framework employs a linear separable reactivity model on a pre-defined neutronics mesh for evaluating the reactivity during the coupled simulation. The input parameters of this model are individual reactivity responses for the neutronics mesh cells under temperature perturbations which are evaluated from a Monte-Carlo *k*-code simulation, here using Serpent2. Using this model, we can derive the reactivity response of a given hydrodynamic configuration from temperature data on the neutronics mesh. In particular, the model takes local temperature effects into account expected to be of high importance for liquid fueled systems.

Finally, since the fundamental form is not expected to change significantly, the shape function and the fission power deposition shape are taken to be fixed parameters in the current model with both quantities extracted from a single high precision *k*-code calculation. The shape function is used to weight the DNP contribution while the fission power shape is used to compute the local deposited heat during each time step of the coupled simulation.

The PK module is the first of three central SEALION modules. The three modules cover:

- **PK module**: This neutronics module is based on the modified point-kinetics approach and is capable of describing the evolution of neutronics parameters, calculating temperature and density feedback coefficients and power evolution. It is also capable of calculating mesh-based delayed neutron precursor transport and concentrations.
- **nCFD module**: An in-house developed solver based on the open source platform OpenFOAM with built-in capabilities of DNP, coolant, fuel and heat transport models.
- **Master iterator (MI) module**: Integrates and couples the PK and nCFD modules. The MI modules handles dynamic time stepping, geometric mesh mapping, operational state specification (steady-state and transient).

In the coupled tool, the PK module will receive temperature and DNP distributions from the nCFD module. Using a reactivity map produced prior to the coupled calculations with a Monte Carlo neutronics code, it will then calculate, among other things, the distribution and evolution of power and neutron flux, which will be read and analysed to produce an updated set of input parameters for the nCFD module. This data analysis and exchange will be controlled by the MI module. The full simulation process flowchart is shown in Figure 5.

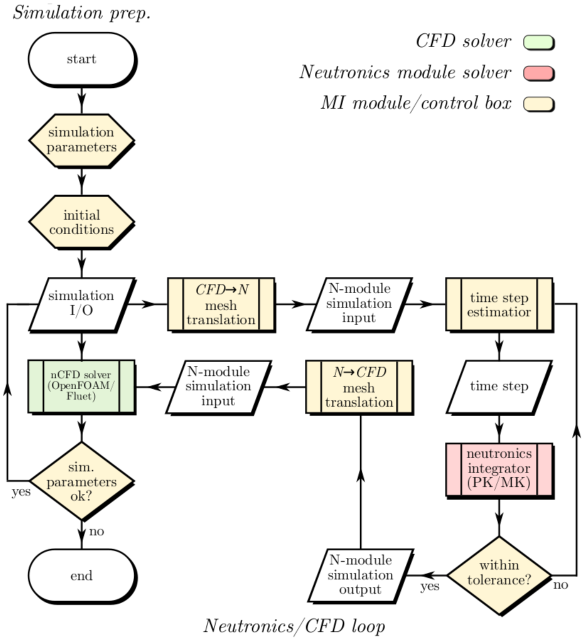


Figure 5: Full simulation flowchart.

The software modules mentioned above are all envisioned to undergo a qualification. However at the current stage only nCFD module is foreseen to undergo that process in the near future.

The starting point for the SEALION computational framework are the general time-dependent point kinetics equations. Expressed in terms of reactor power $P$ and DNP concentration $C_I$ in a given group $I$ the point kinetics equations looks like:

$$\dot{P} = \left(\frac{\rho - \beta}{\Lambda_{(p)}}\right) P + \sum_{I=1}^{N_{DNP}} \lambda_I C_I \tag{5}$$

$$\dot{C}_I = \left(\frac{\beta_I}{\Lambda_{(p)}}\right) P - \lambda_I C_I \tag{6}$$

where $N_{DNP}$ denotes the number of DNP groups which in the current study is set to eight. $\Lambda_{(p)}$ denotes the prompt neutron generation time, $\beta_I$ the relative fraction of delayed neutrons from DNP group $I, \beta = \sum_{I=1}^{N} \beta_I$ is the total delayed neutron fraction and $\lambda_I$ the decay constant of DNP group $I$. Finally, $\rho$ (the reactivity) represents the branching of the fission chains and the amount of multiplication in the system.

In order to incorporate neutron importance effects, the DNP coupling term is convoluted with the adjoint flux, which in turn is approximated by the flux, consistent with a simple one-group approximation, and finally recasted on a mesh:

$$|\dot{\mathbb{P}}\rangle = \left(\frac{\rho-\beta}{\Lambda_{(p)}}\right)|\mathbb{P}\rangle + \sum_{I=1}^{N_{DNP}}(\lambda_I \mathcal{P}\otimes \mathrm{Vol}\psi)|\mathbb{C}_I\rangle \quad (7)$$

$$|\dot{\mathbb{C}}_I\rangle = \left(\frac{\beta_I}{\Lambda_{(p)}}\right)|\mathbb{P}\rangle - \lambda_I|\mathbb{C}_I\rangle \quad (8)$$

Here $|\mathbb{P}\rangle$ and $|\mathbb{C}_I\rangle$ denote the mesh state vectors for the power density and the DNP concentration. Vol $\psi$ denotes an inner product between two vectors Vol and $\psi$ that contains the volumes and neutron flux of each individual mesh cell. $\mathcal{P}$ denotes the power shape unit normalized in a $L_2$ sense.

The deterministic approaches for computing the point-kinetics parameters with the explicit solution for the adjoint flux is considered a difficult task in continuous energy Monte Carlo codes. For these reasons, other approaches (that do not require explicit computation of the adjoint flux) are often employed. In the SEALION framework , the iterated fission probability (IFP) approach within Serpent2 is used to extract values for the point-kinetics parameters (Leppänen et al. (2014)).

The SEALION code relies on data exchange between the CFD and neutronics modules, which in general means employing different meshes. The neutronics mesh is expected to be somewhat coarser that the mesh employed by the CFD software for computational reasons. In order to handle data exchange between the SEALION modules, data on one mesh therefore needs to be translated to the other mesh and vice versa. In general, mesh translation from a fine to a coarse mesh is performed by a clustering algorithm (weighted average) while mesh translation from a coarse to a fine mesh will require an interpolation algorithm. A dedicated module has been developed to perform mesh translations in a systematic way in order to ensure consistency between various meshes used by the SEALION code. The module is based on the OpenFOAM polyMesh format as well as the python packages `NumPy`(Oliphant (2006)) and `SciPy`(Eric Jones (2001)). The first python package is used for the manipulation of multidimensional arrays and the latter for interpolation and clustering algorithms.

*Specifications for benchmark calculations*

The mesh for computing the reactivity response is a coarse structured uniform mesh containing 10 × 10 cells, while the mesh used for the neutronics solver is a finer mesh containing 20 × 20 cells, likewise structured and uniform. The mesh used for the CFD solver is a structured non-uniform mesh containing 200 × 200 cells. Mesh convergence was ensured, similarly to the DTU coupling approach (Figure 4).

The input data needed to the modified point kinetics equations are extracted using Serpent2.1.31 (Leppänen et al. (2014)). The power density is likewise extracted using Serpent2 and translated into mesh state vectors for the power density $|\mathbb{P}\rangle$ using the dedicated mesh translation module.

On the thermo-hydraulics side the SEALION code employs an Euler, corrected linear Gaussian and an upwind Gaussian discretization scheme for the integration of the temporal, diffusive and convective terms, respectively.

The convergence criteria for the coupled solver is influenced by the convergence criteria for the mono-physics solvers individually. The neutronics solver uses a `SciPy` implementation of explicit Runge-Kutta method of order 5(4) (Dormand y Prince (1980)). The convergence is therefore controlled assuming accuracy of the fourth-order method, but steps are taken using the fifth-order accurate formula. On the thermal-hydraulics a tolerance of $10^{-8}$ to $10^{-6}$ was used in OpenFOAM, depending on the quantity is being solved for. For the hydro-static pressure the tolerance was set to $10^{-8}$ whereas for all other quantities, the tolerance was set to $10^{-6}$ . For single quantities a relative tolerance can also be set. In steady-state simulations, the codes are iterated until convergence.

Transient calculations were performed taking a time step equal to 1/200 of the perturbation period.

## 5. Benchmark results

In this section the main results are presented and compared to the benchmark analysis (Tiberga et al. (2020)). For the sake of transparency the step numbers in the figures and tables refer to the step numbering used in the benchmark analysis.

### 5.0.1. Assessing the results

Given the nature of the benchmark no reference solution exists, however, as a measure of any discrepancy of any individual code to the average result, we employ a $\chi^2$ metric,

$$\epsilon_C = \sqrt{\frac{\sum_{i=1}^{N_p}\left(Q_c(\boldsymbol{r}_i) - Q_{avg}(\boldsymbol{r}_i)\right)^2}{\sum_{i=1}^{N_p} Q_{avg}^2(\boldsymbol{r}_i)}} \quad (9)$$

where $\boldsymbol{r}_i$ denotes a point on the sample lines, $N_p = 201$ is the number of points used in sampling of the quantity $Q_C$ for a given code $C$. The average value $Q_{avg}(\boldsymbol{r}_i) = \frac{1}{N_C}\sum_{c=1}^{N_C} Q_c(\boldsymbol{r}_i)$, however excludes the code under scrutiny. A collective measure for the full set of codes is taken to be the averaged value of $\epsilon_C$ : $\epsilon = \frac{1}{N_C}\sum_{C=1}^{N_C}\epsilon_C$.

### 5.1. Phase 0: Single physics benchmark

### 5.1.1. Step 0.1: Velocity field

The first test of the single physics solvers is the steady-state velocity field of the fuel given a fixed lid velocity: $U_{lid} = 0.5ms^{-1}$ .

In Figure 6, the horizontal velocity component along the cavity vertical centerline is shown, providing a good agreement between the results of this work and the benchmark participants. The code-wise discrepancy $\epsilon_C(DTU)$ varies between 0.4% and 1.3% along $AA'$ centerline and between 0.3% and 0.8% along $BB'$ centerline. For the SEALION code the corresponding numbers are $\epsilon_C(SEALION) \in [0.4\%, 0.8\&]$ and $\epsilon_C(SEALION) \in [0.3\%, 0.7\%]$. Respectively, these numbers are to be compared to an average relative variation of 0.35%

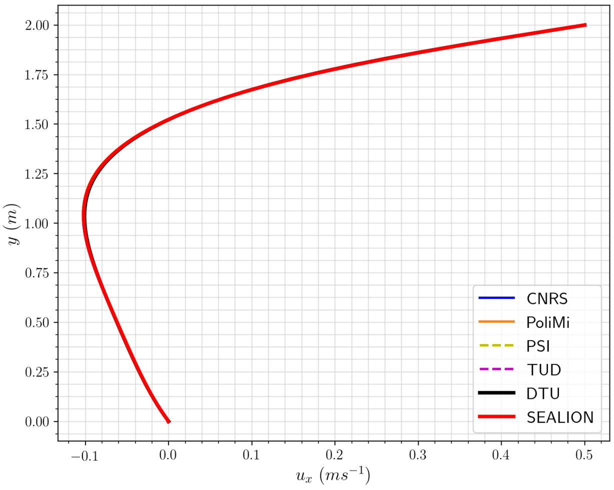


Figure 6: Step 0.1 - Horizontal velocity component along *BB*′.

for velocity profiles along $AA'$ and 0.8% for the $BB'$ profiles for the benchmark participants. A statistical test based on the measure $\epsilon_C$ yields that both the DTU and SEALION code is statistically compatible with the rest of the benchmark participants (p-value ≈ 1). A Kolmogorov-Smirnov test yields the same result. The zero velocity of both codes on the boundaries reflects the no-slip condition imposed on all boundaries expect the lid which moves at and 0.5ms$^{-1}$ (Ghia et al. (1982),Botella y Peyret (1998)).

The velocity values used for the plots are provided in Table in A.11 in Appendix A and are compared for different benchmark steps in Figure B.13 in Appendix B.

### 5.1.2. Step 0.2: Neutronics

The objective of Step 0.2 is to establish a baseline for the heating source from fission events in the cavity and thereby provide a test of the neutronics module. Figure 7 shows the fission rate density and Table 6 shows the excess reactivity.

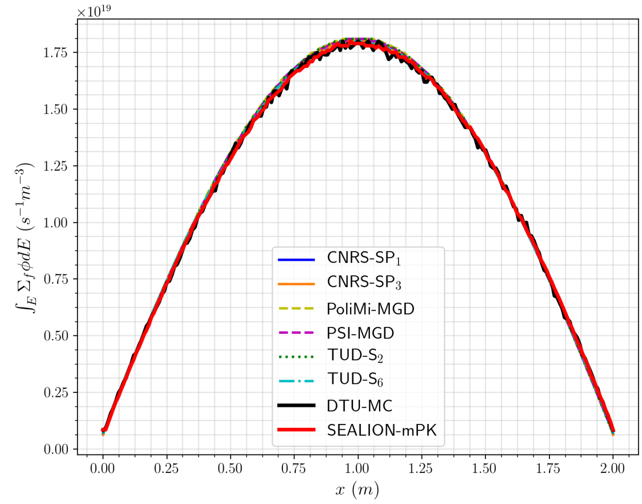


Figure 7: Step 0.2 - Fission rate density along $AA'$

A good agreement with the benchmark is observed both for the SEALION and for the DTU codes. The code-wise discrepancy is $\epsilon_C(DTU) = 1.6\%$ and $\epsilon_C(SEALION) = 1.3\%$ which is at the level of the statistical uncertainty in the individual sample bins. These numbers are to be compared to a code-wise discrepancy for benchmark participants in the range of 0.2%-0.6% and an average relative deviation below 0.4%. At this stage both codes relies on Serpent2 solely to extract the fission rate density. As such it is expected that the two codes give similar results. To ensure that the fission rate density is tallied in a region similar to the deterministic codes a relative small detector volume used. As a consequence hereof relatively large fluctuations can be observed in the results for both the DTU and SEALION code. It should be emphasized that the statistical fluctuations present in Figure 7 will not propagate into the rest of the benchmark as the input to the neutronics solvers relies on the fission rate density extracted on the full volume and not the subset used for tallying. Also note here that both the DTU and SEALION results gives significantly higher values at the boundaries ($x = 0$m and $x = 2$m) when compared to the average of the results from the other codes. This is expected due to the difference in the implementation of the vacuum boundary conditions between diffusion and higher order transport discretizations. As can be seen from Table A.12 in Appendix A, DTU and SEALION results show a better agreement with the higher order codes as expected due to the multi-group and higher order nature of Monte-Carlo codes. A statistical test similar to the one performed in Step 0.1 implies that the hypothesis that the fission rate density distribution from the DTU and SEALION codes stems from the same distribution as the benchmark participants can not be rejected.

Table 6 gives the system excess reactivity for the various codes. The DTU and SEALION codes both yield significantly larger excess reactivity than the other codes. The explanation for the discrepancy is twofold. First, the more advanced transport models are expected to give different results from the diffusion codes. In comparison the DTU and SEALION codes differ a from the average of the other participants by 300 pcm. Another factor contributing to the discrepancy is the different choice of nuclear libraries. The difference in excess reactivity when evaluated with Serpent2 using JEFF3.1 vs JEFF3.2 is 386 pcm which is consistent with the difference observed between the benchmark participants and the DTU and SEALION codes. Further studies have been carried out to estimate the systematic uncertainties arising from the specific choice of nuclear library. Differences in the fission rate density from the different choice of nuclear library are locally below 0.2% and thereby compatible with the code-wise and average relative discrepancy for benchmark participants. The study of systematic uncertainties have also be extended to include some of the later steps in the benchmark. Here the uncertainties from a particular choice of nuclear library fall below the code-wise and average relative discrepancy reported by the benchmark participants. This picture is consistent with the notion that discrepancies between the benchmark participants reported in Table 6 does not propagate to the rest of the benchmark.

| Code | $\rho$ (pcm) |
|---|---|
| DTU | 740 ± 6 |
| SEALION | 741.2 ± 2.5 |
| CNRS-SP$_1$ | 411.3 |
| CNRS-SP$_3$ | 353.7 |
| PoliMi | 421.2 |
| PSI | 411.7 |
| TUD-S$_2$ | 482.6 |
| TUD-S$_6$ | 578.1 |

Table 6: Step 0.2 - System excess reactivity.

| Code | $\rho - \rho_{s_{0.2}}$ (pcm) |
|---|---|
| DTU | -60 ±13 |
| SEALION | -57.6 ± 8.6 |
| CNRS-SP$_1$ | -62.5 |
| CNRS-SP$_3$ | -62.5 |
| PoliMi | -62.0 |
| PSI | -63.0 |
| TUD-S$_2$ | -62.0 |
| TUD-S$_6$ | -62.7 |

Table 7: Step 1.1 - Reactivity change from Step 0.2.

### *5.1.3. Step 0.3: Temperature*

In this step the fixed velocity field from Step 0.1 and the fixed volumetric heat map from Step 0.2 are used to test the ability of the codes to predict the temperature of the fuel. The temperature profile along $BB'$ is shown in Figure 8. The code-wise discrepancy is $\epsilon_C(DTU) \in [0.2\%, 0.4\%]$ and $\epsilon_C(SEALION) \in [0.2\%, 0.6\%]$ along $AA'$ and $BB'$ centerlines. These numbers are to be compared to an average relative discrepancy below 0.1%. A good agreement between this work and the benchmark participants results is observed. It should be noted that the results from the DTU and SEALION codes presents slight deviations from the average of the results from the other participants. However, the deviations are statistically insignificant.

The values for temperature along the two centerlines are given in Table A.13 in Appendix A. Figure B.14 in Appendix B shows the temperature distributions for various benchmark steps.

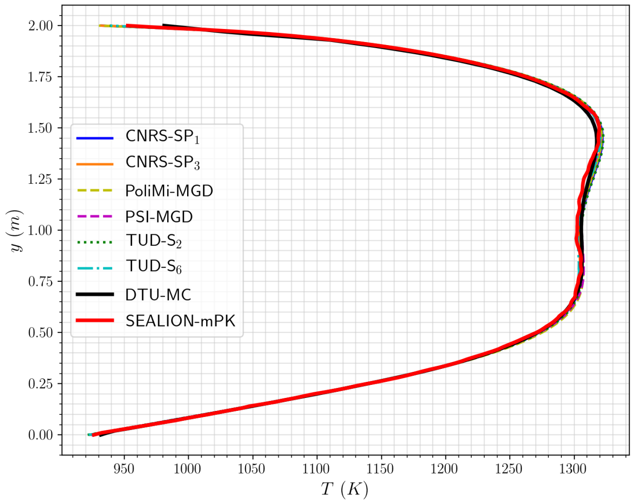


Figure 8: Step 0.3 - Temperature along $BB'$.

### *5.2. Phase 1: Steady-state coupling*

### *5.2.1. Step 1.1: Circulating fuel*

As mentioned in Section 3.2.1 the goal of this step is to investigate the reactivity loss due to the transport of delayed neutron precursors. In order to isolate the effect a simplified version of the coupling is introduced where both the flow field and the temperature distribution are kept fixed. Figure 9 shows the total delayed neutron source along the vertical centerline. The agreement with the benchmark results is satisfactory, with a code-wise discrepancy of $\epsilon_C(DTU) \in [2.4\%, 2.8\%]$ and $\epsilon_C(SEALION) \in [1.2\%, 3.1\%]$ along the $AA'$ and $BB'$ centerlines. These numbers are to be compared to a code-wise discrepancy for benchmark participants of 0.2%-0.7% and an average relative discrepancy below 0.3%. The raw values for the delayed neutron precursor source are provided in Table A.14 in Appendix A. Figure B.15 and B.16 in Appendix B show the delayed neutron precursor distributions for various benchmark steps for the first and fifth family.

