POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Development and Validation of a Simplified Thermal-Hydraulics Solver for the Modelling of Boiling Water Reactor Fuel Channels in DONJON5

CLÉMENT HUET
Département de Génie Mécanique

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Génie Mécanique

Juin 2025

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

Development and Validation of a Simplified Thermal-Hydraulics Solver for the Modelling of Boiling Water Reactor Fuel Channels in DONJON5

présenté par **Clément HUET**
en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Fabian DENNER, président
Cédric BÉGUIN, membre et codirecteur de recherche
Alain HÉBERT, membre et codirecteur de recherche
Stéphane ÉTIENNE, membre

DEDICATION

À mes grands pères . . .

ACKNOWLEDGEMENTS

Je tiens avant tout à remercier chaleureusement mes directeurs de recherches, Messieurs Alain Hébert et Cédric Beguin: Alain Hébert pour sa grande expérience et sa confiance dans l'exploration de ce sujet; Cédric Beguin pour sa disponibilité, ses encouragements et toutes nos conversations particulièrement fertiles.

Merci au Conseil de recherches en sciences naturelles et en génie du Canada (CRNSG) ainsi qu'a la Commission canadienne de sûreté nucléaire du Canada (CNSC) pour le financement de ce projet.

Merci à toute l'équipe du laboratoire et surtout Raphaël Guasch. Guide, collègue et ami qui m'a habilement aidé tout au long de mon projet et qui a su éveiller en moi une grande curiosité intellectuelle.

De France et du Canada, je tiens a remercier mes amis. Sans même vous en rendre compte c'est en partie grâce à vous que j'ai pu faire mon travail sans soucis de l'orage, même si je ne vous cite pas tous le coeur et l'âme y sont. Petite pensée à mes collocs, Maya, Hortense et Ben mes alliés du quotidien. Laury-Ann, merci pour ton inconditionnel soutien. Je te remercie pour le cadeau de ta présence à mes cotés tout les jours.

Finalement je tiens à remercier ma famille. Maxence merci, toute ma vie, sans même le savoir, tu m'as donné de l'amour et de l'énergie de façon diablement efficace. Maman, Papa, je vous dois toutes ces opportunités qui m'ont permis d'être là où j'en suis dans ma vie. Merci pour votre aide, vos sacrifices et votre amour.

RÉSUMÉ

Ce mémoire de recherche présente le développement, la vérification et la validation d'un nouveau solveur de thermohydraulique simplifiée pour la modélisation d'un canal de réacteur nucléaire à eau bouillante (REB) pour le code de simulation Open-Source de cœur Donjon5. L'objectif premier de cette recherche est d'adapter le module de thermohydraulique déjà existant THM: initialement conçu pour la modélisation des réacteur à eau pressurisée (REP) et CANada Deuterium Uranium (CANDU) aux BWR. Les améliorations principales résident dans la modélisation de la chute de pression le long du canal du combustible, la modélisation des écoulements diphasiques, des transferts de chaleurs associés et de l'ébullition le long du canal.

Ce travail comprend une revue de littérature des modèles, des codes de CFD et de thermohydraulique simplifiés existant. La conceptualisation du modèle basée sur le drift flux modèle (DFM) est explicitée ainsi que le processus de développement du code. Le DFM utilisé permet de décrire de façon efficace les interactions entre les phases liquide et vapeur sans nécessiter de trop grandes capacités de calcul. On y résout les équations de conservation de la masse, de la quantité de mouvement et de l'énergie ajustées selon le DFM et résolues en 1D en utilisant la méthode des volumes finis pour une mixture liquide-vapeur.

Afin de vérifier le solveur, des équations analytiques du problème simplifié ont été trouvées pour comparer les résultats. La validation du solveur à été obtenue en comparant la nouvelle version du module THM: au code de calcul multiphysique CFD GeN-Foam et aux résultats expérimentaux issus du benchmark OECD/NUPEC BFBT. Les résultats montrent de bons accords entre fraction de vide, température, pression et vitesse avec les références, bien que certains écarts soient observés, notamment au moment de l'ébullition sous-saturée. Ces écarts pourront faire l'objet de travaux futurs, notamment pour modéliser de façon plus précise le départ en ébullition. Une analyse de sensibilité et de performance a ensuite été réalisée afin d'évaluer l'intérêt du solveur dans un contexte industriel.

Les contributions de ce travail incluent non seulement l'avancement des capacités du code Donjon5, mais aussi la promotion des outils open-source dans le domaine du génie nucléaire, conformément aux principes de la science ouverte.

ABSTRACT

This master thesis presents the development, verification and validation of a new simplified thermal-hydraulics solver for modeling a boiling water nuclear reactor (BWR) channel for the Donjon5 open-source full core simulation code. The main objective of this research is to adapt the existing thermal-hydraulics module `THM:` initially designed for pressurised water reactor (PWR) and Canada Deuterium Uranium (CANDU) modelling to the BWR. The main improvements lie in the modelling of the pressure drop along the fuel channel, the modelling of two-phase flows, associated heat transfers and boiling along the channel.

This work includes a literature review of existing models, CFD and simplified thermal-hydraulic codes. The conceptualisation of the model is based on the drift flux model (DFM), and the code development process is explained. The DFM enables the interactions between the liquid and vapour phases to be described efficiently without excessive computing power. The conservation DFM version of the mass, momentum and energy equations are solved and solved in 1D using the finite volume method for a liquid-vapour mixture.

In order to do the verification of the solver, analytical equations of the simplified problem were found to compare the results. The solver was validated by comparing the new version of the `THM:` module with the CFD GeN-Foam multiphysics calculation code and with experimental results from the OECD/NUPEC BFBT benchmark. The results show good agreement between void fraction, temperature, pressure and velocity with the benchmarks, although some discrepancies are observed, particularly at the time of undersaturated boiling. These deviations could be the subject of future work, in particular to model subcooled boiling more accurately. A sensitivity and performance analysis was then carried out to evaluate the solver's interest in an industrial context.

This work's contributions include advancing the capabilities of the Donjon5 code and promoting open-source tools in nuclear engineering, in line with the principles of open science.

TABLE OF CONTENTS

DEDICATION . . . iv

ACKNOWLEDGEMENTS . . . v

RÉSUMÉ . . . vi

ABSTRACT . . . vii

TABLE OF CONTENTS . . . viii

LIST OF TABLES . . . xi

LIST OF FIGURES . . . xiii

LIST OF SYMBOLS AND ACRONYMS . . . xv

LIST OF APPENDICES . . . xviii

CHAPTER 1 Introduction . . . 1
- 1.1 Context . . . 1
- 1.2 Goal . . . 1
- 1.3 Thesis structure and methodology . . . 3
- 1.4 Open-source approach . . . 3

CHAPTER 2 Literature review . . . 5
- 2.1 Basics of neutronics and reactor physics . . . 5
- 2.2 Basics of two-phase flow in a reactor . . . 7
  - 2.2.1 Definition . . . 7
  - 2.2.2 Dimensionless number . . . 10
  - 2.2.3 Bubble flow . . . 11
- 2.3 BWR Benchmark and full core models . . . 15
  - 2.3.1 OECD NUPEC BWR Full-Size Fine-Mesh Bundle Test Benchmark . . . 16
  - 2.3.2 Atrium 10 benchmarks . . . 18
- 2.4 Existing code to model LWR and BWR . . . 19
  - 2.4.1 Cobra-TF code . . . 19
  - 2.4.2 Trace Thermal-Hydraulic code . . . 19

2.4.3 PARCS/PATHS code 20
2.4.4 TWOPORFLOW 22
2.4.5 Other 23
2.5 OpenFoam 25
2.5.1 The different subsolver 26
2.5.2 Coupled codes 27
2.5.3 Conclusion 28

CHAPTER 3 Mathematical modelisation of two-phases flow 30
3.1 Conservation equations 30
3.1.1 Averaging conservation equation 32
3.1.2 Mixture equation 36
3.2 Drift flux model 37

CHAPTER 4 Developement of the new thermohydraulic solver 45
4.1 Initial mathematical and numerical modelization in `THM:` 46
4.1.1 Thermal conduction equation in the fuel, the gap and the clad 46
4.1.2 Initial convection modelisation in the coolant 48
4.1.3 Limit of the thermohydraulic module 49
4.2 New modelisation of the convection 49
4.2.1 Mathematical modelisation 49
4.2.2 Constitutive equations 50
4.2.3 Friction and two-phase factor correlations 52
4.2.4 1D numerical discretisation 53
4.3 Numerical simulation methodology 57
4.3.1 Methodology 57
4.3.2 Algorithm 59
4.3.3 Numerical resolution 60

CHAPTER 5 Presentation of the validation case and GeN-Foam 62
5.1 Presentation of the validation case 62
5.2 Two-phase simulation using OpenFOAM and the nuclear sub-solver GeN-Foam 65
5.2.1 Thermohydraulic mathematical and numerical modelling in GeN-Foam 66
5.2.2 Validation and application - GeN-Foam test cases 68

CHAPTER 6 Results and code to code validation 71
6.1 `THM:` pressure drop implementation 71

6.1.1 New results compared to the previous code 72
6.1.2 Verification of the code 74
6.2 Complete steady states model validation 76
6.2.1 Simple validation against reference code 77
6.2.2 Validation against the BTBF PWR subchannel benchmark 80
6.2.3 Models and correlation comparison 82
6.3 Perfomance 87
6.4 Discussion 89

CHAPTER 7 Multiphysics modelisation 91
7.1 Introduction 91
7.2 Theory 92
7.2.1 Convection in the coolant 92
7.2.2 Conduction in the fuel pin 93
7.3 Methodology 94
7.3.1 Problem specifications 94
7.3.2 Generating microscopic cross sections : the DRAGON5 lattice code . 95
7.3.3 Full core diffusion calculations in DONJON5 95
7.3.4 Steady state coupled neutronics-TH loop 96
7.3.5 Numerical verification and validation methodology 97
7.4 Numerical results and analysis 98
7.4.1 Coupled neutronics-TH loop verification 98
7.4.2 Validation of the neutronics solution 99
7.4.3 Comparison of `THM prototype` solution to GeN-Foam 100
7.5 Conclusions 102

CHAPTER 8 Conclusion 103
8.1 Summary of Works 103
8.2 Future Research 103

REFERENCES 105

APPENDICES 110

LIST OF TABLES

Table 2.1 Geometric features, thermal-hydraulic data, and heat transfer properties of the OECD NUPEC BWR Full-Size Fine-Mesh Bundle Test Benchmark 18

Table 2.2 Root-Mean-Square Difference and Maximum Difference in Percent Void for PATHS Compared to the Experimental FRIGG Results [1] 21

Table 2.3 Summary of main thermal-hydraulic codes 24

Table 4.1 Coefficient values for different flow regimes 53

Table 5.1 Model Parameters and Properties 64

Table 5.2 Input of the actual `THM:` module and proposition for new 64

Table 5.3 Purpose of new input of the updated `THM:` 64

Table 5.4 Geometrical and heated characteristics of sub-channel 70

Table 6.1 Geometric and Thermohydraulic Parameters for the simple test . . . 77

Table 6.2 Absolute RMS, maximum, and average `THM prototype`-GeNFoam deviations on pressure, void fraction and coolant temperature at power levels of 10.227, 20.453 and 40.907 kW 79

