

Improved methodology for analysis and design of Molten Salt Reactors

Présentée le 10 décembre 2021

Faculté des sciences de base
Laboratoire de physique des réacteurs et de comportement des systèmes
Programme doctoral en énergie

pour l'obtention du grade de Docteur ès Sciences

par

Rodrigo GONZALEZ GONZAGA DE OLIVEIRA

Acceptée sur proposition du jury

Prof. M. Paolone, président du jury
Dr K. Mikityuk, Dr J. Krepel, directeurs de thèse
Prof. E. Shwageraus, rapporteur
Prof. D. Lathouwers, rapporteur
Prof. H.-M. Prasser, rapporteur

Education isn't something
you can finish.
— Isaac Asimov

To my professors,
for broadening my horizons.
To my sensei,
for instilling me with strength.
To my parents,
for nurturing my dreams.

Preface

Every long term scientific project, a thesis for example, should aim to excel in 3 important aspects in my opinion: scientific, practical and didactic.

If you are reading this thesis, just as I read other thesis, you wanted to understand something that you didn't know and articles didn't or can't explain properly. By writing a thesis, I gain significant freedom of writing as long as scientific rigour is maintained. This gives me a lot of leeway in how to pass you everything that I learnt these past few years while trying to make it into a pleasant text. More than just pass information as condensed as possible, I wanted to guide you through the thought process I had, which I was told by other colleagues to be insightful — I hope it was not just politeness on their part.

To whoever is reading this to learn; more than anyone else you were always the real target of this document. I hope you have an enjoyable and informative reading.

Happy research!

Baden, 2 February 2021

R.d.O.

Acknowledgements

Jiří Křepel and Konstantin Mikityuk, for giving me all the support during the development of this work and, most of all, their patience.

Petra Malá who always took me out of office so we could let off the steam during an afternoon walk. You more than me. I thought we would exchange places at some point during my studies, but this change actually never happened and it was always you wanting to jump from the PSI bridge!

Aleksandra Kim, the always caring tomboy princess. To think that our friendship started with me stating that pelmeni is basically dumpling... which we both know I'm right.

Lubomír Bureš, who rekindled the philosophical thinking I had buried for so long due to lack of appropriate company.

Václav Šísl, that Christmas will always be in my heart. If I close my eyes, I can still taste the Goulash!

Boris Aviv Hombourger, my nuke brother.

Professor João Alfredo Medeiros, who had a pivotal role in my early development as a scientist. For sure a part of you can be found all over this thesis.

Shihan Miguel Greg, whose Budo made me unrelenting. Bufu Ikkan.

Meggie Dorothea Krymowski; the last rose-golden egg Switzerland had to give.

My sister, Carolina Gonzalez Gonzaga de Oliveira, who came to this world because I asked. Now the joke is on me and I have to bite the bullet!

Finally, to my parents, João Gonzaga de Oliveira and Damasia Gonzalez dos Santos Oliveira, for supporting and enabling all this adventure.

I am grateful to all. Directly or not, each ones individual contribution made this adventure fun, enriching, unique and memorable. This thesis, that I crafted with everything I am, is a tribute to our time together.

Baden, 18 October 2021

R.d.O.

Abstract

Molten Salt Reactors are a class of advanced reactors that is typically characterised by the presence of a molten inorganic salt of a fissile material as fuel. At the moment, this class of reactors is considered one of the promising options in the long run of nuclear power.

In the context of safety assessment of the MSFR, a task on analysis of selected transients was planned by the SAMOFAR project, to which this thesis should contribute. In addition, fluids with high melting temperature bring with them challenges regarding solidification of the fluid and possible flow blockage, to which a method was implemented during this thesis to evaluate the consequences. The open core cavity with curved shape adds another issue where the core is highly turbulent and requires the use of unstructured mesh codes for routine analysis, adding uncertainty and computational burden to the analytical workflow. Using ATARI, the tool developed during this thesis, we:

1. Analyse the MSFR together with partners in support of a safety assessment task.
2. Consider the impact and how to include solidification/melting in an analytical methodology.
3. Evaluate options to manage/reduce uncertainties and the burden of routine use of high-fidelity codes.

After a careful derivation of the conservation laws and assembling the code main algorithms, ATARI undergoes verification in a limited scope using the Stefan problem, manufactured solutions and energy balances.

The nuclear cavity benchmark was performed between SAMOFAR partners showing equivalency of the codes within the scope of the benchmark. Then the participating institutions perform analysis of the MSFR for the safety assessment task, which ATARI is unable to followup with the transient analysis. The reason for failure is identified as the requirement of appropriate acceleration schemes by OpenFOAM-based multiphysics solvers, and the lack of one in ATARI.

In the lack of accurate data regarding the heat exchangers, the impact of solidification is evaluated in a few simplified cases of a pipe with square cross-section and heat exchanger. For different salt compositions and different boundary conditions, the heat exchanger is

found to recover from blockage if pump maintains its momentum and some cross-flow is allowed. The recovery of cases with abrupt solidification is questionable due to the required pressure head.

The use of flow baffles is investigated as an option to structure the flow inside the reactor and reduce uncertainties originating from turbulence modelling while allowing the use of structured mesh codes for analysis. The concept is tested in a chloride based MSR, which retains its breed and burn mode while showing a quasi-1D flow appropriate for modelling with legacy codes.

We recommend that the nuclear cavity benchmark should be expanded to include a more demanding transient case, representative of MSFR transient analysis requirements. A case is made for homogenised and economical models as used in ATARI to enable the practical use of coupling methods that promote the simultaneous solution of equations. A niche for ATARI is found as a special-mission code for analysis of challenging cases with deforming structure and flow field that justifies the use of unstructured meshes and the added computational burden. Due to design uncertainties and their nature, it is recommended to consider solidification implicitly when analysing the MSFR fuel circuit.

Key words: Methodology, Molten salt, Freezing, Analysis, Design

Résumé

Les réacteurs à sels fondus sont un type de réacteurs avancés qui sont typiquement caractérisés par la présence de sel inorganique fondu de matériau fissile utilisé comme combustible. À l'heure actuelle, ce type de réacteurs est considéré comme étant une des options prometteuses à long terme pour la production d'énergie d'origine nucléaire.

Dans le contexte de l'évaluation de sûreté du MSFR, l'analyse de transitoires choisis a été planifiée dans le cadre du projet SAMOFAR, auquel cette thèse doit contribuer. En outre, les fluides à haut point de fusion présentent des défis particuliers concernant la solidification du fluide et la possibilité de blocage de l'écoulement, problème pour lequel une méthode a été implémentée dans cette thèse afin d'en évaluer les conséquences. Le cœur du réacteur, consistant en une cavité ouverte à la forme incurvée, se rajoute aux défis précédents puisque l'écoulement dans le cœur est hautement turbulent et requiert l'utilisation de codes de calculs à maillages non-structurés pour des analyses de routine, ajoutant de ce fait des incertitudes et une charge de calcul substantielle aux travaux d'analyse et de conception. Grâce à ATARI, l'outil développé pendant cette thèse, nous:

1. analysons le MSFR en collaboration avec des partenaires en support à la tâche d'évaluation de sûreté;
2. considérons l'impact de et comment inclure la solidification et la fusion dans une méthodologie analytique;
3. évaluons les options pour gérer ou réduire les incertitudes et la charge de calcul due à l'utilisation routinière de codes de calcul à haute fidélité.

Après une dérivation des lois de conservation et l'assemblage des algorithmes principaux du code, ATARI est vérifié dans un domaine d'application limité à l'aide du problème de Stefan, de solutions manufacturées et de bilans énergétiques.

Un benchmark sur le cœur en forme de cavité est effectué en collaboration avec les partenaires du projet SAMOFAR, montrant l'équivalence des codes utilisés dans le domaine d'application dudit benchmark. Les institutions participantes produisent ensuite des évaluations de la sûreté du MSFR, ce qu'ATARI s'avère incapable de faire dans le cadre des analyses de transitoires. La raison de cet échec est identifiée et est due au besoin d'algorithmes d'accélération de calculs pour les solveurs multi-physiques basés sur OpenFOAM, et le manque d'un tel outil dans ATARI.

Malgré le manque de données précises concernant les échangeurs de chaleur, l'impact de la solidification est évalué à l'aide de plusieurs cas simples de tuyaux à section carrée et d'échangeurs de chaleurs. Pour plusieurs compositions du sel et conditions de bord, il est montré que l'échangeur se remet d'un blocage si la pompe correspondante maintient sa vitesse et qu'un minimum d'écoulement croisé est permis. La récupération lors de cas où la solidification est particulièrement abrupte est en revanche discutable à cause de la différence de pression nécessaire.

L'utilisation de baffles séparateurs est étudiée en tant qu'option pour structurer l'écoulement dans le cœur et réduire les incertitudes venant de la modélisation des turbulences tout en permettant l'utilisation de codes de calcul à maillages structurés à des fins d'analyse. Le concept est testé dans un réacteur à sels fondus de chlorures, qui garde la capacité de fonctionner en cycle *Breed-and-Burn* tout en démontrant un écoulement quasiment uni-directionnel approprié à la modélisation par des codes de calculs existants.

Nous recommandons que le benchmark sur le cœur en forme de cavité soit étendu pour inclure des transitoires plus complexes, représentatifs des nécessités des évaluations de sûreté du MSFR. Nous défendons ensuite l'utilisation de modèles homogénéisés et économiques tels qu'utilisés dans ATARI pour permettre l'utilisation pratique de méthodes de couplage qui promeuvent la résolution simultanée d'équations. Une utilisation de niche est trouvée pour ATARI en tant que code à usage spécifique pour l'analyse de cas complexes avec déformation des structures et champs de vitesses qui justifient l'utilisation de maillages non-structuré et la charge de calcul qui en découle.

À cause des incertitudes de conception et leur nature, il est recommandé de prendre en compte la solidification implicitement dans l'analyse du circuit combustible du MSFR.

Mots clefs: Méthodologie, Sels fondus, Solidification, Analyse, Conception

Contents

Preface	i
Acknowledgements	iii
1. Introduction	1
1.1. Historical Overview	4
1.2. MSFR challenges	6
1.3. Legacy codes, OpenFOAM, GeN-Foam, and ATARI	7
1.4. Formalism: definitions and notation	10
1.4.1. Mathematics formalism: tensor algebra	11
1.4.2. Physics formalism: conservation laws and the continuity equation	14
1.4.3. Verification, validation and benchmarking	18
1.5. Motivation and objectives	19
1.6. Thesis structure	20
2. Fluid mechanics	23
2.1. Single-phase flow equations	24
2.2. Multiphase flow model	27
2.3. Homogenised model for flow through a structure	29
2.4. Turbulence modelling	35
2.5. Solidification and melting	38
2.6. Discussion	41
3. Neutronics	45
3.1. Neutron transport	46
3.2. The diffusion approximation	49
3.3. Parametrization	52
3.4. Discussion	55
4. Algorithms	57
4.1. Single-mesh solver	58
4.1.1. Implemented equations and flow diagram	59
4.2. Multi-mesh solver	65
4.2.1. Mesh hierarchy	65
4.2.2. Flow diagram and extended algorithm	66
4.3. Discussion	67

5. Verification	71
5.1. Stefan problem	73
5.1.1. 1-phase Stefan problem	74
5.1.2. 2-phase Stefan problem	78
5.2. Recirculating cavity	82
5.2.1. Domain specification	83
5.2.2. Equations and scope	84
5.2.3. Analytical solutions	86
5.2.4. Boundary conditions	87
5.2.5. Parameters	88
5.2.6. Results	90
5.2.7. Mesh studies	93
5.3. Triple circuit heat exchanger	93
5.4. Discussion	97
6. Application	99
6.1. Nuclear cavity benchmark	100
6.1.1. Step 0.1: Fluid dynamics	102
6.1.2. Step 0.2: Neutron diffusion	102
6.1.3. Step 0.3: Energy transport	102
6.1.4. Step 1.1: Neutron and DNP transport	104
6.1.5. Step 1.3: Power coupling	106
6.1.6. Step 1.5: Buoyancy	108
6.1.7. Step 1.7: Full coupling	108
6.1.8. Step 2.0: Transient behaviour	110
6.1.9. Discussion	111
6.2. Molten Salt Fast Reactor	115
6.2.1. Steady State Results	119
6.2.2. Transient problems	123
6.2.3. Discrepancies between studies	123
6.2.4. Discussion	124
6.3. Freezing in Heat Exchanger	129
6.3.1. Reference	132
6.3.2. Freezing with tuned salt	134
6.3.3. Freezing with MSFR salt	140
6.3.4. Discussion	140
6.4. Molten Chloride Fast Reactor conceptual design	146
6.4.1. Materials and Properties	147
6.4.2. Core Geometry and Meshing	148
6.4.3. Boundary Conditions	149
6.4.4. Fuel cycle parameters	149
6.4.5. Reflector and baffles	150

6.4.6. Clear and Baffled Cores Comparison	151
6.4.7. Results	151
6.4.8. Discussion	152
7. Conclusion	155
7.0.1. Future work	157
Appendices	159
A. Derivations	159
A.1. Stress tensor	159
A.2. Conservation of total energy	163
B. MSFR nuclear data	167
B.1. Fuel cross-sections	169
B.2. Heat Exchanger cross-sections	171
C. Personal experiences	173
C.1. Best practices on programming	173
C.1.1. Code structure and Object-Oriented Programming	174
C.1.2. Managing code development and Software Quality Assurance . .	177
C.1.3. The paradox of happiness – building a mountain with strangers .	178
Bibliography	190

List of Figures

1.1. Low-carbon electricity outlook in 2018	2
1.2. The MSFR concept [Allibert et al., 2017]	5
1.3. Change of fuel channel size due to salt freezing in PCHE	6
1.4. Conceptual representation of tensor τ	11
2.1. Continuous enthalpy due to explicit latent heat	39
3.1. Multiscale reactor analysis	49
3.2. Parametrization of cross-sections from a reference point r	53
4.1. Simplified flow diagram of single-mesh solver	61
4.2. Multiple domains of a multi-mesh solver	66
4.3. Simplified flow diagram of multi-mesh solver	68
5.1. Solution for λ	75
5.2. Temperature and Liquid fraction fields at $t = 10\,000\text{ s}$	76
5.3. Temperature and Liquid fraction at $t = 10\,000\text{ s}$	76
5.4. Melting front position as a function of time	77
5.5. Temperatures as a function of time at 4 positions	77
5.6. Spatial Order of Accuracy at last time step	78
5.7. Temperature and Liquid fraction fields at $t = 10\,000\text{ s}$	80
5.8. Plot of Temperature and Liquid fraction at $t = 10\,000\text{ s}$	81
5.9. Melting front position as a function of time	81
5.10. Temperatures as a function of time at 6 positions	81
5.11. Spatial Order of Accuracy at last time step	82
5.12. Temporal Order of Accuracy with 250 cells	82
5.13. Recirculating cavity domain	83
5.14. Flux on the domain and results over line AA'	91
5.15. Velocity magnitude on the domain and results over line AA'	91
5.16. Pressure on the domain and results over line AA'	92
5.17. Temperature on the domain and results over line AA'	92
5.18. L_1 and L_2 norms for velocity	93
5.19. Wall time for different mesh sizes	94
5.20. Triple circuit HEX meshes	94
5.21. Conceptual PCHE with 3 fluids	95
5.22. Solid in the PCHE	96

6.1. Nuclear cavity domain	100
6.2. Step 0.1 velocity field	102
6.3. Step 0.1 comparison of results between participants	103
6.4. Step 0.2 flux summed over energy	103
6.5. Step 0.2 comparison results between participants	104
6.6. Step 0.3 temperature field	104
6.7. Step 0.3 comparison results between participants	105
6.8. Step 1.1 delayed neutron precursors concentration fields	105
6.9. Step 1.1 comparison of results between participants	106
6.10. Step 1.3 fluid density and temperature fields	106
6.11. Step 1.3 comparison of results between participants	107
6.12. Step 1.5 velocity magnitude field	108
6.13. Step 1.5 comparison of results between participants	109
6.14. Step 1.7 full coupling equal velocity plots	110
6.15. Step 1.7 full coupling equal power plots	111
6.16. Step 2.0 power gain Bode diagram	112
6.17. Step 2.0 phase-shift Bode diagram	112
6.18. MSFR geometry cutaway	115
6.19. MSFR cutaway view with tagged regions.	116
6.20. Meshed domain comprising a fuel circuit sector	117
6.21. MSFR pressure and velocity fields with vectors	120
6.22. MSFR volumetric power and temperature fields	120
6.23. MSFR neutron flux of groups 1 and 6	121
6.24. MSFR delayed neutron precursors of groups 1 and 8	122
6.25. MSFR eddy viscosity field	122
6.26. Comparison of MSFR velocity field with literature.	123
6.27. Comparison of MSFR temperature field with literature.	124
6.28. Square pipe and HEX mesh side, front and perspective	131
6.29. Reference field results without freezing at $t = 20$ s	132
6.30. Reference plot results without freezing at $t = 20$ s	133
6.31. Freezing HEX fields at $t = 5$ s	134
6.32. Freezing HEX plots at $t = 5$ s	135
6.33. Freezing HEX fields at $t = 6$ s	136
6.34. Freezing HEX plots at $t = 6$ s	137
6.35. Freezing HEX fields at $t = 20$ s	138
6.36. Freezing HEX plots at $t = 20$ s	139
6.37. MSFR Freezing HEX fields at $t = 5$ s	141
6.38. MSFR Freezing HEX fields at $t = 6$ s	142
6.39. MSFR Freezing HEX fields at $t = 1.2$ s	143
6.40. MSFR Freezing HEX fields at $t = 3$ s	144
6.41. Core geometries with barrel and baffles.	148
6.42. Eddy viscosity in clear (left) and baffled (right) cores.	151

6.43. Flux in group 1 and volumetric power in the core.	152
6.44. Velocity and velocity-dependent fields of temperature and delayed neutron precursor in group 4.	153

List of Tables

1.1. Forms of continuity equation	17
2.1. Parameters of the single-phase Navier-Stokes equations.	25
2.2. Expanded parameters of the multiphase equations.	28
2.3. Parameters of simple heat transfer model.	33
2.4. Nusselt number parameters and empirical correlation.	34
2.5. Parameters of Darcy-Weisbach empirical correlation.	35
2.6. Parameters of the k - ϵ model in homogenised regions.	37
3.1. Parameters of the neutron transport equation.	47
5.1. Stefan problem fluid properties	74
5.2. Recirculating cavity fluid properties	88
5.3. Recirculating cavity neutronic properties	89
5.4. Boundary conditions of the triple heat exchanger	95
5.5. Energy balances in the triple heat exchanger	96
6.1. Nuclear cavity benchmark steps	101
6.2. MSFR fluid properties	117
6.3. Freezing HEX boundary conditions	130
6.4. Freezing HEX fluid properties	131
6.5. MCFR fluid properties	147
6.6. MCFR fluid dynamics boundary conditions	149
6.7. Candidate chloride fuel salts adapted from [B. Hombourger et al., 2019]	150
6.8. Candidate reflectors considered at equilibrium	150
A.1. Strain energy decomposition	165
B.1. MSFR 6 energy group structure and inverse neutron speed	167
B.2. MSFR delayed neutron precursor data	167
B.3. MSFR albedo coefficients	168
B.4. MSFR prompt and delayed neutron spectrum, and fission energy yield .	168
B.5. Fuel cross-sections	169
B.6. Fuel cross-sections parametrization coefficient α	169
B.7. Fuel P_0 scattering matrix $\Sigma_{s,i \leftarrow j}$	170
B.8. Fuel P_0 scattering matrix $\Sigma_{s,i \leftarrow j}$ parametrization coefficient α	170

B.9. HEX cross-sections	171
B.10. HEX cross-sections parametrization coefficient α	171
B.11. HEX P_0 scattering matrix $\Sigma_{s,i \leftarrow j}$	172
B.12. HEX P_0 scattering matrix $\Sigma_{s,i \leftarrow j}$ parametrization coefficient α	172

Nomenclature

ATARI Accessible Tool for Advanced Reactor Investigation

BNB Breed-aNd-Burn

CFD Computational Fluid Dynamics

DNP Delayed Neutron Precursor

GIF Generation IV International Forum

HEX Heat EXchanger

LHS Left Hand Side (of an equation)

MMS Multi-Mesh Solver

MSFR Molten Salt Fast Reactor

MSR Molten Salt Reactor

PCHE Printed Circuit Heat Exchanger

RHS Right Hand Side (of an equation)

SAMOFAR Safety Assessment of the Molten Salt Fast Reactor

SHM snappyHexMesh

SMS Single-Mesh Solver

1. Introduction

Aller Anfang ist schwer.

(German proverb)

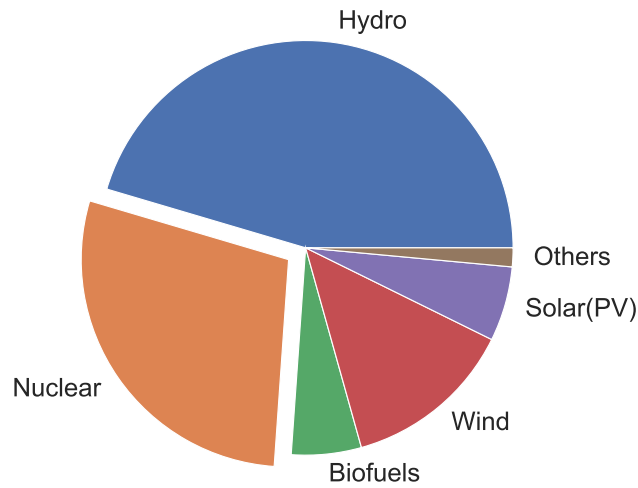
Energy is an essential component of a comfortable life. A strong correlation between standards of living and energy use can be observed around us and around the world itself. Yet today, for most part this energy comes from fossil fuels; resulting in green house gas emissions and long-term change of climate in unpredictable ways. Therefore Humanity faces the double challenge of lifting a significant portion of the population — the majority, really — out of poverty by providing more abundant energy while mitigating its own impact on the surrounding natural environment. In this context, significant effort is directed towards development and deployment of low-carbon energy sources.

In order to achieve this objective economically and on a reasonable time frame, scientific studies to date indicate that wide-scale deployment of various technologies will be necessary [IEA, 2020a], and among these technologies is nuclear power. Despite a number of aberrant studies indicating otherwise, proper studies indicate that there is no reasonable replacement for nuclear energy if emission goals are to be achieved [IEA, 2019; NEA, 2019; Sepulveda et al., 2018; Loftus et al., 2015].

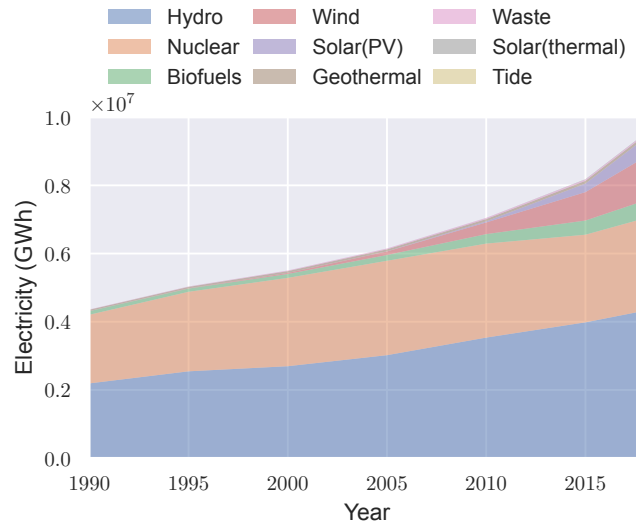
In fact, it is a good moment to reflect on aberrant studies, since these typically require a mix of renewable deployment, energy storage, energy efficiency, behavioural changes, and sometimes even stagnant human development. It is January of 2021 at the time of this writing, and the world is going through a pandemic and struggling to recover. The behavioral changes imposed by the pandemic did have a noticeable impact [IEA, 2020c] but not so much, which is a good indication of just how much behavioural change would be necessary on a long term basis to compliment a working energy transition. Needless to say, if in many places people were extremely resistant to behavioural changes even in face of a life threatening situation, expecting equivalent changes to avoid an abstract threat such as climate change is most likely a pipe dream. After witnessing first-hand how resistant people are to changes that affect their lives, it seems that not counting on it is a much more robust choice. On the positive side, we do not lack technologies that are scalable and up to the task of powering people's lives without behavioural changes, but we do seem to lack the right incentives, market fitness and an amenable proposition regarding by-products (i.e., nuclear waste).

1. Introduction

Today, nuclear power is the second largest source of low-carbon electricity, just after hydropower (figure 1.1a), with its contribution staying mostly constant for decades (figure 1.1b). Although deployment of nuclear energy is not on track to achieve the goals set [IEA, 2020b], its share is still expected to increase significantly until 2040, mostly in China and India at the moment [IEA, 2020c]. However, in western democracies nuclear power is slowly decreasing and only now the right incentives and market fitness are topics that are receiving more attention; after years of inaction.



(a) Distribution by low-carbon energy source as of 2018



(b) Historical trend of low-carbon energy sources

Figure 1.1.: Low-carbon electricity outlook in 2018

Regarding an amenable proposal to by-products, it turns out that Fermi's warning about mankind's feeling towards radioactivity was spot-on [Weinberg, 1994, p. 41], already at

the dawn of nuclear age.

It is not clear that the public will accept an energy source that produces this much radioactivity and that can be subject to diversion of material for bombs.

—Enrico Fermi, Fermi's warning

In the author's own outreach experience in the last few years, it turns out that from people's perspective nuclear waste is a real issue, even if the reasons behind it are illusory. Most people do warm up well to the reality of nuclear waste when presented with well supported facts and arguments, and specially after seeing it themselves. However, there is only so many people the author could take to visit ZWILAG, the Swiss interim storage of nuclear waste, to have the experience of touching a cask of high level waste themselves and realising how harmless it is after proper storage. It is unknown how long it might take for people to start accepting radioactivity as part of the world, and nuclear technology as just another tool with its usefulness and risks, like fire. Careful management of nuclear waste seems necessary, specially if reactor deployment gains traction, even if for no reason other than to make it more amenable to people.

The overwhelming majority of nuclear reactors deployed around the world today are Light Water Reactors using enriched Uranium as fuel. This is a mature technology that has shown a superlative safety record, despite criticism. A triple meltdown in Fukushima Dai-ichi resulting in no deaths directly related to radioactivity is a testament to safety even in face of avoidable project flaws and dire circumstances; a claim that is hardly matched by any other industrial activity. Regardless of its safety, this technology produces a **relatively** significant amount of unnecessary waste downstream (i.e., spent fuel) and upstream (i.e., depleted uranium, mining tails, etc) of the reactor operation; significant because people care about it, and unnecessary because the technologies required for a cleaner fuel cycle exist and have been proven.

There are solutions to downstream waste management [Shwageraus, 2003], even if economic implementation is open to discussion. Between downstream and upstream wastes, downstream one has a definite higher priority since the population shows noticeable aversion to it. However, using depleted uranium from the upstream waste also solves the problem of fuel supply for long-term operation of a large fleet of nuclear reactors, which is an issue [NEA & IAEA, 2020] assuming Uranium extraction from seawater to be uneconomical. In order to address upstream waste and ensure long-term fuel supply, breeder reactors are necessary. These advanced reactors concepts are capable of converting depleted uranium into usable fuel in a self-sustained way as part of their fuel cycle.

In 2000, the Generation IV International Forum (GIF) had its first meeting. The

objective of this task force is to develop innovative reactors that excel in 4 aspects: economics, proliferation resistance, reliability/safety and sustainability. To achieve this, the consortium members selected 6 category of reactors to study:

- Gas-cooled fast reactor (GFR)
- Lead-cooled fast reactor (LFR)
- Molten salt reactor (MSR)
- Sodium-cooled fast reactor (SFR)
- Supercritical-water-cooled reactor (SCWR)
- Very-high-temperature reactor (VHTR)

Out of these, four can be reasonably designed as breeder reactors: GFR, LFR, MSR, and SFR. In addition to the GIF consortium, today several companies are developing their own advanced reactor concept as well.

Switzerland is a member of GIF since signing the charter in 2002, initially with the intention to develop the GFR and VHTR. In 2015 Switzerland left the GFR task force and signed a memorandum of understanding joining the development of MSR [GIF, 2015, p. 11]. As a part of this effort, the Paul Scherrer Institut (PSI) is participant of the Safety Assessment of the Molten Salt Fast Reactor (SAMOFAR) [SAMOFAR, 2015] project funded by the Horizon 2020 EU initiative. This project studies a conceptual Molten Salt Fast Reactor (MSFR) design, which is the flagship MSR of GIF.

1.1. Historical Overview

Molten salt reactors are, as the name suggests, a category of reactor that uses some type of molten salt either as fuel or coolant in the reactor core. In this thesis, we shall limit this definition to reactors that are fuelled by an inorganic salt of a fissile nuclide that is molten and allowed to circulate in the reactor core.

The first incarnation of this idea was the Aircraft Reactor Experiment (ARE) in the fifties. This was a beryllium oxide moderated $2.5 \text{ MW}_{\text{th}}$ nuclear reactor operated at Oak Ridge National Laboratory (ORNL) designed for military aircraft nuclear propulsion. While a nuclear reactor is unsuitable for aircraft propulsion, as was already known at the time, it was understood that a high-temperature reactor was useful for other purposes [Weinberg, 1994, p. 97].

By the end of the decade, interest in a nuclear powered aircraft was replaced by interest

in adapting the developed technology for civilian electricity production, culminating in the Molten Salt Reactor Experiment (MSRE), a graphite moderated 10 MW_{th} reactor also operated at ORNL in the sixties (Weinberg, 1994, p. 126; MacPherson, 1985). No successor to the MSRE has been made, yet it can be considered a very successful experiment that solved multiple challenges related to molten salt reactors while demonstrating the reliability and safety of the concept. Ever since then, the reference MSR design and the one originally adopted by GIF was a graphite-moderated core heavily based on the MSRE.

A significant departure from the MSRE occurred since 2008 [GIF, 2008, p. 37] when the core was re-evaluated. Parametric studies of an MSR lattice [Mathieu et al., 2009], concluded that an MSR without graphite moderator has significant advantages regarding safety and waste generation. As a result the core retained its cylindrical geometry, but the moderator was removed; this was the core adopted for study in the Evaluation and Viability of Liquid Fuel Fast Reactor System (EVOL) project. By the end of the project, the shape of the original cylindrical cavity was refined in order to reduce recirculation zones in the reactor cavity [Rouch et al., 2014], resulting in the curved walls between fuel and blanket salts. Finally, the entire primary circuit was integrated into the vessel in order to reduce the frequency of pipe leak hazards [Merle, 2017]. The result is the integrated MSFR concept shown in figure 1.2, where a sector is composed of a breeding blanket (green), reflector (grey), pump (blue) and heat exchanger (HEX) (yellow). The primary circuit is composed of 16 sectors arranged azimuthally at the inner periphery of the core vessel. This is the latest iteration of the concept and the one studied by the SAMOFAR project.

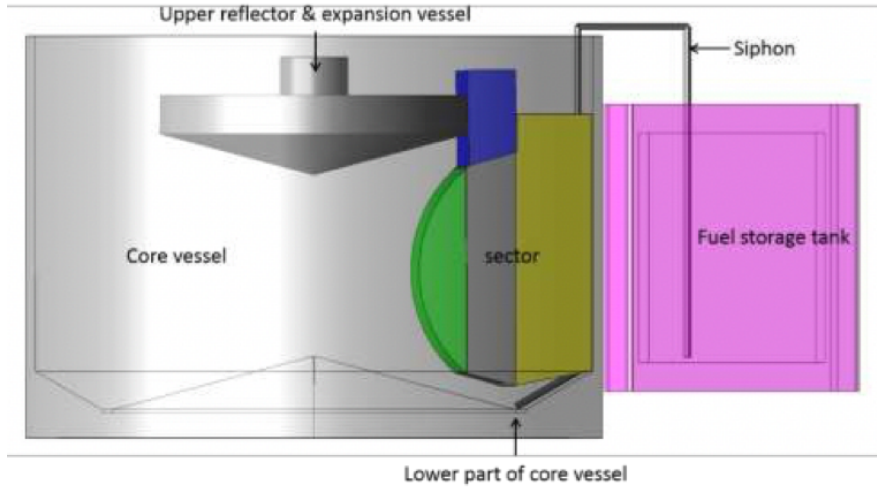


Figure 1.2.: The MSFR concept [Allibert et al., 2017]

1.2. MSFR challenges

An MSR typically has characteristics that depart significantly from other reactor categories due to the molten fuel. This fuel circulates from core cavity to heat exchanger and back into the core, carrying fissile nuclides, fission products, and any other material in the flow. Whereas fission products are locked in place in a typical reactor, in an MSR its movement lead to smearing of delayed neutron precursors and decay heat throughout the flow path.

In the particular case of the MSFR, the core cavity is a large homogeneous zone due to the removal of graphite moderator. As a consequence, the flow in the core is turbulent and has strong 3-dimensional characteristics, i.e., no single component of velocity is so dominant that a good approximate solution can be obtained by abstracting the other components. This invalidates friction and heat transfer correlations used by system codes (e.g., TRACE [US NRC, 2017]) **in the core**; therefore if core behaviour is a subject of study, a more elaborate model is necessary. In the past decade, the use of Computational Fluid Dynamics (CFD) supplemented by some neutronics capability has been used to fill this gap [Aufiero, 2014; Fiorina et al., 2014], particularly OpenFOAM-based applications.

The inorganic molten salts used as fuel in reactors are substances with relatively high melting point; of the order of 600 K at the lower end. Previous simulation of overcooling transients [Aufiero, 2014; Fiorina et al., 2014; Pettersen, 2016] indicate that freezing of this salt inside heat exchangers might occur. During most of this thesis, the proposal for the heat exchanger of the MSFR was of a Printed Circuit Heat Exchanger (PCHE) design. If freezing of fuel salt on the walls were to occur (as schematised in figure 1.3), it could lead to complete blockage. A knowledge gap exists in the consequences of salt freezing inside heat exchangers, although the phenomena has been recently simulated [Cartland-Glover et al., 2019].

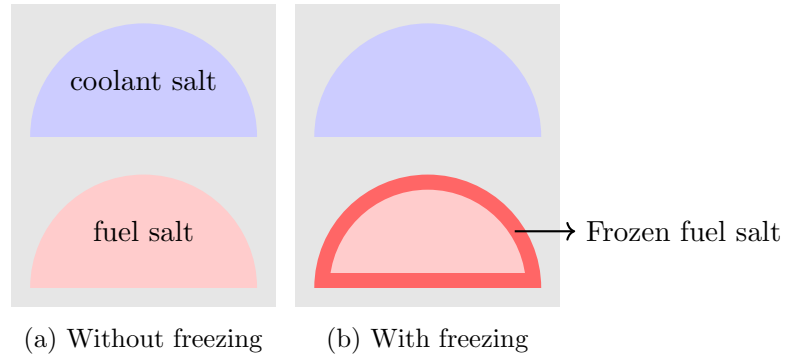


Figure 1.3.: Change of fuel channel size due to salt freezing in PCHE

There are other challenges that could be mentioned: corrosion, gas bubbling, precipita-

tion of salts due to limits of solubility, fouling of heat exchangers, etc. A nuclear reactor is a complex system filled with challenges. All the more so in the case of an evolving design such as the MSFR, where every change brings with it new questions.

Of the many existing challenges that could be addressed, we shall focus on two: solidification of the molten salt, and simulation of the turbulent core cavity with a nuclear energy source. The main tool that will be used in this thesis to tackle these is OpenFOAM.

1.3. Legacy codes, OpenFOAM, GeN-Foam, and ATARI

Nuclear reactors require tools to allow predictions of its phenomena for the purposes of design, maintenance, and decommissioning. Through decades, academia, research laboratories and private institutions have developed their own set of tools to perform those activities. In this thesis, these tools shall be referred to as “legacy tools”. Commonly, these tools implement their own numerical methods, discretisation schemes, specialised models, and several other facilities that allow them to perform with excellent for their designated tasks. Several examples of legacy codes could be given, but the most relevant to this work fit into de categories of sub-channel codes or system codes.

In the past decade, there has been an increased interest in developing new tools by exploring functionalities of modern frameworks. One such framework, and the one used in this thesis, is OpenFOAM [Weller et al., 1998]; a C++ library that implements the finite volume method (FVM) for spatial discretisation of partial differential equations (PDE). Given a discretised spatial domain, commonly referred to as “mesh”¹, OpenFOAM uses the FVM to turn a PDE into a system of algebraic equations.

A lengthy explanation of the finite volume method and CFD will be avoided. Several books² are available [Moukalled et al., 2016; Chung, 2010; Versteeg & Malalasekera, 2007; Date, 2005; Pletcher et al., 1984] explaining the topic with a degree of sophistication that cannot be matched in an introductory chapter. A basic background on the topic is expected, and will be used to address more niche areas.

As a modern framework for CFD, OpenFOAM finds competition in several other frameworks such as ANSYS, COMSOL, MOOSE [Gaston et al., 2009], FEniCS [Alnæs et al., 2015], COOLFluid [Lani et al., 2013], etc. All these frameworks operate with unstructured meshes, which despite the name, does not refer to the **visual** loss of structure due to mixed shapes and polyhedra defining cells. A code that works with

¹In CFD, the term mesh is associated with the spatial discretisation. However, in a broader sense it refers to the discretisation of any dimension. In the context of neutronics it might refer to discretisation of energy, exemplified by the SHEM (Santamarina Hfaiedh Energy Mesh) energy group structure.

²If the reader would like a single suggestion out of these, the book by Moukalled et al. [2016] is targeted specifically at OpenFOAM.

a structured mesh only needs a simple loop for each spatial dimension to iterate over every cell of the mesh. A code that works with an unstructured mesh needs data on cell structure and connectivity such as owner, neighbours, surfaces, volumes, etc. Therefore, regardless the visual appearance, the classification of a mesh as structured or unstructured refers to a **layout in memory**; a mesh may seem structured visually in a code that operates with unstructured mesh, but it will still be unstructured due to memory layout. A code with structured mesh can iterate over the domain much faster than an unstructured one, and are much faster for the same reason. As a trade-off, unstructured meshes have a flexibility to represent spatial domains that cannot be matched by structured ones.

The use of the FVM and unstructured meshes for reactor analysis gives several advantages in exchange for simulation time. Meshes are much finer in the FVM, therefore cusping (i.e., partial insertion of a strong absorber in a coarse cell) is a much smaller problem. The unstructured mesh allows simulating events that lead to arbitrary change of domain (e.g., structural deformation), and some reactors have unorthodox shapes that require this flexibility, such as the MSFR.

This thesis inherited GeN-Foam [Fiorina et al., 2015] as a starting point³, which is an OpenFOAM-based application for nuclear reactor analysis originally developed at PSI and currently under active development at EPFL.

The author cannot comment on the current status of the program, but in 2017 it had several limitations, such as:

1. Lack of standardisation: Code duplication and variable repetition (e.g., 3 variables “gamma”, “porosity”, and “void fraction” expressing the same concept).
2. Weak expressiveness: the equations written are technically correct, but do not emphasise the physical concepts (e.g., fission source not precomputed).
3. Plain functionalities: using compiler “#include” macros instead of the object oriented functionality of the programming language.
4. Difficult to expand: Since models are not classes, changing models implies changing the bulk of the code.

Item 2 is a minor issue from the point of view of performance, and more of a lost opportunity in terms of readability and reinforcement of physical concepts. To make the issue explicit, let us take an example case. Equation 1.1 and 1.2 are the neutron diffusion and delayed neutron drift equation, which will be explained in detail in chapter 3. These equations have a term in common, which is the fission source equation 1.3, which represents the number of neutrons that are generated by fission events per unit

³More specifically, the version from the end of the year 2017.

volume and unit time. By not exploiting the concept of the neutron source, we miss not only the performance gain of computing it only once and using in both equations, but we miss the opportunity to communicate the physical concept that these equations share, which improves readability and understanding. GeN-Foam missed many opportunities to “teach physics with code” whenever these concepts were not taken advantage of.

$$\begin{aligned}
& \frac{1}{V_i} \frac{\partial}{\partial t} \phi_i(\mathbf{r}, t) + \nabla \cdot (-D_i(\mathbf{r}, t) \nabla \phi_i(\mathbf{r}, t)) + \Sigma_{\text{rem},i}(\mathbf{r}, t) \phi_i(\mathbf{r}, t) \\
&= \sum_{i \neq j} \Sigma_{s,i \leftarrow j}(\mathbf{r}, t) \phi_j(\mathbf{r}, t) \\
&+ \chi_{p,i}(\mathbf{r}, t) (1 - \beta(\mathbf{r}, t)) \frac{\sum_{j=1}^J \bar{\nu}_j(\mathbf{r}, t) \Sigma_{f,j}(\mathbf{r}, t) \phi_j(\mathbf{r}, t)}{k} \\
&+ \chi_{d,i}(\mathbf{r}, t) \sum_{g=1}^G \lambda_g C_g(\mathbf{r}, t)
\end{aligned} \tag{1.1}$$

$$\begin{aligned}
& \frac{\partial}{\partial t} C_g(\mathbf{r}, t) + \nabla \cdot (C_g(\mathbf{r}, t) \mathbf{u}) + \nabla \cdot (-D_g(\mathbf{r}, t) \nabla C_g(\mathbf{r}, t)) \\
&= \beta_g(\mathbf{r}, t) \frac{\sum_{j=1}^J \bar{\nu}_j(\mathbf{r}, t) \Sigma_{f,j}(\mathbf{r}, t) \phi_j(\mathbf{r}, t)}{k} \\
&- \lambda_g C_g(\mathbf{r}, t)
\end{aligned} \tag{1.2}$$

$$S_{\text{fis}} = \frac{\sum_{j=1}^J \bar{\nu}_j(\mathbf{r}, t) \Sigma_{f,j}(\mathbf{r}, t) \phi_j(\mathbf{r}, t)}{k} \tag{1.3}$$

Item 3 and 4 are related and are some of the biggest opportunities GeN-Foam missed. GeN-Foam was programmed using mostly compiler “#include” macros, which is essentially a form of “copy and paste”, following more or less what OpenFOAM solvers do. The difference between GeN-Foam and OpenFOAM solvers is size though — comparatively speaking, GeN-Foam is massive! Whereas in OpenFOAM solvers the use of “#include” directives is reserved for fairly small snippets of code that are used once or twice, in GeN-Foam there are entire structures inside these directives, such as a whole neutronics solver with reading of nuclear data, parametrization and all. This results in code that is duplicate, hard to read/understand, easy to break and hard to replace, which is an issue that generally plagues the solver.

The C++ programming language is designed to exploit the concept of classes and objects in order to tightly encapsulate functionality. A well written object not only organises codes, but organises concepts. Creating good classes require good understanding of multiple physical concepts, how they are related, and maybe more important, how to separate them. Proper use of this paradigm allows code that is highly modular, and this is only magnified by the functionality of the “run time selection table” native to the OpenFOAM

library, which allows selecting objects to use at run time, without specifying the options inside the code with a decision tree. The repercussions of using or not these concepts is quite extreme. Not using it effectively results in code that cannot be formally verified⁴, since every change results in a different copy of the old code. On the other hand, code that is properly encapsulated not only becomes verifiable, but also easily replaceable. Therefore, the use of this functionality allows broad experimentation with different models and implementations without breaking working code, which is an important aspect of verification and experimentation in general.

These and other issues severely limited GeN-Foam at the time. This is to be expected since the program was an unpolished proof-of-concept, which is not to be taken as any sort of harsh criticism of the code — the author is sure that there were several reasons for it to be this way. At the same time, this was an opportunity to explore different design philosophies and capabilities, which led to the creation of Accessible Tool for Advanced Reactor Investigation (ATARI).

The design of ATARI focused strongly on a structure that allowed smooth expansion of the code, with the objective of keeping working code safe while allowing unhindered experimentation.

1.4. Formalism: definitions and notation

A fundamental aspect of simulating natural phenomena is posing the right equations that describe conservations of relevant quantities. Most of ATARI is around the implementation of these as a system of equations, and closure relations required to solve the system.

Posing these equations require some limited knowledge of tensor mathematics. Therefore, it is relevant to discuss shortly this topic in order to set the mathematical formalism and notation to be used in later chapters.

Finally, it is necessary to discuss how to test the code in order to assert that it is correct and fit for purpose. The meaning of code verification and validation — two essential activities for scientific code development — will be introduced and defined.

⁴The meaning of verification here should not be taken lightly. In chapter 5 it will be formalised as a mathematical activity with a well defined purpose.

1.4.1. Mathematics formalism: tensor algebra

Tensor mathematics is the basis for modern continuum mechanics, therefore it is important to know how to operate using it. For this work, we can avoid an involved mathematical discussion about what tensors are, which can be found in great detail on multiple books [de Souza Sánchez Filho, 2016; Jeevanjee, 2015; Fleisch, 2011]. The important characteristic of a tensor relevant to understand this thesis is that it is a mathematical object that generalises the concept of a vector. If a vector, velocity for example, can be represented by a 1-dimensional array containing 3 numbers expressing velocity components in different directions of a 3D space, the viscous stress tensor τ can be represented by a 2-dimensional array expressing stresses in the 3 directions plus 6 components of stresses orthogonal to each direction, as shown in figure 1.4.

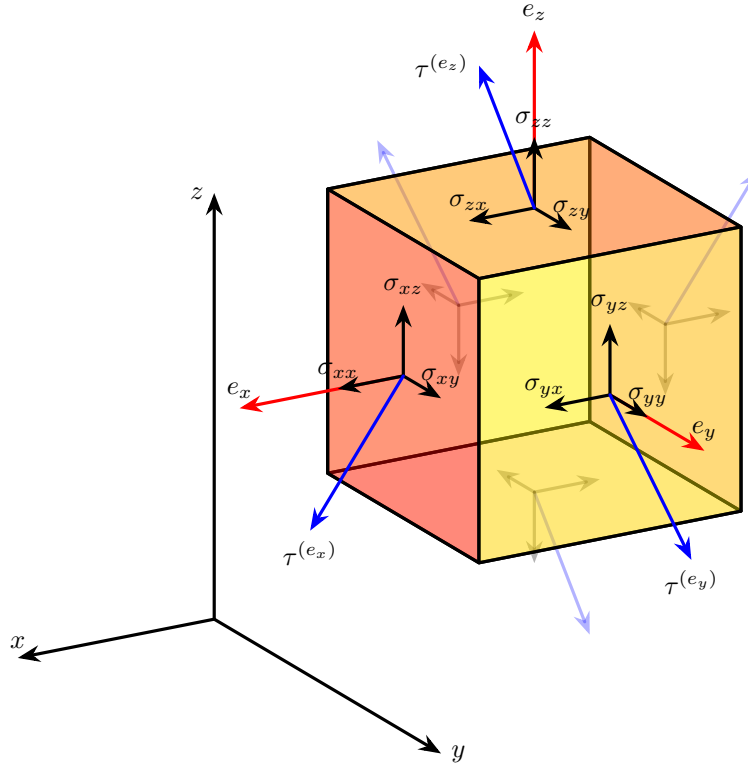


Figure 1.4.: Conceptual representation of tensor τ

Tensors have order equal to the dimensionality of the array needed to express the quantity, therefore order 0 tensors represent scalar quantities, order 1 vector quantities, and order 2 and higher quantities that require 2 or higher array dimensionality.

Notation: scalars, vectors and tensors

Since tensors of order greater than 3 will not be relevant in this work and neither will coordinate independence, the words scalar, vector and tensor will be used as shorthand notation to denote tensors of order 0, 1 and 2. The following standard will be used whenever possible:

- scalar: italic letters t
- vector: bold italic letters \mathbf{t}
- tensor: greek letters τ

In addition, 2 operations regarding tensor algebra are particularly relevant: the outer and inner products. The outer product or tensor product (both are equivalent in this work) represented by \otimes of 2 tensors of order n and k result itself in a tensor of order $n + k$. This operation between a scalar (order 0) and a vector (order 1) results in a scaled vector (order $0 + 1$). The same operation between 2 vectors \mathbf{a} and \mathbf{b} results in a tensor (order $1 + 1$) as shown.

$$\mathbf{a} \otimes \mathbf{b} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \otimes \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} a_1 b_1 & a_1 b_2 & a_1 b_3 \\ a_2 b_1 & a_2 b_2 & a_2 b_3 \\ a_3 b_1 & a_3 b_2 & a_3 b_3 \end{bmatrix} \quad (1.4)$$

The inner product or tensor contraction (also equivalent in this work) represented by a central dot \cdot of 2 tensors of order n and k results in a tensor of order $n + k - 2 \geq 0$. It consists of a generalisation of the trace operation. An example using the same vectors as before is shown.

$$\mathbf{a} \cdot \mathbf{b} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \cdot \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = a_1 b_1 + a_2 b_2 + a_3 b_3 \quad (1.5)$$

An operation denoted by the double dot $:$ also exists, which results in a tensor of order $n + k - 4 \geq 0$. It is only mentioned because it will appear briefly during derivations in appendix A.2, however, the term containing it will be ignored due to irrelevance to the modelling of the phenomena.

Finally, if the nabla or del operator ∇ is defined as a vector of spatial partial derivatives, the gradient and divergent operations can be expressed in this framework as outer and

inner products shown before and clarified below.

$$\nabla \mathbf{a} = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{bmatrix} \otimes \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} \frac{\partial a_1}{\partial x} & \frac{\partial a_2}{\partial x} & \frac{\partial a_3}{\partial x} \\ \frac{\partial a_1}{\partial y} & \frac{\partial a_2}{\partial y} & \frac{\partial a_3}{\partial y} \\ \frac{\partial a_1}{\partial z} & \frac{\partial a_2}{\partial z} & \frac{\partial a_3}{\partial z} \end{bmatrix} \quad (1.6)$$

$$\nabla \cdot \alpha = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{bmatrix} \cdot \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} \end{bmatrix} = \begin{bmatrix} \frac{\partial \alpha_{11}}{\partial x} + \frac{\partial \alpha_{21}}{\partial y} + \frac{\partial \alpha_{31}}{\partial z} \\ \frac{\partial \alpha_{12}}{\partial x} + \frac{\partial \alpha_{22}}{\partial y} + \frac{\partial \alpha_{32}}{\partial z} \\ \frac{\partial \alpha_{13}}{\partial x} + \frac{\partial \alpha_{23}}{\partial y} + \frac{\partial \alpha_{33}}{\partial z} \end{bmatrix} \quad (1.7)$$

It is a good opportunity to mention the differences between matrix algebra and tensor algebra. Since tensors of order 0 to 2 can be intuitively represented by matrices, there is a significant amount of confusion about the differences.

Matrix algebra is concerned with representation of mathematical structures, such as vectors, tensors and the system of equations from a computational point of view. A vector can be stored either as a row matrix or as a column matrix. The inner or outer product of 2 vectors will require one or the other to be transposed for the matrix product operation and will result in an appropriate matrix. Depending on how a problem is laid out as a matrix, it might be symmetric or asymmetric, etc.

Tensor algebra is concerned with how linear spaces operate instead of representation of the object that embodies the said space. For tensor algebra it is not relevant if vector \mathbf{A} is stored as column or row vector. It is just a vector or a tensor of order 1 where representation has been abstracted. Likewise the outer product of 2 vectors is not concerned if vectors are rows or columns. The product will result in a tensor and the necessary transposes for matrix algebra-like operations are also abstracted. A tensor of order 2 can be intuitively represented as a matrix structure, but this is just a consequence of how the structure was designed. A tensor is not just a matrix, it is a mathematical object that can be stored as a matrix but has additional properties, such as invariance under coordinate transformation. From order 3 and above, the representation becomes less intuitive.

Summarising, matrix algebra lays the foundation for **how mathematical operations might be represented and computed** while tensor algebra builds on top of this matrix algebra framework and defines **how particular operations are carried out**. Matrix algebra is not really relevant in this work. All underlying structures are implemented at a low level in the OpenFOAM library and abstracted from the programmer. Of interest to the programmer is how to use the high level operations that are given by the library in a syntax similar to tensor algebra.

1.4.2. Physics formalism: conservation laws and the continuity equation

A conservation law binds the behaviour of a quantity in the system in 2 ways. First, it states that a quantity is conserved in an isolated system. In other words, it cannot appear or disappear from the system, only transform. Second, it states how that quantity moves through space (and time), otherwise a quantity could disappear in one place and appear in another place and still be conserved.

Following Noether's theorem, conserved properties must demonstrate some sort of symmetry, or in other words, have some invariance property. For example, translational invariance and time invariance result in conservation of linear momentum and energy respectively. Therefore, only certain quantities, such as energy, have conservation laws by themselves, whereas enthalpy does not for example. This does not necessarily mean that a conservation law cannot be written using enthalpy, only that the formulation will require enthalpy and some other quantity to express conservation of energy instead. Invariance is also an important property in tensor mathematics.

Conservation laws are mathematically expressed by the continuity equation. In equation 1.8 the continuity condition of a generic conserved quantity ψ is shown.

$$\frac{\partial \psi}{\partial t} + \nabla \cdot \mathbf{j} = S \quad (1.8)$$

In this equation \mathbf{j} is the flux of ψ . Whatever the order n of the generic tensor ψ , \mathbf{j} is a tensor of order $n + 1$ so that the result after divergence operation has order n again, or else the addition operation with the time derivative is not possible. The flux \mathbf{j} can be broken down into 2 parts: advective and diffusive fluxes.

$$\mathbf{j} = \mathbf{j}_{\text{adv}} + \mathbf{j}_{\text{dif}} \quad (1.9)$$

The advective part \mathbf{j}_{adv} describes the transport of ψ by the bulk motion of the medium that contains it. There is a deceptive approximation in the advective flux that is not usually mentioned. In a fluid for example, velocity is only well defined in a point⁵ or for an infinitesimally small control volume, however it is necessary to make the control volume finite in order to solve the problem numerically. When the control volume has finite size, velocity is not clearly defined⁶. It is possible, and even likely, that flow patterns or eddies exist inside the finite control volume and we lost the capability to resolve it by making the control volume finite. Therefore, instead of solving for velocity,

⁵This is different for an ideal rigid body, where velocity is well defined through all the body volume.

⁶Despite the theory, even the case of a moving ideal rigid body velocity might end up ill defined in a simulation anyway. This would occur due to the difficulty of discretising the domain and moving the body in such a way that a control volume is only occupied by either a body or a fluid, but not both.

we actually solve for a quantity called volumetric flux, which has units of $\text{m}^3/\text{m}^2\text{s}$ and is the volume of fluid that flows through the finite control volume face per unit area and time. Notice that the units of volumetric flux reduce to units of velocity m s^{-1} , even though it is not the same. However, volumetric flux is considered to be representative of velocity if the size of the control volume is appropriately small for the problem being modelled or if supplemented by sub-grid scale models⁷. The same approximation propagates to all advected quantities through the advective flux in equation 1.10.

$$\mathbf{j}_{\text{adv}} = \psi \otimes \mathbf{u} \quad (1.10)$$

The formulation of the diffusive part \mathbf{j}_{dif} can be complicated, but regardless of how complicated it is, it will describe the transport of ψ due to differences in some potential. It is a flux that corresponds to smearing and dissipating concentrations of ψ in the system. Common ways of expressing this concept relies on equations similar to 1.11. Notice that the gradient field “points” to the region of increasing potential or “peaks”, therefore the diffusive flux is proportional to the negative of the gradient because the flux “points” to regions of decreasing potential or “valleys”.

$$\mathbf{j}_{\text{dif}} \propto -\nabla\theta \quad (1.11)$$

Constitutive relations of relevance to this work that might model this diffusive flux for various quantities are shown in equations 1.12. All these equations express exactly the same concept. Any potential that is concentrated somewhere, be it mechanical, thermal or chemical, will diffuse to its surroundings according to the negative of the potential gradient multiplied by a proportionality constant or function. From this interpretation arises an elegant perspective to the meaning of viscosity as describing transport of momentum, analogous to how thermal conductivity describes transport of thermal energy. In this sense, fluids with high viscosity effectively diffuse momentum from places with higher velocity to lower velocity and contacting surfaces, leading to stagnation, which is what is intuitively associated with high viscosity fluids.

Notice that Newton’s law of viscosity looks a bit different than the other laws without deliberately making it look the same. The reason is a difference in definition. Heat flux \mathbf{q} is defined from the beginning as a quantity that diffuses from high temperatures to low ones, however the viscous stress tensor τ contains no information about how stress dissipates, only its proportionality to velocity derivatives (i.e., shear rate). In order to add information about its dissipation, we have to change the sign. It is tempting to define a quantity Γ that denotes some linear momentum flux with a relation to the stress tensor, which would be aligned with the meaning of the heat flux by embedding

⁷a model that gives information about phenomena that is too small for the grid to resolve.

information on how stresses diffuses ($\Gamma = -\mu \nabla \mathbf{u} = -\tau$). Despite this consideration, this concept of momentum flux is not usually used in the literature with rare exceptions [Landau & Lifshitz, 1989, p. 11]. Therefore, even though the fundamental difference in the definition of the quantities makes the equations somehow not perfectly elegant, it will stay as is to avoid clashes with the wider literature.

$$\text{Fick's law of diffusion} \qquad \qquad \mathbf{j} = -D \nabla C \qquad (1.12a)$$

$$\text{Fourier's law of heat conduction} \qquad \mathbf{q} = -k \nabla T \qquad (1.12b)$$

$$\text{Newton's law of viscosity} \qquad -\tau = -\mu \nabla \mathbf{u} \qquad (1.12c)$$

It is important to observe at least a few details regarding modelling of diffusion phenomena. The quantities ψ and θ in equations 1.10 and 1.11 might be the same but this is not always the case. A common example of this is the transport of energy, where diffusion might be described as a function of temperature using Fourier's law of heat conduction (equation 1.12b). This will influence the way that the equation will be solved, such that \mathbf{j}_{dif} might be an implicit term (i.e., part of matrix) if ψ and θ are the same or can be made to be the same with some reasonable approximation, or an explicit term (i.e., part of the source) if ψ and θ are different.

Another consideration is that the proportionality constant/function (i.e., D , k , μ , etc) is only a scalar quantity if diffusion is an isotropic phenomena or can be approximated as such. In cases that diffusion cannot be approximated as isotropic, D will be a tensor to describe anisotropy.

A good example from both, phenomenological and modelling perspectives, can be discussed regarding heat transfer. In a graphite crystal lattice, typically a stack of planes, heat transfer phenomena is highly anisotropic. This would be true for any scale of heat transfer where crystal characteristics are dominant, or manufacture techniques that result in crystal alignment. In such case the scalar thermal conductivity k would have to be generalised to a thermal conductivity tensor κ , and require a generalised Fourier's law of heat conduction of the form $\mathbf{q} = -\kappa \cdot \nabla T$ or some other anisotropic generalised model. However, if the graphite structure under study is a multi-crystal one manufactured with techniques that emphasise random arrangement of the individual crystals in the structure, typical of nuclear graphite (Shen et al., 2015; Konings, 2012, vol. 4, p. 329, sec. 4.11.3; Haag et al., 1990), heat transfer can be treated as isotropic with negligible loss of accuracy. A different but related example of anisotropic phenomena could be diffusion or dissipation of momentum when modelling plastic flow of a polymer, glass, magma or metal forming, which is likely to be described by a tensor.

A different type of approximation relates to how laws formulated as equation 1.11

violate special relativity, since according to the equation a perturbation of θ is felt at infinity instantaneously [Ali & Zhang, 2005]. However, this only becomes relevant when the perturbation is highly localised for the dimensions of the system. Fourier's law of heat conduction would not be appropriate for modelling of pulsed laser heating of a sample significantly bigger than the heating spot for example. Such case would require a constitutive relation with more complex formulation.

Many possible examples of approximations could be mentioned, but the important observation is that modelling diffusion might require special considerations regarding the phenomena and the various degrees of approximation that are possible.