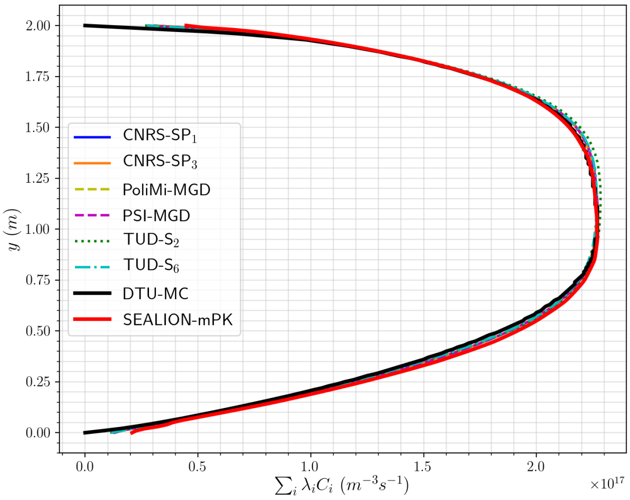


Figure 9: Step 1.1 - Delayed neutron source along $BB'$.

Table 7 summarizes the reactivity loss with respect to the static case presented in Step 0.2. The results from the DTU and SEALION codes both differ from the average of the other participants by 2-7 pcm. The objective of this step in the benchmark is to assess the capabilities of the codes to correctly model the reactivity loss due to the transport of delayed neutron precursors. A discrepancy in the either the fission rate density or the steady state flow field will propagate into the results for the reactivity loss/gain. However, the results are compatible with the benchmark participants within the uncertainties.

### 5.2.2. Step 1.2: Power coupling

The impact of a non-uniform temperature on the flux and in turn the effect of the flux deformation on the temperature is evaluated. The temperature impact on the reactivity is assessed via the fuel density feedback.

The left plot in Figure 10 shows the temperature profile along the horizontal centerline. A good agreement with the benchmark results is noted, with DTU and SEALION results yielding a code-wise discrepancy below 0.4% for both codes, which is comparable to the average relative discrepancy below 0.1% reported by the benchmark participants. The right plot in Figure 10 shows the fission rate density change compared to the standalone neutronics test in Step 0.2 with static fuel. Due to the nature of the Monte-Carlo based neutronics approach in the DTU code it is expected that the change in fission rate density, which is more than an order of magnitude smaller than the fission rate density itself, is to have large statistical fluctuations. The same is true for the SEALION code which cannot solve for the change in fission rate density directly but infers the results from an additional Monte-Carlo simulation. Error-bands representing the two standard deviation limit have been imposed on the figure to illustrate the magnitude of the statistical uncertainties. The overall agreement between the DTU/SEALION results and the benchmark participant results is sub-optimal yet compatible within the statistical uncertainties. The code-wise discrepancy are in both cases larger than the average discrepancy of 1%-2% reported by the benchmark participants. The overall shape of the change in fission rate is is largely reproduced by both the DTU and the SEALION code. This means that the feedback mechanism is captured correctly; increasing the temperature causes the power to decrease via a negative density feedback coefficient. This claim is supported by the results from Section 5.1.3 and 5.2.1 where it was demonstrated that both the temperature distribution and delayed neutron source agree with the benchmark results.

Computing the change in fission rate density is a stress test of the codes; and testing the capability of the codes to model higher order effects, which are more than a magnitude smaller than the fission rate density itself. The SEALION code does not take small changes to the fission cross section due to local temperature changes, into account. The SEALION code therefore assumes that the fundamental form does not change during a simulation and the code is as such not expected to reproduce higher order effects. In the case of the DTU code the quality of the results are limited by a compromise between computational resources and the desired precision. An intermediate conclusion would be that both the DTU and the SEALION code falls short in describing the the higher order perturbation to the fission rate density caused by the shift of DNPs from a region of high neutron importance in the center of the cavity to the region of lower neutron importance away from the center. It is thus expected that higher order deterministic codes perform better in describing high precision results such as the change in fission rate density. Table A.15 in Appendix A shows the temperature and change in fission rate density for the various codes.

Table 8 gives the reactivity changes compared to the first step of the steady-state phase. The results from the DTU and SEALION codes are compatible with the results from the benchmark participants within 2-3 standard deviations.

| Code | $\rho - \rho_{s_{1.1}}$ (pcm) |
|---|---|
| DTU | - 1094.0±103 |
| SEALION | -1217.5 ± 30 |
| CNRS-SP$_1$ | -1152.0 |
| CNRS-SP$_3$ | -1152.7 |
| PoliMi | -1161.0 |
| PSI | -1154.8 |
| TUD-S$_2$ | -1145.2 |
| TUD-S$_6$ | -1122.0 |

Table 8: Step 1.2 - Reactivity change from Step 1.1.

All values used in plots for this step can be found in Table A.12 in Appendix A.

### 5.2.3. Step 1.3: Buoyancy

Buoyancy effects are considered, and the velocity and temperature fields are assessed along with the reactivity changes due to DNP transport under the assumption of natural convection only. In Figure 11 the vertical component of the velocity and the temperature are shown. In terms of velocity, the agreement is fair yielding a code-wise discrepancy of $\epsilon_C(DTU) \in [2.5\%, 5.2\%]$ along the $AA'$ centerline and $\epsilon_C(DTU) = 4.2\%$ along the $BB'$ centerline. For the SEALION code the corresponding numbers are $\epsilon_C(SEALION) \in [2.1\%, 5.3\%]$ and $\epsilon_C(SEALION) = 3.4\%$. These numbers are to be compared to an code-wise discrepancy for benchmark participants of 0.2%-1.8% and an average relative discrepancy below 0.6% The $x$-component of the velocity field along the $BB'$ centerline is omitted from the benchmark due the symmetry of the solution as can be seen from Figure B.13. In Figure 11 also the temperature profile along the horizontal centerline is shown. Here the agreement between the results is very good with code-wise discrepancies below 0.4% for both codes. The slight overshoot of the DTU code of $u_y$ along the horizontal centerline can be explained by the slight overshoot of the temperature profile along the same centerline. In Figure 11, the delayed neutron precursor source is shown along the horizontal centerline. The agreement between the participants is good; the benchmark participants report 0.5 % discrepancies between the results, DTU results show some deviation close to the center which can be attributed to a non-converged solution. In this step the SEALION results agrees slightly better with the benchmark participants yielding code-wise discrepancies of no larger than a few procent.

Table 9 reports the change in reactivity compared to the standalone neutronics test. There is around 50 pcm of difference reported between the benchmark participants. The estimated reactivity loss due to buoyancy effects for both the DTU and SEALION codes are statistically compatible with the results from the benchmark participants.

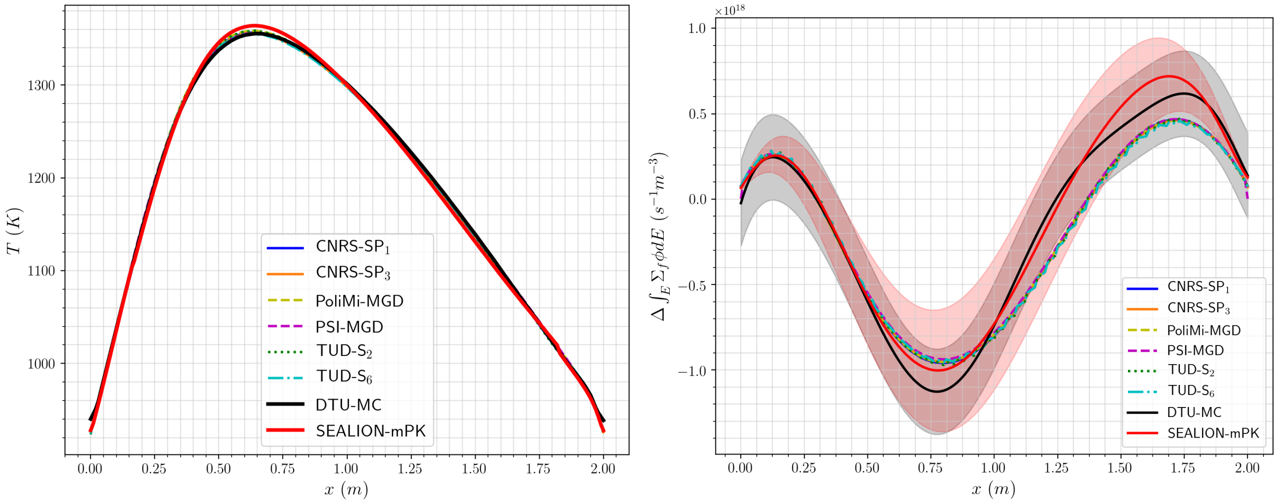


Figure 10: Step 1.2 - Temperature and change of fission rate density along $AA\prime$. The colored band around the curves show the statistical uncertainty.

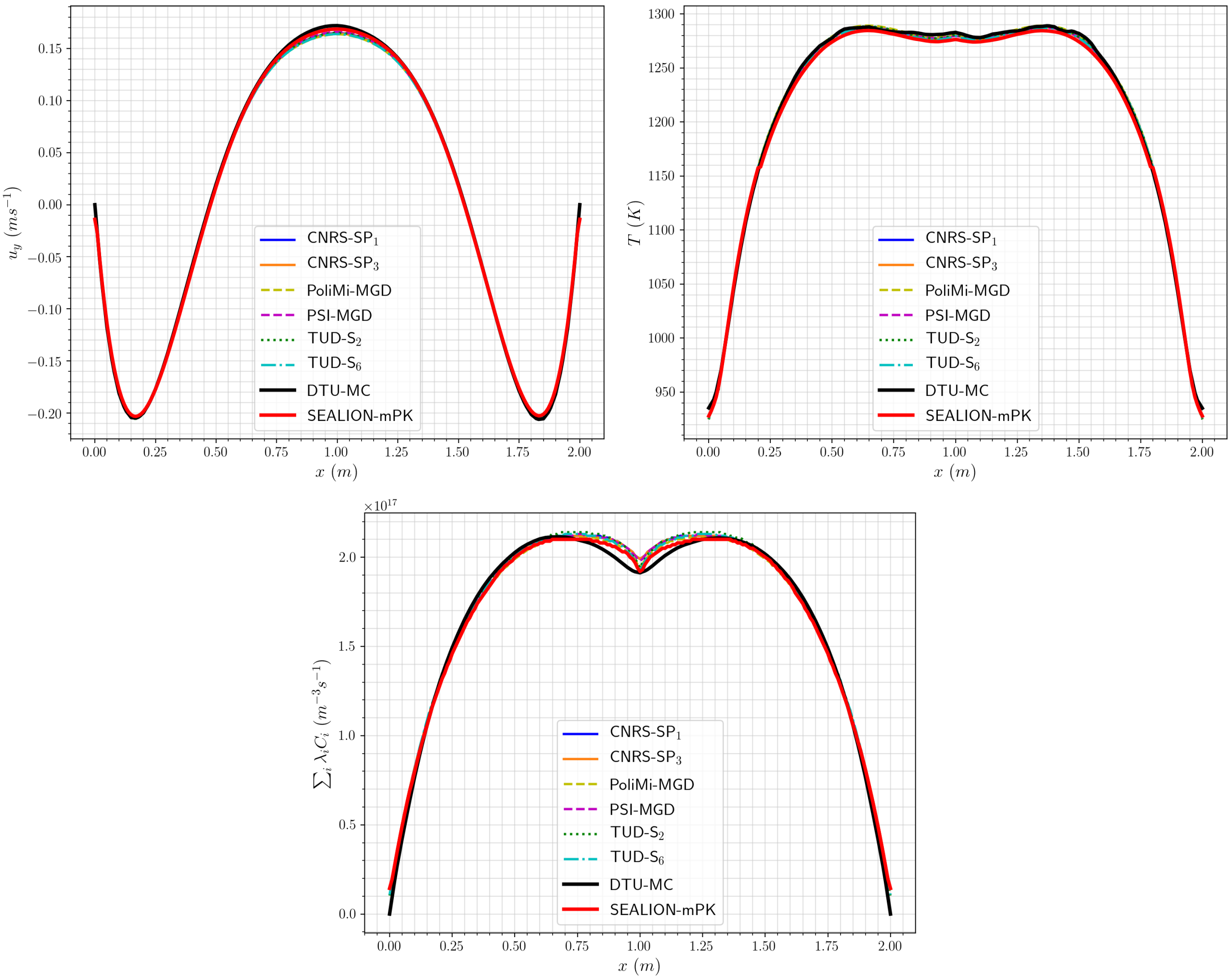


Figure 11: Step 1.3 - Vertical velocity component, temperature, and delayed neutron source along $AA'$.

All temperature, velocity and delayed neutron source values from this step are shown in Table A.13 and Table A.14 in Appendix A.

### *5.2.4. Step 1.4: Full coupling*

The objective of this step is to assess the capabilities of the codes to predict the loss of reactivity in a setting that con-

| Code | $\rho - \rho_{s_{0.2}}$ (pcm) |
|---|---|
| DTU | -1249±25 |
| SEALION | -1218.2 ± 22 |
| CNRS-SP$_1$ | -1220.5 |
| CNRS-SP$_3$ | -1220.5 |
| PoliMi | -1227.0 |
| PSI | -1219.6 |
| TUD-S$_2$ | -1208.5 |
| TUD-S$_6$ | -1184.4 |

Table 9: Step 1.3 - Reactivity change from Step 0.2.

stitutes a realistic reactor simulation under different operating conditions. The different operating conditions are emulated at this stage by increasing the velocity of the lid $U_{lid}$ from 0 to 1ms$^{-1}$ in steps of 0.1ms$^{-1}$ and the power from 0 to 1GW with steps of 0.2GW . The impact on the system reactivity is investigated. The change in reactivity as a function of power is provided in Table 10 for cavity lid velocities $U_{lid}$=0ms$^{-1}$, 0.3ms$^{-1}$ and $U_{lid}$=0.5ms$^{-1}$.

When inspecting the reactivity differences $\Delta\rho_{s_{0.2}} = \rho - \rho_{s_{0.2}}$ in Table 10, two interesting phenomena appear. In general, an increase in power level leads to an increase in the temperature of the system, causing fuel salt expansion and increased buoyancy. The increased buoyancy effectively drives the DNPs away from the cavity center onto regions of lower neutron importance, lowering the reactivity and leading to larger negative value of $\Delta\rho_{s_{0.2}}$ for higher power levels. The same effect in the reactivity caused by a shift of the DNPs to regions of lower importance is observed at lower power levels (column 1) when the forced convection is increased by changing the speed of the cavity lid from 0ms$^{-1}$ to 0.5 ms$^{-1}$. At higher power levels, starting from P=0.6 GW, higher velocities result in slightly less negative reactivity. This is due to two competing processes: a high velocity shifts the DNPs away from the cavity center, whereas a high power reduces the temperature in the regions of higher importance. Depending which effect is stronger, the negative reactivity level will increase or decrease (Tiberga et al. (2020)).

In terms of reactivity, the code-wise discrepancies for the DTU code are $\epsilon_C(DTU) \in [7\%, 9\%]$ and $\epsilon_C(SEALION) \in [4\%, 8\%]$ for SEALION, cf. Table 10. These numbers are to be compared with an average relative discrepancy for benchmark participants of 1%-13%. The discrepancy between the SEALION code and the benchmark participants, can be ascribed to the fact that the shape of the power distribution is assumed to remain unchanged, thereby ignoring second order perturbations to the fundamental form. The relative importance of the second order effects to the fundamental form increases at lover power levels for higher levels of forced convection. This explains the slight difficulties of the SEALION code to model the increase in $|\Delta\rho_{s_{0.2}}|$ for increasing $U_{lid}$. The results from the DTU replicate the trend from the benchmark participants. This difference can be explained by the slight difference in the in velocity , fission density and delayed neutron source as discussed in Sections 5.2.1 - 5.2.3.

## 6. Phase 2: Time dependent coupling

The left panel in Figure 12 shows the power gain as a function of frequency. The right panel in Figure 12 compares the results for the power phase shift compared to the $\gamma$ wave. A discrepancy for the SEALION code of $\epsilon_C(SEALION) = 4.6\%$ is observed for the gain diagrams. These numbers are to be compared to a code-wise and average relative discrepancy for benchmark participants of less than 1%. For the phase diagrams the agreement is better yielding a code-wise discrepancy for the SEALION code of $\epsilon_C(SEALION) = 8.2\%$. However, these numbers are to be compared to a code-wise discrepancy for the benchmark participants of 1.3%-3.2% and an average relative discrepancy below 2.2%.

As for DTU, due to the necessity to do very fine power change sampling and the computational costs related to that, at this stage only one frequency is compared; 0.025 Hz. For this frequency, the gain computed by coupled Serpent2-OpenFOAM approach is 0.713 and the phase shift is 27 ° as compared to the average of 0.835 and 18.6 ° obtained by the benchmark participants. This relatively large discrepancies are mainly attributed to the sampling time-step used in Serpent2-OpenFOAM coupling approach to extract the data points for the power change and is expected to become more pronounced at higher frequencies. However, it can be appreciated that the coupling technique is capable to capture reasonably well the physical phenomena related to the non-uniform salt cooling in a frequency range where the DNPs an equilibrium at each instant. This is expected since the period for the slow transients are comparable to the half-lives of the longest lived DNP groups giving a gain of ∼ 1 and small phase-shift. The SEALION results agrees fairly well with the benchmarks participants meaning that the SEALION code to a large extend is able to replicate the sinusoidal power trend induced by the negative density feedback coefficient from the slow transient to the high frequencies domain where the kinetics are govern by short lived DNPs. The SEALION code overshoots the phase shift slightly which can be explained by the fact that due to the assumption on the fundamental form in the modified point-kinetics solver the effects of DNPs movement and other second order effects are underestimated in this approach. These effects are expected to only influence the shape of the power transfer function and not the reduction in the system response amplitude which can also be observed since the SEALION code describes the gain well over a large frequency range.

## 7. Conclusion

This paper presents the results of a code-to-code verification of two multiphysics coupling techniques for molten salt reactor modeling. The coupled Serpent2-OpenFOAM method and an in-house developed SEALION software are compared for several multiphysics software features in a benchmark defined for a generic fast MSR (Tiberga et al. (2020)).

The standalone physics modules yield almost identical results with the benchmark participants. Gradual introduction of coupling terms shows that both coupling schemes well

| Code | | $\rho - \rho_{s_{0.2}}$ (pcm) | | | | |
|---|---|---|---|---|---|---|
| | | $P$ (GW) | | | | |
| | $U_{lid}(ms^{-1})$ | 0.2 | 0.4 | 0.6 | 0.8 | 1.0 |
| DTU | 0.0 | -206 ± 28 | -440 ± 30 | -728 ± 42 | -913 ± 27 | -1118 ± 29 |
| SEALION | | -269.6 ± 22 | –523.3 ± 29 | -770.1 ± 27 | -1013.2 ± 24 | -1253.71 ± 26 |
| CNRS-SP$_1$ | | -264.5 | -503.4 | -737.8 | -974.2 | -1220.5 |
| CNRS-SP$_3$ | | -265.8 | -503.4 | -738.5 | -976.2 | -1220.7 |
| PoliMi | | -266.0 | -498 | -734 | -975 | -1227 |
| PSI | | -268.0 | -504.8 | -739 | -975 | -1215.1 |
| TUD-S$_2$ | | -263.7 | -498.1 | -731.1 | -967.2 | -1208.5 |
| TUD-S$_6$ | | -258.0 | -487.8 | -716.3 | -947.9 | -1184.4 |
| DTU | 0.3 | -265 ± 28 | -451 ± 32 | -671 ± 30 | -850 ± 29 | -1098 ± 29 |
| SEALION | | -260.7 ± 26 | -512.1 ± 24 | -757.7 ± 27 | -1000 ± 24 | -1239.8 ± 26 |
| CNRS-SP$_1$ | | -269.5 | -502.9 | -735.2 | -970.9 | -1212.1 |
| CNRS-SP$_3$ | | -269.8 | -503.5 | -735.3 | -971.0 | -1212.4 |
| PoliMi | | -278.0 | -503.0 | -734.0 | -970.0 | -1219.0 |
| PSI | | -274.0 | -504.8 | -735.6 | -969.1 | -1206.8 |
| TUD-S$_2$ | | -269.5 | -498.1 | -727.7 | -961.2 | -1200.2 |
| TUD-S$_6$ | | -263.8 | -488.0 | -713.2 | -942.2 | -1176.4 |
| DTU | 0.5 | -289 ± 30 | -468 ± 27 | -655 ± 28 | -836 ± 31 | -1093 ± 33 |
| SEALION | | -254.4 ± 28 | -503.2 ± 24 | -746.1 ± 30 | -985.9 ± 26 | -1223.1± 27 |
| CNRS-SP$_1$ | | -276.5 | -503.5 | -732.9 | -966.3 | -1204.8 |
| CNRS-SP$_3$ | | -276.8 | -503.5 | -733 | -966.5 | -1205.2 |
| PoliMi | | -284 | -508 | -737 | -972 | -1214 |
| PSI | | -278.1 | -504.6 | -733.1 | -964.3 | -1199.8 |
| TUD-S$_2$ | | -273.1 | -497.8 | -725.2 | -956.4 | -1193 |
| TUD-S$_6$ | | -267.5 | -487.8 | -710.8 | -937.6 | -1169.7 |

Table 10: Step 1.4 - Reactivity change from Step 0.2 as a function of P and $U_{lid}$.