Table 6.3 Geometric and Thermohydraulic Parameters for the void fraction model correlation 83

Table 6.4 RMS and maximum error on void fraction for the different void fraction correlations 84

Table 6.5 RMS and maximum error pressure for the different two-phase multiplier and friction factor correlations 86

Table 6.6 Main computational bottleneck 89

Table 7.1 Specifications of the case studied : geometric features, thermal-hydraulic data, and heat transfer properties. 95

Table 7.2 Tabulation points for T_F, T_m, ρ_m, used in lattice calculations. 95

Table 7.3 Convergence criteria set for multi-physics iterations 97

Table 7.4 Dependence of Δ_{RMS} on axial mesh discretisation for 10 kW and 35 kW power normalizations. 99

Table 7.5 Evolution of Δk_{eff} (pcm) with increasing number of axial slices for power levels 10kW and 35kW. 99

Table 7.6 Absolute RMS, maximum, and average `THM prototype`-GeNFoam deviations on pressure, void fraction and coolant temperature at power levels 10kW and 35kW. 101

Table F.1 Geometric and Thermohydraulic Parameters for the TwoPorFlow test case 121
Table F.2 RMS %, maximum % deviations on coolant temperature, void fraction, pressure and velocities at power levels 3.8, 9.6, 19.2, 38.4 kW 123

LIST OF FIGURES

Figure 1.1 Schematic representation of the BWR reactor 2
Figure 2.1 Control volume and interface 8
Figure 2.2 Flow patterns in a upward co-current flow in vertical pipes [2] 12
Figure 2.3 Flow patterns in a vertical heated tube [2] 13
Figure 2.4 Flow pattern diagram for steam-water flow for a diameter of $D = 2.5cm$ from Taitel work [3] 14
Figure 2.5 OpenCFD Limited. OpenFoam User Guide, 1.7.1 edition, 2010. . . . 26
Figure 4.1 Mesh-Centered Finite Difference 46
Figure 4.2 Algorithm flowchart and link with the oriented-object code 60
Figure 5.1 Schem of the geometry studied 63
Figure 5.2 Single subchannel geometry used to validate the boiling capacities of GeN-Foam 69
Figure 6.1 Pressure, temperature, void fraction and liquid velocity comparison between the previous `THM:` module and the updated one 73
Figure 6.2 Numerical solution of $\texttt{THM}_f$ for 10 and 100 volumes compared to the analytical solution 75
Figure 6.3 Convergence of the numerical solution compared to the analytic solution 76
Figure 6.4 Temperature, void fraction, pressure and liquid velocity comparison against GeN-Foam test case for 10.227, 20.453 and 40.907 kW fuel power 78
Figure 6.5 Comparison between $\texttt{THM}_p$ and GeN-Foam with a simulation starting at 0.8 m with inlet void fraction equal to the $\texttt{THM}_p$ void fraction at 0.8 m 80
Figure 6.6 Terminal void fraction (a) and density (b) comparison against the Nupec OECD BFBT benchmark 81
Figure 6.7 Error in void fraction ($\varepsilon_{simulated} - \varepsilon_{experimental}$) as a function of energy. 82
Figure 6.8 Pressure and temperature mean relative error compared to GeN-Foam test cases for a range of $1GW/m^3$ to $5GW/m^3$. 83
Figure 6.9 Void fraction mean error compared to GeN-Foam test cases for $1GW/m^3$ to $5GW/m^3$. 84
Figure 6.10 Void fraction and temperature error compared to GeN-Foam test cases for a range of $1GW/m^3$ to $5GW/m^3$. 85
Figure 6.11 Relative mean pressure error compared to GeN-Foam test cases for a range of $1GW/m^3$ to $5GW/m^3$. 86
Figure 6.12 $\texttt{THM}_p$ profiling 88

Figure 7.1 Multi-physics coupling loop : data flow in *PyGan* procedure. Steps performed in the main Python3 procedure, calls to our `THM prototype`, to the *Python3 LCM* API and to procedures through the *Python3 CLE-2000* API are respectively indicated by yellow, blue, orange and green boxes. 96

Figure 7.2 Power density (a), void fraction (b) evolution along the axial direction. Spatial convergence analysis for increasingly fine meshes. Power normalizations $10kW$ and $35kW$. 98

Figure 7.3 D5-S2 RMS deviation between power density distributions (a) and axial evolution of nodal relative deviation on power densities (b). 100

Figure 7.4 Void fraction (a), pressure (b), and coolant temperature (c) evolution along the axial dimension. Full and dashed lines respectively correspond to the $P_{tot} = 10kW$ and $P_{tot} = 35kW$ cases. The green and red colors are associated to the `THM prototype` and to the GeN-Foam solutions respectively. 101

Figure B.1 Code for mass conservation modelling 112

Figure B.2 Code for momentum conservation modelisation 113

Figure F.1 Temperature, void fraction, pressure and liquid velocity comparison against TwoPorFlow test case for 3.8, 9.6, 19.2, 38.4 kW fuel power . 122

LIST OF SYMBOLS AND ACRONYMS

The next list describes several symbols that will be used later within the body of the document

Latin and Greek symbol

Φ	porosity	
V_k	volume of the phase k	Pa
r_w	radius of the channel for a circular channel, pitch for a square channel	m
r_c	clad radius	m
r_f	fuel radius	m
D_h	hydraulic diameter	m
A	area	m^2
g	gravitational acceleration	$m \cdot s^{-2}$
Q	volumetric flow rate	m^3/s
j	volumetric flux, superficial velocity	m/s
ρ	density	$kg \cdot m^{-3}$
v	velocity	m/s
h	enthalpy	J/kg
T	temperature	K
P	pressure	Pa
x_s	static quality	
x_d	flow quality	
x_{th}	thermodynamic quality	
k	thermal conductivity	$W/m/K$
μ	dynamic viscosity	$kg/m/s$
ν	kinematic viscosity	$Pa.s$
ε	mean void fraction	
ε_k	fraction of phase k	
$\overline{\overline{\tau_k}}$	stress tensor	
$\Gamma_{v\ell}$	evaporation rate per unit volume	$kg/m^{-3}.s-1$
$\Gamma_{\ell v}$	condensation rate per unit volume	$kg/m^{-3}.s-1$
Γ_k	mass creation rate	$kg/m^{-3}.s-1$
$\vec{M_k^i}$	momentum transfer	$kg.m^{-2}.s^{-1}$
$\vec{n_k^i}$	normal vector to the gas liquid interface	
$\vec{n_k^w}$	normal vector to the wall and phase k interface	

δ	dirac function	
χ	indicator function	
$\langle j \rangle$	total volumetric flux of the two phase mixture	m/s
V_m	velocity of the centre of mass of the mixture	m/s
c_0	distribution parameter	
V_{vj}	drift flux velocity	m/s

Subscript

k	phase (liquid or vapour)
$_\ell$	liquid phase
$_v$	vapor phase
$_g$	gas phase
$_m$	mixture
$_{sat}$	saturation
$_{MC}$	mass center
$_{VC}$	volume center

Superscript

^{eff}	effective
^{turb}	turbulent
$'$	temporal derivative
^{i}	gas liquid interface
^{w}	wall interface

Notation

$\tilde{\psi}$	local value (not yet averaged) of the variable ψ
$\langle \tilde{\psi} \rangle$	surface averaged value of ψ
ψ	void fraction weighted mean value

Acronym

LWR	Light Water Reactor
BWR	Boiling Water Reactor
SMR	Small Modular Reactor
$PARCS$	Purdue Advanced Reactor Core Simulator

$PATHS$	PARCS Advanced Thermal Hydraulic Solver
$CANDU$	CANada Deuterium Uranium
THM	THermohydraulic Module
$FAIR$	Findable, Accessible, Interoperable, Reusable
BTE	Botlzman Transport Equatio
$NUPEC$	Nuclear Power Engineering Corporation
$OECD$	Organisation for Economic Co-operation and Development
NEA	Nu- clear Energy Agency
SMR	Small Modular Reactor
$TRACE$	Tansient Reactor Analysis Code
$EPRI$	Electric Power Research Institute
GE	General Eletric
HEM	Homogeneous Equilibrium Model
COV	Covariance
FEM	Finite element method
KIT	Karlsruhe Institute of Technology
TPF	TwoPorFlow
CFD	Computational fluid dynamic
CHF	Critical heat flux

LIST OF APPENDICES

Appendix A Void Fraction correlation 110

Appendix B Mass and momentum concervation equation code in the previous `THM:` model 112

Appendix C Numerical discretisation for transient 114

Appendix D Justification for Neglecting the Surface Tension Term $\vec{\kappa}$ 117

Appendix E Turbulence 119

Appendix F Validation using the porous media code TWOPORFLOW 120

CHAPTER 1 Introduction

1.1 Context

Faced with growing energy needs, dwindling fossil fuel reserves and the spectre of global warming, nuclear power is seen as a viable solution to our energy and environmental problems. Indeed, nuclear energy production has many advantages. It is a safe form of energy, monitored by independent safety organizations such as the Canadian Nuclear Safety Commission in Canada and the Autorité de Sûreté Nucléaire in France. What's more, the absence of greenhouse gas emissions during the operation of power plants makes it one of the most carbon-free forms of energy. For constant energy production, emissions from the sector are ten times lower than those from fossil fuels and similar to those from renewable energies [4]. Finally, the associated production costs are relatively similar to those of other energy production methods [5]. Although there are still some problems with fuel extraction and waste treatment, the benefits of this technology are still growing.

The main type of water-cooled reactor is the Light Water Reactor (LWR), which accounts for two-thirds of the world's total. They are divided into two kinds: Boiling water reactors (BWR) and Pressurized water reactors (PWR). In a BWR, the reactor core directly heats the water and boils it to produce steam, which drives a turbine. Its scheme is presented figure 1.1. Conversely, in a PWR, the reactor core heats the water under high pressure, transferring its heat to a separate system and transforming it into steam, which then powers the turbine.

This change significantly affects a reactor's neutronics, material interactions, and energy distribution/generation. In addition, in BWRs, the presence of large water rods in the assembly is necessary. These rods occupy a significant part of the water flow zone and increase local power by facilitating more efficient neutron moderation. As a result, these rods induce variations in neutronics and overall heat transfer in the vicinity. In Canada the provinces of Ontario and Saskatchewan have selected a specific BWR model. They plan to expand their nuclear fleet by building General Electric-Hitachi's BWRX-300 small modular reactor (SMR).