Joining equations 1.8-1.10 into a single one, a consolidated continuity equation 1.13 is formed. How flux will be expressed, if it is an implicit or an explicit term, etc, are all modelling decisions to be made.

$$\frac{\partial \psi}{\partial t} + \nabla \cdot (\psi \otimes \mathbf{u}) + \nabla \cdot \mathbf{j}_{\text{dif}} = S \quad (1.13)$$

Considering appropriate advective and diffusive terms for a phenomenon, all the basic equations used in this work arise from the continuity equation, one way or another, with examples shown in table 1.1. The previously undefined quantities in this table are e the specific internal energy, K the specific kinetic energy, and \mathbf{w} is the work flux, which will be explained in the appendix. These equations will be discussed in more detail in chapters 2, 3, and appendix A.2, which is entirely dedicated to the equation for conservation of total energy and its terms.

Table 1.1.: Forms of continuity equation

Property	ψ	\mathbf{j}_{adv}	\mathbf{j}_{dif}
Mass	ρ	$\rho \mathbf{u}$	0
Linear Momentum	$\rho \mathbf{u}$	$\rho \mathbf{u} \otimes \mathbf{u}$	$-\sigma$
Total Energy	$\rho(e + K)$	$\rho(e + K)\mathbf{u}$	$\mathbf{q} + \mathbf{w}$

The continuity equation will arise in other common contexts as well, such as electromagnetism and neutron diffusion (covered in chapter 3), and in not obvious ones, such as conservation of wave function probability in quantum mechanics. The power of the continuity equation as a mathematical expression of conservation laws cannot be overstated.

1.4.3. Verification, validation and benchmarking

Scientific code development is a challenging task that needs a multitude of checks in order to evaluate the progress from different perspectives. Verification and validation are two essential activities in every code development targeting the simulation of physical phenomena.

Code verification can be divided into 2 branches: numerical algorithm verification and software quality assurance [Oberkampf & Roy, 2010]. Numerical algorithm verification is a task that produces evidence that algorithms in the code are **mathematically** correct and free of errors. In order to accomplish this task, one needs an **objectively correct** reference solution. A more precise description of what this means and the ways it can be accomplished will be dealt with in chapter 5.

Software Quality Assurance (SQA) involves implementing a development environment that ensures program routines do what they should and prevents errors from being introduced (or re-introduced) into the code. The main objectives of SQA can be achieved based on consistent good practices. Unit, component and system testing, regression tests, testing harness, code coverage, etc, are all necessary components of a robust environment for code development. Commonly overlooked in academia, this is a big part of the difference between the degree of belief given to commercial codes, where it is assumed to have been properly performed, even if it's not necessarily true. In this work, all verification cases and relevant cases that exercise parts of the code that are difficult to verify were used as testing procedure during development. While it takes a significant amount of time to automate verification and implement even a partial SQA procedure, this simply cannot be skipped or else conclusions derived can be fundamentally challenged (and flawed). Further considerations on the topic of SQA are shortly presented in appendix C.1 but should be sought in the software engineering literature by anyone engaged with code development.

Validation consists on evaluating if a model predicts **physical** reality with accuracy matching or exceeding an established criteria. In order to accomplish this task, one needs **validation criteria and experimental data**.

Therefore, we formalise verification as a mathematics problem and validation as a physics problem. Verification answers the question “am I solving the equations right?”, whatever equations these might be. Validation answers the question “am I solving the right equations?”. Clearly, verification must be carried out before validation.

Benchmarking or code comparison is a task that is many times confusingly referred to as “verification” [Malicki et al., 2019; Hedayat, 2016; Ma et al., 2015; Dahmani et al., 2011]. **Code comparison is not verification** [Oberkampf et al., 2003]. The reason is that any number of codes “agreeing” on a result, whatever arbitrary/subjective

criteria for agreement is established, is not any sort of objective proof of correctness. Fundamentally, no subjective criteria can lead to objective conclusions. This does not imply that code comparisons are not useful. For example, a code comparison may indicate that code X can be replaced by faster code Y with a certain loss of accuracy. It is a useful information! However, it says nothing about which code is correct.

1.5. Motivation and objectives

With recently renewed interest in MSRs and the study of the MSFR, the motivation of this thesis is to fill gaps in knowledge that could contribute to better understanding and future commercialisation of such reactor.

The MSFR curved walls and turbulent open cavity require the use of specialised codes such as ATARI to model its phenomena. During the SAMOFAR project, a task to analyse selected transients of interest in the MSFR was proposed, to which this thesis should contribute as best as possible to evaluating the safety of the concept.

Due to the high melting point of inorganic molten salts, it is likely that several transient scenarios might lead to at least partial freezing of the salt. The possibility of this salt freezing and the lack of information on the consequences of heat exchanger blockage represents an unknown. Current analytical methodologies do not take this into account, and it is important to consider if this is an issue that requires special attention and how to tackle it.

Finally, the requirement of unstructured meshes and turbulence models, adds a significant source of uncertainty and greatly increases the computational burden of routine analysis. It is unclear if the uncertainties introduced are small enough to allow a robust safety case. It is also uncertain if such wide application of these specialised codes would be practical. It would be desirable to allow routine analysis of the reactor by legacy codes, and reserve the use of specialised codes to exceptional situations in order to avoid an analytical workflow that is too expensive/time consuming/burdensome.

Taking these issues into consideration, **the objectives of this thesis are (1) to analyse the MSFR together with partners to evaluate the safety of the concept, (2) consider how to include solidification and melting in an analytical methodology, and (3) evaluate options that could improve reactor design by reducing uncertainties and the burden of routine use of ATARI.**

The ATARI code developed is the tool to execute the analyses for this thesis. Even if its application is relatively costly, it could still fit a niche in a wider analytical methodology if its models are carefully considered. We shall try to make this judicious choice and develop an efficient tool by design, not chance, to carry out the required analysis in this

investigation. To do this, we shall pay special attention to the models, their assumptions and how to tie all parts into a whole application purposefully designed.

A significant portion of the material is applicable to nuclear reactors in general, but it will focus on MSRs whenever the scope is narrowed.

1.6. Thesis structure

To develop a thorough understanding of the models in ATARI, the initial part of the thesis focuses on the theoretical background to craft the tool.

In chapter 2 we use the understanding of the continuity equation as a general conservation law gained in this introduction to discuss single-phase flow from a unified perspective. The discussion is extended to a multi-phase flow model, where the understanding of the continuity equation is once again used to understand the averaging process of a general quantity and obtain correct diffusion terms. We narrow down the scope of the multi-phase model to the particular case of a fluid permeating a solid structure due to its practical importance to reactor analysis. Additional consideration will be given to solidification and melting of the fluid, where a computationally economical model will be chosen to tackle the phenomena and adapt it to our needs.

Chapter 3 discusses neutronics. Equations, cross-section parametrization, approximations and special considerations will be presented. The unique situation where the fissile material is a flowing fluid will be of particular relevance. A significant part of the discussion is devoted into understanding why the so well known diffusion approximation is such a powerful model and why it fits our approach to fluid dynamics.

The equations of chapters 2 and 3 are put together to develop two multiphysics solvers in chapter 4. This chapter will presents the program flow, loop arrangement and how the solution is controlled to reach convergence in an economical way. Architecture of the solver is given particular importance to allow easy modularity and expansion of ATARI.

Once the theoretical background of the solver is established, its use begins. In chapter 5, analytical solutions of a coupled problem and a semi-analytical solution to solidification in 1D are used to test ATARI's mathematical correctness in a limited scope. The intention is to verify the basic functionalities and the solidification model implemented. A test harness is constructed from these to perform regression tests during further developments.

In chapter 6 ATARI is applied to different case studies. A comparative benchmark is performed between the codes participating in the SAMOFAR project, and the same

codes later perform the analysis of the MSFR. The case study of the freezing of a simplified heat exchanger case is performed to study the possible effects of freezing on the MSFR heat exchanger, which is not explicitly simulated. To conclude the chapter, a new reactor design is proposed using flow baffles to suppress turbulence and arrive at a design that might be analysed by system codes instead of ATARI.

Finally, chapter 7 concludes this work, summarising the main findings of this research and charting the way forward.

2. Fluid mechanics

Πάντα ρεῖ.

(Heraclitus of Ephesus)

Fluids are substances that continuously deform under the action of stress coplanar to its cross section (i.e., shear stress). This is a very succinct way to categorise fluids by the result of an action that causes deformation, which is their defining characteristic. The action of stress normal to a cross section causes isotropic compression/expansion without deformation, therefore cannot be used as a criteria. It also differentiates fluids from solids, which do not continuously deform under shear stress (i.e., they break at some point). Some of the central ideas or assumptions in the topic are conveyed by this definition, such as:

- A lot of substances are or can be made to behave like fluids, even solids under fluidising conditions (e.g., quicksand, soil liquefaction during earthquakes and particles in a fluidised bed reactor).
- Fluids are treated as continuous materials (i.e. continuum assumption). Its discrete atomistic nature is abstracted into macroscopic properties.
- Shearing properties are fundamental to understanding the phenomena, including turbulence (i.e. shear induced instabilities).

Fluid mechanics is the study of these substances at rest (i.e. fluid statics) or in motion (i.e. fluid dynamics). The latter generally more relevant, specially for this work which needs to model it.

In this chapter, the equations implemented in ATARI will be derived in detail. Such detailed derivation has not been done before and a thorough explanation of the procedure and steps required is currently missing, scattered in multiple works or incomplete.

We will start by obtaining single-phase flow equations, which will later be expanded into a particular model of multiphase flow that considers interpenetrating substances [Drew & Passman, 1999]. This multiphase model will be simplified to the particular case of a fluid permeating through a solid structure, which will be called "homogenised fluid dynamics". The model can also be found by the name of "porous medium" in the literature [Fiorina et al., 2015], but in this thesis we consider this term appropriate for

flow through materials like sand, where there is some sort of particle distribution and statistical description of the medium material, instead of a known structure like we are interested.

As will be discussed in chapter 3, description of the flow through the homogenised fluid dynamics model is well aligned with the diffusion approximation to simulate neutron behaviour, where homogenisation of the spatial heterogeneity is necessary in order to eliminate any explicit description of strong neutron absorbers.

Finally, a model for solidification and melting of a fluid will be shown following the enthalpy-porosity approach, and using the understanding obtained from the derivation of multiphase flow equations, it will be easily expanded to take into account the homogenised fluid-solid model.

To follow this chapter, a basic understanding of fluid mechanics is assumed, and can be obtained in multiple books [Landau & Lifshitz, 1989; Kundu et al., 2016; Moukalled et al., 2016; Pletcher et al., 1984].

2.1. Single-phase flow equations

In order to model the fluid system, a set of equations describing the conservation of mass, linear momentum and total energy are required. These are collectively referred by modern fluid dynamics literature as the Navier-Stokes equations¹ and shown respectively as equations 2.1, 2.2, and 2.3. The equations arise from substituting properties shown on table 1.1 into the continuity equation 1.13.

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

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla \cdot (-\sigma) = \mathbf{S}_u \quad (2.2)$$

$$\frac{\partial \rho(e + K)}{\partial t} + \nabla \cdot (\rho(e + K)\mathbf{u}) + \nabla \cdot (\mathbf{q} + \mathbf{w}) = S_{eK} \quad (2.3)$$

Details of the parameters, such as meaning and units, are found on table 2.1. These equations are used in modelling of single-phase flows with heat transfer. At this point, equations 2.1, 2.2 and 2.3 contain undefined diffusion terms. The **source terms** S_m , \mathbf{S}_u and S_{eK} **are arbitrary and can contain any number of sources abstracted into it**, which will be expanded when relevant.

¹Historically, the Navier-Stokes equation refers to only conservation of momentum.

Table 2.1.: Parameters of the single-phase Navier-Stokes equations.

Parameter	Units	Meaning
ρ	kg m^{-3}	Density
S_m	$\text{kg m}^{-3} \text{s}^{-1}$	Mass source
σ	$\text{Pa}, \text{N m}^{-2}, \text{kg m}^{-1} \text{s}^{-2}$	(Total) Stress tensor
τ	$\text{Pa}, \text{N m}^{-2}, \text{kg m}^{-1} \text{s}^{-2}$	Viscous stress tensor
μ	$\text{Pa s}, \text{kg m}^{-1} \text{s}^{-1}$	Viscosity
\mathbf{g}	m s^{-2}	Gravitational acceleration
\mathbf{S}_u	$\text{N m}^{-3}, \text{kg m}^{-2} \text{s}^{-2}$	Momentum source
\mathbf{q}	$\text{W m}^{-2}, \text{J m}^{-2} \text{s}^{-1}, \text{kg s}^{-3}$	Heat flux
k	$\text{W m}^{-1} \text{K}^{-1}, \text{kg m s}^{-3} \text{K}^{-1}$	Thermal conductivity
c_p	$\text{J kg}^{-1} \text{K}^{-1}, \text{m}^2 \text{s}^{-2} \text{K}^{-1}$	Specific heat capacity
\mathbf{w}	$\text{W m}^{-2}, \text{J m}^{-2} \text{s}^{-1}, \text{kg s}^{-3}$	Work flux
K	$\text{J kg}^{-1}, \text{m}^2 \text{s}^{-2}$	Specific Kinetic energy
S_{eK}	$\text{W m}^{-3}, \text{J m}^{-3} \text{s}^{-1}, \text{kg m}^{-1} \text{s}^{-3}$	Energy source based on internal energy
S_{he}	$\text{W m}^{-3}, \text{J m}^{-3} \text{s}^{-1}, \text{kg m}^{-1} \text{s}^{-3}$	Energy source based on enthalpy
\mathbf{u}	m s^{-1} (or $\text{m}^3 \text{m}^{-2} \text{s}^{-1}$)	Velocity (or advective flux)
p	$\text{Pa}, \text{N m}^{-2}, \text{kg m}^{-1} \text{s}^{-2}$	Pressure
e	$\text{J kg}^{-1}, \text{m}^2 \text{s}^{-2}$	Specific Internal Energy
h	$\text{J kg}^{-1}, \text{m}^2 \text{s}^{-2}$	Specific Enthalpy

The equation for conservation of mass typically has no diffusion term. Mass can diffuse, however the time scales of mass diffusion are too large to be relevant in the usual flow simulation, unless this phenomena in particular is targeted (e.g., mass diffusion through a membrane). The source S_m is considered 0 in equation 2.4, assuming no ingress of fluid or changes between fluid phases, which is appropriate for now.

For the conservation of momentum, the stress tensor σ in equation 2.2 is treated as detailed in appendix A.1, under the assumption of an isotropic newtonian incompressible fluid with constant viscosity. A buoyancy term $\mathbf{S}_u = \rho \mathbf{g}$ is expanded from the arbitrary source to consider the gravity force as well, resulting in equation 2.5. Additional momentum sources, such as pumps, are certainly possible but will stay abstracted for now.

The internal energy e in equation 2.3 becomes enthalpy h in equation 2.6 through a process detailed in appendix A.2. The viscous dissipation $\nabla \cdot (-\tau \cdot \mathbf{u})$ is neglected since it is marginal compared to all other contributions in the typical reactor.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (2.4)$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) - \mu \nabla^2 \mathbf{u} = -\nabla p + \rho \mathbf{g} + \mathbf{S}_u \quad (2.5)$$

$$\begin{aligned} & \frac{\partial \rho h}{\partial t} - \frac{\partial p}{\partial t} + \nabla \cdot (\rho h \mathbf{u}) + \nabla \cdot \mathbf{q} \\ & + \frac{\partial \rho K}{\partial t} + \nabla \cdot (\rho K \mathbf{u}) = \rho \mathbf{g} \cdot \mathbf{u} + S_{he} \end{aligned} \quad (2.6)$$

Only the heat flux \mathbf{q} in equation 2.6 is left undefined. This term could be replaced by Fourier's law of heat conduction, with the limitation that it would have to be an explicit term. In OpenFOAM, an assumption of approximately constant specific heat capacity c_p is used, allowing the algebraic manipulation shown in equation 2.7. This manipulation recasts Fourier's law of heat conduction into a gradient of enthalpy under the constant c_p assumption, allowing heat flux to be treated implicitly.

$$\mathbf{q} = -k \nabla T = -\frac{c_p}{c_p} k \nabla T = -\frac{k}{c_p} \nabla c_p T = -\alpha \nabla h \quad (2.7)$$

It must be emphasized that in the fluid dynamics literature α is commonly defined as $\alpha = \frac{k}{\rho c_p}$, which is the thermal diffusivity with units of $\text{m}^2 \text{s}^{-1}$. Thermal diffusivity is a measure of thermal inertia, or how fast temperature differences smoothens, just as momentum diffusivity $\nu = \frac{\mu}{\rho}$ (i.e., kinematic viscosity), with the exact same units, is a measure of how fast momentum differences smoothens. This is **not** the concept that α in equation 2.7 represents. Confusingly enough, it is possible to find both interpretations of α inside OpenFOAM; the thermal diffusivity one is used in specialised incompressible solvers, and the one given by 2.7 is used in general solvers capable of compressible and incompressible flows. The base solver for ATARI is a general one, therefore it will use the interpretation of 2.7, with units of $\text{kg m}^{-1} \text{s}^{-1}$, which is **nothing but an artifice** in order to allow implicit treatment of heat flux. Nevertheless, if this artifice is used, equation 2.6 turns into 2.8.

$$\begin{aligned} & \frac{\partial \rho h}{\partial t} - \frac{\partial p}{\partial t} + \nabla \cdot (\rho h \mathbf{u}) + \nabla \cdot (-\alpha \nabla h) \\ & + \frac{\partial \rho K}{\partial t} + \nabla \cdot (\rho K \mathbf{u}) = \rho \mathbf{g} \cdot \mathbf{u} + S_{he} \end{aligned} \quad (2.8)$$

Leaving turbulence aside, which is a topic on its own, these equations allow the simulation of single-phase flow with mild assumptions. In chapter 3 we will tackle neutronics, which will result in the definition of a nuclear heat source to be included in equation 2.6 through S_{he} .

2.2. Multiphase flow model

There exists multiple ways of simulating multiphase flow phenomena, and one of the main criteria that will drive decisions is the need (or lack of) resolution of the interface. For 2 fluids, the presence of a moving interface is very challenging and a major topic of research. In our particular application, we would like to simulate a fluid flowing through a static solid structure; therefore, resolution of the interface is ironically trivial — that is what traditional single-phase CFD simulations are. However, our intent is exactly the opposite; to not resolve the interface between a fluid and a solid in a region of space. This choice results in a simplified model that is computationally cheap, with uncertainties that are compatible with the expectations of a full core simulation, and aligned with the neutronics model that will be shown on chapter 3. Despite the differences, the foundation of this simplified model is similar to the approach of interpenetrating fluids in what is called Eulerian multiphase modelling.

An averaged multiphase flow model is described below, following previous works [Drew, 1983; Saurel & Abgrall, 1999; Saurel & Lemetayer, 2001]. These conservation equations describe the flow of N interpenetrating fluids through a phase-averaging process, but a description of topology is missing. The topological equation 2.9 is an approximate model for topology, and is the ensemble-average of a more general model involving a discrete description of topology with phase indicator functions. The advantage of this model is that simplifications will lead directly to a solid phase with known structure (topology), like heat exchangers, whereas an approach such as the Darcy-Forchheimer model will lead to a proper porous medium, like sand.

$$\frac{\partial \gamma_n}{\partial t} + \mathbf{u}_i \nabla \gamma_n = 0 \quad (2.9)$$

$$\frac{\partial \gamma_n \rho_n}{\partial t} + \nabla \cdot (\gamma_n \rho_n \mathbf{u}_n) = \dot{m}_{Nn} \quad (2.10)$$

$$\frac{\partial \gamma_n \rho_n \mathbf{u}_n}{\partial t} + \nabla \cdot (\gamma_n \rho_n \mathbf{u}_n \otimes \mathbf{u}_n) = -\nabla \gamma_n p_n + p_i \nabla \gamma_n + \dot{m}_{Nn} \mathbf{u}_i + F_{\text{drag},i} + \mathbf{S}_{\mathbf{u},n} \quad (2.11)$$

$$\begin{aligned} & \frac{\partial \gamma_n \rho_n E_n}{\partial t} + \nabla \cdot (\gamma_n \rho_n E_n \mathbf{u}_n) \\ &= -\nabla \cdot \gamma_n p_n \mathbf{u}_n + (p_i \nabla \gamma_n) \cdot \mathbf{u}_i + \dot{m}_{Nn} E_i + F_{\text{drag},i} \cdot \mathbf{u}_i + Q_i + S_{eK,n} \end{aligned} \quad (2.12)$$

where for a phase n , γ_n is the phase fraction, and $E_n = e_n + K_n$ the total energy. Parameters shown on table 2.1 are also present with a subscript n to denote the phase, and additional parameters are detailed on table 2.2.

Table 2.2.: Expanded parameters of the multiphase equations.

Parameter	Units	Meaning
γ_n	m^{-3}	Volumetric phase fraction
\dot{m}_{Nn}	$\text{kg m}^{-3} \text{s}^{-1}$	Mass transfer source between phase n and all other N phases
\mathbf{u}_i	m s^{-1}	Interfacial velocity
p_i	$\text{Pa}, \text{N m}^{-2}, \text{kg m}^{-1} \text{s}^{-2}$	Interfacial pressure
$F_{\text{drag},i}$	$\text{N m}^{-3}, \text{kg m}^{-2} \text{s}^{-2}$	Interfacial drag force
Q_i	$\text{W m}^{-3}, \text{J m}^{-3} \text{s}^{-1}, \text{kg m}^{-1} \text{s}^{-3}$	Interfacial heat source

Equations 2.10, 2.11 and 2.12 are phase average equivalent to the single-phase equations presented previously. The topological equation 2.9 describes a change of topology, where the phase fraction is such that:

$$\sum_{n=1}^N \gamma_n = 1 \quad (2.13)$$

Crucial in these equations are the interfacial terms with subscript i , that need to be modelled somehow. When resolving the interface is an objective, interfacial quantities might be obtainable from mechanistic approaches, which is not our intent. For a non-resolved interface, these quantities will be part of Sub-Grid Scale (SGS) models, which uses assumptions and correlations to recover information from phenomena happening at a scale that is too small to be explicitly resolved; turbulence models are a common example. One approach is to solve additional equations, also derived from the continuity equation, in order to obtain $F_{\text{drag},i}$ and Q_i as a function of the characteristics of some dispersed quantity (e.g., bubbles or droplets), such as size. The interfacial pressure is discussed by Saurel & Abgrall [1999], where an average pressure for the mixture is suggested as shown below.

$$p_i = \sum \gamma_n p_n \quad (2.14)$$

In this interpenetrating model, it is necessary to solve N mass, momentum and energy conservations, and $N - 1$ phase fraction ones since at least one must be obtained from relation 2.13 in order to not over-specify the system. This is similar to how it is not possible to specify the 3 angles of a triangle, only 2, and the last angle is obtained by subtracting from 180 degrees; or how it is not possible to specify pressure, temperature and density of water, where one of the quantities is obtained from an equation of state tying them all together.

Although it seems like the original model proposed by the articles made no statement regarding diffusive terms, this is not entirely true since conservation of momentum did have a pressure term, which originates from the isotropic contributions of the stress tensor. The meaning behind the multiphase equations 2.10-2.12 is that the continuity equation 1.8 can be phase-averaged as shown in equation 2.15 as long as appropriate sources are given to model the exchange of ψ between phases, including the diffusive terms.

$$\frac{\partial \gamma_n \psi_n}{\partial t} + \nabla \cdot (\gamma_n \mathbf{j}_n) = S \quad (2.15)$$

Using this idea, even though no explicit statement regarding multiphase diffusion was made in the original articles, it is still perfectly reasonable to recover it. The caveat is that the proportionality constant for the diffusive flux might be difficult to obtain and anisotropic in many cases.

2.3. Homogenised model for flow through a structure

Using the previous concepts, it is possible to specialize the general multiphase model into one that is fit for the purpose of a fluid flowing through a static solid structure, which we shall do in a step-wise manner.

Considering first the 2 fluids case, denoted by subscripts 1 and 2, it would be necessary to solve at least 7 equations: conservation of mass, momentum and energy in each fluid and a single equation for advection of fluid fraction, since $\gamma_2 = 1 - \gamma_1$. The reciprocity between mass transfers terms $\dot{m}_{21} = -\dot{m}_{12}$ also arises in this case.

$$\frac{\partial \gamma_1}{\partial t} + \mathbf{u}_i \nabla \gamma_1 = 0 \quad (2.16)$$

$$\frac{\partial \gamma_1 \rho_1}{\partial t} + \nabla \cdot (\gamma_1 \rho_1 \mathbf{u}_1) = \dot{m}_{12} \quad (2.17)$$

$$\frac{\partial \gamma_1 \rho_1 \mathbf{u}_1}{\partial t} + \nabla \cdot (\gamma_1 \rho_1 \mathbf{u}_1 \otimes \mathbf{u}_1) = -\nabla \gamma_1 p_1 + p_i \nabla \gamma_1 + \dot{m}_{12} \mathbf{u}_i + F_{\text{drag},i} + \mathbf{S}_{\mathbf{u},1} \quad (2.18)$$

$$\begin{aligned} & \frac{\partial \gamma_1 \rho_1 E_1}{\partial t} + \nabla \cdot (\gamma_1 \rho_1 E_1 \mathbf{u}_1) \\ &= -\nabla \cdot \gamma_1 p_1 \mathbf{u}_1 + (p_i \nabla \gamma_1) \cdot \mathbf{u}_i + \dot{m}_{12} E_i + F_{\text{drag},i} \cdot \mathbf{u}_i + Q_i + S_{eK,1} \end{aligned} \quad (2.19)$$

$$\frac{\partial \gamma_2 \rho_2}{\partial t} + \nabla \cdot (\gamma_2 \rho_2 \mathbf{u}_2) = -\dot{m}_{12} \quad (2.20)$$

$$\frac{\partial \gamma_2 \rho_2 \mathbf{u}_2}{\partial t} + \nabla \cdot (\gamma_2 \rho_2 \mathbf{u}_2 \otimes \mathbf{u}_2) = -\nabla \gamma_2 p_2 + p_i \nabla \gamma_2 - \dot{m}_{12} \mathbf{u}_i + F_{\text{drag},i} + \mathbf{S}_{\mathbf{u},2} \quad (2.21)$$

$$\begin{aligned} & \frac{\partial \gamma_2 \rho_2 E_2}{\partial t} + \nabla \cdot (\gamma_2 \rho_2 E_2 \mathbf{u}_2) \\ &= -\nabla \cdot \gamma_2 p_2 \mathbf{u}_2 + (p_i \nabla \gamma_2) \cdot \mathbf{u}_i - \dot{m}_{12} E_i + F_{\text{drag},i} \cdot \mathbf{u}_i + Q_i + S_{eK,2} \end{aligned} \quad (2.22)$$

Considering the particular 2 fluids case where the fluids do not exchange mass with each other, all mass transfer terms \dot{m} are eliminated. We shall go one step further and consider the case where we turn fluid 2 into a static solid in order to simulate a structure. If this solid is absolutely static, with no significant vibrations, all terms depending on interface velocity \mathbf{u}_i are also eliminated, because the fluid-solid interface is known and static. In this case, it is not necessary to solve conservation of phase fraction, mass and momentum for the solid, and there is no need to consider kinetic energy in conservation of total energy of the solid either.

$$\frac{\partial \gamma_1 \rho_1}{\partial t} + \nabla \cdot (\gamma_1 \rho_1 \mathbf{u}_1) = 0 \quad (2.23)$$

$$\frac{\partial \gamma_1 \rho_1 \mathbf{u}_1}{\partial t} + \nabla \cdot (\gamma_1 \rho_1 \mathbf{u}_1 \otimes \mathbf{u}_1) = -\nabla \gamma_1 p_1 + p_i \nabla \gamma_1 + F_{\text{drag},i} + \mathbf{S}_{\mathbf{u},1} \quad (2.24)$$

$$\frac{\partial \gamma_1 \rho_1 E_1}{\partial t} + \nabla \cdot (\gamma_1 \rho_1 E_1 \mathbf{u}_1) = -\nabla \cdot \gamma_1 p_1 \mathbf{u}_1 + Q_i + S_{eK,1} \quad (2.25)$$

$$\frac{\partial \gamma_2 \rho_2 e_2}{\partial t} = Q_i + S_{e,2} \quad (2.26)$$

We can further simplify the pressure terms in equation for conservation of linear momentum 2.24 considering that we assumed a static solid as the second phase, therefore, transmission of pressure waves/vibrations in the liquid-solid interface is ignored, acoustic effects between phases are ignored in other words. As a consequence, the interfacial pressure is equal to the fluid pressure $p_i = p_1$, and the subscript for pressure can simply be dropped, since it became redundant. Using equation 2.27, we can simplify the pressure terms into a single one as shown in equation 2.28.

$$\nabla \gamma p = \gamma \nabla p + p \nabla \gamma \implies -\nabla \gamma p + p \nabla \gamma = -\gamma \nabla p \quad (2.27)$$

$$\frac{\partial \gamma_1 \rho_1 \mathbf{u}_1}{\partial t} + \nabla \cdot (\gamma_1 \rho_1 \mathbf{u}_1 \otimes \mathbf{u}_1) = -\gamma_1 \nabla p + F_{\text{drag},i} + \mathbf{S}_{\mathbf{u},1} \quad (2.28)$$

The equation for energy in the solid structure can also be simplified by considering an ideal non-expanding solid, whose internal energy e is expressed by equation 2.29, where c and T are the heat capacity (considered constant) and temperature of the solid respectively. This allows redefining equation 2.26 into 2.30.

$$de = c dT \quad (2.29)$$

$$\frac{\partial \gamma_2 \rho_2 c_2 T_2}{\partial t} = Q_i + S_{e,2} \quad (2.30)$$

Until this point, only the original equations have been manipulated to reach the desired model. However, the model as written is incomplete due to the lack of diffusive terms. Whereas the original articles focused on flows strongly dominated by advective terms, such as detonations, our interest lies in flows of much lower velocity, therefore diffusive terms must be recovered.

With the insight obtained from equation 2.15, we will simply add the diffusive terms by multiplying the diffusive fluxes by the phase fraction. In addition, we will also recover the buoyancy terms. This model expansion leads to equations 2.31-2.34, where the subscript 1 for fluid has been replaced by “f”, and 2 for solid by “s”.

$$\frac{\partial \gamma_f \rho_f}{\partial t} + \nabla \cdot (\gamma_f \rho_f \mathbf{u}_f) = 0 \quad (2.31)$$

$$\frac{\partial \gamma_f \rho_f \mathbf{u}_f}{\partial t} + \nabla \cdot (\gamma_f \rho_f \mathbf{u}_f \otimes \mathbf{u}_f) + \nabla \cdot (-\gamma_f \mathbf{T}) = -\gamma_f \nabla p + \gamma_f \rho_f \mathbf{g} + F_{\text{drag},i} + \mathbf{S}_{\mathbf{u}} \quad (2.32)$$

$$\frac{\partial \gamma_f \rho_f E_f}{\partial t} + \nabla \cdot (\gamma_f \rho_f E_f \mathbf{u}_f) + \nabla \cdot (\gamma_f \mathbf{q}) = -\nabla \cdot (\gamma_f p \mathbf{u}_f) + \gamma_f \rho_f \mathbf{g} \cdot \mathbf{u}_f + Q_i + S_{eK} \quad (2.33)$$

$$\frac{\partial \gamma_s \rho_s c_s T_s}{\partial t} - \nabla \cdot (\gamma_s k_s \nabla T_s) = Q_i + S_s \quad (2.34)$$

As in the single-phase case, it is convenient to reformulate equation 2.33 using enthalpy as a variable, leading to equation 2.35.

$$\begin{aligned} & \frac{\partial \gamma_f \rho_f h_f}{\partial t} - \frac{\partial \gamma_f p}{\partial t} + \nabla \cdot (\gamma_f \rho_f h_f \mathbf{u}_f) + \nabla \cdot (\gamma_f \mathbf{q}) \\ & + \frac{\partial \gamma_f \rho_f K}{\partial t} + \nabla \cdot (\gamma_f \rho_f K \mathbf{u}_f) = \gamma_f \rho_f \mathbf{g} \cdot \mathbf{u}_f + Q_i + S_{he} \end{aligned} \quad (2.35)$$

We shall replace the diffusive fluxes $-\tau$ and \mathbf{q} in equations 2.32-2.33 by the terms discussed in section 2.1 to prepare for an algebraic manipulation that is coming. In addition, we shall assume γ_f to be approximately constant, which is reasonable considering the solid structures typical of nuclear reactors (e.g.: heat exchangers, fuel assemblies, etc). This allows γ_f to be moved into and out of derivatives. At the boundaries between regions where γ_f cannot be assumed to be constant, a special treatment is applied using “pressure baffles” (i.e., a form of pressure boundary condition that is internal to the domain).

$$\frac{\partial \gamma_f \rho_f \mathbf{u}_f}{\partial t} + \nabla \cdot (\gamma_f \rho_f \mathbf{u}_f \otimes \mathbf{u}_f) - \gamma_f \mu \nabla^2 \mathbf{u}_f = -\gamma_f \nabla p + \gamma_f \rho_f \mathbf{g} + F_{\text{drag},i} + \mathbf{S}_u \quad (2.36)$$

$$\begin{aligned} & \frac{\partial \gamma_f \rho_f h_f}{\partial t} - \frac{\partial \gamma_f p}{\partial t} + \nabla \cdot (\gamma_f \rho_f h_f \mathbf{u}_f) + \nabla \cdot (-\gamma_f \alpha \nabla h) \\ & + \frac{\partial \gamma_f \rho_f K}{\partial t} + \nabla \cdot (\gamma_f \rho_f K \mathbf{u}_f) = \gamma_f \rho_f \mathbf{g} \cdot \mathbf{u}_f + Q_i + S_{he} \end{aligned} \quad (2.37)$$

A numerical problem will most likely arise from solving these equations. Typical systems of interest for modelling of a fluid phase passing through a solid one (e.g., water passing through a rod bundle or a heat exchanger), involve fluid fraction that changes drastically at the inlet and outlet of regions containing a solid structure, even though it is mostly constant in between. This results in an abrupt discontinuity of the velocity field at inlet and outlet interfaces. In order to address this issue, it is possible to solve for a homogenised velocity as defined in equation 2.38, which is continuous.

$$\mathbf{u}_{h,f} = \gamma_f \mathbf{u}_f \quad (2.38)$$

Considering the substitution of equality 2.38 into equations 2.31, 2.36 and 2.37 leads to equations 2.39-2.41. One has to take care to calculate the value of kinetic energy K properly, using real fluid velocities instead of homogenised ones.

$$\frac{\partial \gamma_f \rho_f}{\partial t} + \nabla \cdot (\rho_f \mathbf{u}_{h,f}) = 0 \quad (2.39)$$

$$\frac{\partial \rho_f \mathbf{u}_{h,f}}{\partial t} + \nabla \cdot \left(\frac{\rho_f \mathbf{u}_{h,f}}{\gamma_f} \otimes \mathbf{u}_{h,f} \right) - \mu \nabla^2 \mathbf{u}_{h,f} = -\gamma_f \nabla p + \gamma_f \rho_f \mathbf{g} + F_{\text{drag},i} + \mathbf{S}_u \quad (2.40)$$

$$\begin{aligned} & \frac{\partial \gamma_f \rho_f h_f}{\partial t} - \frac{\partial p}{\partial t} + \nabla \cdot (\rho_f h_f \mathbf{u}_{h,f}) + \nabla \cdot (-\gamma_f \alpha \nabla h) \\ & + \frac{\partial \gamma_f \rho K}{\partial t} + \nabla \cdot (\rho K \mathbf{u}_{h,f}) = \rho \mathbf{g} \cdot \mathbf{u}_{h,f} + Q_i + S_{he} \end{aligned} \quad (2.41)$$

Finally, it is necessary to address the interfacial drag force $F_{\text{drag},i}$ and interfacial heat source Q_i between phases. While there are multiple models that could be considered, as a first approximation for Q_i we shall adopt a simple volumetric heat transfer rate 2.42, with parameters detailed on table 2.3.

$$Q_i = U A_i (\Delta T) \quad (2.42)$$

This is a 0-dimensional (point) model that abstracts the conduction path and assumes that heat flow between solid and fluid are equal, resulting in relation $Q_{i,f \rightarrow s} = -Q_{i,s \rightarrow f}$. In this simplified description, the only parameter that depends on interfacial quantities is the interfacial volumetric area, which is the heated surface of a heat exchanger in contact with fluid in a certain volume.

Table 2.3.: Parameters of simple heat transfer model.

Parameter	Units	Meaning
Q_i	W m^{-3}	Interfacial heat source
U	$\text{W m}^{-2} \text{K}^{-1}$	Overall heat transfer coefficient
A_i	m^{-1} (or $\text{m}^2 \text{m}^{-3}$)	Interfacial volumetric area
ΔT	K	Temperature difference between materials

The quantity U can be defined in multiple ways depending on the problem but, in the simplest case, it will be equal to the heat transfer coefficient h (not to be confused with specific enthalpy) between fluid and solid, which can be obtained from Nusselt number shown in equation 2.43. This dimensionless number can be estimated from empirical correlations involving other dimensionless numbers such as Reynolds number in forced convection cases, shown in equation 2.44, or Rayleigh number for natural convection. The parameters are described on table 2.4.

$$\text{Nu} = \frac{hL}{k} \quad (2.43)$$

$$\text{Nu} = a\text{Re}^b\text{Pr}^c + d \quad (2.44)$$

Table 2.4.: Nusselt number parameters and empirical correlation.

Parameter	Units	Meaning
Nu	-	Nusselt number
h	$\text{W m}^{-2} \text{K}^{-1}$	Heat transfer coefficient
L	m	Characteristic length
k	$\text{W m}^{-1} \text{K}^{-1}$	Fluid thermal conductivity
a, b, c, d	-	Empirical coefficients
Re	-	Reynolds number
Pr	-	Prandtl number

A more elaborate model might consider a higher dimensional approach between solid and liquid, for example, where heat flux might be unequal and pass through the 1D domain, which would allow for time-dependent heating or cooling of the path of conduction. Higher dimensional models are particularly beneficial if we are interested in transient thermal stresses in order to evaluate structural integrity.

The final interfacial quantity that needs to be defined is the interfacial drag force $F_{\text{drag},i}$. Similar to Q_i , there are multiple ways to approach this term depending on the problem. For the common problem of a fluid moving through some channel or pipe, the drag force can be interpreted as a compound of viscous forces and surface roughness at the fluid-wall interface. The result of this wall friction is a pressure loss at the outlet channel, which can be evaluated for a fully developed steady incompressible flow in a pipe by the Darcy-Weisbach equation 2.45, with parameters shown on table 2.5.

$$\frac{dp}{dx} = -f_D \frac{\rho}{2D_h} \mathbf{u}^2 = -f_D \frac{\rho}{2D_h} \frac{\mathbf{u}_{h,f}^2}{\gamma_f^2} \quad (2.45)$$

It is important to use the correct friction factor value and remember that the Darcy friction factor is 4 times the Fanning one, because Fanning was derived using hydraulic radius instead of diameter. The friction factor can be obtained by different methods, such as Moody charts or correlations applicable to certain situations [McKEON et al., 2005].

In ATARI, f_D is actually a tensor, not a scalar, allowing differential treatment of the drag force in multiple directions (i.e., anisotropic drag force). While obtaining the values of a full arbitrary tensor might be difficult, if the flow path is well determined, it could

Table 2.5.: Parameters of Darcy-Weisbach empirical correlation.

Parameter	Units	Meaning
$\frac{dp}{dx}$	Pa m^{-1}	Linear pressure drop
f_D	-	Darcy friction factor
D_h	m	Hydraulic diameter

be possible to diagonalise the tensor by taking its flow direction as eigenbasis and using it for a coordinate transformation.

2.4. Turbulence modelling

Turbulence was left aside until now, but it is necessary to discuss it and its place in the overall model. In section 2.1 the equations for single-phase flow were given. What was not mentioned was that despite being correct, they are not tractable as **they require resolving all scales of motion** (i.e., turbulent flow structures such as eddies and vortices) all the way to the Komolgorov scales. Looking at equations 2.2 and 1.13, this means that \mathbf{u} must be fully resolved; therefore we must understand what that implies, why it is not practical, and what are the practical alternatives.

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla \cdot (-\sigma) = \mathbf{S}_u \quad (2.2 \text{ revisited})$$

$$\frac{\partial \psi}{\partial t} + \nabla \cdot (\psi \otimes \mathbf{u}) + \nabla \cdot \mathbf{j}_{\text{dif}} = S \quad (1.13 \text{ revisited})$$

At the introduction, it was mentioned that \mathbf{u} is actually the advective flux, not velocity, because velocity is only well defined at points, not volumes. However, \mathbf{u} can be representative of velocity if the volume is small enough. This is exactly what it means to resolve velocity, and in order to resolve all the scales of turbulent motion, the cell volumes must be very small. Simulations with such resolution fall into the category of Direct Numerical Simulations (DNS), where domain size might be smaller than a cubic millimetre depending on the application — far from the size of a nuclear reactor!

An alternative that enables simulating fluid flows of practical importance is necessary, and that is the idea of turbulence models. **Turbulence models aim at modelling the effect of turbulent motion on flow properties, rather than resolving them.** There are multiple approaches with various degree of level of detail and completeness, but all of them finally adjust \mathbf{j}_{dif} in such a way that turbulent dissipation of quantities are accounted for.

Since the homogenised fluid dynamics model is essentially an extension of single-phase CFD, a first reasonable approach can be a simple extension of a standard single-phase turbulence models to account for homogenised regions. The standard k - ϵ model is the most widely used complete turbulence model [Pope, 2000, p. 373], and belongs to the category of turbulent viscosity models. As a relatively simple model that demonstrates completeness and wide applicability, it is an appropriate candidate for an extension to use with homogenised regions, before considering less applicable models or a category of turbulence models more elaborate than turbulent viscosity. The standard model is extended by Fiorina et al. [2015], and it is inherited by ATARI with slight modifications.

The overall idea of the k - ϵ model is to solve a continuity equation for turbulent kinetic energy k and for the rate of dissipation of turbulent kinetic energy ϵ . Due to the fluid-solid treatment, we are particularly interested in adjusting the k - ϵ model to account for a homogenised region, and have good estimates for k and ϵ at the outlet of the region. A possible approach is to force a convergence of k and ϵ to equilibrium values inside the homogenised region by modifying the typical conservation equations for turbulent kinetic energy and turbulent dissipation energy as shown in equations 2.46 and 2.47. We introduce a boolean variable $H = \{0, 1\}$, where the function takes the value of 1 in homogenised regions (i.e., regions with fluid and solid) and 0 in homogeneous regions (i.e., regions with only fluid). Using H , we selectively promote the convergence of k and ϵ to equilibrium values k_0 and ϵ_0 at rate λ in homogenised regions, or leave the usual sources and sinks of the model S_k and S_ϵ .

$$\frac{\partial \rho k}{\partial t} + \nabla \cdot (\rho k \mathbf{u}) + \nabla \cdot (\rho D_k (-\nabla k)) = H \rho \lambda (k_0 - k) + (1 - H) S_k \quad (2.46)$$

$$\frac{\partial \rho \epsilon}{\partial t} + \nabla \cdot (\rho \epsilon \mathbf{u}) + \nabla \cdot (\rho D_\epsilon (-\nabla \epsilon)) = H \rho \lambda (\epsilon_0 - \epsilon) + (1 - H) S_\epsilon \quad (2.47)$$

$$D_k = \frac{\nu_t}{\sigma_k} + \nu_l$$

$$D_\epsilon = \frac{\nu_t}{\sigma_\epsilon} + \nu_l$$

The resulting values of k and ϵ are used to calculate the turbulent viscosity or eddy viscosity ν_t through equation 2.49, where C_μ is a model constant (and defaults to 0.09). Mind that the eddy viscosity is an artificial **kinematic** viscosity (i.e., momentum diffusivity) that models the effect of turbulence on the diffusion of momentum.

$$\nu_t = C_\mu \frac{k^2}{\epsilon} \quad (2.49)$$

Table 2.6.: Parameters of the k- ϵ model in homogenised regions.

Parameter	Units	Meaning
ρ	kg m^{-3}	Density
H	-	Boolean variable for switching behaviour
λ	s^{-1}	Equilibrium convergence rate
k_0	$\text{m}^2 \text{s}^{-2}$	Equilibrium turbulent kinetic energy in homogenized zones
ϵ_0	$\text{m}^2 \text{s}^{-3}$	Equilibrium turbulent kinetic energy dissipation rate in homogenized zones
D_k	$\text{m}^2 \text{s}^{-8}$	Diffusivity of turbulent kinetic energy
σ_k	-	Model constant (defaults to 1.0)
D_ϵ	$\text{m}^2 \text{s}^{-8}$	Diffusivity of turbulent kinetic energy dissipation rate
σ_ϵ	-	Model constant (defaults to 1.3)
ν_t	$\text{m}^2 \text{s}^{-1}$	Eddy viscosity
ν_l	$\text{m}^2 \text{s}^{-1}$	Kinematic viscosity (i.e., Momentum diffusivity)
S_k	$\text{m}^2 \text{s}^{-3}$	Typical sources/sinks of k
S_ϵ	$\text{m}^2 \text{s}^{-4}$	Typical sources/sinks of ϵ
k	$\text{m}^2 \text{s}^{-2}$	Turbulent kinetic energy
ϵ	$\text{m}^2 \text{s}^{-3}$	Turbulent kinetic energy dissipation rate

The eddy viscosity is used to find a turbulent dynamic viscosity μ_t through equation 2.50, which is then used to find an effective dynamic viscosity through equation 2.51. The quantity μ_l appearing in equation 2.51 is the laminar dynamic viscosity, which is just the usual dynamic viscosity that is calculated from thermodynamic data only.

$$\mu_t = \rho \nu_t \quad (2.50)$$

$$\mu_{\text{eff}} = \mu_l + \mu_t \quad (2.51)$$

Finally, we would like to obtain the impact of turbulent motion on diffusion of thermal energy. This is done through equation 2.52, where Pr_t is typically 1.0 [Pope, 2000, p. 95]. Keep in mind that α_t is **not** a turbulent thermal diffusivity, because α_l is the same α discussed in section 2.1, which is not the thermal diffusivity; instead, it is an artifice to allow enthalpy to be solved implicitly. Both are used to find α_{eff} in equation 2.53, which is then used in a corrected heat flux shown in equation 2.54.

$$\alpha_t = \frac{\rho \nu_t}{\text{Pr}_t} \quad (2.52)$$

$$\alpha_{\text{eff}} = \alpha_l + \alpha_t \quad (2.53)$$

$$\mathbf{q} = -\alpha_{\text{eff}} \nabla h \quad (2.54)$$

From this point onwards, whenever the equations of fluid dynamics are shown it should be understood that μ and α implicitly mean μ_{eff} and α_{eff} . For cases where flow is laminar, a turbulence model is not used so μ_t and α_t are zero, and the effective value degenerates to the ones obtained from thermodynamic quantities only.

2.5. Solidification and melting

As a task in this project, a model for solidification and melting was introduced with the intent of evaluating the freezing of molten salts with high melting temperature. As usual in modelling, there were multiple ways to approach this, however, the so called enthalpy-porosity approach was chosen [Alexiades & Solomon, 1993; Swaminathan & Voller, 1992; Voller et al., 1987; Voller & Prakash, 1987]. This approach has advantages since it benefits from OpenFOAM's capability of dealing with arbitrary sources, and also numerical advantages related to the propagation of phase-change in the media.

In order to derive the relevant source, we start by rewriting equation A.15 for conservation of internal energy using enthalpy instead, making the substitution $h = e + \frac{p}{\rho}$. As mentioned in appendix A.2, this equation is incomplete, missing a contribution from mechanical energy to internal energy. Regardless of this imperfection, it will serve just fine in the end because the mechanical energy will not play a role in the derivation.

$$\frac{\partial \rho h}{\partial t} + \nabla \cdot (\rho h \mathbf{u}) + \nabla \cdot \mathbf{q} - \frac{\partial p}{\partial t} - \nabla \cdot (p \mathbf{u}) = S_{he} \quad (2.55)$$

Considering $h = h_s + L$, where h is the specific enthalpy h_s plus the phase change enthalpy L , also known as latent heat. We replace this into equation 2.55 to obtain:

$$\frac{\partial \rho(h_s + L)}{\partial t} + \nabla \cdot (\rho(h_s + L) \mathbf{u}) + \nabla \cdot \mathbf{q} - \frac{\partial p}{\partial t} - \nabla \cdot (p \mathbf{u}) = S_{he} \quad (2.56)$$

The term $\nabla \cdot \mathbf{q}$ was left untouched because phase change is not a diffusive phenomena. We continue by expanding the derivatives in preparation for algebraic manipulations.

$$\frac{\partial \rho h_s}{\partial t} + \frac{\partial \rho L}{\partial t} + \nabla \cdot (\rho h_s \mathbf{u}) + \nabla \cdot (\rho L \mathbf{u}) + \nabla \cdot \mathbf{q} - \frac{\partial p}{\partial t} - \nabla \cdot (p \mathbf{u}) = S_{he} \quad (2.57)$$

Passing the latent heat terms to right side as explicit terms, we end up in a form similar to the starting equation 2.55 with some additional terms related to phase change.

$$\frac{\partial \rho h_s}{\partial t} + \nabla \cdot (\rho h_s \mathbf{u}) + \nabla \cdot \mathbf{q} - \frac{\partial p}{\partial t} - \nabla \cdot (p \mathbf{u}) = S_{he} - \frac{\partial \rho L}{\partial t} - \nabla \cdot (\rho L \mathbf{u}) \quad (2.58)$$

It is possible to incorporate all terms that do not belong to the original equation into a source such that:

$$S_{he} = -\frac{\partial \rho L}{\partial t} - \nabla \cdot (\rho L \mathbf{u}) \quad (2.59)$$

Due to the derivation arriving back at the form of the starting conservation equation plus a source composed of some extra terms, this source is appropriate for any formulation of the energy equation that uses enthalpy to express thermal energy.

This model has the advantage of moving the enthalpy jump originating from latent heat to an explicit source and leave “sensible” enthalpy as a continuous quantity. This is illustrated by figure 2.1, of a conceptual 1-dimensional domain where some liquid solidifies at point s . The blue curve shows the continuous solution the method provides, and the red curve the discontinuous solution if latent heat were implicit.

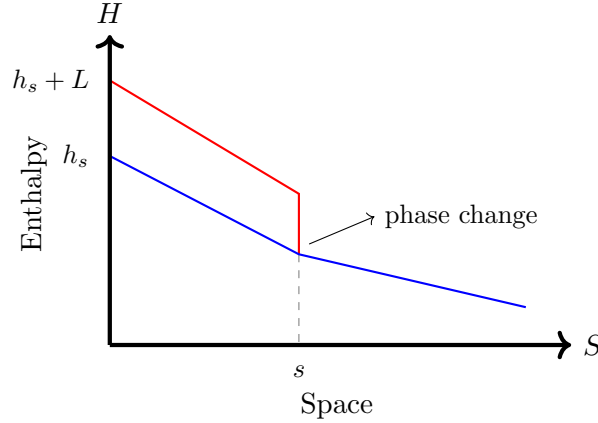


Figure 2.1.: Continuous enthalpy due to explicit latent heat

Despite this elegance, there is a need for an additional consideration. The source is mathematically correct, but as written it would require the solid-liquid interface location to be known with very high precision due to the enthalpy jump across the surface.

As discussed in section 2.2, resolving a sharp interface would be an option, although in this case an entirely different approach would be more appropriate. However, resolving the interface is not our intent in any case. It is not in line with the approach we seek of a homogenised model, neither is it practical for the analysis of large systems. Therefore,

we shall allow the interface to cross the cell without being explicitly resolved and adapt the source given by equation 2.59 to a viable numerical algorithm by introducing the concept of liquid fraction $\alpha[0,1]$ (not to be confused with thermal diffusivity), where:

$$\begin{cases} \alpha = 0 & \text{for solid} \\ 0 < \alpha < 1 & \text{for mushy} \\ \alpha = 1 & \text{for liquid} \end{cases} \quad (2.60)$$

A mushy region is a 2-phase region resulting as an artefact of the method chosen. Using this concept, we replace the latent heat L from the perfectly sharp concept for an approximate one $\Delta h(T)$, which represents the fraction α of the latent heat present in a cell, defined as:

$$\Delta h(T) = \alpha L, \text{ where } \Delta h \text{ is the enthalpy fraction} \quad (2.61)$$

Replacing L for αL to account for this numerical artefact, the source then becomes:

$$S_{he} = -\frac{\partial \rho \alpha L}{\partial t} - \nabla \cdot (\rho \alpha L \mathbf{u}) \quad (2.62)$$

This can be simplified to:

$$S_{he} = -L \left(\frac{\partial \rho \alpha}{\partial t} + \nabla \cdot (\rho \alpha \mathbf{u}) \right) \quad (2.63)$$

In ATARI, the advective term $\nabla \cdot (\rho \alpha \mathbf{u})$ is neglected. The fluid fraction α in each cell is calculated through iterations between equation 2.64 and the energy conservation equation supplemented by the source derived.

$$\alpha = {}^{-1}\alpha + R \frac{c_p}{L} (T - T_{\text{melt}}) \quad (2.64)$$

where ${}^{-1}\alpha$ is alpha at the previous iteration and R is a relaxation factor (usually 0.9). We must remember to forcefully bound α between 0 and 1.

The momentum equation must also be modified to account for the influence of phase change into the momentum. Essentially, any method that would eliminate momentum in the limit where $\alpha = 0$ would work. The method chosen was to add a porosity source

S_u to model this phenomenon.

$$S_u = -C_u \frac{(1 - \alpha)^2}{(\alpha + q)^3} \quad (2.65)$$

In this source, C_u is a constant that expresses how sensitive it is to a change of phase fraction. It is dependent, among other things on:

- geometry of the domain (e.g., small or large hydraulic diameter)
- surface phenomena (e.g., wetting, roughness)

Without empirical data or a model of C_u , the best that can be done is a parametric study on the sensitivity of the system to it.

In this model, enthalpy is transported using equation 2.41 with the solidification/melting source in order to find temperatures at cells. The temperature field is used to find the mushy zone (i.e., interfacial cells) through an iterative procedure using equation 2.64 to calculate the liquid fraction. After enough iterations, the domain and solution are found.

As concluded in the end of section 2.2, in order to work with the homogenised flow model presented, it is necessary to adjust the source so that fluid fraction is taken into account. We achieve this by noticing that according to equation 2.15, the enthalpy value for a phase has been scaled by the phase fraction, therefore, the source 2.63 must also be scaled by the same value as shown below.

$$S_{he} = -\gamma_f L \frac{\partial \rho \alpha}{\partial t} \quad (2.66)$$

In principle, equation 2.65 for linear momentum sink also needs to be scaled by the fluid fraction. However, since it is scaled by an **arbitrary** value C_u , it doesn't make any practical difference. The main purpose of the momentum sink is to terminate momentum in a frozen region, not slow it down by a precise measure.

2.6. Discussion

In this chapter, the main equations that are implemented in order to model homogenised fluid flow through a static structure were presented. While the model does present some additional considerations at the interfaces between fluid and percolated solid due to discontinuities, it also presents some advantages compared to an entirely heterogeneous

model. The model could be expanded to consider the percolating fluid as multiphase using the same foundation and a slightly more elaborate derivation.

The homogenised model fits well with the approach to neutron diffusion that will be shown in the next chapter, where a similar approach is taken to predict neutron population. One could reasonably argue that it makes little sense to have an entirely heterogeneous fluid flow model of extremely high accuracy, and pair it with a neutronics model that might present an accuracy bottleneck overall.

The topic of turbulence has been briefly touched in order to explain how ATARI treats it and why. An important concept to remember always is the advective flux, how it relates to velocity, and the considerations regarding its resolution and turbulence. The model shown and DNS are the two extremes of a range; one resolves all scales of eddies through advection, the other approximates the effects of eddies by tuning momentum diffusivity. An intermediate approach exists as well, called Large Eddy Simulation (LES), where large eddies are resolved but small eddies are filtered out and their effect accounted for by an increased diffusivity similar to the model presented.

Turbulence is vast topic, and its effects on modelling fluid flows is an important part of achieving an accurate simulation. While ATARI has no models besides the $k-\epsilon$ one inherited from GeN-Foam, expanding the selection will be relevant in the future.

The approach chosen for solidification and melting model is an established method, but has been modified to take into account the homogenised regions. This model has the advantage of keeping “sensible” enthalpy as a continuous quantity as far as phase change is concerned; a discontinuity caused by transition between zones of different fluid fraction is still possible.

Other approaches to model phase-change could be considered. For example, the “apparent heat capacity” model is another option, which gives heat capacity c_p as a function of temperature with peak centred at the melting temperature and the product of temperature and peak integral to be equal to the latent heat. It has the advantage of being readily usable, at least as long as c_p can be given as a function. However the method has downsides such as a strong reliance on discretisation since enthalpy will have a steep gradient (proportional to the sharpness of the peak). A sharp enough peak will result in a gradient that is numerically equivalent to a discontinuity.

Another approach would be to use an elaborate multiphase model with a full set of equations for solid. Such an approach would allow solids to move with the flow, but would require significant development and could prove **extremely** challenging depending on the level of detail. It is possible to imagine solids flowing, colliding and fragmenting for example. While solid-solid collisions could probably be neglected in a flow with small amount of solids, it would still be necessary to account for solid-wall collisions, which would be harder to neglect. With the exception of flows of low kinetic energy,

where fragmentation of colliding solids could be neglected, a fragmentation model would be necessary. Such a detailed model would also have an accuracy disproportional to the uncertainty of design and data used in this thesis — and probably most other studies for that matter.

3. Neutronics

... to devote its entire heart and mind to find the way by which the miraculous inventiveness of man shall not be dedicated to his death, but consecrated to his life.

*(Dwight D. Eisenhower
Atoms for Peace Speech)*

Neutronics, analogous in meaning to fluid or thermal mechanics but not as well defined, is the body of knowledge related to neutron behaviour in a system, be it in nuclear reactor, the topic of focus, or a neutron beam line.

The behaviour of neutrons inside a nuclear reactor is a vast, interesting and challenging topic. On average, neutrons are generated from fission events with energy of the order of MeV and are absorbed, possibly causing another fission, with energy orders of magnitude lower, and a lot happens in between these. Through lowering of energy due to collisions, the spatial region of influence of a neutron is reduced until it becomes relatively small compared to the dimensions of a reactor. While the multiplication factor itself is an integral system parameter that is not very sensitive to local phenomena, various safety parameters are local and sensitive to this subtle temporal balance of neutrons. Depending on the application and parameters of interest, modelling of neutron behaviour in a system requires accuracy in energy, direction, space and time.

In this chapter, we shall present and discuss how to model neutron behaviour and several assumptions and approximations that happen in the process. The intent is not to give an in depth derivation of everything on the topic. Background knowledge is assumed such as the concepts of multiplication factor, angular and scalar neutron flux, neutron spectrum, nuclear data (i.e., microscopic and macroscopic cross-section, resonance treatment, self-shielding and doppler broadening), condensation, homogenization, and burnup, which can be found in several good books [Hébert, 2020; Stacey, 2018; Cacuci, 2010]. The aim is to use some basic background to give the reader an understanding of ATARI's capabilities, subtle aspects of neutronics, and insights into the topic.

3.1. Neutron transport

In order to model neutron behaviour in a reactor, we are interested in neutron movement inside a medium composed of atoms and molecules where neutron and medium particles interact during collision events. This is done through the Boltzmann transport equation, where transport is taken as ballistic between interactions, and quantum properties are taken into account by the cross-sections during interaction resolution. This interaction can result in any number of nuclear reactions, such as scattering, capture, fission, etc.