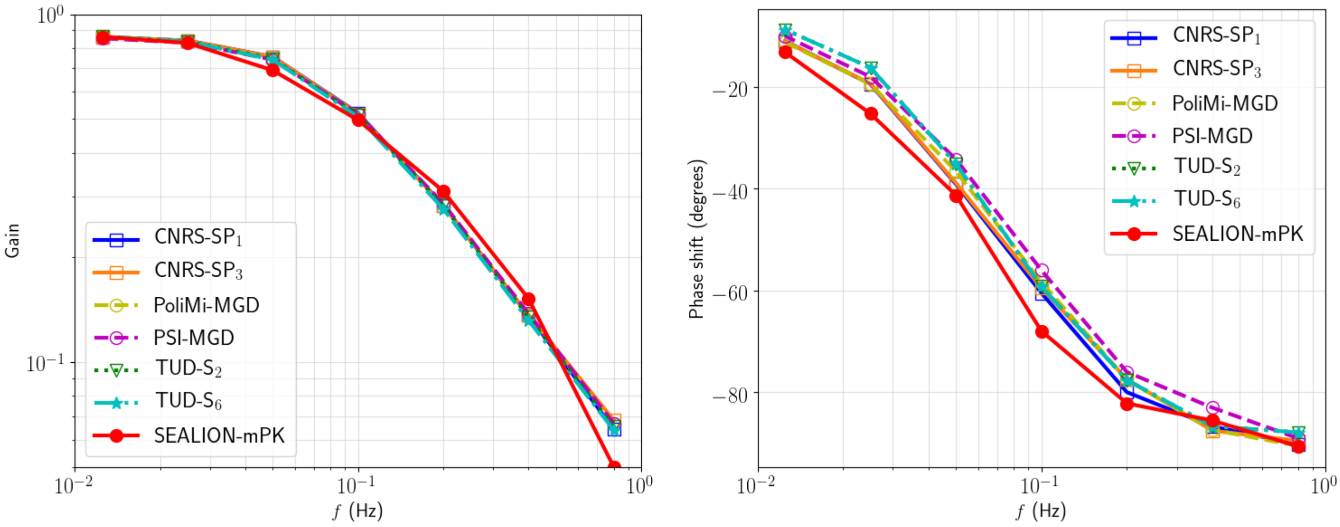


Figure 12: Step 2.1 - Bode diagrams of power gain and phase-shift as a function of the frequency of the $\gamma$ wave.

describe the neutronics feedback mechanisms. Both the DTU and the SEALION approaches shows acceptable agreement with the other participants for steady-state steps. The solvers capture most of the intrinsic features of the liquid fuel, *inter alia* the reactivity loss due to the circulation of the DNPs, the reactor feedback mechanisms and the distribution of the DNP fields. Given the simple benchmark geometry, the coupling schemes applied to the benchmark including the DTU and SEALION are able to model the physics of the liquid fueled reactor as long as any perturbation to the system is not both large and local. However, both the DTU and the SEALION code have limitations in describing the higher order effects in the fission rate density caused by a shift of DNPs from a region of high neutron importance in the center of the cavity to the region of lower neutron importance away from the center.

## Appendix A. Additional tables

In the Appendix A several tables are presented containing raw data for velocity, temperature and delayed neutron precursor source values by all participants for the various benchmark steps.

## Appendix B. Observables fields

In the Appendix B several figures are presented containing raw data for velocity, temperature and delayed neutron precursor source values by all participants for the various benchmark steps.

Table A.11: Step 0.1 - Velocity components along centerlines *AA*' and *BB*'.

| Observable | Code | Results along *AA*' (points coordinates are expressed in m) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | (0.0,1) | (0.25,1) | (0.5,1) | (0.75,1) | (1.0,1) | (1.25,1) | (1.5,1) | (1.75,1) | (2.0,1) |
| $u_x$ $(ms^{-1})$ | DTU | 0.000E+00 | -1.939E-02 | -5.358E-02 | -8.281E-02 | -1.008E-01 | -1.023E-01 | -7.853E-02 | -3.084E-02 | 0.000E+00 |
| | SEALION | 0.000E+00 | -1.929E-02 | -5.367E-02 | -8.335E-02 | -1.018E-01 | -1.034E-01 | -7.913E-02 | -3.072E-02 | 0.000E+00 |
| | CNRS | 0.000E+00 | -1.924E-02 | -5.372E-02 | -8.369E-02 | -1.025E-01 | -1.043E-01 | -7.972E-02 | -3.080E-02 | 0.000E+00 |
| | PoliMi | 0.000E+00 | -1.922E-02 | -5.365E-02 | -8.357E-02 | -1.023E-01 | -1.041E-01 | -7.947E-02 | -3.066E-02 | 0.000E+00 |
| | PSI | 0.000E+00 | -1.929E-02 | -5.366E-02 | -8.332E-02 | -1.018E-01 | -1.034E-01 | -7.912E-02 | -3.072E-02 | 0.000E+00 |
| | TUD | 1.002E-06 | -1.922E-02 | -5.372E-02 | -8.371E-02 | -1.025E-01 | -1.044E-01 | -7.977E-02 | -3.081E-02 | 4.198E-06 |
| $u_y$ $(ms^{-1})$ | DTU | 0.000E+00 | 7.279E-02 | 8.506E-02 | 5.975E-02 | 1.211E-02 | -4.683E-02 | -9.465E-02 | -8.753E-02 | 0.000E+00 |
| | SEALION | 0.000E+00 | 7.254E-02 | 8.526E-02 | 6.021E-02 | 1.232E-02 | -4.732E-02 | -9.529E-02 | -8.711E-02 | 0.000E+00 |
| | CNRS | 0.000E+00 | 7.266E-02 | 8.575E-02 | 6.084E-02 | 1.251E-02 | -4.789E-02 | -9.606E-02 | -8.722E-02 | 0.000E+00 |
| | PoliMi | 0.000E+00 | 7.139E-02 | 8.433E-02 | 6.007E-02 | 1.269E-02 | -4.691E-02 | -9.472E-02 | -8.621E-02 | 0.000E+00 |
| | PSI | 0.000E+00 | 7.265E-02 | 8.534E-02 | 6.021E-02 | 1.230E-02 | -4.734E-02 | -9.536E-02 | -8.720E-02 | 0.000E+00 |
| | TUD | 5.877E-06 | 7.269E-02 | 8.580E-02 | 6.089E-02 | 1.252E-02 | -4.794E-02 | -9.613E-02 | -8.726E-02 | -1.013E-05 |
| Observable | Code | Results along *BB*' (points coordinates are expressed in m) | | | | | | | | |
| | | (1,0.0) | (1,0.25) | (1,0.5) | (1,0.75) | (1,1.0) | (1,1.25) | (1,1.5) | (1,1.75) | (1,2.0) |
| $u_x$ $(ms^{-1})$ | DTU | 0.000E+00 | -3.530E-02 | -6.223E-02 | -8.629E-02 | -1.008E-01 | -8.611E-02 | -1.123E-02 | 1.696E-01 | 5.000E-01 |
| | SEALION | 0.000E+00 | -3.504E-02 | -6.209E-02 | -8.664E-02 | -1.018E-01 | -8.739E-02 | -1.196E-02 | 1.705E-01 | 5.000E-01 |
| | CNRS | 0.000E+00 | -3.517E-02 | -6.242E-02 | -8.720E-02 | -1.025E-01 | -8.766E-02 | -1.147E-02 | 1.717E-01 | 5.000E-01 |
| | PoliMi | 0.000E+00 | -3.423E-02 | -6.107E-02 | -8.613E-02 | -1.023E-01 | -8.861E-02 | -1.299E-02 | 1.706E-01 | 5.000E-01 |
| | PSI | 0.000E+00 | -3.511E-02 | -6.217E-02 | -8.667E-02 | -1.018E-01 | -8.731E-02 | -1.191E-02 | 1.705E-01 | 5.000E-01 |
| | TUD | 1.494E-06 | -3.519E-02 | -6.244E-02 | -8.724E-02 | -1.025E-01 | -8.770E-02 | -1.146E-02 | 1.718E-01 | 5.000E-01 |
| $u_y$ $(ms^{-1})$ | DTU | 0.000E+00 | 1.077E-04 | 7.568E-04 | 3.921E-03 | 1.211E-02 | 2.411E-02 | 2.928E-02 | 1.478E-02 | 0.000E+00 |
| | SEALION | 0.000E+00 | 7.850E-05 | 6.866E-04 | 3.882E-03 | 1.232E-02 | 2.475E-02 | 2.996E-02 | 1.482E-02 | 0.000E+00 |
| | CNRS | 0.000E+00 | 5.641E-05 | 6.309E-04 | 3.862E-03 | 1.251E-02 | 2.524E-02 | 3.048E-02 | 1.500E-02 | 0.000E+00 |
| | PoliMi | 0.000E+00 | 9.118E-05 | 7.484E-04 | 4.046E-03 | 1.269E-02 | 2.534E-02 | 3.050E-02 | 1.500E-02 | 0.000E+00 |
| | PSI | 0.000E+00 | 7.727E-05 | 6.822E-04 | 3.875E-03 | 1.230E-02 | 2.472E-02 | 2.994E-02 | 1.481E-02 | 0.000E+00 |
| | TUD | 1.501E-06 | 5.260E-05 | 6.209E-04 | 3.853E-03 | 1.252E-02 | 2.528E-02 | 3.053E-02 | 1.502E-02 | 7.987E-06 |

Table A.12: Step 0.2 - Fission rate density along *AA*'.

| Observable | Code | Results along *AA*' (points coordinates are expressed in m) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | (0.0,1) | (0.25,1) | (0.5,1) | (0.75,1) | (1.0,1) | (1.25,1) | (1.5,1) | (1.75,1) | (2.0,1) |
| $\int_E \Sigma_f \phi dE$ $(s^{-1}m^{-3})$ | DTU-MC | 8.800E+17 | 7.410E+18 | 1.280E+19 | 1.660E+19 | 1.790E+19 | 1.660E+19 | 1.300E+19 | 7.560E+18 | 8.800E+17 |
| | SEALION-mPK | 8.510E+17 | 7.320E+18 | 1.280E+19 | 1.660E+19 | 1.790E+19 | 1.660E+19 | 1.310E+19 | 7.590E+18 | 8.510E+17 |
| | CNRS-SP$_1$ | 6.900E+17 | 7.440E+18 | 1.310E+19 | 1.680E+19 | 1.810E+19 | 1.680E+19 | 1.310E+19 | 7.440E+18 | 6.900E+17 |
| | CNRS-SP$_3$ | 6.210E+17 | 7.450E+18 | 1.300E+19 | 1.670E+19 | 1.800E+19 | 1.670E+19 | 1.300E+19 | 7.450E+18 | 6.210E+17 |
| | PoliMi-MGD | 7.780E+17 | 7.470E+18 | 1.310E+19 | 1.680E+19 | 1.810E+19 | 1.680E+19 | 1.310E+19 | 7.470E+18 | 7.780E+17 |
| | PSI-MGD | 8.620E+17 | 7.440E+18 | 1.310E+19 | 1.680E+19 | 1.810E+19 | 1.680E+19 | 1.310E+19 | 7.440E+18 | 8.620E+17 |
| | TUD-S$_2$ | 6.630E+17 | 7.430E+18 | 1.310E+19 | 1.680E+19 | 1.810E+19 | 1.680E+19 | 1.310E+19 | 7.430E+18 | 6.630E+17 |
| | TUD-S$_6$ | 6.830E+17 | 7.460E+18 | 1.300E+19 | 1.670E+19 | 1.800E+19 | 1.670E+19 | 1.300E+19 | 7.460E+18 | 6.830E+17 |

Table A.13: Step 0.3 - Temperature distribution along centerlines *AA*' and *BB*'

| Observable | Code | Results along *AA*' (points coordinates are expressed in m) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | (0.0,1) | (0.25,1) | (0.5,1) | (0.75,1) | (1.0,1) | (1.25,1) | (1.5,1) | (1.75,1) | (2.0,1) |
| $T$ $(K)$ | DTU-MC | 9.363E+02 | 1.192E+03 | 1.354E+03 | 1.362E+03 | 1.306E+03 | 1.225E+03 | 1.132E+03 | 1.034E+03 | 9.345E+02 |
| | SEALION-mPK | 9.236E+02 | 1.195E+03 | 1.355E+03 | 1.358E+03 | 1.302E+03 | 1.224E+03 | 1.132E+03 | 1.039E+03 | 9.264E+02 |
| | CNRS-SP$_1$ | 9.253E+02 | 1.194E+03 | 1.358E+03 | 1.363E+03 | 1.305E+03 | 1.224E+03 | 1.131E+03 | 1.034E+03 | 9.251E+02 |
| | CNRS-SP$_3$ | 9.236E+02 | 1.194E+03 | 1.357E+03 | 1.361E+03 | 1.304E+03 | 1.224E+03 | 1.131E+03 | 1.034E+03 | 9.235E+02 |
| | PoliMi-MGD | 9.253E+02 | 1.196E+03 | 1.361E+03 | 1.364E+03 | 1.305E+03 | 1.224E+03 | 1.132E+03 | 1.035E+03 | 9.252E+02 |
| | PSI-MGD | 9.253E+02 | 1.196E+03 | 1.356E+03 | 1.363E+03 | 1.306E+03 | 1.226E+03 | 1.133E+03 | 1.037E+03 | 9.252E+02 |
| | TUD-S$_2$ | 9.212E+02 | 1.194E+03 | 1.359E+03 | 1.364E+03 | 1.305E+03 | 1.224E+03 | 1.131E+03 | 1.032E+03 | 9.225E+02 |
| | TUD-S$_6$ | 9.219E+02 | 1.194E+03 | 1.356E+03 | 1.360E+03 | 1.303E+03 | 1.223E+03 | 1.131E+03 | 1.034E+03 | 9.233E+02 |
| Observable | Code | Results along *BB*' (points coordinates are expressed in m) | | | | | | | | |
| | | (1,0.0) | (1,0.25) | (1,0.5) | (1,0.75) | (1,1.0) | (1,1.25) | (1,1.5) | (1,1.75) | (1,2.0) |
| $T$ $(K)$ | DTU-MC | 9.318E+02 | 1.139E+03 | 1.271E+03 | 1.305E+03 | 1.306E+03 | 1.311E+03 | 1.316E+03 | 1.260E+03 | 9.810E+02 |
| | SEALION-mPK | 9.263E+02 | 1.139E+03 | 1.271E+03 | 1.303E+03 | 1.302E+03 | 1.307E+03 | 1.319E+03 | 1.263E+03 | 9.527E+02 |
| | CNRS-SP$_1$ | 9.252E+02 | 1.139E+03 | 1.273E+03 | 1.305E+03 | 1.305E+03 | 1.314E+03 | 1.321E+03 | 1.265E+03 | 9.322E+02 |
| | CNRS-SP$_3$ | 9.236E+02 | 1.140E+03 | 1.272E+03 | 1.304E+03 | 1.304E+03 | 1.313E+03 | 1.320E+03 | 1.265E+03 | 9.322E+02 |
| | PoliMi-MGD | 9.253E+02 | 1.140E+03 | 1.275E+03 | 1.307E+03 | 1.305E+03 | 1.313E+03 | 1.321E+03 | 1.265E+03 | 9.303E+02 |
| | PSI-MGD | 9.252E+02 | 1.139E+03 | 1.273E+03 | 1.307E+03 | 1.306E+03 | 1.312E+03 | 1.319E+03 | 1.263E+03 | 9.481E+02 |
| | TUD-S$_2$ | 9.215E+02 | 1.139E+03 | 1.273E+03 | 1.306E+03 | 1.305E+03 | 1.315E+03 | 1.322E+03 | 1.265E+03 | 9.374E+02 |
| | TUD-S$_6$ | 9.222E+02 | 1.140E+03 | 1.272E+03 | 1.303E+03 | 1.303E+03 | 1.312E+03 | 1.319E+03 | 1.264E+03 | 9.390E+02 |

Table A.14: Step 1.1 - Delayed neutron source along *AA*' and *BB*.

| Observable | Code | Results along *AA*' (points coordinates are expressed in m) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | (0.0,1) | (0.25,1) | (0.5,1) | (0.75,1) | (1.0,1) | (1.25,1) | (1.5,1) | (1.75,1) | (2.0,1) |
| $\sum_i \lambda_i C_i$ ($m^{-3}s^{-1}$) | DTU-MC | 1.69E+016 | 1.430E+17 | 2.200E+17 | 2.420E+17 | 2.270E+17 | 1.920E+17 | 1.430E+17 | 8.060E+16 | 1.22E+016 |
| | SEALION-mPK | 1.649E+16 | 1.440E+17 | 2.204E+17 | 2.412E+17 | 2.268E+17 | 1.928E+17 | 1.468E+17 | 9.472E+16 | 1.949E+16 |
| | CNRS-SP$_1$ | 1.335E+16 | 1.452E+17 | 2.212E+17 | 2.411E+17 | 2.268E+17 | 1.923E+17 | 1.461E+17 | 9.214E+16 | 1.316E+16 |
| | CNRS-SP$_3$ | 1.251E+16 | 1.454E+17 | 2.209E+17 | 2.406E+17 | 2.264E+17 | 1.921E+17 | 1.462E+17 | 9.245E+16 | 1.233E+16 |
| | PoliMi-MGD | 1.321E+16 | 1.450E+17 | 2.219E+17 | 2.414E+17 | 2.266E+17 | 1.920E+17 | 1.459E+17 | 9.188E+16 | 1.292E+16 |
| | PSI-MGD | 1.325E+16 | 1.453E+17 | 2.214E+17 | 2.413E+17 | 2.270E+17 | 1.925E+17 | 1.463E+17 | 9.218E+16 | 1.314E+16 |
| | TUD-S$_2$ | 1.093E+16 | 1.438E+17 | 2.228E+17 | 2.426E+17 | 2.278E+17 | 1.927E+17 | 1.464E+17 | 8.968E+16 | 1.184E+16 |
| | TUD-S$_6$ | 1.132E+16 | 1.437E+17 | 2.212E+17 | 2.405E+17 | 2.261E+17 | 1.916E+17 | 1.461E+17 | 9.029E+16 | 1.224E+16 |

| Observable | Code | Results along *BB*' (points coordinates are expressed in m) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | (1,0.0) | (1,0.25) | (1,0.5) | (1,0.75) | (1,1.0) | (1,1.25) | (1,1.5) | (1,1.75) | (1,2.0) |
| $\sum_i \lambda_i C_i$ ($m^{-3}s^{-1}$) | DTU-MC | 1.55E+016 | 1.150E+17 | 1.840E+17 | 2.180E+17 | 2.270E+17 | 2.250E+17 | 2.150E+17 | 1.740E+17 | 1.79E+016 |
| | SEALION-mPK | 2.081E+16 | 1.231E+17 | 1.915E+17 | 2.213E+17 | 2.268E+17 | 2.252E+17 | 2.145E+17 | 1.743E+17 | 4.482E+16 |
| | CNRS-SP$_1$ | 1.306E+16 | 1.190E+17 | 1.881E+17 | 2.193E+17 | 2.268E+17 | 2.261E+17 | 2.178E+17 | 1.754E+17 | 3.079E+16 |
| | CNRS-SP$_3$ | 1.222E+16 | 1.193E+17 | 1.879E+17 | 2.189E+17 | 2.264E+17 | 2.257E+17 | 2.175E+17 | 1.753E+17 | 3.072E+16 |
| | PoliMi-MGD | 1.297E+16 | 1.186E+17 | 1.881E+17 | 2.194E+17 | 2.266E+17 | 2.260E+17 | 2.177E+17 | 1.756E+17 | 2.805E+16 |
| | PSI-MGD | 1.299E+16 | 1.189E+17 | 1.881E+17 | 2.195E+17 | 2.270E+17 | 2.261E+17 | 2.176E+17 | 1.752E+17 | 2.730E+16 |
| | TUD-S$_2$ | 1.109E+16 | 1.174E+17 | 1.882E+17 | 2.203E+17 | 2.278E+17 | 2.281E+17 | 2.193E+17 | 1.768E+17 | 2.655E+16 |
| | TUD-S$_6$ | 1.143E+16 | 1.178E+17 | 1.872E+17 | 2.186E+17 | 2.261E+17 | 2.264E+17 | 2.179E+17 | 1.761E+17 | 2.728E+16 |