1.2 Goal

The safety analysis and design of nuclear reactors require a good prediction of their physics. This coupling must be implemented on calculation codes to validate reactor operation in nominal and accident conditions. Daily simulation of the reactor operation are based on sim-

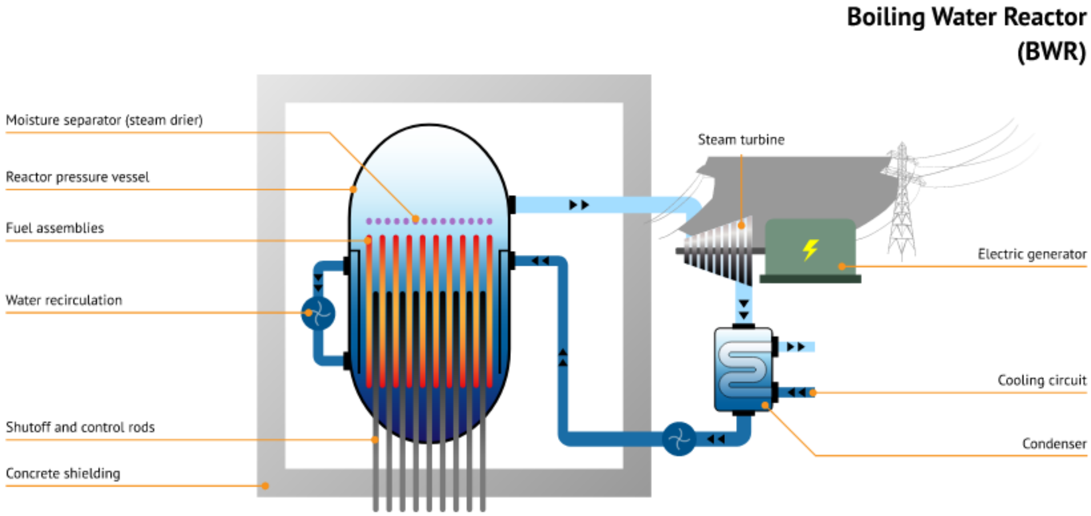


Figure 1.1 Schematic representation of the BWR reactor

plified thermal-hydraulic models. Numerous simplified thermal-hydraulic calculation codes exist, such as the PATHS module of the PARCS code [1] or the CTF module of the COBRA-TF code family [6], which we will mention later.

The subject of this master's thesis is to adapt the `THM:` [7] module of the existing DONJON5 code, validated for PWRs, to the physics of BWRs. The DONJON5 code is a core calculation code developed by the Nuclear Engineering Institute of the Ecole Polytechnique de Montréal. It is associated with the DRAGON5 assembly/lattice calculation code [8] and from the VERSION5 code system.

Since the 1990s, these codes have been developed mainly to accurately represent the accurately represent the Canada Deuterium Uranium (CANDU) reactor. The aim of this master's thesis is a sub-goal of a project to simulate Canada's new BWR reactors: adapt the `THM:` code to the BWR physics and validate results using code to code validation. The second main objective of this master's thesis will be to develop a two-phase flow simulation using another calculation code. The chosen computational tool is OpenFOAM, an open-source framework. OpenFOAM is a set of open-source libraries and solvers for solving the partial differential equations governing mass, heat and momentum transfer. Mastering this computational tool will be essential to simulate the flows that could exist in the reactors studied as faithfully as possible. Modification of the `THM:` code will then be validated using OpenFOAM.

1.3 Thesis structure and methodology

With the aim of adapting existing `THM:` code: the first task will be to review existing simplified thermal-hydraulic code applied to BWR. These codes will probably enable us to develop a robust mathematical model for the `THM:` changeover. The main features of these codes should be their ability to model phase change and heat transfer between liquid and gaseous phases.

After reviewing the existing coupled neutronics/thermohydraulics code we need to choose one which will be the reference for the modification of `THM:`. The goal will on one hand the development of `THM:` and in other hand the case creation in a different code. These two aspects will be done simultaneously. As mentioned above, cases will be created using OpenFOAM. However, it would be essential to benchmark the various existing solvers and additional libraries.

In order to operate modification in `THM:` it will be first important to list the main limit of the simulation to make the good research to tackle them. Then, modifying the code will be necessary. Insofar as the existing `THM:` code is written in Fortran, the strategy adopted will be to port `THM:` to the Python language in order to make modifications in a more accessible language. In this master thesis, the ported and updated code on Python will be called $\texttt{THM}_p$ or $\texttt{THM}_{prototype}$. Once these modifications have been considered satisfying, the code will be ported back to Fortran and the VERSION5 environment.

1.4 Open-source approach

The VERSION5 [9] [8] environment is an open-source code development platform. One goal of this master's thesis is to provide even more open-source tools to researchers and industry professionals worldwide. Open-source codes are powerful tools to provide free and collaborative software solutions. It defines free, accessible, modifiable and redistributable software by the public or the scientific community. The open-source concept is a part of the open-science movement, which is aimed at promoting the reuse and interoperability of data such as codes, scientific publications and tools in the scientific field. By promoting open science, researchers seek to improve the efficiency and impact of scientific research by making it easier to find, reuse, and integrate data from diverse sources.

Open-source codes are a part of the data exchanged by the researcher. It allows them to use and improve code from companies and universities worldwide. This work contributes to democratising knowledge: Open-source tools make cutting-edge methodologies accessible to researchers, including those from institutions in developing nations, fostering inclusivity and bridging resource gaps. The collaboration between universities can also foster innovation, cutting countries' boundaries and merging the knowledge of many labs. Because every Open-source code can be tested and reviewed by everyone, it improves the transparency and trust we can have in those codes. The ability to scrutinise and verify source code enhances the reliability of simulation outcomes. It is critical in a safety-conscious field like nuclear engineering.

In nuclear science, open source seems particularly useful for the scientific and industrial community. As highlighted in [10], open-source codes are increasingly utilised for nuclear reactor analysis. It can replace proprietary software that often poses barriers due to high costs, limited access, and opacity. Catching this barrier allows researchers, public institutions, and developing nations to have access to high scientific knowledge to build their own nuclear industry and tackle the technological gap. Open-source initiatives in the nuclear domain strongly align with Open Science principles, such as the FAIR guidelines: Findable, Accessible, Interoperable, and Reusable. For instance, platforms like MOOSE (Multiphysics Object-Oriented Simulation Environment) [11], Salome [12] and OpenFOAM [13] provide frameworks for collaborative research. Sharing and doing repeated and mutual control of the code allow scientists to build shared tools which are very safe and efficient. In the case of reactor analysis, this efficiency and safety analysis plays a critical role. We will mention the commercial and open source code in the section 2.4

During this master thesis, I will commit to using only open-source tools, whether for CAD, code-to-code validation, or throughout the entire development process. The open source VERSION5 environment [9] [8] is used by public organisations, researchers and private companies and aims to spread the nuclear computational capability and knowledge. Contributing to the open-source code Donjon5 by developing a new simplified thermal-hydraulic solver, this master's thesis can help the open-source community, particularly in nuclear science and engineering.

CHAPTER 2 Literature review

The main objective of this research work is to contribute to developing the `THM:` module in Donjon5 code for the simulation of two-phase bubble flows in a nuclear environment. To begin with, we need to understand the neutronics concepts we will be using. These concepts will be used throughout our work and serve as input variables for many calculations. We will also present the benchmarks on which we will base our comparison. Then, we will define the variables and parameters that are important to model the fluids. Finally, we will present a brief, non-exhaustive literature review of the thermal-hydraulic codes used to model BWRs. This literature review will give the key to understanding the coupling between neutronics and thermal-hydraulics and summarize existing modelisation for two-phase flow. Knowing that the $\texttt{THM}_p$ python prototype will be coded from scratch, it will be interesting to recall some modelling tricks to simulate the two-phase flow in the single-pin cell.

2.1 Basics of neutronics and reactor physics

Basics of neutronics are necessary for understanding and analysing the behaviour of nuclear reactors, including boiling water reactors (BWRS). In thermal-hydraulics, *fission power*, *neutron transport*, and *interactions* play a fundamental role in simulating reactor performance and safety. In the context of this master's thesis, we won't discuss the mathematical model of the neutronics calculation, although it is coupled with our thermal-hydraulic calculations. However, some basic notions are necessary to understand the vocabulary in the following sections. The following paragraph will delve into various concepts essential for comprehending neutronics in the context of the thermohydraulics of BWRs.

Neutron transport is a significant component of neutronics, describing the movement of neutrons through the reactor. This complex process considers neutron interactions with atomic nuclei, including diffusion, absorption and fission. This behaviour is modelled by the *Boltzmann Transport Equation (BTE).* This equation uses statistical mechanics to describe the neutron transport inside the material of a reactor core. In neutronics analysis, the reactor core is often discretized into small spatial regions called *lattice cells* to the *lattice assemblies.* The lattice calculation step involves solving the *neutron transport equation* for each cell, considering the *neutron flux* in its fundamental mode. Solutions mostly depend on various factors like material compositions, fuel configurations, and geometric arrangements. To simplify this equation, diffusion formalism is often used, enabling a more straightforward resolution while maintaining an adequate level of accuracy. This project will perform the lattice calculation step using the Dragon5 [8] lattice code. The full core simulation step will be performed using Donjon5 [9], and its associated finite element method (FEM) solver Trivac5 [14].

The neutron flux in the reactor corresponds to the probability of a neutron being measured per volume unit multiplied by the neutron velocity. In other words, it's a population density multiplied by a velocity. *Neutron calculations* can be done at different scales, from the complete reactor core to a single fuel assembly. They are essential for assessing reactor performance and optimizing design. A commonly used numerical resolution method is the *characteristic method*, a solution of the Boltzmann transport equation over long (infinite) trajectories.

Neutronics also relies on fundamental data such as *cross sections*, which describe the probability of a neutron interacting with an atomic nucleus. Each cross-section is associated with a reaction. They are pretty different for capture, fission and diffusion reactions. These data are crucial for characterizing the nuclear properties of the materials used in the reactor.

The *fission power* in a nuclear fuel is a measure of the energy released by the fission of any fissile material present in the fuel. Fission power is calculated using the fission rate. This reaction rate is the product of the neutron flux and the fission *cross-section* weighted by a nu factor (ν = number of neutrons released). This gives us the neutron source in a volume element associated with this fission. To determine the fission source, we integrate this calculation for the reactor under consideration and then multiply it by the fission energy. This fission energy is an indication of the reaction being considered.

Moreover, a higher *cross-section* promotes greater fission power. The *fission fraction*, which designates the proportion of neutrons produced by the fission of a fissile nucleus that will in turn cause the fission of other fissile nuclei, is also crucial to maintaining a self-sustaining chain reaction and hence stable fission power.

In addition, the *reactivity coefficient* measures the variation in reactor reactivity in response to a change in operating conditions, such as temperature or fuel density. Positive reactivity can lead to unwanted fluctuations in fission power, while negative reactivity can contribute to reactor stability.

In neutronics k_{eff} , often referred to as the effective multiplication factor, is a critical parameter that quantifies the behavior of the chain reaction in a nuclear reactor. It represents the ratio of the average number of neutrons produced by fission in one generation to the average number of neutrons lost by absorption or leakage in the same generation.

Symbolically, k_{eff} is expressed as:

$$k_{eff} = \frac{\text{Neutrons production rate by fission}}{\text{Neutrons absorption and leakage rate}} \tag{2.1}$$

When k_{eff} equals 1, the reactor is in a critical state, meaning that the number of neutrons produced by fission is balanced by the number of neutrons lost and absorbed. In this condition, the reactor power remains constant over time. If k_{eff} is greater than 1, the reactor is supercritical, indicating that the neutron population is increasing from one generation to the next, leading to a progressive increase in reactor power. Conversely, if k_{eff} is less than 1, the reactor is subcritical, indicating that the neutron population decreases from generation to generation, reducing reactor power. Maintaining k_{eff} at or near 1 is crucial for reactor safety and stability. Reactor operators adjust control rods and other parameters to achieve this balance and ensure safe and efficient reactor operation.