In practice, we are interested in a particular form of the equation called the Linear Boltzmann Equation (LBE), shown in 3.1 already complemented with a delayed neutron source. This equation does not take into account multiple effects, among them: (1) neutron-neutron interactions (thus why it is a linear version), (2) bulk movement of the medium, and (3) relativistic motion. These are appropriate approximations for reactor modelling and would only become a problem in other applications such as: (1) modelling of a neutron star in astrophysics, (2) disassembly stage of a nuclear explosive, and (3) relativistic neutrons in high energy particle physics.

The delayed neutron source in equation 3.1 couples it with 3.2, which models the dynamics of delayed neutron precursors (DNP) concentration (i.e., fission products that emit additional neutrons from radioactive decay). A detailed derivation of both equations is readily available in the literature [Cacuci, 2010, Section 5.2], however equation 3.2 has been generalised to consider the effect of medium convection into the spatial distribution of precursors.

$$\begin{aligned}
& \frac{1}{V(E)} \frac{\partial}{\partial t} \psi(\mathbf{r}, \hat{\Omega}, E, t) + \hat{\Omega} \cdot \nabla \psi(\mathbf{r}, \hat{\Omega}, E, t) + \Sigma_t(\mathbf{r}, E, t) \psi(\mathbf{r}, \hat{\Omega}, E, t) \\
&= \int_0^\infty \int_{4\pi} \Sigma_s(\mathbf{r}, \hat{\Omega}' \cdot \hat{\Omega}, E' \rightarrow E, t) \psi(\mathbf{r}, \hat{\Omega}', E', t) d\hat{\Omega}' dE' \\
&+ \frac{1}{k} \frac{\chi_p(\mathbf{r}, E, t)}{4\pi} \int_0^\infty \int_{4\pi} (1 - \beta(\mathbf{r}, E', t)) \bar{\nu}(\mathbf{r}, E', t) \Sigma_f(\mathbf{r}, E', t) \psi(\mathbf{r}, \hat{\Omega}', E', t) d\hat{\Omega}' dE' \\
&+ \frac{1}{4\pi} \sum_{g=1}^G \chi_{d,g}(\mathbf{r}, E, t) \lambda_g C_g(\mathbf{r}, t)
\end{aligned} \tag{3.1}$$

$$\begin{aligned}
& \frac{\partial}{\partial t} C_g(\mathbf{r}, t) + \nabla \cdot (C_g(\mathbf{r}, t) \mathbf{u}) + \nabla \cdot (-D_g \nabla C_g(\mathbf{r}, t)) \\
&= \frac{1}{k} \int_0^\infty \int_{4\pi} \beta_g(\mathbf{r}, E', t) \bar{\nu}(\mathbf{r}, E', t) \Sigma_f(\mathbf{r}, E', t) \psi(\mathbf{r}, \hat{\Omega}', E', t) d\hat{\Omega}' dE' \\
&- \lambda_g C_g(\mathbf{r}, t)
\end{aligned} \tag{3.2}$$

The parameters involved are shown on table 3.1. The first block of parameters lists the 7 dimensions of the problem: 3 dimensions of space represented by position vector \mathbf{r} , 2 dimensions of direction-of-flight in unit vector $\hat{\Omega}$, neutron energy E , and time t . The second block lists parameters that are given as data. The third and final block shows unknown quantities to be determined: k , ψ and C .

Table 3.1.: Parameters of the neutron transport equation.

Parameter	Units	Meaning
\mathbf{r}	m	Position vector
$\hat{\Omega}$	-	Direction-of-flight unit vector ($ \hat{\Omega} = 1$)
E	eV	Neutron energy
t	s	Time
$V(E)$	m s^{-1}	Neutron speed
$\Sigma_{(t,f)}(\mathbf{r}, E, t)$	m^{-1}	Macroscopic cross-section for total or fission reactions
$\Sigma_s(\mathbf{r}, \hat{\Omega}' \cdot \hat{\Omega}, E' \rightarrow E, t)$	m^{-1}	Macroscopic cross-section for scattering reactions
$\bar{\nu}(\mathbf{r}, E', t)$	-	Average number of neutrons produced in fission reactions
$\chi_p(\mathbf{r}, E, t)$	-	Energy spectrum for emission of prompt neutron
$\beta(\mathbf{r}, E', t)$	-	Fraction of Delayed neutrons
$\chi_{d,g}(\mathbf{r}, E, t)$	-	Energy spectrum for emission of delayed neutron from group g
$D_g(\mathbf{r}, t)$	-	Diffusion coefficient of delayed neutron precursors in group g
$\beta_g(\mathbf{r}, E', t)$	-	Fraction of Delayed neutrons from group g
λ_g	s^{-1}	Decay constant of delayed neutron precursors from group g
k	-	Multiplication factor
$\psi(\mathbf{r}, \hat{\Omega}, E, t)$	$\text{m}^{-2}\text{s}^{-1}\text{sr}^{-1}\text{eV}^{-1}$	Angular flux
$C_g(\mathbf{r}, t)$	m^{-3}	Concentration of Delayed neutron precursors in group g

Usually, equation 3.1 is shown either in its steady-state formulation, without time derivative $\frac{1}{V(E)} \frac{\partial}{\partial t} \psi(\mathbf{r}, \hat{\Omega}, E, t)$, or in its transient formulation, without eigenvalue k . Despite the usual presentation, both terms are left in the equation since that is what the most general problem requires, however, the terms are mutually exclusive as free

parameters - it is not possible for both to vary at the same time. A pseudo-steady-state simulation must be performed first to find the eigenvalue, then a transient simulation can be performed where the eigenvalue found previously is kept fixed.

The diffusion coefficient D_g in equation 3.2, which describes the diffusion of delayed neutron precursors in the system, is obtained from equation 3.3 in a similar way as α_{eff} in equation 2.53. The diffusion coefficient is separated into a laminar and a turbulent component, where the later corrects the diffusion constant by taking the turbulent motion into account. In equation 3.3, Sc is the Schmidt number, which relates the momentum diffusivity and the mass diffusivity. In most cases of interest, at least in this thesis, the diffusion of mass during laminar flow is negligible, leading to a very high Schmidt number. However, the turbulent effect can be reasonably approximated to be similar to the one for the diffusion of energy, leading to a turbulent Schmidt number of the order of unity.

$$D_g = D_l + D_t = \frac{\nu}{Sc} + \frac{\nu_t}{Sc_t} \quad (3.3)$$

Together, equations 3.1 and 3.2 represent a general model of the problem. Scattering is expressed with arbitrary anisotropy, and each delayed neutron precursor is a single isotope with its own neutron emission energy spectrum. The multiple methods used to solve the neutron transport equation are discussed in the literature with detail [Sanchez & McCormick, 1982]. The computational power required to solve a large scale problem with such dimensionality is very high, therefore it should be used sparingly.

Recognising existing computational limitations, the typical approach to reactor analysis breaks down the problem in different scales, such as shown in figure 3.1, to achieve an approximate solution that is reasonably accurate. Different approaches exist to achieve this objective, but the shared concept is the same: (1) solve one or more small scale problems with very detailed energy and anisotropy representation, and (2) use the results to condense the energy spectrum and homogenise spatial heterogeneities, generating a new approximate set of nuclear data for use in larger scale problems.

In the context of light water reactors (LWR), the method of Collision Probability is typically used in a cylindrical square-equivalent pin-cell in order to condense energy groups (from hundreds to tens of groups). Then the method of characteristics is used on a lattice of pins to further reduce the number of energy groups (from tens to typically 2 to 8) and homogenise in space. The resulting condensed homogenised cross-sections can be used for full reactor analysis, which typically uses either a diffusion approximation or simplified spherical harmonics SP_N .

Unlike an LWR, the MSFR is a homogeneous reactor; therefore, a simpler strategy is used. Condensation from hundreds to few energy groups and homogenization can

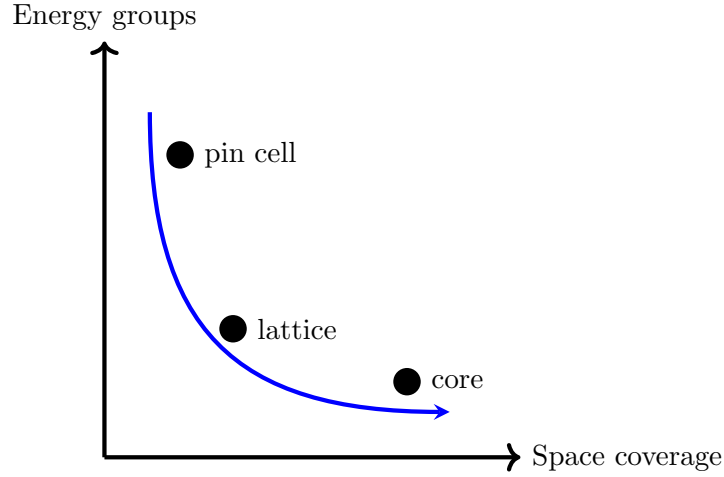


Figure 3.1.: Multiscale reactor analysis

happen in a single step. In addition, the neutron diffusion approximation is valid without caveats as we will discuss.

3.2. The diffusion approximation

Large scale problems (i.e., reactor size) commonly use Fick's law, equation 1.12a, in order to reduce the transport problem of equation 3.1 into a diffusion one as shown in equation 3.4. As can be seen, dependency on $\hat{\Omega}$ has been eliminated and with it most of the anisotropy description. In this process, the corpuscular nature of neutrons is completely lost; its behaviour approximated to a formless quantity, analogous to heat, that simply permeates the medium. It is possible to use techniques to generate transport corrected cross-sections, as it is also possible to use the same technique in certain isotropic formulations of the transport equation (e.g., method of characteristics). It is also possible to make $D(\mathbf{r}, E, t)$ a tensor, instead of a scalar. However, no matter what technique is used, the anisotropy expressiveness of the neutron transport equation has been irrecoverably lost.

The diffusion approximation requires some strong conditions to give accurate results. More specifically, the medium must be diffusive, i.e., $\Sigma_s \gg \Sigma_a$, and it must lack strong localised absorbers and streaming effects.

A typical reactor contains a highly heterogeneous lattices, with control rods, burnable poison, and other similar material that result in a localised large gradient of neutron flux. For the diffusion approximation to be valid in such reactors, these heterogeneities must be smoothened through spatial homogenisation in small scale simulations. As a side effect in these reactors, given a typical assembly square lattice of 17×17 pins for

example, there is a “sweet spot” of how much space should be homogenised (e.g., whole assembly, 4 subassemblies, 9 subassemblies, etc) in order to obtain accurate results through the diffusion approximation.

For homogeneous reactors, such as the MSFR studied in later chapters, the condition for diffusion is easily satisfied, and smaller homogenised volumes do not result in invalidating the method. However, care must still be taken at the interface between core and blanket if accurate results are important in this location.

Some additional approximations have been made regarding delayed neutron precursors. The original problem posed by equation 3.1 divided delayed neutron precursors in such a detail that it would be possible to model each precursor isotope separately, which is too general. In equation 3.4, isotopes with similar decay dynamics have been lumped together into delayed neutron precursor groups. The delayed neutron energy spectrum $\chi_d(\mathbf{r}, E, t)$ has been brought out of the summation, implying that instead of each $C_g(\mathbf{r}, t)$ having its own spectrum upon decay, all delayed neutron precursors share the same emission spectrum. $\beta(\mathbf{r}, t)$ and $\beta_g(\mathbf{r}, t)$ have been integrated and brought out of the fission source integral, therefore, fissions of any energy contributes equally to the production of a certain $C_g(\mathbf{r}, t)$ and all precursor production, regardless of energy, must be subtracted from the fission source.

$$\begin{aligned}
& \frac{1}{V(E)} \frac{\partial}{\partial t} \phi(\mathbf{r}, E, t) + \nabla \cdot (-D(\mathbf{r}, E, t) \nabla \phi(\mathbf{r}, E, t)) + \Sigma_t(\mathbf{r}, E, t) \phi(\mathbf{r}, E, t) \\
&= \int_0^\infty \Sigma_s(\mathbf{r}, E' \rightarrow E, t) \phi(\mathbf{r}, E', t) dE' \\
&+ \chi_p(\mathbf{r}, E, t) (1 - \beta(\mathbf{r}, t)) \frac{\int_0^\infty \bar{\nu}(\mathbf{r}, E', t) \Sigma_f(\mathbf{r}, E', t) \phi(\mathbf{r}, E', t) dE'}{k} \\
&+ \chi_d(\mathbf{r}, E, t) \sum \lambda_g C_g(\mathbf{r}, t)
\end{aligned} \tag{3.4}$$

$$\begin{aligned}
& \frac{\partial}{\partial t} C_g(\mathbf{r}, t) + \nabla \cdot (C_g(\mathbf{r}, t) \mathbf{u}) + \nabla \cdot (-D_g \nabla C_g(\mathbf{r}, t)) \\
&= \beta_g(\mathbf{r}, t) \frac{\int_0^\infty \bar{\nu}(\mathbf{r}, E', t) \Sigma_f(\mathbf{r}, E', t) \phi(\mathbf{r}, E', t) dE'}{k} \\
&- \lambda_g C_g(\mathbf{r}, t)
\end{aligned} \tag{3.5}$$

After condensation and homogenisation, the cross-sections Σ in equations 3.1 and 3.4 are not the same. During condensation, information regarding flux distribution has been used. In a similar manner, during homogenisation, information regarding geometry has been used. Therefore, while the cross-sections in equation 3.1 are, in principle, fairly general and require only isotope concentration and some mild approximations in nuclear data, the cross-sections in equation 3.4 are specific to the geometrical location

where homogenisation was performed, and not only a function of isotope concentration anymore. For this reason, changing the cross-section value due to a change in the system requires some sort of parametrization, discussed in section 3.3.

In order to solve equations 3.4 and 3.5, it is necessary to discretise its dimensions. Spatial discretisation is a functionality of multiple frameworks to solve partial differential equations (OpenFOAM in the case of ATARI), therefore, we shall seek only a discretisation in energy in equations 3.6 and 3.7.

$$\begin{aligned}
 & \frac{1}{V_i} \frac{\partial}{\partial t} \phi_i(\mathbf{r}, t) + \nabla \cdot (-D_i(\mathbf{r}, t) \nabla \phi_i(\mathbf{r}, t)) + \Sigma_{t,i}(\mathbf{r}, t) \phi_i(\mathbf{r}, t) \\
 &= \sum_{j=1}^J \Sigma_{s,i \leftarrow j}(\mathbf{r}, t) \phi_j(\mathbf{r}, t) \\
 &+ \chi_{p,i}(\mathbf{r}, t) (1 - \beta(\mathbf{r}, t)) \frac{\sum_{j=1}^J \bar{\nu}_j(\mathbf{r}, t) \Sigma_{f,j}(\mathbf{r}, t) \phi_j(\mathbf{r}, t)}{k} \\
 &+ \chi_{d,i}(\mathbf{r}, t) \sum_{g=1}^G \lambda_g C_g(\mathbf{r}, t)
 \end{aligned} \tag{3.6}$$

$$\begin{aligned}
 & \frac{\partial}{\partial t} C_g(\mathbf{r}, t) + \nabla \cdot (C_g(\mathbf{r}, t) \mathbf{u}) + \nabla \cdot (-D_g(\mathbf{r}, t) \nabla C_g(\mathbf{r}, t)) \\
 &= \beta_g(\mathbf{r}, t) \frac{\sum_{j=1}^J \bar{\nu}_j(\mathbf{r}, t) \Sigma_{f,j}(\mathbf{r}, t) \phi_j(\mathbf{r}, t)}{k} \\
 &- \lambda_g C_g(\mathbf{r}, t)
 \end{aligned} \tag{3.7}$$

Equation 3.6 is typically modified by subtracting the diagonal terms of the scattering source from the term containing the total cross-section, leading to what is called “removal cross-section”, shown in equation 3.8. This leads to a slightly modified equation 3.9, where the scattering term does not contain terms where $i = j$, expressing scattering inside an energy group, leaving only scattering that leads to a change of energy group.

$$\Sigma_{\text{rem},i}(\mathbf{r}, t) = \Sigma_{t,i}(\mathbf{r}, t) - \Sigma_{s,i=j}(\mathbf{r}, t) \tag{3.8}$$

$$\begin{aligned}
& \frac{1}{V_i} \frac{\partial}{\partial t} \phi_i(\mathbf{r}, t) + \nabla \cdot (-D_i(\mathbf{r}, t) \nabla \phi_i(\mathbf{r}, t)) + \Sigma_{\text{rem}, i}(\mathbf{r}, t) \phi_i(\mathbf{r}, t) \\
&= \sum_{i \neq j} \Sigma_{s, i \leftarrow j}(\mathbf{r}, t) \phi_j(\mathbf{r}, t) \\
&+ \chi_{p, i}(\mathbf{r}, t) (1 - \beta(\mathbf{r}, t)) \frac{\sum_{j=1}^J \bar{\nu}_j(\mathbf{r}, t) \Sigma_{f, j}(\mathbf{r}, t) \phi_j(\mathbf{r}, t)}{k} \\
&+ \chi_{d, i}(\mathbf{r}, t) \sum_{g=1}^G \lambda_g C_g(\mathbf{r}, t)
\end{aligned} \tag{3.9}$$

Finally, these equations must be coupled with the homogenised equations of fluid dynamics from chapter 2. Remember that those equations do not differentiate in space between moving fluid and static solid. This concept is shared by $\phi_i(\mathbf{r}, t)$ and $C_g(\mathbf{r}, t)$, whose quantities are homogenised in space already, or per total volume, similarly to $\mathbf{u}_{h, f}$. However, convection of $C_g(\mathbf{r}, t)$ is confined to its concentration in fluids only. The advection term of equation 3.7 is already correct as long as we keep in mind that $\mathbf{u}_f = \frac{\mathbf{u}_{h, f}}{\gamma_f}$. The diffusion term needs correction to find the concentration of $C_g(\mathbf{r}, t)$ that is representative of its diffusion, as shown in equation 3.10, but changing $C_g(\mathbf{r}, t)$ is inconvenient since it's a quantity that is already continuous. Therefore, we maintain it and change the diffusion constant instead.

$$\begin{aligned}
& \frac{\partial}{\partial t} C_g(\mathbf{r}, t) + \nabla \cdot (C_g(\mathbf{r}, t) \mathbf{u}_f) + \nabla \cdot \left(-\frac{D_g(\mathbf{r}, t)}{\gamma_f} \nabla C_g(\mathbf{r}, t) \right) \\
&= \beta_g(\mathbf{r}, t) \frac{\sum_{j=1}^J \bar{\nu}_j(\mathbf{r}, t) \Sigma_{f, j}(\mathbf{r}, t) \phi_j(\mathbf{r}, t)}{k} \\
&- \lambda_g C_g(\mathbf{r}, t)
\end{aligned} \tag{3.10}$$

3.3. Parametrization

As mentioned in the previous section, once condensation and homogenisation are performed, a naive variation of Σ becomes impossible. A common way is to parametrise few groups homogenised cross-sections as shown in figure 3.2. In this figure, the value of a cross-section Σ varies as a function of a state variable S . What exactly a state variable is will depend on the reactor. Some state variables are common to many reactors, such as coolant density, coolant and fuel temperatures, and burnup, where burnup requires special consideration that will be mentioned later. However, some state variables are specific to a particular reactor design, such as control rod insertion, expansion or contraction of some structural material or concentration of some strong absorber such as Samarium or Xenon. In more general terms, a state variable is a

parameter that upon a change of value results in a system response (i.e., change of reactor state) that cannot be ignored to achieve the required modelling accuracy. In other words, a state variable is a parameter to which the system response is sensitive to. Determining state variables requires at least an accuracy criteria and some engineering sense, since high sensitivity parameters tend to be obvious, but might also require a sensitivity study for lower sensitivity ones.

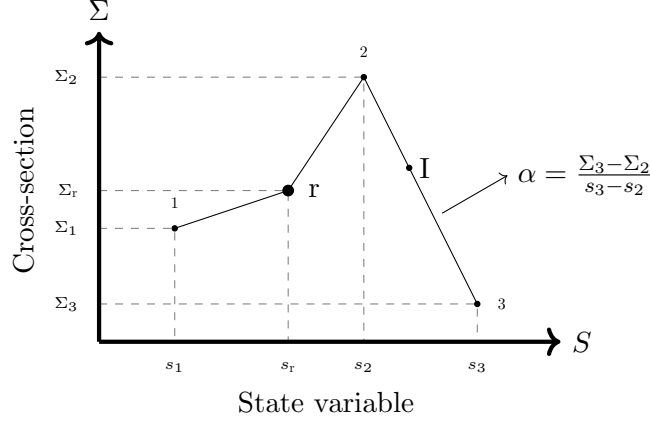


Figure 3.2.: Parametrization of cross-sections from a reference point r

In figure 3.2, the homogenised cross-section values have been calculated for 4 values of a state variable, a reference one and 3 others in the range of interest. The reference value is usually the expected value during nominal operation of the reactor. The additional values and the range of interest are determined by how much this parameter is expected to vary during a simulation and how many points are necessary to capture features of importance such as peaks, gradients and inflections.

We would like to be able to determine the unknown cross-section at a point of interest Σ_I , from a given value of the state variable s_I , the known data points, and some interpolation assumption. This is typically implemented as the sum of a reference cross-section Σ_r at state s_r plus some correction term Σ' as shown in equation 3.11.

$$\Sigma_I = \Sigma_r + \Sigma' \quad (3.11)$$

Let us forget for a moment about the reference cross-section and write an equation that would work to determine Σ_I from the closest known cross-section value whose state value is lower than s_r . In the case of figure 3.2, this is represented as equation 3.12

$$\Sigma_I = \Sigma_2 + \alpha(s_I - s_2) \quad (3.12)$$

where α is the slope shown in figure 3.2. By adding and subtracting the term Σ_r and

rearranging them, we recover the form desired by equation 3.11.

$$\Sigma_I = \Sigma_2 + \alpha(s_I - s_2) + \Sigma_r - \Sigma_r \implies \Sigma_I = \Sigma_r + (\Sigma_2 - \Sigma_r) + \alpha(s_I - s_2) \quad (3.13)$$

We now know what the correction term Σ' is. As shown in equation 3.14, it is composed of an offset and a slope times difference in state variables. The slope and the offset can either be stored as an interpolation table data file or precomputed at program start from point data.

$$\Sigma' = (\Sigma_2 - \Sigma_r) + \alpha(s_I - s_2) \quad (3.14)$$

In this form, interpolation was assumed to be linear for the state variable, which is always an option as long as distance between points is small enough. By changing the state axis of S from a linear scale to square root or logarithmic scale, the interpolation can be generalised. This results in the following change for logarithmic:

$$\Sigma' = (\Sigma_2 - \Sigma_r) + \alpha(\ln s_I - \ln s_2) \quad (3.15)$$

$$\text{where } \alpha = \frac{\Sigma_3 - \Sigma_2}{\ln s_3 - \ln s_2} \quad (3.16)$$

Likewise for square root:

$$\Sigma' = (\Sigma_2 - \Sigma_r) + \alpha(\sqrt{s_I} - \sqrt{s_2}) \quad (3.17)$$

$$\text{where } \alpha = \frac{\Sigma_3 - \Sigma_2}{\sqrt{s_3} - \sqrt{s_2}} \quad (3.18)$$

Here, it was shown how to calculate Σ_I as a function of a single state variable. Normally, there will be more state variables of interest. This requires the method to be extended to n state variables by adding more correction terms in the equation, such that:

$$\Sigma_I = \Sigma_r + \Sigma'_1 + \dots + \Sigma'_n \quad (3.19)$$

where each correction term is calculated by an offset from the reference value and a slope in the same way as the simple case derived. An important assumption of this method, at least as presented, is that the responses of Σ_I to perturbations of state variables must be independent of one another. In principle, a more general method could be used with

higher order perturbations in order to relax this assumption, however in most cases this first order approach is suitable.

The parametrization scheme presented is usually called “Multidimensional Interpolation Table”. It has as many dimensions as n , the interpolation for each dimension can have different forms (i.e., linear, logarithmic or square root) and the offsets and slopes are stored as a data table.

However, burnup is a special state variable for 2 unique characteristics: (1) it is an overarching state variable that influences the value at the reference point and features of interest of all other state variables (i.e., peaks, gradients and inflections might change) and, (2) unlike common state variables that might change quickly during a transient state, burnup progresses slowly during reactor operation. Therefore, burnup can be assumed as static for a calculation at any combination of common state variables and parametrisations for state variables are calculated at each burnup state¹. The result of such calculations is a series of figures similar to 3.2, where peaks, gradients and inflections change location or value. Therefore, parametrization results in 1-dimensional curve for a certain burnup state, and branching stacks multiple of these figures in a burnup axis, resulting in a 2-dimensional surface for each state variable. To perform simulations for an arbitrary burnup, the value of Σ_I can be determined as a burnup-weighted mean from 2 bounding burnups, or hypothetical values for references, offsets and slopes can be pre-interpolated.

3.4. Discussion

An overview of the approaches to model neutron behaviour in a nuclear reactor has been presented, with a focus on assumptions. Only deterministic methods were mentioned, with a noticeable lack of a discussion regarding Monte Carlo methods.

Today, the Monte Carlo method is also commonly used for neutron transport problems, which avoids solving the neutron transport equation and instead simulates neutron behaviour through stochastic means and “brute force” computational power. This method excels in situations where the analyst is ignorant about the reactor characteristics, or in non-routine analysis such as design of experiments, where fewer assumptions in resonance treatment and energy group structure in the nuclear data are beneficial.

Irradiation experiments are relatively expensive, therefore the cost justifies the expense of using the Monte Carlo simulations for its design. Frequent changes in experimental conditions might also prevent establishing appropriate assumptions that would allow

¹Parametrization for burnup is tightly related to evolution of isotopic composition, which varies slowly. However, consideration might be given to particular isotopes, such as ¹³⁵Xe and ¹⁴⁹Sm, that might receive their own special treatment.

design of experiments to meet a desired uncertainty criteria.

In cases where routine analysis is the target, deterministic methods excel due to their superior speed. The downside is that certain assumptions are necessary to make the problem tractable, which requires a good understanding of the system. The analysis of power reactors, and many research reactors as well, falls into this category. Therefore it is important to understand the system well enough to allow a deterministic analytical methodology to be developed.

While one may argue about the benefits of different neutron transport methods on small scale simulations — and there is a lot to be considered — codes using the diffusion approximation are the workhorse of large scale full-core reactor analysis still today. Even in reactors with heterogeneous lattices, the diffusion approximation can be made to be valid with appropriate spatial treatment. There is a limit to how far heterogeneities can be spatially homogenised and yet retain an accurate solution, but within the design boundaries of commercial reactors it has proven to be a very powerful and computationally cheap method.

The diffusion approximation also pairs well with the homogenised fluid-solid model developed in the previous chapter, which follows a similar paradigm. For a heterogeneous reactor core using fuel pin lattices, it would be necessary to reconstruct the power profile at below mesh resolution and an additional pin model in the fluid dynamics solver to benefit from such information.

For homogeneous reactors such as the MSFR, it is difficult to justify other methods of reactor analysis, except in special circumstances. No pins are present, and no additional model to predict phenomena below mesh resolution in the core is required either. Homogeneous reactors create the best conditions for the diffusion approximation to be valid by design [Tiberga, Lathouwers, & Kloosterman, 2020].

4. Algorithms

An algorithm must be seen to be believed.

(Donald Knuth)

An algorithm is a series of operations, transformations, evaluations and sub-algorithms that are performed over a certain input, leading to a result. In other words, algorithms are well defined processes. Computer programs and problem solving in mathematics are algorithms as well by this definition. A solver, in this thesis, is considered to be a piece of algorithm that provides a solution to a well defined mathematical problem. Therefore, a solver may provide a solution to a neutronics problem, to a fluid dynamics one, or both if coupled.

ATARI inherits a significant number of mathematical algorithms from the OpenFOAM library, implements many new ones and ultimately, assembles everything into 2 main algorithms/programs: the single-mesh solver (SMS) and the multi-mesh solver (MMS). Visualizations of the algorithms as a flow diagram are provided to help understanding their inner workings.

In this chapter, we shall first explore the single-mesh solver, which uses all equations previously presented in order to set up a simple coupled neutronics and fluid dynamics solver. This solver is such a complete implementation of the concept that very few addendums have to be made in order to extend its logic to the MMS introduced later on, which extends the scope to problems with multiple regions.

By the end, the reader should have a good grasp of the inner logic behind the main solvers in ATARI, their niche of application, limitations, and possible expansions.

Notation: implicit and explicit terms

Equation 4.1 represents a linear system, which might represent a partial differential equation discretised in space and/or time. Terms that are part of matrix A are called implicit terms. Terms that are either known or unknown but calculated with fields from previous iterations or time steps and simply added to the source b are called explicit.

$$Ax = b \quad (4.1)$$

For clarity, any partial differential equation that is discretised and solved for will be written as follows:

- implicit terms on the left hand side (LHS) of the equality
- explicit terms on the right hand side (RHS) of the equality

4.1. Single-mesh solver

This solver is a simplified approach to tackle a coupled neutronics and fluid dynamics problem. It uses a single mesh in both the neutronics and fluid dynamics sub-solvers in order to do so, therefore, it is limited in scope to the primary circuit of the reactor, which is a well defined boundary problem. Despite this disadvantage, it is also significantly easier to set up than a more elaborate one using multiple meshes, making it a good option to prototype complicated cases.

A single mesh is usually not appropriate for such a coupled simulation, since the fuel and active core of typical reactors is limited to a portion of the primary circuit. Despite this, using a single mesh at this stage has advantages, as it is easier to explain and later expand to a problem that deals with multiple meshes. In addition, since MSRs are an important topic in this thesis, and these reactors circulate a fissile fluid throughout the entire primary circuit, the convection of delayed neutron precursors makes it unavoidable that the fluid dynamics and neutronics mesh would have identical domains.

If the only difference between 2 completely overlapping meshes to model such a reactor would be the cell size, there is only a marginal gain in having an individual mesh for each. Therefore, this solver is quite suited for introduction of the solver algorithm and also for practical MSR analysis.

Otherwise, typical requirements of mesh determine stability and accuracy of the simulation. Stability considerations are:

- Velocity is part of the Courant number or Courant–Friedrichs–Lewy (CFL) condition, which defines numerical stability in fluid dynamics.

- The neutron mean free path influences the diffusion coefficient, which is part of a Diffusion number that defines stability in neutronics using diffusion models.
- The quantities mentioned (i.e., diffusion coefficient and velocity) interact with cell size (mesh) and time step to define the respective stability conditions.

In addition to these stability conditions, relaxation factors also increase instability in exchange for reducing the rate of convergence. Stability is the minimal condition to perform a numerical simulation. Once stability is satisfied, accuracy becomes the next priority criterion, taking the following into consideration:

- Resolution of gradient of quantities will limit cell size required to obtain an accurate solution
- For fluid dynamics, the gradient of velocity or temperature are criteria for choosing mesh, especially close to the walls
- Close to the walls, the gradient of transported quantities leads to the idea of “wall functions”, whose purpose is to reduce the need for a very fine mesh close to the wall to model this gradient by assuming a certain shape function — an idea that is conceptually similar to the use of shape functions by Finite Element Method (FEM)
- Gradients of neutron flux might or might not be a problem close to the wall, depending on the neutronic properties of the system.

Numerical schemes interact with stability and accuracy, where higher order numerical schemes improve accuracy for a given mesh while sacrificing stability. It is usually recommended to start a simulation using lower order numerical schemes that offer higher stability. Simulations can always be restarted in higher numerical schemes later, once the fields are well converged.

Something else to consider is resolving the geometry, which is not always necessary to capture the phenomena of interest. The analyst should always consider the possibility and degree of geometry defeaturing (i.e., selective loss of detail) during meshing procedure to reduce cell count.

4.1.1. Implemented equations and flow diagram

The governing equations necessary for a coupled simulation of fluid dynamics (2.39-2.41 and 2.30) and neutronics (3.9 and 3.10) have been presented in the previous chapters. In order to implement them, it is necessary to define what are the implicit and explicit terms, how explicit terms are calculated and the coupling mechanism.

$$\frac{\partial \gamma_f \rho_f}{\partial t} + \nabla \cdot (\rho_f \mathbf{u}_{h,f}) = 0 \quad (2.39 \text{ revisited})$$

$$\frac{\partial \rho_f \mathbf{u}_{h,f}}{\partial t} + \nabla \cdot \left(\frac{\rho_f \mathbf{u}_{h,f}}{\gamma_f} \otimes \mathbf{u}_{h,f} \right) - \mu \nabla^2 \mathbf{u}_{h,f} = -\gamma_f \nabla p + \gamma_f \rho_f g + F_{\text{drag},i} + \mathbf{S}_u \quad (2.40 \text{ revisited})$$

$$\begin{aligned} & \frac{\partial \gamma_f \rho_f h_f}{\partial t} - \frac{\partial p}{\partial t} + \nabla \cdot (\rho_f h_f \mathbf{u}_{h,f}) + \nabla \cdot (-\gamma_f \alpha \nabla h) \\ & + \frac{\partial \gamma_f \rho K}{\partial t} + \nabla \cdot (\rho K \mathbf{u}_{h,f}) = \rho g \cdot \mathbf{u}_{h,f} - Q_i + S_{he} \end{aligned} \quad (2.41 \text{ revisited})$$

$$\frac{\partial \gamma_s \rho_s c_s T_s}{\partial t} - \nabla \cdot (\gamma_s k_s \nabla T_s) = Q_i + S_s \quad (2.30 \text{ revisited})$$

$$\begin{aligned} & \frac{1}{V_i} \frac{\partial}{\partial t} \phi_i(\mathbf{r}, t) + \nabla \cdot (-D_i(\mathbf{r}, t) \nabla \phi_i(\mathbf{r}, t)) + \Sigma_{\text{rem},i}(\mathbf{r}, t) \phi_i(\mathbf{r}, t) \\ & = \sum_{i \neq j} \Sigma_{s,i \leftarrow j}(\mathbf{r}, t) \phi_j(\mathbf{r}, t) \\ & + \chi_{p,i}(\mathbf{r}, t) (1 - \beta(\mathbf{r}, t)) \frac{\sum_{j=1}^J \bar{\nu}_j(\mathbf{r}, t) \Sigma_{f,j}(\mathbf{r}, t) \phi_j(\mathbf{r}, t)}{k} \\ & + \chi_{d,i}(\mathbf{r}, t) \sum_{g=1}^G \lambda_g C_g(\mathbf{r}, t) \\ & \frac{\partial}{\partial t} C_g(\mathbf{r}, t) + \nabla \cdot (C_g(\mathbf{r}, t) \mathbf{u}_f) + \nabla \cdot \left(-\frac{D_g(\mathbf{r}, t)}{\gamma_f} \nabla C_g(\mathbf{r}, t) \right) \\ & = \frac{1}{k} \beta_g(\mathbf{r}, t) \sum_{j=1}^J \bar{\nu}_j(\mathbf{r}, t) \Sigma_{f,j}(\mathbf{r}, t) \phi_j(\mathbf{r}, t) \\ & - \lambda_g C_g(\mathbf{r}, t) \end{aligned} \quad (3.9 \text{ revisited})$$

$$\begin{aligned} & \frac{\partial}{\partial t} C_g(\mathbf{r}, t) + \nabla \cdot (C_g(\mathbf{r}, t) \mathbf{u}_f) + \nabla \cdot \left(-\frac{D_g(\mathbf{r}, t)}{\gamma_f} \nabla C_g(\mathbf{r}, t) \right) \\ & = \frac{1}{k} \beta_g(\mathbf{r}, t) \sum_{j=1}^J \bar{\nu}_j(\mathbf{r}, t) \Sigma_{f,j}(\mathbf{r}, t) \phi_j(\mathbf{r}, t) \\ & - \lambda_g C_g(\mathbf{r}, t) \end{aligned} \quad (3.10 \text{ revisited})$$

As mentioned, OpenFOAM facilities to discretize and assemble a matrix-vector system for each equation are quite well developed, however, as of version 8 it is incapable of assembling multiple matrices in what is called a “block matrix” in order to solve multiple equations together.

If we are to be rigorous, that is only partially true. OpenFOAM is capable of solving equations for the different components of a vector quantity simultaneously, such as velocity in the momentum equations. What it is not capable of is assembling dissimilar quantities, such as velocity, pressure, enthalpy, neutron fluxes, precursors concentration, and the eigenvalue, all in a block matrix-vector system for simultaneous solution.

Being incapable of doing so results in a need to solve each equation separately in what is known as a “segregated” solution. Generally, this approach uses the implicit parameters of one equation to generate explicit sources for other equations, and so on. By iterating each equation sequentially in this manner, it is expected that the procedure will converge to an approximate global solution.

A simplified flow diagram of the algorithm is shown on figure 4.1 and a detailed version is available at the very end of the thesis. In this diagram, each coloured box represents a process in the algorithm and each background light colour represents a solver scope. The grey colour is at the scope of the entire SMS application, the blue area at the scope of the fluid dynamics sub-solver inside of it, and the red area bounds the scope of the neutronics “sub-sub-solver”, which exists inside the fluid dynamics one.

This nested hierarchy allows the development of multiple fluid dynamics and neutronics solvers with slightly different models or capabilities that can be combined in the main solver to form a coupled one fit for the intended purpose.

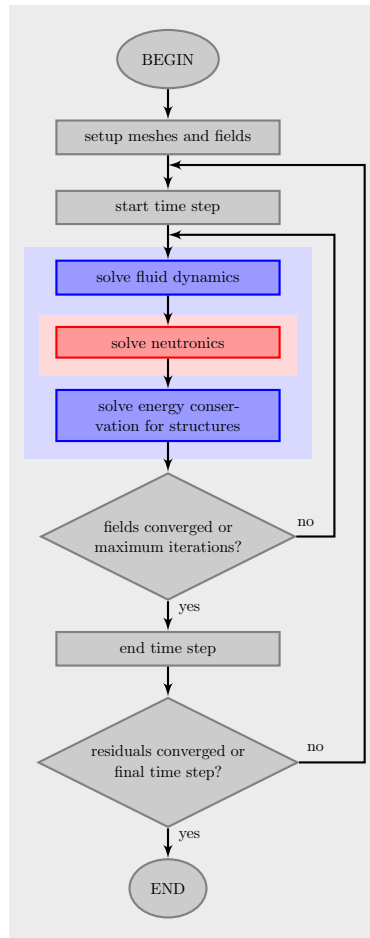


Figure 4.1.: Simplified flow diagram of single-mesh solver

4. Algorithms

After setting up the mesh and fields, the algorithm moves on to start the procedures for solving a fluid dynamics problem, encapsulated by the blue area. It starts by solving mass conservation and moving on to find velocities solving the linear momentum equation.

The equation for conservation of momentum requires special care. Velocity can be directly solved using the momentum equation, but pressure cannot be directly solved from the mass continuity equation, leading to methods such as the SIMPLE (Semi Implicit Method for Pressure Linked Equations) and the PISO (Pressure Implicit Split Operator) widely described in the literature [Moukalled et al., 2016, p. 561]. As we can see, equation 2.39 was manipulated into 4.2, leaving the advection of mass as an explicit term that is calculated using the velocities obtained from solving the momentum equation (or from initial conditions during the first iteration). The PISO method is used to solve the momentum equation, where velocities are calculated with a predicted pressure, which is later corrected in order to enforce mass continuity.

$$\frac{\partial \gamma_f \rho_f}{\partial t} = -\nabla \cdot (\rho_f \mathbf{u}_{h,f}) \quad (4.2)$$

In the solution of equation 2.40, it is necessary to decide how to treat $F_{\text{drag},i}$. A viable approach is to manipulate the Darcy-Weisbach correlation 2.45 as shown on equation 4.3. In this formulation, we split the square of the homogenised velocity into a solution at a previous iteration $^{-1}\mathbf{u}_{h,f}$ and a current one. The previous solution is used to calculate the tensor ζ , which is included as an implicit term on the LHS, as shown in equation 4.4. This is called “semi-implicit” method.

$$F_{\text{drag},i} = \frac{dp}{dx} = -f_D \frac{\rho}{2D_h} \frac{\mathbf{u}_{h,f}^2}{\gamma_f^2} = -f_D \frac{\rho}{2D_h} \frac{^{-1}\mathbf{u}_{h,f}}{\gamma_f^2} \mathbf{u}_{h,f} = \zeta \mathbf{u}_{h,f} \quad (4.3)$$

$$\frac{\partial \rho_f \mathbf{u}_{h,f}}{\partial t} + \nabla \cdot \left(\frac{\rho_f \mathbf{u}_{h,f}}{\gamma_f} \otimes \mathbf{u}_{h,f} \right) - \mu \nabla^2 \mathbf{u}_{h,f} - F_{\text{drag},i} = -\gamma_f \nabla p + \gamma_f \rho_f g + \mathbf{S}_u \quad (4.4)$$

After solving for velocity using a predicted pressure, fluid velocity, density, temperature, and any other state variables of interest are mapped onto the neutronics solver, which is also a solver object encapsulating the red area. The mapping functionality is extremely versatile, allowing access and parametrization of cross-sections by **any scalar field** that exists in the fluid dynamics scope using a lookup feature. Cross-sections are updated with the mapped parameters, and the explicit sources in equation 4.5 are calculated

first. Once these are set, equation 4.6 is solved for each energy group.

$$S_{\text{scat}} = \sum_{i \neq j} \Sigma_{s,i \leftarrow j}(\mathbf{r}, t) \phi_j(\mathbf{r}, t) \quad (4.5a)$$

$$S_{\text{fis}} = \frac{\sum_{j=1}^J \bar{\nu}_j(\mathbf{r}, t) \Sigma_{f,j}(\mathbf{r}, t) \phi_j(\mathbf{r}, t)}{k} \quad (4.5b)$$

$$S_{\text{dnp}} = \sum_{g=1}^G \lambda_g C_g(\mathbf{r}, t) \quad (4.5c)$$

$$\begin{aligned} & \frac{1}{V_i} \frac{\partial}{\partial t} \phi_i(\mathbf{r}, t) + \nabla \cdot (-D_i(\mathbf{r}, t) \nabla \phi_i(\mathbf{r}, t)) + \Sigma_{\text{rem},i}(\mathbf{r}, t) \phi_i(\mathbf{r}, t) \\ &= S_{\text{scat}} + \chi_{p,i}(\mathbf{r}, t)(1 - \beta(\mathbf{r}, t))S_{\text{fis}} + \chi_{d,i}(\mathbf{r}, t)S_{\text{dnp}} \end{aligned} \quad (4.6)$$

The volumetric power from fission reactions $Q_{fis}(\mathbf{r}, t)$ is found through equation 4.7, where $\kappa_j(\mathbf{r}, t)$ is the amount of energy released locally per fission event in group j . It is part of the nuclear data, provided with cross-sections. For a reactor where the liquid fuel is advected, a further elaboration could be made where power is divided into a prompt component that is deposited locally, and a delayed component that is advected. This elaboration has not been pursued during this work.

The time derivative term $\frac{1}{V_i} \frac{\partial}{\partial t} \phi_i(\mathbf{r}, t)$ and eigenvalue k are mutually exclusive as free parameters. A pseudo-steady-state simulation must be performed, where the eigenvalue k is found via a “power iteration” method [Stacey, 2018, p. 78]. This is a very limited algorithm. It can only find the fundamental eigenvalue and eigenvector, and does so with slow convergence. However, it is also one of the most practical for large problems. Some integral quantity to be given for normalisation in order to scale the eigenvector to the right value — the desired integral power P_{norm} in our case. The integral power **at the iteration** P_{iter} is calculated and the deviation from the desired power is used to scale the eigenvalue and eigenvector in equations 4.8.

$$Q_{fis}(\mathbf{r}, t) = \sum_{j=1}^J \kappa_j(\mathbf{r}, t) \Sigma_{f,j}(\mathbf{r}, t) \phi_j(\mathbf{r}, t) \quad (4.7)$$

$$P_{\text{iter}} = \sum_{\mathbf{r}}^R Q_{fis}(\mathbf{r}, t) \quad (4.8a)$$

$$k = {}^{-1}k \frac{P_{\text{iter}}}{P_{\text{norm}}} \quad (4.8b)$$

$$\phi_j(\mathbf{r}, t) = {}^{-1}\phi_j(\mathbf{r}, t) \frac{P_{\text{norm}}}{P_{\text{iter}}} \quad (4.8c)$$

A transient simulation requires a steady-state one to be performed in order to find the multiplication factor k . Once found, k is fixed at the steady-state value in order to keep sources and sinks balanced, and the time derivative term is allowed to vary.

The solution of neutron flux, or more precisely, the fission source, is used to solve a modified version of equation 3.10, shown in equation 4.9. In this equation, the decay term has been moved to the LHS, where it is an implicit term. An important criteria for doing so is that the LHS must stay diagonally dominant to guarantee convergence of iterative numerical solvers. This not only happens, but the term reenforces diagonal dominance.

$$\begin{aligned} & \frac{\partial}{\partial t} C_g(\mathbf{r}, t) + \nabla \cdot (C_g(\mathbf{r}, t) \mathbf{u}_f) + \nabla \cdot \left(-\frac{D_g(\mathbf{r}, t)}{\gamma_f} \nabla C_g(\mathbf{r}, t) \right) + \lambda_g C_g(\mathbf{r}, t) \\ & = \beta_g(\mathbf{r}, t) S_{\text{fis}} \end{aligned} \quad (4.9)$$

Once the neutronics solution is found, the volumetric power field is mapped onto the fluid dynamics solver, where it is used as an energy source Q_{fis} unfolded from S_{he} in equation 4.11. The interfacial source Q_i is defined as the volumetric heat transfer rate given by equation 4.10 between fluid and solid, where $A_{f \rightarrow s}$ is given as input data and U_f calculated from appropriate flow correlations. An arbitrary number of additional sources, such as the solidification/melting one, can be used in order to simulate additional energy-related phenomena.

Once all desired sources have been included, enthalpy is found by solving equation 4.11.

$$Q_{f \rightarrow s} = U_f A_{f \rightarrow s} (T_f - T_s) \quad (4.10)$$

$$\begin{aligned} & \frac{\partial \gamma_f \rho_f h_f}{\partial t} + \nabla \cdot (\rho_f h_f \mathbf{u}_{h,f}) + \nabla \cdot (-\gamma_f \alpha \nabla h) \\ & = \frac{\partial \gamma_f \rho K}{\partial t} + \nabla \cdot (\rho K \mathbf{u}_{h,f}) + \frac{\partial p}{\partial t} + \rho g \cdot \mathbf{u}_{h,f} + Q_{fis} - Q_{f \rightarrow s} + S_{he} \end{aligned} \quad (4.11)$$

In order to find the temperature of solid structures, the same heat transfer rate $Q_{f \rightarrow s}$ is used. The term $Q_{s \rightarrow \text{ext}}$, given by equation 4.12, is optional and designed as a simple way to simulate a heat exchanger with uniform external temperature. The terms U_{ext} , $A_{s \rightarrow \text{ext}}$, and T_{ext} must all be given as input parameter.

If $Q_{s \rightarrow \text{ext}}$ is not used, some other form of heat balance must be provided, either through an arbitrary source S_s or an appropriate boundary condition. When all appropriate conditions are met, equation 4.13 is solved for T_s .

$$Q_{s \rightarrow \text{ext}} = U_{\text{ext}} A_{s \rightarrow \text{ext}} (T_s - T_{\text{ext}}) \quad (4.12)$$

$$\frac{\partial \gamma_s \rho_s c_s T_s}{\partial t} - \nabla \cdot (\gamma_s k_s \nabla T_s) = Q_{f \rightarrow s} - Q_{s \rightarrow \text{ext}} + S_s \quad (4.13)$$

Solid temperature is solved for and fluid temperature is calculated from the fluid enthalpy. These are used to recalculate $Q_{f \rightarrow s}$ and $Q_{s \rightarrow \text{ext}}$ at the next iteration.

Once all quantities solved for, $\mathbf{u}_{h,f}$, p , $\phi_i(\mathbf{r}, t)$ at every group i , $C_g(\mathbf{r}, t)$ at every group g , h_f and T_s have a residual lower than an arbitrary criteria given, the iteration is considered to have converged. Solution controls classes responsible for convergence tolerance checks for fluid solver are inherited from OpenFOAM and the ones to control the neutronics solver are custom-made based on the fluid ones.

Once an iteration is finished, the algorithm advances a time step and the iterative process re-starts.

4.2. Multi-mesh solver

Extending the single-mesh solver, the multi-mesh solver uses multiple meshes to represent the system while relying on field projections functionalities available in the OpenFOAM library, to project values from one mesh onto another for data exchange. The primary circuit is not the limit anymore, being able to model, at least in theory, a system with any number of circuits, any of which can have a nuclear source of heat inserted at the modeller's discretion.

4.2.1. Mesh hierarchy

In order to implement this MMS, a hierarchy needed to be established between phenomena in respect to spatial reach. In a nuclear reactor, the reactor core is typically either

a small part of the primary circuit (e.g., light water reactors or liquid metal cooled fast reactors) or at most the entire primary circuit (e.g., molten salt reactors). While there are variations of the examples, they still fall under one category or another. For this reason, the neutronics mesh is contained by the fluid dynamics one, with full or partial overlap. Information is exchanged by projecting parametrised state variables from the fluid dynamics solver into the neutronics one, and projecting volumetric power field back from the neutronics mesh to the fluid dynamics one as heat source.

The solid mesh at the moment is responsible mainly for simulating heat exchanger between fluid meshes (e.g., primary and secondary circuits). This mesh can potentially envelop all fluid domain in order to perform a thermo-mechanical analysis of the system.

In figure 4.2 we have a problem involving the MMS and 4 meshes: neutronics, fluid 1, fluid 2, and solid. The volumes of fluid 1, solid and fluid 2 overlap in space but are shown separated for illustrative purposes only. The solid mesh is used by the MMS to exchange energy between fluid 1 and fluid 2.

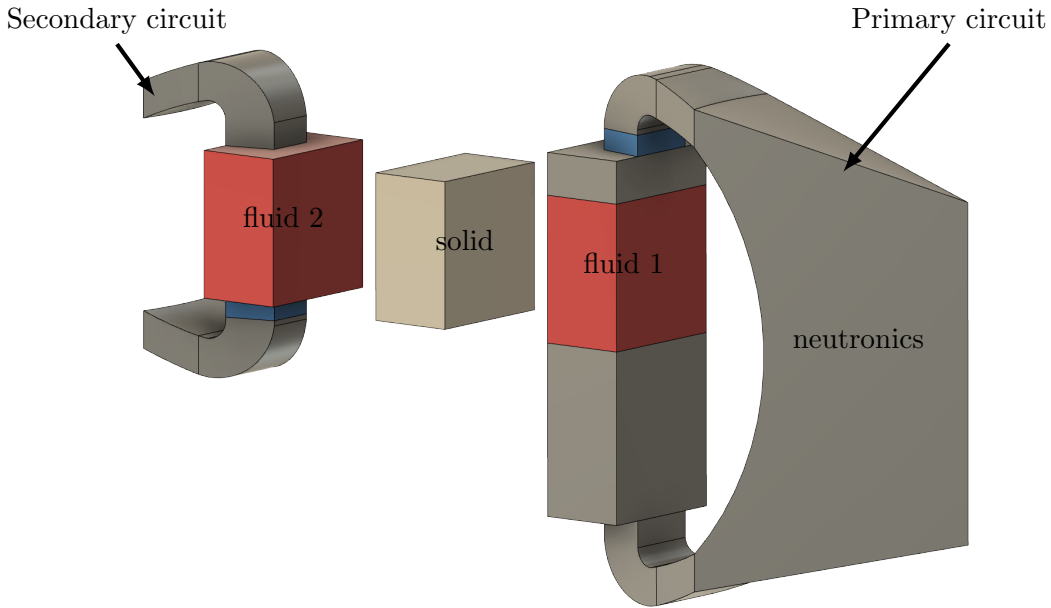


Figure 4.2.: Multiple domains of a multi-mesh solver

4.2.2. Flow diagram and extended algorithm

For most part, the algorithm of this extended solver is the same as the simplified one. Figure 4.3 shows a simplified diagram of the MMS, with a detailed diagram at the

end of the thesis. Like before, the solver starts by creating meshes and fields. There is just more “bookkeeping” to be done at this stage. The major difference happens when it would be time to solve for temperature of the solids. In this solver, designed with modelling of a heat exchanger in mind, the solids are regions that overlap between different fluid meshes and exchange heat between them. Let us assume, for the sake of discussion and simplicity, that we have 2 fluids, and fluid 1 has a nuclear heat source contained somewhere inside, either localised in solid fuelled reactors, or distributed in liquid fuelled ones.

The solver predicts velocity, neutron flux and precursor concentration in each group, enthalpy and corrects pressure for fluid 1, in this order. Then it does the same for fluid 2.

After solving for each fluid, it proceeds to map fluid temperatures to the main solver in the grey area, shown in figure 4.3. After this mapping, it solves an equation 4.14 for solid temperature, which is similar to the previous equation 4.13 but contains as many $Q_{f \rightarrow s}$ as there are fluids.

$$\begin{aligned} & \frac{\partial \gamma_s \rho_s c_s T_s}{\partial t} - \nabla \cdot (\gamma_s k_s \nabla T_s) \\ & = Q_{f1 \rightarrow s} + Q_{f2 \rightarrow s} + \dots \\ & + Q_{s \rightarrow \text{ext}} + S_s \end{aligned} \tag{4.14}$$

It proceeds then to iterate until global convergence is achieved, which might require many iterations depending on how many fluid circuits exist and their interaction. Once this convergence is achieved, it proceeds to the next time step.

4.3. Discussion

In general, a fairly versatile design of the solvers was achieved. The decision of encapsulating solvers through a hierarchy, specially the nuclear one, was a premeditated decision in order to allow other heat sources besides nuclear to be treated in a similar approach. When you remove all its complications and intricacies, nuclear fission in a power reactor is nothing but a fancy way of heating a working fluid. In a system analysis framework, it is nothing but a heat source. It is easy to imagine other heat sources, such as chemical or electrical, as being part of a fluid circuit as well, and the current software architecture allows designing such solvers to replace the nuclear one. The electrical one would be particularly strategical in a circuit containing a working fluid with high melting temperature.

It would be possible, at least in principle, for the MMS to have a different design or

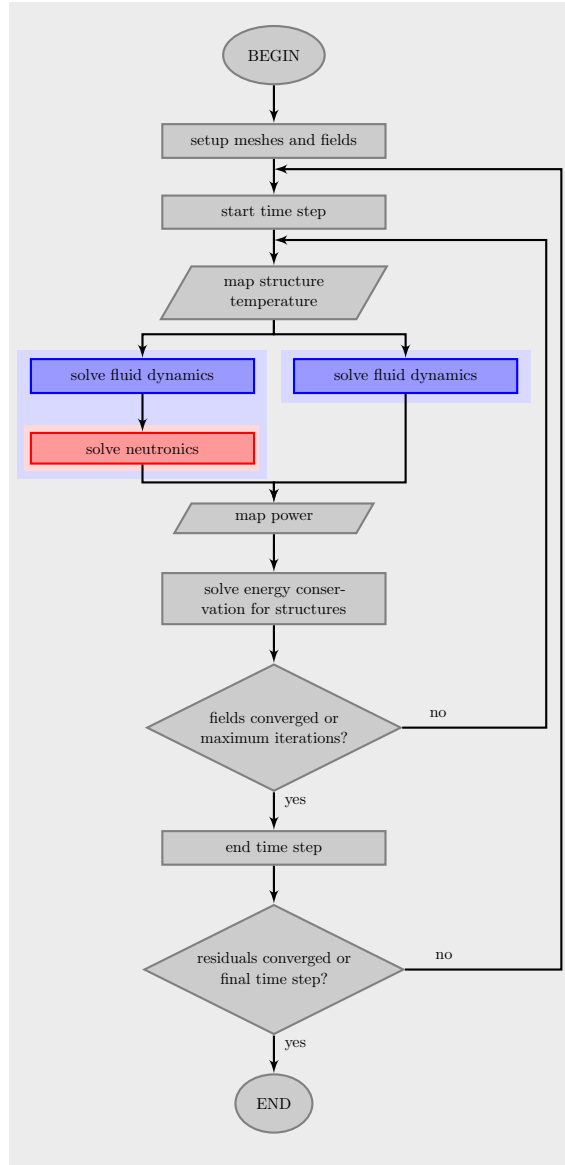


Figure 4.3.: Simplified flow diagram of multi-mesh solver

program flow. For example, to solve for fluid 1, then temperature of solid structure connecting fluid 1 to 2 using data from fluid 2, then solve for fluid 2 and so forth. However, this scheme assumes that fluid 1 is connected only to fluid 2. Most other topologies would become clunky exceptions.

Therefore, it was decided that solving for all fluids first and then solve for temperature of the overlapping structure with update fluid temperatures allowed a more general topology (see Appendix 5.3 for an example of this capability). This does not necessarily imply a better design. It could turn out that this arrangement is more numerically unstable for larger problems, although no investigation of this sort has been performed.

However, this decision was to leave the foundations necessary to eventually allow thermo-mechanical analysis by deforming solid structures. If solids were disconnected from one another, their interaction would become either ambiguous or impossible to predict. Therefore, having a single field of solid temperatures encompassing the entire solid structure was attractive. In principle, this architecture could also allow parallelisation of solution of fluids, by solving multiple fluids in parallel and project all their values to the solid mesh in the end. However to achieve this, it is necessary to go beyond the functionalities of OpenFOAM and use the parallel capabilities of the C++ programming language.

Despite the theoretical capability to model an arbitrary number of fluids, when the number of coupled systems increases, convergence becomes harder and takes longer. If a full reactor is built with all fluid circuits explicitly described, or even a partial one but still with fair amount of complexity, it might simply suffer from numerical instability and diverge for practical problems.

The MMS also suffers from a limitation originating from OpenFOAM itself. During set up of multiple meshes, a field projection object must be created as well. Sometimes the creation of this object fails for reasons apparently connected to the discretisation of the different meshes. In other words, this problem is complicated and connected to the OpenFOAM library on a low level. It is so complicated to the point that the first time it happened, it spurred the decision to create a solver using a single mesh. In other words, regardless of the chapter narrative, the simplified solver was not made as a prototype to the complex one. It was made later, after experiencing the possible problems that could arise from having multiple meshes, without any clear or even an indication of possible solution. It just so happened that this problem could be simply evaded for several cases that did not require multiple meshes in the first place. The biggest downside for coupled problems solved in a single mesh is that fluid dynamics typically determines the cell size either to maintain numerical stability or resolve boundary layers, which results in a mesh that is overly fine for neutronics.

In a steady-state simulation, the time steps are considered to be part of a time-marching algorithm to reach an asymptotic converged steady-state. Convergence tolerance inside a “time step” is not particularly important, although good choices of relaxation schemes do lead to faster convergence — sometimes much faster. However, each time step must be converged within a strict tolerance before advancing in time in a transient simulation. Iterations inside a time step (i.e., SIMPLE loop) are particularly relevant when dealing with the solidification and melting source described in section 2.5. This loop defines the number of iterations between the energy equation and the equation for α , where a few iterations are usually necessary to obtain a good estimation of the mushy region. Convergence for the solidification and melting source is controlled by the residual of enthalpy, which must have a tight tolerance.

No mention was made so far of stability conditions, since it is assumed to be part of background knowledge, but typical conditions must be met. For a fluid simulation, Courant number must be smaller than or equal to 1. This parameter is calculated by both solvers; and for each fluid in the multi-mesh one. It is possible to change cell size, but not practical. Most likely, time steps should be adjusted accordingly instead. While a variable time stepping functionality is part of the solver, it is strongly suggested to restrict its use to transient simulations. Spurious instabilities, especially at the beginning of a steady-state simulation, can negatively affect the time march.

In addition to the fluid stability condition, a Von Neumann stability analysis also results in a similar condition for the diffusion of neutrons [Hutchinson, 2015, p. 58 and 120], or any other quantity. At the moment, no functionality has been implemented to prevent this source of instability as it has not been an issue. Courant number has been the limiting factor for every analysis performed so far.