Table A.15: Step 1.2 - Profiles along *AA*' and *BB*' of the temperature and the change of fission rate density with respect to the solution obtained at Step 0.2.

| Observable | Code | Results along *AA*' (points coordinates are expressed in m) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | (0.0,1) | (0.25,1) | (0.5,1) | (0.75,1) | (1.0,1) | (1.25,1) | (1.5,1) | (1.75,1) | (2.0,1) |
| $T$ ($K$) | DTU-MC | 9.401E+02 | 1.193E+03 | 1.338E+03 | 1.349E+03 | 1.301E+03 | 1.227E+03 | 1.139E+03 | 1.044E+03 | 9.387E+02 |
| | SEALION-mPK | 9.280E+02 | 1.190E+03 | 1.346E+03 | 1.356E+03 | 1.300E+03 | 1.221E+03 | 1.131E+03 | 1.043E+03 | 9.278E+02 |
| | CNRS-SP$_1$ | 9.280E+02 | 1.195E+03 | 1.341E+03 | 1.349E+03 | 1.298E+03 | 1.225E+03 | 1.136E+03 | 1.041E+03 | 9.278E+02 |
| | CNRS-SP$_3$ | 9.262E+02 | 1.195E+03 | 1.341E+03 | 1.348E+03 | 1.298E+03 | 1.225E+03 | 1.137E+03 | 1.042E+03 | 9.260E+02 |
| | PoliMi-MGD | 9.281E+02 | 1.198E+03 | 1.343E+03 | 1.350E+03 | 1.300E+03 | 1.226E+03 | 1.138E+03 | 1.045E+03 | 9.280E+02 |
| | PSI-MGD | 9.282E+02 | 1.197E+03 | 1.340E+03 | 1.349E+03 | 1.300E+03 | 1.227E+03 | 1.139E+03 | 1.045E+03 | 9.280E+02 |
| | TUD-S$_2$ | 9.235E+02 | 1.196E+03 | 1.343E+03 | 1.350E+03 | 1.300E+03 | 1.226E+03 | 1.137E+03 | 1.041E+03 | 9.250E+02 |
| | TUD-S$_6$ | 9.243E+02 | 1.196E+03 | 1.340E+03 | 1.347E+03 | 1.298E+03 | 1.225E+03 | 1.137E+03 | 1.042E+03 | 9.258E+02 |
| $\Delta \int_E \Sigma_f \phi dE$ ($s^{-1}m^{-3}$) | DTU-MC | 4.630E+16 | 9.190E+16 | -8.580E+17 | -1.150E+18 | -1.050E+18 | -2.430E+17 | 4.560E+17 | 4.040E+17 | 5.930E+16 |
| | SEALION-mPK | 4.567E+16 | 1.503E+17 | -5.552E+17 | -9.958E+17 | -7.411E+17 | -5.098E+16 | 5.362E+17 | 6.960E+17 | 1.267E+17 |
| | CNRS-SP$_1$ | 7.800E+16 | 1.170E+17 | -5.690E+17 | -9.490E+17 | -7.980E+17 | -2.850E+17 | 2.630E+17 | 4.610E+17 | 7.780E+16 |
| | CNRS-SP$_3$ | 7.020E+16 | 1.160E+17 | -5.670E+17 | -9.430E+17 | -7.940E+17 | -2.870E+17 | 2.580E+17 | 4.580E+17 | 6.990E+16 |
| | PoliMi-MGD | 7.560E+16 | 1.140E+17 | -5.680E+17 | -9.440E+17 | -7.910E+17 | -2.800E+17 | 2.630E+17 | 4.560E+17 | 7.520E+16 |
| | PSI-MGD | 2.188E+15 | 1.230E+17 | -5.490E+17 | -9.270E+17 | -7.810E+17 | -2.720E+17 | 2.700E+17 | 4.630E+17 | 1.835E+15 |
| | TUD-S$_2$ | 7.200E+16 | 1.070E+17 | -5.650E+17 | -9.530E+17 | -8.190E+17 | -2.720E+17 | 2.630E+17 | 4.530E+17 | 7.190E+16 |
| | TUD-S$_6$ | 7.300E+16 | 1.040E+17 | -5.570E+17 | -9.380E+17 | -8.100E+17 | -2.790E+17 | 2.470E+17 | 4.430E+17 | 7.270E+16 |

| Observable | Code | Results along *BB*' (points coordinates are expressed in m) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | (1,0.0) | (1,0.25) | (1,0.5) | (1,0.75) | (1,1.0) | (1,1.25) | (1,1.5) | (1,1.75) | (1,2.0) |
| $T$ ($K$) | DTU-MC | 9.362E+02 | 1.152E+03 | 1.275E+03 | 1.305E+03 | 1.301E+03 | 1.302E+03 | 1.301E+03 | 1.245E+03 | 9.834E+02 |
| | SEALION-mPK | 9.281E+02 | 1.139E+03 | 1.268E+03 | 1.301E+03 | 1.300E+03 | 1.305E+03 | 1.310E+03 | 1.252E+03 | 9.350E+02 |
| | CNRS-SP$_1$ | 9.281E+02 | 1.148E+03 | 1.272E+03 | 1.301E+03 | 1.298E+03 | 1.304E+03 | 1.307E+03 | 1.253E+03 | 9.350E+02 |
| | CNRS-SP$_3$ | 9.262E+02 | 1.149E+03 | 1.272E+03 | 1.300E+03 | 1.298E+03 | 1.303E+03 | 1.306E+03 | 1.253E+03 | 9.351E+02 |
| | PoliMi-MGD | 9.281E+02 | 1.150E+03 | 1.275E+03 | 1.304E+03 | 1.300E+03 | 1.304E+03 | 1.307E+03 | 1.253E+03 | 9.470E+02 |
| | PSI-MGD | 9.282E+02 | 1.148E+03 | 1.273E+03 | 1.303E+03 | 1.300E+03 | 1.303E+03 | 1.306E+03 | 1.252E+03 | 9.517E+02 |
| | TUD-S$_2$ | 9.240E+02 | 1.148E+03 | 1.274E+03 | 1.302E+03 | 1.300E+03 | 1.306E+03 | 1.309E+03 | 1.254E+03 | 9.424E+02 |
| | TUD-S$_6$ | 9.247E+02 | 1.149E+03 | 1.272E+03 | 1.300E+03 | 1.298E+03 | 1.303E+03 | 1.306E+03 | 1.253E+03 | 9.442E+02 |
| $\Delta \int_E \Sigma_f \phi dE$ ($s^{-1}m^{-3}$) | DTU-MC | 1.730E+14 | 2.710E+17 | -4.730E+17 | -4.280E+17 | -1.110E+18 | -1.150E+18 | -4.500E+17 | -6.450E+16 | 6.730E+16 |
| | SEALION-mPK | 1.185E+17 | 3.795E+17 | -7.906E+16 | -5.642E+17 | -6.375E+17 | -6.511E+17 | -5.618E+17 | -8.285E+16 | 6.022E+16 |
| | CNRS-SP$_1$ | 8.200E+16 | 2.640E+17 | -2.260E+17 | -6.430E+17 | -7.980E+17 | -7.560E+17 | -4.920E+17 | -3.950E+16 | 8.350E+16 |
| | CNRS-SP$_3$ | 7.380E+16 | 2.620E+17 | -2.270E+17 | -6.410E+17 | -7.940E+17 | -7.520E+17 | -4.900E+17 | -3.920E+16 | 7.490E+16 |
| | PoliMi-MGD | 7.960E+16 | 2.640E+17 | -2.260E+17 | -6.390E+17 | -7.940E+17 | -7.540E+17 | -4.920E+17 | -4.160E+16 | 8.050E+16 |
| | PSI-MGD | 9.170E+16 | 2.640E+17 | -2.290E+17 | -6.510E+17 | -8.020E+17 | -7.480E+17 | -4.860E+17 | -3.480E+16 | 9.060E+16 |
| | TUD-S$_2$ | 7.590E+16 | 2.510E+17 | -2.340E+17 | -6.590E+17 | -8.190E+17 | -7.700E+17 | -4.960E+17 | -4.660E+16 | 6.930E+16 |
| | TUD-S$_6$ | 7.680E+16 | 2.440E+17 | -2.370E+17 | -6.540E+17 | -8.100E+17 | -7.610E+17 | -4.910E+17 | -4.530E+16 | 6.990E+16 |

Table A.16: Step 1.3 - Velocity components, temperature distribution, and delayed neutron source along *AA*'.

| Observable | Code | Results along *AA*' (points coordinates are expressed in m) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | (0.0,1) | (0.25,1) | (0.5,1) | (0.75,1) | (1.0,1) | (1.25,1) | (1.5,1) | (1.75,1) | (2.0,1) |
| $u_x$ $(ms^{-1})$ | DTU-MC | 0.000E+00 | 1.558E-02 | 2.162E-02 | 1.472E-02 | 2.666E-04 | -1.476E-02 | -2.189E-02 | -1.563E-02 | 0.000E+00 |
| | SEALION-mPK | 9.781E-06 | 1.785E-02 | 2.497E-02 | 1.667E-02 | 2.912E-04 | -1.616E-02 | -2.468E-02 | -1.776E-02 | -9.938E-06 |
| | CNRS-SP$_1$ | 0.000E+00 | 1.641E-02 | 2.310E-02 | 1.579E-02 | 1.250E-09 | -1.579E-02 | -2.310E-02 | -1.641E-02 | 0.000E+00 |
| | CNRS-SP$_3$ | 0.000E+00 | 1.636E-02 | 2.302E-02 | 1.572E-02 | 1.000E-09 | -1.572E-02 | -2.302E-02 | -1.636E-02 | 0.000E+00 |
| | PoliMi-MGD | 0.000E+00 | 1.637E-02 | 2.312E-02 | 1.578E-02 | -1.130E-10 | -1.578E-02 | -2.312E-02 | -1.637E-02 | 0.000E+00 |
| | PSI-MGD | 0.000E+00 | 1.630E-02 | 2.263E-02 | 1.519E-02 | -8.530E-09 | -1.519E-02 | -2.263E-02 | -1.630E-02 | 0.000E+00 |
| | TUD-S$_2$ | 6.050E-06 | 1.644E-02 | 2.316E-02 | 1.584E-02 | -2.220E-06 | -1.584E-02 | -2.316E-02 | -1.644E-02 | -6.050E-06 |
| | TUD-S$_6$ | 5.980E-06 | 1.631E-02 | 2.295E-02 | 1.566E-02 | -2.200E-06 | -1.566E-02 | -2.295E-02 | -1.631E-02 | -5.980E-06 |
| $u_y$ $(ms^{-1})$ | DTU-MC | 0.000E+00 | -1.772E-01 | 1.972E-02 | 1.418E-01 | 1.721E-01 | 1.408E-01 | 1.908E-02 | -1.789E-01 | 0.000E+00 |
| | SEALION-mPK | -1.409E-02 | -1.773E-01 | 1.733E-02 | 1.402E-01 | 1.687E-01 | 1.402E-01 | 1.786E-02 | -1.766E-01 | -1.408E-02 |
| | CNRS-SP$_1$ | 0.000E+00 | -1.777E-01 | 1.721E-02 | 1.376E-01 | 1.649E-01 | 1.376E-01 | 1.721E-02 | -1.777E-01 | 0.000E+00 |
| | CNRS-SP$_3$ | 0.000E+00 | -1.771E-01 | 1.708E-02 | 1.372E-01 | 1.645E-01 | 1.372E-01 | 1.708E-02 | -1.771E-01 | 0.000E+00 |
| | PoliMi-MGD | 0.000E+00 | -1.767E-01 | 1.741E-02 | 1.368E-01 | 1.638E-01 | 1.368E-01 | 1.741E-02 | -1.767E-01 | 0.000E+00 |
| | PSI-MGD | 0.000E+00 | -1.779E-01 | 1.662E-02 | 1.376E-01 | 1.659E-01 | 1.376E-01 | 1.662E-02 | -1.779E-01 | 0.000E+00 |
| | TUD-S$_2$ | -2.890E-05 | -1.780E-01 | 1.735E-02 | 1.379E-01 | 1.650E-01 | 1.379E-01 | 1.735E-02 | -1.780E-01 | -2.890E-05 |
| | TUD-S$_6$ | -2.930E-05 | -1.766E-01 | 1.694E-02 | 1.368E-01 | 1.639E-01 | 1.368E-01 | 1.694E-02 | -1.766E-01 | -2.930E-05 |
| $T$ $(K)$ | DTU-MC | 9.354E+02 | 1.190E+03 | 1.278E+03 | 1.283E+03 | 1.283E+03 | 1.284E+03 | 1.282E+03 | 1.189E+03 | 9.351E+02 |
| | SEALION-mPK | 9.279E+02 | 1.186E+03 | 1.274E+03 | 1.281E+03 | 1.276E+03 | 1.281E+03 | 1.274E+03 | 1.186E+03 | 9.279E+02 |
| | CNRS-SP$_1$ | 9.279E+02 | 1.193E+03 | 1.278E+03 | 1.284E+03 | 1.280E+03 | 1.284E+03 | 1.278E+03 | 1.193E+03 | 9.279E+02 |
| | CNRS-SP$_3$ | 9.260E+02 | 1.193E+03 | 1.278E+03 | 1.284E+03 | 1.280E+03 | 1.284E+03 | 1.278E+03 | 1.193E+03 | 9.260E+02 |
| | PoliMi-MGD | 9.279E+02 | 1.193E+03 | 1.279E+03 | 1.286E+03 | 1.282E+03 | 1.286E+03 | 1.279E+03 | 1.193E+03 | 9.279E+02 |
| | PSI-MGD | 9.279E+02 | 1.191E+03 | 1.278E+03 | 1.284E+03 | 1.280E+03 | 1.284E+03 | 1.278E+03 | 1.191E+03 | 9.279E+02 |
| | TUD-S$_2$ | 9.248E+02 | 1.193E+03 | 1.279E+03 | 1.285E+03 | 1.281E+03 | 1.285E+03 | 1.279E+03 | 1.193E+03 | 9.248E+02 |
| | TUD-S$_6$ | 9.257E+02 | 1.192E+03 | 1.277E+03 | 1.283E+03 | 1.280E+03 | 1.283E+03 | 1.277E+03 | 1.192E+03 | 9.257E+02 |
| $\sum_i \lambda_i C_i$ $(m^{-3}s^{-1})$ | DTU-MC | 0.000E+00 | 1.493E+17 | 2.026E+17 | 2.098E+17 | 1.913E+17 | 2.100E+17 | 2.017E+17 | 1.489E+17 | 0.000E+00 |
| | SEALION-mPK | 1.440E+16 | 1.460E+17 | 2.000E+17 | 2.100E+17 | 1.920E+17 | 2.100E+17 | 2.000E+17 | 1.460E+17 | 1.440E+16 |
| | CNRS-SP$_1$ | 1.500E+16 | 1.470E+17 | 2.000E+17 | 2.120E+17 | 1.990E+17 | 2.120E+17 | 2.000E+17 | 1.470E+17 | 1.500E+16 |
| | CNRS-SP$_3$ | 1.410E+16 | 1.470E+17 | 2.000E+17 | 2.120E+17 | 1.990E+17 | 2.120E+17 | 2.000E+17 | 1.470E+17 | 1.410E+16 |
| | PoliMi-MGD | 1.440E+16 | 1.460E+17 | 1.990E+17 | 2.110E+17 | 1.940E+17 | 2.110E+17 | 1.990E+17 | 1.460E+17 | 1.440E+16 |
| | PSI-MGD | 1.480E+16 | 1.470E+17 | 2.000E+17 | 2.130E+17 | 1.980E+17 | 2.130E+17 | 2.000E+17 | 1.470E+17 | 1.480E+16 |
| | TUD-S$_2$ | 1.040E+16 | 1.490E+17 | 2.020E+17 | 2.140E+17 | 1.940E+17 | 2.140E+17 | 2.020E+17 | 1.490E+17 | 1.040E+16 |
| | TUD-S$_6$ | 1.090E+16 | 1.480E+17 | 2.010E+17 | 2.130E+17 | 1.930E+17 | 2.130E+17 | 2.010E+17 | 1.480E+17 | 1.090E+16 |
| Observable | Code | Results along *BB*' (points coordinates are expressed in m) | | | | | | | | |
| | | (1,0.0) | (1,0.25) | (1,0.5) | (1,0.75) | (1,1.0) | (1,1.25) | (1,1.5) | (1,1.75) | (1,2.0) |
| $u_y$ $(ms^{-1})$ | DTU-MC | 0.000E+00 | 3.690E-02 | 9.397E-02 | 1.422E-01 | 1.721E-01 | 1.738E-01 | 1.368E-01 | 6.071E-02 | 0.000E+00 |
| | SEALION-mPK | 1.119E-05 | 3.478E-02 | 8.972E-02 | 1.377E-01 | 1.687E-01 | 1.723E-01 | 1.368E-01 | 6.075E-02 | 1.909E-05 |
| | CNRS-SP$_1$ | 0.000E+00 | 3.512E-02 | 8.947E-02 | 1.359E-01 | 1.649E-01 | 1.665E-01 | 1.307E-01 | 5.756E-02 | 0.000E+00 |
| | CNRS-SP$_3$ | 0.000E+00 | 3.510E-02 | 8.933E-02 | 1.356E-01 | 1.645E-01 | 1.660E-01 | 1.303E-01 | 5.740E-02 | 0.000E+00 |
| | PoliMi-MGD | 0.000E+00 | 3.507E-02 | 8.909E-02 | 1.351E-01 | 1.638E-01 | 1.656E-01 | 1.302E-01 | 5.743E-02 | 0.000E+00 |
| | PSI-MGD | 0.000E+00 | 3.537E-02 | 9.055E-02 | 1.374E-01 | 1.659E-01 | 1.669E-01 | 1.309E-01 | 5.780E-02 | 0.000E+00 |
| | TUD-S$_2$ | -1.230E-05 | 3.510E-02 | 8.950E-02 | 1.360E-01 | 1.650E-01 | 1.667E-01 | 1.308E-01 | 5.763E-02 | -2.810E-05 |
| | TUD-S$_6$ | -1.330E-05 | 3.506E-02 | 8.912E-02 | 1.352E-01 | 1.639E-01 | 1.655E-01 | 1.299E-01 | 5.719E-02 | -2.870E-05 |
| T(K) | DTU-MC | 9.352E+02 | 1.068E+03 | 1.158E+03 | 1.229E+03 | 1.283E+03 | 1.316E+03 | 1.322E+03 | 1.276E+03 | 9.573E+02 |
| | SEALION-mPK | 9.280E+02 | 1.063E+03 | 1.148E+03 | 1.219E+03 | 1.276E+03 | 1.316E+03 | 1.328E+03 | 1.285E+03 | 9.284E+02 |
| | CNRS-SP$_1$ | 9.280E+02 | 1.067E+03 | 1.156E+03 | 1.226E+03 | 1.280E+03 | 1.315E+03 | 1.326E+03 | 1.283E+03 | 9.284E+02 |
| | CNRS-SP$_3$ | 9.261E+02 | 1.067E+03 | 1.156E+03 | 1.226E+03 | 1.280E+03 | 1.315E+03 | 1.325E+03 | 1.282E+03 | 9.266E+02 |
| | PoliMi-MGD | 9.280E+02 | 1.067E+03 | 1.157E+03 | 1.228E+03 | 1.282E+03 | 1.317E+03 | 1.327E+03 | 1.284E+03 | 9.282E+02 |
| | PSI-MGD | 9.281E+02 | 1.068E+03 | 1.156E+03 | 1.226E+03 | 1.280E+03 | 1.314E+03 | 1.324E+03 | 1.281E+03 | 9.287E+02 |
| | TUD-S$_2$ | 9.250E+02 | 1.066E+03 | 1.156E+03 | 1.227E+03 | 1.281E+03 | 1.316E+03 | 1.327E+03 | 1.283E+03 | 9.137E+02 |
| | TUD-S$_6$ | 9.258E+02 | 1.069E+03 | 1.157E+03 | 1.226E+03 | 1.280E+03 | 1.314E+03 | 1.325E+03 | 1.282E+03 | 9.149E+02 |
| $\sum_i \lambda_i C_i$ $(m^{-3}s^{-1})$ | DTU-MC | 0.000E+00 | 7.838E+16 | 1.258E+17 | 1.639E+17 | 1.913E+17 | 2.052E+17 | 2.020E+17 | 1.683E+17 | 0.000E+00 |
| | SEALION-mPK | 1.050E+16 | 8.020E+16 | 1.280E+17 | 1.660E+17 | 1.920E+17 | 2.050E+17 | 2.000E+17 | 1.650E+17 | 1.580E+16 |
| | CNRS-SP$_1$ | 1.480E+16 | 8.800E+16 | 1.380E+17 | 1.750E+17 | 1.990E+17 | 2.090E+17 | 2.020E+17 | 1.660E+17 | 1.720E+16 |
| | CNRS-SP$_3$ | 1.380E+16 | 8.840E+16 | 1.380E+17 | 1.750E+17 | 1.990E+17 | 2.080E+17 | 2.010E+17 | 1.660E+17 | 1.630E+16 |
| | PoliMi-MGD | 1.450E+16 | 8.620E+16 | 1.340E+17 | 1.710E+17 | 1.940E+17 | 2.050E+17 | 1.980E+17 | 1.620E+17 | 1.580E+16 |
| | PSI-MGD | 1.480E+16 | 8.780E+16 | 1.370E+17 | 1.740E+17 | 1.980E+17 | 2.080E+17 | 2.010E+17 | 1.660E+17 | 1.690E+16 |
| | TUD-S$_2$ | 1.340E+16 | 8.510E+16 | 1.330E+17 | 1.700E+17 | 1.940E+17 | 2.050E+17 | 1.980E+17 | 1.620E+17 | 1.240E+16 |
| | TUD-S$_6$ | 1.380E+16 | 8.600E+16 | 1.340E+17 | 1.690E+17 | 1.930E+17 | 2.030E+17 | 1.970E+17 | 1.620E+17 | 1.300E+16 |