2.2 Basics of two-phase flow in a reactor

Two-phase flow in a reactor describes a dynamic scenario in which three distinct states of matter coexist within a system: solid vapour and liquid. A phase represents a fundamental state of matter. Unique thermodynamic properties can distinguish each state. The phases may be of the same composition but have different physical states or compositions. In the following, we will study two-phase water-steam flows.

2.2.1 Definition

The study focuses on a fuel rod (solid) surrounded by water (liquid) or steam (vapour). The water + steam system will be considered as a two-phase mixture. Then, instead of modelling a separate vapour-solid or liquid-solid interface, we will consider a mixture-solid interface noted δ^w. The following scheme shows the different interfaces we will mention later in this thesis.

To be able to space average we define a controle volume V illustrated in figure 2.1. We define as V_g, V_ℓ, V_s respectively the volume occupied by the vapour, liquid and solid.

The porosity Φ is the fraction of the control volume occupied by the two fluids at a given time:

$$\Phi = \frac{V_g + V_\ell}{V} \tag{2.2}$$

The average void ratio ε is the fraction of the control volume occupied by the vapour phase

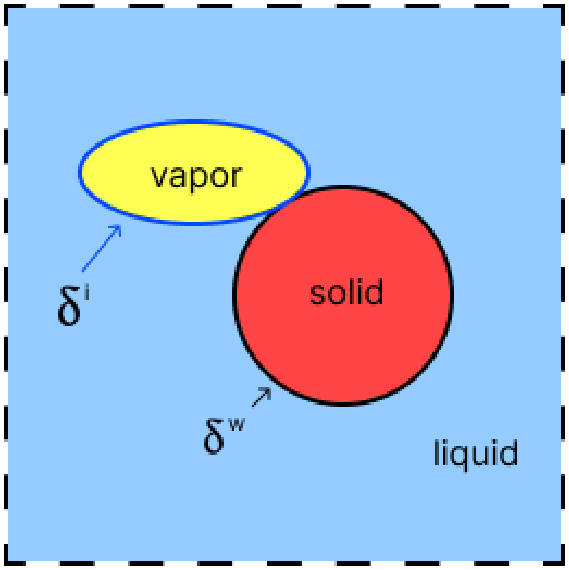


Figure 2.1 Control volume and interface

over the fluid volume at a given time. The average void ratio ε_k is the volume occupied by the phase k over the fluid volume:

$$\varepsilon_k = \frac{V_k}{V_g + V_\ell} = \frac{V_k}{\Phi V} \tag{2.3}$$

by definition $V = V_g + V_\ell + V_s$. We will note $\varepsilon_g = \varepsilon$ and $\varepsilon_\ell = 1 - \varepsilon$

The static quality [1]is a way to represent the instantaneous vapour mass fraction based on local void fraction. It is defined as:

$$x_s = \frac{\varepsilon \rho_g}{\varepsilon \rho_g + (1-\varepsilon)\rho_\ell} = \frac{m_g}{m_g + m_\ell} = \frac{\varepsilon}{\varepsilon + (1-\varepsilon)\rho_\ell/\rho_g} \tag{2.4}$$

The flow quality [2]describes the vapour's share of the total mass flow rate:

$$x_d = \frac{\int_A \varepsilon_g \rho_g \vec{v_g} dA}{\sum_k \int_A \varepsilon_k \rho_k \vec{v_k} dA} = \frac{\dot{m}_g}{\dot{m}_g + \dot{m}_\ell} \tag{2.5}$$

The mixture enthalpy is defined as:

$$h_m = \frac{\varepsilon \rho_g h_g + (1-\varepsilon)\rho_\ell h_\ell}{\varepsilon \rho_g + (1-\varepsilon)\rho_\ell} = x_s h_g + (1-x_s) h_\ell \tag{2.6}$$

[1] "Fraction massique" in French

[2] "Titre massique" in French

The flow enthalpy is different:

$$h_d = x_d h_g + (1 - x_d) h_\ell \tag{2.7}$$

To determine the thermodynamic quality [3], we consider the mixture enthalpy. It infers the vapour fraction from the mixture's enthalpy state. We can then define this quality use the specific enthalpy of saturated water and specific enthalpy of saturated steam, respectively $(h_\ell)_{sat}, (h_g)_{sat}$ as:

$$x_{th} = \frac{h_m - (h_\ell)_{sat}}{(h_g)_{sat} - (h_\ell)_{sat}} \tag{2.8}$$

We define the area (or volume) averaged operator $\langle\rangle$ for a quantity local $\tilde{\psi}$:

$$\langle\tilde{\psi}_k\rangle = \frac{1}{A}\iint_A \tilde{\psi}_k dA \tag{2.9}$$

.

So $\langle\chi_k\rangle = \varepsilon$. Then we can define also the void fraction weighted averaged quantity $\bar{\psi}$:

$$\phi_k = \frac{\langle\chi_k\tilde{\phi}_k\rangle}{\langle\chi_k\rangle} \tag{2.10}$$

The mean volumetric flux, or so-called superficial velocity, j_g and j_ℓ, is defined as the volumetric flow rate of the phase divided by the total cross-sectional flow area of the pipe:

$$\begin{aligned} j_g &= \frac{Q_g}{A} \\ j_\ell &= \frac{Q_\ell}{A} \end{aligned} \tag{2.11}$$

Where the volumetric flow rate is defined as $Q_k = A_k\langle\tilde{v}_k\rangle$ with $\langle\tilde{v}_k\rangle$ the average velocity in the cross-section area of the phase k. The total surface velocity of the mix is $j = j_g + j_\ell$.

The hydraulic diameter D_h is defined as 4 times the cross-section of the flow divided by the wet perimeter $D_h = \dfrac{4(A_g + A_\ell)}{P_{wet}}$. In the case of this study, we can have circular or square channels. For circular channel, with r_w the outer radius of the channel and r_c the inner radius of the channel:

[3] "Titre enthalpique" in French

$$D_h = \frac{4\pi(r_w^2 - r_c^2)}{2\pi(r_w + r_c)} = 2(r_w - r_c) \tag{2.12}$$

in the case of a square channel surrounding a cylindrical pincell, with r_w the side of the square channel:

$$D_h = \frac{2(r_w^2 - \pi r_c^2)}{2r_w + \pi r_c} \tag{2.13}$$

2.2.2 Dimensionless number

Flow characteristics depend in particular on the following main physical quantities. Control of the following features is critical, in particular the density ρ_g and ρ_ℓ, kinematic viscosity of both phases μ_g and μ_ℓ, the gravity of Earth g, a characteristic length D_h, the velocity of phase k $\vec{v_k}$ and the surface tension σ.

Knowledge of the flow regime and flow quality is required to obtain the value of the heat transfer coefficients used in Newton's law of cooling. These coefficients are functions of the following non-dimensional numbers:

Nusselt number

$$\mathrm{N}u = \frac{H\,D_h}{k_\ell} \tag{2.14}$$

where
H = heat transfer coefficient (W/m^2/K)
D_h = hydraulic diameter (m)
k_ℓ = thermal conductivity of the liquid (W/m/K)

The Nusselt number represents the ratio of convective to conductive heat transfer at a boundary. Nu determines the efficiency of heat transfer. A high Nu indicates dominant convection, while a low Nu suggests conduction dominates.

Mixture Reynolds number

$$Re_m = \frac{\dot{m}\,D_h}{\mu_m} = \frac{\rho_m u_m D_h}{\mu_m} \tag{2.15}$$

where
$\dot{m}$ = mass flow rate (kg/m^2/s)
μ_m = dynamic viscosity of the mixture (kg/m/s) defined as $\mu_m = \varepsilon\mu_g + (1-\varepsilon)\mu_\ell$

The Reynolds number is the ratio of inertial forces to viscous forces in a fluid. Reynolds number helps evaluate the regime of the flow. It can consider the mixture in this whole or one of the phases.

Vapour Reynolds number

$$Re_g = \varepsilon \, \frac{\dot{m} \, D_h}{\mu_g} = \frac{\rho_g u_m D_h}{\mu_g} \tag{2.16}$$

where
$\dot{m}$ = mass flow rate (kg/m^2/s)
μ_g = dynamic viscosity of the vapour (kg/m/s)

Liquid Reynolds number

$$Re_\ell = (1 - \varepsilon) \, \frac{\dot{m} \, D_h}{\mu_\ell} = \frac{\rho_\ell u_m D_h}{\mu_\ell} \tag{2.17}$$

where
$\dot{m}$ = mass flow rate (kg/m^2/s)
μ_ℓ = dynamic viscosity of the liquid (kg/m/s)

Liquid Prandl number

The Prandtl number is the ratio of momentum diffusivity (viscosity) to thermal diffusivity in a fluid.

$$Pr = \frac{\mu_\ell C_\ell}{k_\ell} \tag{2.18}$$

where
ρ_ℓ = mass density of the liquid (kg/m^3)
ρ_g = mass density of the vapour (kg/m^3)
μ_g = dynamic viscosity of vapour

A low Pr means heat diffuses faster than momentum (e.g., gases).

2.2.3 Bubble flow

The configuration of multiphase flows is determined by their topology and geometric structure. For those flows, factors such as fluid mechanics laws, phase interactions, conduit shape, flow direction, and type of flow, whether liquid-gas, liquid-liquid,, or liquid-solid, play crucial roles.

Designing efficient and safe industrial multiphase systems requires a thorough understanding of the flow's nature and its dependence on operational parameters. Multiphase flows, such as those involving liquid-gas or liquid-liquid phases, are common in industries like oil and gas, chemical processing, and power generation. Factors like interfacial tension, viscosity, and density differences between phases influence the characteristics of these flows.

Engineers use flow regime maps and advanced measurement techniques to predict and manage these complex systems. Noninvasive methods such as particle image velocimetry (PIV) and laser Doppler anemometry (LDA) help visualise and measure flow characteristics without disrupting the flow. Invasive methods, including phase fraction probes and conductivity sensors, directly measure phase distribution and flow rates.

In our case, for a cocurrent vertical liquid-vapour mixture, the flow can take different configurations. Those configurations are summarised in the figure 2.2 from the book [2].

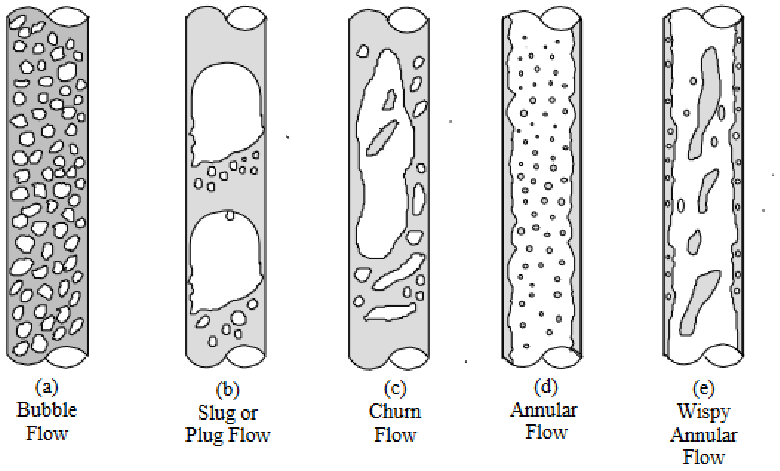


Figure 2.2 Flow patterns in a upward co-current flow in vertical pipes [2]

In our case, there is creation of a two-phase mixture through vapour generation in a uniformly heated vertical tube, as shown schematically in figure 2.3. Initially, the tube is supplied with liquid below its saturation temperature.