While OpenFOAM is capable of parallelising problems through domain decomposition, it is important to mention that **the eigenvalue k depends on domain decomposition**. A difference of the order of pcm can result from different decompositions, which is a very large imbalance. **It is important for the steady-state and transient simulations to be performed with the exact same domain decomposition** in order to maintain the equation balanced. A different decomposition with the same number of domains will not work either. It must be exactly the same. Therefore, it is recommended to decompose the domain, and perform the steady-state and any transient simulation before merging the decomposed domains.

5. Verification

Доверяй, но проверяй.

(Russian proverb)

In scientific computing, verification is a task that produces evidence that algorithms in a code are **mathematically** correct and error-free. To do so, an objective reference of correctness is necessary, which is not obtainable from experiments or other codes; it is only obtainable from mathematics itself — solutions to the equations the code embodies.

A very important notion to have from the start is that mathematics is the central topic of verification, and physics is not important at this point. **The objective of verification is to stress-test the discretisation schemes and numerical methods of the code.** The equations being solved are not really important for now as long as it is an equation that we plan to use, and we can obtain a reference solution to it. Physics will take the central stage during validation where, after testing all numerical methods, the equation/model will be evaluated against experimental data to see how accurately it reproduced a natural phenomena. Verification must be performed before validation because during validation it will not be possible to judge if the code is mathematically correct anymore; it should have been done at the verification stage.

The literature in scientific code verification has numerous articles describing different problems in detail, some of them well known (e.g., Sod shock tube, Stefan problem), but a very limited number of works formalising the task of verification itself [Oberkampf & Roy, 2010; Knupp & Salari, 2002; Roache, 1998]. A major focus on computational fluid dynamics is present, mostly done by the aerospace industry, however there are examples available of verification done for radiation transport codes as well, with notable works by Ganapol [2008] and others [Pautz, 2001; Schunert & Azmy, 2011; Wang et al., 2016].

Analytical solutions to the equations are recognised as the “golden standard” of objective reference to demonstrate correctness. Despite this, the literature does not perfectly agree in what fits this description, and might even contradict itself. For example, the exponential function is unanimously understood as analytical while an infinite series is sometimes considered to be approximate [Oberkampf & Roy, 2010, p. 208], even though the exponential function itself is defined by the infinite power series in equation 5.1. The literature might even go as far as dividing these into “exact-analytical”, “near-analytical”

and “semi-analytical” [Ganapol, 2008, p. XXI].

$$e^x := \sum_{k=0}^{\infty} \frac{x^k}{k!} \quad (5.1)$$

Faced with these inconsistencies, it is better to take a step back and peer through the underlying logic. Our code, composed of many pieces and sub-algorithms, produces a numerical solution over a discretised domain, containing many sources of uncertainty — specially from the discretisation itself. The characteristic that we desire from a reference solution is, at least ideally, “unquestionable trust”. If a numerical solution agrees with a reference solution “it is correct”, and if it disagrees “it is the code’s fault”. In order to obtain such reference, we shall consider 2 types of solutions, **distinct by their accuracy**:

1. Analytical: a continuous form solution that is either exact or can be evaluated with arbitrary accuracy and error estimate.
2. Semi-analytical: a continuous form solution that can be evaluated with only finite accuracy, albeit very high one, by a numerical method such as an iterative algorithm.

A type 1 solution has arbitrary accuracy, implying that it can be evaluated up to infinite precision with no constraints besides time of evaluation. Therefore any inaccuracy of the solution is negligible compared to results of the numerical simulation. For an analytical solution, unquestionability rests on the **solution** only.

A type 2 solution has finite accuracy constrained by the numerical method used to find it. This is not the numerical method of the code we would like to verify but an external numerical method that is used to evaluate a reference solution that cannot be solved analytically (e.g., a transcendental equation). While it is understandable and debatable if such a solution would fit the description of an “objectively correct reference”, its continuous nature does eliminate a major source of errors — the discretisation. Therefore, a semi-analytical solution can have an accuracy much higher than the numerical solution provided by the code we would like to verify, but the numerical method used to find it is questionable. The way to resolve this is to first verify the numerical method itself using an analytical solution, which becomes a required pillar of trust of a semi-analytical solution whose unquestionability rests on the **solution and numerical method**.

The challenge now lies on finding such reference solutions for the equations to be solved. This again can be divided in 2 overall approaches:

1. Forward problem: find a solution that satisfies the equation.

2. Backward problem: find the source that satisfies a prescribed solution.

To exemplify this, let us consider a simple steady-state heat diffusion equation 5.2 in a uniform isotropic medium.

$$-k\nabla^2 T = S \tag{5.2}$$

Depending on geometry, boundary conditions and source S , the equation has well known analytical solutions $T(\mathbf{r})$. This is the forward approach; the typical way one would think of verifying a code that implements this heat diffusion equation, which is fine for a limited number of cases.

In the backward approach, one would select a domain and prescribe an analytical solution $T(\mathbf{r})$ to the problem. This prescribed solution would be evaluated at the domain boundaries to find the appropriate boundary conditions, and would be inserted into the equation in order to find S . This approach is also called “method of manufactured solutions” (Oberkampf & Roy, 2010, p. 219; Knupp & Salari, 2002, p. 43), a highly versatile method for code verification.

In this chapter, two problems are used to verify ATARI. The first is the “Stefan problem”, a forward problem of the phase-change category that exercises the solidification/melting source in a very constrained scope. Its solution is semi-analytical, by the definition set above, since one of the parameters of the solution depends on an iterative evaluation of a transcendental equation. The second is a backward problem with imposed analytical solution. As published [de Oliveira & Mikityuk, 2018], it is proposed as a starting point to verify coupled neutronics and fluid dynamics codes.

These two problems are far from enough to reach even a decent coverage of all the numerical methods in ATARI, but it represents a starting point to improve upon.

5.1. Stefan problem

Phase change problems belong to a class of mathematically challenging problems known as “free boundary” problems where the domain is not fully known a priori. Instead, it is found as part of the solution itself. An example of such problem is known as “Stefan problem” after Jožef Stefan, a Slovenian physicist that studied this class of problems in the context of water-ice transition. In order to obtain a closed form explicit solution for the problem, the following conditions must be satisfied: 1-dimensional, semi-infinite domain (i.e., infinite in one direction or defined beginning but no defined end), uniform initial temperature, constant temperature at the defined boundary and constant thermophysical properties in each phase [Alexiades & Solomon, 1993, p. 33].

In this section, two problems with explicit solution will be used, known in the literature as the “1-phase Stefan problem” and the “2-phase Stefan problem. The 1-phase Stefan problem is characterised by starting with the solid at $T = T_{\text{melt}}$, while the 2-phase one is characterised by starting with the solid at $T < T_{\text{melt}}$. Both problems are technically 2-phase problems, but the literature names them in this way by referring to “active phases”, since in the 1-phase problem the solid is at melting point and does not change temperature, only melts.

In these problems we consider a semi-infinite 1-dimensional domain, where a substance is solidified at the start of the numerical experiment. The temperature of the domain boundary is instantaneously increased to a temperature above the melting point of the solid, causing heat to diffuse into the domain, melt the solid from the defined boundary, and propagate the phase-change interface through the semi-infinite domain. Since in OpenFOAM the domain cannot be infinite in any direction, each problem will have a special consideration that will relax this condition.

A description that is complete enough to allow implementation of the problem will be provided in order to keep the text relatively complete for future reference.

5.1.1. 1-phase Stefan problem

A domain of 1 m length is solid at 500 K, same as the melting temperature. The left boundary is at a fixed temperature T_l above the melting point, at 600 K in this case. Due to the domain already being at the melting temperature, the semi-infinite domain condition can be relaxed by simply preventing the melting front from reaching the right boundary, which will be taken as adiabatic (i.e., insulated) but could also be at fixed temperature equal to melting temperature with same effect in this case. Therefore, as soon as the simulation starts, heat diffuses into the domain from the left boundary and gradually melts it. An elaborate description of the problem, with solution, can be found in [Alexiades & Solomon, 1993, p. 34]. Properties used for the fluid are given in table 5.1.

Table 5.1.: Stefan problem fluid properties

Density (ρ)	10^3 kg m^{-3}
Specific heat capacity (c)	$10^3 \text{ J kg}^{-1} \text{ K}^{-1}$
Thermal conductivity (k)	$10^2 \text{ W m}^{-1} \text{ K}^{-1}$
Latent heat (L)	10^6 J kg^{-1}
Melting temperature (T_{melt})	500 K

The semi-analytical solution to the problem predicts the position of the melting front

$X(t)$ at time t to be given by equation 5.3.

$$X(t) = 2\lambda\sqrt{\alpha_1 t} \quad (5.3)$$

where the thermal diffusivity $\alpha_1 = \frac{k}{\rho c}$, and λ is a parameter that will be further explained later. The temperature at a distance x and time t from the fixed temperature boundary condition on the left is given by:

$$T(x, t) = T_1 - \Delta T_1 \frac{\operatorname{erf}\left(\frac{x}{2\sqrt{\alpha_1 t}}\right)}{\operatorname{erf}(\lambda)} \quad (5.4)$$

where $\Delta T_1 = (T_1 - T_{\text{melt}})$ and erf denotes the error function $\operatorname{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-s^2} ds$. The parameter λ is given numerically by the roots of the transcendental equation 5.5, which can be found by an iterative numerical method such as Newton-Raphson or Bisection.

$$\frac{\operatorname{St}_l}{e^{\lambda^2} \operatorname{erf}(\lambda)} = \lambda\sqrt{\pi} \quad (5.5)$$

In this equation, we can see the Stefan number for the liquid, $\operatorname{St}_l = \frac{c}{L} \Delta T_1$. The solution to this equation can be approximated by:

$$\lambda \approx 0.706\sqrt{\operatorname{St}_l}(1 - 0.21(0.5642 \cdot \operatorname{St}_l)^{0.93-0.15 \cdot \operatorname{St}_l}) \quad (5.6)$$

As can be seen on figure 5.1, the error of the approximation decreases for low St_l . In our case $\operatorname{St}_l = 0.1$, resulting in $\lambda \approx 0.2199$ with negligible error.

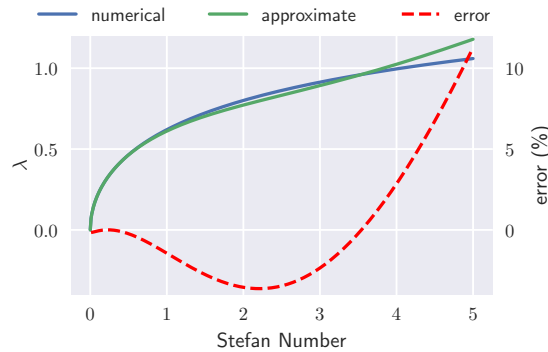


Figure 5.1.: Solution for λ

5. Verification

For the numerical solution of the problem, the domain is discretised into 1000 cells, resulting in cells that are 1 mm thick. As expected, figures 5.2 and 5.3 show the melting front moving away from the higher temperature boundary on the left, in the direction of the insulated boundary on the right. The latent heat and parameters for thermal diffusivity were purposely chosen in order to sharpen the melting front of the numerical solution and allow a close to unambiguous evaluation of its position.

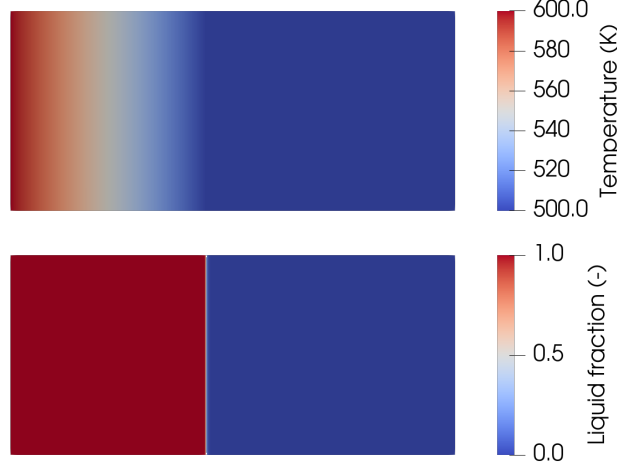


Figure 5.2.: Temperature and Liquid fraction fields at $t = 10\,000\text{ s}$

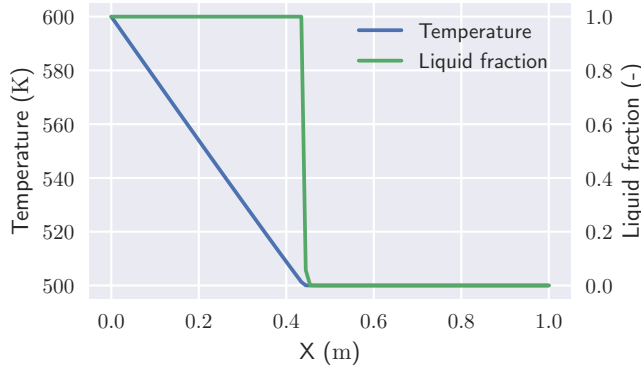


Figure 5.3.: Temperature and Liquid fraction at $t = 10\,000\text{ s}$

Figures 5.4 and 5.5 show the comparison of the reference solution (black marker x) against the results of the numerical simulation (shown as the line). Figure 5.4 shows the change of position of the melting front as a function of time as it moves to the right. Figure 5.5 shows the evolution of temperature at 4 locations (0.1, 0.2, 0.3 and 0.4 m) away from the left boundary. As can be seen, an accurate agreement between numerical and reference solution was achieved.

In addition to the comparison of results with the reference solutions, an analysis of the L_1 and L_2 norms with grid refinement was made. These norms are calculated as shown

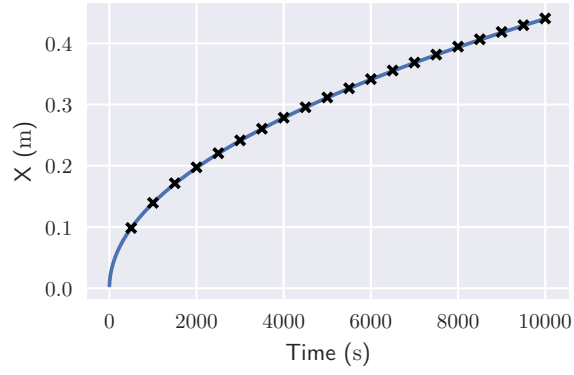


Figure 5.4.: Melting front position as a function of time

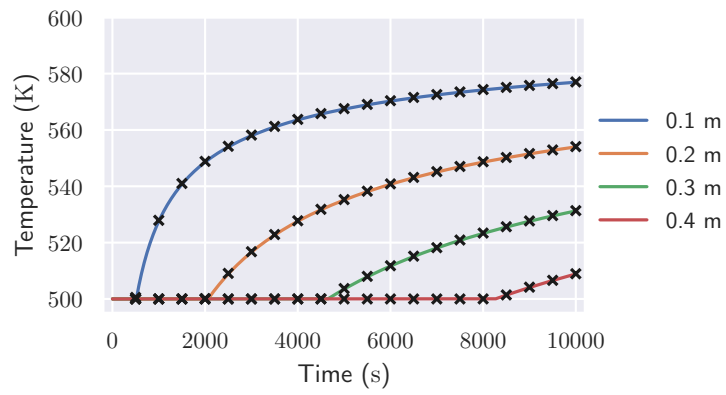


Figure 5.5.: Temperatures as a function of time at 4 positions

5. Verification

in equations 5.7 and 5.8, where ω_n is the cell volume and Ω is the domain volume.

$$\|r - r_{\text{ref}}\|_1 = \frac{1}{\Omega} \sum_{n=1}^N \omega_n |r_n - r_{n,\text{ref}}| \quad (5.7)$$

$$\|r - r_{\text{ref}}\|_2 = \left(\frac{1}{\Omega} \sum_{n=1}^N \omega_n |r_n - r_{n,\text{ref}}|^2 \right)^{\frac{1}{2}} \quad (5.8)$$

As can be seen, the L_1 norm represents the mean volume-weighted absolute error, and the L_2 norm the root-mean-square of the volume-weighted absolute error. For the particular case of uniform meshes, as used in this study, the equations for the norms degenerate into simple averages over the cells.

Doubling the number cells during mesh refinement should divide the norm of the error by 2 for a first order method \mathcal{O} and divide by 4 for second order method \mathcal{O}^2 . An accuracy of first order for space is recovered by the method as shown in figure 5.6. Time step, mesh refinement and number of outer iterations interact in a complex way to determine the order of accuracy.

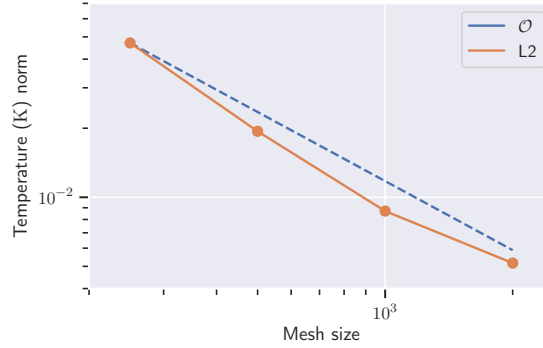


Figure 5.6.: Spatial Order of Accuracy at last time step

5.1.2. 2-phase Stefan problem

If the 1-phase Stefan problem was characterised by starting with the solid at $T = T_{\text{melt}}$, the 2-phase one is characterised by starting with $T < T_{\text{melt}}$. An initial temperature of 400 K will be used in this simulation. For a semi-infinite domain, the solutions to the problem are equations 5.9 and 5.10 [Alexiades & Solomon, 1993, p. 46]. In this case, the semi-infinite domain condition can be relaxed by programming the right boundary, as will be discussed soon.

Equation 5.9 describes the position of the melting front as a function of time. The function has the same form as the solution for the 1-phase problem (Equation 5.3), however λ differs and will be discussed.

$$X(t) = 2\lambda\sqrt{\alpha_1 t} \quad (5.9)$$

Equation 5.10 describes the evolution of temperature as a function of time and position in the domain. The solution for $0 < x < X(t)$ is of the same form as for the 1-phase problem, but with different λ . This solution describes the behaviour of the domain as it rises above melting temperature. When $x > X(t)$ we have an additional solution that describes the increasing temperature of the domain that is below melting temperature.

$$T(x, t) = \begin{cases} T_l - \Delta T_l \frac{\operatorname{erf}\left(\frac{x}{2\sqrt{\alpha_1 t}}\right)}{\operatorname{erf}(\lambda)} & \text{for } 0 < x < X(t) \\ T_s + \Delta T_s \frac{\operatorname{erfc}\left(\frac{x}{2\sqrt{\alpha_s t}}\right)}{\operatorname{erfc}(\nu\lambda)} & \text{for } X(t) < x \end{cases} \quad (5.10)$$

where T_l and T_s are temperatures of the liquid and solid respectively. In this solution we introduce $\Delta T_s = (T_{\text{melt}} - T_s)$ (note that it is slightly different than ΔT_l), $\nu = \sqrt{\alpha_1/\alpha_s}$, and make use of the complimentary error function $\operatorname{erfc}(z) = 1 - \operatorname{erf}(z)$. The parameter λ gains an additional term in equation 5.11 if compared to equation 5.5.

$$\frac{\operatorname{St}_l}{e^{\lambda^2} \operatorname{erf}(\lambda)} - \frac{\operatorname{St}_s}{\nu e^{\nu^2 \lambda^2} \operatorname{erfc}(\nu\lambda)} = \lambda\sqrt{\pi} \quad (5.11)$$

where the Stefan number for the solid is $\operatorname{St}_s = \frac{c}{L}\Delta T_s$. Whereas for the 1-phase case λ could be approximated by a fitting that was accurate for a wide range of Stefan numbers, in the 2-phase problem such fitting has a fairly limited validity. Therefore, it was decided to solve λ numerically using the Newton-Raphson method. The iterative solver was verified using a few polynomials.

In the 1-phase problem, the semi-infinite domain could be easily taken to be finite as long as the melting front did not reach the right boundary. In the 2-phase problem, this is not possible because the temperature at the right boundary is changing at all times during the simulation. The way to achieve a finite domain despite this situation is to program the right boundary to have a temperature given by the reference solution 5.10. This solution depends on t , x and position of $X(t)$ relative to x . We know x , which is the position of the right boundary. If the simulation is set up in such a way that the melting front does not reach the right boundary, it is possible to eliminate the

5. Verification

dependence on the melting front position since $X(t) < x$ during the entire simulation. Therefore, we can program a simplified equation 5.12, that depends on t only.

$$T_{bc}(t) = T_s + \Delta T_s \frac{\operatorname{erfc}\left(\frac{x_{bc}}{2\sqrt{\alpha_s t}}\right)}{\operatorname{erfc}(\nu\lambda)} \quad (5.12)$$

The domain is discretised in the same way as for the 1-phase problem. Compared to the 1-phase problem, the only changes are the initial temperature of the solid and the boundary condition on the right. From the amount of similarities, including the solutions, it is reasonable to expect the results to look similar.

Figures 5.7 and 5.8 show the melting front moving away from the higher temperature boundary on the left, in the direction of the boundary on the right. It is very similar to the 1-phase problem except that in this case the temperature rises in the solid part of the domain.

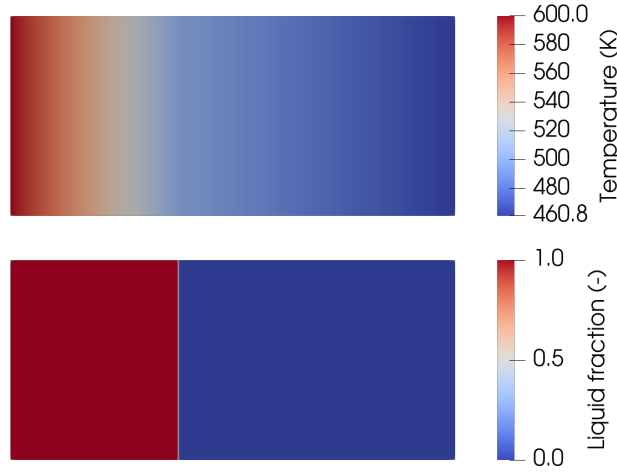


Figure 5.7.: Temperature and Liquid fraction fields at $t = 10\,000$ s

As before, figures 5.9 and 5.10 show the comparison of the reference solution to the results of the numerical simulation. In figure 5.9 we notice that the agreement for the movement of the melting front is accurate. Figure 5.10 shows the evolution of temperature at 6 locations (0.1, 0.2, 0.3, 0.4, 0.6 and 0.8 m) away from the left boundary. Half of the locations are inside the liquid region and the other half in the solid region by the end of the simulation. As predicted, the curve is continuous but not smooth, having a “kink” in the location of phase transition, where it is not differentiable. The agreement between numerical and reference solution is evident.

Similar to the 1-phase problem, we found that the method recovers a spatial order of accuracy of approximately first order as shown in figure 5.11. In addition, by halving

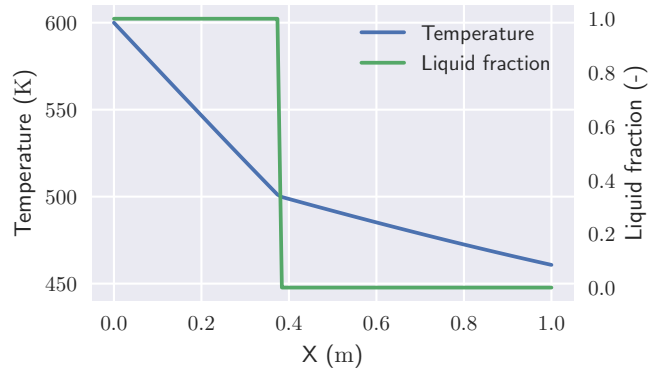
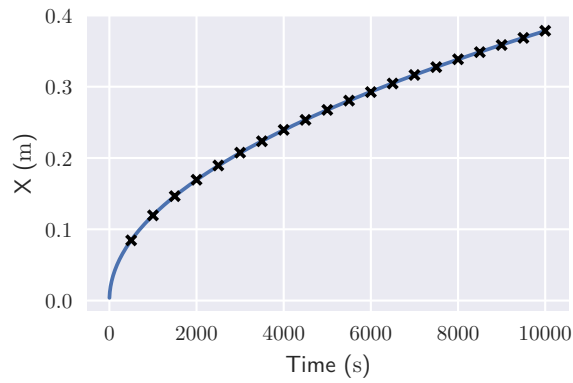
Figure 5.8.: Plot of Temperature and Liquid fraction at $t = 10\,000$ s

Figure 5.9.: Melting front position as a function of time

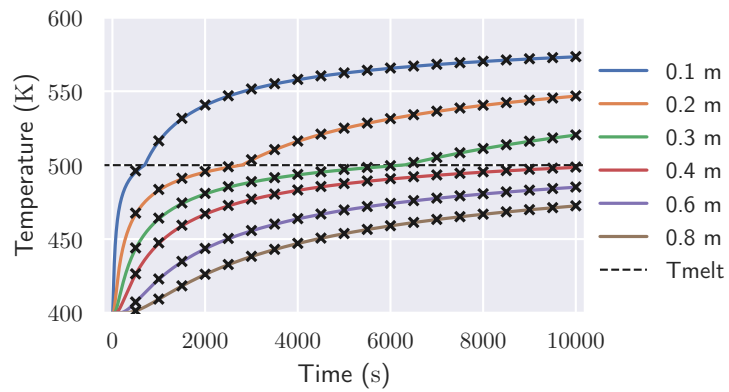


Figure 5.10.: Temperatures as a function of time at 6 positions

the time step (i.e., temporal mesh refinement) we can see a first order temporal accuracy in figure 5.12. A complex interaction between time step, mesh refinement and number of outer iterations happen, and the main bottleneck to improvement of norms is the error at the melting front position.

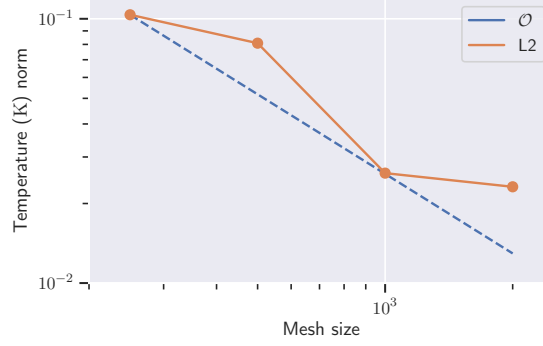


Figure 5.11.: Spatial Order of Accuracy at last time step

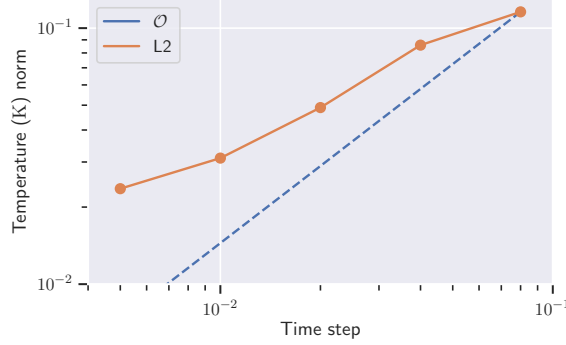


Figure 5.12.: Temporal Order of Accuracy with 250 cells

5.2. Recirculating cavity

This problem uses a prescribed analytical solution, also known as manufactured solution, that purposely results in a very simple source and can be easily evaluated at the domain boundaries.

Special considerations are necessary when performing this procedure for the first time. On one hand, manufactured solutions should use functions that do not disappear upon differentiation, therefore the complexity of the source tends to escalate quickly; usually creating these solutions require the support of symbolic mathematics packages. On the other hand, the equations to be solved are not implemented on the symbolic mathematics program when one first attempts this. Therefore it is necessary to verify their appropriate implementation and manage the complexity of the source, else it becomes error-prone.

The process is not trivial due to these opposing needs. Our strategy will consist of basing the solutions on trigonometric functions (e.g., sines and cosines), which never disappear upon differentiation and will cancel each other when properly chosen.

In the next few sections we discuss the domain shape, analytical solutions, equations and source terms, and boundary conditions. Although an attempt at separating these different aspects is made, there is a high degree of interaction between them when simplicity is targeted.

5.2.1. Domain specification

Inspired by the lid-driven cavity (the most basic OpenFOAM tutorial), the domain proposed is a square centred at point $(0,0)$, with dimensions of $\pi \times \pi$ meters, as shown in figure 5.13.

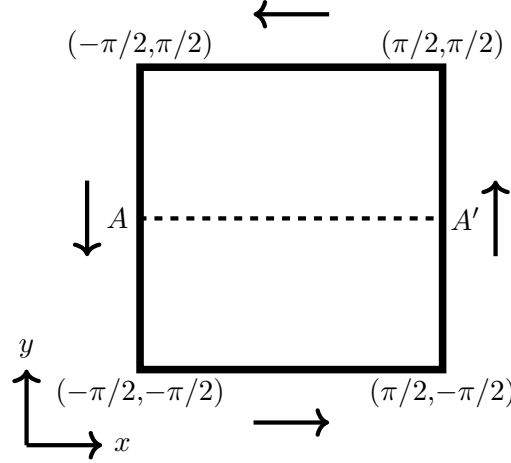


Figure 5.13.: Recirculating cavity domain

This simplicity is desirable for verification activities, at least during initial stages. It reduces the complexities introduced by the mesh into the procedure, and allows its systematic refinement.

For later reference, it is convenient to define the point at the left-bottom corner of the domain as follows:

$$(x_{\min}, y_{\min}) = -\pi/2, -\pi/2 \quad (5.13)$$

Care was taken in selecting a domain that is centred on $(0,0)$ instead of placing the (x_{\min}, y_{\min}) point at $(0,0)$. If the second option had been taken, an asymmetrical

round-off error would be introduced since the code can compute $\sin(0)$ exactly, but not $\sin(\pi)$.

5.2.2. Equations and scope

The equations presented in chapter 4 are too complex to be verified in their entirety straightaway, even disregarding our lofty objective of keeping the source simple. In order to meet the objectives, some significant simplifications are necessary in the scope of the verification exercise and the associated equations.

We shall start by considering the single phase flow case, where there are no solid structures homogenised with the fluid. In addition, it shall be limited to the case of a steady-state incompressible flow in laminar regime of a newtonian isotropic fluid with constant properties. In order to enforce incompressible flow conditions, equation 2.1 is simplified into equation 5.14, which in turn is used to simplify equations 2.2 and 2.3 into equations 5.15 and 5.16 after expansion of divergence operator with calculus identities.

$$\nabla \cdot \mathbf{u} = 0 \quad (5.14)$$

$$\rho(\mathbf{u} \cdot \nabla)\mathbf{u} - \mu \nabla^2 \mathbf{u} = -\nabla p + \mathbf{S}_u \quad (5.15)$$

$$\rho c_p(\mathbf{u} \cdot \nabla)T - k \nabla^2 T = Q_{\text{fis}} \quad (5.16)$$

The source \mathbf{S}_u in equation 5.15, sometimes referred to as “forcing term” in the literature, is what we would like to find in the backwards problem. Other equations could have their own arbitrary source but, as we will see later, the solutions chosen will result in only the momentum equation having one.

Compared with the fluid dynamics, the neutron diffusion equation requires more care since the diffusion condition is satisfied only by judicious choice of parameters. Let us revisit equations 3.6 and 3.7 below, and simplify them appropriately.

$$\begin{aligned}
& \frac{1}{V_i} \frac{\partial}{\partial t} \phi_i(\mathbf{r}, t) + \nabla \cdot (-D_i(\mathbf{r}, t) \nabla \phi_i(\mathbf{r}, t)) + \Sigma_{t,i}(\mathbf{r}, t) \phi_i(\mathbf{r}, t) \\
&= \sum_{j=1}^J \Sigma_{s,i \leftarrow j}(\mathbf{r}, t) \phi_j(\mathbf{r}, t) \\
&+ \chi_{p,i}(\mathbf{r}, t) (1 - \beta(\mathbf{r}, t)) \frac{\sum_{j=1}^J \bar{\nu}_j(\mathbf{r}, t) \Sigma_{f,j}(\mathbf{r}, t) \phi_j(\mathbf{r}, t)}{k} \\
&+ \chi_{d,i}(\mathbf{r}, t) \sum_{g=1}^G \lambda_g C_g(\mathbf{r}, t)
\end{aligned} \tag{3.6 revisited}$$

$$\begin{aligned}
& \frac{\partial}{\partial t} C_g(\mathbf{r}, t) + \nabla \cdot (C_g(\mathbf{r}, t) \mathbf{u}) + \nabla \cdot (-D_g(\mathbf{r}, t) \nabla C_g(\mathbf{r}, t)) \\
&= \beta_g(\mathbf{r}, t) \frac{\sum_{j=1}^J \bar{\nu}_j(\mathbf{r}, t) \Sigma_{f,j}(\mathbf{r}, t) \phi_j(\mathbf{r}, t)}{k} \\
&- \lambda_g C_g(\mathbf{r}, t)
\end{aligned} \tag{3.7 revisited}$$

Let us consider the following simplifications:

1. Steady-state case: drop time dependencies (i.e., $f(\mathbf{r}, t) \rightarrow f(\mathbf{r})$) and time derivatives, just as was done with the fluid dynamics equations.
2. No delayed neutron precursors: all neutrons come from prompt fission, which eliminates the delayed neutron source and sets $\beta(\mathbf{r}) = 0$ in equation 3.6, and eliminates equation 3.7.
3. 1 energy group: drop all energy subscripts and $\chi_p(\mathbf{r}) = 1$.
4. Constant properties: diffusion coefficient can be brought out of the divergence.

Taken together, these turn equation 3.6 into 5.17.

$$\begin{aligned}
& -D(\mathbf{r}) \nabla^2 (\phi(\mathbf{r})) + \Sigma_t(\mathbf{r}) \phi(\mathbf{r}) \\
&= \Sigma_s(\mathbf{r}) \phi(\mathbf{r}) + \frac{\bar{\nu}(\mathbf{r}) \Sigma_f(\mathbf{r}) \phi(\mathbf{r})}{k}
\end{aligned} \tag{5.17}$$

For the sake of simplicity, dependency on \mathbf{r} will be left implicit; everything but the eigenvalue depends on position. Let us perform the subtraction $\Sigma_t - \Sigma_s = \Sigma_a$, where Σ_a is the absorption cross-section, leading to equation 5.18.

$$-D \nabla^2 \phi + \Sigma_a \phi = \frac{\bar{\nu} \Sigma_f \phi}{k} \tag{5.18}$$

Finally, it is interesting to aggregate all neutronics parameters into a single lumped term, as shown in equation 5.19.

$$\nabla^2 \phi + \frac{\bar{\nu}\Sigma_f - \Sigma_a}{D} \phi = 0 \quad (5.19)$$

This lumped parameter in equation 5.20 is called the “material buckling” B_m , which only depends on material properties. The concept of buckling will be very useful when deciding the neutronic properties.

$$\nabla^2 \phi + B_m^2 \phi = 0 \quad (5.20)$$

Equations 5.14, 5.15, 5.16, and 5.20 represent the multiphysics¹ problem that is to be verified in this section.

5.2.3. Analytical solutions

The set of analytical solutions imposed for velocity, pressure, temperature, and neutron flux are shown in equations 5.22-5.25. These have been parametrised by α and β as defined in equations 5.21, with the objective of making the solutions invariant to translation (i.e., variations of x_{\min} or y_{\min}) or stretching (i.e., variations of a or b) of the domain. In this parametrization of space variables, a and b are the dimensions of the domain on the x and y directions. For the particular case proposed $a = b = \pi$ m, resulting in a immediate simplification of equations 5.21.

$$\begin{aligned} \alpha(x) &= \frac{\pi}{a}(x - x_{\min}) \\ \beta(y) &= \frac{\pi}{b}(y - y_{\min}) \end{aligned} \quad (5.21)$$

In equation 5.23, P_c represents the pressure at the centre of the cavity, which is the minimum pressure and also the reference one. Similarly, T_b in equation 5.24 represents the constant temperature at the boundaries of the domain, which will be specified later on. The amplitude above T_b at the centre ΔT_c depends on simulation parameters (e.g., thermal conductivity) to be discussed. Equation 5.25 is the flux in the domain, and ϕ_c is flux in the centre, or an amplitude above a baseline along which flux is taken as 0.

$$\begin{aligned} u_x(x, y) &= \sin(\alpha) \cos(\beta) \\ u_y(x, y) &= -\cos(\alpha) \sin(\beta) \end{aligned} \quad (5.22)$$

¹The problem deals with one-way coupling only at this stage.

$$P(x, y) = P_c + \rho \frac{\cos^2(\alpha) + \cos^2(\beta)}{2} \quad (5.23)$$

$$T(x, y) = T_b + \Delta T_c \sin(\alpha) \sin(\beta) \quad (5.24)$$

$$\phi(x, y) = \phi_c \sin(\alpha) \sin(\beta) \quad (5.25)$$

The set of equations is arbitrary and does not have to be a physical problem in any sense. The only requirement is for **solutions to be differentiable**, which might imply continuity or smoothness. Another desirable property, although not strictly necessary, is for **solutions to be infinitely differentiable**, avoiding the disappearance of any term of the differential equations.

By inserting the analytical solutions 5.22 and 5.23 into equation 5.15, the resulting source term \mathbf{S}_u is given by equation 5.26, which is very compact.

$$\mathbf{S}_u = 2\mu \mathbf{u} \quad (5.26)$$

5.2.4. Boundary conditions

Evaluating equations 5.22 at the domain boundaries, it is possible to specify the velocity boundary conditions as shown in equations 5.27.

From the boundary conditions for velocity, it is possible to infer qualitatively that, in this problem, the fluid flows in a counter-clockwise pattern, as shown in figure 5.13, with velocity reducing to 0 at the corners of the domain.

$$\begin{aligned} u_x(x, -\pi/2) &= \sin(\alpha) \\ u_x(x, \pi/2) &= -\sin(\alpha) \\ u_y(-\pi/2, y) &= -\sin(\beta) \\ u_y(\pi/2, y) &= \sin(\beta) \end{aligned} \quad (5.27)$$

The absolute pressure of the system is not important for an incompressible problem, but a reference pressure of 10^5 Pa was given since the code requires some value as baseline. What actually matters is the pressure difference created by dynamic forces in the system, commonly known as gauge pressure or, more rigorously, as sealed gauge pressure with the reference pressure as specified. The appropriate boundary condition

for this system sets the gradient of pressure normal to the wall to 0. This represents an uniform Neumann boundary condition, which also satisfies equation 5.23.

For temperature, a uniform Dirichlet boundary condition of $T_b = 1000$ K was used, which results from evaluating equation 5.24 at the boundaries. This effectively represents an infinite heat sink outside the system with the set temperature.

The neutron flux is set to 0 at the boundaries, also resulting from evaluation of equation 5.25. The benefits of this choice and the next steps will be discussed at the end of the chapter.

5.2.5. Parameters

Table 5.2 shows the parameters used for the fluid in this problem. All the parameters are arbitrary, but their choice requires some consideration.

Table 5.2.: Recirculating cavity fluid properties

Parameter	Value
Density (ρ)	10^3 kg m^{-3}
Specific heat capacity (c_p)	$1 \text{ J kg}^{-1} \text{ K}^{-1}$
Thermal conductivity (k)	$50 \text{ W m}^{-1} \text{ K}^{-1}$
Dynamic viscosity (μ)	50 Pa s

Compared to physically sensible values, dynamic viscosity is high in order to force the flow regime to be laminar despite such a large domain. In addition, all the parameters in the table are related in choice in order to keep the advective and diffusive terms in the equations in balance so that both are similarly exercised. For example, the choice of specific heat is related to the choice of density and thermal conductivity in order to keep the terms of equation 5.16 in balance. If this balance is not taken into account and, for example, a high value is given to specific heat while keeping density and thermal conductivity unchanged, numerical noise in the advection term will become significant. This happens because the advection term of the energy conservation equation sums to 0 analytically, but numerically will be a very small non-zero number (of the order of 10^{-8}). This very small number (numerical noise) will be multiplied by a very high value (density \cdot specific heat capacity) and the term will become significant compared to the diffusion one. This type of carelessness negatively impacts the solution and was experienced first handedly during the design of this exercise.

The solution of equation 5.20 requires neutronics parameters, which are also arbitrary to a certain extent. Equation 5.28 is the geometrical buckling of a 2D square homogenous unreflected reactor with extrapolation distance 0. From the equation, and reminding

that $a = b = \pi \text{ m}$, the problem has $B_g^2 = 2 \text{ m}^{-2}$.

$$B_g^2 = \left(\frac{\pi}{a}\right)^2 + \left(\frac{\pi}{b}\right)^2 \quad (5.28)$$

In order for the reactor to be critical, the material buckling given by equation 5.29 has to be equal to the previously calculated geometrical one ($B_g^2 = B_m^2$) [Stacey, 2018].

$$B_m^2 = \frac{\frac{\bar{\nu}\Sigma_f}{k} - \Sigma_a}{D} \quad (5.29)$$

The neutron diffusion coefficient D in equation 5.29, considering isotropic scattering, is given by:

$$D = \frac{1}{3(\Sigma_s + \Sigma_a)} \quad (5.30)$$

From equations 5.29 and 5.30, we generate arbitrary one-group cross-sections considering the following criteria:

1. $k = 1$
2. $D = 2 \times 10^{-2} \text{ m}$
3. $\Sigma_a = 1 \text{ m}^{-1}$

where item 1 is for convenience, 2 is to work with a diffusion coefficient of the same order of magnitude as the mesh size, and 3 is to have $\Sigma_a \ll \Sigma_s$, which is a condition for the diffusion theory to be valid. The neutronic parameters prescribed by the criteria and the calculated ones are consolidated in table 5.3.

Table 5.3.: Recirculating cavity neutronic properties

Parameter	Value
D	$2 \times 10^{-2} \text{ m}$
Σ_a	1 m^{-1}
Σ_s	$15.666 \dots \text{ m}^{-1}$
$\bar{\nu}\Sigma_f$	1.04 m^{-1}
$\kappa\Sigma_f$	1 J m^{-1}

5. Verification

Inserting the analytical solutions 5.22, 5.24, and 5.25 into equation 5.16, we get:

$$\kappa\Sigma_f\phi_c = 2k\Delta T_c \left(\left(\frac{\pi}{a} \right)^2 + \left(\frac{\pi}{b} \right)^2 \right) \quad (5.31)$$

For a flux to power conversion factor (commonly known as kappa-fission) $\kappa\Sigma_f = 1 \text{ J m}^{-1}$, $k = 50 \text{ W m}^{-1} \text{ K}^{-1}$ and an arbitrary $\Delta T_c = 100 \text{ K}$, we get a $\phi_c = 10^4 \text{ m}^{-2} \text{ s}^{-1}$.

Finally, it is necessary to calculate the integral power used to normalise flux in ATARI. Since $\kappa\Sigma_f$ is constant, we can find this parameter by integrating the flux $\phi(x, y)$ given by equation 5.25 over the domain, getting the integral flux Φ , as in equation 5.32. It is important to mention that although the domain is conceptually 2D, all parameter were given with 3D units for better understanding. However, this implies that in order to achieve strict unit matching the integration has to cover 3 dimensions, where the third dimension over a pseudo-axis z measures 1 m, being numerically irrelevant.

$$\Phi = \int_0^1 \int_{-\pi/2}^{\pi/2} \int_{-\pi/2}^{\pi/2} \phi_c \sin(\alpha) \sin(\beta) dx dy dz = 4\phi_c \quad (5.32)$$

If $\phi_c = 10^4 \text{ m}^{-2} \text{ s}^{-1}$, as previously mentioned, then $\Phi = 4 \times 10^4 \text{ m s}^{-1}$. Considering $\kappa\Sigma_f = 1 \text{ J m}^{-1}$ once again, power will be equal to $\kappa\Sigma_f\Phi = 4 \times 10^4 \text{ W}$.

5.2.6. Results

In order to compare the results of the numerical simulation with the analytical one, the respective fields will be shown besides a plot of values over line AA' shown on figure 5.13. This will allow a mix of qualitative view of the fields, which we have an intuition from the analytical solutions of how they should look like, and a quantitative comparison over a well defined path. The relative error ϵ in a cell n is calculated by $\epsilon_n = |r_n - r_{n,\text{ref}}| / r_{n,\text{ref}} \cdot 100\%$, where r_n and $r_{n,\text{ref}}$ are the numerical and analytical values respectively of a response quantity in the cell.

On figure 5.14 we can see that the flux has the expected behaviour of a cosine shape, showing a peak at the centre and 0 at the border. The shape of the flux is not the typical “chopped cosine” since no extrapolated distance was given to the case. The relative error is very low (below 0.01%), and as we can see, the numerical solution is indistinguishable from the analytical one. The effective multiplication factor was also found with very high accuracy, showing an over-prediction of 0.3 pcm for the reference mesh.

The correct solution of the momentum equation under the conditions of this problem

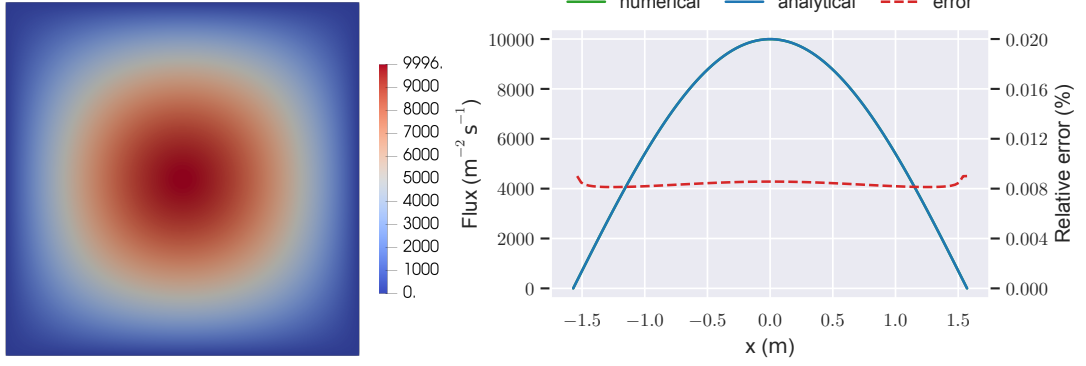


Figure 5.14.: Flux on the domain and results over line AA'

is quite demanding on the divergence discretisation scheme of velocity. A first order upwind scheme fails to generate a correct flow and pressure fields, requiring the use of a second-order upwind scheme with results presented in figures 5.15 and 5.16. It is relevant to mention that the velocity plot in figure 5.15 has a blunt tip in the centre due to the mesh having an even number of cells, therefore there is no cell in the centre of the domain. A simple test was made with a cell in the centre, offering no benefit to the solution. With the lack of any benefit, it was decided not to have a cell in the centre to avoid the problem of defining a relative error in the centre of the domain, where the analytical solution to velocity is 0.

The relative error graphic for velocity shows that it peaks quickly and reduces to an approximately constant value away from the boundary. While a finer mesh at the boundary could appear to offer modest improvements, even though errors are already very low, this improvement was not observed. A mesh refinement at the walls or a mesh with refinement at the walls and in the centre did not convey any improvement over a uniform one.

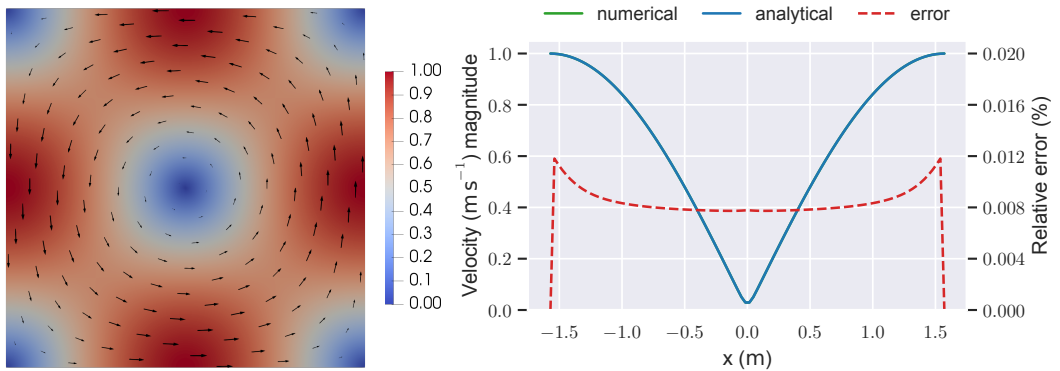


Figure 5.15.: Velocity magnitude on the domain and results over line AA'

5. Verification

Pressure has the lowest error, at approximately 0.0001%, with an unusual curve shape. Upon closer investigation of the numerical and analytical values, it is clear that the numerical value is higher than the analytical closer to the boundary while the reverse is true for the centre. Therefore, the zone of 0 error simply represents an intersection where both happen to be equal.

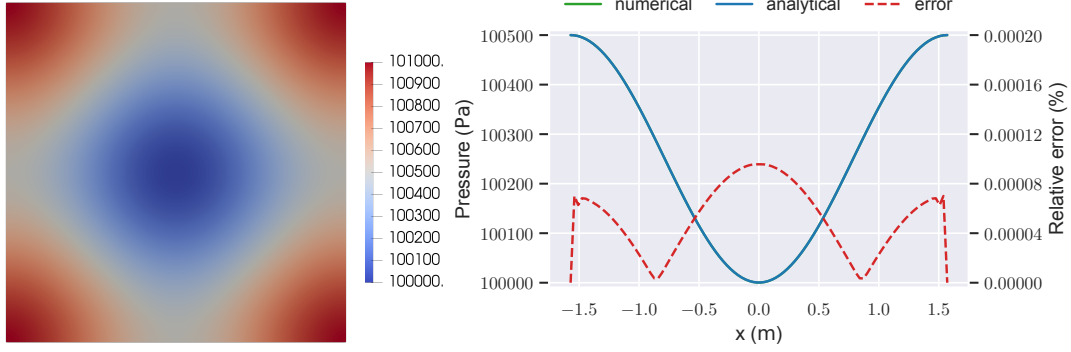


Figure 5.16.: Pressure on the domain and results over line AA'

Solving the temperature field reveals some challenging characteristics of this case. Term dominance can easily be a problem since $\rho c_p \gg k$ usually, resulting in advection strongly dominating over diffusion. Advection of energy should sum to 0 analytically, but numerically becomes a very low non-zero number. Multiplying this numerical noise by a relatively big number results in unacceptable noise levels that prevent the calculation of the correct temperature field. Despite the temperature field being resolved to a very low relative error as shown at figure 5.17, indicating convergence, the error field of this case has a unique shape that is strongly influenced by the advection noise. Investigating the best way to deal with this problem should be a priority in the future in order to increase the robustness of the proposed case.

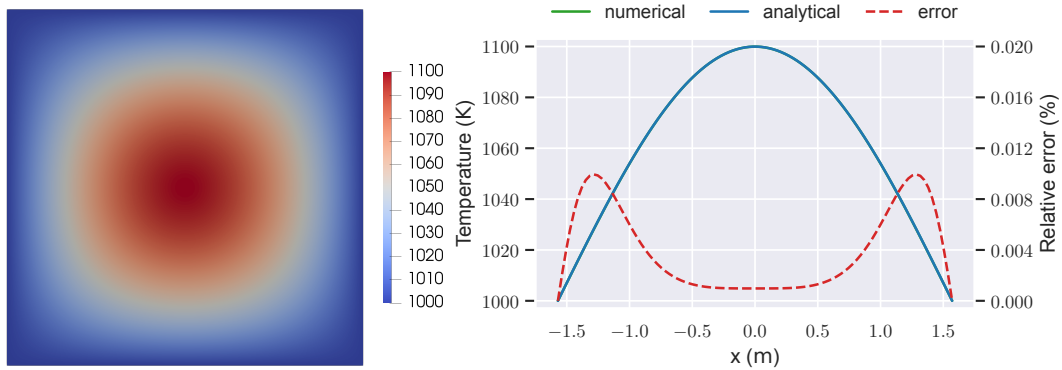


Figure 5.17.: Temperature on the domain and results over line AA'

5.2.7. Mesh studies

The same studies regarding the change of norms L_1 and L_2 norms with grid refinement was made. The intention was to find the coarsest grid that showed converged results in order to minimise execution time and use the case as a basic regression test during code development.

Figure 5.18 shows the L_2 norms of velocity on the domain where mesh size indicates the number of uniform cells in the x and y direction. As can be seen on the figure, the norms decrease quadratically with mesh refinement, following the expected behaviour of the second order scheme used. Norms for other responses show a similar trend.

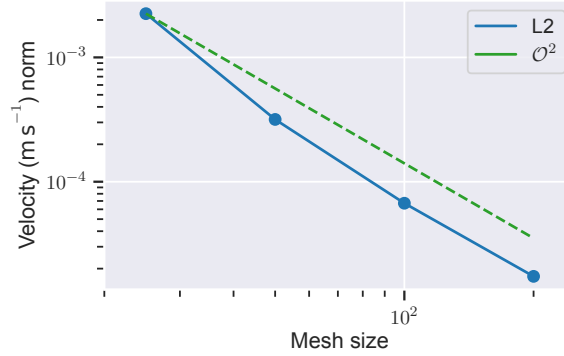


Figure 5.18.: L_1 and L_2 norms for velocity

The increase in wall time from a mesh refinement is shown in figure 5.19. Using a logarithmic scale does not linearise the trend due to the overhead present during startup of the program. At mesh size 25, the minimum wall time is limited by startup operations, where a mapping of values between the fluid mechanics and neutronics meshes is established, and coded sources and boundary conditions are compiled. At mesh size 200 the wall time is limited by processing power. At mesh size 50, the case runs at approximately 50 seconds in a common desktop, making it a good candidate for use as a fast regression testing during code development. At mesh size 100, the problem takes only 180 seconds, making it the appropriate resolution for further developments in this problem in general, where accuracy is more important and finer refinement brings only marginal gains.

5.3. Triple circuit heat exchanger

This rough verification case is also a limited demonstration of the unique capability of ATARI's multi-mesh solver to handle multiple regions that exchange heat.

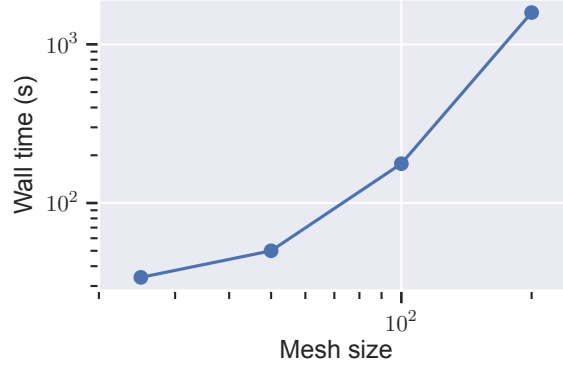
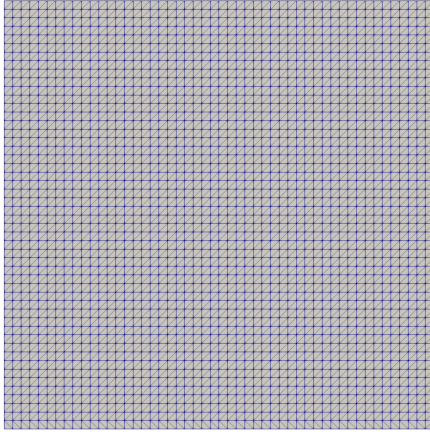


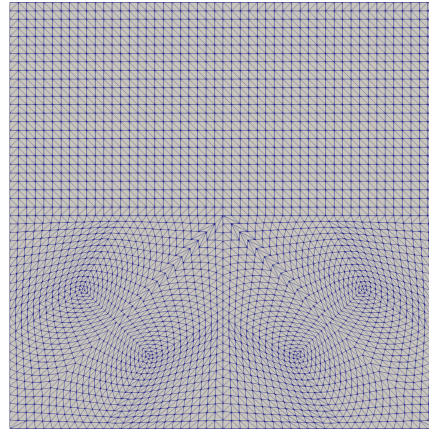
Figure 5.19.: Wall time for different mesh sizes

For this case, we select a heat exchanger with 3 fluids going through it. The conceptual image of the author was of a printed circuit heat exchanger (PCHE) with a hot and cold side, plus what is called a “guard” circuit in between. For a PCHE, the guard circuit stays in between the hot and cold side and prevents ingress of the fluid of one into another in case of channel rupture. For a reactor, it could be used for a decay heat removal system or some other safety-related task, such as melting of molten salt that solidified in one of the circuits.

For such a case, 4 meshes are necessary: one for each fluid and one for the overlapping solid. The mesh for the hot, cold and solid are in figure 5.20a, and the mesh for the guard circuit is in figure 5.20b. The boundary conditions are given in table 5.4 and flow is assumed to be laminar.



(a) Mesh of the hot and cold side, and solid



(b) Mesh of the guard side

Figure 5.20.: Triple circuit HEX meshes

Figures in 5.21 show the temperature results for the 3 fluids interacting through the energy equation of the solid, and figure 5.22 shows the temperature of the solid of the

Table 5.4.: Boundary conditions of the triple heat exchanger

Circuit	Field	Walls	Inlet	Outlet
Hot side	Pressure	zero gradient	zero gradient	1 bar
	Velocity	slip	$(0.2, 0, 0) \text{ m s}^{-1}$	zero gradient
	Energy	zero gradient	600 K	zero gradient
Cold side	Pressure	zero gradient	zero gradient	1 bar
	Velocity	slip	$(-0.2, 0, 0) \text{ m s}^{-1}$	zero gradient
	Energy	zero gradient	400 K	zero gradient
Guard	Pressure	zero gradient	zero gradient	1 bar
	Velocity	slip	$(0, -0.2, 0) \text{ m s}^{-1}$	zero gradient
	Energy	zero gradient	400 K	zero gradient
Solid	Energy	zero gradient		

heat exchanger acting as interface material.

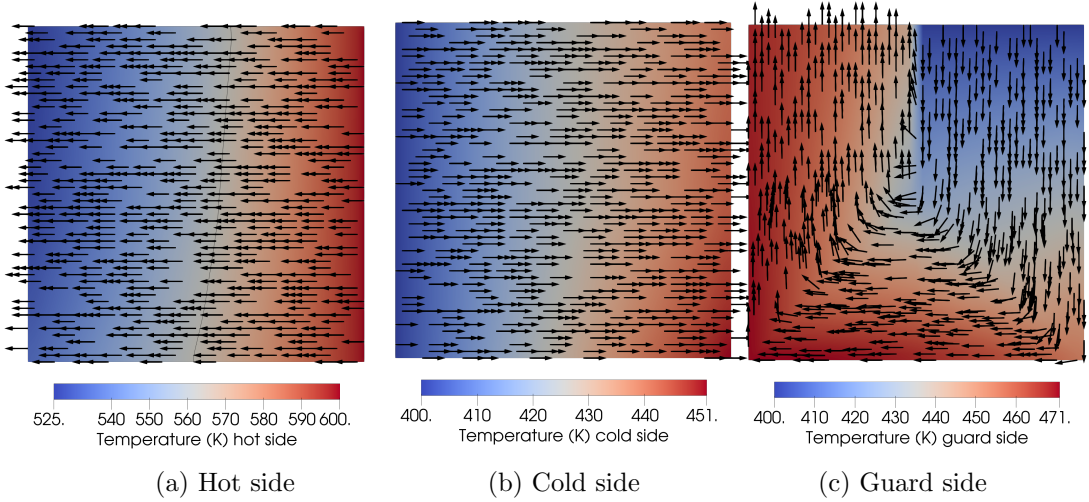


Figure 5.21.: Conceptual PCHE with 3 fluids

Integrating enthalpy values at inlet and outlet of different circuits allow us to form table 5.5, where we can check if energy is being properly conserved. We can see in the table that the total difference is very small, indicating that the energy equation is working properly. The difference can be made smaller by enforcing a tighter convergence criterion.