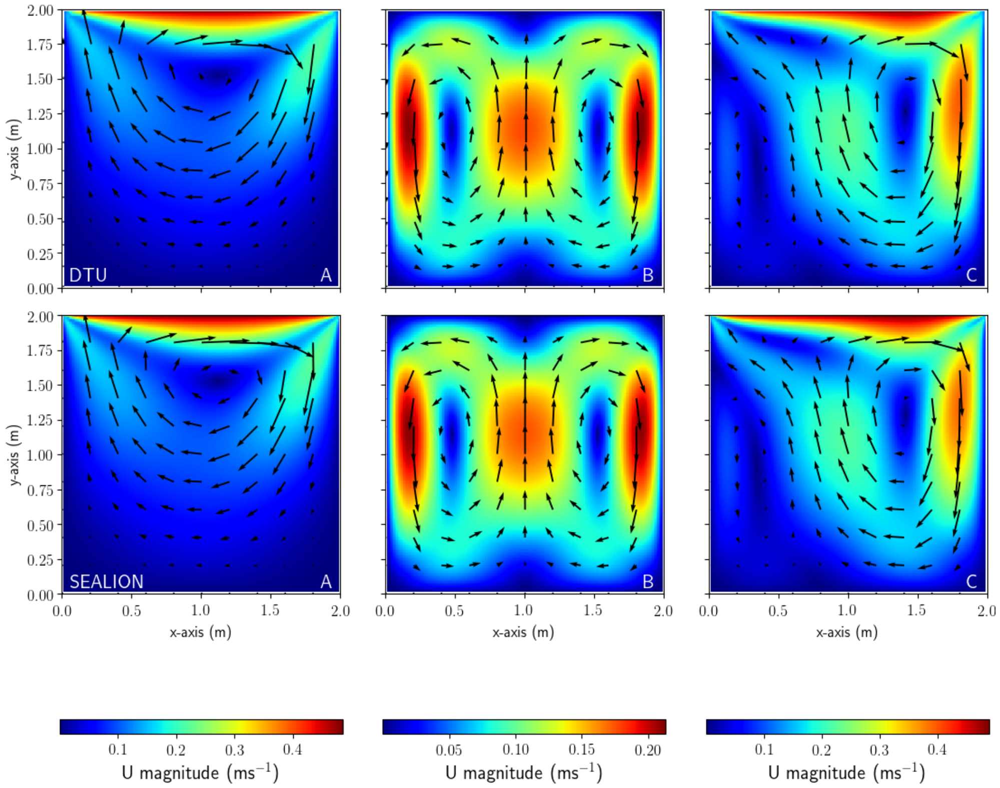


Figure B.13: Velocity magnitude field and flow streamlines obtained by each code in three different benchmark steps. From left to right: Step 0.1(A); Step 1.3(B); and Step 1.4(C) with P = 1 GW and $U_{lid}$ = 0.5ms$^{-1}$. DTU results are shown in the top row and results from SEALION are shown in the bottom row.

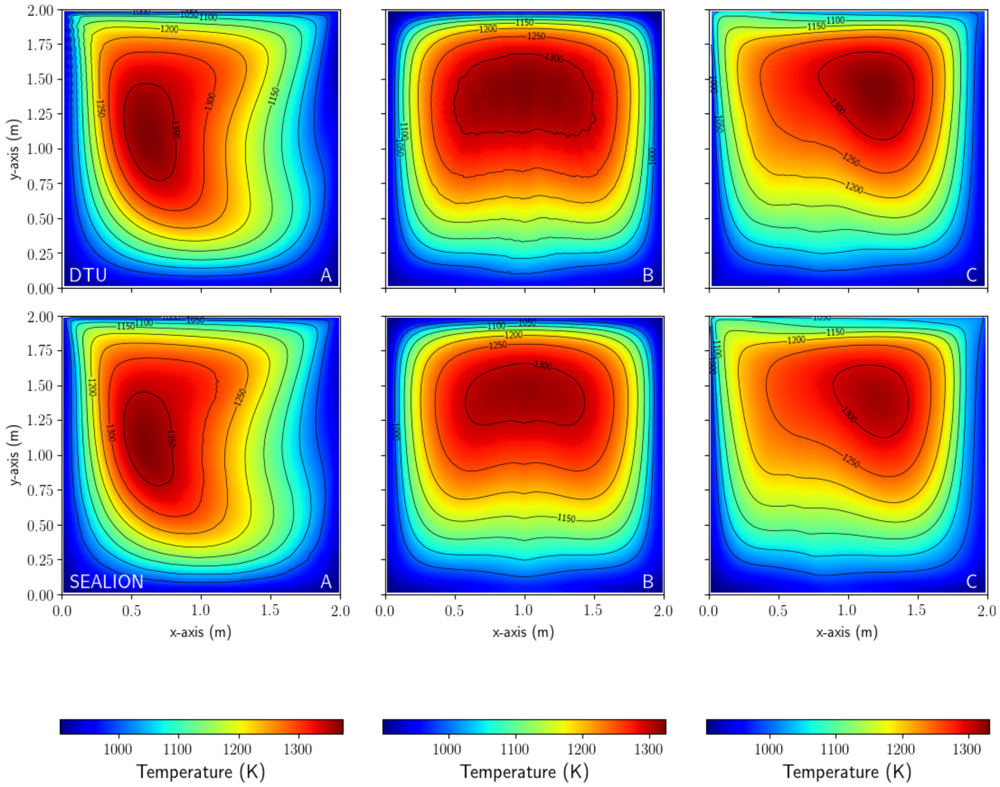


Figure B.14: Temperature field with isolines obtained by each code in three different benchmark steps. From left to right: Step 0.3(A); Step 1.3(B); and Step 1.4(C) with P = 1 GW and $U_{lid}$ = 0.5ms$^{-1}$. DTU results are shown in the top row and results from SEALION are shown in the bottom row.

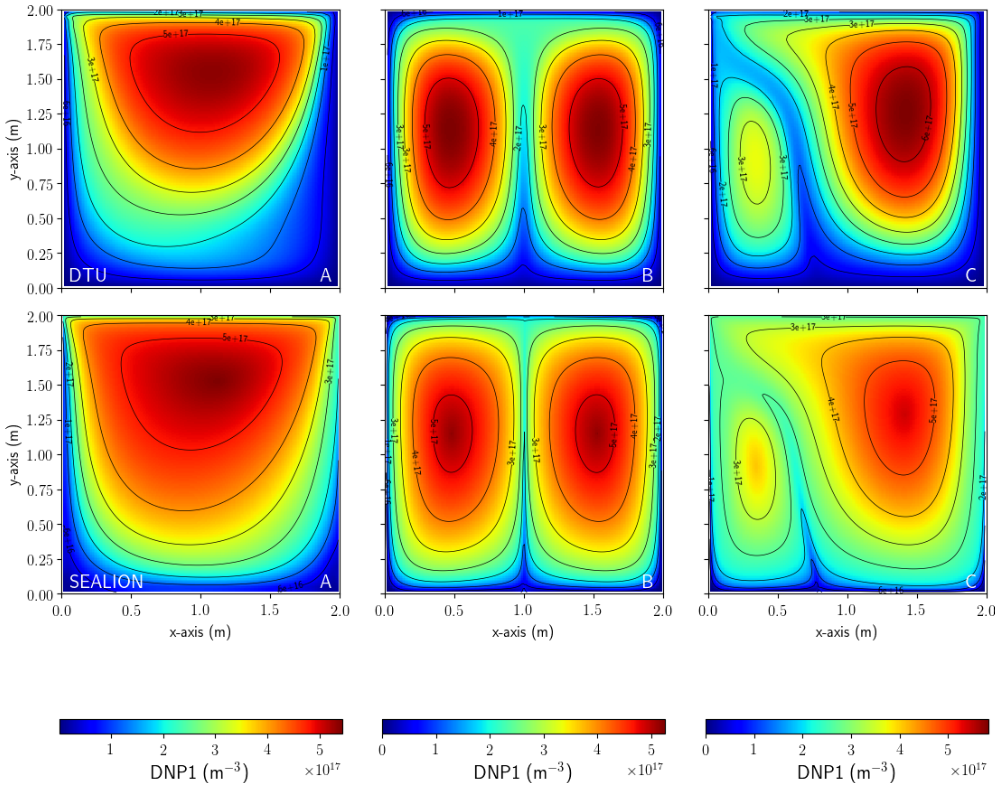


Figure B.15: Distribution of the first family of delayed neutron precursors ($T_{1/2} = 55.6$s) with isolines obtained by each code in three different benchmark steps. From left to right: Step 1.1(A); Step 1.3(B); and Step 1.4(C) with P = 1 GW and $U_{lid} = 0.5\text{ms}^{-1}$. DTU results are shown in the top row and results from SEALION are shown in the bottom row.

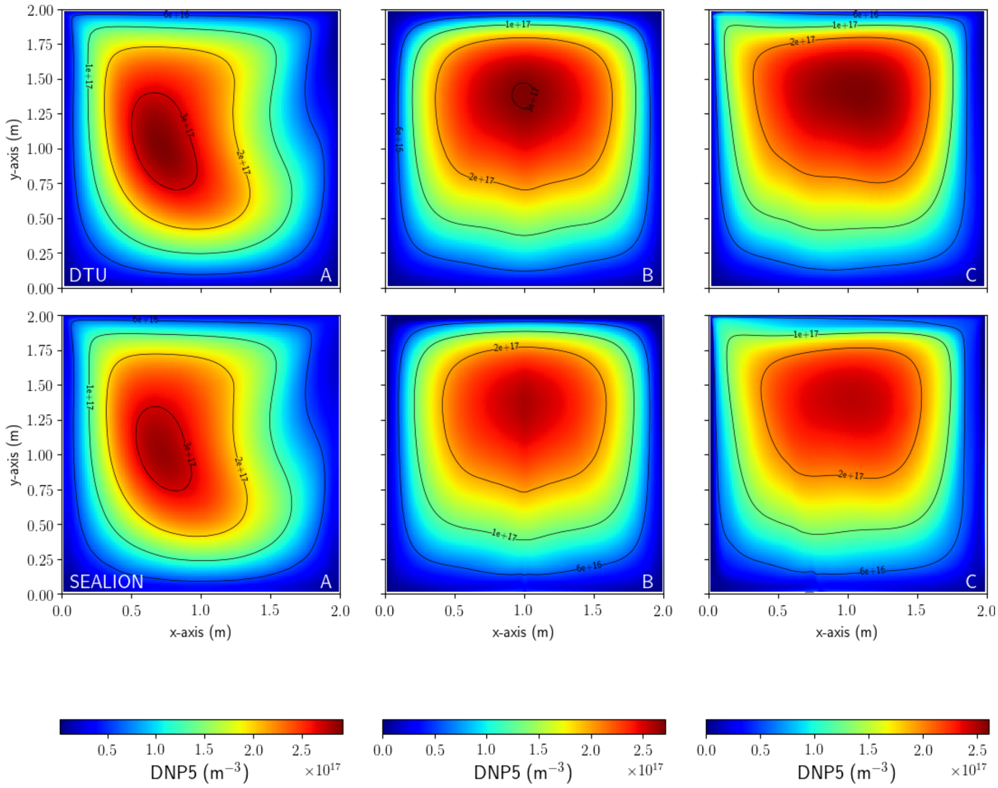


Figure B.16: Distribution of the fifth family of delayed neutron precursors ($T_{1/2} = 2.37$s) with isolines obtained by each code in three different benchmark steps. From left to right: Step 1.1(A); Step 1.3(B); and Step 1.4(C) with P = 1 GW and $U_{lid} = 0.5\text{ms}^{-1}$ DTU results are shown in the top row and results from SEALION are shown in the bottom row.

## References

A. Koning, R. Forrest, M. K., 2006. The jeff-3.1 nuclear data library. technical report jeff report 21 nea-oecd. Tech. rep., NEA-OECD.

A.J.M. Plompen, O. Cabellos, C. D. S. J., 2020. The joint evaluated fission and fusion nuclear data library, jeff-3.3. The European Physical Journal A 56 (181).
DOI: `https://doi.org/10.1140/epja/s10050-020-00141-9`

Ash, M., 1979. Nuclear Reactor Kinetics. Advanced book program. McGraw-Hill.

Aufiero, M., 2015. Serpent-openfoam coupling for criticality accidents modelling â definition of a benchmark for msrs multiphysics modelling. Serpent and Multiphysics meeting.

Aufiero, M., Rubiolo, P., 2018. Testing and verification of multiphysics tools for fast-spectrum msrs: The cnrs benchmark. ANS Annual Meeting.

Blanco, J., Rubiolo, P., Dumonteil, E., 2020. Neutronic modeling strategies for a liquid fuel transient calculation. En: Physics of Reactors Conference. PHYSOR.

Botella, O., Peyret, R., 1998. Benchmark spectral results on the lid-driven cavity flow. Computers  Fluids 27 (4), 421–433.
URL: `https://www.sciencedirect.com/science/article/pii/S0045793098000024`
DOI: `https://doi.org/10.1016/S0045-7930(98)00002-4`

Boussier, H., Delpech, S., Ghetta, V., Serp, J., 2012. The molten salt reactor in generation iv: Overview and perspectives. En: Generation4 International Forum Symposium, San Diego, USA. Generation4 International Forum Symposium, San Diego, USA.

Cervi, E., Lorenzi, S., Cammi, A., Luzzi, L., 2019a. Development of a multiphysics model for the study of fuel compressibility effects in the molten salt fast reactor. Chem. Eng. Sci 142, 379–393.
DOI: `https://doi.org/10.1016/j.ces.2018.09.025`

Cervi, E., Lorenzi, S., Cammi, A., Luzzi, L., 2019b. Development of an sp3 neutron transport solver for the analysis of the molten salt fast reactor. Nuclear Engineering and Design 346, 209 – 219.
URL: `http://www.sciencedirect.com/science/article/pii/S0029549319300354`
DOI: `https://doi.org/10.1016/j.nucengdes.2019.03.001`

Diniz, R. C., da Cruz GonÃ§alves, A., de Souza da Rosa, F. S., 2020. Neutron point kinetics model with precursorsâ shape function update for molten salt reactor. Nuclear Engineering and Design 360, 110466.
URL: `https://www.sciencedirect.com/science/article/pii/S0029549319304972`
DOI: `https://doi.org/10.1016/j.nucengdes.2019.110466`

Dormand, J. R., Prince, P. J., 1980. A family of embedded runge-kutta formulae. Journal of Computational and Applied Mathematics 6 (1), 19–26.

Eric Jones, Travis Oliphant, P. P., 2001. Scipy: Open source scientific tools for python.
URL: `http://www.scipy.org/`

Fiorina, C., Aufiero, M., Cammi, A., Guerrieri, C., Krepel, J., Luzzi, L., Mikityuk, K., Ricotti, M., 2012. Analysis of the msfr core neutronics adopting different neutron transport models. En: International Conference on Nuclear Engineering, ICONE. Anaheim, CA, USA. International Conference on Nuclear Engineering, ICONE. Anaheim, CA, USA.
DOI: `https://doi.org/10.1115/ICONE20-POWER2012-54519`

Fiorina, C., Clifford, I., Aufiero, M., Mikityuk, K., 2015. Gen-foam: a novel openfoam based multi-physics solver for 2d/3d transient analysis of nuclear reactors. Nuclear Engineering and Design 294, 24 – 37.
URL: `http://www.sciencedirect.com/science/article/pii/S0029549315003829`
DOI: `https://doi.org/10.1016/j.nucengdes.2015.05.035`

Fiorina, C., Lathouwers, D., Aufiero, M., Cammi, A., Guerrieri, C., Kloosterman, J. L., Luzzi, L., Ricotti, M. E., 2014. Modelling and analysis of the msfr transient behaviour. Annals of Nuclear Energy 64, 485 – 498.
URL: `http://www.sciencedirect.com/science/article/pii/S0306454913004118`
DOI: `https://doi.org/10.1016/j.anucene.2013.08.003`

Fletcher, C. D., Schultz, R. R., 1995. Relap5/mod3 code manual. Tech. rep., Idaho National Engineering Laboratory.
DOI: `10.2172/1466669`

Ghia, U., Ghia, K., Shin, C., 1982. High-re solutions for incompressible flow using the navier-stokes equations and a multigrid method. Journal of Computational Physics 48 (3), 387–411.
URL: `https://www.sciencedirect.com/science/article/pii/0021999182900584`
DOI: `https://doi.org/10.1016/0021-9991(82)90058-4`

Laureau, A., 2015. DÃ©veloppement de modeles neutroniques pour le couplage thermohydraulique du msfr et le calcul de parametres cinetiques effectifs. Ph.D. thesis, Grenoble Alpes University, France.