In the single-phase region, the liquid is heated to reach the saturation temperature, forming a thermal boundary layer at the wall and establishing a radial temperature profile. At a certain point along the tube, the wall temperature surpasses the saturation temperature, leading to the nucleation of vapour bubbles at the wall. These bubbles form at preferred sites, grow, and detach to create a *bubbly flow.* In the bubbly flow, the gas phase is distributed as discrete bubbles in a continuous liquid phase. Bubbles are usually spherical. There are two bubbly flow patterns, one interacting (if the bubbles are close enough) and one non-interacting.

As vapour production increases, the bubble population grows, leading to coalescence and the formation of *slug flow*. *Slug flow* occurs when bubble diameters approach that of the tube. The region between two successive bubbles is filled with liquid. It eventually transitions to *an annular flow*. In *annular flow* a liquid film forms at the pipe wall with a continuous central gas core but the flow is mainly composed of vapour and the liquid film surface is near the walls. Beyond the *slug-annular transition point*, bubble nucleation at the wall may stop, and further vapour formation occurs through evaporation at the liquid film-vapour core interface. The increasing vapour core velocities entrain liquid droplets. The liquid film is depleted through entrainment and evaporation, eventually drying out completely. The remaining droplets persist after dry out and gradually evaporate until only single-phase vapour remains. In our project, in the case of nuclear reactor core and pipe, the single-phase vapour point is rarely reached.

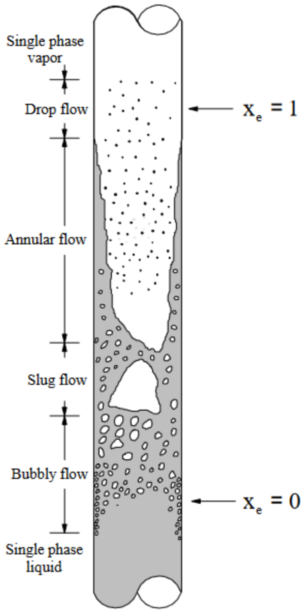


Figure 2.3 Flow patterns in a vertical heated tube [2]

Knowledge of flow patterns and the transition zones between two successive patterns is an important piece of information when modelling two-phase systems, so flow regime maps are used. These are two-dimensional representations of the domains of existence of flow configurations. They are also an interesting tool for considering the physical properties of the flow, the geometry and the operating conditions of the system. Figure 2.4 shows an example of a flow regime map for an ascending cocurrent two-phase vapour-liquid mixture.

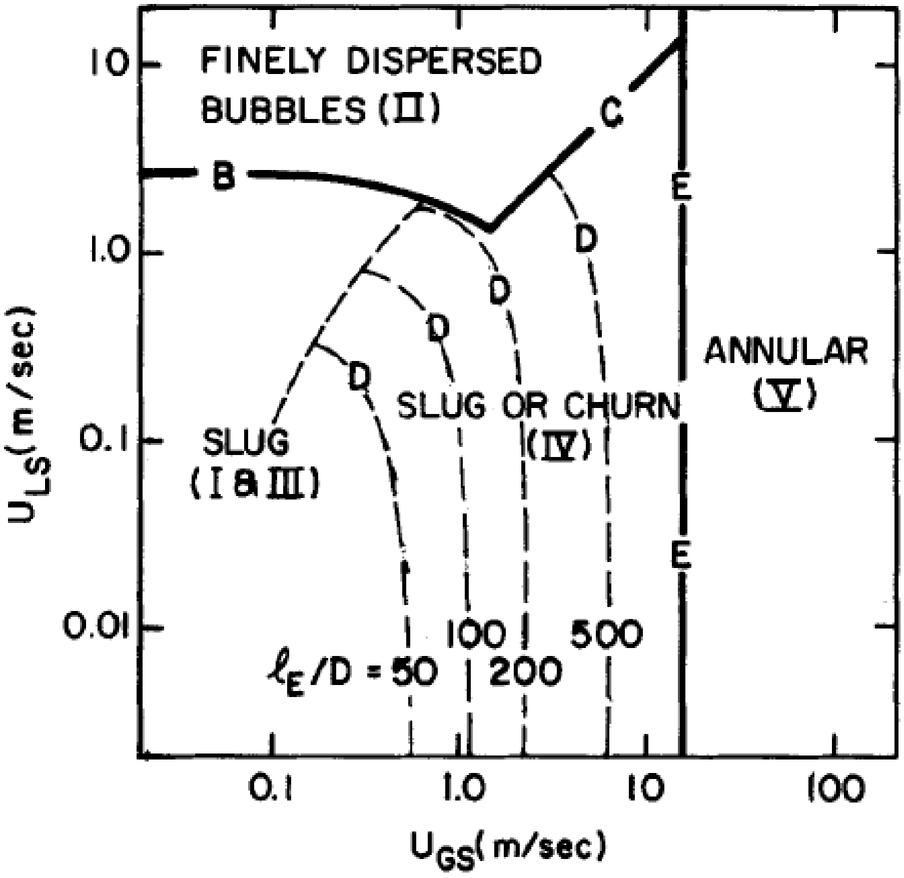


Figure 2.4 Flow pattern diagram for steam-water flow for a diameter of $D = 2.5cm$ from Taitel work [3]

This flow map is a function of the velocity of the two phases, but depending on the data and the situation being studied, it can also be obtained as a function of the thermodynamic quality, mass flow rate, the Froude number and the volume titer.

This project used a simplified version of the flow map presented. The thermodynamic quality was properly calculated and considered in calculating the void fraction. However, the correlations used in this thesis assume a slug flow regime most of the time. This makes the problem simpler by considering only the transition between a bubbly flow and an annular flow.

Taitel's [3] model assumes that transitions between flow patterns are governed by physical mechanisms and geometrical conditions such as the diameter size. According to Taitel, in slug flow, the characteristic length, which is usually the bubble diameter, can be approximated by the hydraulic diameter, particularly when calculating parameters such as the drift velocity. This distinction is critical because large Taylor bubbles characterising slug flow have diameters nearly equal to the pipe diameter. They are separated by liquid slugs containing smaller bubbles. According to Taitel, this means that the characteristic length is not the bubble diameter but rather the hydraulic diameter when calculating, inter alia, the drift velocity. The theoretical framework provided by Taitel offers a more generalizable approach which can be linked to empirical maps. It can be useful for work, considering different flow patterns and transitions. In fact, it accounts for fluid properties, pipe size, and flow rates explicitly and offers a new range of flow maps useful to model numerically two-phase flow.

This document further complements this section with a mathematical modelling of two-phase flows to obtain the equations and tools required for code development in section 3.

2.3 BWR Benchmark and full core models

The final goal of the overarching project, of which this Master's research is a part, is the multiphysics modelling of the complete BWRX-300 reactor developed by GE Hitachi Nuclear Energy. The BWRX-300 is a 300 MWe boiling water reactor. It is a proprietary design, meaning that no specific details about the assembly or the reactor's exact configurations are available. Therefore, this study will rely on open references found in the literature. The purpose of the following section is to describe these references, aiding in determining the essential characteristics and data to be modelled for our current work. Ultimately, we will use all the benchmarks to compare the new solver to create a test case with physically consistent parameters. In our hypotheses, we will focus on a single channel, which may in the future represent an equivalent channel of an assembly.

2.3.1 OECD NUPEC BWR Full-Size Fine-Mesh Bundle Test Benchmark

The OECD/NEA NUPEC BWR Full-size Fine-mesh Bundle Test (BFBT) Benchmark [15] was developed by the Nuclear Energy Agency (NEA) to address critical challenges in the simulation and validation of boiling water reactor systems. The benchmark is based on comprehensive experimental data generated by the Nuclear Power Engineering Corporation (NUPEC) in Japan. These experiments were performed on full-size BWR fuel assemblies under well-controlled conditions.

The goal of the NUPEC BWR Full-Size Fine-mesh Bundle Test (BFBT) benchmark is to improve the understanding and the accuracy of the prevision of two-phase flow in BWR. The paper focuses on void fraction distributions and critical power measurements. The benchmark is conducted on full-size BWR fuel assemblies to ensure the applicability of results to real reactor problems. The fine-mesh measurement details the spatial resolution of void fraction along the fuel bundle. It allows precise model validation.

The main objective of this benchmark is to allow researchers to model with high-fidelity variables such as:

- Void Fraction distribution: Spatial variations of vapour phase concentration in the assembly
- Critical power prediction: Conditions under which boiling crises occur, potentially leading to departure from nucleate boiling
- Local flow dynamics: Analysis of coolant velocity and turbulence in multiphase flow
- Heat transfer performance: understanding the multiphysics of heat generation and coolant removal

The experimental dataset comprises void fraction measurements, pressure drops and critical power observations across various operating conditions. The experiments are designed to be easily replicated and used in numerical simulation. This benchmark aims to realistically replicate the conditions in BWR fuel assemblies.

The benchmark is organised into several phases to facilitate incremental validation.

Phase I

Phase I focuses on the void fraction distribution within the reactor core under steady states and transient conditions. It gives data about single-pin, 2x2, and assembly configuration experiments. This phase is very interesting for understanding the behaviour of the coolant within the fuel assembly when the water transits between the liquid and vapour phases.

Phase II

Phase II addresses the measurement of the critical power and pressure drops within the fuel assemblies. Critical power is particularly important because it is a key safety limit in BWRs. It represents the point at which the coolant can no longer remove heat from the fuel rods. It no longer behaves like a moderator. It leads to a rapid rise in the temperature of the system. Phase II provides the necessary data to validate precise computational models that predict these limits.

These experiments were conducted in the NUPEC facility in Japan. It was designed specifically for thermal hydraulic testing. Full-size fuel assemblies, typically used in boiling water reactors, were used to ensure that the experimental data were relevant to actual reactor operations. Tests were carried out at pressures of approximately 7 MPa. It is representative of actual BWR conditions. Different mass flow rates, cooling rates, inlet temperatures and power levels were used to reproduce BWR operational scenarios.

These measurements were obtained using X-ray and gamma-ray densitometry, which provide high-resolution and non-intrusive imaging of void distribution inside the tested bundles. These measurements were complemented by additional sensors, such as thermocouples and pressure transducers, to measure temperature distributions and pressure drops in the fuel bundles. Finally, the BFBT benchmark plays a crucial role in advancing the accuracy and reliability of thermal-hydraulic models used in BWR reactors.

Transient Analysis includes optional cases where dynamic phenomena occur.