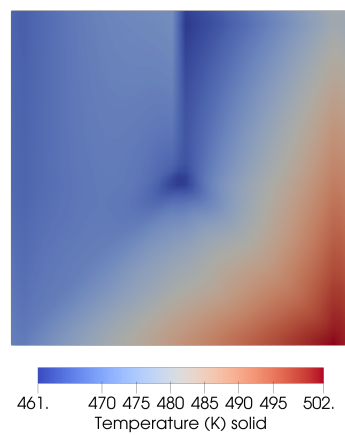


Figure 5.22.: Solid in the PCHE

Table 5.5.: Energy balances in the triple heat exchanger

Circuit	Enthalpy (J)		
	Inlet	Outlet	Difference
Hot side	1.85×10^6	1.62×10^6	-2.22×10^5
Cold side	1.23×10^6	1.37×10^6	1.35×10^5
Guard	6.15×10^5	7.03×10^6	8.76×10^4
Total			5.29×10^2

5.4. Discussion

From the exercises, we can see that ATARI performs well within the limited scope that was verified. Code verification in a broad scope of models is a very challenging task, demanding investment of time and human resources that is beyond what any individual thesis can achieve.

While the Stefan problem is a very simplified problem, the level of agreement achieved is a solid first step towards a more sophisticated verification of the solver. For the approximately 1D flow characteristics of a narrow pipe such as presented in chapter 6, this verification is satisfactory and brings enough evidence of correctness to cautiously proceed.

Verification exercises should be performed with very controlled steps whenever the scope is to be broadened. In this sense, the 1-phase and 2-phase Stefan problems are complimentary. The former can be implemented with minimal requirements in code capability besides the appropriate source. The later requires the availability of the error function and programmable boundary conditions, which is an added level of complexity. This allows knowing with relatively high confidence that the source is working for example, before evaluating special functions and boundary conditions. A step-wise approach greatly helps narrowing on the problem when something **does not work**, which can be very frustrating and time consuming to diagnose.

More generally, both problems can be formulated with the entire domain in one phase, where the boundary condition causes the domain to transition to the other phase. The case where the domain is solidified and caused to melt by diffusing energy into the domain from the boundary condition was picked for no particular reason, except that a choice had to be made. The alternative, where the domain is liquid and caused to solidify by diffusing energy out of the domain to the boundary condition at lower temperature is mathematically the same.

The “recirculating cavity” problem proposed is another verification exercise that is severely limited in scope. It is limited to one-way coupling and certainly there are more models that it does not verify than models it does; but that is how it should be at this point. Future development should progressively broaden the scope and relax assumptions one at a time.

Using this simple case to implement and verify a supporting symbolic mathematics package should prove highly beneficial, and is most likely a requirement for more complicated cases. Once this is done, the need to manage complexity in the source essentially disappears. In that case, the exact opposite is desirable. The source should become very complex so that it becomes numerically challenging for the code to reproduce the reference solution, with the intent of emphasising imperfections. Always keep in

mind that the objective of verification is to reveal problems related to numerical methods and discretisation schemes.

In order to use the method of manufactured solutions for code verification, it is necessary that the code allows an arbitrary source to be inserted into the equations. This capability is present in the OpenFOAM library through the “fvOptions” functionality, which ATARI inherits. Likewise, each boundary condition for velocity is a non-uniform Dirichlet boundary condition that is relatively complex to implement, requiring the use of the “codedFixedValue” boundary condition. Implementing this verification case requires very advanced level of understanding of both functionalities. Additional information and templates covering this usage can be found in the test case files at the author’s GitHub repository².

The neutron diffusion model used a rather unusual boundary condition, setting a fixed value of neutron flux at 0. While unusual, it must be emphasised that at this stage the main targets of this verification are the solvers, not the boundary conditions available. Verifying a Marshak or albedo boundary condition is an additional layer of complexity that should be performed in a future development.

Finally, the multi-mesh capability of the code was rudimentarily tested using energy balances of the different circuits, and was found to be correct in a steady state. A future development in the short term should include the energy that is transferred into the solid, which will require an analysis of balance in time as well. The final objective should be to move to more robust methods of verification as was used in the other exercises of this chapter. Energy balances are a very rough method of verification, and ideally an analytical solution should be developed to allow studies of order of accuracy.

Regardless of the limited scope achieved, rigorous code verification is a task that brings significant benefits in understanding the exercised models. If one’s intention is to reach a deep understanding of the equations and all its parameters, such as the neutron diffusion, performing a verification task is definitely a good approach. It is just important to relinquish the urge to maintain physical realism. In a normal analysis, physical meaning is obviously the objective, however during verification one must understand that the objective is different. Realism must be set aside and give way to other needs.

²<https://github.com/deOliveira-R/recirculatingCavity>

6. Application

Souvenez-vous qu'il n'existe pas de sciences appliquées, mais seulement des applications de la science.

(Louis Pasteur)

After a thorough redesign and verification of basic functionalities of the code, it is finally put to use in some specific problems where some useful conclusions can be drawn.

A code-to-code benchmark performed in the framework of the SAMOFAR project, which shall be named “Nuclear Cavity benchmark” in this thesis, is presented in section 6.1. It consists of a sequence of problems of gradually increasing complexity in order to test the coupling between different parts of the code. Despite not being a verification exercise, coupling is an aspect of the algorithm that is hard to evaluate. While a benchmark is incapable of producing evidence of correctness, it is capable of producing evidence of incorrectness, which is useful in its own sense. Therefore this exercise has a dual purpose of gauging how aligned the different participating codes are in terms of models, and to indicate mistakes in coupling schemes.

The design proposed by the SAMOFAR project, the Molten Salt Fast Reactor (MSFR), is analysed in steady-state conditions in section 6.2. Performed in support of the project Work Package 4 [Cammi et al., 2019], it should allow an overview of the reference design, and reveal some deficiencies in the concept and important missing data.

In section 6.3, a square pipe with pump and heat exchanger, based roughly on the heat exchanger section of the MSFR, is solidified due to excessive heat extraction. This is a first approach to the study of the solidification phenomena in a heat exchanger, with the objective of identifying important missing data and overall expected transient behaviour.

Finally, a conceptual design of a fast spectrum MSR using chloride salt and flow baffles is proposed in section 6.4. Most fast spectrum MSR concepts are based on a clear core with no structural materials in the flow field, resulting in a very turbulent flow. As a consequence, analyses of these cores are very reliant on specialised codes based on CFD, such as ATARI. We will explore the drawbacks of these characteristics, and how, with a simple change, it is possible to design cores that are “well behaved”.

Due to the diversity of the topics in the chapter, each section will have its own discussion in order to call the readers attention to the most important lessons learnt.

6.1. Nuclear cavity benchmark

The benchmark was performed as a task for the SAMOFAR project [Tiberga, de Oliveira, et al., 2020; Lathouwers et al., 2018] in cooperation with Politecnico di Milano (PoliMi), and Technische Universiteit Delft (TUDelft). Originally it was performed using GeN-Foam, but repeated once ATARI reached feature parity in regards to the capabilities required.

Like the recirculating cavity in chapter 5, it is also inspired by the 2D lid-driven cavity case, which is part of the OpenFOAM tutorials. Its domain is a square 2×2 meters containing a fissile fluid inside, shown on figure 6.1, which is a very simple geometry that minimizes the influence of the domain on the interpretation of results.

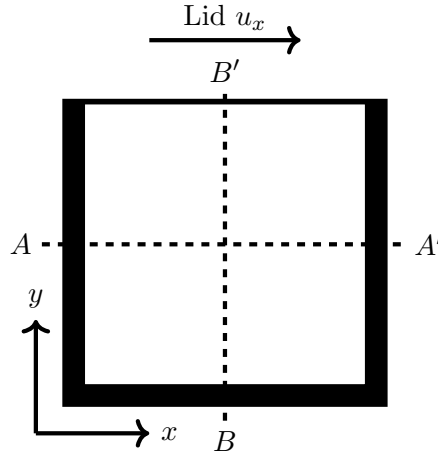


Figure 6.1.: Nuclear cavity domain

Dirichlet boundary conditions are applied for fluid dynamics, where left, right and bottom boundary conditions are “no-slip” type and the top lid moves to the right with prescribed velocity u_x . For energy transport, the boundaries of the cavity are insulating (i.e., Zero-gradient Neumann boundary conditions), with uniform volumetric heat extraction from the fluid shown in equation 6.1, where $UA = 10^6 \text{ W m}^{-3} \text{ K}^{-1}$ is volumetric heat transfer coefficient¹, and $T_{\text{ext}} = 900 \text{ K}$. This volumetric heat sink has been implemented directly using OpenFOAM’s “fvOptions” functionality, since it will be varied later. Vacuum boundary conditions (i.e., no incoming neutron current) are

¹ UA has the same concept and units of the product in equation 4.10 or 4.12

applied on all sides for neutron diffusion.

$$Q = U A (T_f - T_{\text{ext}}) \quad (6.1)$$

For this benchmark, all codes have in common the use of incompressible Navier-Stokes equations, laminar flow model, and use of the Boussinesq approximation to avoid a problem of mass conservation in a closed domain with incompressible fluid. In this approximation, changes of density are neglected in all terms of the conservation equations, except for the buoyancy force in the conservation of momentum and parametrization of nuclear data in the neutronics model.

The codes differ in neutronics capabilities and formulation of the discretisation scheme. Whereas ATARI and PoliMi's code inherit the finite volume formulation from OpenFOAM and use a diffusion approximation for neutronics, TUDelft's code is developed entirely in-house, uses a finite element formulation and a discrete ordinates (S_n) neutronics solver.

The benchmark is composed of multiple steps of incremental complexity, where an overview of the models exercised in each step can be seen on table 6.1. It starts with isolated single-physics steps of fluid mechanics and neutronics and progresses to coupled multi-physics problems.

Table 6.1.: Nuclear cavity benchmark steps

Step	Fluid dynamics			Neutronics	
	Momentum	Energy	Buoyancy	Neutron transport	DNP transport
0.1	Yes	No	No	No	No
0.2	No	No	No	Yes	No
0.3	Imposed	Yes	No	Imposed	No
1.1	Imposed	No	No	Yes	Yes
1.3	Imposed	Yes	No	Yes	Yes
1.5	Yes	Yes	Yes	Yes	Yes
1.7	Yes	Yes	Yes	Yes	Yes
2.0	Yes	Yes	Yes	Yes	Yes

Plots over dashed lines AA' and BB', crossing the middle of the cavity, are used as quantitative comparison with other institutions participating in the benchmark.

6.1.1. Step 0.1: Fluid dynamics

The benchmark starts pure fluid dynamics problem with all other solvers disabled. In this first step, the top lid is moving at $u_x = 0.5 \text{ m s}^{-1}$ inducing a recirculating flow field inside the cavity. Figure 6.2 shows the velocity magnitude of this field, with arrow glyphs indicating the flow direction. As figure 6.3 shows, the results of the participating institutions are in excellent agreement.

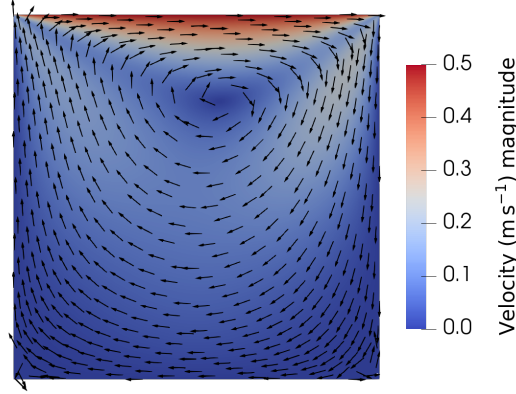


Figure 6.2.: Step 0.1 velocity field

6.1.2. Step 0.2: Neutron diffusion

On the second step, the benchmark investigates a pure neutron diffusion problem. Figure 6.4 shows that the results from the participants match very closely, and that neutron flux reaches its maximum value at the centre of the cavity. The chopped cosine volumetric fission rate $\sum_{j=0}^J \Sigma_{f,j} \phi_j$ along both axes is shown in figure 6.5.

Of particular importance is that TUDelft's results from discrete ordinates S_6 with P_3 scattering and S_2 with P_1 scattering are a closely matched, and to know that S_2P_1 is essentially equivalent to the diffusion approximation with P_1 scattering transport correction. This match is an important predicted aspect of homogeneous reactors, as discussed at the end of chapter 3.

6.1.3. Step 0.3: Energy transport

In the third and final single physics step, the velocity field of step 0.1 (figure 6.2) and the neutron flux from step 0.2 (figure 6.4) are mapped to the respective fields of the problem. The objective is investigating the resulting temperature field simulated by a pure energy transport problem, where fluid dynamics and neutronics are imposed

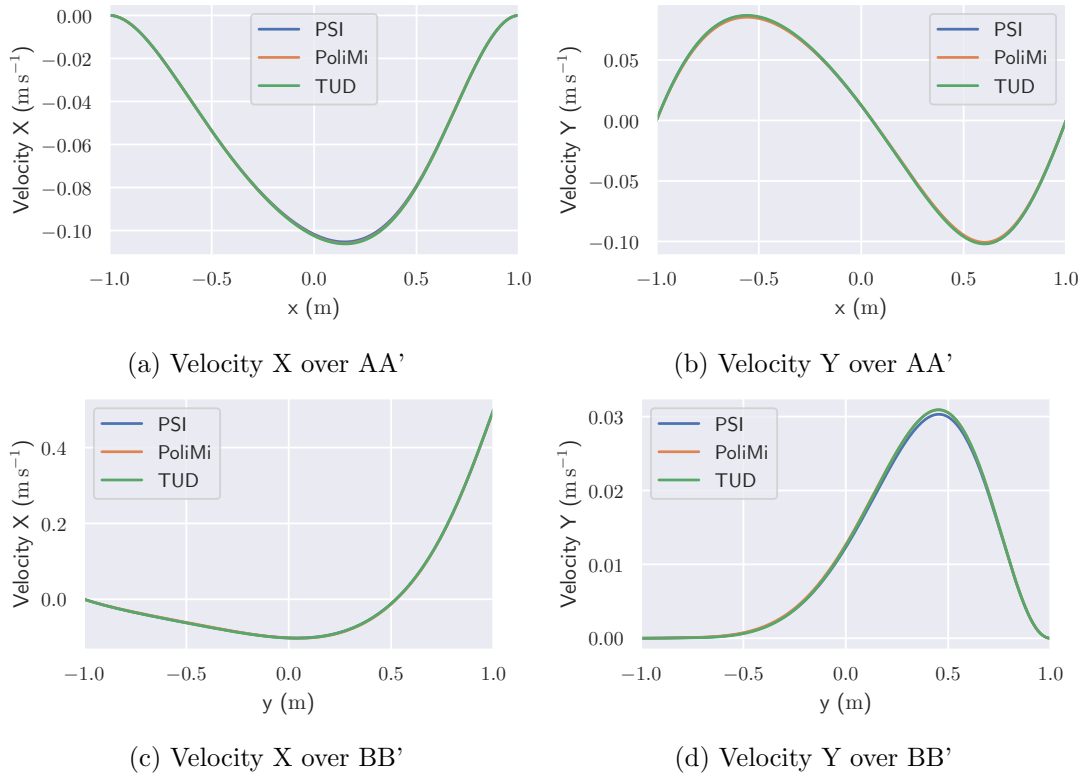


Figure 6.3.: Step 0.1 comparison of results between participants

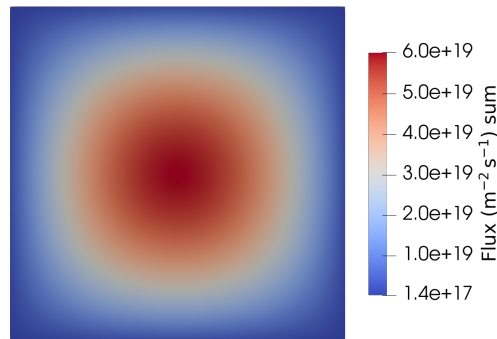


Figure 6.4.: Step 0.2 flux summed over energy

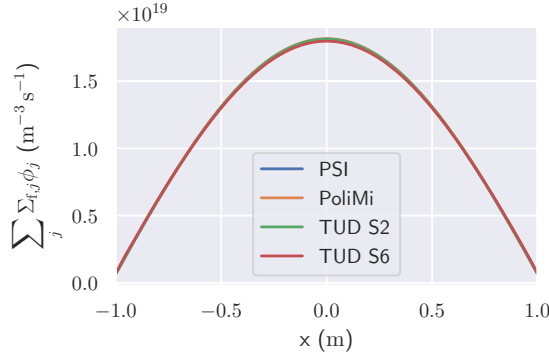


Figure 6.5.: Step 0.2 comparison results between participants

but not solved for. Figure 6.6 shows that the energy in the cavity is distributed as a results of the imposed velocity field, and figure 6.7 the excellent agreement between institutions.

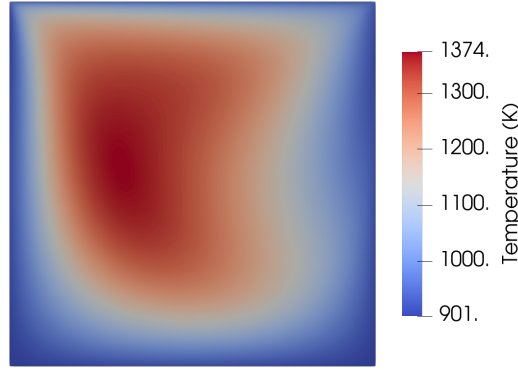


Figure 6.6.: Step 0.3 temperature field

6.1.4. Step 1.1: Neutron and DNP transport

In this step, neutron and DNP transport are solved under an imposed flow field. Similar to step 0.3, DNP transport is also a scalar transport problem, where the transported quantities are precursors concentrations. As can be seen on figure 6.8, the quantities are successfully transported and the longest lived group 1 spreads around the flow region while the shortest lived group 8 barely moves from its origin. Group 5 shows an intermediate behaviour.

Figure 6.9 shows the delayed neutron source $S_{\text{dnp}} = \sum_{g=1}^G \lambda_g C_g(\mathbf{r}, t)$ over the comparison lines. All institutions show closely matched results.

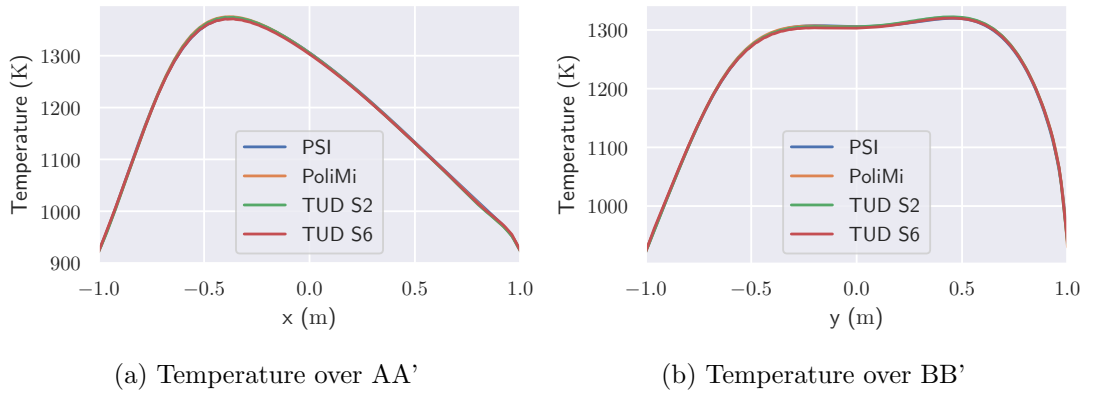


Figure 6.7.: Step 0.3 comparison results between participants

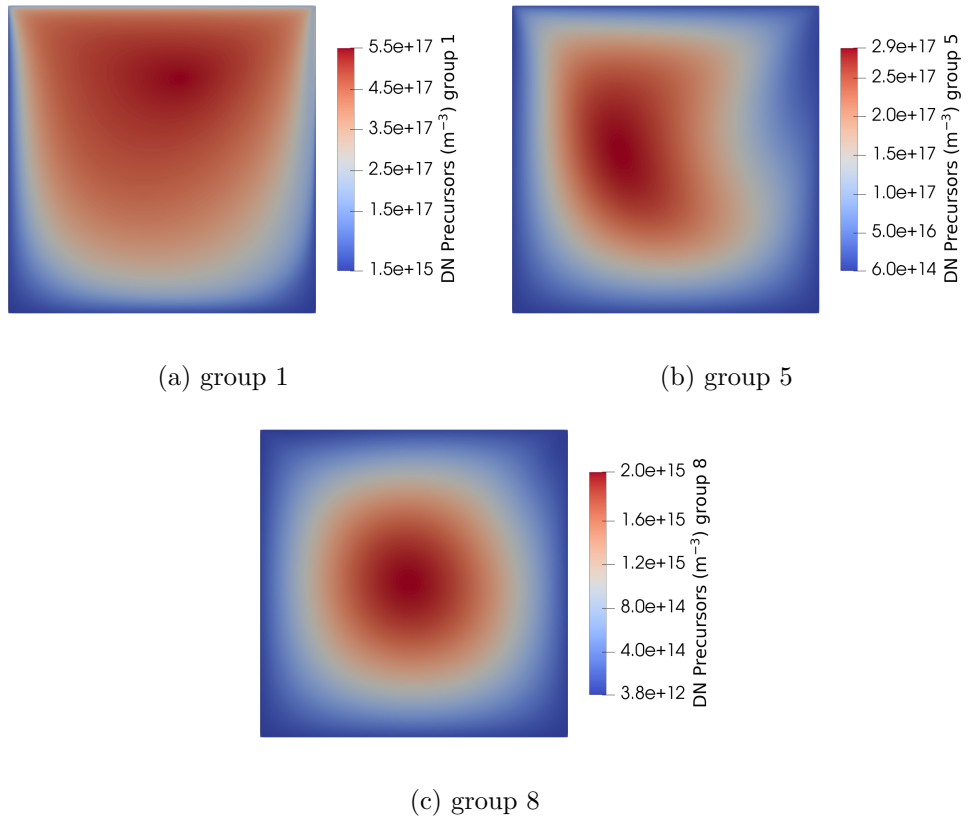


Figure 6.8.: Step 1.1 delayed neutron precursors concentration fields

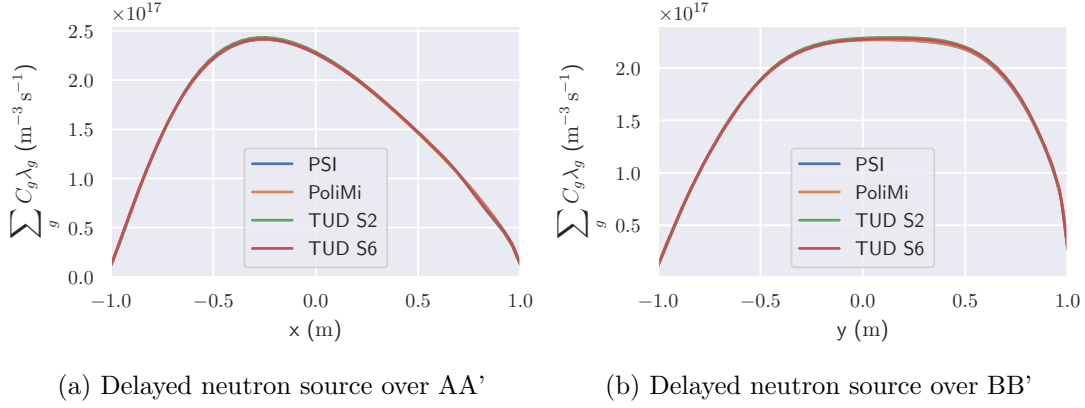


Figure 6.9.: Step 1.1 comparison of results between participants

6.1.5. Step 1.3: Power coupling

Influence of a non-uniform density field to neutronics is investigated in this step, where neutronics and energy transport models are enabled.

Figures 6.11 show the excellent agreement between institutions, where figure 6.11c shows the difference in fission rate density compared to step 0.2.

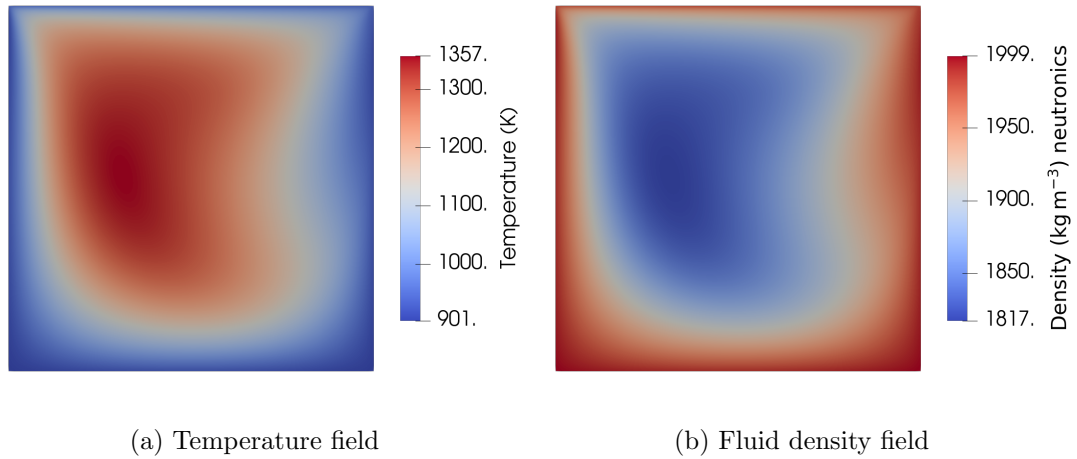


Figure 6.10.: Step 1.3 fluid density and temperature fields

This step identified a mistake in the original GeN-Foam application (from 2017, not current one) when mapping density from fluid dynamics to neutronics in the case of incompressible flow. Before correcting the field mapping routine, figure 6.10b would show a uniform field with initial density value, despite the temperature field shown in figure 6.10a. After correcting the issue, the density field is properly mapped.

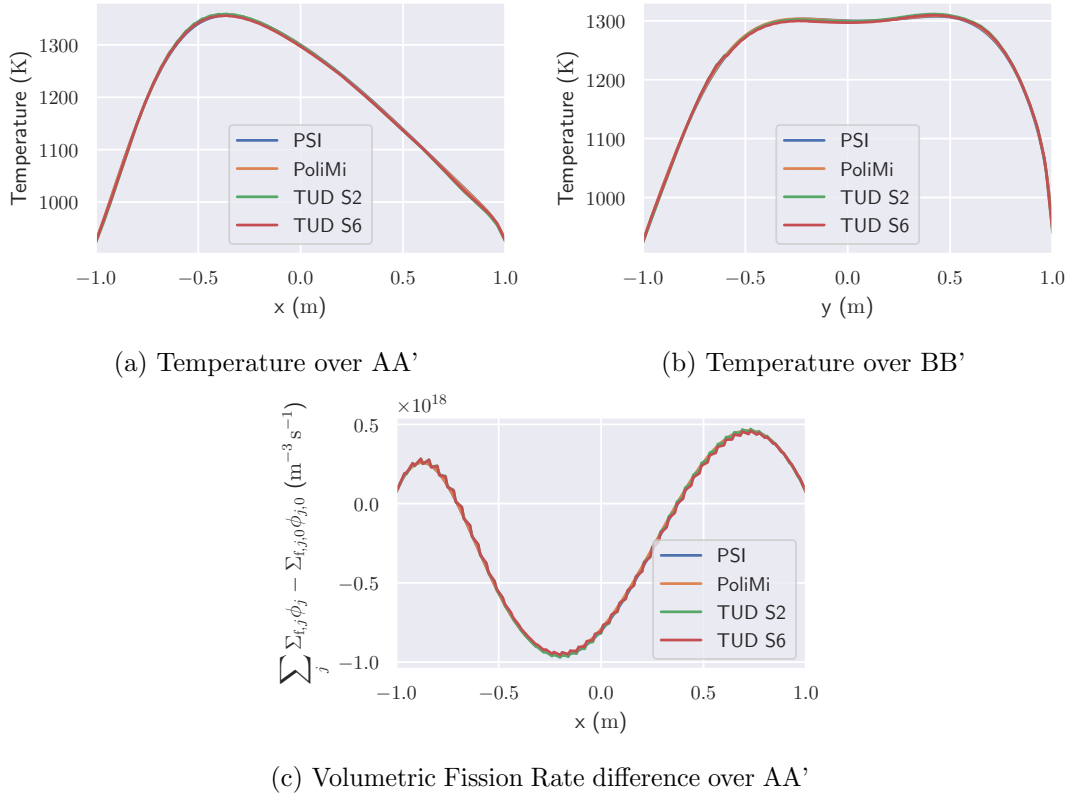


Figure 6.11.: Step 1.3 comparison of results between participants

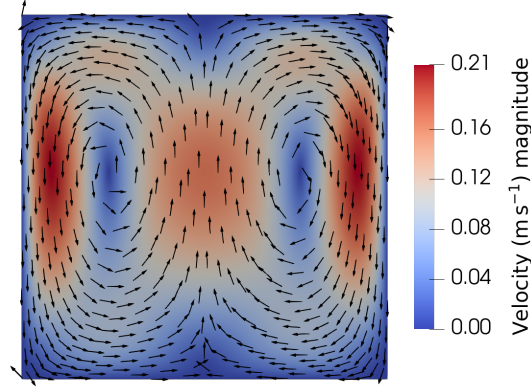


Figure 6.12.: Step 1.5 velocity magnitude field

6.1.6. Step 1.5: Buoyancy

Natural circulation is the focus of this step. The top lid is static ($u_x = 0 \text{ m s}^{-1}$), and Earth's average gravity acceleration is imposed in the -Y direction, resulting in an axisymmetric flow response (figure 6.12) due to the strongly centred power production. The velocity direction, indicated by arrow glyphs on the same figure, also show the expected flow behaviour of natural circulation in the domain.

All institutions agree very well as shown on figure 6.13, as expected from the result of previous steps.

6.1.7. Step 1.7: Full coupling

Finally, the last step is composed of a series of 36 steady-state cases where all solvers are enabled. In this set of cases, power is increased from 0 to 1 GW in intervals of 0.2 GW and lid velocity is increased from 0 to 0.5 m/s in intervals of 0.1 m/s. The observable quantity is the reactivity change $\rho - \rho_{0.2}$, where $\rho_{0.2}$ is the reactivity of step 0.2 used as a reference.

The comparison between partners will be put aside in this step, since there are too many cases and the only conclusion from the comparison is that partners are in very high agreement [Tiberga, de Oliveira, et al., 2020]. Instead, the values obtained by the author will be shown in figures in order to discuss trends.

In figure 6.14, each plot is an equal lid velocity case, where only power is varying. Reactivity decreases with power due to decrease in density of the fluid and, to a smaller degree, due to the resulting buoyancy effects on the velocity field. The decrement is almost linear, with a deviation only for zero power cases where a uniform density field is

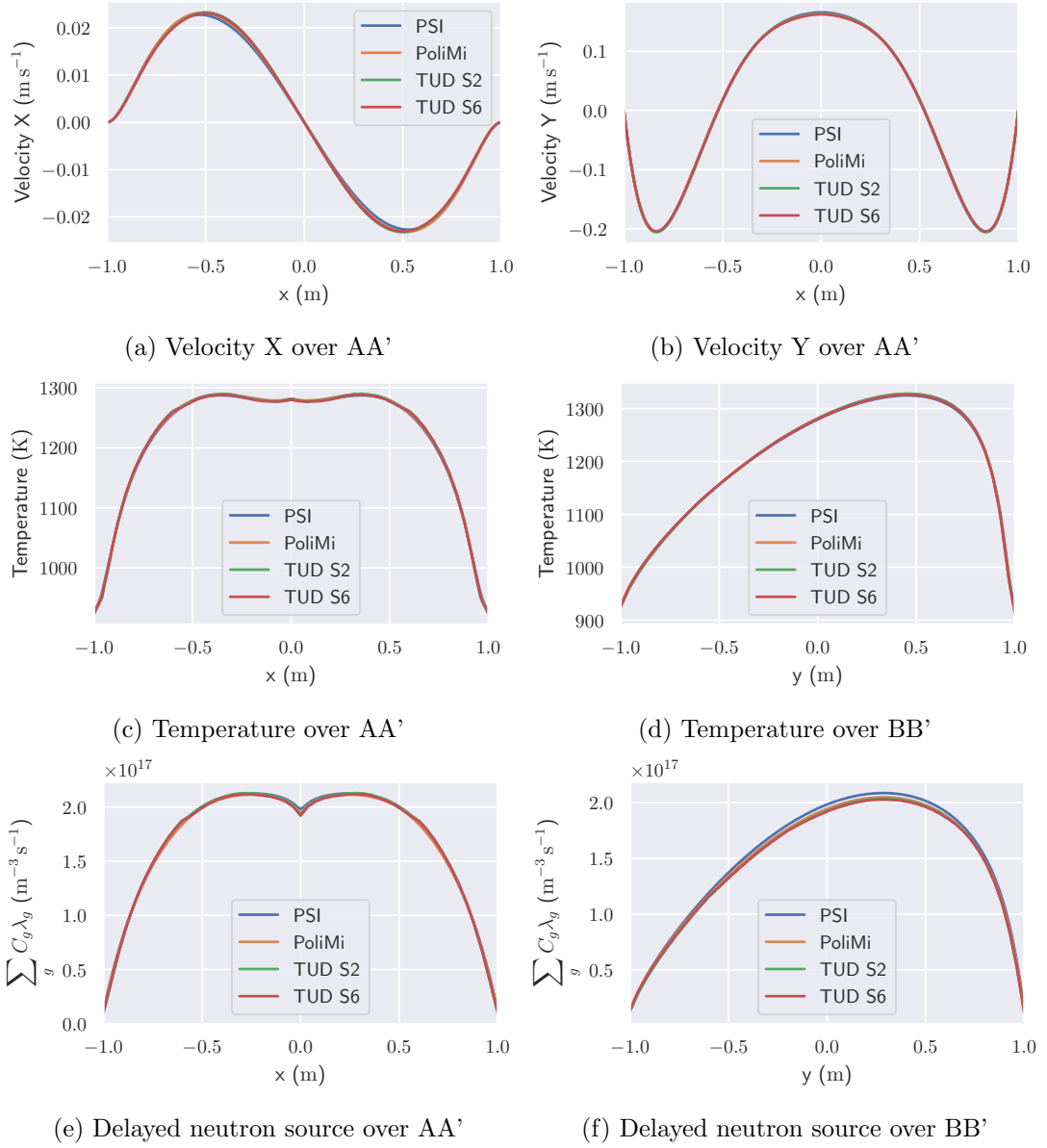


Figure 6.13.: Step 1.5 comparison of results between participants

present in the cavity, in which case reactivity decrement is solely due to DNP removal from the region of high importance for neutronics.

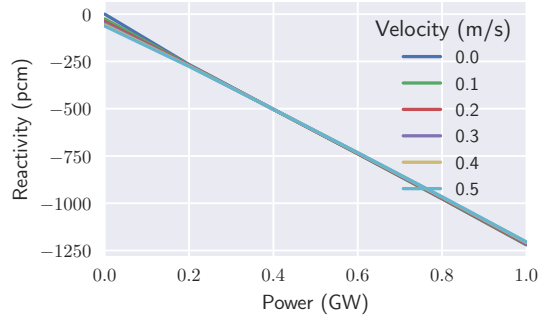


Figure 6.14.: Step 1.7 full coupling equal velocity plots

Plots of equal power are shown in figure 6.15, where only lid velocity is varying. It is possible to notice that for low power cases, reactivity decreases with increasing lid velocity as previously explained. For high power cases, the situation reverses and an increment in lid velocity also increases reactivity. At high power cases, natural circulation is significant and the imposed lid velocity changes the recirculation pattern, resulting in more DNP in the centre of the cavity. At 0.4 GW the intermediate state can be seen where an increase in lid velocity initially removes DNP from the centre cavity, until a certain point where an additional increment in lid velocity causes DNP to recirculate back into the centre of the cavity. However, reactivity varies only by a fraction of pcm in this case, therefore, changes are very subtle.

6.1.8. Step 2.0: Transient behaviour

In the final step, the volumetric heat transfer coefficient is varied by a sine wave of amplitude 10% of its value (i.e., 10^5), according to equation 6.2. A total of 7 cases are simulated with sine wave periods of 1.25, 2.5, 5, 10, 20, 40, and 80 seconds. The intention is to induce a power oscillation that, depending on the sine wave period, follows heat sink oscillation with a certain gain, given by equation 6.3, and phase-shift. The gain and phase-shift can be analysed in the frequency domain as shown in figures 6.16 and 6.17, which are called Bode diagrams.

A general good agreement between partners can be seen, always limited to a 5% discrepancy. We can see in the figures that at low frequencies, power follows the sine wave closely (i.e., almost no phase-shift) and kinetics is driven by delayed neutron precursors since these have the time required to find equilibrium in a quasi-static manner, resulting in a gain of approximately 1. At high frequencies, a significant phase-shift develops, up to almost 90° , and kinetics is driven by prompt neutrons causing gain to

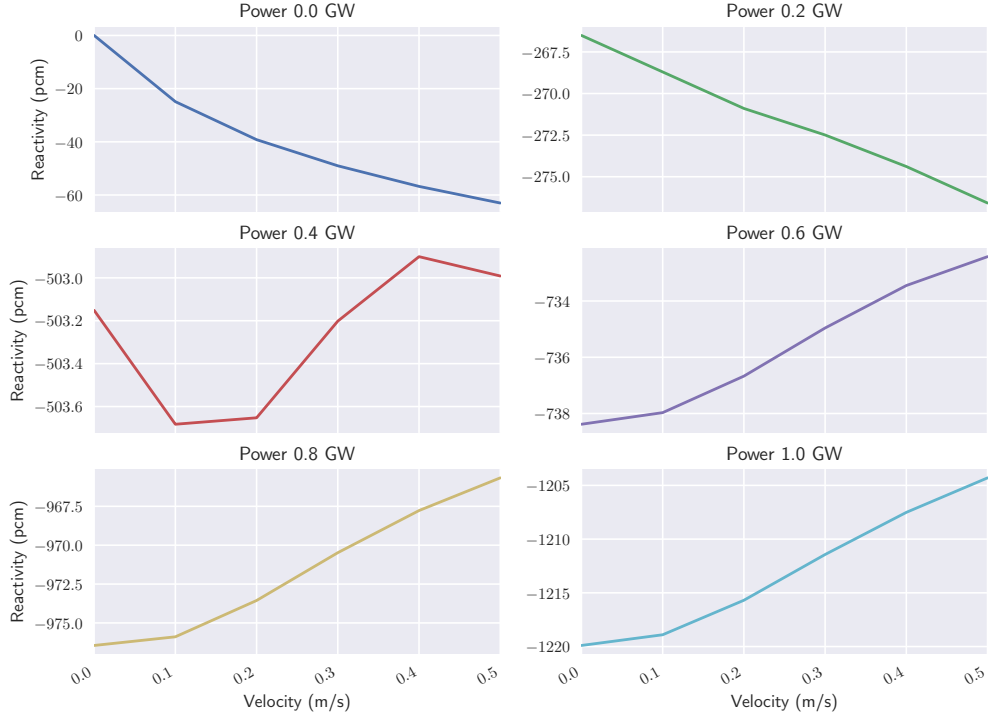


Figure 6.15.: Step 1.7 full coupling equal power plots

drop significantly due to the suppressed influence of delayed neutrons.

$$(UA)_{\text{eff}} = UA \left(1 + 10\% \sin \left(\frac{2\pi}{T} \right) \right) \quad (6.2)$$

$$G_P = 10 \left(\frac{P_{\text{max}} - P_{\text{ref}}}{P_{\text{ref}}} \right) = 10 \left(\frac{P_{\text{max}}}{P_{\text{ref}}} - 1 \right) \quad (6.3)$$

6.1.9. Discussion

The agreement between the participants of the benchmark was considered excellent in general. The codes involved are essentially equivalent from the perspective of accuracy, which is what the benchmark intended to demonstrate. Criteria that has not been investigated or reported include rate of convergence and execution time for each code, a result that would have been quite valuable but the partners did not think of it at the time.

The agreement between results of step 0.2 reinforces the argument made in chapter 3. Homogeneous systems will typically demonstrate small anisotropy and the diffusion

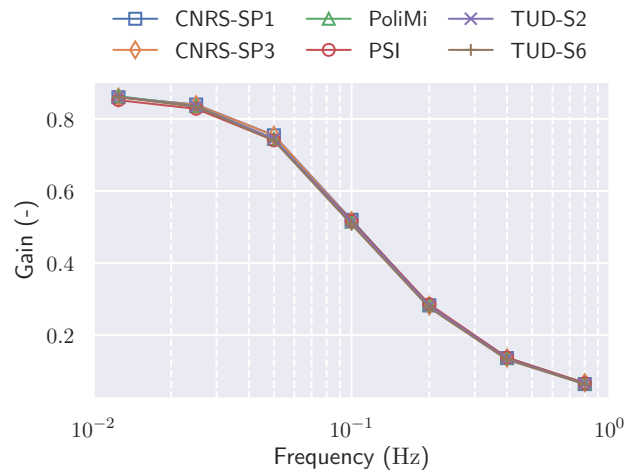


Figure 6.16.: Step 2.0 power gain Bode diagram

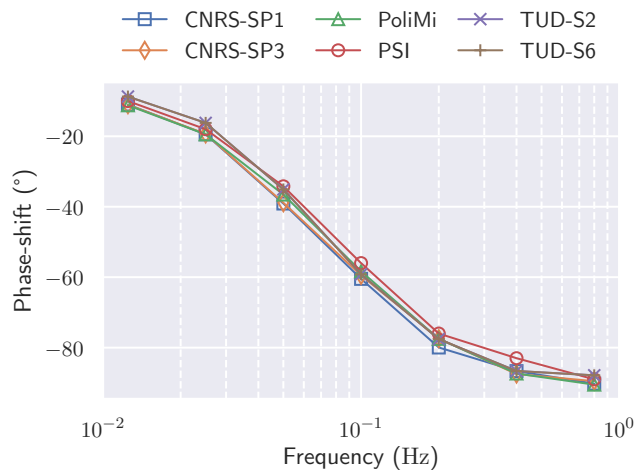


Figure 6.17.: Step 2.0 phase-shift Bode diagram

approximation will perform remarkably well. Results from S_6P_3 shows essentially no discrepancy compared to the use of transport-corrected diffusion. In addition, let us consider that this benchmark was limited to laminar flow; if turbulence is introduced, the uncertainty embedded in a turbulence model is such that the uncertainty in the neutronics model is comparatively negligible. In the next section the analysis of the MSFR design will be presented, where this exact observation will come into play in a more realistic case.

The issue discovered in step 1.3, section 6.1.5 is a good example of utility and limits of benchmarks. It produced evidence of incorrectness, and after fixing that evidence was gone; which does not mean that the code is correct, just not obviously incorrect. To produce evidence of correctness, verification techniques are required. The reason is very simple: for a question that has an objective answer, such as a mathematical one, there is essentially an unlimited number of ways of being wrong, but only 1, or at most a few, ways of being right. Since one cannot test, or even predict, the uncountable ways of being wrong, there is no amount of benchmarking, or no lack of evidence of incorrectness, that will ever prove one to be correct. One can only show correctness by knowing precisely what correct means, and showing evidence of matching it.

The nuclear cavity benchmark was performed very early during studies, when GeN-Foam was still used for the thesis, and its benefits were clear. For this reason, a script has been made that enables automated execution of all steps reported, with post-processing scripts that allowed automatic generation of every figure presented. Together, these were incorporated into a software quality assurance procedure in the form of a series of regression tests. Such a gradual benchmark proved remarkably useful during development of ATARI later on.

This logic of using a benchmark as a regression test cannot go very far though, as at some point the costs outweigh the benefits. The execution of a single step of the benchmark took hours and complete execution of the benchmark took days, whereas the recirculating cavity from chapter 5 took a minute. This means that, for practical reasons, the benchmark was very rarely fully executed. Most of the time, its execution was limited to the step or steps that would likely narrow down changes with numerical consequences. In other words, it took hours to narrow down a change and evaluate if it was reasonable to proceed further, without even knowing if it was strictly correct or not, whereas a case that demonstrated correctness, even if it did so in a very limited aspect, did so in a minute — such is the benefit of knowing the precise meaning of correctness.

At any time frame longer than a single thesis, the investment of devising a set of verification cases, which might individually have a very limited scope but a combined wider one, will quickly payoff. However, it is undeniable that the execution of the benchmark and identification of flawed logic saved what would be a significant amount

of time hunting for issues later on during this thesis. It may not show correctness, but shows that a change had numerical repercussions, which helped narrowing down the cause so that at least the likelihood of some flawed logic being introduced could be evaluated. This is far from perfect, but it is better than complete ignorance, which critically is a very low standard and shows just how much work is still to be done.

6.2. Molten Salt Fast Reactor

The MSFR studied by the SAMOFAR project is a 3 GW reactor concept consisting of an integrated circuit where 16 modules, each containing blanket, shielding, and heat exchanger, are immersed into a cylindrical vessel, shown in figure 1.2 at the introduction.

In figure 6.18, the geometry studied in this thesis is presented, which was inherited from previous works [Laureau et al., 2017; Rouch et al., 2014; Aufiero, 2014]. The original STL file was provided by another institution but was too difficult to mesh due to several issues that made it not leak-tight (e.g., holes, disconnected vertices, and internal surfaces). A new CAD file was built by the author based on that damaged geometry, with assured symmetry and leak-tightness, allowing it to be meshed without issues.

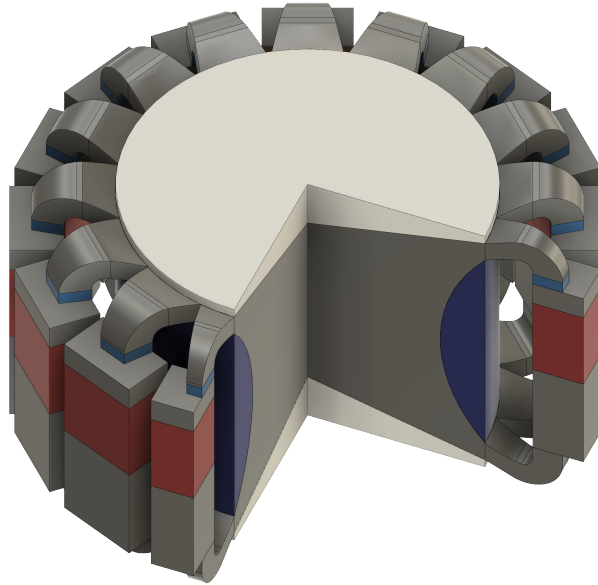


Figure 6.18.: MSFR geometry cutaway

In figure 6.18, the molten salt containing a fissile nuclide occupies the dark grey volume, where it is allowed to circulate into a pump in the blue volume, through a heat exchanger in the red volume, and recirculates back into the core region where fission takes place. A radial blanket is present around the core in purple, and further than that but not shown would be a radial reflector and absorber. The axial neutron reflectors are shown in white.

The curved walls in the core were intended to prevent recirculations and improve thermal profile at the walls [Rouch et al., 2014; Brovchenko et al., 2013]. Heat exchanger was first supposed to take axially all the volume between the expansion below the pump, and contraction. Currently, it is conceptually limited to the red part in order to locate

the thermal center of the heat sink above the mid-height of the core (i.e.: heat source thermal center) and allow natural circulation; at least in theory, but its size will not make a difference in this analysis. The pump is located above the heat exchanger, and the lower pressure region created at the pump inlet offers an advantageous place for degassing purposes in order to avoid bubble formation inside the heat exchanger [Allibert et al., 2017, p. 20]. Both the pump and the heat exchanger design are still subjects of investigation by the project.

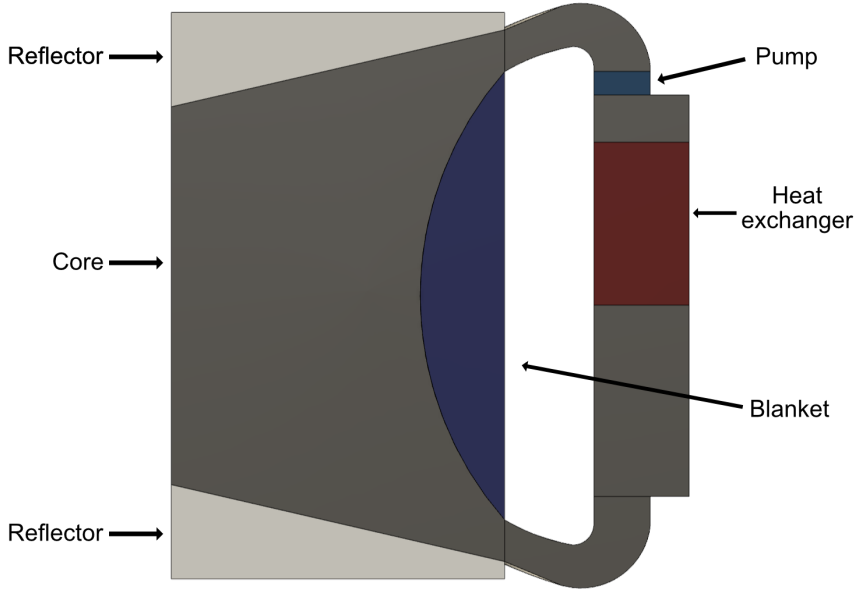


Figure 6.19.: MSFR cutaway view with tagged regions.

A $\frac{1}{16}$ azimuthal sector of the core **comprised only of the fuel circuit** (i.e., dark grey, blue and red on figure 6.19) was meshed using the OpenFOAM mesher, “snappyHexMesh”, maintaining symmetry on the wedges. The resulted in a mesh of 99512 cells shown in figure 6.20, composed of approximately 75% hexahedra, which is the basic starting polyhedra of the mesher, with the 25% a mix of prisms and various other polyhedra. A boundary layer 3 cells is present at the walls, which with the help of wall-functions, was found to be sufficient for the convergence of fields of interest. Convergence was monitored through the pressure difference between inlet and outlet.

The properties of salt composition 2 at 973 K described in the SAMOFAR deliverable [Allibert et al., 2017, p. 9] are used, shown on table 6.2 below. The influence of density variations on the fluid flow are modelled using the Boussinesq approximation.

Transport corrected diffusion coefficients and homogenised cross-sections for the fuel salt are obtained in 6 energy groups from a Serpent 2 model using the JEFF-3.1.1 library; credit is given to Politecnico di Milano for performing the Serpent analysis and providing the data that is reproduced in appendix B. A single value of a reaction cross-section in

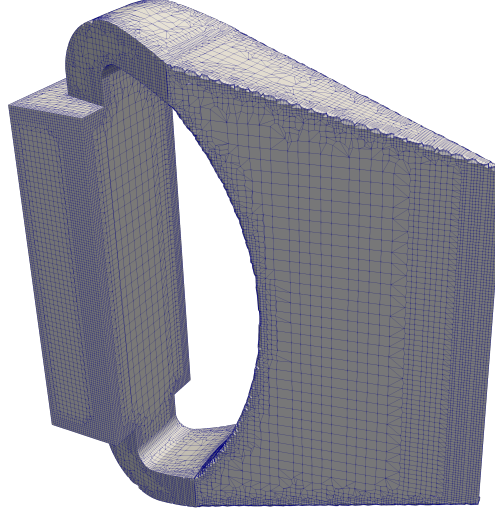


Figure 6.20.: Meshed domain comprising a fuel circuit sector

Table 6.2.: MSFR fluid properties

Parameter	Equation of State	Value at 973 K	Unit
Density	$(5108 - 0.8234 T)$	4306.7	kg m^{-3}
Dynamic viscosity	$\left(6.187 \cdot 10^{-4} \exp\left(\frac{772.2}{T-765.2}\right)\right)$	$2.536 \cdot 10^{-4}$	Pas
Specific heat capacity	-	1594	$\text{J kg}^{-1} \text{K}^{-1}$
Thermal conductivity	-	1.7	$\text{W m}^{-1} \text{K}^{-1}$

the core is considered a reasonable approximation at the moment — there is no reason to believe that the neutron spectrum varies significantly in the bulk of the core and we can expect much larger sources of uncertainty from the fluid dynamics modelling. More importantly though, it is known that the power production will be in the centre of the cavity, and that will also be the region of high importance for the results of interest. Unless our interests shift to regions of lower neutronics importance, such as the intention to accurately model blanket or reflector phenomena for some special reason, there should be no need for finer detail.

A particular parametrization model is used, first introduced by Aufiero [2014] and shown in equation 6.4. In this equation, the density ratio is obtained from the Boussinesq approximation and the α coefficients are obtained from parametrising the data for temperature perturbations. The reference cross-sections are calculated at $T_0 = 900$ K, and the coefficients from perturbations at 1200 K.

$$\Sigma(T) = \frac{\rho(T)}{\rho(T_{ref})} \left(\Sigma(T_0) + \alpha \log\left(\frac{T}{T_0}\right) \right) \quad (6.4)$$

The heat exchanger is modelled as a homogenised region of printed circuit heat exchanger (PCHE) and fuel salt, and the pump is modelled as a simple momentum source with downwards vector. The pump momentum source and heat exchanger friction factor need to be adjusted in tandem so that a volumetric flow rate of $4.5 \text{ m}^3 \text{ s}^{-1}$ with a pressure drop of 4 bar is obtained. An initial estimation of the friction factor can be obtained from the Darcy-Weisbach equation 2.45, taking an average fluid velocity of 1.14 m s^{-1} , hydraulic diameter $D_h = 1.22 \text{ mm}$, and a length $x = 0.69 \text{ m}$, however, it has to be calibrated during simulation. The friction factor is set up in such a way that cross-flow is not possible, as would be expected of typical PCHE. The fluid fraction γ_f is not used since it would result in unnecessary differences from the analysis performed by partner institutions.

$$\frac{dp}{dx} = -f_D \frac{\rho}{2D_h} \mathbf{u}^2 = -f_D \frac{\rho}{2D_h} \frac{\mathbf{u}_{h,f}^2}{\gamma_f^2} \quad (2.45 \text{ revisited})$$

A target temperature difference of 100 K across the heat exchanger is intended. To achieve it, an external temperature of 635°C is given. The external temperature of equation 4.12 is set up to the given value $T_{\text{ext}} = 908 \text{ K}$ (635°C), and the heat transfer coefficient is calibrated to obtain the correct ΔT .

$$Q_{s \rightarrow \text{ext}} = U_{\text{ext}} A_{s \rightarrow \text{ext}} (T_s - T_{\text{ext}}) \quad (4.12 \text{ revisited})$$

A symmetry boundary condition is applied on the surfaces of the wedges, making this

problem axisymmetric. This special boundary condition is applied for every quantity, leaving the boundary condition on the walls to be determined.

Fluid dynamics boundary conditions are taken as “no slip” on the walls, and “free slip” in the heat exchanger region. In the heat exchanger region, modelled as a homogenised region, it is unphysical to apply a boundary condition that models walls, since “there is no wall”; it has been homogenised into the volume and lost its shape. A friction relation, the Darcy-Weisbach in this case, is used to model momentum loss in this region. For the same reason, the k - ϵ turbulence model uses standard wall functions everywhere except the heat exchanger, where a “zero gradient” Neumann boundary condition is used. Pressure boundary condition is “zero gradient” on every wall due to choice of incompressible flow. Likewise, the walls are considered adiabatic and a “zero gradient” boundary condition is applied for energy.

The influence of the blanket and reflector in the neutronic analysis was modelled using albedo boundary conditions in the contacting surfaces, provided in appendix B. Vacuum boundary conditions are used on the other surfaces. Eliminating the reflector and blanket from the geometry by using albedos greatly reduces the number of unnecessary cells in the domain.

Power in this axisymmetric geometry is normalised to 187.5 MW (i.e., $\frac{1}{16}$ of nominal power) during eigenvalue calculations, which ignores any power produced in the blanket. For the 3 GW nominal power of the whole reactor, previous studies have found the power production in the blanket to be around 20 MW [Pettersen, 2016, p. 28]. Therefore, ignoring power production in the blanket is a reasonable approximation, whose consequence is a negligible overestimation of the flux in the core. Regardless of the nominal power of the reactor including the blanket or not, the other partners used the same conditions.

6.2.1. Steady State Results

Figure 6.21 shows the pressure and velocity fields. The velocity field show that a recirculation is still present at the core inlet, which was known to exist. Although the curved wall was intended to eliminate recirculation, it cannot do so fully as-is. Several recirculation vortices are spread above and below the heat exchanger as a result of the decision to cut heat exchanger size without redesigning the region, which was expected as well. A significant pressure drop at the heat exchanger is present, as expected from the friction factor imposed; cross-flow is also absent as intended. Measurements of pressure at heat exchanger inlet and outlet indicate a pressure drop of 4 bar as desired. The pressure field shows inconsistencies with what would be normally expected, such as lacking a pressure increase in the pump. While this is an unphysical result, the simulation is fully incompressible, therefore pressure assumes the meaning of a Lagrange multiplier,

rather than its usual thermodynamic one. Since the model used a few functionalities that were unverified, it is difficult to say how these affected the Lagrange multiplier without much deeper studies focusing on this issue specifically. While velocity was found to generally agree with the other institutions, we can expect it to contain inaccuracies due to problems in the pressure field, which will affect transported quantities.

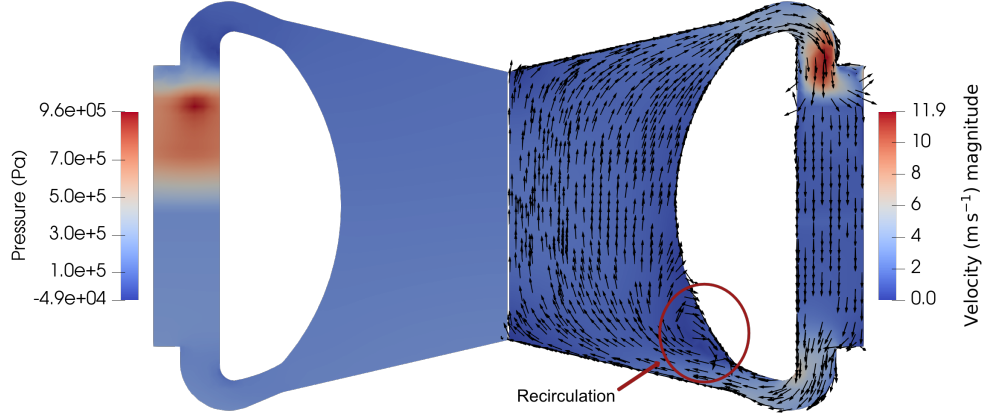


Figure 6.21.: MSFR pressure and velocity fields with vectors

The volumetric power of the cavity is strongly centred as figure 6.22 shows. As a result, that is where temperature increases the most. Temperature isolines are drawn in the figure as well, where we can see that the top reflector will be exposed to salt at a very high temperature. Turbulent mixing homogenizes fluid temperatures at the core outlet.

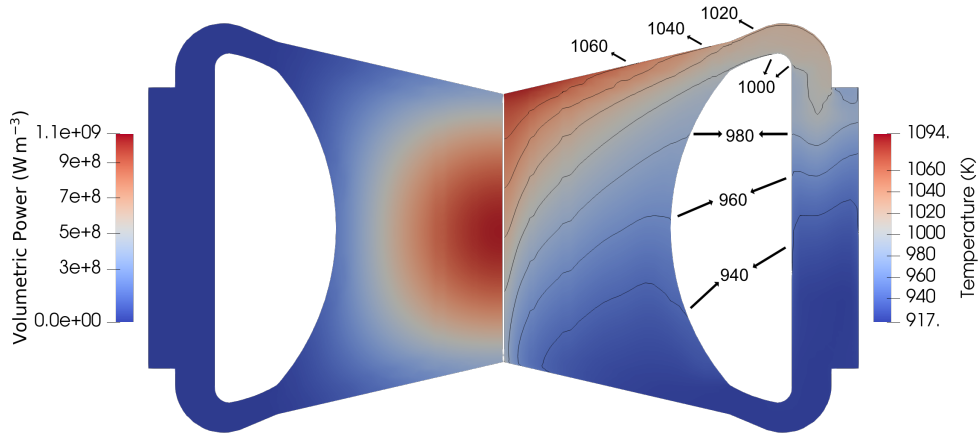


Figure 6.22.: MSFR volumetric power and temperature fields

As indicated by the volumetric power, fluxes are also strongly centred as shown in figure 6.23. Of interest in this figure is that the flux in the most thermal group shows

an increase near the blanket and especially near the reflector. This is the expected behaviour as well, and confirms that the albedo boundary conditions successfully models this phenomenon.