Leppänen, J., Aufiero, M., Fridman, E., Rachamin, R., van der Marck, S., 2014. Calculation of effective point kinetics parameters in the serpent 2 monte carlo code. Annals of Nuclear Energy 65, 272 – 279.
URL: `http://www.sciencedirect.com/science/article/pii/S0306454913005628`
DOI: `https://doi.org/10.1016/j.anucene.2013.10.032`

Leppänen, J., Pusa, M., Viitanen, T., Valtavirta, V., Kaltiaisenaho, T., 2015. The serpent monte carlo code: Status, development and applications in 2013. Annals of Nuclear Energy 82, 142 – 150, joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo 2013, SNA + MC 2013. Pluri- and Trans-disciplinarity, Towards New Modeling and Numerical Simulation Paradigms.
URL: `http://www.sciencedirect.com/science/article/pii/S0306454914004095`
DOI: `https://doi.org/10.1016/j.anucene.2014.08.024`

Nalbandyan, A., Groth-Jensen, J., Klinkby, E. B., Lauritzen, B., Steyn, R., 2019. Coupled neutronics thermal hydraulics assessment of graphite molten salt reactors. En: Proceedings of the NURETH18 conference. Proceedings of the NURETH18 conference.

Oliphant, T., 2006. Numpy: A guide to numpy.
URL: `http://www.numpy.org/`

SAMOFAR, 2019. A paradigm shift in nuclear reactor safety with the molten salt fast reactor. Tech. rep., SAMOFAR.

Tiberga, M., de Oliveira, R., E. Cervi, J. B., Loreniz, S., M.Aufiero, Lathouwers, D., Rubilo, P., 2020. Results from a multiphysics numerical benchmark for codes dedicated to molten salt fast reactors. Annals of Nuclear Energy 142, 107428.
DOI: `10.1016/j.anucene.2020.107428`

Tiberga, M., Lathouwers, D., Kloosterman, J., 2019. A discontinuous galerkin fem multi-physics solver for the molten salt fast reactor. En: International Conference on Mathematics and Computational Methods applied to Nuclear Science and Engineering. M&C.

Tuominen, R., Valtavirta, V., Peltola, J., Leppänen, J., 2016. Coupling serpent and openfoam for neutronics - cfd multi-physics calculations. En: PHYSOR 2016: Unifying Theory and Experiments in the 21st Century. American Nuclear Society (ANS), United States, pp. 255–269, sDA: SHP: SASUNE Nuclear Project : 100502 ; International Conference on the Physics of Reactors, PHYSOR 2016 : Unifying Theory and Experiments in the 21st Century, PHYSOR 2016 ; Conference date: 01-05-2016 Through 05-05-2016.

Valocchi, G., Tommasi, J., Ravetto, P., 2020. Reduced order models in reactor kinetics: A comparison between point kinetics and multipoint kinetics. Annals of Nuclear Energy 147, 107702.
URL: `https://www.sciencedirect.com/science/article/pii/S030645492030400X`
DOI: `https://doi.org/10.1016/j.anucene.2020.107702`

Valtavirta, V., 2016. Multiphysics capabilities of serpent 2. En: Serpent Workshop at PHYSOR 2016. PHYSOR 2016.

Weller, H., Tabor, G., Jasak, H., Fureby, C., 1998. A tensorial approach to computational continuum mechanics using object-oriented techniques. Computers in Physics 12 (6), 620–631.
URL: `https://aip.scitation.org/doi/abs/10.1063/1.168744`
DOI: `10.1063/1.168744`

Zhang, H., Andrs, D., Hansel, J., Zou, L., 2018. Relap-7 user's guide. Tech. rep., Idaho National Lab. (INL), Idaho Falls, ID.
DOI: `10.2172/1466669`

Zhong, Z., Talamo, A., Gohar, Y., Jul 2011. One-run monte carlo calculation of effective delayed neutron fraction and area-ratio reactivity.

Chemical Engineering Science 00 (2021) 1–11

# Computational Fluid Dynamics Modelling of the DYNASTY Loop

A. Nalbandyan[a], A. Cammi[b], S. Lorenzi[b], E.B.Klinkby[a], B.Lauritzen[a]

[a] *DTU Physics, Frederiksborgvej 399, 4000 Roskilde, Denmark*
[b] *Politecnico di Milano, Department of Energy, Via Lambruschini, 4, 20156 Milan, Italy*

## Abstract

In this paper, CFD assessment of the DYNASTY natural circulation loop, adopting RANS-based turbulence models is performed using the OpenFOAM open source toolbox. The CFD simulation results are benchmarked against experimental data and compared to the results of analytical and numerical approaches. The DYNASTY facility is designed to investigate the stability and dynamics of heat-generating fluids, in particular molten salts, in a natural or forced circulation regime and as such, it is a first in kind large scale natural circulation facility built for molten-salts. In this work, the aim is to setup a CFD model of the facility and to validate the model by comparing the modelling results to experimental data obtained during the initial testing campaign of the facility, with water as working fluid. In particular, the equilibrium state of the system is investigated and studied in terms of the mass flow rate dynamic behaviour and the temperature difference across the cooler section of the loop. It is shown that the CFD simulations adopting the $k - \omega \; SST$ turbulence model best reflects the experimental results. The CFD results are in agreement with the 1D modeling as well as the analytical solution.

*Keywords:*
DYNASTY loop, Natural Circulation, CFD, Validation and Benchmarking

## Nomenclature

| | |
|---|---|
| $\alpha$ | thermal diffusivity ($m^2/s$) |
| $\nu$ | dynamic viscosity ($Pas$) |
| $\rho$ | density ($kg/m^3$) |
| $\tau_{ij}$ | viscous stress tensor ($Pa$) |
| $\tau_{tij}$ | turbulent stress tensor ($Pa$) |
| $\zeta$ | friction coefficient (-) |
| $C_f$ | skin friction factor (-) |
| $D$ | Darcy coefficient (-) |
| $F$ | Forchheimer coefficient (-) |
| $g$ | gravitational acceleration ($m/s^2$) |
| $h$ | fluid enthalpy ($J$) |
| $k$ | fluid kinetic energy ($J$) |
| $Nu$ | Nusselt number (-) |
| $Pr$ | Prandtl number (-) |
| $Q$ | heat source ($W$) |
| $Re$ | Reynolds number (-) |
| $Re_\theta$ | transition Reynolds number (-) |
| $RS$ | radiation heat source ($W$) |
| $T_f$ | fluid temperature ($K$) |
| $T_w$ | wall temperature ($K$) |
| $u$ | velocity ($m/s$) |
| $U_\tau$ | fluid friction velocity ($m/s$) |
| $u_r$ | relative velocity ($m/s$) |
| $y^+$ | dimensionless wall distance (-) |

*Email Addresses:* `asnalb@dtu.dk` (A. Nalbandyan), `antonio.cammi@polimi.it` (A. Cammi), `stefano.lorenzi@polimi.it` (S. Lorenzi), `esbe@dtu.dk` (E.B.Klinkby), `blau@dtu.dk` (B.Lauritzen)

## 1. Introduction

Passive systems operating without active driving components, such as pumps, are of interest for engineering applications where system autonomy and reliability have to be ensured. This is particularly relevant in the nuclear sector, where power plant safety functions (reactivity control, core and containment cooling, prevention of radioactive release, ...) should be ensured in different situations, ranging from operational conditions to accidental scenarios [1]. For example, the cooling function, and related passive decay heat removal systems, play a relevant role in increasing the reliability of a nuclear reactor, as highlighted in Fukushima event.

Some advanced Generation III and III+ water reactors such as the AP1000 and the ESBWR as well as the Generation IV reactors foresee passive systems for core decay heat removal after reactor shutdown [2], [3],[7]. These kind of passive systems rely on natural circulation features in order to ensure the cooling capability. Being the absence of a pump the main advantage - in terms of passive safety but also in terms of simplicity and cost reduction -, as main drawback, the natural circulation systems are prone to unstable behavior due to the strong coupling between the buoyancy forces (that depends on density differences) and friction forces [8]. The equilibrium stability of a natural circulation loop for a single phase flow can be altered by the nature of the heat source and the cold sink as well as the friction in the loop. Whereas in the majority of the cases the heat source and the heat sink are localized, for some specific applications, distributed heat source might be present in the loop, e.g. due to the internal heat generation by the working fluid. This situation is particularly relevant to the Generation IV Molten Salt Reactor (MSR), where the liquid fuel acts also as coolant causing the fission heat generation to be internally delivered in the fluid [4]. The unstable behaviour is characterized by large oscillations of the main flow parameters such as the mass flow rate and temperature. This is not a desirable situation for any engineering application since the system can reach conditions that may jeopardize the structural components. Thus, it is important to be able to correctly predict the regime of natural circulation systems. In this context, a useful equipment is represented by a Natural Circulation Loop (NCL), i.e., a rectangular or a toroidal loop with a heat sink and a heat source where the fluid flow is driven by natural circulation as balance of inertial, friction and buoyancy forces [8]. The simplicity of these systems allows focusing on the physical and phenomenological basis of the buoyancy-driven convection and represents an ideal validation benchmark for testing modelling capabilities. Among the NCLs, the DYNASTY facility [9] located at Politecnico di Milano (Figure 1) is aimed at investigating the stability and the dynamics of natural circulation in presence of distributed heating, in particular to investigate the decay heat removal mechanisms for the Molten Salt Fast Reactor[4]. Previous studies on the stability of single-phase NCLs with both localized and distributed heat sources applying analytical and numerical methods [3], [4], indicate, that natural circulation with internal heat generation is potentially more prone to the instability with respect to the case with localized sources. Thus it is of interest and practical importance to further investigate the stability of NCLs with distributed heating. In order to gain in depth information on the flow behavior inside the facility, in this paper a CFD model of DYNASTY - based on the OpenFOAM CFD framework - is presented with a comparison of several Reynolds-Averaged Navier Stokes (RANS) turbulent models. The CFD approach allows investigating 3D phenomena as well as the radial profile of temperature which plays a relevant role in case of distributed heating. As a major outcome, the CFD model is validated against experimental results carried out using water as working fluid. These experimental data represent the first data obtained with DYNASTY in distributed heating mode. As a further verification, the modeling results are also compared to previously developed and tested methods, e.g. the stability maps and the 1D Object Oriented Modeling approach [3]. As such, this work presents a unique study of an NCL with distributed heat source comparing not only analytical and 1D modelling methods but also a full-scale detailed CFD model.

The paper is organized as follows. In Section 2, the description of DYNASTY facility is provided, followed by a detailed description of the CFD model whereas in Section 3: the model geometry, meshing approach, the turbulence modeling, modeling of pressure and heat losses as well as the initial and boundary conditions together with the numerical schemes used by the solver are presented. Section 4 refers to the numerical and analytical modeling approaches and finally Section 5 presents and discusses the simulation results compared to the experiment, followed by a conclusion given in Section 6.

## 2. Description of DYNASTY Facility

DYNASTY is a large NCL facility designed to operate with molten salt as the thermal carrier but with the flexibility to run with multiple thermal carriers as water or glycol. All the components are made of stainless steel (AISI 316) to withstand the operational temperatures. The main geometrical dimensions of the facility and operational parameters for water are summarized in Table 1.

A schematic view of DYNASTY is presented in Figure 2. The facility is divided into five sections, namely the cooler, the downcomer (pipe 1), the riser (pipe 3), the horizontal leg (pipe 2), and the pump leg. The two vertical legs are labelled riser and downcomer, assuming a clockwise flow of fluid inside the loop. The bottom part of DYNASTY presents two parallel sections (horizontal leg and pump leg) to be used for natural circulation (NC, as in this work) and forced circulation experiments respectively. Each of the two sections can be isolated from the rest of the loop through valves placed at both ends of the section. The

| Parameter | Dimension | Unit |
|---|---|---|
| Height | 3.0900 | m |
| Width | 3.1000 | m |
| Pipe inner diameter | 0.0382 | m |
| Pipe wall thickness | 0.0020 | m |

Table 1: Main geometrical data for DYNASTY

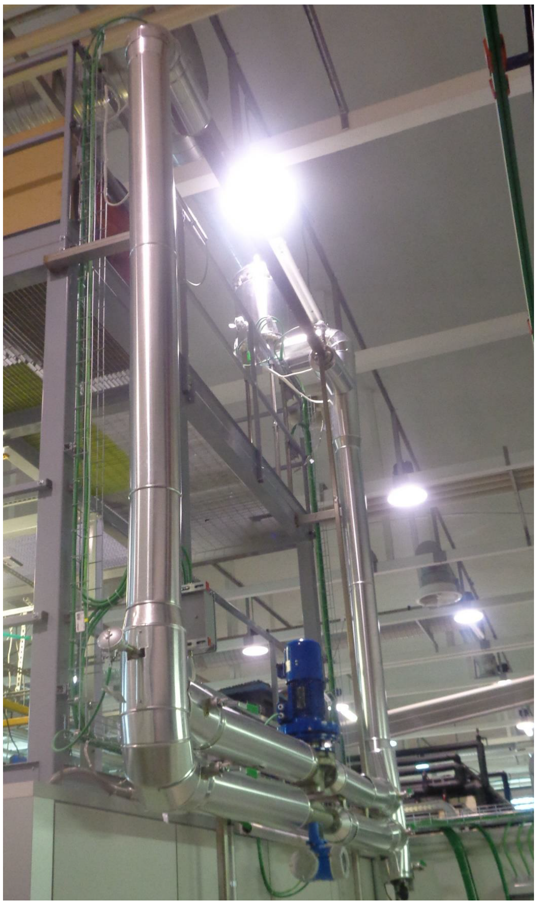


Figure 1: DYNASTY facility

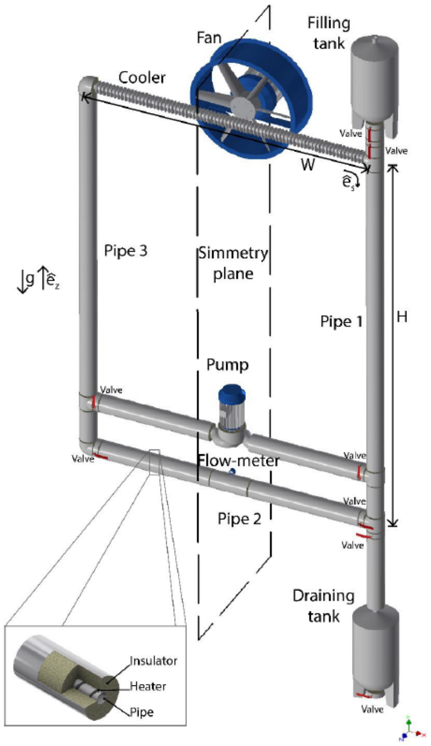


Figure 2: Schematic view of DYNASTY

DYNASTY cooler is the top horizontal section and it is a finned pipe coupled to a fan in cross-flow, with the air being driven to the cooler at ambient temperature.
Heat is provided by electrical strips installed on the exterior of the pipes to mimic the internal heat generation. On the outside of the pipes and the heating system, a rockwool thermal insulator is applied to reduce thermal dispersion to the environment. The heating elements are grouped in four sections GV1, GV2, GO1 and GO2 that are responsible for heating the riser and top part adjacent to it, the downcomer and the top part adjacent to it, the horizontal leg and the pump leg (Figure 3). Each of these sections can be powered independently from one another, and the provided power can be distributed between the sections. The cooler section cannot be heated and thus does not have any heating stripes. Thanks to the independent powering of each leg of the system, it is possible to operate the facility with a wide range of power distributions ranging from localized configuration (e.g., Horizontal Heating Horizontal Cooling or Vertical Heating Horizontal Cooling) to distributed heating condition powering all three legs.

| Parameter | Value | Unit |
|---|---|---|
| Pressure (filling tank outlet) | 101325 | Pa |
| Temperature range | 299.15-353.15 | K |

Table 2: Main operational data for DYNASTY (water as working fluid)

As for the instrumentation, DYNASTY is equipped with temperature sensors for the fluid and for the pipes, and with a Coriolis-effect mass flow meter on the bottom horizontal leg. The fluid temperature is measured with four resistance temperature detectors (TC1, TC2, TC3 and TC4) and are placed in positions relevant with respect to the heat exchange section, i.e., cooler inlet, cooler outlet, downcomer outlet and riser inlet (Figure 3). These locations are relevant since they allow computing the fluid temperature difference, which is the driving force of NC for the different heating configuration. In case of distributed heating, the driving fluid temperature difference would be TC2-TC1 (assuming counterclockwise flow) as the thermal gap across the cooler.The favorable flow direction can be determined by the provided power difference between the vertical legs.

## 3. The CFD model

As stated in [3], the stability of natural circulation is influenced by the heat exchange between the fluid and the solid wall. To this aim, a Conjugate Heat Transfer (CHT) approach is employed for the CFD model. In particular, the *chtMultiRegionFoam* heat transfer solver [16], available in the OpenFOAM v1806 version, is used in this work. The main equations [15] for the fluid body are:

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot \rho u \tag{1}$$

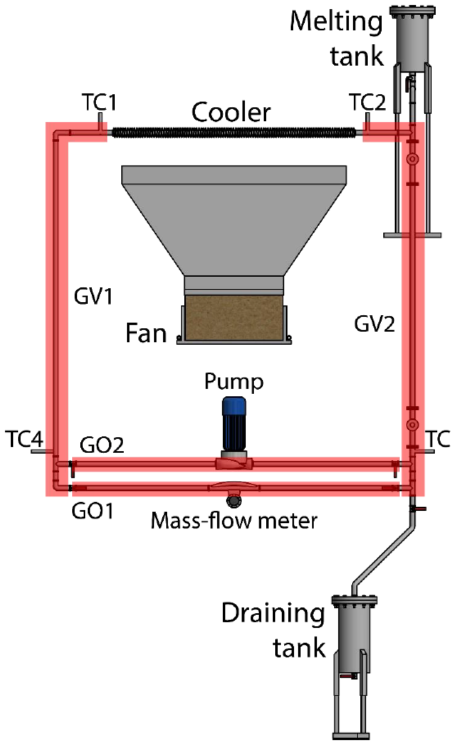


Figure 3: Heating and instrumentation in DYNASTY facility

$$\frac{\partial \rho u}{\partial t} + \nabla \cdot (\rho u \times u) = -\nabla \cdot \tau - \nabla p + \rho g \tag{2}$$

$$\frac{\partial(\rho h)}{\partial t} + \nabla \cdot (\rho u h) + \frac{\partial(\rho k)}{\partial t} + \nabla \cdot (\rho u k) - \frac{\partial p}{\partial t} = -\nabla \cdot q + \rho g \cdot u - \nabla \cdot (\tau \cdot u) + \rho Q + RS \tag{3}$$

Equation (1) is the mass conservation equation where $\rho$ is the fluid density, $u_x$, $u_y$, $u_z$ are the velocity components. Equation (2) is the momentum conservation equation, $g$ is the gravitational acceleration, $\tau$ is the stress tensor. Equation (3) is the energy conservation equation, where $h$ is the fluid enthalpy, $k$ is the kinetic energy, $q$ is the heat transferred by conduction, Q is the heat source term and the RS is the term representing heat due to radiation.

For the solid steel pipes, only the energy conservation equation is discretized and solved:

$$\frac{\partial(\rho h)}{\partial t} = \nabla(\alpha \nabla h) \tag{4}$$

where $h$ is the specific enthalpy and $\alpha = k/C_p$ is the thermal diffusivity defined as the ratio between the thermal conductivity and the specific heat capacity for the steel. The coupling between the solid and fluid bodies is realized by means of additional conditions, at the fluid to solid interface the temperatures are assumed equal:

$$T_f = T_w \tag{5}$$

Similarly, the heat flux entering the domain in one region should be equal to the heat flux leaving the domain in the other region:

$$k_f \frac{\partial T_f}{\partial n} = k_s \frac{\partial T_w}{\partial n} \tag{6}$$

where $k_f$ and $k_s$ are the fluid and solid thermal conjunctivitis respectively, $T_f$ is the fluid temperature next to the wall, $T_w$ is the solid wall temperature, and $n$ is the directional normal to the wall.