To develop the thermal-hydraulic solver, we will incrementally compare our results to the BWR design and parameters. The main thermal-hydraulic parameters are shown in the table 2.1

<table>
<tr><th colspan="4">Specifications of the studied case</th></tr>
<tr><th>Thermal-hydraulic properties</th><th>Value</th><th>Geometric data</th><th>Value</th></tr>
<tr><td>Inlet temperature (K)</td><td>$T_{in} = 592.75$</td><td>Channel pitch (cm)</td><td>$r_w = 1.33409$</td></tr>
<tr><td>Outlet pressure (MPa)</td><td>$P_{out} = 14.719781$</td><td>Fuel radius (cm)</td><td>$r_f = 2.171155$</td></tr>
<tr><td>Inlet velocity (m/s)</td><td>$v_{in} = 4.467$</td><td>Cladding radius (cm)</td><td>$r_c = 0.47798$</td></tr>
<tr><td>Hydraulic diameter (cm)</td><td>$D_h = 1.4355$</td><td>Channel length (m)</td><td>$h = 1.555$</td></tr>
<tr><th colspan="4">Heat transfer properties</th></tr>
<tr><td colspan="4">Fuel thermal conductivity (W/mK): $k_{fuel} = 4.18$</td></tr>
<tr><td colspan="4">Gap heat exchange coefficient (W/m^2K): $H_{gap} = 10000$</td></tr>
<tr><td colspan="4">Cladding thermal conductivity (W/mK): $k_{clad} = 21.5$</td></tr>
</table>

Table 2.1 Geometric features, thermal-hydraulic data, and heat transfer properties of the OECD NUPEC BWR Full-Size Fine-Mesh Bundle Test Benchmark

2.3.2 Atrium 10 benchmarks

The Atrium 10 benchmark [16] is a benchmark of the BWR developed by Framatome. The Atrium 10 assembly represent a significant advancement in nuclear technology. It integrates features to improve neutron economy, thermal performance and operational flexibility. This benchmark can be useful to validate both neutronics and thermal-hydraulic simulation codes.

One of the key features of the Atrium 10 is its design. This denser lattice ameliorates the neutron moderation process and improves the reactor's overall efficiency. From a thermal hydraulic approach, the Atrium 10 assembly integrates innovative spacer and mixing vane designs to improve coolant flow distribution and enhance heat transfer.

The benchmark dataset derived from the Atrium 10 design is comprehensive and includes relevant data such as void fraction distributions, pressure drop measurements and heat transfer data. This benchmark can be very useful for validating coupled thermal-hydraulic and neutronic solvers because it provides detailed information on power distributions, flux profiles and reactivity coefficients. It provides a lot of information on the geometry, which can be easily replicable. The operational characteristics are comparable to modern BWR assemblies such as the BWRX-300 making it particularly relevant for studies which aim to test new solver capabilities.

Mentioning this benchmark is interesting because it's an important candidate for the cooperative work which this master's thesis will be part of. Indeed, it is used by other contributors to the Version5 project working on neutronics and coupling, such as Raphaël GUASCH [17]. A simplified dataset of the Atrium 10 has been obtained in the paper [18]

2.4 Existing code to model LWR and BWR

Some open-source codes already exist to model nuclear reactors. They have been listed in a publication by the IAEA [10]. Proprietary codes are also reviewed using the associated public publication and data. A non-exhaustive description of codes is provided below.

2.4.1 Cobra-TF code

CTF, a thermal-hydraulic sub-channel code known as COBRA-TF, is a collaborative effort between Pennsylvania State University (PSU) and Oak Ridge National Laboratory (ORNL), developed for the U.S. Department of Energy (DOE) Consortium for Advanced Simulation of Light Water Reactors (CASL) project.

The CTF code can simulate BWR thermal-hydraulics in normal and steady-state operation. This version employs a two-fluid, three-dimensional representation of two-phase flow, enabling it to accurately simulate two-phase flow in Boiling Water Reactors (BWR). For each spatial dimension, CTF addresses three equations - each for a dimension - for momentum conservation, three for mass conservation, and two for energy conservation. It assumes thermal equilibrium between the continuous liquid field and the droplet field, resulting in the solution of eight equations for the fluid portion.

The article [19] compares critical test power and void distribution, using the values in this database as an example. This is an interesting article concerning the methodology of code validation by comparison. The author uses code-to-code validation with trace code, which will be discussed in section 2.4.2, and validation using benchmarks. The methodology consists of comparing several models, a single assembly on a resolved pin-cell level and a network of assemblies on the resolved pin-cell level (2x2). It has been validated and compared to various experimental cases, including the U.S. Nuclear Regulatory Commission (NRC) / Nuclear Energy Agency of the Organization for Economic Co-operation and Development (NEA-OECD) Boiling Water Reactor Full Bundle Tests (BFBT) Benchmark [15] or with the Oskarashamn-2 Benchmark [20] .

2.4.2 Trace Thermal-Hydraulic code

TRACE (Transient Reactor Analysis Code) [21] is a calculation code developed by the U.S. Nuclear Regulatory Commission (NRC). It is mainly used for transient analysis in nuclear reactor systems. It is generally coupled with the PARCS code, which models the full core reactor, like Version 5's Donjon5 code.

TRACE is designed to simulate the thermal-hydraulic behaviour of nuclear reactor systems, particularly during transients, i.e. rapid changes in reactor operating conditions. These transients can include loss-of-coolant accidents, pipe ruptures, start-up and shutdown sequences, etc. The TRACE code uses advanced models to represent the physics of thermal-hydraulic phenomena in reactors, including single-phase and two-phase thermal-hydraulics, heat transfers, phase change phenomena (evapouration, condensation), fluid movements, etc.

The TRACE code implements a six-equation two-fluid model, distinguishing between vapour and liquid phases with separate equations for mass, momentum, and energy fields. This model necessitates intrinsic constitutive relations detailing the material properties of the fluid phases and the properties of solid structures interacting with the fluid. Additionally, it involves seven extrinsic constitutive relations, including three for the transfer of mass, momentum, and energy between the two phases, and four for boundary conditions governing the exchange of momentum and energy between structural walls and each contacting phase of the fluid.

The user manual serves as a comprehensive guide for both novice and experienced users. The paper [22] reviews the main equations used by the TRACE code.

2.4.3 PARCS/PATHS code

The PATHS (PARCS Advanced Thermal Hydraulic Solver) [1] code was created at the University of Michigan as part of research carried out for the U.S. Nuclear Regulatory Commission. Its objective is to solve the steady-state two-phase thermal hydraulic equations specific to a boiling water reactor (BWR), and to provide thermal-hydraulic feedback for BWR burnout calculations in combination with the PARCS (Purdue Advanced Reactor Core Simulator) neutronic code. The validation and main features of the code are presented in the article [1]. The code gives steady-state solutions of a two-phase flow. It uses spatial averaging of equations and a face-based numerical scheme. For thermal-hydraulic modelling, the code uses a drift flux model. The equation definition will be discussed in section 4. The modelling is similar in some respects to that of the `THM:` module.

The results of the PATHS code were experimentally validated using the experimental data from the FRIGG experimental facility [1]. The authors use a simple single-channel model consisting of a vertical channel with fixed inlet flow, enthalpy, and outlet pressure. They tested many test cases, changing the flow conditions, channel power, flow rate, pressure, and inlet. They performed those tests using four void correlations. All of those correlations showed strong agreement in void fraction with the experiment.

	Root-Mean-Square Error	**Maximum Error**
PATHS (EPRI)	3.5%	15.5%
PATHS (GE-ramp)	5.1%	17.2%
PATHS (Modified Bestion)	13.1%	22.3%
PATHS (HEM)	4.9%	8.8%

Table 2.2 Root-Mean-Square Difference and Maximum Difference in Percent Void for PATHS Compared to the Experimental FRIGG Results [1]

Table 5.4 shows the root-mean-square errors and the maximum per cent difference between the four different correlations used in PATHS and the experimental results from FRIGG. We note that the lowest error is achieved with the correlations of the EPRI model, while the maximum error is one of the lowest.

They also compare the result to the BFBT Validation case. Once again, the simulations show good accuracy compared to this benchmark.

A prevalent aspect found in contemporary BWR fuel designs involves employing partial-length fuel rods to enhance the coolant flow area within the upper sections of the flow channel. This approach aims to mitigate bundle pressure drop. The authors showed how they were able to deal with this modification of length. This length modification involves an axially varying geometry and flow area. The authors manage to simulate this anomaly by developing relationships for the pressure and velocity at the interface of two cells with an abrupt area change.

A basic single-channel scenario was established in both PATHS and TRACE to confirm the correct implementation of the variable flow area.

Simulate-3 is the full core simulation code of Studvisk similar to PARCS. PATHS have test their treatment of part-length rods on an existing SIMULATE-3 full-core model. The PATHS model exhibits outstanding performance compared to both the SIMULATE and TRACE models. When they simulate the mass flow rate, the TRACE and SIMULATE models both include a leakage path for flow to escape to the bypass. They also include an exchange of water between the active flow and the central water rod. Consequently, a mass flow rate correction was necessary to facilitate a meaningful code-to-code comparison.

A full-core PATHS model has been developed. It shows remarkable performance compared to SIMULATE and emphasizes the excellent methodology used in PATHS.

The PARCS/PATHS code is the most interesting due to its good convergence, results, and simple model. It is also closest to what we intend to do in this project. Therefore, it is possible to take inspiration from the equations and resolutions used in the code. We will present these equations when we return to drift flux model in the section 4. The main limitation of PATHS is the absence of the transient aspect of the simulation and it's proprietary design.

2.4.4 TWOPORFLOW

TWOPORFLOW is a 3D porous medium code for modelling two-phase flow in Boiling Water Reactor (BWR) cores for normal and accidental conditions conducted by Karlsruhe Institute of Technology(KIT) [23]. It solves six conservation equations (vapour and liquid mass, vapour and liquid momentum, and vapour and liquid energy) numerically on a Cartesian grid. It considers volume and area porosity for solid structures (e.g., fuel rods). The geometric complications are addressed through the Fractional Area Volume Obstacle Representation (FAVOR) technique and are closed out by empirical relationships. Major hypotheses are:

- Equal pressure for vapour and liquid phases.
- Heat transfer correlations are dependent on flow regime (e.g., Chen's correlation for nucleate boiling and Lockhart-Martinelli for interfacial drag).
- Turbulent viscosity from a reduced mixing-length model.
- Subcooled boiling was simulated through Unal's correlation for vapour generation.

For the numerical implementation, the code uses a semi-implicit ICE method (Implicit Continuous Eulerian) and staggered velocity grids. The steady-state results are obtained through pseudo-transient simulations. The flow characteristics are captured through subchannel-centered discretization (for example, 9×9 subchannels in BFBT tests), while empirical coefficients (for instance, mixing coefficient $\beta = 0.007$) account for the spacer impacts.

The code was validated for the BWR Full-size Fine-mesh Bundle Test (BFBT) benchmark [15]. Computed void fractions at assembly exits had a standard deviation of $\leq 10\%$ to experiments. The largest deviations were 10% for low void fractions ($\varepsilon = 0.4$) due to unheated rods and water channels. Results were comparable with COBRA-TF for accuracy.

For the transient tests the time-averaged errors are between 4.5-11.2 % over axial levels. Pressure drop trends conformed to experimental data, notwithstanding scatter in measurements. TWOPORFLOW demonstrated sufficient accuracy in predicting BWR thermal-hydraulic behaviour and was comparable to reference codes. Its porous media formulation is balanced between computational cost and resolution and, therefore, appropriate for core-wide calculations. Future work includes Critical Heat Flux (CHF) modelling and extension to Pressurized Water Reactor (PWR) conditions.