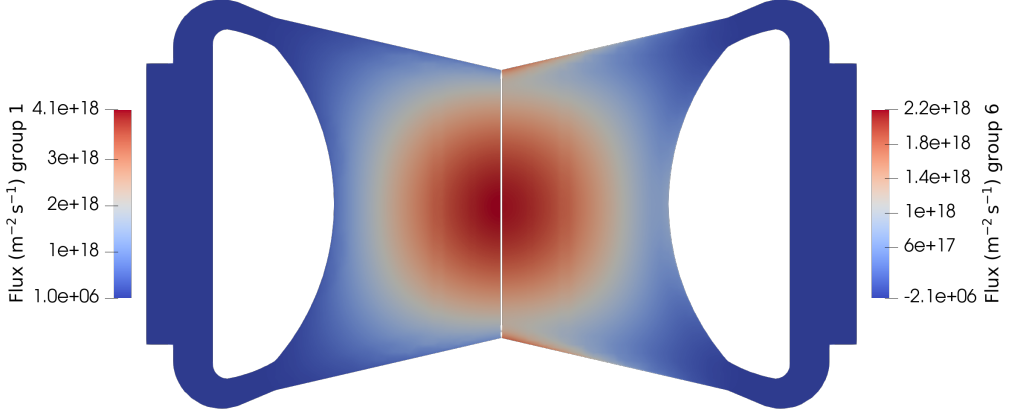


Figure 6.23.: MSFR neutron flux of groups 1 and 6

In figure 6.24 we can see that the longest lived delayed neutron precursors (group 1) essentially smears through the cavity. The slightly distorted shape of the shortest lived one (group 8) is also clear, by comparing it with the static profile, which would match the volumetric power. Therefore, the shortest lived group drifts from its origin slightly, but the longest lived one does spread and its concentration is 2 orders of magnitude higher. We can expect that in a transient where flow is lost, the reduced drift of the relatively long lived groups will have a measurable impact. A concentration of the long lived delayed neutron precursors in the heat exchanger is also noticeable, which is likely a compound effect of (1) errors in the velocity field and (2) loss of diagonal dominance in the matrix, which is required by Krylov solvers. The problem of diagonal dominance can be easily solved by using a non-Krylov solver, however, the error in the velocity field is a deeper issue requiring a more detailed verification effort. While unintended, the impact of this artefact is rather small and the values in the core are of the same order of magnitude compared to partners PoliMi and TUDelft.

The eddy viscosity field is shown in figure 6.25, which serves to emphasize the effect of turbulence in the flow field and quantities of interest. The exact value is not so important, rather its relative distribution allows greater insight into the behaviour of other fields. For example, it is evident that the location at the core inlet where figure 6.21 shows recirculation is also a location where eddy viscosity is particularly high. However, the turbulent mixing close to the top reflector is the reason for the homogenization of the temperature field until the core outlet, which is not necessarily obvious considering that the velocity field does not show any vortices in the region.

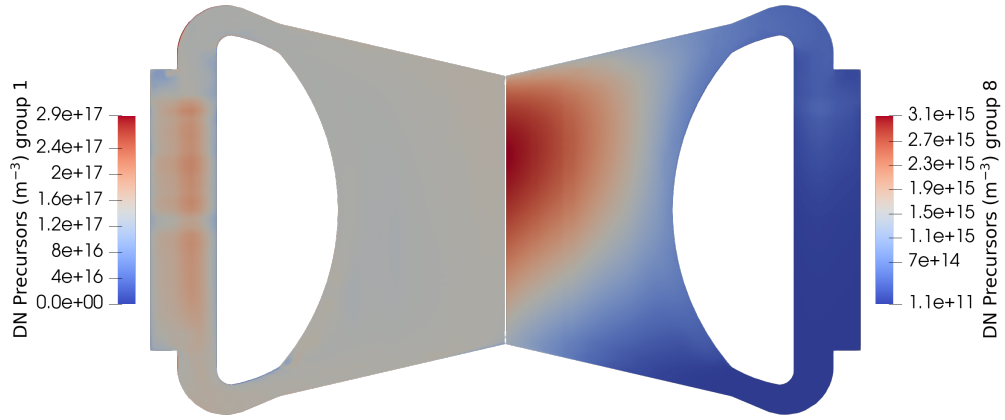


Figure 6.24.: MSFR delayed neutron precursors of groups 1 and 8

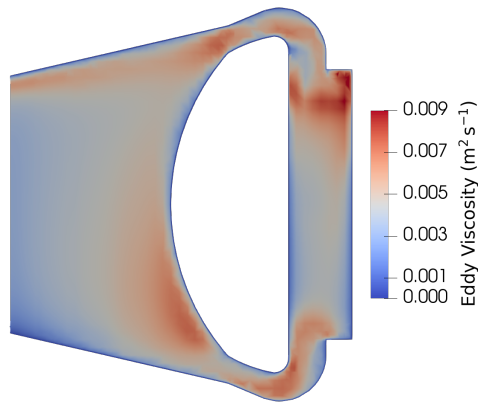


Figure 6.25.: MSFR eddy viscosity field

6.2.2. Transient problems

The transient simulations of the MSFR were aborted during execution. It proved impractical to perform the analysis and ensure systematic convergence of results without incurring a disproportionate computational cost, which indicates a flaw in the methodology used by the author.

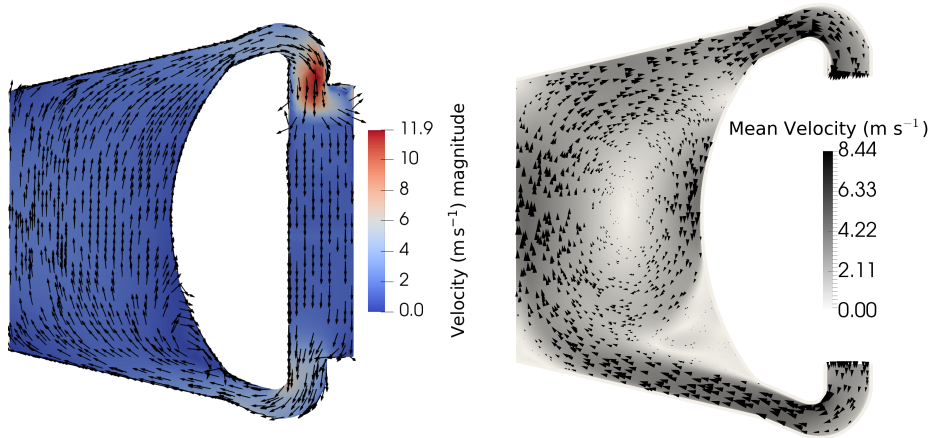
The cause was identified to be the coupling method and the requirement of a suitable acceleration scheme. The issue will be discussed in detail at the end of the section.

6.2.3. Discrepancies between studies

Before starting an elaborate discussion, it is relevant to mention explicitly another work also analysing the MSFR in order to point out the differences and draw constructive criticism of the current state, and limitations of analysis.

In the work of Cartland-Glover et al. [2019] a frozen wall is modelled as a boundary condition. As a result of this choice, we can see an extremely different behaviour of the velocity and temperature fields as compared to the results obtained in this thesis (and in the SAMOFAR project).

In figure 6.26b we can see recirculation in the bulk of the reactor, not just close to the inlet as in figure 6.21. As a result of this bulk recirculation, the temperature field in figure 6.27b is also very different than the one shown in figure 6.22.



(a) SAMOFAR results

(b) Results from Cartland-Glover et al. [2019].
Reproduced with permission.

Figure 6.26.: Comparison of MSFR velocity field with literature.

While it is hard to say how such changes would impact the transient behaviour of

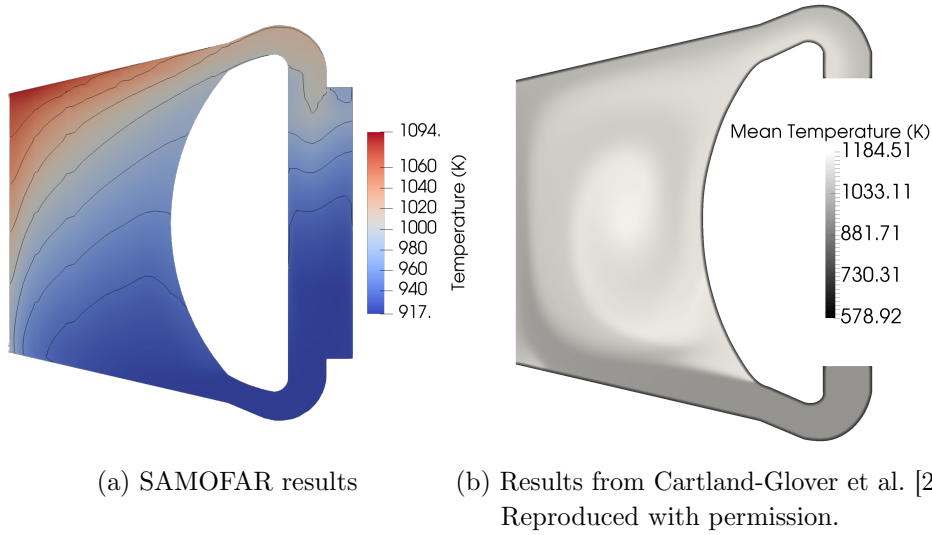


Figure 6.27.: Comparison of MSFR temperature field with literature.

the core, it is reasonable to expect that results would be different — maybe even substantially different.

6.2.4. Discussion

The steady state of the MSFR was simulated in this section, and several conclusions can be drawn from it.

Not all functionalities required by this simulation were formally verified, resulting in inaccuracies in modelling of pressure drop in the heat exchanger. This negatively affected the pressure and velocity fields and related quantities, such as delayed neutron precursor transport. A more detailed and systematic verification of the implemented correlations, such as friction, is required.

Most of the recirculation in the MSFR core has been eliminated by the curved walls if walls are adiabatic, as intended by the design changes. Some recirculation at the inlet is still present, but recirculation of high temperature salt is eliminated in the conditions used.

If the walls are considered to have a temperature that is significantly different than the salt, results from the literature indicate that bulk recirculation is present again, as shown in section 6.2.3. In both cases, the walls are not in contact with high temperature salt, either because the recirculation was eliminated or because there is a layer of frozen salt at the wall, however, it is reasonable to expect the transient behaviour of these 2 cases to be different.

A study to resolve uncertainties in the boundary conditions and turbulence model seem necessary, as the uncertainties present at the moment are so high that the results can be obviously different. Alternatively, a study of options to eliminate these issues by design could be considered.

The flux shape inside the MSFR is just as one would imagine. To emphasise just how expected it was, in [Tiberga, Lathouwers, & Kloosterman, 2020] it is shown that the deviation from an analytical solution to a diffusion equation in a cylinder is negligible, despite the shape of the walls. Therefore, there is no reason to use spatial kinetics to model an axisymmetric MSFR core; a point kinetics model changing the amplitude of an analytical solution will give the same result with negligible loss of accuracy and much better efficiency. Even if the intent is to model a non-axisymmetric transient, a point kinetics approach will be of benefit at least for obtaining a fast and very precise initial guess of the steady state condition, which can then be used to start a spatial kinetics solver in steady state conditions and progress to non-axisymmetric problems. This may seem to give marginal benefit, since the steady state condition is calculated only once for several transients departing from this equilibrium. However, in order to find a suitable design candidate that is worth doing transient analysis of, it might be required to go through steady state studies of several geometries instead. Transient analysis may be a big part of analysis of a mature reactor design, however, steady state analysis is a big part of design studies to find a design worth maturing.

The lack of a proper design of the heat exchanger and surrounding region is very detrimental to modelling of the fuel circuit. In this analysis, the size of the heat exchanger was limited to an upper section in order to bring the thermal centre of the heat sink to a higher position and promote natural circulation. However, the same decision of limiting the size is detrimental to natural circulation by resulting in a heat exchanger with very high pressure drop (approximately 4 bar in this case); as expected of any compact heat exchanger capable of removing 187.5 MW and fit in a mere 0.188 m³. For this analysis, it was decided by the participating institutions to proceed regardless of these considerations, however, redesigning the heat extraction of the primary circuit should be a high priority. The lack of information and design of the pump is understandable, at least compared to the heat exchanger. The top “pipe” could probably be redesigned as needed to accommodate different pumps with little consequences, unlike the heat exchanger design, which the geometry and this analysis is very sensitive to. Therefore, while this analysis has been performed, its results are mostly a representation of the capabilities of the codes involved.

Broad conclusions that are generally true, such as the flux shape matching an analytical solution for a cylinder, could be inferred with confidence without any previous numerical analysis. Specific conclusions that could impact safety considerations and cannot be inferred by engineering judgement, requiring numerical simulation, cannot be derived with confidence with the current state of knowledge, because even the steady state is

uncertain as pointed out.

After the steady state analysis was performed, it was the intention of the partners to proceed to analysis of transients considered important by other studies of the SAMOFAR project. This is the point where this thesis deviates from the other partners. ATARI could not perform the transients successfully even after significant effort. Therefore, it was decided to study the origin of this limitation, which was thought to be a more fundamental issue, while the partners continued to perform the transient analysis.

The reason was found to be the way how ATARI (and any other OpenFOAM-based solver) deals with transients of coupled neutronics and fluid dynamics problems. At the moment, these coupled solvers rely on acceleration schemes to perform transient analysis. **Without an appropriate acceleration scheme, the solver does not reach convergence inside a time step.** In section 6.1, the same codes participating in this MSFR analysis were used in a benchmark, where oscillating transients were induced by varying the heat exchange coefficient of the volumetric heat removal. That benchmark transient scaled the amplitude of the fields, but did not result in spatial shift of the quantities. In other words, the transient could be represented as an integral quantity oscillating in time. For that analysis, ATARI used the Aitken Extrapolation to predict the integral power during iterations in a time step and scale the flux accordingly, with great success.

However, these integral conditions are not representative of the MSFR transients, where spatial shift of quantities occur. Spatial changes in flow field result in spatial changes of delayed neutron precursors concentration and temperature fields, both with their own significant feedbacks. The Aitken Extrapolation is not suitable for these transient simulations. This was also found in previous studies [Aufiero, 2014, p. 57], where the explicit, singly diagonally implicit Runge–Kutta (ESDIRK) [Kennedy & Carpenter, 2003] acceleration scheme was used successfully. Unfortunately, such important finding was not reflected in the conclusions of the work.

The inability to solve this problem in a reasonable and systematic way and the requirement for appropriate acceleration schemes is a reflection of the limitations imposed by the OpenFOAM library. As explained in chapter 4, as of version 8, the framework only allows solving equations in a segregated manner (also known as operator-splitting scheme), that is, solve the equation for each quantity separately and couple them by the explicit source. This coupling approach has slow convergence [Hopkins et al., 2007, p. 22], low efficiency due to residual over-solving [Wang et al., 2020], and in extreme cases fail to accurately describe the transient [Aufiero, 2014, p. 57]. If a case has a phenomenon that strongly dominates all others, this characteristic does not appear as the dominant phenomenon marches in time and the others follow with little feedback. The problem arises when there is no or relatively little difference in importance (i.e., no dominance). OpenFOAM itself realises this, and for this reason the solution of the

3 components of velocity can be performed simultaneously, which increases the rate of convergence in flows that are strongly 3-dimensional, where one component is not particularly dominant over others (The OpenFOAM Foundation, 2013; Uroić, 2019, p. 182).

The segregated approach requiring the use of acceleration schemes has a few additional problems. Each acceleration scheme is an additional user decision that does not result in systematically predictable outcomes, requiring trial, error and experience. The use of segregated approach is inherently ill-suited due to difficulty or impossibility to properly verify the code by making spatial and temporal order of accuracy ambiguous [Hopkins et al., 2007, p. 89]. Even if it was not ambiguous, the requirement — not the optional use — of acceleration schemes makes it effectively impossible to formally verify the order of accuracy of temporal schemes, since what is verified is not the temporal scheme, but the temporal scheme together with the acceleration scheme. In addition, accuracy of results is dubious anyway because it cannot be guaranteed that the acceleration scheme used was appropriate for resolving all parts of the transient. Finally, asserting convergence in a time step is challenging. This is particularly important because the reason for doing a transient simulation usually lies in the information contained in one or some time steps, not the asymptotic behaviour. Furthermore, the most valuable information will typically be contained in the time steps that are hardest to converge, due to some strongly changing quantity. If ensuring convergence in the most challenging time step is not systematically achievable, the confidence in the simulation is dubious.

The normal justification for using segregated solutions, instead of simultaneous one, is the reduced computational requirement. The time required to solve a matrix increases rapidly with matrix size; sometimes up to $\mathcal{O}(n^3)$ [McClarren, 2018, p. 368]. There are 2 main influences to matrix size: cell count and coupled equations. Cell count is pretty much self-explanatory; a mesh with more cells will result in a larger matrix. Coupled equations that are solved simultaneously results in what is called a “block matrix”, which is a matrix of matrices; more coupled equations result in larger block matrix.

Cell count is typically not chosen but a result of requirements in spatial convergence of quantities, however, we do have more control over the number of coupled equations; at least apparently. Therefore, to manage computational requirements, the segregated solution of 2 coupled equations results in solving 2 small problems, which is computationally cheaper than solving 1 big one — the caveat is that no mention of convergence **in time** was made. A simultaneous solution is clearly beneficial if iterating the solution of 2 equations/small problems until convergence in time requires longer program run times than converging a single big one. In nuclear reactor problems, tightly and loosely coupled situations can arise depending on design, therefore, having the option of choosing a segregated or a simultaneous solution is necessary.

All these considered, we are essentially optimising a balance of cell counts and coupled

equations to manage matrix size and keep the problem practically solvable. In other words, we want to minimise cell count and coupled equations at the same time, which brings us to 2 main topics: homogenised models and economical models. Homogenised models, where fine geometrical details are not resolved, significantly reduces the cell count and matrix size by virtue of the method — it’s a method to tackle the requirements of spatial discretisation. Economical models bring into discussion the need for parsimony when solving engineering problems. If “all models are wrong, but some are useful” [Box, 1976], we have to carefully consider what is the simplest model that represents the problem to be investigated. For example, the nuclear cavity benchmark from section 6.1 shows that to model that problem there is little or no benefit in using an elaborate model such as discrete ordinates over a rather simple transport-corrected diffusion model, which significantly reduces the number of coupled equations. This also emphasises the benefits of models even simpler than diffusion, such as point kinetics.

The reduction of cell count given by homogenised methods even limit the possibility of parallelism in segregated solvers. Domain decomposition only brings significant benefits when the run time is limited by processing power. After a certain amount of decomposition, the matrix present in each CPU is small enough that solution is limited by the transfer of information of the boundaries between split matrices, instead of processing power. As evidence of this bottleneck, the MSFR case was executed in a mere 6 CPU cores, but run time was reduced by a third, not a sixth. Further increase of CPU cores gave even smaller benefits. Therefore, the increase in matrix size as a result of simultaneous solutions counteract the reduced matrix size of the homogenised models and realigns the entire concept with modern paradigms of parallelism and high performance computing.

It is not known at the moment what level of simultaneous solution is required in order to solve tightly coupled problems systematically and with high confidence. For example, solving fluxes and delayed neutron precursors simultaneously, or solving these coupled with the Navier-Stokes equations all at once. What is certain is that solving every quantity in a segregated way is not the right way forward. This conclusion is not entirely new [Ragusa & Mahadevan, 2009], however, previously it was contextualised for legacy codes, whereas here we contextualise it to a modern framework. This can be done because, despite the modern framework, segregated solvers in OpenFOAM are coupled in the same way as independent legacy codes.

This puts ATARI and the usage of OpenFOAM in coupled neutronics and fluid dynamics simulations in an awkward position. The niche where unstructured meshes and high fidelity multiphysics simulations in a CFD-like framework fit the best are challenging problems that require a flexible spatial description due to spatial shift of quantities, such as a changing flow field or deforming structure. However, these are precisely the situations where the current framework is only partially or totally unsuited for, since it **requires** different acceleration schemes to deal with different cases. For situations

where unstructured meshes are not needed, the niche is filled by other codes where structured meshes bring an absolute advantage in run time and, despite criticism, excel at the task they were designed for, such as several system codes (e.g., TRACE, Cathare, Athlet).

The idea of solving quantities simultaneously also aligns with the expected computational requirements of executing a special-purpose high fidelity code, such as ATARI. In the niche identified for ATARI of challenging simulations with spatial shift of quantities, the expectation of fast results are relaxed as long as run time can be kept reasonable. Therefore, these are already situations where the analyst is already amenable to the penalty of higher computational requirement of simultaneous solution of coupled equations.

The problems that emerged during the simulation of the MSFR allowed, for the first time, a clear view of the benefits of the methods chosen, the current flaws in implementation of ATARI and similar codes, and the niche where these codes fit the best. While the idea behind the concept is promising, it needs to be reimplemented in a framework that allows proper leveraging of its benefits.

A secondary conclusion is that, while the nuclear cavity benchmark from section 6.1 has its usefulness, the transient case proposed is not representative of the requirements for transient simulation of the MSFR. The benchmark has to be supplemented by a transient case with enough spatial shift to challenge coupling methods and acceleration schemes.

6.3. Freezing in Heat Exchanger

Several accidents of the fuel salt circuit of the MSFR are hypothesised to result in solidification of the salt in the heat exchanger [Allibert et al., 2017], therefore, one of the intentions of this thesis was to explore the repercussions of this kind of accidents. However, if the current heat exchanger data is insufficient for detailed analysis, it is even more deficient when it comes to phase-change simulations. These simulations are very sensitive to heat exchanger design; particularly to channel cross-flow resistance. The current design is of a very compact printed circuit heat exchanger — so in principle no cross-flow — and with needle-sized channels. In such a design, it is foregone that excessive heat extraction or loss of flow will result in quick solidification of the percolating molten salt and flow blockage.

The heat exchanger design is being actively worked. One of the main reasons is the large pressure drop of the heat exchanger, and the small height difference between thermal centres of heat source and sink. These characteristics of the current design limit the use of natural circulation as a means of passive safety in case of loss of pumps [Ghetta

et al., 2019, p. 25], which is a desired feature. A partial or total loss of heat sink, and failure to dissipate decay heat, will lead to melting of the freeze plugs in the bottom of the reactor and draining of the fuel salt to the emergency draining vessel below the core, which is a measure of last resort.

With the current design where solidification is foregone, and as long as the heat exchanger design is possibly subjected to big changes, there is little point in performing a detailed simulation. Instead, it was decided to perform an exploratory study of solidification, which even given the current lack of information, might give meaningful insights into future decisions on heat exchanger design.

The proposed case is a momentum-driven pipe with geometry shown in figure 6.28. It consists of a 2 m long pipe with square cross-sections of 0.20 m side, reminiscent of the heat exchanger region of the MSFR. The geometry is meshed in 20 graded (i.e., non-uniform) cells in the xy -plane for both axes, with increasing refinement near the walls, resulting in 400 cells at the face. Along the yz -plane, the mesh is discretised in 200 uniform cells along the z axis. The resulting mesh consists of 80000 hexahedral cells with perfect orthogonality.

The red and blue zones represent heat exchanger and “pump” (actually a momentum source) regions respectively, which are zones where fluid is homogenised with a solid structure representing the heat exchanger structure. In the other regions, fluid is not homogenised. The boundary conditions in these different regions are given in table 6.3.

Fluid comes from the pump side, passes through the pump, flows through the heat exchanger and leaves on the opposite side. Table 6.4 shows the properties of the fluid utilised for the simulation where T_{ext} is the temperature of the secondary side of the heat exchanger. Additionally, the heat exchanger zone has a fluid fraction of 0.3 and **friction factors that allow a very small amount of cross-flow**. The simulations use a laminar approximation because the main objective of using a turbulence model would be to homogenise by turbulent mixing the inlet of the heat exchanger, however this can also be artificially achieved using boundary conditions later on.

Table 6.3.: Freezing HEX boundary conditions

Field	Inlet	Outlet	Homogenized Walls	Other Walls
Pressure	zero gradient	1 bar	zero gradient	zero gradient
Velocity	zero gradient	zero gradient	slip	no slip
Energy	600 K	zero gradient	zero gradient	zero gradient

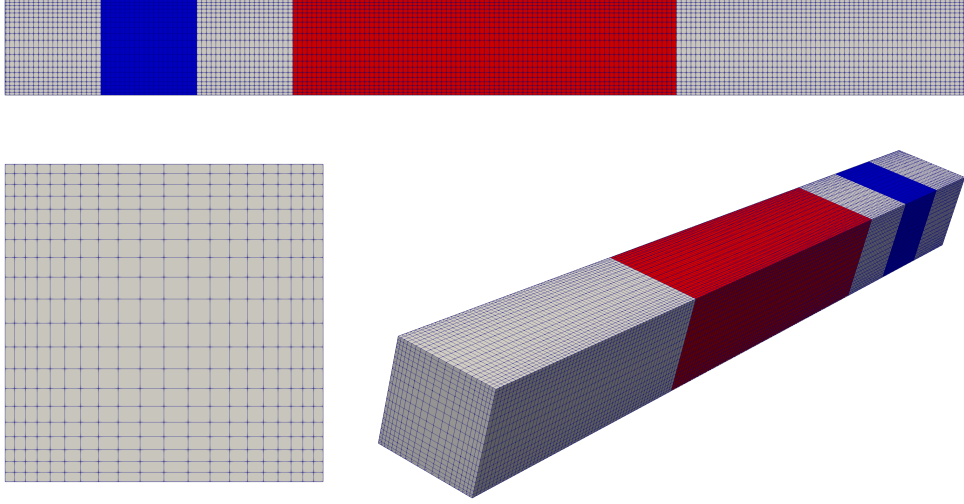


Figure 6.28.: Square pipe and HEX mesh side, front and perspective

Table 6.4.: Freezing HEX fluid properties

ρ	10^3 kg m^{-3}
c_p	$10^3 \text{ J kg}^{-1} \text{ K}^{-1}$
μ	10 Pas
Pr	10^4
L	10^5 J kg^{-1}
T_{melt}	525 K
T_{ext}	500 K

6.3.1. Reference

The figures below present what would be the results of the simulation if freezing model was not used. Its purpose is to serve as a reference so that the results of the simulation involving the freezing model can be compared against.

In figures 6.29 and 6.30 it is possible to see that pressure increases between the pump and the heat exchanger, while dropping throughout the heat exchanger. Temperature decreases almost uniformly as the fluid passes through the heat exchanger, entering with 600 K and leaving with approximately 523 K. The temperature T_s of the solid material composing the heat exchanger follows almost the same trend as the fluid. Flow is approximately 1D and flow of fluid inside the homogenised zone given by the sub-grid scale velocity (i.e., SGS Velocity) is accordingly much faster than in zones without a solid structure. Reminding that velocity in this solver is a homogenised velocity represented by $u_f = \gamma u$, where γ is the fraction of the homogenised zone that is occupied by fluid.

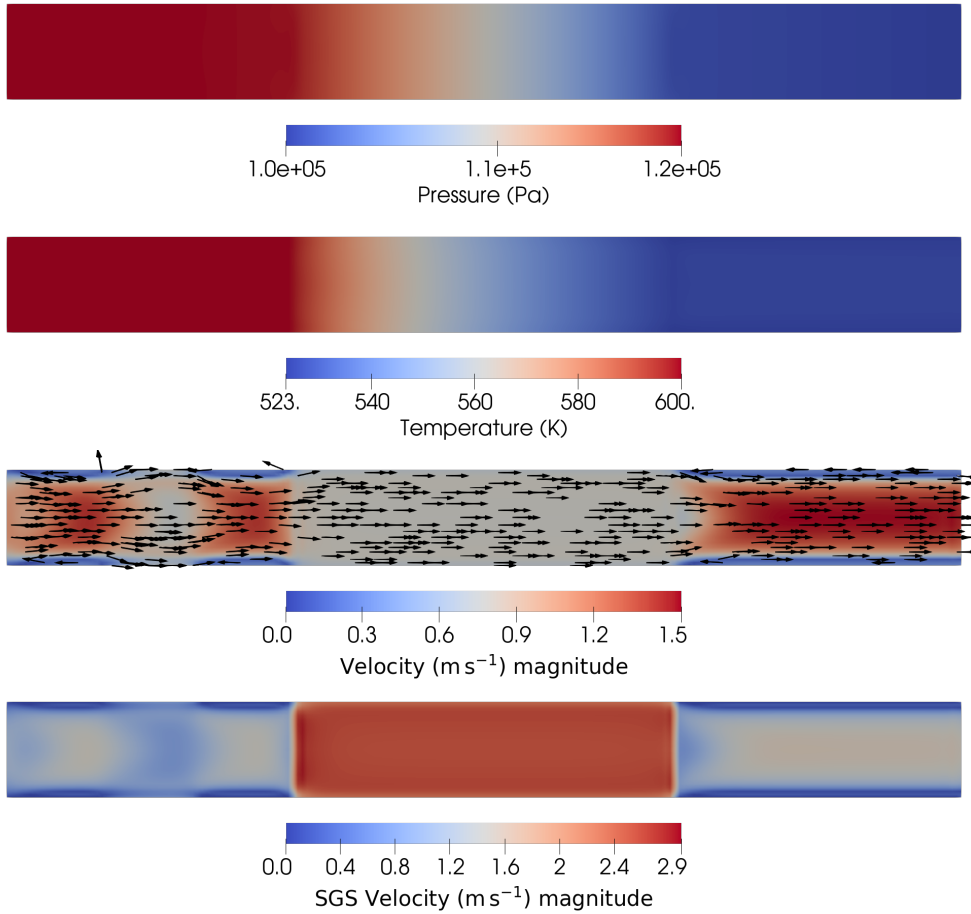
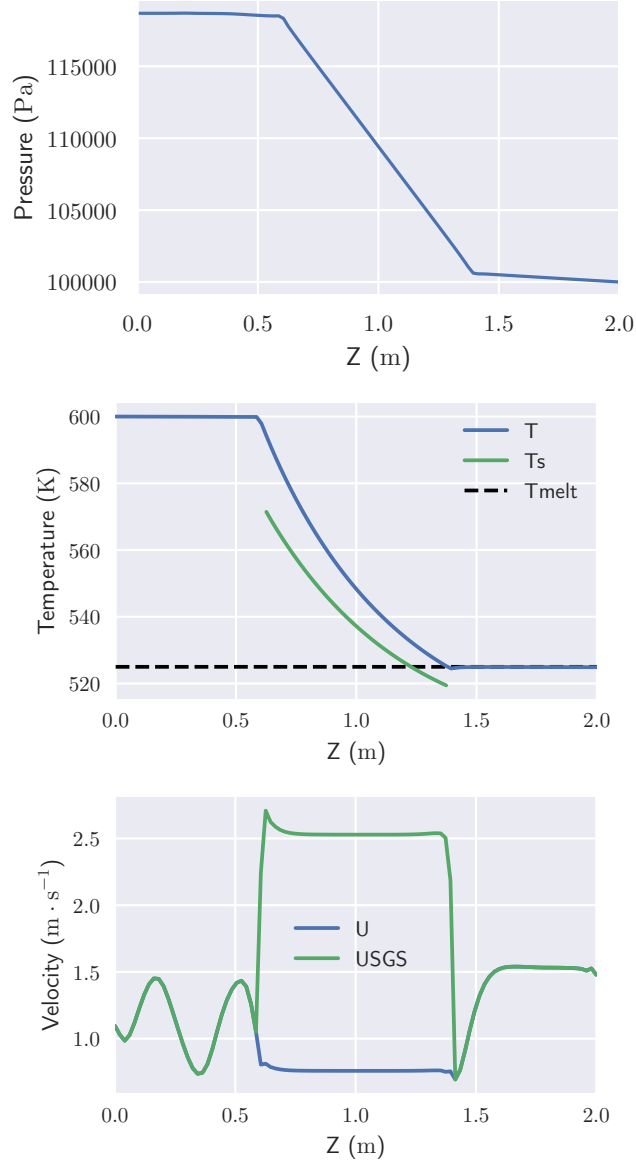


Figure 6.29.: Reference field results without freezing at $t = 20$ s

Figure 6.30.: Reference plot results without freezing at $t = 20$ s

6.3.2. Freezing with tuned salt

In this part, simulation is done with freezing model activated and results are shown at $t = 5, 6$ and 20 s.

In figures 6.31 and 6.32 it is possible to see what happens when the outlet of the heat exchanger is just about to freeze due to the temperature field at $t = 5$ s. Liquid fraction in this zone approaches 0, forming a mushy zone with partial blockage. Back pressure increases as a result of this partial blockage and the flow field forms a focus point where velocity increases 4 times approximately.

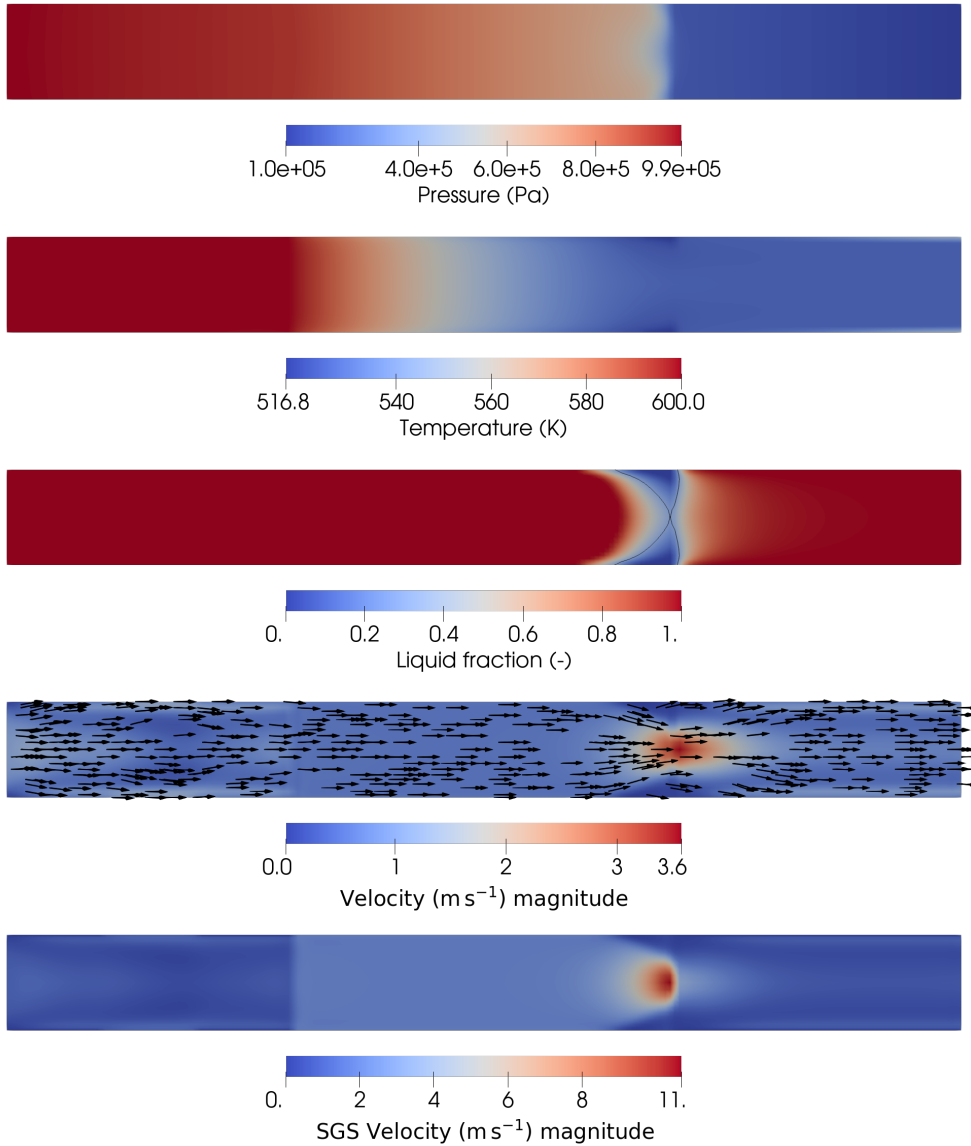
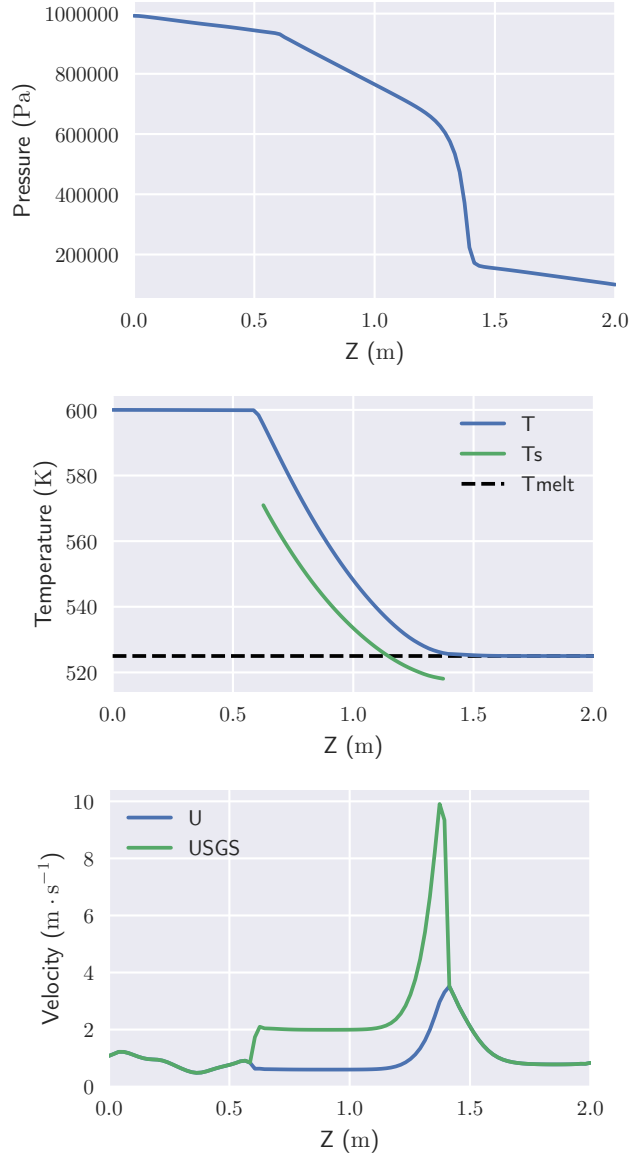


Figure 6.31.: Freezing HEX fields at $t = 5$ s

Figure 6.32.: Freezing HEX plots at $t = 5$ s

At $t = 6$ s, pressure spikes to its maximum, leading the flow pattern to focus at a single point for breakthrough. Fluid velocity increases to 12 times its nominal value, generating a fast rate of renewal of fluid at the choking region, preventing further temperature drop and solidification. A jet effect is formed at the outlet of the heat exchanger as shows the vector arrows of the velocity field.

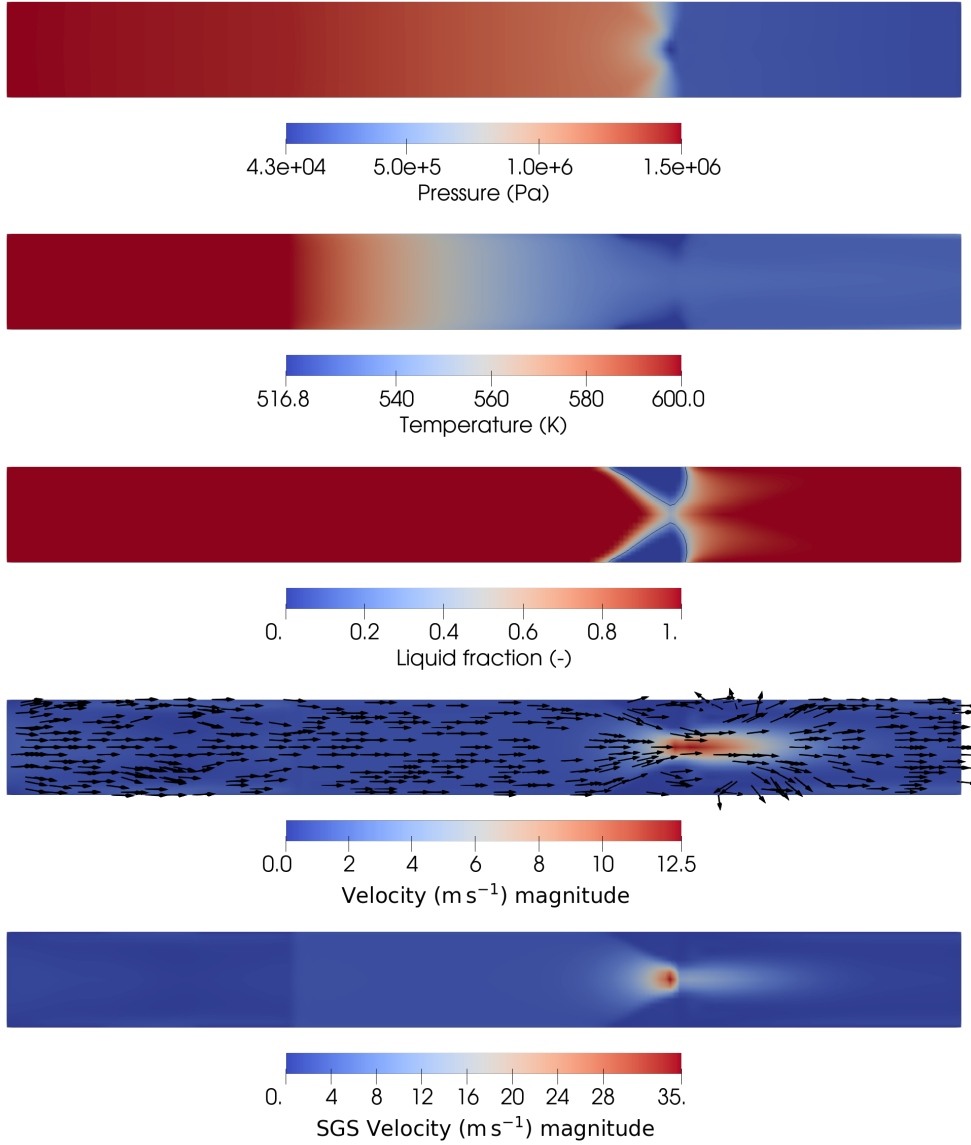
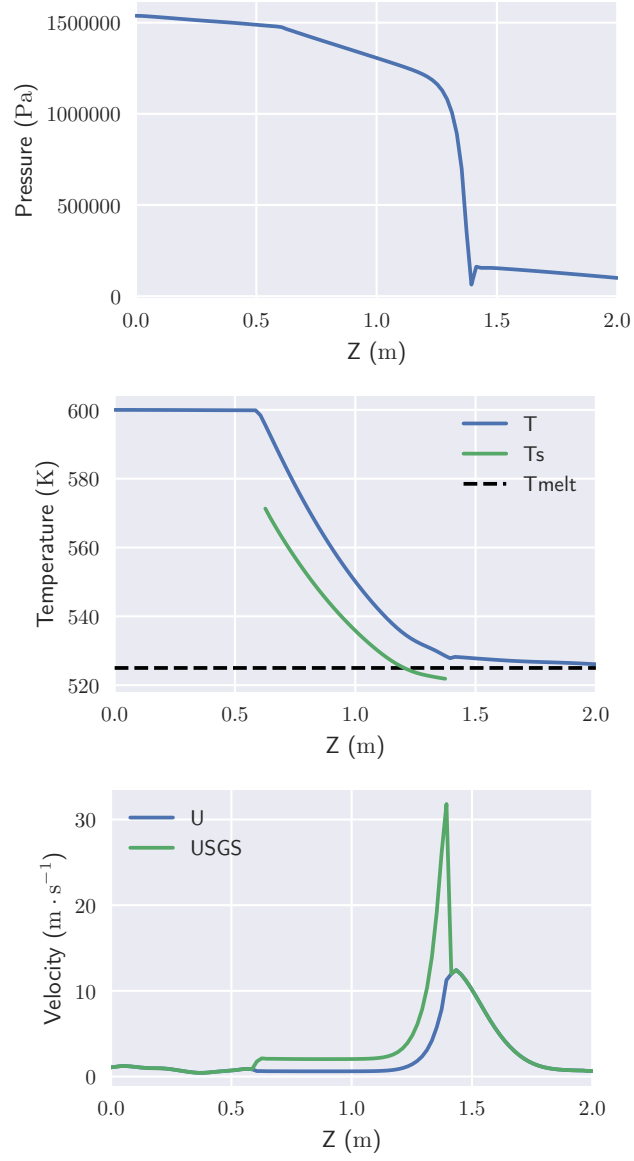


Figure 6.33.: Freezing HEX fields at $t = 6$ s

Finally, at $t = 20$ s, the end of the simulation, flow has stabilised. Breakthrough is achieved and the focus point from the time of onset of freezing opens up, partially undoing the flow choking. Pressure is still higher than nominal values but the flow is, in principle, stable if this condition can be sustained.

Figure 6.34.: Freezing HEX plots at $t = 6$ s

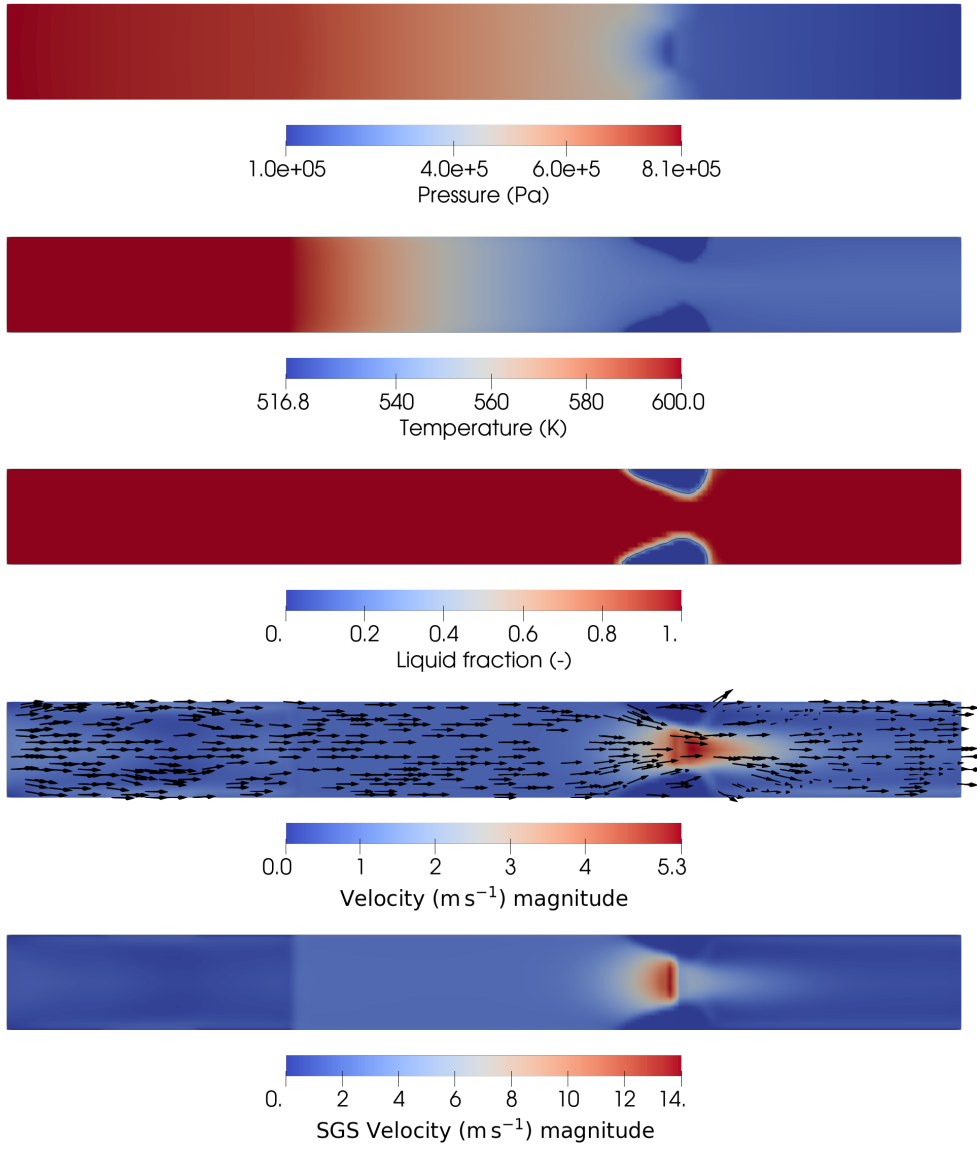
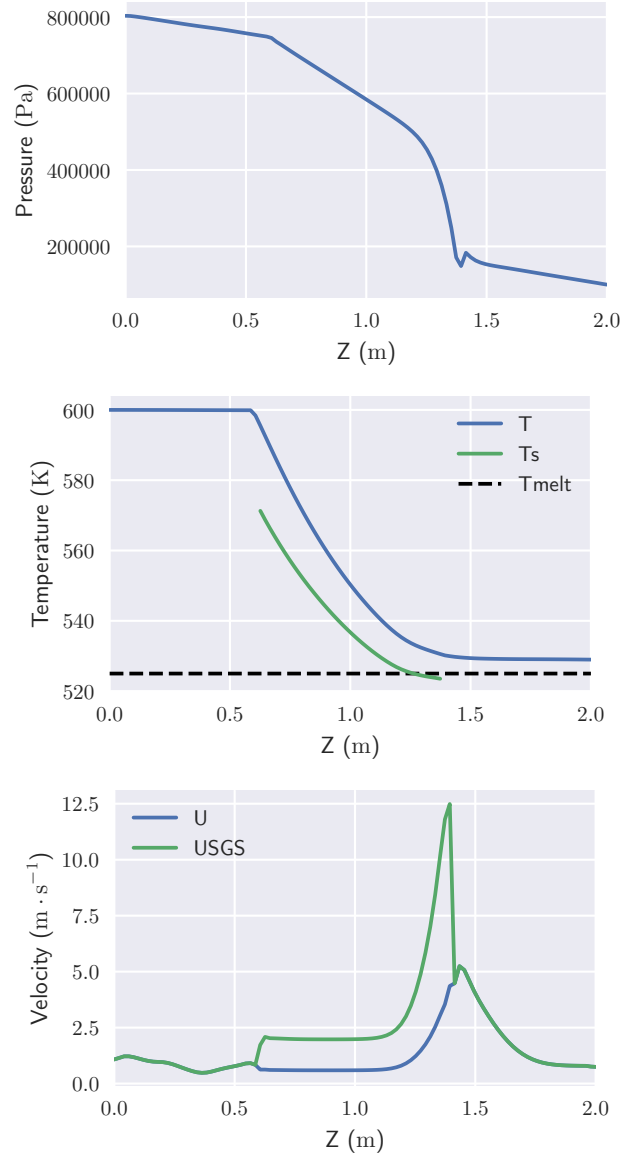


Figure 6.35.: Freezing HEX fields at $t = 20$ s

Figure 6.36.: Freezing HEX plots at $t = 20$ s

6.3.3. Freezing with MSFR salt

Like in the previous section, simulation is done with freezing model activated but using MSFR salt properties. Results are shown at $t = 5$ and 6 s, since an asymptotic state is quickly reached.

Compared to the previous demonstration, a much larger volume of the fuel salt is solidified. This should not be a surprise as thermophysical properties of previous case was chosen to result in rather simple phenomenology.

Essentially full blockage takes place at $t = 5$, leading to a pressure at inlet that is significantly higher than previous demonstrations.

At $t = 6$ s the blockage has been recovered with a full passage available.

We can see that previous trends are respected regarding breakthrough. Both solidification and melting of the salt are more abrupt.

In figure 6.39 we have the same MSFR salt as before, however all pipe boundary conditions are slip conditions. This prevents a flow profile from developing at the inlet of the heat exchanger, therefore, the salt is equivalent to a perfectly mixed and uniform fluid. The intention was to make a surrogate to perfect mixing due to turbulence, but without actually using a turbulence model, since all that can be expected of the turbulence model is a better representation of the heat exchanger outlet after breakthrough, when a jet forms.

Due to the uniformity of the flow, it is much more difficult to cause solidification of the salt. Heat transfer parameters has to be significantly increased in order to extract enough heat to cause solidification, with blockage at $t = 1.2$ s. Due to the uniformity of the properties, blockage is very abrupt and causes a massive 4×10^7 bar at the inlet.

At $t = 3$ s the blockage has been recovered with a full passage available (at least in principle) With the abruptness of the solidification of the heat exchanger, it is possible to interpret the recovery of the passage as merely a numerical artefact. Still, the trend of recovery of the heat exchanger is seemingly maintained among all test cases.

6.3.4. Discussion

While at first the geometry might be seen as highly idealised, it is about as idealised as the current description of the MSFR itself, where the heat exchanger location is a “pipe with rectangular cross-section”.