Note, that the CFD model models steel pipes but for the sake of simplicity the heating stripes and the insulator are only mimicked via appropriate boundary conditions. For modeling the all-external heat flux in the heated regions of the loop, the so called *externalWallHeatFluxTemperature* [16] boundary condition is used on the external surfaces of heated pipe sections. This boundary condition can operate in three modes: heat flux, power and coefficient mode.

The power mode is used for the heated sections, whereas the cooler outer surface is set to coefficient mode with the overall heat transfer coefficient and the ambient temperature as input parameters. The heat losses are treated by reducing the net power provided to the heated sections of the loop.

In Equation (2), the shear stress term is written as a sum of viscous and turbulent components. The turbulent component, also known as Reynolds stress, is usually modelled rather than solved directly for. Direct numerical simulation (DNS) of unsteady Navier-Stokes equations often implies a high computational burden. For homogeneous turbulent flow, the DNS computational efforts are proportional to $Re^3$ and the non-homogeneous effects such as wall flow are further adding to the computational efforts [10]. This makes DNS unsuitable for the vast majority of engineering applications.

There are several ways to avoid using purely DNS approach, e.g. Large Eddy Simulation (LES) model, wherein the small scale fluctuations that do not considerably affect the solution are filtered out and approximated via modelling, whereas the large scale fluctuations are solved. Another common way around this complication is to adopt the Reynolds Averaged Numerical Simulation (RANS) approach, where the Reynolds stress variable is represented as a sum of the mean steady value and the unsteady fluctuations. These unsteady fluctuations are represented by the Reynolds stress tensor and require additional closure models to relate the stress tensor to the mean values of the flow variables. These closure models are known as turbulence models and they commonly rely on the introduction of a parameter that relates the Reynolds stress to the mean flow, called the eddy viscosity. Similar to the molecular viscosity, the eddy viscosity is responsible for internal momentum transfer via eddies that are formed in turbulent flow. The eddy-viscosity turbulence models are classified as zero up to five equation models depending on the number of transport equations they solve.

The *chtMultiRegionFoam* solver considers polynomial dependence on the temperature for the thermo-physical properties. The main thermo-physical properties for the water and the piping steel used in this work are presented in Table 3.

| Property | Water | Value for water at 60 °$C$ | Steel |
|---|---|---|---|
| Density (kg m$^{-3}$) | 1122-0.4159·T(K) | 983.44 | 8238 |
| Dynamic viscosity (Pa s) | 4.064·10$^{-3}$-1.086·10$^{-5}$·T(K) | 0.0004656 | - |
| Specific heat capacity (J kg$^{-1}$ K$^{-1}$) | 4213 | 4213 | 468 |
| Thermal conductivity (W m$^{-1}$ K$^{-1}$) | 0.2549+1.192·10$^{-3}$·T(K) | 0.652 | 13.4 |

Table 3: Thermo-physical properties of water and steel [3]

### 3.1. RANS Turbulence Modeling

In this work the two-equation eddy-viscosity turbulence models are tested, namely: *k–ω SST*, *k–ω SSTLM* and *realizable k–ϵ* models. A short description of each of these models is provided below.

#### 3.1.1. $k - \omega$ Shear Stress Transport ($k - \omega$ SST) turbulence model

This two-equation model solves for the turbulent kinetic energy $k$ and the specific turbulence dissipation rate $\omega$. The implementation adopted in OpenFOAM is based on the solution of the transport equations for $k$ and $\omega$ with adoption of a blending function which depends on the distance to the wall [11]. When moving away from the walls, this blending function has a value of 0, which corresponds to applying the $k - \epsilon$ model, while next to the wall the value of the blending function is 1 and subsequently the $k - \omega$ turbulence model is used. A second blending function is used as well, which prevents buildup of turbulence in stagnation zones. The $k - \omega\ SST$ method thus combines the advantages of normal $k - \omega$ model in the near wall region with the accurate performance of $k - \epsilon$ model in the free stream region. This model is used in many industrial flow simulations especially when flow separation or adverse pressure gradients are present.

#### 3.1.2. Langtry-Menter $k - \omega$ Shear Stress Trasnport turbulence model ($k - \omega$ SST LM)

Another turbulence model tested in this work is the Langtry-Menter four equation $k - \omega\ SST$ model also known as $\gamma Re_\theta$ model. This model has particularly been developed for unstructured CFD meshes and it is a transition model, aimed to capture flow transition from the laminar to the turbulent regime [12]. This is particularly interesting to investigate for the natural circulation regime with internal heat generation in the fluid, as the transition from laminar to turbulent for this scenario is not well investigated. The model introduces the transition $Re$ number as:

$$Re_\theta = \begin{cases} 1173.51 - 589.428Tu + \frac{0.2196}{Tu^2} & Tu \leq 1.3 \\ \frac{331.5}{(Tu - 0.5658)^{0.671}} & Tu > 1.3 \end{cases} \tag{7}$$

where the $Tu$ factor depends on the free stream velocity and the turbulent kinetic energy and is defined as :

$$Tu = \frac{\sqrt(2k/3)}{|u_\infty|} \tag{8}$$

The model performs well at the low-Re numbers and in predicting flow transition [13]. Based on the experimental data and the main geometrical parameters of the loop, the transition $Re$ number is calculated to be 584, which again points to the fact that in the case of natural circulation with distributed heating the transition $Re$ number can be much lower than e.g. for an infinitely long straight pipe, for which the transition $Re$ is around 2300 (for water).

#### 3.1.3. Realizable $k - \epsilon$ turbulence model

Finally, the third model tested in this work is the realizable $k - \epsilon$ turbulence model. It is a two equation model solving for the turbulence kinetic energy $k$ and the energy dissipation rate $\epsilon$.

$$\epsilon = \frac{C_\mu^{(0.75)} k^{(1.5)}}{l} \tag{9}$$

In contrast to the standard $k - \epsilon$ model this model adopts a new, more exact transport equation for the dissipation rate and assumes the turbulent viscosity being a function of the mean flow properties.

- The *realizable* $k - \epsilon$ model is more accurate and reliable for many application than the standard $k - \epsilon$ model.
- It is especially success full in describing complex flows with rotation, vortexes and stagnation zones.

### 3.2. Model geometry and Mesh

The geometrical model of DYNASTY (depicted in Figure 4) features several simplifications compared to the actual geometry presented in Figure 2. The pump leg is not modeled because in the scope of this work only natural circulation regime is considered. The mass flow rate meter is included by means of porous medium instead of modeling the actual complex structure of the device as the only impact on the flow is the pressure loss caused by the mass flow rate meter and this can be modeled using the vendor provided information. The draining tank is not modeled as well. The finned structure of the cooler is also not included in the CFD model for the sake of simplicity. The impact of the fins is taken into account in the calculations of the heat transfer coefficient.

The meshing is facilitated using ANSYS 19.2 Workbench [19]. The cooler section which includes the water filling tank, is sliced, so that the largest part of it can be meshed using hexahedral structured grid, meanwhile the tank itself is meshed using an unstructured tetrahedral mesh (Figure 5).

Meshes of different coarseness are generated to test the model for mesh independence purpose. The mesh average orthogonal quality and the maximum skewness are reported in Table 4 and are complying with standard meshing guidelines for ANSYS

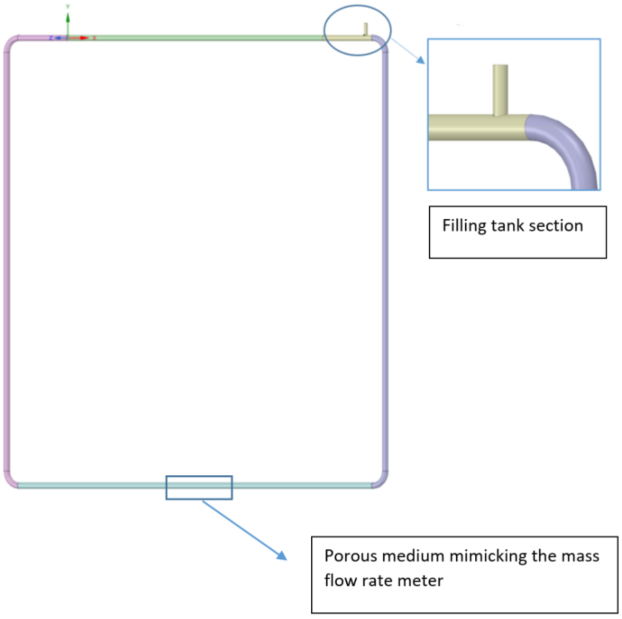


Figure 4: CFD model of DYNASTY

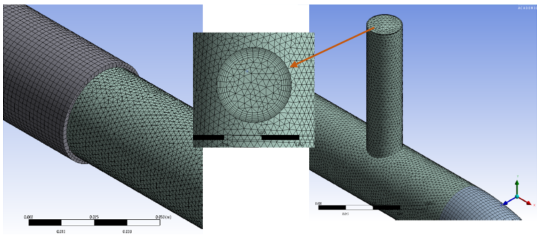


Figure 5: DYNASTY mesh: Closeup at inflation layers applied on water body

Workbench suggesting an average orthogonal quality above 0.2 and a maximal skewness less than 0.95. These two characteristics are very important to control the mesh quality, as they show how close the mesh elements are to the optimal size and shape.

Another important meshing criterion is the dimensionless wall distance otherwise known as the wall $y^+$ value. It is defined as

$$y^+ = \frac{\rho U_\tau \Delta y}{\mu} \tag{10}$$

where $\Delta y$ is the first cell height of the mesh boundary layer, $\rho$ is the density, $\mu$ is the dynamic viscosity and $U_\tau$ is the frictional velocity which can be calculated as:

$$U_\tau = \sqrt{\frac{C_f}{2}}U \tag{11}$$

where U is the free stream velocity and $C_f$ is the skin friction factor; an empirical coefficient determined for internal flows as [14]:

$$C_f = 0.079Re^{-0.25} \tag{12}$$

The dimensionless wall distance helps describing the near wall flow, which is subject to numerical and modelling challenges due to the viscosity induced effects. Usually, the near wall flow can either be modelled by resolving the viscous sub-layer, or by adopting wall functions to approximate the flow behavior across it. If the viscous sub-layer is to be resolved, very fine mesh is usually required, with a $y^+ \approx 1$. This is frequently not viable for a large industrial model and thus wall functions are generally adopted for near wall flow modelling. The validity of applying wall functions depends on the flow regime (Re number) and the turbulence model being used and in this work it is ensured that the wall $y^+$ value is consistent with the turbulence model applied.

| Mesh | Coarsest | Coarse | Fine | Finest |
|---|---|---|---|---|
| Element Number | $\approx 1{\cdot}10^6$ | $\approx 4{\cdot}10^6$ | $\approx 6{\cdot}10^6$ | $\approx 8{\cdot}10^6$ |
| Skewness | 0.68 | 0.87 | 0.84 | 0.81 |
| Orthogonal quality | 0.96 | 0.947 | 0.950 | 0.93 |

Table 4: Mesh Data

### 3.3. Pressure loss modeling

The pressure losses in the loop are mainly due to the water filling tank and the mass flow rate meter. These localized pressure drops can be written as:

$$\Delta P = \zeta\frac{\rho U^2}{2} \tag{13}$$

where $\zeta$ is the resistance coefficient. In the OpenFOAM model the pressure losses can be introduced as a porous zone with the Darcy-Forchheimer model [17]:

$$\nabla P = (\mu D + \frac{1}{2}\rho F U)U = \mu D U + \frac{1}{2}\rho F U^2 \tag{14}$$

where $D$ and $F$ are the friction coefficients; $D$ is for the viscous losses (also known as the Darcy component) and $F$ is for inertial losses (also known as the Forchheimer component).
For e.g. pressure loss in x direction:

$$\Delta P = \Delta x(\mu D U + \frac{1}{2}\rho F U^2) = dU + fU^2 \tag{15}$$

For the water filling tank the pressure loss is calculated using empirical correlations available in hydraulic handbooks [10] as a function of the flow characteristics, mainly the Re number as well as the roughness of the pipes. For the mass flow rate meter, the mass flow rate dependent pressure drop correlation is provided by the vendor. Including the detailed structure of the mass flow rate meter in the CAD model for the CFD simulations is not convenient, as it will add complexity to the meshing. The only impact of the mass flow rate meter on the flow is in terms of the pressure losses and these can well be modeled using porous medium approach.

The $D$ and $F$ coefficients can be calculated from Eq.14, if the velocity dependent pressure loss is known, as follows:

$$D = \frac{d}{\mu \Delta x} \tag{16}$$

$$F = \frac{2f}{\rho \Delta x} \tag{17}$$

### *3.4. Case initialization*

A transient experimental test case is modelled in this work, wherein the total power provided to the loop is 450 *Watt*, distributed 2/3 to the hot leg and 1/3 to the cold leg. The heat losses are taken into account by decreasing the net power provided to the heated legs. The cooler is modeled by providing the convective heat transfer coefficient calculated based on semi-empirical correlations [5].

The solver accounts for the conjugate heat transfer between solid and fluid regions as well as the buoyancy and the turbulence effects adopting a segregated solution strategy; first the equations for the fluid region are solved followed by the solution for a solid region. For the fluid region, the PIMPLE algorithm [15] with three correctors is adopted for pressure,velocity and energy equations, meaning that the pressure is corrected three times within the PIMPLE loop. The outer correctors are two, meaning that within a time step the PIMPLE loop is performed two times for the whole set of the equations before moving to the next time step. As for the numerical schemes, first order implicit Euler scheme is used for the time derivative terms, second order unbounded Gauss linear and first order bounded Gauss upwind schemes are used for the divergence terms.

For the boundary conditions, a special temperature coupling boundary condition available in the OpenFOAM, *TemperatureCoupledBaffleMixed* [16], is used for the temperature on all contact zones between the fluid and the solid.This boundary condition represents the coupling condition for the temperature given by Eq.6. On the tank outlet an *inletOutlet* boundary condition is used. This boundary condition usually behaves as a zero gradient Neumann boundary condition, except when there is backflow into the domain; then the *inletOutlet* boundary condition changes to a fixed value to prevent a non-physical flow reentry situation. The boundary condition for the velocity in the liquid zone is set to uniform zero fixed value on all boundaries and to *pressureInletOutletVelocity* on the tank outlet boundary; this boundary condition applies a zero gradient condition on the outflow and for the inflow a velocity derived from an internal cell values is applied. For the cooler external surface a coefficient boundary condition mode is used: the overall heat transfer coefficient and the ambient temperature are specified.

Both the fluid and the solid zones are initially set to 333.15$K$ temperature, and the fluid velocity is set to 0.

## 4. Analytical and Object Oriented Modeling Approaches

An additional verification of the CFD model is performed by comparing the results to the simulations performed with 1-D object oriented DYNASTY model and semi-analytical calculations based on stability maps [9]. In this section the two methods are described briefly.

### *4.1. Stability Maps*

Stability maps are a simple but powerful tool aimed at provide information about the stability of natural circulation over a large range of conditions, being very useful in the design phase. They are graph drawn in the space spanned by two parameters (e.g., Reynolds and Prandtl number) where a neutral stability curve separates the region of asymptotically stable equilibrium points of the system from the unstable ones (Figure 6). They rely on the modal linear analysis approach and on a semi-analytical treatment based on some simplifying assumptions (incompressible fluid and mono-directional, same flow regime in any point of the loop, Boussinesq approximation). Differently from CFD simulations, the stability maps require pressure drop and heat transfer correlations for the analysis. Starting from the conservation equations for mass, momentum and energy for the fluid and energy equation for the solid wall, a steady state can be found (in terms of mass flow rate and temperatures) which correspond to the equilibrium point investigated by the stability map. The stability analysis involved the linearization of the time-dependent version of the aforementioned equation with the solution of every state variable represented as a steady-state solution plus a time dependent perturbation. The perturbation of a generic state variable can be written in the following form:

$$\delta\Gamma(s,t) = \Gamma_0(s) + \Gamma(s)e^{(\omega t)} \tag{18}$$

being $\omega$ the complex pulsation (or mode) of the perturbation. If the Equation 18 is substituted into the governing equation, the system can be solve for $\omega$. If the real part of $\omega$ is greater than zero, the amplitude of a perturbation grows exponentially in time, hence the internal dynamics of the NCL amplifies perturbations and the steady state considered is unstable. On the contrary, if the real part of $\omega$ is negative, the exponential is decreasing in time, the perturbation is dampened, and the steady state is stable. $\omega = 0$ represent the neutrally stable curve. Detailed description of stability maps derivation process can be found in e.g. [3], [4].

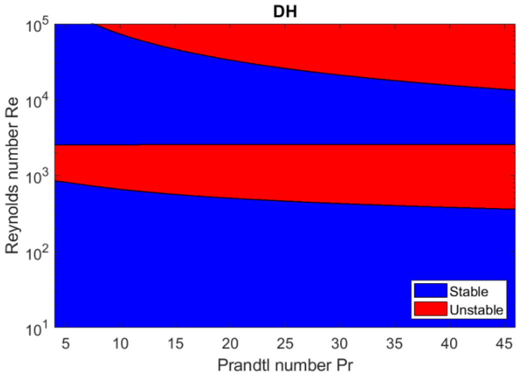


Figure 6: DYNASTY stability map example: distributed heating configuration, molten salt as working fluid

### 4.2. 1D Object Oriented Modeling

One drawback of the stability map approach is the limitation to an asymptotic analysis. To overcome this problem, and to be able to track the time evolution of the main variable of interest, a 1D modelling has been developed in the past, based on the Modelica object oriented language in the Dymola environment. This approach consists in the solution of the one-dimensional, time dependent, non-linear governing equation (mass, momentum and energy balance for the fluid, and energy balance for the solid wall). Similar to the stability maps, also the one dimensional approach rely on pressure drop and heat transfer correlations. Modelica is an object-oriented, acausal, equation-based language used to simulate physical systems. It relies on a wide range of validated libraries containing the model of different multi-engineering components (thermal hydraulics, electrical, mechanical, ...). The 1D object-oriented DYNASTY model is presented in Figure 17. The thermal hydraulics components of the model are taken from the ThermoPower library [18] along with other components specifically developed for distributed heating and collected in the ThermoPowerIHG library. The development and validation of the model is described in [3]. The main component in the model is represented by a pipe, which implements the mass, momentum and energy balance equations for the fluid which is provided with a model for the metal tube and for the volumetric heat source. Additional components are added in order to represent the cooler and the fan, and the localized pressure drop caused by the mass flow meter, the elbow and the junctions. Pressure drop and heat transfer are modeled through dedicated semi-empirical correlations, e.g. the Darcy friction factor for the transition zone is modeled by employing the correlation derived in [4], whereas for determining the convective heat transfer coefficient the correlation for $Nu$ number according to Churchill-Bernstein is used [5].

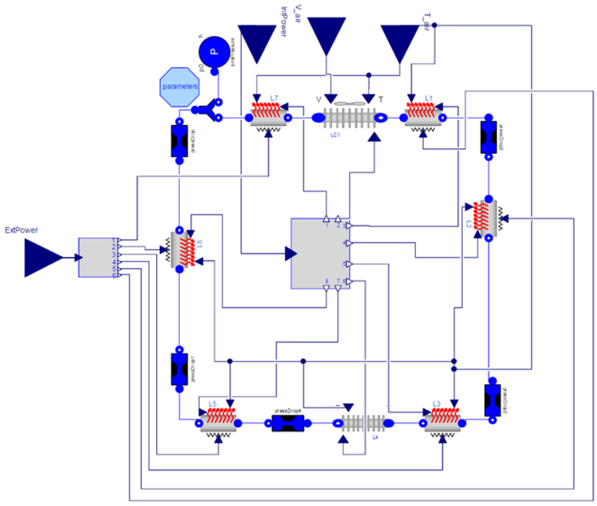


Figure 7: 1-D model of the facility

## 5. Results and Discussion

The initial temperature of the water as well as the solid pipes is set to 60 ° C. The power provided to the system is 450 Watt (considering the losses), distributed in ratio of 2/3rd to the hot leg and 1/3rd to the cold leg. The flow evolution as a function of time is investigated to establish whether the flow stabilizes i.e., whether the mass flow rate and the temperature difference across the cooler stabilize around some mean value.