2.4.5 Other

We can also cite the code CATHARE from the CEA [24]. In the section 5 the code GeN-FOAM [25] will be mentioned and used to compare to our results. An overall list of these codes is presented in table 2.3. Note that only PATHS can be considered as a "simplified" thermal hydraulic code and can be considered as the equivalent of `THM`:

Code	**modelling**	**Main Equations**	**Strengths**	**Limitations**
COBRA-TF (CTF)	BWR thermal-hydraulics in steady-state operation	Two-fluid, three-field model: mass conservation (x3), momentum (x3), energy (x2)	High resolution at pin-cell level, experimentally validated	No transient modelling, computationally heavy
TRACE	Transient thermal-hydraulics	Two-fluid, six-equation model (mass, momentum, energy for each phase)	Capable of simulating accidents (LOCA, pipe ruptures)	Requires coupling with a neutron code for full reactor analysis
PARCS/ PATHS	Steady-state two-phase BWR flow	Drift-Flux model, face-based numerical scheme	Excellent convergence, advanced modelling of geometry and flow variations	No transient modelling, requires complementary models for rapid dynamics
CATHARE	Detailed thermal-hydraulics, transient simulations	Two-fluid, six-equation model with advanced closure laws	High precision, used for PWR safety analysis	Very complex, requires significant computational resources
Two porous flow	Two-phase flow in porous media and channels	Two-fluid model with closure equations for porous media	Accounts for flows in porous fuel and channel geometries	More specialized for compact cores or porous fuel applications

Table 2.3 Summary of main thermal-hydraulic codes

2.5 OpenFoam

OpenFoam [13] is a freely available computational fluid dynamics (CFD) software developed by OpenCFD Ltd. It was created at Imperial College in London during the 1980s, with its inaugural commercial release in 2004.

The main characteristics are:

- free of charge;
- open-source, i.e. the code can easily be modified in order to improve existing solvers and peer-reviewed;
- object-oriented through which the users can introduce new models and solvers (user-selectable) without changing the main code independently from the discretization scheme used, providing great flexibility and simplicity of use.

OpenFoam is written in C++ and runs only on Linux distributions. It uses the finite volume method (FVM) to discretize and solve complex fluid dynamics problems. OpenFoam incorporates its own mesh generator. However, it may be more appropriate to develop our own mesh generation tool. OpenFoam employs the FVM on a colocated grid arrangement, storing all relevant variables (pressure, velocity, etc.) at the centre of each cell to minimize computational cost. This approach contrasts with the staggered grid arrangement, where different variables can be defined on separate grids. While the staggered method exists, colocated grids offer several advantages, making them the preferred choice for most CFD software.

One key benefit of the colocated arrangement is the simplified pressure-velocity coupling, eliminating the "checkerboard instability" (pressure field oscillations). Techniques like the Rhie and Chow cure successfully address this issue. Another advantage is the reduced computational effort, as storing all variables on the same control volume minimizes memory usage and computation time. Additionally, colocated grids excel at handling complex geometries, particularly those with discontinuous boundary conditions.

OpenFoam's structure mirrors typical CFD software, comprising pre-processing, solver, and post-processing stages. In pre-processing, users define the mesh, initial and boundary conditions, and fluid properties. The solver specifies and discretizes the governing equations using FVM, solving them numerically. Post-processing handled externally to OpenFoam, involves visualization and plotting of results using tools like ParaView. Additionally, the sample utility extracts raw data from desired regions for further analysis and comparison with experimental results. The OpenFOAM structure is presented in fig 2.5.

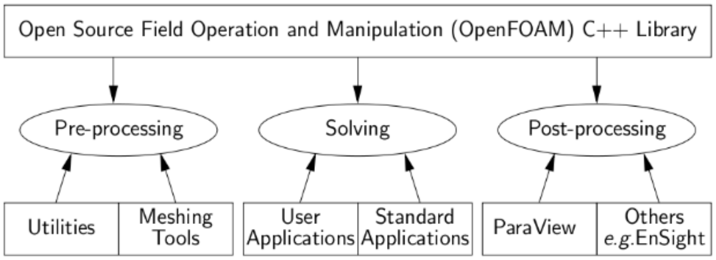


Figure 2.5 OpenCFD Limited. OpenFoam User Guide, 1.7.1 edition, 2010.

It offers a wide range of solvers and utilities for various fluid flow phenomena, such as incompressible and compressible flows, turbulence, multiphase flows, heat transfer, and combustion. Key features include its modular architecture, extensibility for customization, support for parallel computing, pre-processing, and post-processing tools.

2.5.1 The different subsolver

In the field of computational fluid dynamics, selecting the appropriate solver is crucial for accurately modelling complex physical phenomena. Below is a detailed explanation of the differences and characteristics of three OpenFOAM solvers: twoPhasesEulerFoam, reactingTwoPhaseEulerFoam, chtMultiRegionFoam, and chtMultiRegionTwoPhaseEulerFoam.

The twoPhaseEulerFoam

twoPhaseEulerFoam is a solver for a system of 2 non-reacting compressible fluid phases. One phase in this system is always dispersed. So, it is a good candidate for simulating bubble columns in gas/liquid systems or fluidized beds and spouted beds in gas-solid systems. The solver also solves the energy equation (enthalpy or internal energy) for both phases and couple them by the one-film resistance heat transfer model.

Various sub-models for interphase coupling are also provided, allowing the system to be modeled using a wide range of physical models. The solver can also be extended to incorporate custom sub-models. The source code of this solver can be found at: *source code*

The reactingTwoPhaseEulerFoam

The reactingTwoPhaseEulerFoam solver is designed to model systems involving two compressible fluid phases. These phases need to share a common pressure but can have distinct properties such as velocity and temperature. It employs the Euler-Euler approach to model the interactions between phases, making it particularly suitable for two-phase flows where exchange (momentum and energy) between the two phases is non-negligible.

It can handle compressible two-phase flows with a shared pressure field and support a huge variety of species. In fact only liquid and gaseous water will be interesting for this study. It offers runtime-selectable models for phase interactions, including momentum, heat and mass transfer between phases. The user can customize the model according to the specific needs of the simulation.

The chtMultiRegionFoam

The *chtMultiRegionFoam* is a transient solver for buoyant, turbulent fluid flow and solid heat conduction with conjugate heat transfer between solid and fluid regions. It allows the coupling of conduction in a solid medium with the thermal convection in a fluid. To use the solver, it is important to divide the domain into multiple distinct regions, each with its own properties and transport equations.

The solver incorporates models for turbulent flows and buoyancy effects in fluids, enabling accurate simulation of heat transfer in systems with these significant factors.

The chtMultiRegionTwoPhaseEulerFoam solver

The *chtMultiRegionTwoPhaseEulerFoam* is a powerful solver that merges the capabilities of two existing solvers: *reactingTwoPhaseEulerFoam* and *chtMultiRegionFoam*. This allows it to tackle simulations involving coupled solid and fluid systems, including reactions. The solver achieves this by employing a special boundary condition called *turbulentTemperatureTwoPhaseRadCoupledMixed* to handle interactions between different regions.

The *reactingTwoPhaseEulerFoam* solver is used for a system of two compressible fluid phases with a common pressure, but otherwise separate properties. The type of phase model is run-time selectable and can optionally represent multiple species and in-phase reactions. The phase system is also run-time selectable and can optionally represent different types of momentum, heat and mass transfer.

2.5.2 Coupled codes

The article [26] provides a benchmark of existing coupled code with OpenFoam. The article reviews the main features of Open-Foam, such as the structure of the code, the finite volume discretization, unstructured meshes, operator splitting, parallelization, and the multi-zone and multi-region treatment. Open-Foam-based code such as GeN-Foam solves momentum, mass and energy conservation equations using the available operators for laplacian, divergence and gradient with a standard merged PISO-SIMPLE loop for pressure-velocity coupling. Turbulent flows are modelled using $k - \epsilon$ or $k - \omega$ models. The code has been recently updated for two-phase flow and uses a porous medium approach. Concerning neutronics and fuel burnup, the development of solvers for multi-group diffusion, simplified P_3 (SP3) and discrete ordinates methods can be achieved within OpenFoam by leveraging the availability of Laplacian and divergence operators. This means that Open-Foam provides the necessary mathematical tools, making it feasible to efficiently implement solvers for these radiation transport methods. The author lists the existing simulation and neutronic modelisation using Open-Foam. The article shows the main features of the code, such as geometrical flexibility, full-matrix solutions, matrix preconditioners, multi-physics simulation and moving mesh. Thermo-mechanics is mentioned, but it is not the case of the master thesis.

OpenFOAM's modular design facilitates coupling with external codes. This capability is mainly due to the use of finite volumes (to improve information exchange), the use of unstructured meshes (allows tailoring of the geometry), the integration of mesh conversion tools, the use of C++ (object-oriented code allows OpenFoam to be suited for cross-compilation) and surface-based (inlet/outlet) coupling schemes. We can also mention that some research in the nuclear field were applied to developing software and methods for model Order Reduction. It is a set of methodologies employed to reduce the computational footprint of numerical analysis.

In 2022, according to this article [26] the main advanced solvers were:

- High-temperature Ga Cooled Reactors
- Gen-Foam solver
- ContainmentFOAM solver for containment modelling
- OFFBEAT solver for fuel modelling

In the remainder of this study, we will focus on GeN-Foam, which appears to be the most advanced neutronic/thermohydraulic OpenFoam-based solver available today. GeN-FOAM will be mentioned and described later section 5

2.5.3 Conclusion

This literature review outlines the key concepts of reactor physics and thermohydraulics. It captures the main theory and existing tools to model the phase flows in a nuclear reactor core. From the introduction of neutronics and thermohydraulics, which are fundamental for coupled simulations, through the benchmark experiment to the existing thermal-hydraulic codes, this literature review shows a gap in the code for nuclear science.

In fact, some CFD codes are open-source and open-access, but are pretty hard to use for reactor physics. Moreover, as mentioned in section 1.4 and in this chapter, no open source nuclear core code can model the thermohydraulics of a BWR. This master thesis modestly attempts to help to fill this gap in available open-source solutions.

CHAPTER 3 Mathematical modelisation of two-phases flow

In general, two different approaches can be used to model two-phase flows, each offering different advantages and disadvantages depending on the technical considerations and the problem studied. The first method, called the particle-source-cell method, uses a lagrangian approach for dispersed phases. It's a way of following the movement of each particle one by one in a mix. The continuous phase is studied using an Eulerian approach. This Euler-Lagrange approach is intuitive but requires considerable computational power, mainly because of the way in which the dispersed phase is modelled and the necessity to model each particle-particle interaction.

The second method uses the Eulerian treatment of both phases. This method simplifies calculations and is widely used for its practicality. However, it may require additional modelling and correlation to represent phase interactions. Using an Eulerian modelling approach creates several levels of complexity in numerical modeling. The most basic method is to model a homogeneous mixture. Additional information can be added, such as correlations and terms that model phase velocities and mass and energy exchanges. For more precise modeling, separate concervation equations can be used for the fluids studied. Finally, when we want to take account of the different types of bubble in our modeling, the MUSIG (Multiple Size Group) method can be used. This is a population balance approach used to simulate poly-dispersed fluids, in particular bubble flows, taking into account bubble coalescence and rupture.

3.1 Conservation equations

The local equations for two-phase flow are obtained by applying the Reynolds transport theorem [27] to a control volume V_c. Local equations for two-phase flow are the Naviers-Stokes equations.