Putting aside a mere demonstration of the capability of the model, the analysis, while

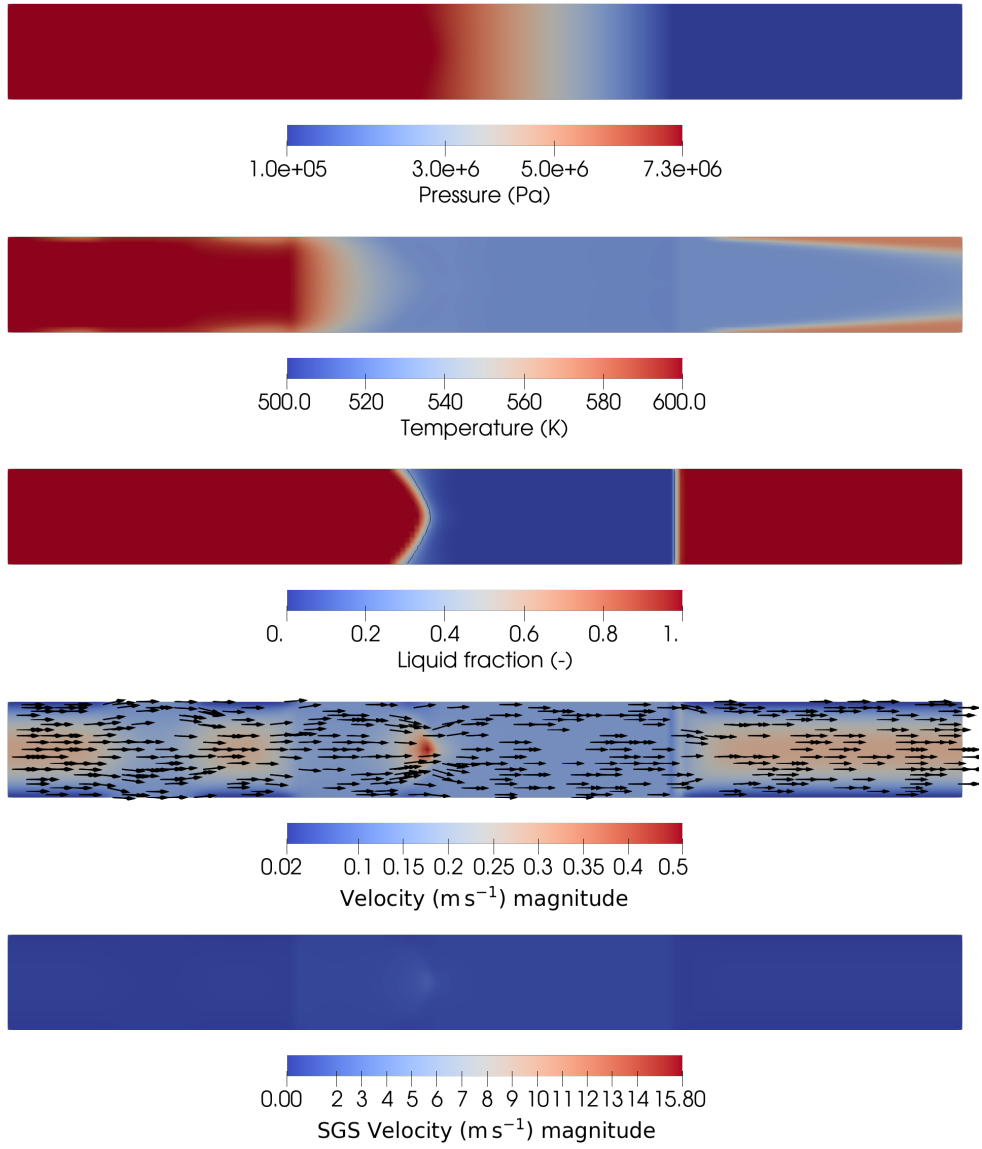


Figure 6.37.: MSFR Freezing HEX fields at $t = 5$ s

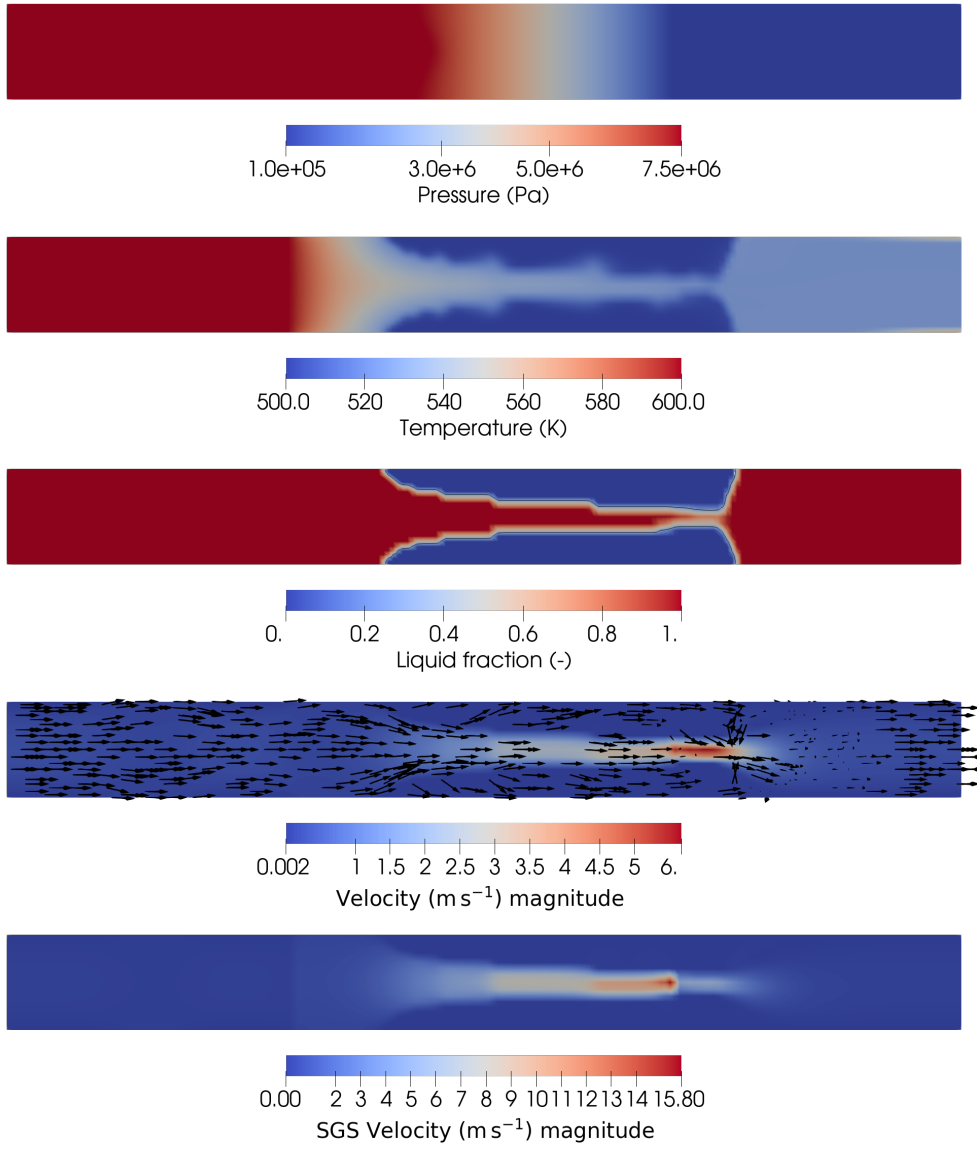


Figure 6.38.: MSFR Freezing HEX fields at $t = 6$ s

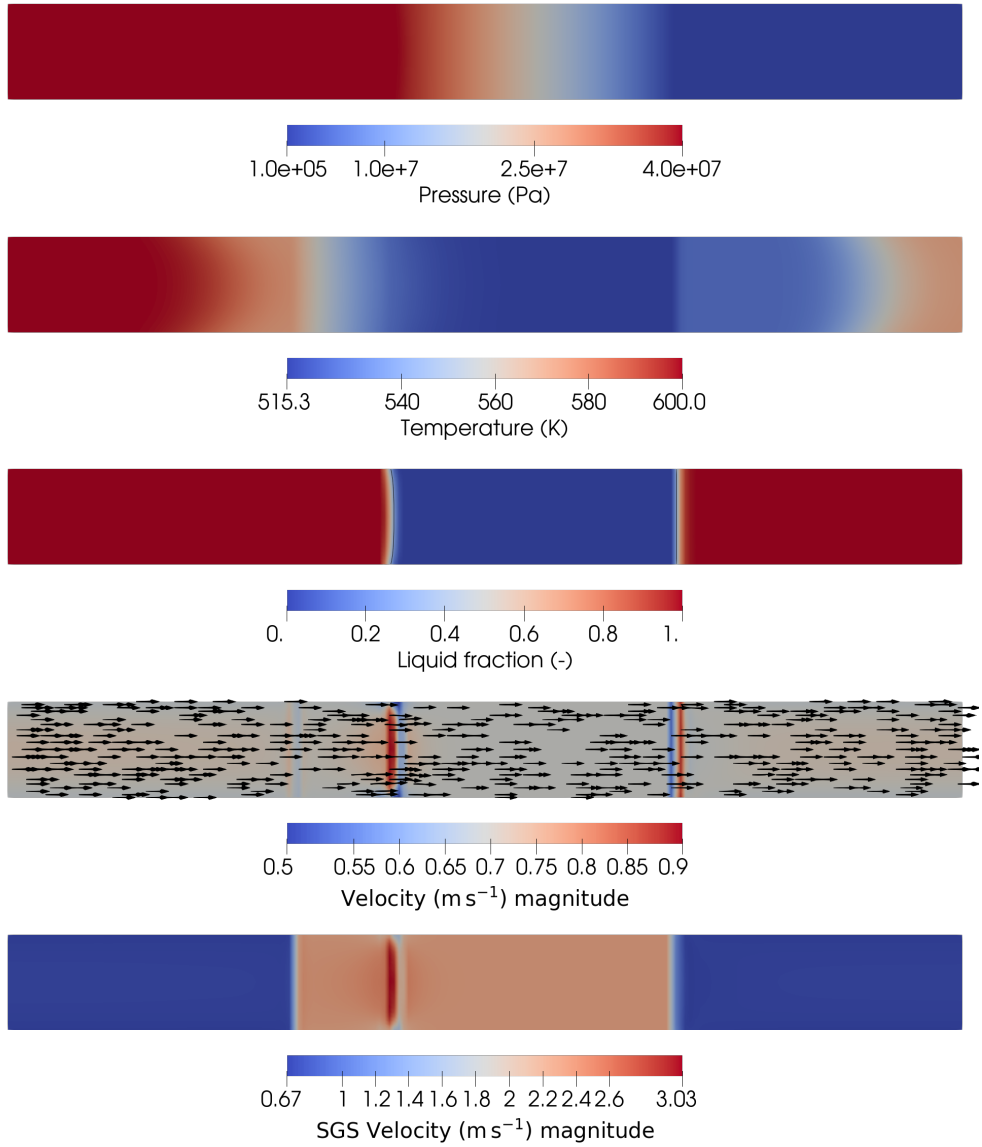


Figure 6.39.: MSFR Freezing HEX fields at $t = 1.2$ s

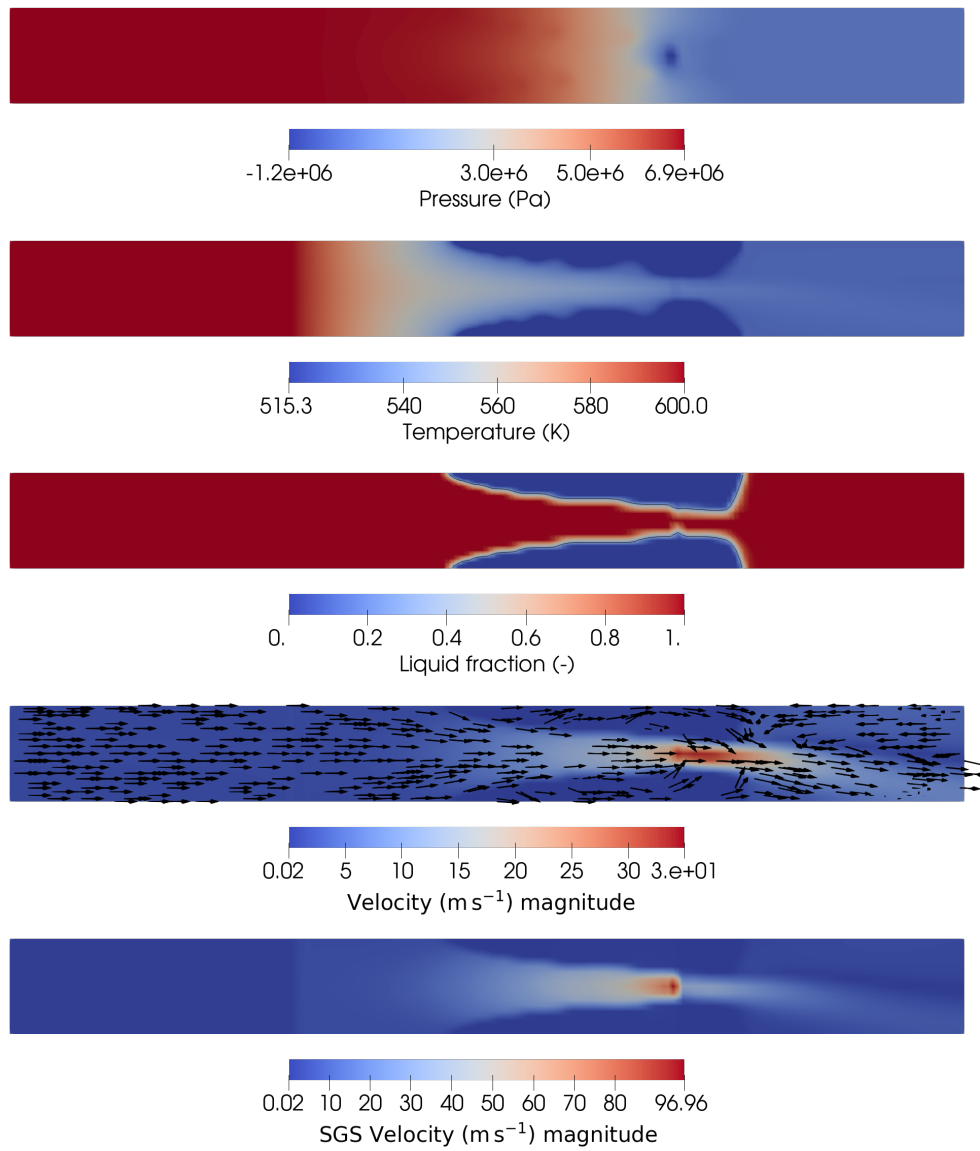


Figure 6.40.: MSFR Freezing HEX fields at $t = 3$ s

simplified, presents relevant results. In total 4 cases were shown: a reference case that does not solidify, and 3 demonstration cases that have difference properties. One has salt properties that are tuned like the reference case to result in a simplified phenomenology, and 2 using MSFR salt properties. One of the MSFR salt cases uses a slip boundary condition to simulate conditions analogous to an inlet with perfect turbulent mixing. The moment of onset of blockage is the critical event, where pressure is highest. All results indicate that the heat exchanger recovers, but it is unlikely that a hypothetical equipment could endure the pressure imposed by the system in cases where solidification is very abrupt.

While there can be many conditions for this, including the heat transfer characteristics of the heat exchanger, at least 2 criteria can be deduced. First, the pump must be able to maintain its momentum despite increase in flow resistance and second, the heat exchanger must allow some degree of cross-flow. The momentum condition of the pump can be achieved for short periods of time by increasing the size of the fly-wheel attached to its axle, without necessarily over-designing its power rating. This is desirable in any case to obtain a grace time for smooth flow decrease in case of a pump trip. Inertia will not keep a pump momentum indefinitely, but it is beneficial at least during the critical moment at the onset of blockage.

The cross-flow conditions for the heat exchanger are not necessarily straightforward. It can be achieved to a certain extent in most designs of compact heat exchanger. In the particular case of a printed circuit heat exchangers, there are designs available where flow is allowed to bypass channel blockage with engineered passages such as ones designed by Alfa Laval to deal with heat exchanger icing [Alfa Laval, 2019].

These broad conclusions are mostly insensitive to changes of salt properties. Such changes result in changes of pressure loss amplitude, however, flow was maintained for the range of values. Considering this insensitivity, salt properties were chosen in such a way as to induce sharpness of the mushy region (i.e.: increase the gradient of the liquid fraction) in order to better elucidate its shape.

Despite the results, a significant amount of study has to be made in order to engineer a mechanism to prevent flow blockage upon freezing using the method shown. In particular, the response to heat transfer characteristics has to be further investigated. More importantly though, while the method shown might prevent the freezing of a heat exchanger due to overcooling of the secondary side, it does not prevent freezing in case of an unprotected pump trip where other measures have to be taken.

The benefits of making a molten salt reactor resistant to flow blockage through cross-flow and pump rotational inertia will have to be balanced by the compromises it requires and alternative options. The increase in heat exchanger size to allow cross-flow or a pump with a large fly-wheel compete with accepting that salt might solidify inside the heat

exchanger in the primary circuit but re-molten by electric heating on the secondary side or some other measure, which might be a desirable or required provision in any case [Allibert et al., 2017, p. 58]. Of interest is that for the heat exchanger to have easier recovery, the fluid should develop a profile at the inlet, but for the heat exchanger to be resistant to solidification, the fluid should be homogeneous.

Given the uncertainties in heat exchanger and reactor design, as shown in the previous section, it might not make sense to use a phase-change model to analyse solidification in the fuel circuit from the perspective of safety, leaving it instead for exploratory studies in the heat exchanger as performed. If we can accept that the heat exchanger will suffer blockage as discussed previously, it might make more sense to simulate the consequences of a complete flow blockage conservatively by imposing appropriate boundary conditions on top and bottom of the heat exchanger, without the need of any phase-change model. Overall, such implicit considerations of solidification in the heat exchanger might lead to a more robust design and safety case, rather than a safety case that models phase-change explicitly but stands on uncertain data.

6.4. Molten Chloride Fast Reactor conceptual design

Usually fast spectrum molten salt reactor cores are completely clear, with only the homogeneous fuel flowing freely inside. One of the potential advantages of fast MSRs is possibility of operating on Breed-and-Burn mode (BNB), achieving high fuel utilisation with an open cycle. Due to the neutron economy required to operate in this fuel cycle, chloride-based MSRs offer significant advantages when implementing this concept.

In light of this potential, this section explores the conceptual design of a single fluid, Molten Chloride Fast Reactor (MCFR) operating on a BNB mode. Studies presenting the fuel-cycle of such reactors in a more detailed way are available [B. A. Hombourger, 2018]. The main driver for a single fluid design is the simplicity and resistance to proliferation that such design offers. Multi-fluid designs compromise on these in exchange for a different set of advantages.

One of the challenges of these fast open-cavity reactors is that legacy tools (e.g., subchannel or system codes) are not capable of modelling such free flow. Therefore, modelling of these reactors use frameworks that are primarily designed for CFD in the context of nuclear reactor multiphysics. [Fiorina et al., 2014; Aufiero, 2014; Bao, 2016; Hu et al., 2017]

This approach presents challenges, because the computational requirements of CFD are much higher than legacy tools. If there is a need to model accurately the flow inside such reactors for safety demonstration, particularly if uncertainty quantification is needed,

significant time and resources are likely to be required. Considering this, we will explore the possibility of shaping the flow in the core using baffles hoping to attain a quasi-1D flow that could be modelled by legacy tools while still preserving the necessary neutron economy for BNB operation. This is an original proposal, specially in the context of fast MSRs [de Oliveira & Hombourger, 2020].

The Serpent Monte Carlo neutron transport code [Leppänen et al., 2015] and the MATLAB-based EQL0D procedure [B. A. Hombourger, 2018; B. Hombourger et al., 2020] were used to obtain critical dimensions and few-groups cross-sections for the equilibrium fuel composition of a BNB cycle chloride-fuelled MSR. EQL0D is a fuel cycle procedure dedicated to MSRs, simulating fuel evolution using a point-like representation of the fuel. Continuous removal of volatile and insoluble fission products and refuelling operations are simulated using user-input removal rates.

All input files and additional information related to this study, such as CAD geometry, are publicly available at GitHub².

6.4.1. Materials and Properties

Considering that this is only a conceptual study, accurate values for properties are not fundamental. The proposed fuel used is 60-40 mol % NaCl and UCl₃ with ²³⁵U enriched to 10.7% and 100% ³⁷Cl as a starting point [B. A. Hombourger, 2018; B. Hombourger et al., 2019]. Table 6.5 shows the parameters used for the fluid, which are on the ballpark of molten salt values but not particular of any. Of these, the most important estimate was of density since it would also impact neutron transport simulations with Serpent, the estimation of the critical core size, and ultimately the geometry. The density value was taken as a molar % linear combination of the pure salts [Beneš & Konings, 2008]. Density variation is modelled using a Boussinesq approximation considering the salt to have an expansion coefficient of $2 \times 10^{-4} \text{ K}^{-1}$ at 923 K.

Table 6.5.: MCFR fluid properties

Parameter	Value
Density	3640 kg m ⁻³
Specific heat capacity	1500 J kg ⁻¹ K ⁻¹
Dynamic viscosity	0.025 Pa s
Prandtl number	15

Structural materials in the core are given the properties of Hasteloy-N without further consideration at this stage. The core reflector used is PbO; a choice that will be justified

²More information at <https://github.com/deOliveira-R/MCFR>

in section 6.4.5.

6.4.2. Core Geometry and Meshing

Figure 6.41 shows the proposed geometry for the reactor core. It does not look much different than a typical reactor, having a cylindrical vessel of 4.8 m diameter as base with a 2:1 height-to-diameter ratio (HDR) after critical core size calculations with Serpent. As can be seen on the Serpent geometry, it also includes a 1 m thick reflector (shown in yellow) that was not included in the geometry to limit the number of cells. This choice of aspect ratio was made to allow the addition of a heat exchanger above the middle level of the core, and reduce pipe length simultaneously. The choice should promote natural circulation during postulated accident scenarios in a planned future work,

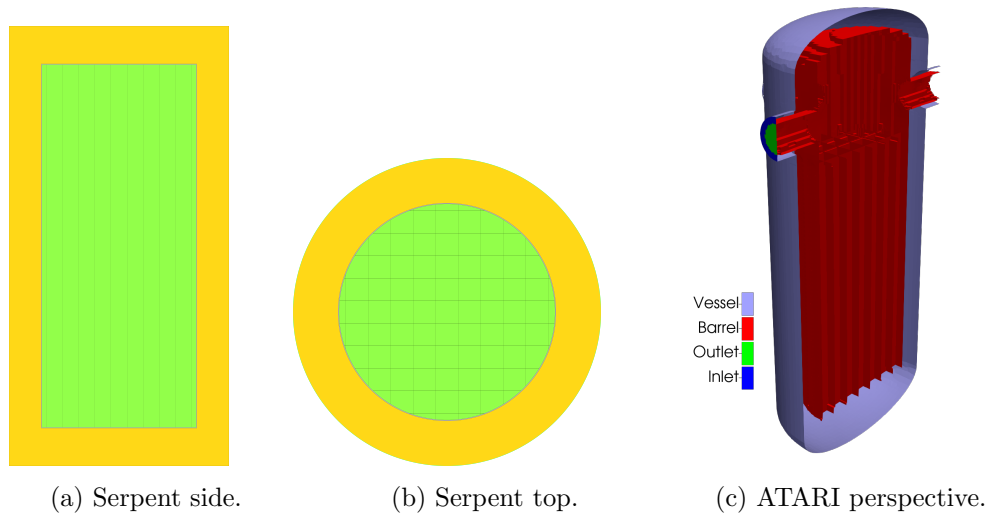


Figure 6.41.: Core geometries with barrel and baffles.

The design borrows some old concepts and introduces new ones. The concentric pipe and core barrel are present in high temperature gas reactors (e.g., GT-MHR), which should allow the temperature of vessel walls to be close to inlet temperature. The rationale is that this would preserve the external parts, which are harder to exchange, and “sacrifice” the internal parts.

In addition to temperature management, radiation damage is another aspect to be considered. It is difficult to estimate how much of a problem radiation damage would be, since there are atom displacements in the crystal lattice due to particle collisions and annealing of defects due to high temperature. Balancing these issues, the baffles do not serve a structural function, having only the purpose of appropriately shaping the flow. Therefore, while strength and ductility are desirable characteristics, they are not necessarily crucial for a baffle that mostly needs to support itself. It is expected

that it will be necessary to exchange the baffles at intervals in any case; either due to corrosion, radiation damage, or the combined effect. In the current design, the baffles have a surface area of approximately 250 m^2 . For a thickness of 1 mm, the baffles would have a mass of approximately 2 tonnes of metal, which is a modest amount to exchange depending on how frequent it should happen.

In order to mesh the CAD geometry, the standard OpenFOAM mesher was used called “snappyHexMesh” (SHM). The resulting mesh contains only about 200 thousand cells.

6.4.3. Boundary Conditions

In Serpent, vacuum boundary conditions are applied after the reflector. In ATARI, vacuum boundary conditions are applied directly after the vessel walls due to Serpent not generating albedos for cylinders. An integral power of 3 GW is given for eigenvalue calculations as a starting point considering the reactor size.

This particular core requires 2 meshes to be simulated, one for the fluid dynamics solver and another for the neutronics one. The reason is that the neutron transport is not guided by the internal barrel and baffles like fluid flow, therefore, the internal boundary should only exist for the fluid simulation.

The boundary conditions for fluid dynamics are presented on table 6.6. Zero gradient boundary conditions means that the gradient normal to the surface is zero. The mass flow rate boundary condition uses the fluid density to calculate the appropriate fluid velocity normal to the inlet surface.

Table 6.6.: MCFR fluid dynamics boundary conditions

Field	Inlets	Outlets	Walls
Pressure	zero gradient	1 bar	zero gradient
Velocity	mass flow rate 3200 kg s^{-1}	zero gradient	no slip
Energy	temperature 923 K	zero gradient	zero gradient
k and ϵ	zero gradient	zero gradient	standard wall functions

6.4.4. Fuel cycle parameters

The starting point were results obtained previously of the BNB cycle equilibrium using the EQL0D procedure. These results are shown in terms of burn-up, fuel salts, and reflector materials [B. Hombourger et al., 2019] shown on table 6.7.

Providing adequate performance, the NaCl-UCl_3 mixture (60-40 mol %) with 100%

Table 6.7.: Candidate chloride fuel salts adapted from [B. Hombourger et al., 2019]

Composition (mol%)	Density (kg m ⁻³)	T_{melt} (K)	Discharge BU (%FIMA)	Critical radius (m)
NaCl–UCl ₃	68-32	3320	793	33.6
NaCl–UCl ₃	60-40	3640	863	33.8
NaCl–UCl ₃ –UCl ₄	70-15-15	3640	773	35
NaCl–UCl ₃ –ThCl ₄	50-25-25	3160	773	35.8
				3.00

enrichment in ³⁷Cl was selected as the most feasible based on critical radius (or inventory) and the lack of UCl₄, which has a low boiling point. A discharge burn-up³ of 33%FIMA was also selected because it leads to minimal critical dimensions at equilibrium.

6.4.5. Reflector and baffles

The possibility of using different reflectors materials was investigated using a 0.92 HDR (optimum according to diffusion theory) and assuming a 1 m thick reflector of a given material.

The critical radius of the core at equilibrium of the BNB cycle with fixed discharge burn-up was compared for several high-temperature candidate reflector materials. PbO was selected as a reflector material due to its acceptable performance and higher melting point than the selected molten salt.

Table 6.8.: Candidate reflectors considered at equilibrium

Reflector	Critical radius (m)	T_{melt} (K)
Lead (reference)	2.50	601
Lead monoxide	2.60	1161
Barium chloride	3.00	1235
Barium oxide	3.25	2196
No reflector	3.50	-

The impact of the baffles on the neutronics was evaluated, leading to a choice of square lattice of 1 mm thin baffles with 50 cm side.

Finally, the case using PbO reflector, baffles and 2:1 HDR was used to generate 8-group cross-sections and 6-group delayed neutron precursor data for core simulation.

³The average burn-up of discharge volume, assuming the reactor is fed with fresh fuel salt which mixes instantaneously with the reactor salt.

6.4.6. Clear and Baffled Cores Comparison

In figure 6.42 we can see the effect that the insertion of baffles has on the flow. Using the eddy viscosity as a measure of turbulence, this design choice reduced the values of this parameter by at least an order of magnitude. Fluid flow has a quasi-1D behaviour as expected. Other fields do not show a significant difference, at least during steady state.

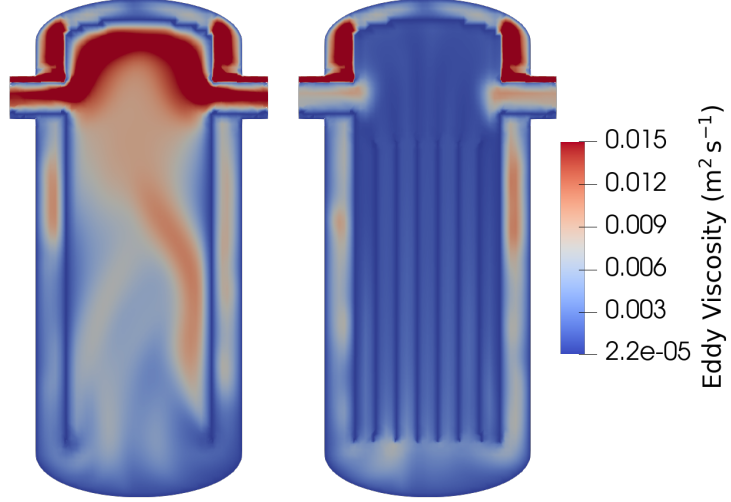


Figure 6.42.: Eddy viscosity in clear (left) and baffled (right) cores.

In the figure, it is possible to appreciate the impact of the mesh on the results. Due to the blocky rough mesh at the inlet, a region of extremely high velocity, turbulence and Courant number is present. Using SHM there are 2 known approaches to mesh the geometry and include the barrel and baffles as internal boundaries. One is to mesh the geometry without the internal boundaries and add them after meshing with the help of the “topoSet” and “createBaffles” utilities. This approach works, but results in the rather blocky barrel shown in figure 6.41c. Another approach would be to generate the mesh with internal boundaries directly. This approach does generate a mesh that is smooth and resolves the curvature of the barrel, however, this mesh also results in a floating point exception during solution of the pressure equation. At the moment, only the first method has been successfully applied. Meshing in SHM is not straightforward, debugging even worse.

6.4.7. Results

Figure 6.43 shows the converged flux (group 1) and volumetric power fields. It is immediately noticeable that they look the same, as they should since convection of decay power is not being modelled at the moment. ATARI calculated a k_{eff} of 0.97640

whereas Serpent a value of 0.97876 for the unreflected core (and 1.00391 ± 0.00039 for the reflected one). A difference of 236 pcm is considered acceptable and expected at this stage where the few-group energy structure has not been optimized.

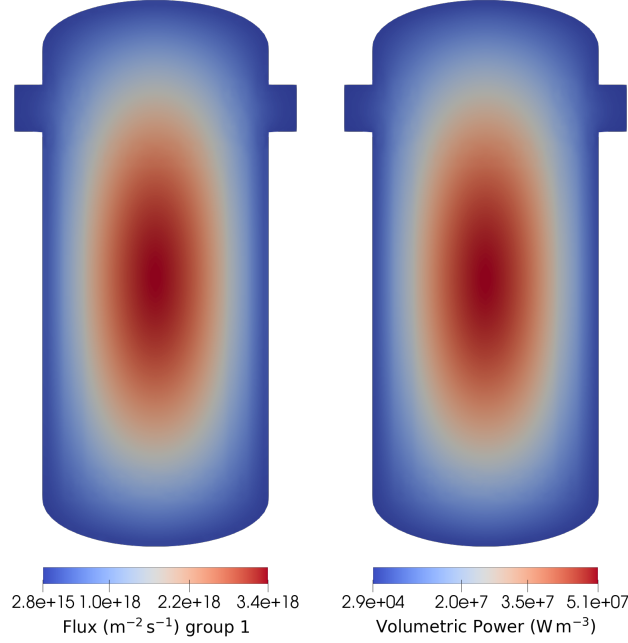


Figure 6.43.: Flux in group 1 and volumetric power in the core.

Figure 6.44 shows the flow velocity in the core, which is rather slow at mostly below 1 m s^{-1} giving this reactor a significant leeway to increase mass flow rate and power. We can also see that the delayed neutron precursors are drifted from their place of origin and that the vessel walls stay close to inlet temperature. Mesh defects again impacts the inlets, affecting the velocity field locally, and consequentially the convection of delayed neutron precursors.

The reactor could be made modular by leaving open the possibility of increasing the power by adding more inlets and outlets. However, there are advantages in terms of safety in having a lower power density as well and these should be considered.

6.4.8. Discussion

Neutronics calculations show that the integration of baffles and the improvement of core geometry for the purpose of improved natural circulation are possible in a Breed-and-Burn reactor, at the cost of a slightly higher inventory of fissile salt due to increased critical radius (from 2.5 m to 2.6 m).

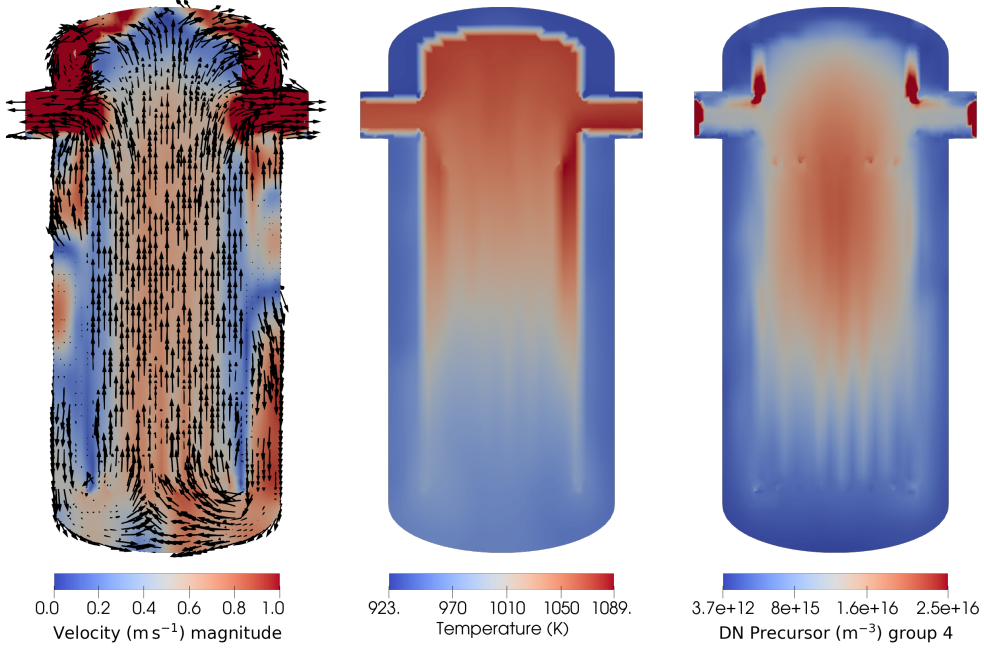


Figure 6.44.: Velocity and velocity-dependent fields of temperature and delayed neutron precursor in group 4.

Fluid dynamics simulation confirms the expectation of a quasi-1D flow in a baffled configuration, opening the possibility to model this core using fast running legacy tools in the future. ATARI would still be useful to analyse very special cases, such as transients with deforming/damaged baffle for example.

Multiphysics simulations demonstrate that the external wall temperature is close to inlet temperature as designed. The results also show the expected behaviour for the simulation including k_{eff} close to criticality, axially elongated flux and volumetric power fields, drift of delayed neutron precursors and the temperature rise in the core; all predicted behaviour.

The simulations with ATARI still presents some issues. From a fluid dynamics aspect, the mesh show problems close to the inlets. This is seen as one of the biggest deficits currently, and generating a smooth curvilinear mesh with internal boundaries directly should be a focus of future developments. From a neutronics aspect, imposing vacuum boundary conditions at the walls is seen as the main issue and proper albedo coefficients must be generated. In order to circumvent Serpent limitations regarding albedos in a cylinder, we envision simplifying the problem to a 2D approximated one or finding the cylinder-equivalent square prism for the case. A criteria for equivalency however, is not yet decided, but buckling is a sound candidate.

This case shows that it is possible to create a “well behaved” core, where flow has

predictable qualities and can be modelled routinely using fast-running legacy tools. The BNB case is used to demonstrate that even a design that requires ultimate neutron economy can still make use of this idea. The suitability of this idea to a reactor is naturally something that has to be considered by the designer among a multitude of other factors. It is understandable that just as some designers might give priority to simplicity of the structures and decide that a clear core with no structures is more advantageous, others might favour simplicity of the flow.

The introduction of flow baffles is also meant to be provocative given the uncertainties of the MSFR highlighted in section 6.2. It is dubious if uncertainties can be made low enough that a robust safety case can be built on it, given their nature (i.e., boundary conditions and RANS turbulence model). Therefore, it is worth reminding the case of the Maples reactors:

This goes to the heart of the issue, says nuclear engineer Jean Koclas at the École Polytechnique in Montreal. “It is not the fact that the coefficient is positive or negative that is a problem. The problem is that you cannot calculate it,” he told the parliamentary committee. “When you find yourself in a situation where you cannot predict as simple a measure as the power coefficient, then can you be sure that the nuclear safety analyses, which are based on calculations, are correct? It was naive to think that such a sensitive coefficient could be calculated within such a high degree of accuracy.”[Alison Motluk, 2010]

Likewise, the velocity field inside the reactor is a rather basic data, but it is unlikely to be known with low uncertainty in a highly turbulent cavity with unknown wall temperatures. A fragile safety case for the MSFR will result from expecting to know the velocity field and all fields that depend on it (e.g., delayed neutron precursor concentration or temperature) with low uncertainty.

The idea presented is merely a proof-of-concept and not meant to be taken at face-value. However, it is worth exploring the design space searching for unorthodox ideas such as this, which can reduce uncertainties to either acceptable or desirable levels.

7. Conclusion

L'arte non è mai finita, ma solo abbandonata.

(Leonardo da Vinci)

The homogenised fluid dynamics model for reactor applications was derived and described in detail in chapter 2. Previously it was scattered between multiple works, lacking terms or containing small mistakes, whereas here its description was consolidated.

ATARI was created based on the homogenised fluid dynamics and neutron diffusion models described. It is not just a redesign, but a bottom-up construction of a multi-physics solver in OpenFOAM. Its modular architecture allows studies into different approaches and models without breaking or invalidating existing working code. The high degree of modularity achieved also allowed the development of a multi-mesh solver to analyse reactors with multiple circuits — a capability not previously available in high-fidelity solvers based on OpenFOAM.

ATARI's capabilities were verified in chapter 5 with a classical phase-change problem with analytical solution and a novel multi-physics problem using the method of manufactured solutions. The novel multi-mesh capability was assessed through energy balances and found to be correct.

The SAMOFAR nuclear cavity benchmark was performed in section 6.1, where ATARI was found to be equivalent to the partners codes within the scope of the benchmark. A few problems in the algorithms were unveiled during the benchmark, demonstrating its value as a diagnosis tool to complement formal verification exercises.

The steady state of the MSFR was analysed with ATARI in section 6.2, where only a small recirculation at core inlet was found to be present. The small recirculation did not result in high temperatures at the wall between core and blanket. The uncertainties in boundary conditions and turbulence model limit the degree of belief in the current analysis.

The steady state MSFR analysis did not progress into a transient analysis as originally intended. Instead, the focus shifted to finding the current limitations of ATARI and OpenFOAM as a framework for reactor analysis,

which was considered a more important and fundamental problem. Coupled neutronics and fluid dynamics solvers in OpenFOAM rely on acceleration schemes to perform transient analysis. ATARI was not capable of performing the transient analysis of the MSFR since the Aitken acceleration scheme is not appropriate for the design.

The niche where codes such as ATARI excel is identified to be challenging simulations where deformation and loss of structure require unstructured mesh capabilities for a flexible description of geometry. Situations not demanding such flexible description of the domain benefit from using codes limited to structured meshes, which offer an unmatched advantage in speed. Codes such as ATARI should be seen as complementary and fill a very special category of simulations that justify the additional computational requirement.

Temporal schemes and transient capabilities of coupled neutronics and fluid dynamics solvers in OpenFOAM cannot be formally verified. What can be currently verified is the temporal scheme in conjunction with the acceleration scheme.

The transient case of the nuclear cavity benchmark from section 6.1 is not representative of the MSFR. The benchmark needs to be supplemented with a transient case that challenges coupling methods and acceleration schemes in a way that is representative of the design characteristics.

Solidification of molten salt in a heat exchanger was investigated in section 6.3. It was found that as long as the heat exchanger allows some degree of cross-flow and pumping power is maintained, solidification was found to reach a stable state with partial blockage. Different models are studied, with different salt properties and different boundary conditions, but all are found to recover if the system can maintain the required conditions. Cases where a flow profile at inlet was present had a softer blockage but it was easier to cause the blockage event. Cases where the inlet was homogenous were very resistant to blockage, but once blockage occurred the pressure went very high. The viability of maintaining such conditions is dubious in cases of abrupt blockage, however cases with abrupt blockage are also the ones with the most resistance of a blockage occurring.

Due to the nature of uncertainties in the MSFR, it is recommended to consider solidification implicitly using conservative approaches, such as blocking the heat exchanger using boundary conditions on top and bottom of the heat exchanger. The use of phase-change models is still a valuable tool for the assessment of heat exchanger characteristics, where explicit modelling of solidification is useful to guide decisions on heat exchanger design.

A fast MSR with baffled core was proposed in section 6.4 with the dual objective of limiting turbulence to manage uncertainties and structure the flow to allow the use of efficient legacy codes for routine analysis. A molten chloride core is used to demonstrate

the value of the concept from a fluid dynamics perspective while preserving breed and burn capability, showing that the effect on neutronics is small and does not invalidate the concept.

The requirement of codes such as ATARI for routine analysis should be taken as a warning. This should be read carefully, as requirement implies that it is not optional, and routine analysis implies a significant portion of analytical workflow. Codes using structured meshes are much faster than codes using unstructured meshes by nature of the mesh layout in memory. If both can give a satisfying answer, the one using structured mesh will be more efficient always; which is something to consider if uncertainty quantification and sensitivity analysis is required. In addition, the requirement of unstructured meshes or turbulence models imply that a reactor sacrificed geometrical and/or flow structure, leading to increased uncertainties. The need for tools such as ATARI for routine analysis might indicate that the design methodology lacked consideration for these issues. While this does not necessarily imply that a reactor design should conform to code capabilities without considering simply further development of the code, it does imply that limits to computational capabilities should be realistically considered. Even if one can argue that everything can be modelled — for which there is no guarantee —, certainly not everything can be modelled with low uncertainty, and not everything can be modelled economically/efficiently.

7.0.1. Future work

An significant effort in extending the verification performed for ATARI must be pursued. The functionalities related to heat exchanger modelling, such as friction correlations, seem particularly urgent.

Rate of convergence and run time should be included as data in future developments of the nuclear cavity benchmarks, since once accuracy is equivalent between participating codes, efficiency becomes the next criterion.

The transient case of the nuclear cavity benchmark is not representative of the MSFR. The integral nature of the variations imposed allow acceleration schemes using integral properties to succeed in the benchmark, while the same schemes fail to perform transients in the MSFR. The benchmark needs to be extended to include transient cases that are challenging enough on coupling methods and acceleration schemes to depend more sophisticated approaches.

Analysis of the MSFR is severely limited by the lack of data in heat exchanger design, where the current one does not allow desired properties to emerge, such as natural circulation. Focus should be given to maturing the heat exchanger design to justify further analysis of the concept.

The boundary condition between core and blanket of the MSFR, and the turbulence models should be a focus of future projects. Current discrepancies make the steady state of the core obviously uncertain, which would result in different dynamics of the system.

The use of homogenised and economical models is identified as an enabler of coupling methods that deal with simultaneous solutions of equations, by managing the cell count and allowing the formation of block matrices of practical size. ATARI needs to be reimplemented in a framework that offers this functionality, such as MOOSE, to allow leveraging the virtues of the method.

A. Derivations

A.1. Stress tensor

The Cauchy stress tensor $\sigma(\nabla u, p)$, referred to simply as stress tensor, completely defines stresses at a point in all 9 directional components shown in figure 1.4. One of the properties of this tensor is that, according to Cauchy's first law of motion applied to a fluid in hydrostatic equilibrium, the tensor takes the form shown in equation A.1.

$$\sigma(p) = -p\delta_{ij} \quad (\text{A.1})$$

This equation implies that:

1. In hydrostatic equilibrium (i.e., $\nabla u = 0$), stresses depend only on pressure.
2. Diagonal tensor components (i.e., normal components) depend on pressure.
3. $\sigma < 0$ denotes compressive stresses.

In hydrostatic equilibrium, shear stresses were absent due to lack of strain rate, leaving only normal stresses due to pressure. This suggests that decomposing the stress tensor of a fluid that is not in hydrostatic equilibrium into a pressure component and a viscous one might be reasonable. We achieve this by first adding and subtracting $p\delta_{ij}$, which is a net 0, as shown in equation A.2.

$$\sigma(\nabla u, p) = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} + \begin{bmatrix} p & 0 & 0 \\ 0 & p & 0 \\ 0 & 0 & p \end{bmatrix} - \begin{bmatrix} p & 0 & 0 \\ 0 & p & 0 \\ 0 & 0 & p \end{bmatrix} \quad (\text{A.2})$$

We then add $\sigma_{ij} + p\delta_{ij}$ into a single term τ_{ij} . By doing so, we counteracted the pressure component in the tensor diagonal, found to be $-p\delta_{ij}$ in the hydrostatic equilibrium, leaving only viscous stresses in τ . The other term $-p\delta_{ij}$ is actually the term related to hydrostatic or thermodynamic pressure as discussed, but the decomposition demonstrated is independent of this previous knowledge. It was just purposefully given to

make this decomposition insightful instead of arbitrary.

$$\begin{aligned}
\sigma(\nabla \mathbf{u}, p) &= \begin{bmatrix} \sigma_{xx} + p & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} + p & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} + p \end{bmatrix} - \begin{bmatrix} p & 0 & 0 \\ 0 & p & 0 \\ 0 & 0 & p \end{bmatrix} \\
&= \begin{bmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \tau_{zz} \end{bmatrix} - \begin{bmatrix} p & 0 & 0 \\ 0 & p & 0 \\ 0 & 0 & p \end{bmatrix} \\
&= \tau - p\delta_{ij}
\end{aligned} \tag{A.3}$$

It is necessary to further develop τ and at this step approximations start to appear. For example, if the fluid is **treated as continuum** and behaves as an **isotropic Newtonian fluid**, the following relation is typically used [Landau & Lifshitz, 1989, p. 45]:

$$\tau = \mu \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T - \frac{2}{3} (\nabla \cdot \mathbf{u}) \delta_{ij} \right) + \kappa (\nabla \cdot \mathbf{u}) \delta_{ij} \tag{A.4}$$

The equation A.4 has, maybe surprisingly, 2 independent coefficients for viscosity:

- μ is the commonly known viscosity, more specifically called dynamic viscosity or first viscosity. It is related to translational motion (i.e., collision) of particles, which is the origin of fluid internal friction.
- κ is rather obscure outside of specific literature, where it is known as second viscosity. It arises due to the finite time necessary for a fluid to reestablish thermodynamic equilibrium when such equilibrium is disturbed [Landau & Lifshitz, 1989, p. 81]. Mechanistically, this represents the time scales for transfer of energy between translational motion/collision (i.e., kinetic energy) to vibrational and rotational motion (i.e., internal energy).

It might be convenient to reorganise equation A.4 where terms are laid out by viscosity mechanism as explained, into a form that focuses on how the mechanisms affects an arbitrary volume of fluid, shown in equation A.5. We do this by grouping together the terms with isotropic stresses $\nabla \cdot \mathbf{u}$ representing volume expansion/compression, consequently isolating terms with strain rate $\nabla \mathbf{u}$ representing volume deformation.

$$\tau = \mu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) + \left(\kappa - \frac{2}{3} \mu \right) (\nabla \cdot \mathbf{u}) \delta_{ij} \tag{A.5}$$

We can conveniently lump together the coefficients for the isotropic terms as a single

value λ :

$$\tau = \mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^T) + \lambda(\nabla \cdot \mathbf{u})\delta_{ij} \quad (\text{A.6})$$

$$\lambda = \kappa - \frac{2}{3}\mu \quad (\text{A.7})$$

Unlike μ where the literature agrees unanimously on the name dynamic viscosity or first viscosity, the name for κ and λ is not standardised¹. Therefore, κ shall be called as second viscosity as mentioned earlier, and λ as bulk viscosity, since it acts on the volume isotropically.

There are multiple possible approximations in this stress tensor. Second viscosity might be important when the fluid is **not in thermodynamic equilibrium** so compression and expansion (i.e., changes of density) happens in time scales comparable to or smaller than that of transfer of kinetic energy to internal energy. Topics such as acoustics, shockwaves and detonations might look like notable candidates for taking second viscosity into consideration. However, even in such topics it will be necessary to judge if shockwave thickness is relevant compared to the blast radius and if it will change this radius in any noticeable way for example. In practice, the so called “Stokes’ hypothesis”, which is essentially an hypothesis of negligible second viscosity or thermodynamic equilibrium, holds true for most cases of interest. An even stronger approximation is to consider the flow as incompressible (i.e., $\nabla \cdot \mathbf{u} = 0$), therefore dropping the term with bulk viscosity entirely. In this study, changes of density are relatively small and time scales large therefore bulk viscosity can be safely ignored.

It is also interesting to consider that the main purpose of the stress tensor $\sigma(\nabla \mathbf{u}, p)$ for fluid dynamics is to introduce it into the equation for the conservation of momentum, where its negative goes into a divergence term $\nabla \cdot (-\sigma)$ expressing diffusion of momentum. Therefore, let us consider this divergence term and substitute τ using the form in equation A.5.

$$\begin{aligned} \nabla \cdot (-\sigma) &= \nabla \cdot (p\delta_{ij} - \tau) \\ &= \nabla \cdot (p\delta_{ij}) - \nabla \cdot (\mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)) - \nabla \cdot \left(\left(\kappa - \frac{2}{3}\mu \right) (\nabla \cdot \mathbf{u})\delta_{ij} \right) \end{aligned} \quad (\text{A.8})$$

Turning our attention to the term with divergence of the pressure-scaled Kronecker

¹It is possible to find κ and λ called as second viscosity and bulk viscosity as in this work, but it is also possible to find them called the inverse [Pletcher et al., 1984, p. 250]. Any name will work as long as the meaning is clear and the modeller knows exactly what he is doing.

delta and using proof A.9, we show that $\nabla \cdot (p\delta_{ij}) = \nabla p$, simplifying equation A.8 into A.10.

$$\begin{aligned}\nabla \cdot (s\delta_{ij}) &= \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{bmatrix} \cdot \left(s \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{bmatrix} \cdot \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & s \end{bmatrix} \\ &= \begin{bmatrix} \frac{\partial s}{\partial x} + 0 + 0 \\ 0 + \frac{\partial s}{\partial y} + 0 \\ 0 + 0 + \frac{\partial s}{\partial z} \end{bmatrix} = \begin{bmatrix} \frac{\partial s}{\partial x} \\ \frac{\partial s}{\partial y} \\ \frac{\partial s}{\partial z} \end{bmatrix} = \nabla s\end{aligned}\quad (\text{A.9})$$

$$\nabla \cdot (-\sigma) = \nabla p - \nabla \cdot (\mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)) - \nabla \cdot \left(\left(\kappa - \frac{2}{3}\mu \right) (\nabla \cdot \mathbf{u}) \right) \quad (\text{A.10})$$

If μ and κ can be considered as constants, they can be taken out of the derivatives and the term with strain rate can be expanded.

$$\begin{aligned}\nabla \cdot (-\sigma) &= \nabla p - \mu \nabla \cdot (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) - \left(\kappa - \frac{2}{3}\mu \right) \nabla (\nabla \cdot \mathbf{u}) \\ &= \nabla p - \mu \nabla^2 \mathbf{u} - \mu \nabla \cdot (\nabla \mathbf{u})^T - \left(\kappa - \frac{2}{3}\mu \right) \nabla (\nabla \cdot \mathbf{u})\end{aligned}\quad (\text{A.11})$$

Finally, by using proof A.12 we show that $\nabla \cdot (\nabla \mathbf{u})^T = \nabla (\nabla \cdot \mathbf{u})$, reducing equation A.11 into A.13.

$$\begin{aligned}\nabla \cdot (\nabla \mathbf{a})^T &= \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{bmatrix} \cdot \left(\begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{bmatrix} \otimes \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \right)^T = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{bmatrix} \cdot \left(\begin{bmatrix} \frac{\partial}{\partial x} a_1 & \frac{\partial}{\partial x} a_2 & \frac{\partial}{\partial x} a_3 \\ \frac{\partial}{\partial y} a_1 & \frac{\partial}{\partial y} a_2 & \frac{\partial}{\partial y} a_3 \\ \frac{\partial}{\partial z} a_1 & \frac{\partial}{\partial z} a_2 & \frac{\partial}{\partial z} a_3 \end{bmatrix} \right)^T \\ &= \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial}{\partial x} a_1 & \frac{\partial}{\partial y} a_1 & \frac{\partial}{\partial z} a_1 \\ \frac{\partial}{\partial x} a_2 & \frac{\partial}{\partial y} a_2 & \frac{\partial}{\partial z} a_2 \\ \frac{\partial}{\partial x} a_3 & \frac{\partial}{\partial y} a_3 & \frac{\partial}{\partial z} a_3 \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} \frac{\partial}{\partial x} a_1 + \frac{\partial}{\partial y} \frac{\partial}{\partial x} a_2 + \frac{\partial}{\partial z} \frac{\partial}{\partial x} a_3 \\ \frac{\partial}{\partial x} \frac{\partial}{\partial y} a_1 + \frac{\partial}{\partial y} \frac{\partial}{\partial y} a_2 + \frac{\partial}{\partial z} \frac{\partial}{\partial y} a_3 \\ \frac{\partial}{\partial x} \frac{\partial}{\partial z} a_1 + \frac{\partial}{\partial y} \frac{\partial}{\partial z} a_2 + \frac{\partial}{\partial z} \frac{\partial}{\partial z} a_3 \end{bmatrix} \\ &= \begin{bmatrix} \frac{\partial}{\partial x} \left(\frac{\partial}{\partial x} a_1 + \frac{\partial}{\partial y} a_2 + \frac{\partial}{\partial z} a_3 \right) \\ \frac{\partial}{\partial y} \left(\frac{\partial}{\partial x} a_1 + \frac{\partial}{\partial y} a_2 + \frac{\partial}{\partial z} a_3 \right) \\ \frac{\partial}{\partial z} \left(\frac{\partial}{\partial x} a_1 + \frac{\partial}{\partial y} a_2 + \frac{\partial}{\partial z} a_3 \right) \end{bmatrix} = \nabla (\nabla \cdot \mathbf{a})\end{aligned}\quad (\text{A.12})$$

$$\begin{aligned}\nabla \cdot (-\sigma) &= \nabla p - \mu \nabla^2 \mathbf{u} - \mu \nabla (\nabla \cdot \mathbf{u}) - \left(\kappa - \frac{2}{3}\mu \right) \nabla (\nabla \cdot \mathbf{u}) \\ &= \nabla p - \mu \nabla^2 \mathbf{u} - \left(\kappa + \frac{1}{3}\mu \right) \nabla (\nabla \cdot \mathbf{u})\end{aligned}\quad (\text{A.13})$$

If the the flow is considered as incompressible, we arrive at the equation that is typically used to model incompressible flow. It is not usually mentioned that it is only strictly valid if μ and κ can be considered as constants, but that is the case as demonstrated.

$$\nabla \cdot (-\sigma) = \nabla p - \mu \nabla^2 \mathbf{u} \quad (\text{A.14})$$

In this work, all conditions for equation A.14 to be valid are satisfied.

A.2. Conservation of total energy

In order to model transport of energy in a fluid it is convenient, and commonly better, to formulate an equation that expresses conservation total energy (i.e., internal plus mechanical energy). It is tempting to try to reach such an equation by deriving an equation for conservation of internal energy and mechanical energy independently of each other and simply adding both, however these 2 quantities are not entirely separable and our wish/need to reach an equation for their sum should be an indication of this. The reason lies in the possibility of converting mechanical energy into thermal energy, which originates from the atomistic nature of matter. Microscopically, thermal energy is nothing more than kinetic energy associated with atomic motion, however that is abstracted by continuum mechanics' macroscopic treatment and somehow we need to recover the macroscopic expression of this property.

While there probably exists an extremely elegant approach to derive a total energy equation from statistical thermodynamics, this will be avoided. It is also unnecessary for understanding of this work. Instead, we will take a rather heuristic approach while pointing out the imperfections that arise from this method. To achieve this, we can start by writing a hypothetical continuity equation for internal energy below.

$$\frac{\partial \rho e}{\partial t} + \nabla \cdot (\rho e \mathbf{u}) + \nabla \cdot \mathbf{q} = S_e \quad (\text{A.15})$$

Where e expresses the specific internal energy of the system and S_e some source of internal energy, such as volumetric heat source Q (e.g., chemical or nuclear heat source) or any other. This equation expresses the typical processes that the continuity equation explicitly describes, such as advection and diffusion, therefore we seem to be missing a term in the source that describes the conversion of mechanic energy that we desire. We shall do the same for mechanical energy and write its continuity equation A.16.

$$\frac{\partial \rho K}{\partial t} + \nabla \cdot (\rho K \mathbf{u}) + \nabla \cdot \mathbf{w} = \rho \mathbf{g} \cdot \mathbf{u} \quad (\text{A.16})$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla \cdot (-\sigma) = \rho \mathbf{g} \quad (\text{A.17})$$

Where $K = \frac{|\mathbf{u}|^2}{2}$, with units of $\text{kg}^2 \text{s}^{-2}$ or equivalently J kg^{-1} , implying that K expresses the specific kinetic energy. The term \mathbf{w} that appears in the equation, formulated as $\mathbf{w} = -\sigma \cdot \mathbf{u}$, seems to have a meaning somewhat analogous to heat flux. If we make a comparison, it seems that equation A.16 is equivalent to the inner product of velocity with equation A.17 for conservation of linear momentum with buoyancy force source.

The deceptively simple term $\mathbf{w} = -\sigma \cdot \mathbf{u}$ obscures a rather complicated meaning, as could be guessed from any term that contains the stress tensor, but Dimensional analysis techniques will uncover it. One way to interpret it is to notice that the stress tensor σ has units of pressure or force per area kg m/s^2 , therefore together with units of velocity m s^{-1} it expresses the force applied on the volume surfaces that causes some displacement per second. This interpretation can be refined by considering that force times displacement is work, therefore $\sigma \cdot \mathbf{u}$ also expresses the work that is done on a surface area per second to cause some displacement. Through this analysis, we conclude that $\sigma \cdot \mathbf{u}$ is work flux that is performed on a closed surface in order to deform the bound volume. Its negative $\mathbf{w} = -\sigma \cdot \mathbf{u}$ expresses the flux of work from regions of higher deformation to regions of lower deformation, or work dissipation/diffusion. This is a meaning for \mathbf{w} that is harmonious with the meaning of heat flux \mathbf{q} as well.

Since we know that \mathbf{w} expresses deformation work flux, the term $\nabla \cdot \mathbf{w}$ in equation A.16 expresses the diffusion of strain energy, the potential energy of deformation. We can decompose this term in 2 ways:

- Through the stress tensor $\sigma = \tau - p\delta_{ij}$, emphasising directionality such as normal or parallel to a cross section plane.
- Through calculus identities, emphasising terms that dissipate through bulk motion or internal energy.

Equation A.18 shows the intermediate step that emphasises different aspects and the total decomposition after applying both methods. The steps are shown organised on table A.1 as well.

$$\begin{aligned} \nabla \cdot \mathbf{w} = \nabla \cdot (-\sigma \cdot \mathbf{u}) &= \begin{cases} \nabla \cdot (p\delta_{ij} \cdot \mathbf{u}) + \nabla \cdot (-\tau \cdot \mathbf{u}) \\ \mathbf{u} \cdot \nabla \cdot (-\sigma) - \sigma : \nabla \mathbf{u} \end{cases} \\ &= p\nabla \cdot \mathbf{u} + \mathbf{u} \cdot \nabla p - \tau : \nabla \mathbf{u} + \mathbf{u} \cdot \nabla \cdot (-\tau) \end{aligned} \quad (\text{A.18})$$

The bulk motion column is familiar since the factor $\nabla \cdot (-\sigma)$ represents diffusion of stresses detailed in the previous appendix A.1, made of a compressive and shearing

Table A.1.: Strain energy decomposition

	$\mathbf{u} \cdot \nabla \cdot (-\sigma)$ bulk motion	$-\sigma : \nabla \mathbf{u}$ internal energy
$\nabla \cdot (p\delta_{ij} \cdot \mathbf{u})$ normal	$\mathbf{u} \cdot \nabla p$	$p\nabla \cdot \mathbf{u}$
$\nabla \cdot (-\tau \cdot \mathbf{u})$ parallel	$\mathbf{u} \cdot \nabla \cdot (-\tau)$	$-\tau : \nabla \mathbf{u}$

components. The column representing internal energy is the “missing link” to the internal energy equation and the reason why a clean separation between mechanical and internal energy is not possible.

The row representing normal components contain propagation of force imbalance and compression/expansion work. These are at least partially reversible, with a component transported through bulk motion and a component stored as internal energy. The row of parallel components representing shear deformation is irreversible work dissipated either as bulk motion of eddies² or shear heating through fluid internal friction. The presence of reversible and irreversible components further emphasises the character of \mathbf{w} as work flux, since work can never be fully converted into kinetic energy without losses.

The individual terms represent:

- $\mathbf{u} \cdot \nabla p$ is the bulk motion of force imbalance, where an extreme example is the propagation of a shock wave.
- $p\nabla \cdot \mathbf{u}$ is the compression/expansion work, such as in a gas compressor or turbine and adiabatic compression during atmospheric reentry.
- $\mathbf{u} \cdot \nabla \cdot (-\tau)$ is the bulk motion of eddies/vortices, such as high velocity turbulent flow around obstacles in aerodynamic breaking.
- $-\tau : \nabla \mathbf{u}$ is the viscous dissipation of deformation work as heat through fluid internal friction, such as in pumps/kitchen blender and skin friction during atmospheric reentry.

Ironically, it turns out that it was possible to reach an equation for total energy A.19 by naively adding the incomplete internal energy equation A.15 and the equation for mechanical energy A.16 containing the missing internal energy term. What would have turned out to be wrong would be to use the incomplete equation for conservation of

²In an elastic solid, the work of shear deformation would be recoverable, however a fluid does not store shear as potential energy since the fluid deforms continuously.

internal energy by itself.

$$\begin{aligned} & \frac{\partial \rho e}{\partial t} + \nabla \cdot (\rho e \mathbf{u}) + \nabla \cdot \mathbf{q} \\ & + \frac{\partial \rho K}{\partial t} + \nabla \cdot (\rho K \mathbf{u}) + \nabla \cdot \mathbf{w} = S_e + \rho \mathbf{g} \cdot \mathbf{u} \end{aligned} \quad (\text{A.19})$$

It is usually convenient to rewrite the total energy equation using specific enthalpy h instead of specific internal energy e . This can be done using the equivalency $h = e + \frac{p}{\rho}$, which expresses enthalpy as the sum of internal energy and the work required to open space for the system to occupy its appropriate volume. We also expand the scope of the internal energy source S_e into a generalised one S_{he} .

$$\begin{aligned} & \frac{\partial \rho h}{\partial t} - \frac{\partial}{\partial t} \rho \frac{p}{\rho} + \nabla \cdot (\rho h \mathbf{u}) - \nabla \cdot \left(\rho \frac{p}{\rho} \mathbf{u} \right) + \nabla \cdot \mathbf{q} \\ & + \frac{\partial \rho K}{\partial t} + \nabla \cdot (\rho K \mathbf{u}) + \nabla \cdot \mathbf{w} = S_{he} + \rho \mathbf{g} \cdot \mathbf{u} \end{aligned} \quad (\text{A.20})$$

If we partially decompose $\nabla \cdot \mathbf{w} = \nabla \cdot (-\sigma \cdot \mathbf{u})$ using calculus identities as shown in equation A.3, some simplification will be possible.

$$\begin{aligned} & \frac{\partial \rho h}{\partial t} - \frac{\partial}{\partial t} \rho \frac{p}{\rho} + \nabla \cdot (\rho h \mathbf{u}) - \nabla \cdot \left(\rho \frac{p}{\rho} \mathbf{u} \right) + \nabla \cdot \mathbf{q} \\ & + \frac{\partial \rho K}{\partial t} + \nabla \cdot (\rho K \mathbf{u}) + \nabla \cdot (-\tau \cdot \mathbf{u}) + \nabla \cdot (p \delta_{ij} \cdot \mathbf{u}) = S_{he} + \rho \mathbf{g} \cdot \mathbf{u} \end{aligned} \quad (\text{A.21})$$

The term $\nabla \cdot (p \delta_{ij} \cdot \mathbf{u})$ can be simplified to $\nabla \cdot (p \mathbf{u})$ as shown in proof A.9 and canceled with the equivalent negative that originated from the enthalpy substitution, leading to the final form of the equation for conservation of total energy expressed with specific enthalpy.

$$\begin{aligned} & \frac{\partial \rho h}{\partial t} - \frac{\partial p}{\partial t} + \nabla \cdot (\rho h \mathbf{u}) + \nabla \cdot \mathbf{q} \\ & + \frac{\partial \rho K}{\partial t} + \nabla \cdot (\rho K \mathbf{u}) + \nabla \cdot (-\tau \cdot \mathbf{u}) = \rho \mathbf{g} \cdot \mathbf{u} + S_{he} \end{aligned} \quad (\text{A.22})$$

For all cases of interest in this work, the term expressing irreversible work losses through viscous forces $\nabla \cdot (-\tau \cdot \mathbf{u})$ is negligible from either a modelling or uncertainty perspective, and therefore ignored.