Mesh independence is checked by comparing the average mass flow rate in the loop for different meshes and is reported in Figure 8. The fine mesh with $6E + 06$ elements is chosen for the simulations. The simulations are run on a HP Proliant SL230 Gen. 8 cluster with Intel Ivy-Bridge Xeon e-5-2880v2 2.GHz processors. The CPU time is reported in Table 5 per each turbulent model, for the same mesh.

In Figure 9 the mass flow rate of the water is shown as a function of time, with the last 1000 s depicted in Figure 10 for the purpose of more clear comparison. The CFD results are compared to the experimental data and Modelica results. Both CFD and Modelica simulations reflect the initial oscillation of the mass flow rate due to the typical initial transient of the natural circulation where hot and cold fluid plug start circulating in the circuit. As for the turbulence modelling comparison, the realizable $k-\epsilon$ turbulence model overestimates the peak mass flow rate value by almost two times both at the initial largest peak around 500 s and for the second one occurring around 900 s. On the other hand, the two $k-\omega$ SST models seem to better reproduce this initial transients also in terms of characteristic frequency. However, it should be pointed out that the exact values of the maximum and minimum peak strongly depends on the initial conditions of the system, especially in terms of the turbulence parameters. In this light, a more appropriate figure of merit for the validation of the CFD model are the steady-state values. The mass flow rate stabilizes starting from t=2000 s, with the realizable $k-\epsilon$ turbulence model further exhibiting some small oscillations and finally stabilizing after 4000 s with the average stable mass flow rate being 0.034 kg/s. This is ≈ 16 % higher than the experimental average of 0.029 kg/s. The

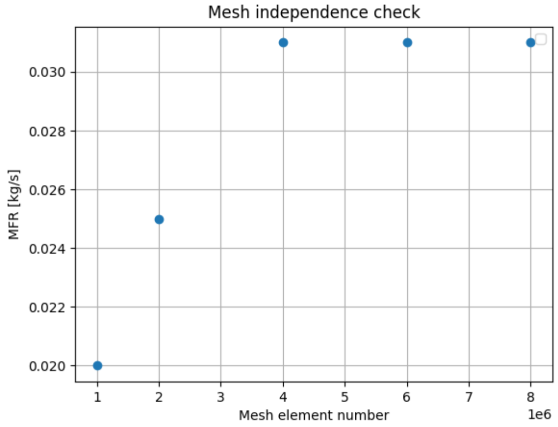


Figure 8: Mesh Independence check

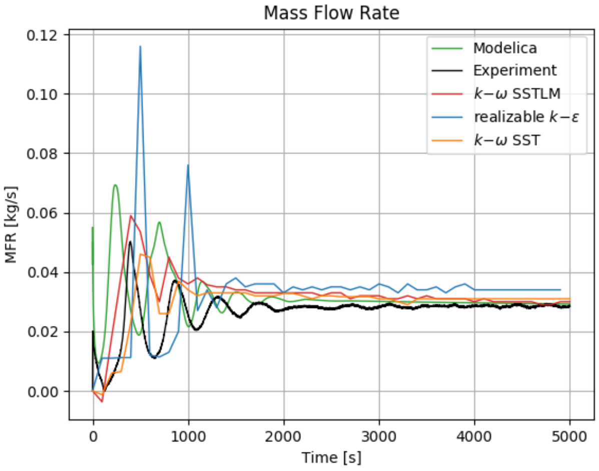


Figure 9: Mass flow rate

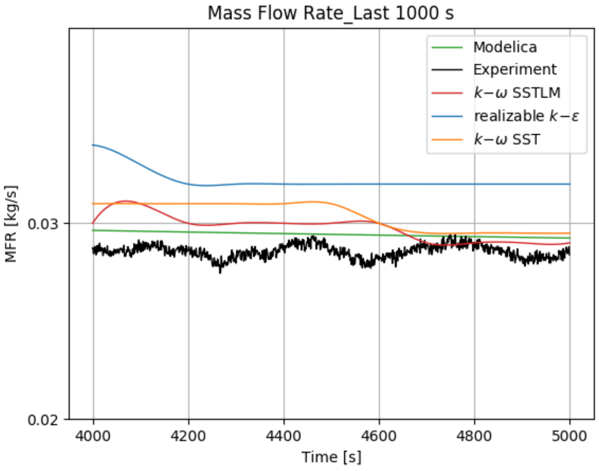


Figure 10: Mass flow rate for the last 1000 s of the simulation

$k-\omega$ models are in better agreement with the experimental results, overestimating the mass flow rate less than 10 %. The relative error calculation is done based on the results from 2000 s to 5000 s interval, where the mass flow rate does not exhibit large oscillations (see Table 6 for a result overview). The evaluation of the stabilized mass flow rate is strongly dependent on the estimation of the temperature and the pressure drop.

In Figure 11 the temperature difference between the cooler inlet and outlet sections is depicted, with a close-up view on the last 1000 s provided in Figure 12. Here the difference between the CFD and Modelica compared to the experiment is larger, with the $k-\omega$ turbulence models results having an average difference of 2 K (see Table 7 for a result overview) and the realizable $k-\epsilon$ having a slightly lower difference of 1.6 K due to the prediction of higher mass flow rate. These discrepancies between the CFD and the experimental results are most likely due to the evaluation and the modeling of the heat losses. A more precise heat loss model implementation can help improve of the current results, however for the first assessment of the CFD model an overall agreement in the temporal behavior of the heat transfer across the cooler is achieved. The comparison of the CFD results to the experiment suggests, that the realizable $k-\epsilon$ turbulence model is not a suitable turbulence model for this case, most likely due to the fact that it is intended for high Re number flows, it is sensitive for initial conditions and in general less stable than other turbulence models; it exhibits larger initial oscillations and overestimates the mass flow rate significantly.

| Turbulence model | Wall time (h) | Processors number |
|---|---|---|
| $realizablek-\epsilon$ | 30 | 64 |
| $k-\omega SST$ | 35 | 64 |
| $k-\omega SSTLM$ | 48 | 64 |

Table 5: Simulation CPU time for each turbulence model

| Case | Mass flow rate (kg/s) | Relative error |
|---|---|---|
| Experiment | 0.0290 | - |
| $k-\omega$ SST | 0.0311 | 7.2% |
| $k-\omega$ SSTLM | 0.0310 | 6.9% |
| realizable $k-\epsilon$ | 0.0337 | 16.2% |
| Modelica | 0.0300 | 1% |

Table 6: Average stabilized mass flow rate compared to the experiment

The Modelica on the other hand delivers the closest results to the experiment probably because of the way the heat losses are modeled, by taking into account the convection and irradiation that occur on the outer shell of the pipe.

In Figure 13 and Figure 14, the velocity and temperature profiles (at 1000 s) obtained with the $k-\omega SST$ CFD model are shown. The CFD approach is able to highlight the strong stratification occurring in the cooler that may impact the heat exchange in this relevant part of the circuit.

In Figure 15 the localized pressure losses due to the tank and the mass flow rate meter as a function of time are shown. Only the modelling and simulation results are compared, as the pres-

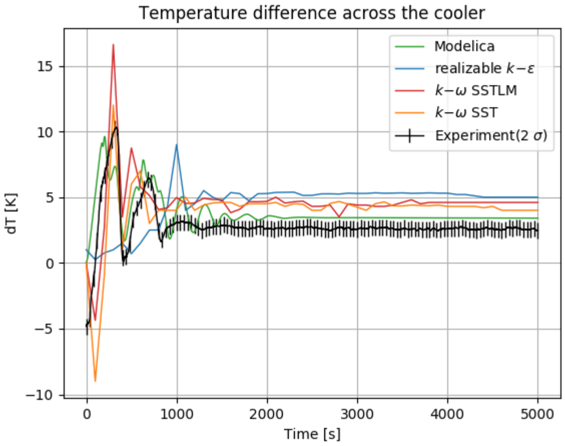


Figure 11: Temperature difference across the cooler

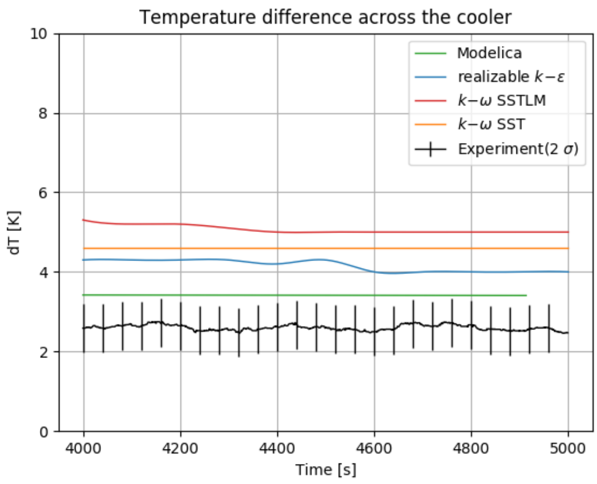


Figure 12: Temperature difference across the cooler for the last 1000 s

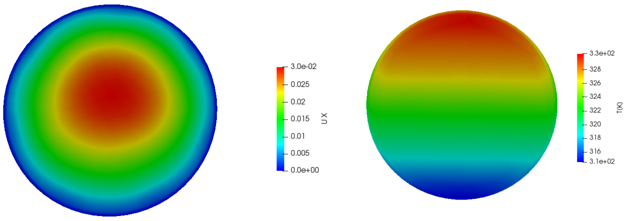


Figure 13: Velocity (sx) and temperature (dx) profiles at the cooler inlet at t=1000 s for $k - \omega SST$ turbulence model

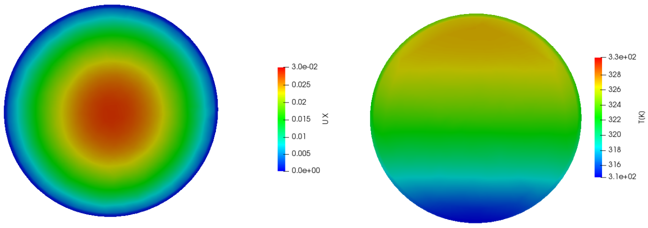


Figure 14: Velocity (sx) and temperature (dx) profiles at the cooler outlet at t=1000 s

sure losses are not experimentally measured. While $k - \omega SST$ and $k - \omega SSTLM$ are in good agreement with Modelica, the $realizablek/\epsilon$ overestimates the pressure loss and exhibits oscillations even after the three other models converge. This is in agreement with the mass flow rate overestimation by the $realizablek/\epsilon$ model. In Figure 16 the adiabatic mixing temperature (weighted average of the equilibrium) temperature in the loop is shown for the stabilized flow regime. Some deviation between the CFD model and Modelica is present, the CFD predicting around 2 K higher temperature in the cold and the hot legs. However, both models show that stability in terms of energy balance is established. The deviation can be contributed to the initial conditions and the heat loss modeling, as well as the impact of the turbulence models in the CFD part.

| Case | Average $\Delta T$ (K) | Difference (K) |
|---|---|---|
| Experiment | 2.7±0.7 | - |
| realizable $k{-}\epsilon$ | 4.3 | 1.6 |
| $k{-}\omega$ SSTLM | 5.0 | 2.3 |
| $k{-}\omega$ SST | 4.5 | 1.8 |
| Modelica | 3.5 | 0.8 |

Table 7: The average stabilized temperature difference

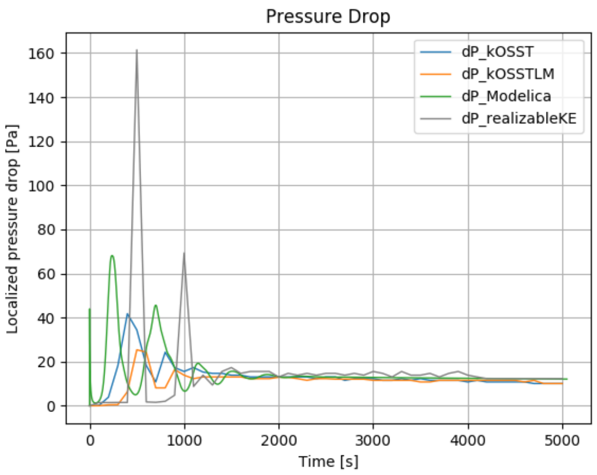


Figure 15: Pressure drop in the loop

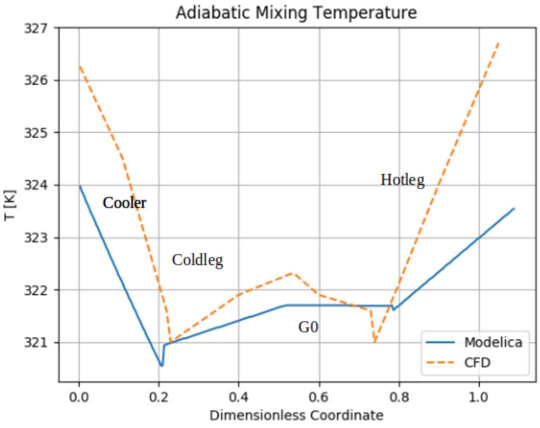


Figure 16: Adiabatic mixing temperature for stabilized flow

The Re and Pr numbers are calculated based on the average stabilized MFR values and are 1145 and 3.45 respectively. This corresponds to the stable regime on the stability map presented in Figure 8.

## 6. Conclusion

In this paper the experimental results with distributed heat source obtained from DYNASTY natural circulation loop are used to setup and validate CFD simulations of the facility in

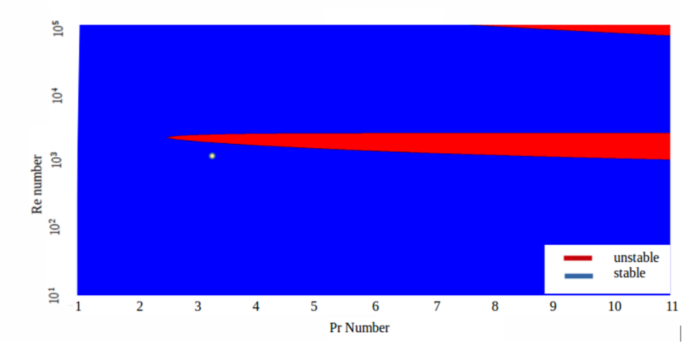


Figure 17: Stability Map

OpenFOAM. The simulation results are also compared 1-D modeling approach using Modelica object-oriented environment and analytical stability maps.The paper provides first complete assessment of a novel natural circulation facility, DYNASTY, for analyzing stability of fluids with internal heat generation. The simulation results in terms of temporal behaviour of mean flow parameters, e.g. the mass flow rate and the temperature difference across the cooler section are compared to the experiment. The objective of the study is to assess the capability of the CFD model to reflect correctly the equilibrium state of a natural circulation loop in presence of distributed heating. The mass flow rate oscillations and stabilization is very well captured by all modeling approaches. For the CFD part, three turbulence models are considered and it is shown that for the modeling of a natural circulation loop with distributed heating the $k-\omega$ SST turbulence model seems to deliver the most accurate results. Localized pressure losses are well captured with the porous media modeling adapted in CFD. The adiabatic mixing temperature profiles provide further details in the thermal balance established in the loop. The temperature difference across the cooler is overestimated by the CFD model most likely due to the heat loss not being adequately modeled. Overall, the CFD model as well as the 1-D model and the stability maps predict stable equilibrium for the considered test case scenario. It is also shown that for natural circulation regime with distributed heating the $k-\omega$ turbulence models can be suggested.

## References

[1] Progress in Methodologies for the Assessment of Passive Safety System Reliability in Advanced Reactors, IAEA-TECDOC-1752 978-92-0-108614-3, 2014

[2] J. Sierchua, Analysis of passive residual heat removal system in AP1000 nuclear power plant 2019 IOP Conf. Ser.: Earth Environ. Sci. 214 012095

[3] L. Burgazzi, Reliability of Passive Systems in Nuclear Power Plants, DOI:10.5772/47862

[4] D. Gerardin et.al, Design Evolutions of the Molten Salt Fast Reactor, International Conference on Fast Reactors and Related Fuel Cycles: Next Generation Nuclear Systems for Sustainable Development (FR17), 2017

[1] DYMOLA User Manual Volume 1,Dassault Systemes AB

[2] Casella, F., Leva, L., Modelling of thermo-hydraulic power generation processes using Modelica, Mathematical and Computer Modelling of Dynamical Systems, Vol. 12, 2006, pp. 19-33

[3] Pini, A., Cammi, A., Luzzi, L., Analytical and numerical investigation of the heat exchange effect on the dynamic behaviour of natural circulation with internally heated fluids, Chemical Engineering Science, Vol. 145, 2016, pp. 108-125

[4] Ruiz, D., Cammi, A., Luzzi, L., 2015. Dynamic stability of natural circulation loops for singlephase fluids with internal heat generation. Chemical Engineering Science 126, 573-583

[5] Bergman, T. L., Lavine, A. S., Incropera, F. P., DeWitt, D. P., 2011. Fundamentals of Heat and Mass Transfer. John Wiley  Sons, Inc., Hoboken, NJ, United States

[6] Vijayan, P. K., 2002. Experimental observations on the general trends of the steady state and stability behaviour of single-phase natural circulation loops, Nuclear Engineering and Design, Vol. 215, 2002, pp.139-152.

[7] J.N. Lillington, G.R. Kimber, Passive decay heat removal in advanced nuclear reactors, Journal of Hydraulic Research Volume 35, 1997 - Issue 6: Section on Fluid Phenomena in Energy Exchanges

[8] M. Misale, Overview on single-phase natural circulation loops, Proc. of the Intl. Conf. on Advances In Mechanical And Automation Engineering â MAE 2014.

[9] A. Cammi, L.Luzzi, M. Cauzzi, A.Pini, DYNASTY: An Experimental Loop for the Study of Natural Circulation with Internally Heated Fluids

[10] C. Hirsch, Numerical Computation of Internal and External Flows, 2nd Edition, 2007

[11] F.R. Menter, M. Kuntz, and R. Langtry. Ten years of industrial experience with the SST turbulence model. In Proceedings of the fourth international symposium on turbulence, heat and mass transfer, pages 625â632, Antalya, Turkey, 2003. Begell House.

[12] R.B. Langtry and F.R. Menter. Correlation-Based Transition Modeling for Unstructured Parallelized Computational Fluid Dynamics Codes. AIAA Journal, 47(12):2894â2906, 2009.

[13] P. Ranjan, W. J. Warton, K. A. James, A Comparison of Physics- and Correlation-Based Turbulence Models at Low Reynolds Numbers, Journal of Aircraft,2020, Vol. 57:1, pp. 1â2

[14] T. von Karman, Turbulence and skin friction, Guggenheim Aeronautical Laboratory, 1934

[15] T. Holtzman, "Mathematics, Numerics, Derivation and OpenFOAM(R)", Holzmann CFD, Leoben, fourth edition, July 2017. URL www.holzmann-cfd.de

[16] OpenFOAM API Guide v2006, *https://www.openfoam.com/documentation/guides/latest/doc/index.html*

[17] H.E.Hafsteinsson, Porous Media in OpenFOAM, Proceedings of CFD with OpenSource Software, 2008, Edited by Nilsson H., *http://dx.doi.org/10.17196/OS CFDYEAR 2008*

[18] F. Casella, A. Leva; Modelling of distributed thermo-hydraulic processes using Modelica, Proceedings of the MathMod 03 Conference, Wienna , Austria, February 2003.

[19] URL *https : //www.ansys.com*