B. MSFR nuclear data

The nuclear data for the MSFR analysis was generated with Serpent 2 Monte-Carlo neutron transport code [Leppänen et al., 2015] in a 3D geometry, at 900 K, using JEFF-3.1.1 library.

Table B.1.: MSFR 6 energy group structure and inverse neutron speed

Group	Energy (MeV) Upper bound	$\frac{1}{V_j}$ (cm ⁻¹ s)
1	20	$3.981\,14 \times 10^{-10}$
2	2.231	$7.520\,35 \times 10^{-10}$
3	4.979×10^{-1}	$2.687\,98 \times 10^{-9}$
4	2.479×10^{-2}	$6.620\,39 \times 10^{-9}$
5	5.531×10^{-3}	$1.494\,69 \times 10^{-8}$
6	7.485×10^{-4}	$3.657\,57 \times 10^{-8}$

Table B.2.: MSFR delayed neutron precursor data

Group	β_g	λ_g (s ⁻¹)
1	$1.229\,58 \times 10^{-4}$	$1.246\,67 \times 10^{-2}$
2	$7.144\,91 \times 10^{-4}$	$2.829\,17 \times 10^{-2}$
3	$3.596\,36 \times 10^{-4}$	$4.252\,44 \times 10^{-2}$
4	$7.940\,58 \times 10^{-4}$	$1.330\,42 \times 10^{-1}$
5	$1.474\,06 \times 10^{-3}$	$2.924\,67 \times 10^{-1}$
6	$5.145\,17 \times 10^{-4}$	$6.664\,88 \times 10^{-1}$
7	$1.229\,58 \times 10^{-4}$	1.634 78
8	$7.144\,91 \times 10^{-4}$	3.554 60

Table B.3.: MSFR albedo coefficients

Group	Core-Reflector	Core-Blanket
1	$3.736\,47 \times 10^{-1}$	$3.886\,50 \times 10^{-1}$
2	$1.473\,28 \times 10^{-1}$	$2.217\,96 \times 10^{-1}$
3	$6.297\,90 \times 10^{-2}$	$9.674\,90 \times 10^{-2}$
4	$8.078\,71 \times 10^{-2}$	$6.757\,40 \times 10^{-2}$
5	$4.559\,60 \times 10^{-3}$	$3.355\,40 \times 10^{-2}$
6	$-1.119\,07 \times 10^{-1}$	$-6.981\,00 \times 10^{-2}$

Table B.4.: MSFR prompt and delayed neutron spectrum, and fission energy yield

Group	$\chi_{p,j}$	$\chi_{d,j}$	κ_j (J)
1	$3.606\,73 \times 10^{-1}$	$4.645\,04 \times 10^{-3}$	$3.304\,41 \times 10^{-11}$
2	$5.171\,57 \times 10^{-1}$	$3.939\,07 \times 10^{-1}$	$3.314\,52 \times 10^{-11}$
3	$1.206\,43 \times 10^{-1}$	$5.753\,88 \times 10^{-1}$	$3.299\,25 \times 10^{-11}$
4	$1.368\,68 \times 10^{-3}$	$2.306\,94 \times 10^{-2}$	$3.294\,01 \times 10^{-11}$
5	$1.514\,93 \times 10^{-4}$	$2.674\,09 \times 10^{-3}$	$3.293\,18 \times 10^{-11}$
6	$7.457\,16 \times 10^{-6}$	$3.162\,59 \times 10^{-4}$	$3.294\,88 \times 10^{-11}$

B.1. Fuel cross-sections

Table B.5.: Fuel cross-sections

Group	D_j (cm)	$\Sigma_{t,j}$ (cm ⁻¹)	$\bar{\nu}_j \Sigma_{f,j}$ (cm ⁻¹)
1	2.314 56	$2.062\,47 \times 10^{-1}$	$1.367\,77 \times 10^{-2}$
2	1.539 90	$2.613\,99 \times 10^{-1}$	$7.206\,38 \times 10^{-3}$
3	0.976 00	$3.684\,88 \times 10^{-1}$	$4.886\,12 \times 10^{-3}$
4	1.180 04	$2.923\,72 \times 10^{-1}$	$6.996\,12 \times 10^{-3}$
5	1.102 61	$3.113\,60 \times 10^{-1}$	$1.387\,05 \times 10^{-2}$
6	1.026 29	$3.342\,89 \times 10^{-1}$	$3.726\,00 \times 10^{-2}$

Table B.6.: Fuel cross-sections parametrization coefficient α

Group	D_j (cm)	$\Sigma_{t,j}$ (cm ⁻¹)	$\bar{\nu}_j \Sigma_{f,j}$ (cm ⁻¹)
1	$-1.181\,86 \times 10^{-3}$	$2.085\,64 \times 10^{-5}$	$-4.518\,88 \times 10^{-6}$
2	$3.823\,67 \times 10^{-4}$	$-6.952\,12 \times 10^{-6}$	$-7.994\,94 \times 10^{-7}$
3	$-3.476\,06 \times 10^{-4}$	$1.077\,58 \times 10^{-4}$	$1.042\,82 \times 10^{-7}$
4	$-3.927\,95 \times 10^{-3}$	$9.246\,32 \times 10^{-4}$	$2.294\,20 \times 10^{-6}$
5	$-1.390\,42 \times 10^{-2}$	$3.827\,14 \times 10^{-3}$	$-1.911\,83 \times 10^{-5}$
6	$-2.346\,34 \times 10^{-2}$	$7.664\,71 \times 10^{-3}$	$-2.773\,90 \times 10^{-4}$

Table B.7.: Fuel P₀ scattering matrix $\Sigma_{s,i \leftarrow j}$

group \ j i	1	2	3	4	5	6
1	$1.355\,47 \times 10^{-1}$	0	0	0	0	0
2	$5.805\,42 \times 10^{-2}$	$2.204\,16 \times 10^{-1}$	0	0	0	0
3	$6.157\,42 \times 10^{-3}$	$3.765\,56 \times 10^{-2}$	$3.503\,94 \times 10^{-1}$	0	0	0
4	$4.172\,87 \times 10^{-5}$	$6.430\,06 \times 10^{-5}$	$1.411\,78 \times 10^{-2}$	$2.718\,93 \times 10^{-1}$	0	0
5	$4.293\,99 \times 10^{-6}$	$5.358\,27 \times 10^{-6}$	$2.365\,85 \times 10^{-5}$	$1.275\,35 \times 10^{-2}$	$2.901\,87 \times 10^{-1}$	0
6	$3.617\,95 \times 10^{-7}$	$2.384\,91 \times 10^{-7}$	$1.490\,37 \times 10^{-6}$	$2.403\,35 \times 10^{-6}$	$5.692\,98 \times 10^{-3}$	$2.955\,10 \times 10^{-1}$

Table B.8.: Fuel P₀ scattering matrix $\Sigma_{s,i \leftarrow j}$ parametrization coefficient α

group \ j i	1	2	3	4	5	6
1	$-1.008\,06 \times 10^{-4}$	0	0	0	0	0
2	$1.046\,29 \times 10^{-4}$	$-2.085\,64 \times 10^{-5}$	0	0	0	0
3	$2.735\,66 \times 10^{-5}$	$1.529\,47 \times 10^{-5}$	$9.732\,97 \times 10^{-5}$	0	0	0
4	$-9.611\,30 \times 10^{-7}$	$-1.317\,08 \times 10^{-6}$	$2.780\,85 \times 10^{-6}$	$8.863\,95 \times 10^{-4}$	0	0
5	$3.700\,61 \times 10^{-7}$	$2.341\,13 \times 10^{-7}$	$-3.163\,21 \times 10^{-8}$	$-1.842\,31 \times 10^{-5}$	$3.559\,48 \times 10^{-3}$	0
6	$-1.416\,49 \times 10^{-7}$	$-3.172\,60 \times 10^{-8}$	$1.353\,93 \times 10^{-7}$	$-3.229\,95 \times 10^{-7}$	$-1.585\,43 \times 10^{-4}$	$5.241\,90 \times 10^{-3}$

B.2. Heat Exchanger cross-sections

Table B.9.: HEX cross-sections

Group	D_j (cm)	$\Sigma_{t,j}$ (cm^{-1})	$\bar{\nu}_j \Sigma_{f,j}$ (cm^{-1})
1	2.314 56	$2.062\,47 \times 10^{-1}$	$1.367\,77 \times 10^{-2}$
2	1.539 90	$2.613\,99 \times 10^{-1}$	$7.206\,38 \times 10^{-3}$
3	0.976 00	$3.684\,88 \times 10^{-1}$	$4.886\,12 \times 10^{-3}$
4	1.180 04	$2.923\,72 \times 10^{-1}$	$6.996\,12 \times 10^{-3}$
5	1.102 61	$3.113\,60 \times 10^{-1}$	$1.387\,05 \times 10^{-2}$
6	1.026 29	$3.342\,89 \times 10^{-1}$	$3.726\,00 \times 10^{-2}$

Table B.10.: HEX cross-sections parametrization coefficient α

Group	D_j (cm)	$\Sigma_{t,j}$ (cm^{-1})	$\bar{\nu}_j \Sigma_{f,j}$ (cm^{-1})
1	$-1.181\,86 \times 10^{-3}$	$2.085\,64 \times 10^{-5}$	$-4.518\,88 \times 10^{-6}$
2	$3.823\,67 \times 10^{-4}$	$-6.952\,12 \times 10^{-6}$	$-7.994\,94 \times 10^{-7}$
3	$-3.476\,06 \times 10^{-4}$	$1.077\,58 \times 10^{-4}$	$1.042\,82 \times 10^{-7}$
4	$-3.927\,95 \times 10^{-3}$	$9.246\,32 \times 10^{-4}$	$2.294\,20 \times 10^{-6}$
5	$-1.390\,42 \times 10^{-2}$	$3.827\,14 \times 10^{-3}$	$-1.911\,83 \times 10^{-5}$
6	$-2.346\,34 \times 10^{-2}$	$7.664\,71 \times 10^{-3}$	$-2.773\,90 \times 10^{-4}$

Table B.11.: HEX P_0 scattering matrix $\Sigma_{s,i \leftarrow j}$

group \ j i	1	2	3	4	5	6
1	$1.355\,47 \times 10^{-1}$	0	0	0	0	0
2	$5.805\,42 \times 10^{-2}$	$2.204\,16 \times 10^{-1}$	0	0	0	0
3	$6.157\,42 \times 10^{-3}$	$3.765\,56 \times 10^{-2}$	$3.503\,94 \times 10^{-1}$	0	0	0
4	$4.172\,87 \times 10^{-5}$	$6.430\,06 \times 10^{-5}$	$1.411\,78 \times 10^{-2}$	$2.718\,93 \times 10^{-1}$	0	0
5	$4.293\,99 \times 10^{-6}$	$5.358\,27 \times 10^{-6}$	$2.365\,85 \times 10^{-5}$	$1.275\,35 \times 10^{-2}$	$2.901\,87 \times 10^{-1}$	0
6	$3.617\,95 \times 10^{-7}$	$2.384\,91 \times 10^{-7}$	$1.490\,37 \times 10^{-6}$	$2.403\,35 \times 10^{-6}$	$5.692\,98 \times 10^{-3}$	$2.955\,10 \times 10^{-1}$

Table B.12.: HEX P_0 scattering matrix $\Sigma_{s,i \leftarrow j}$ parametrization coefficient α

group \ j i	1	2	3	4	5	6
1	$-1.008\,06 \times 10^{-4}$	0	0	0	0	0
2	$1.046\,29 \times 10^{-4}$	$-2.085\,64 \times 10^{-5}$	0	0	0	0
3	$2.735\,66 \times 10^{-5}$	$1.529\,47 \times 10^{-5}$	$9.732\,97 \times 10^{-5}$	0	0	0
4	$-9.611\,30 \times 10^{-7}$	$-1.317\,08 \times 10^{-6}$	$2.780\,85 \times 10^{-6}$	$8.863\,95 \times 10^{-4}$	0	0
5	$3.700\,61 \times 10^{-7}$	$2.341\,13 \times 10^{-7}$	$-3.163\,21 \times 10^{-8}$	$-1.842\,31 \times 10^{-5}$	$3.559\,48 \times 10^{-3}$	0
6	$-1.416\,49 \times 10^{-7}$	$-3.172\,60 \times 10^{-8}$	$1.353\,93 \times 10^{-7}$	$-3.229\,95 \times 10^{-7}$	$-1.585\,43 \times 10^{-4}$	$5.241\,90 \times 10^{-3}$

C. Personal experiences

This appendix is a supplement to the thesis containing thoughts and recommendations of the author. Scientific rigour and writing style give way to a more personal message when convenient.

They are presented in hopes that, although based on the author's observation and perspective, the message might prevent unnecessary frustration or improve the overall quality of work.

C.1. Best practices on programming

Every design starts in a blank sheet of paper, and programs are not exceptions. It is necessary to reflect on how to organise data, compartmentalise the program in blocks (e.g., neutronics, fluid dynamics, solvers, etc) and interfaces to exchange data with these blocks. Computer programs are algorithms, that is, structured sets of information and procedures to solve a problem. Programs typically become unstructured and confusing by cutting short the design process and coding it without a clear vision of structure in mind. The clearer the vision, the clearer the resulting code.

Even without proper strategies in place to emphasise structure, one might get away with a simple executable that, at least apparently, solves the problem. That is how many scientific codes start. However, there is a trap embedded in this thinking. It is tempting to skip testing and pretty much everything else related to Software Quality Assurance (SQA).

These problems - lack of structure and SQA - are hardly surprising. It is merely a reflection of the “principle of least effort” and “instant gratification”, our human nature. Everything that is not least effort and that delays gratification requires increasing amount of motivation, patience and discipline. Research and academia in general do not demand this, or worse, the push to publish articles fast effectively incentivises “quick and dirty” behaviour. Therefore, motivation for proper coding in this environment is primarily internal.

I (the author) am a human as well! I understand very well how least effort and instant

gratification works, and how we tend to create excuses to yield to it. Nothing can be done about our human nature; it is what it is. We can only work our way around it by trying to find something even stronger to overcome it.

This section will attempt to expose the main points of why code structure, with a focus on Object-Oriented Programming, and SQA, with a focus on testing and version control, are important. More than just contributing to the quality of the end result, they help us to become better scientists. If self-improvement as a scientist/engineer/mathematician is one's objective, there is a case to apply these techniques in code development as a tool to diagnose one's own gap in knowledge.

C.1.1. Code structure and Object-Oriented Programming

Once a design is clear enough to start programming, the program will grow in size. This will be a relatively slow process so it might not be immediately obvious but, at some point, a code to solve a non-trivial problem will become big. The bigger the code, the bigger the need to clearly identify structures and organise it. **Writing large code bases is, first and foremost, an exercise in managing complexity!** In order to organise code, computer scientists have come up with many paradigms. One of the most import ones, and the one the author hears the most criticism about, is object-oriented programming (OOP).

A significant amount of criticism to OOP originates from people that did not spend the time to understand it. Senior scientists that spent most of their career using Fortran language, which did not have support for it for a long time, are a good example. Since they did not use it, and yet could solve their problems, it is understandable not seeing the need for it. However, it is important to realise that scientists typically write code to solve their own problems, do not have formal education in programming besides the scientific perspective and have limited or no liability for external users. Under these circumstances, code readability and maintenance become a low priority.

Regardless of criticism, object-oriented programming is fundamental for large code bases. The top programming languages, as well as most modern commercial programs are object-oriented as evidence of this. If most modern professional codes use this paradigm, one that does not understands the need for it has to take a step back and reflect. It is unlikely that most professional programmers are incompetent or masochists.

The idea of OOP is simple: group data and procedures that fundamentally related. Let us put the concept in context, using a financial transaction as an analogy.

Someone (probably the author!) wants to buy a beer at a kiosk. The person grabs the beer and goes to the cashier to pay for it. It's time for the transaction. The person

takes the wallet, opens it, takes only the money to pay the beer and hands it to the cashier. Done. Beer time!

The wallet is our example of an object that follows the concept of object-oriented programming. The choice of it was intentional because the wallet is a object that everyone behaves as they should in OOP, since it contains money and sensitive information that must be carefully handled. From a wallet it is typically possible to get at least money, credit cards and an ID. For this reason, nobody would pay for the beer by handing the wallet to the cashier. One would open the wallet and hand only the necessary content. If a stranger would ask for something inside your wallet, you would only give the necessary, if necessary. If a stranger asks how much money you have inside of it, you would not say it, unless it was necessary. Inside a computer program, “everyone is a stranger”. By defining objects, one immediately implements the concept of boundaries of trust and access. In addition, the limits of interaction are defined by the objects’ procedures. One does not asks the wallet to cool the beer, because it cannot do that. That is a procedure that belongs to the fridge object. Objects might contain other objects, for example, the “object” person might contain objects wallet, car keys, pistol and police badge.

Existing physical objects are usually straightforward to understand the virtual implementation. However the capabilities of OOP to organise code really appears when one comprehends how data and procedures are related and how these are organised as objects. For example, a geometrical domain could be an object that contains vertices points and have procedures to provide distance between points, surfaces and volumes. A finite difference method solver might be an object that asks the domain object for deltas between points and has procedures to calculate finite differences, store matrices and solves problems given a numerical solver.

The concept should be sound until now. Criticism regarding OOP typically arises not due to these concepts, but due to subclasses, which are essentially the concept of object specialisations. For example, maybe a specialised subclass of wallet has a hidden compartment that would allow one to hide a bill in case the visible money was stolen, or the a specialised wallet could block RFID attempts to read the information from the ID contained inside. On the first, the specialised object might have extra data that is hidden (i.e. hidden money), while the second might have limited procedures to get ID info from inside (i.e. ID cannot be read by some RFID scanner). In this context, the wallet class is a “base” class to these specialised wallet subclasses. Maybe the finite difference method solver is a subclass specialisation of a discrete solver base class, and this discrete solver might have other specialisations available such as finite volume method solver or finite element method solver.

Subclassing is a system to eliminate code duplication. All subclasses of wallets store money but might have slightly different extra data or procedures. All subclasses of discrete solvers need to ask information to a domain, store a matrix and solve a problem,

but do so differently. Here lies the criticism to OOP. To understand a subclass, it might be necessary to understand multiple parent classes, and this is not trivial after a few layers.

While this frustration is understandable, it is also necessary to understand why it became like that and what would be the consequence of not using OOP. This criticism is a result of arising complexity in the structure of the program, and while organising the program in objects seem complex, organising the same program without objects is much worse and prone to errors simultaneously! **Every growing program will become progressively more complex — one cannot eliminate this complexity, only manage it by implementing good strategies.** Without objects, every variable, procedure and flag is a moving piece in the program. By grouping into objects, the number of moving pieces is drastically reduced. This is the fundamental reason for why OOP is so common in professional programming where program size is typically large.

However, there are very good reasons for using OOP even in small programs, such as:

1. there might be little or no subclasses in small program, therefore the original criticism becomes invalid.
2. cleverly employed OOP strongly emphasises program structure and, even for small programs, significantly increases readability.
3. if the program is written from the beginning using OOP, it will not have to be reprogrammed from scratch once it reaches a size where the lack of it becomes very penalising for continued growth.

Of these, item 2 is particularly important. During the design of the the program, the appearance of object patterns is a good indication that the program structure is solidifying because a significant understanding data-procedures relationship is required for these objects to arise. **Not being able to see object patterns arising during code design indicates a gap in knowledge of data-procedures relationship.** In the context of scientific programming, recognising object patterns require clear understanding of the different topics of science and mathematics that are necessary to solve the problem (e.g., particles, materials, domain, discretisation, numerical schemes, solvers, etc) and the boundary in between.

This is the “selfish” case for good programming practices. Even if one cares about nothing but oneself, good programming practices will serve as a self-diagnosis tool to identify knowledge gaps.

C.1.2. Managing code development and Software Quality Assurance

During the programming of a scientific software there are 2 main tasks related to SQA that are essential for sustainable growth of the code: testing and version control.

Testing consists of checking if the program performs a set of tasks as expected. Verification and Validation are tests, for example. One tests mathematical correctness, while the other tests model adequacy. Testing is an essential part of code development because every piece of code written has a purpose, and it's important to ensure its strict fulfilment. **Not being able to figure out exactly how to test a code indicates a gap in knowledge of what the code is expected to do.** When developing codes, its testing should be considered simultaneously as it will influence design to an extent. By not taking testing into consideration during design and development, it is completely possible, and even likely, to end up with code that is cumbersome to test. This is particularly true for objects that contain other objects.

For example, when testing a domain object that is supposed to return surface values, all domain points can be given and surfaces calculated with arbitrary precision. When testing a discrete solver that relies on this domain, it will be necessary to employ techniques and strategies specific to it (e.g., Order Of Accuracy tests), and it might be necessary to give this solver object a domain object that has been already tested or a dummy object. Without a good strategy, this can become cumbersome after a few layers of object dependency.

When a code grows to a size where testing during development becomes non-trivial, it is a good idea to consider investing some time into making a testing harness. This utility, maybe a script or an auxiliary testing program, should execute tests with high degree of automation. It will be of great value by highlighting unintentional flaws and differences in results that inevitably occur during development.

Version Control System (VCS) is a tool to manage code development, having 2 key features:

1. Traceability: VCS allows safe return to any point in the development history.
2. Safe development environment: VCS separates code that is known to be working (and hopefully tested!) from developing (and probably flawed) one.

The first allows the user to create snapshots at any point during the development and return to these states on demand. In a large program it is entirely possible for a flaw to be introduced and pass unnoticed by the testing harness, unless testing coverage is very high. Going back in history might allow finding the source of the problem and fix it.

The second involves a tree system that allows one to create timelines with different

purposes. Tested working code, untested code under development, and maybe even some experimental ideas that are not clear yet how they will progress, are all possible timelines of development that can interact. VCS allows one to keep all these safe without the creation of a myriad of folders, where the user is prone to lose track of.

VCS is absolutely necessary for any code developer. Without it, any sort of verification and validation becomes easily questionable. When any test of the code can be questioned, so can all the results it provides. Any degree of belief in the code effectively evaporates.

This has profound implications. Research done with external resources (i.e., money) is not a hobby. It is a professional activity, with associated responsibilities where professionals are accountable for. While in research there has to be a certain degree of freedom in order to allow creative innovation, this is not an excuse for sloppiness.

VCS uses actually go much further than just code development since it can be applied to any text-based file, which happens to be the vast majority of input files for scientific code. Therefore VCS can and should be used during the development of such input files. When VCS is used on a code and its input file simultaneously, it is possible to achieve **perfect reproducibility of results**.

In order to correctly use VCS, it is necessary to establish a workflow. There are enough examples of workflow on the internet to satisfy or inspire anyone. The development of ATARI follows a standard workflow called “Git Flow”. Whatever workflow is chosen, follow it with discipline. The benefits will be felt in the very first unnoticed “bug” — the first of many.

C.1.3. The paradox of happiness – building a mountain with strangers

Hopefully, enough arguments have been made to warm the reader to the idea of following good development practices. Focus was given to topics the author considers particularly problematic or prone to misunderstanding, but more recommendations can be found in specialised literature [Oberkampff & Roy, 2010, p. 757]. However, some might not be convinced by these arguments. One can still be concerned that implementing all these practices takes a toll on the speed of development. It is true. Proper design, testing and management requires time, specially at the beginning. To this concern, the author would ask; What is the hurry? Let us distance ourselves from the methods and look at an overarching topic of managing research.

It is attractive to believe the result of one’s research will have an impact. Maybe change the world! If one can do more, maybe he will have more impact. Having such optimism is an important source of motivation. At the same time, healthy skepticism and keeping expectations real is also important to avoid needless disappointment. Most likely, one’s

results won't change the world. Research is an area where individual results have a low chance of making any difference. Even when one person seemingly had a massive impact, it is possible to observe that it was a culmination of a mountain of research. It just so happens that the one that puts the first pebbles on the base or the last pebble on the top tends to get the glory for the whole mountain. However, it is still only the compound results of many people's work that creates the mountain regardless. It is natural that in the process a lot goes unnoticed.

High expectations are a good source of motivation but also disappointment. Low expectations are a terrible source of motivation but minimises disappointment, which is commonly considered the key to happiness. Right there is the paradox of happiness and motivation. Somehow, one should have the enthusiasm of changing the world while not being disappointed that most likely it won't happen. The answer to this paradox is personal, but the only way the author found to re-conciliate motivation with low expectation was to focus on the process of research, not the results. Enjoy the process of putting pebbles and making a mountain with others, most of them strangers.

One should realise that it is true that implementing these practices takes time. However, this process is necessary to obtain reliable results. While skipping this process might allow one to have more results, without this process, results are likely to be unreliable and meaningless anyway! For the result of a research to have an impact, it must outgrow the original researcher and be used by others. Others are more likely to have an interest in meaningful reliable results. Therefore, one needs to exercise this balancing act of focusing on process and results. Your pebble is only useful when it becomes footing for you or someone else to step on and put a higher pebble, so make it a solid footing for the climb.

Bibliography

- Alexiades, V., & Solomon, A. D. (1993). *Mathematical modelling of melting and freezing processes*. Hemisphere Pub. Corp.
- Alfa Laval. (2019, July). *Printed Circuit Heat Exchanger product leaflet*. Retrieved from https://www.alfalaval.com/globalassets/documents/products/heat-transfer/plate-heat-exchangers/pche/printed-circuit-heat-exchanger_product-leaflet_en.pdf
- Ali, Y. M., & Zhang, L. C. (2005, June). Relativistic heat conduction. *International Journal of Heat and Mass Transfer*, 48(12), 2397–2406. doi: doi:10.1016/j.ijheatmasstransfer.2005.02.003
- Alison Motluk. (2010). What killed the Maples reactors? *New Scientist*, 205(2743), 30–33. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0262407910601212>
- Allibert, M., Gérardin, D., Heuer, D., Huffer, E., Laureau, A., Merle, E., ... Tiberi, V. (2017, February). *D1.1 Description of initial reference design and identification of safety aspects* (Tech. Rep.). SAMOFAR.
- Alnæs, M., Blechta, J., Hake, J., Johansson, A., Kehlet, B., Logg, A., ... Wells, G. N. (2015, December). The FEniCS Project Version 1.5. *Archive of Numerical Software*, 3(100), 9–23. doi: 10.11588/ans.2015.100.20553
- Aufiero, M. (2014). *Development of Advanced Simulation tools for Circulating-fuel Nuclear Reactors* (Doctoral Thesis, Politecnico di Milano, Milan, Italy). Retrieved 2017-08-11, from <https://www.politesi.polimi.it/handle/10589/89364>
- Bao, J. (2016). *Development of the model for the multi-physics analysis of Molten Salt Reactor Experiment using GeN-Foam code* (Unpublished master’s thesis). EPFL-ETHZ.
- Beneš, O., & Konings, R. (2008, April). Thermodynamic evaluation of the NaCl–MgCl₂–UCl₃–PuCl₃ system. *Journal of Nuclear Materials*, 375(2), 202–208. Retrieved 2019-10-30, from <https://linkinghub.elsevier.com/retrieve/pii/S002231150800010X> doi: 10.1016/j.jnucmat.2008.01.007

- Box, G. E. P. (1976, December). Science and Statistics. *Journal of the American Statistical Association*, 71(356), 791–799. Retrieved 2019-08-14, from <http://www.tandfonline.com/doi/abs/10.1080/01621459.1976.10480949> doi: 10.1080/01621459.1976.10480949
- Brovchenko, M., Merle-Lucotte, E., Rouch, H., Alcaro, F., Allibert, M., Aufiero, M., ... Yamaji, B. (2013, September). *D2.2 Optimization of the pre-conceptual design of the MSFR.pdf* (Tech. Rep.). EVOL.
- Cacuci, D. G. (Ed.). (2010). *Handbook of Nuclear Engineering*. Springer Science & Business Media. Retrieved 2017-04-03, from <http://books.google.com/books?hl=en&lr=&id=pu9BWuf2gdkC&oi=fnd&pg=PR1&dq=%22nuclear+engineers+needs+to+be+educated,+a+generation+which+is+to+succeed+the%22+%22for+energy,+needs+for+competences,+needs+for+education+and+training,+needs+for%22+&ots=v0JMb0VbBX&sig=vpCWkjrw101X0xXVJQASqmHtWE>
- Cammi, A., Cervi, E., Křepel, J., Lathouwers, D., LeCarpentier, D., Lorenzi, S., ... Wang, S. (2019, August). *D4.4 Analysis of accidental transients* (Tech. Rep.). SAMOFAR.
- Cartland-Glover, G., Skillen, A., Litskevich, D., Rolfo, S., Emerson, D. R., Merk, B., & Moulinec, C. (2019, December). On the numerical modelling of frozen walls in a molten salt fast reactor. *Nuclear Engineering and Design*, 355, 110290. Retrieved 2019-11-09, from <https://linkinghub.elsevier.com/retrieve/pii/S0029549319303073> doi: 10.1016/j.nucengdes.2019.110290
- Chung, T. J. (2010). *Computational Fluid Dynamics* (2nd ed.). Cambridge, UK: Cambridge University Press. Retrieved from <https://www.cambridge.org/core/books/computational-fluid-dynamics/5C396317EE111C5ED1192FA7F8853944>
- Dahmani, M., Phelps, B., & Shen, W. (2011, November). Verification and validation of the flux reconstruction method for CANDU® applications. *Annals of Nuclear Energy*, 38(11), 2410–2416. Retrieved 2017-09-18, from <http://linkinghub.elsevier.com/retrieve/pii/S0306454911002714> doi: 10.1016/j.anucene.2011.07.011
- Date, A. W. (2005). *Introduction to Computational Fluid Dynamics*. Cambridge, UK: Cambridge University Press.
- de Oliveira, R. G., & Hombourger, B. A. (2020, March). Fuel tap: a simplified breed and burn MSR. In *Transition to a Scalable Nuclear Future*. Cambridge, UK.
- de Oliveira, R. G., & Mikityuk, K. (2018, November). Analytical solutions to a coupled fluid dynamics and neutron transport problem with application to GeN-Foam verification. *Annals of Nuclear Energy*, 121, 446–451. Retrieved 2018-09-03, from <https://linkinghub.elsevier.com/retrieve/pii/S0306454918303852> doi: 10.1016/j.anucene.2018.07.036

- de Souza Sánchez Filho, E. (2016). *Tensor Calculus for Engineers and Physicists*. Cham: Springer International Publishing. Retrieved 2021-01-24, from <http://link.springer.com/10.1007/978-3-319-31520-1> doi: 10.1007/978-3-319-31520-1
- Drew, D. A. (1983). Mathematical Modeling of Two-Phase Flow. *Annual Review of Fluid Mechanics*, 15, 261–291.
- Drew, D. A., & Passman, S. L. (1999). *Theory of multicomponent fluids* (No. 135). New York: Springer.
- 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. Retrieved 2017-03-26, from <http://linkinghub.elsevier.com/retrieve/pii/S0029549315003829> doi: 10.1016/j.nucengdes.2015.05.035
- Fiorina, C., Lathouwers, D., Aufiero, M., Cammi, A., Guerrieri, C., Kloosterman, J. L., ... Ricotti, M. E. (2014, February). Modelling and analysis of the MSFR transient behaviour. *Annals of Nuclear Energy*, 64, 485–498. Retrieved 2016-02-28, from <http://linkinghub.elsevier.com/retrieve/pii/S0306454913004118> doi: 10.1016/j.anucene.2013.08.003
- Fleisch, D. (2011). *A Student's Guide to Vectors and Tensors*. Cambridge: Cambridge University Press. Retrieved 2017-04-01, from <http://ebooks.cambridge.org/ref/id/CB09781139031035> doi: 10.1017/CBO9781139031035
- Ganapol, B. D. (2008). *Analytical Benchmarks for Nuclear Engineering Applications* (No. NEA/DB/DOC(2008)1). OECD. Retrieved 2017-09-05, from <https://www.oecdnea.org/databank/docs/2008/db-doc2008-1.pdf>
- Gaston, D., Newman, C., Hansen, G., & Lebrun-Grandié, D. (2009, October). MOOSE: A parallel computational framework for coupled systems of nonlinear equations. *Nuclear Engineering and Design*, 239(10), 1768–1778. Retrieved 2017-03-27, from <http://linkinghub.elsevier.com/retrieve/pii/S0029549309002635> doi: 10.1016/j.nucengdes.2009.05.021
- Ghetta, V., Giraud, J., Rubiolo, P., Tano-Retamales, M., & Lorenzi, S. (2019, July). *D3.7 Recommendations for the MSFR shut down and DHR passive system* (Tech. Rep.). SAMOFAR.
- GIF. (2008). *2008 Annual Report* (Annual Report). Generation IV International Forum. Retrieved from https://www.gen-4.org/gif/jcms/c_44720/annual-reports
- GIF. (2015). *2015 Annual Report* (Annual Report). Generation IV International Forum. Retrieved from https://www.gen-4.org/gif/jcms/c_44720/annual-reports

- Haag, G., Mindermann, D., Wilhelmi, G., Persicke, H., & Ulsamer, W. (1990, April). Development of reactor graphite. *Journal of Nuclear Materials*, 171(1), 41–48. Retrieved 2020-12-09, from <https://linkinghub.elsevier.com/retrieve/pii/002231159090345N> doi: 10.1016/0022-3115(90)90345-N
- Hedayat, A. (2016, April). Benchmarking verification of the control rod effects on the MTR core parameters using the MTR-PC and MCNP codes throughout 3D core modeling and rod-drop experiment. *Progress in Nuclear Energy*, 88, 183–190. Retrieved 2016-02-28, from <http://linkinghub.elsevier.com/retrieve/pii/S0149197015301190> doi: 10.1016/j.pnucene.2015.12.001
- Hombourger, B., Křepel, J., & Pautz, A. (2019). Breed-and-Burn fuel cycle in Molten Salt Reactors. *European Journal of Physics - Nuclear, In Press*.
- Hombourger, B., Křepel, J., & Pautz, A. (2020, September). The EQL0D fuel cycle procedure and its application to the transition to equilibrium of selected molten salt reactor designs. *Annals of Nuclear Energy*, 144, 107504. Retrieved 2020-07-14, from <https://linkinghub.elsevier.com/retrieve/pii/S0306454920302024> doi: 10.1016/j.anucene.2020.107504
- Hombourger, B. A. (2018). *Conceptual Design of a Sustainable Waste Burning Molten Salt Reactor* (Doctoral Thesis, EPFL, Lausanne, Switzerland). Retrieved from <https://infoscience.epfl.ch/record/255939?ln=en>
- Hopkins, M. M., Moffat, H. K., Carnes, B., Hooper, R. W., & Pawlowski, R. P. (2007, November). *Final report on LDRD project : coupling strategies for multi-physics applications*. (Tech. Rep. Nos. SAND2007-7146, 1029797). Retrieved 2021-06-22, from <http://www.osti.gov/servlets/purl/1029797> doi: 10.2172/1029797
- Hu, T., Cao, L., Wu, H., Du, X., & He, M. (2017, November). Coupled neutronics and thermal-hydraulics simulation of molten salt reactors based on OpenMC/TANSY. *Annals of Nuclear Energy*, 109, 260–276. Retrieved 2018-09-03, from <https://linkinghub.elsevier.com/retrieve/pii/S0306454916309975> doi: 10.1016/j.anucene.2017.05.002
- Hutchinson, I. H. (2015). *A Student's Guide to Numerical Methods*. Cambridge, UK: Cambridge University Press.
- Hébert, A. (2020). *Applied Reactor Physics* (3rd ed.). Montréal, Canada: Presses internationales Polytechnique.
- IEA. (2019, May). *Nuclear Power in a Clean Energy System* (Tech. Rep.). OECD. Retrieved from <https://webstore.iea.org/nuclear-power-in-a-clean-energy-system>

- IEA. (2020a, September). *Energy Technology Perspectives 2020* (Tech. Rep.). Paris, France: OECD. Retrieved from <https://www.iea.org/reports/energy-technology-perspectives-2020>
- IEA. (2020b, June). *Tracking Power 2020* (Tech. Rep.). Paris, France: OECD. Retrieved from <https://www.iea.org/reports/tracking-power-2020>
- IEA. (2020c, October). *World Energy Outlook 2020* (Tech. Rep.). Paris, France: IEA. Retrieved from <https://www.iea.org/reports/world-energy-outlook-2020>
- Jeevanjee, N. (2015). *An Introduction to Tensors and Group Theory for Physicists* (2nd ed.). Cham: Springer. Retrieved 2021-01-24, from <http://link.springer.com/10.1007/978-3-319-14794-9> doi: 10.1007/978-3-319-14794-9
- Kennedy, C. A., & Carpenter, M. H. (2003, January). Additive Runge–Kutta schemes for convection–diffusion–reaction equations. *Applied Numerical Mathematics*, 44(1-2), 139–181. Retrieved 2021-08-01, from <https://linkinghub.elsevier.com/retrieve/pii/S0168927402001381> doi: 10.1016/S0168-9274(02)00138-1
- Knupp, P., & Salari, K. (2002). *Verification of Computer Codes in Computational Science and Engineering* (Vol. 20023155). Chapman and Hall/CRC. Retrieved 2018-03-27, from <https://www.taylorfrancis.com/books/9781420035421> doi: 10.1201/9781420035421
- Konings, R. J. M. (2012). *Comprehensive nuclear materials*. Elsevier.
- Kundu, P. K., Cohen, I. M., Dowling, D. R., & Tryggvason, G. (2016). *Fluid Mechanics* (Sixth edition ed.). Amsterdam ; Boston: Elsevier/AP.
- Landau, L. D., & Lifshitz, E. M. (1989). *Fluid Mechanics* (2nd ed., Vol. 6). Oxford, UK: Pergamon Press.
- Lani, A., Villedie, N., Bensassi, K., Koloszar, L., Vymazal, M., Yalim, S. M., & Panesi, M. (2013, June). COOLFluid: an open computational platform for multi-physics simulation and research. American Institute of Aeronautics and Astronautics. Retrieved 2017-12-22, from <http://arc.aiaa.org/doi/10.2514/6.2013-2589> doi: 10.2514/6.2013-2589
- Lathouwers, D., Cervi, E., Lorenzi, S., de Oliveira, R. G., Tiberga, M., Rubiolo, P., ... Laureau, A. (2018, September). *D4.3 Verification and Validation studies* (Tech. Rep.). SAMOFAR.
- Laureau, A., Heuer, D., Merle-Lucotte, E., Rubiolo, P., Allibert, M., & Aufiero, M. (2017, May). Transient coupled calculations of the Molten Salt Fast Reactor using the Transient Fission Matrix approach. *Nuclear Engineering and Design*, 316, 112–124. Retrieved 2017-09-19, from <http://linkinghub.elsevier.com/retrieve/pii/S002954931730081X> doi: 10.1016/j.nucengdes.2017.02.022

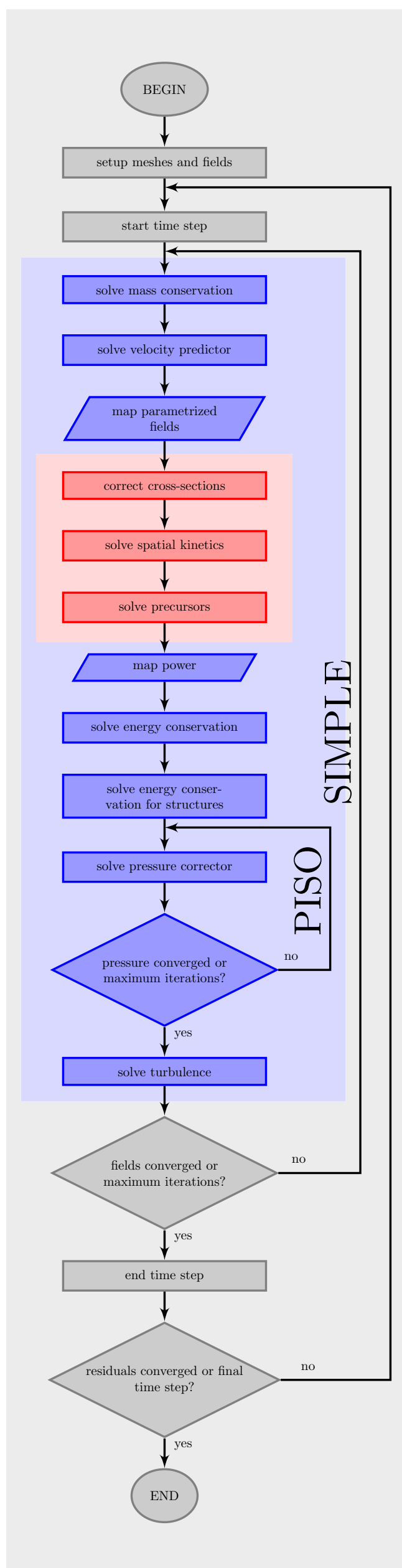
- Leppänen, J., Pusa, M., Viitanen, T., Valtavirta, V., & Kaltiaisenaho, T. (2015, August). The Serpent Monte Carlo code: Status, development and applications in 2013. *Annals of Nuclear Energy*, 82, 142–150. Retrieved 2016-02-28, from <http://linkinghub.elsevier.com/retrieve/pii/S0306454914004095> doi: 10.1016/j.anucene.2014.08.024
- Loftus, P. J., Cohen, A. M., Long, J. C. S., & Jenkins, J. D. (2015, January). A critical review of global decarbonization scenarios: what do they tell us about feasibility?: A critical review of global decarbonization scenarios. *Wiley Interdisciplinary Reviews: Climate Change*, 6(1), 93–112. Retrieved 2019-11-02, from <http://doi.wiley.com/10.1002/wcc.324> doi: 10.1002/wcc.324
- Ma, Z., Yue, N., Zheng, M., Hu, B., Su, G., & Qiu, S. (2015, February). Basic verification of THACS for sodium-cooled fast reactor system analysis. *Annals of Nuclear Energy*, 76, 1–11. Retrieved 2016-02-28, from <http://linkinghub.elsevier.com/retrieve/pii/S0306454914005040> doi: 10.1016/j.anucene.2014.09.025
- MacPherson, H. G. (1985). The molten salt reactor adventure. *Nuclear Science and engineering*, 90(4), 374–380.
- Malicki, M., Pieńkowski, L., Skolik, K., & Trivedi, A. (2019, June). Development of MELCOR thermal hydraulic model of AP1000 and its verification for a DECL break. *Annals of Nuclear Energy*, 128, 44–52. Retrieved 2019-11-02, from <https://linkinghub.elsevier.com/retrieve/pii/S0306454918307059> doi: 10.1016/j.anucene.2018.12.041
- Mathieu, L., Heuer, D., Merle-Lucotte, E., Brissot, R., Le Brun, C., Liatard, E., ... Lecarpentier, D. (2009, January). Possible Configurations for the Thorium Molten Salt Reactor and Advantages of the Fast Nonmoderated Version. *Nuclear Science and Engineering*, 161(1), 78–89. Retrieved 2017-11-30, from <https://www.tandfonline.com/doi/full/10.13182/NSE07-49> doi: 10.13182/NSE07-49
- McClarren, R. G. (2018). *Computational Nuclear Engineering and Radiological Science Using Python*. Elsevier. Retrieved 2021-01-23, from <https://linkinghub.elsevier.com/retrieve/pii/C20160035076> doi: 10.1016/C2016-0-03507-6
- McKEON, B. J., Zagarola, M. V., & Smits, A. J. (2005, August). A new friction factor relationship for fully developed pipe flow. *Journal of Fluid Mechanics*, 538(-1), 429. Retrieved 2020-12-08, from http://www.journals.cambridge.org/abstract_S0022112005005501 doi: 10.1017/S0022112005005501
- Merle, E. (2017, May). *Molten Salt Reactors (MSR)* [Webinar]. Retrieved from https://www.gen-4.org/gif/jcms/c_82831/webinars
- Moukalled, F., Mangani, L., & Darwish, M. (2016). *The Finite Volume Method in Computational Fluid Dynamics* (Vol. 113). Cham: Springer International Publishing.

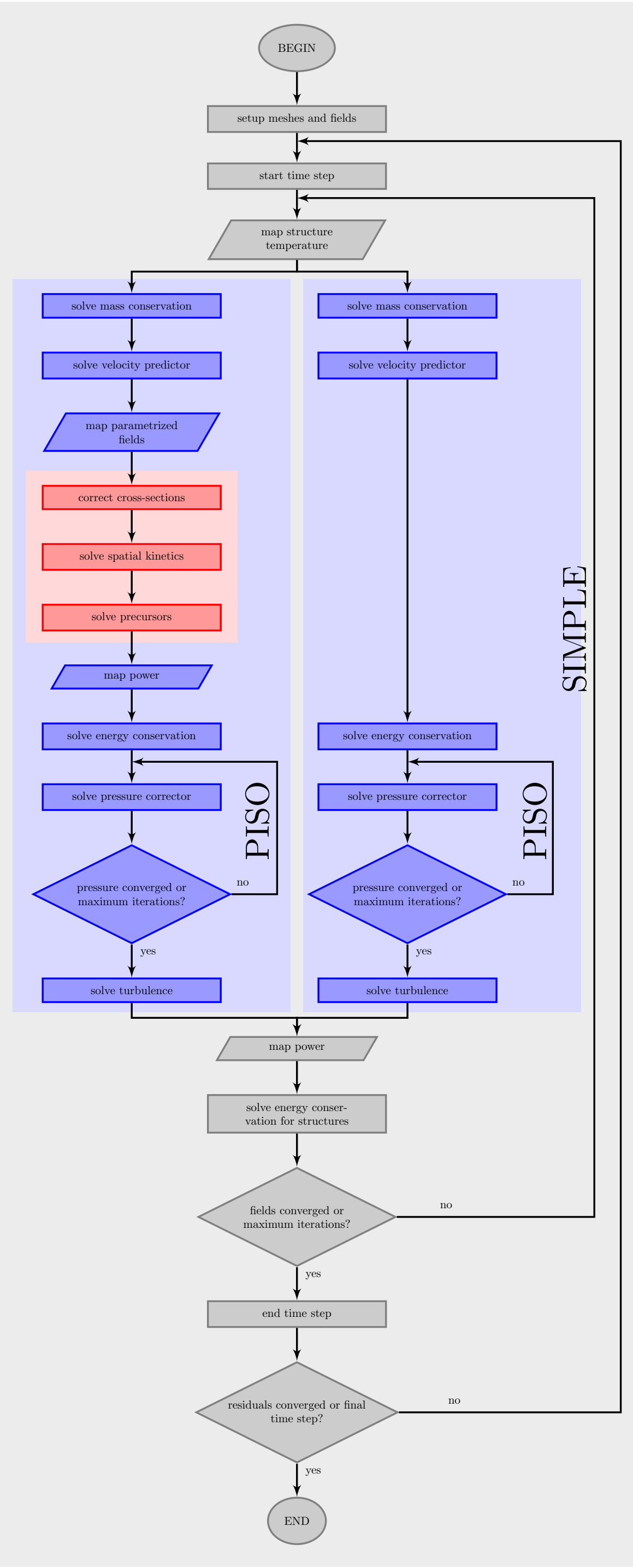
- Retrieved 2017-05-11, from <http://link.springer.com/10.1007/978-3-319-16874-6> doi: 10.1007/978-3-319-16874-6
- NEA. (2019, January). *The Costs of Decarbonisation: System Costs with High Shares of Nuclear and Renewables* (Tech. Rep.). OECD. Retrieved 2019-11-02, from https://www.oecd-ilibrary.org/nuclear-energy/the-costs-of-decarbonisation_9789264312180-en doi: 10.1787/9789264312180-en
- NEA, & IAEA. (2020, December). *Uranium 2020: Resources, Production and Demand* (Tech. Rep.). Paris, France: OECD NEA. Retrieved from https://www.oecd-nea.org/jcms/pl_52718/uranium-2020-resources-production-and-demand
- Oberkampff, W. L., & Roy, C. J. (2010). *Verification and Validation in Scientific Computing*. Cambridge: Cambridge University Press. Retrieved 2017-10-17, from <http://ebooks.cambridge.org/ref/id/CBO9780511760396> doi: 10.1017/CBO9780511760396
- Oberkampff, W. L., Trucano, T. G., & Pilch, M. M. (2003, August). *On the role of code comparisons in verification and validation*. (Tech. Rep. Nos. SAND2003-2752, 918244). Retrieved 2018-06-13, from <http://www.osti.gov/servlets/purl/918244-tbRwVC/> doi: 10.2172/918244
- Pautz, S. D. (2001, March). *Verification of Transport Codes by the Method of Manufactured Solutions: The ATTLA Experience* (Tech. Rep. No. LA-UR-01-1487). Los Alamos, USA: Los Alamos National Laboratory. Retrieved from <http://lib-www.lanl.gov/la-pubs/00796073.pdf>
- Pettersen, E. E. (2016). *Coupled multi-physics simulations of the Molten Salt Fast Reactor using coarse-mesh thermal-hydraulics and spatial neutronics* (Master's thesis, Université Paris-Saclay, Paris, France). Retrieved from <https://www.psi.ch/sites/default/files/import/fast/PublicationsEN/FB-DOC-16-016.pdf>
- Pletcher, R. H., Tannehill, J. C., & Anderson, D. A. (1984). *Computational Fluid Mechanics and Heat Transfer* (3rd ed.). CRC press Boca Raton. Retrieved 2017-05-11, from <http://www.crcnetbase.com/doi/pdf/10.1201/b10158-1>
- Pope, S. B. (2000). *Turbulent Flows*. Cambridge University Press.
- Ragusa, J. C., & Mahadevan, V. S. (2009, March). Consistent and accurate schemes for coupled neutronics thermal-hydraulics reactor analysis. *Nuclear Engineering and Design*, 239(3), 566–579. Retrieved 2017-03-27, from <http://linkinghub.elsevier.com/retrieve/pii/S0029549308005645> doi: 10.1016/j.nucengdes.2008.11.006
- Roache, P. J. (1998). *Verification and validation in computational science and engineering*. Albuquerque, USA: Hermosa Publishers.

- Rouch, H., Geoffroy, O., Rubiolo, P., Laureau, A., Brovchenko, M., Heuer, D., & Merle-Lucotte, E. (2014, February). Preliminary thermal-hydraulic core design of the Molten Salt Fast Reactor (MSFR). *Annals of Nuclear Energy*, 64, 449–456. Retrieved 2016-02-28, from <http://linkinghub.elsevier.com/retrieve/pii/S0306454913004829> doi: 10.1016/j.anucene.2013.09.012
- SAMOFAR. (2015). *A Paradigm Shift in Reactor Safety with the Molten Salt Fast Reactor*. Retrieved from samofar.eu
- Sanchez, R., & McCormick, N. J. (1982, April). A Review of Neutron Transport Approximations. *Nuclear Science and Engineering*, 80(4), 481–535. Retrieved 2019-11-03, from <https://www.tandfonline.com/doi/full/10.13182/NSE80-04-481> doi: 10.13182/NSE80-04-481
- Saurel, R., & Abgrall, R. (1999, April). A Multiphase Godunov Method for Compressible Multifluid and Multiphase Flows. *Journal of Computational Physics*, 150(2), 425–467. Retrieved 2018-12-10, from <http://linkinghub.elsevier.com/retrieve/pii/S0021999199961879> doi: 10.1006/jcph.1999.6187
- Saurel, R., & Lemetayer, O. (2001, March). A multiphase model for compressible flows with interfaces, shocks, detonation waves and cavitation. *Journal of Fluid Mechanics*, 431, 239–271. Retrieved 2018-12-10, from http://www.journals.cambridge.org/abstract_S0022112000003098 doi: 10.1017/S0022112000003098
- Schunert, S., & Azmy, Y. Y. (2011, May). A two-dimensional method of manufactured solutions benchmark suite based on variations of Larsen’s benchmark with escalating order of smoothness of the exact solution. Rio de Janeiro, Brazil.
- Sepulveda, N. A., Jenkins, J. D., de Sisternes, F. J., & Lester, R. K. (2018, November). The Role of Firm Low-Carbon Electricity Resources in Deep Decarbonization of Power Generation. *Joule*, 2(11), 2403–2420. Retrieved 2019-11-02, from <https://linkinghub.elsevier.com/retrieve/pii/S2542435118303866> doi: 10.1016/j.joule.2018.08.006
- Shen, K., Huang, Z.-H., Shen, W., Yang, J., Yang, G., Yu, S., & Kang, F. (2015, November). Homogenous and highly isotropic graphite produced from mesocarbon microbeads. *Carbon*, 94, 18–26. Retrieved 2020-12-09, from <https://linkinghub.elsevier.com/retrieve/pii/S0008622315005497> doi: 10.1016/j.carbon.2015.06.034
- Shwageraus, E. (2003). *Rethinking the LWR Fuel Cycle* (Doctoral Thesis, Massachusetts Institute of Technology. Department of Engineering., Cambridge, United States of America) Retrieved 2016-08-09, from <http://dspace.mit.edu/handle/1721.1/75166>
- Stacey, W. M. (2018). *Nuclear Reactor Physics* (3rd ed.). Weinheim: Wiley-VCH.

- Swaminathan, C. R., & Voller, V. R. (1992, October). A general enthalpy method for modeling solidification processes. *Metallurgical Transactions B*, 23(5), 651–664. Retrieved 2018-12-06, from <http://link.springer.com/10.1007/BF02649725> doi: 10.1007/BF02649725
- The OpenFOAM Foundation. (2013, March). *OpenFOAM 2.2.0: Matrix Solvers*. Retrieved from <https://openfoam.org/release/2-2-0/matrix-solvers/>
- Tiberga, M., de Oliveira, R. G. G., Cervi, E., Blanco, J. A., Lorenzi, S., Aufiero, M., ... Rubiolo, P. (2020, July). Results from a multi-physics numerical benchmark for codes dedicated to molten salt fast reactors. *Annals of Nuclear Energy*, 142, 107428. Retrieved 2020-07-14, from <https://linkinghub.elsevier.com/retrieve/pii/S0306454920301262> doi: 10.1016/j.anucene.2020.107428
- Tiberga, M., Lathouwers, D., & Kloosterman, J. L. (2020, September). A multi-physics solver for liquid-fueled fast systems based on the discontinuous Galerkin FEM discretization. *Progress in Nuclear Energy*, 127, 103427. Retrieved 2020-10-18, from <https://linkinghub.elsevier.com/retrieve/pii/S0149197020301797> doi: 10.1016/j.pnucene.2020.103427
- Uroić, T. (2019). *Implicitly Coupled Finite Volume Algorithms* (Doctoral Thesis). University of Zagreb, Zagreb, Croatia.
- US NRC. (2017). *TRACE V5.0 PATCH 5 USER'S MANUAL*. U.S. Nuclear Regulatory Commission.
- Versteeg, H. K., & Malalasekera, W. (2007). *An Introduction to Computational Fluid Dynamics - The Finite Volume Method* (2nd ed.). Harlow: Pearson/Prentice Hall. (OCLC: 263715061)
- Voller, V. R., Cross, M., & Markatos, N. C. (1987, January). An enthalpy method for convection/diffusion phase change. *International Journal for Numerical Methods in Engineering*, 24(1), 271–284. Retrieved 2018-12-06, from <http://doi.wiley.com/10.1002/nme.1620240119> doi: 10.1002/nme.1620240119
- Voller, V. R., & Prakash, C. (1987). A fixed grid numerical modelling methodology for convection-diffusion mushy region phase-change problems. *International Journal of Heat and Mass Transfer*, 30(8), 1709–1719.
- Wang, J., Martin, W., & Collins, B. (2016, May). Application of the method of manufactured solutions to the 1D Sn equation. In *PHYSOR 2016*. Sun Valley, Idaho, USA.
- Wang, J., Wang, Q., & Ding, M. (2020, March). Review on neutronic/thermal-hydraulic coupling simulation methods for nuclear reactor analysis. *Annals of Nuclear Energy*, 137, 107165. Retrieved 2020-01-23, from <https://linkinghub.elsevier.com/retrieve/pii/S0306454919306759> doi: 10.1016/j.anucene.2019.107165

- Weinberg, A. M. (1994). *The First Nuclear Era: The Life and Times of a Technological Fixer*. New York, USA: Springer.
- Weller, H. G., 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. Retrieved 2017-08-15, from <http://scitation.aip.org/content/aip/journal/cip/12/6/10.1063/1.168744>





Rodrigo G.G. de Oliveira



Email: oliveira.r@pm.me

Website: www.de-oliveira.me

Education

2017 – 2021 Thesis	Ph.D. in Nuclear Engineering École Polytechnique Fédérale de Lausanne Improved methodology for analysis and design of Molten Salt Reactors	Lausanne, Switzerland
2015 – 2016 Thesis	M.Phil. in Nuclear Energy University of Cambridge Neutronic Thermal-Hydraulic Coupling of 3D MoC WIMS and Sub-channel Analysis COBRA-EN codes	Cambridge, UK
Topics	Reactor Physics, Reactor Engineering, Nuclear Materials, Nuclear Fuel Cycle and Decommissioning, Nuclear Safety, Computational Reactor Modelling, and Advanced Reactors	
2014 – 2015 Thesis	M.Sc. in Sustainability Engineering Heriot-Watt University Recovery of Proteins from Pot Ale by Ion Exchange	Edinburgh, UK
Topics	Design Methods, Foundations of Energy, Process Engineering, Process Intensification, Sustainable Processing, Principles of Recycling, Environmental Impact Assessment and Economics of Renewable Energy	
2007 – 2014 Thesis	B.Sc. in Chemistry Federal University of Rio de Janeiro The use of Anodic Stripping Voltammetry for the Analysis of Produced Waters	Rio de Janeiro, Brazil

Experience:

2021	National Institute of Chemical Physics and Biophysics Scientist	Tallinn, Estonia
	<ul style="list-style-type: none">• Lead of the AI Nuclear Operator project• Gives a course on Reactor Physics	
2017	Paul Scherrer Institut PhD work and teaching assistant	Villigen, Switzerland
	<ul style="list-style-type: none">• Development of an OpenFOAM-based Multiphysics tool for simulation of reactor systems including neutronics, fluid dynamics and thermal-mechanics.• Application of Method of Manufactured Solutions to code verification.• Modelling of freezing phenomena in multiphase flow.• Teaching assistant of Special Topics in Reactor Physics at ETH Zurich (Swiss Federal Institute of Technology in Zurich).	
2015	Horizon Proteins Summer Project / Thesis	Edinburgh, UK

- Development of 1 mL protein separation method adapted from radioactive waste and rare-earth separation and purification techniques previously learnt
- Scale-up to 1 L protein separation method, producing 250 g of protein
- Scale-up to 400 L pilot plant trials producing tens of kilograms of protein
- Improvement of efficiency and accuracy of protein quantification analysis

2011 – 2014

Laboratory of Mineral and Environmental Analysis

Rio de Janeiro, Brazil

Internship

- Separation/Analysis: Precious metals, Rare Earth, Thorium, Uranium
- Caesium decontamination by ion-exchange
- Environmental analysis for Angra Nuclear Power Station
- Development of process control methodologies
- Supervision of a team of analysts, delivering comprehensive reports of consistent high quality meeting deadlines

Main Skills

Nuclear

- Serpent 2, WIMS, COBRA, and others

Programming

- Advanced C++, Python, Bash, LaTeX and GitHub

Software

- Advanced user of Unix and Windows

Qualifications and Additional Skills

Professional

- Associate Member of the Royal Society of Chemistry (UK)

Memberships

- Associate Member of the Institute of Physics (UK)

Languages

- Native Portuguese and fluent English

Interests

- Travel and culture exchange
- Open water SCUBA diving and nature photography
- Hobbyist inventor (in home laboratory in Brazil)

References

Dr. Marti Jeltsov

National Institute of Chemical Physics and Biophysics

marti.jeltsov@kbfi.ee

Dr. Konstantin Mikityuk

Paul Scherrer Institut

konstantin.mikityuk@psi.ch