To these equations, we perform an averaging called "Reynolds-averaged". The goal of the Reynolds-averaged Navier-Stokes equations (RANS) is to reasonably describe the averaged transport, energy, momentum, or other equation without too many computational resources. The RANS equations are used to describe turbulent flow. There are different types of averaged equations. They can be volume averaged and time-averaged.

Most theoretical studies rely on the Navier-Stokes equations describing both fluids and averages in space. The indices $k = g$ (or ℓ) refer respectively to gas (or liquid), the superscripts

i and w indicate that the variable is related to the gas-liquid interface or the wall. Variables not yet averaged are denoted with $\sim$.

In the case where gravity is the only mass force, the Navier-Stokes equations are expressed as follows:

$$\begin{cases} \text{- Mass conservation:} \\ \dfrac{\partial \tilde{\rho}_k}{\partial t} + \nabla \cdot (\tilde{\rho}_k \tilde{\vec{v_k}}) = 0 \\ \\ \text{- Momentum conservation :} \\ \dfrac{\partial \tilde{\rho}_k \tilde{\vec{v_k}}}{\partial t} + \nabla \cdot (\tilde{\rho}_k \tilde{\vec{v}}_k \otimes \tilde{\vec{v_k}}) = \tilde{\rho}_k \vec{g} - \nabla \tilde{P}_k + \nabla \cdot \tilde{\bar{\bar{\tau}}}_{\mathbf{k}} \\ \\ \text{- Energy conservation :} \\ \dfrac{\partial}{\partial t}\left(\tilde{\rho}_k \tilde{h}_k\right) + \nabla \cdot \left(\tilde{\rho}_k \tilde{\vec{v}}_k \tilde{h}_k\right) = \dfrac{d}{dt}\tilde{P}_k + \tilde{\bar{\bar{\tau}}}_{\mathbf{k}} \cdot \nabla \cdot \tilde{\vec{v_k}} + \nabla \cdot (\tilde{k}_k \nabla \tilde{T}_k) \\ \\ \text{- Equations of states (from water-steam tables):} \\ \tilde{T}_k(\tilde{P}_k, \tilde{h}_k) \ ; \ \tilde{\rho}_k(\tilde{P}_k, \tilde{T}_k) \ ; \ \tilde{\mu}_k(\tilde{P}_k, \tilde{T}_k) \ ; \ \tilde{k}_k(\tilde{P}_k, \tilde{T}_k) \end{cases} \tag{3.1}$$

For the phase k we denote its local density by $\tilde{\rho}_k$, its local velocity by $\tilde{\vec{v}}_k$, its local enthalpy by $\tilde{h}_k$, its local pressure by $\tilde{P}_k$, the local viscous stress term $\tilde{\bar{\bar{}}}_{\mathbf{k}}$, $\tilde{k}_k$ its local thermal conductivity and $\tilde{T}_k$ its local temperature. $\vec{g}$ is the gravitational force constant.

The term $-\tilde{k}_k \nabla \tilde{T}_k$ represents the heat flux by conduction where $\tilde{k}_k$ is the fluid's thermal conductivity coefficient (Fourier's law).

Finally, the boundary conditions (noting $[\![f_k]\!] = f_g + f_\ell$) are:

- For mass transfer:

$$\begin{gathered} [\![\tilde{\rho}_k (\tilde{\vec{v_k}} - \vec{v^i}) \cdot \vec{n_k^i}]\!] = 0 \\ \tilde{\rho}_k (\tilde{\vec{v_k}} - \vec{v^i}) \cdot \vec{n_k^i} = \Gamma_k \end{gathered} \tag{3.2}$$

This means that the mass leaving phase g through the unit surface element of the interface is equal to that entering phase l and vice versa. Γ_k represents the mass flux transforming from phase 'k' to the other phase (e.g., vaporization).

- For momentum transfer:

$$[\![\tilde{P}_k \vec{n_k^i} + \tilde{\bar{\bar{\tau}}}_{\mathbf{k}} \cdot \vec{n_k^i} - \tilde{\rho}_k \tilde{\vec{v_k}} (\tilde{\vec{v_k}} - \vec{v^i}) \cdot \vec{n_k^i}]\!] = \gamma C_i \vec{n_\ell^i} \tag{3.3}$$

Where γ is the surface tension and C_i is the local curvature of the gas.

- For enthalpy transfer :

$$\left[\!\left[\tilde{P_k}\tilde{\vec{v}}_k \cdot \vec{n_k^i} - \tilde{\vec{v_k}} \cdot \left(\tilde{\bar{\bar{\tau}}}_{\mathbf{k}} \cdot \vec{n_k^i}\right) - \tilde{k}_k \nabla \tilde{T}_k \cdot \vec{n_k^i} + \tilde{h}_k \tilde{\rho}_k (\tilde{\vec{v_k}} - \vec{v^i}) \cdot \vec{n_k^i}\right]\!\right] = 0 \tag{3.4}$$

We introduce the indicator function χ_k of the fluid phase ($k = g, \ell$):

$$\chi_k(\mathbf{x}, t) = \begin{cases} 1, & \text{if phase } k \text{ is present at } (\mathbf{x}, t) \\ 0, & \text{otherwise} \end{cases} \tag{3.5}$$

The derivatives of χ_k are zero everywhere except at the interfaces. Assuming the walls are immobile, thus $\vec{v_k}\delta^w = 0$, we obtain:

$$\begin{array}{rcccc} \frac{\partial \chi_k}{\partial t} & = & -\vec{v^i}.\vec{n_k^i}\delta^i & + & 0 \\ \nabla \chi_k & = & \vec{n_k^i}\delta^i & + & \vec{n_k^w}\delta^w \\ \tilde{\vec{v_k}}\nabla \chi_k & = & \tilde{\vec{v_k}}\vec{n_k^i}\delta^i & + & 0 \\ \tilde{\vec{v_k}}\tilde{\vec{v_k}}\nabla \chi_k & = & \underbrace{\tilde{\vec{v_k}}\tilde{\vec{v_k}}\vec{n_k^i}\delta^i}_{\text{gas-liquid interface}} & + & \underbrace{\quad 0 \quad}_{\text{wall-fluid interface}} \end{array} \tag{3.6}$$

With δ^i and δ^w being Dirac functions respectively non-zero at the gas-liquid interface and at the wall.

3.1.1 Averaging conservation equation

Space averaged variables are denoted with $\langle - \rangle$. Let's denote spatial averaging using a cube of sides L as the control volume.

$$\langle \tilde{\phi} \rangle(\vec{x}, t) = \frac{1}{L^3} \int_{x_1+L/2}^{x_1-L/2} \int_{x_2+L/2}^{x_2-L/2} \int_{x_3+L/2}^{x_3-L/2} \tilde{\phi}(\vec{x}', t) dx_3' dx_2' dx_1' \tag{3.7}$$

To integrate over a control volume independent of phase presence, we multiply the Navier-Stokes equations by χ_k ($k = g$ or ℓ). Then, we perform spatial averaging.

$$\begin{aligned} &\langle \chi_k \frac{\partial \tilde{\rho}_k}{\partial t} \rangle + \langle \chi_k \nabla \cdot (\tilde{\rho}_k \tilde{\vec{v_k}}) \rangle = 0 \\ &\langle \chi_k \frac{\partial \tilde{\rho}_k \tilde{\vec{v_k}}}{\partial t} \rangle + \langle \chi_k \nabla \cdot (\tilde{\rho}_k \tilde{\vec{v_k}} \otimes \tilde{\vec{v_k}}) \rangle = \langle \chi_k \tilde{\rho}_k \vec{g} \rangle - \langle \chi_k \nabla \tilde{P}_k \rangle + \langle \chi_k \nabla \cdot \tilde{\bar{\bar{\tau}}}_k \rangle \\ &\langle \chi_k \frac{\partial}{\partial t}\left(\tilde{\rho}_k \tilde{h}_k\right) \rangle + \langle \chi_k \nabla \cdot \left(\tilde{\rho}_k \tilde{h}_k \tilde{\vec{v_k}}\right) \rangle = \langle \chi_k \frac{d}{dt} \tilde{P}_k \rangle + \langle \chi_k \nabla \cdot \tilde{\bar{\bar{\tau}}}_k \tilde{\vec{v_k}} \rangle - \langle \chi_k \nabla \cdot (-\tilde{k_k} \nabla \tilde{T}_k) \rangle \end{aligned} \tag{3.8}$$

Note that:

$$
\begin{aligned}
\langle \chi_k \frac{\partial \tilde{\rho}_k}{\partial t} \rangle &= \frac{\partial \langle \chi_k \tilde{\rho}_k \rangle}{\partial t} - \langle \tilde{\rho}_k \frac{\partial \chi_k}{\partial t} \rangle = \frac{\partial \langle \chi_k \tilde{\rho}_k \rangle}{\partial t} + \langle \tilde{\rho}_k \tilde{\vec{v}}^i \cdot \vec{n}^i_k \delta^i \rangle \\
\langle \chi_k \frac{\partial \tilde{\rho}_k \tilde{\vec{v}}_k}{\partial t} \rangle &= \frac{\partial \langle \chi_k \tilde{\rho}_k \tilde{\vec{v}}_k \rangle}{\partial t} - \langle \tilde{\rho}_k \tilde{\vec{v}}_k \frac{\partial \chi_k}{\partial t} \rangle = \frac{\partial \langle \chi_k \tilde{\rho}_k \tilde{\vec{v}}_k \rangle}{\partial t} + \langle \tilde{\rho}_k \tilde{\vec{v}}_k \tilde{\vec{v}}^i \cdot \vec{n}^i_k \delta^i \rangle \\
\langle \chi_k \nabla \cdot (\tilde{\rho}_k \tilde{\vec{v}}_k) \rangle &= \nabla \cdot \langle \chi_k \tilde{\rho}_k \tilde{\vec{v}}_k \rangle - \langle \tilde{\rho}_k \tilde{\vec{v}}_k \cdot \vec{n}^i_k \delta^i \rangle \\
\langle \chi_k \nabla \cdot (\tilde{\rho}_k \tilde{\vec{v}}_k \otimes \tilde{\vec{v}}_k) \rangle &= \nabla \cdot \langle \chi_k \tilde{\rho}_k \tilde{\vec{v}}_k \otimes \tilde{\vec{v}}_k \rangle - \langle \tilde{\rho}_k \tilde{\vec{v}}_k \otimes \tilde{\vec{v}}_k \cdot \vec{n}^i_k \delta^i \rangle \\
\langle \tilde{P}_k \nabla \chi_k \rangle &= \langle \tilde{P}_k \vec{n}^i_k \delta^i \rangle + \langle \tilde{P}_k \vec{n}^w_k \delta^w \rangle \\
\langle \tilde{\bar{\bar{\tau}}}_k \cdot \nabla \chi_k \rangle &= \langle \tilde{\bar{\bar{\tau}}}_k \cdot \vec{n}^i_k \delta^i \rangle + \langle \tilde{\bar{\bar{\tau}}}_k \cdot \vec{n}^w_k \delta^w \rangle \\
\langle \chi_k \frac{\partial \tilde{\rho}_k \tilde{h}_k}{\partial t} \rangle &= \frac{\partial \langle \chi_k \tilde{\rho}_k \tilde{h}_k \rangle}{\partial t} - \langle \frac{\partial \chi_k}{\partial t} \tilde{\rho}_k \tilde{h}_k \rangle = \frac{\partial \langle \chi_k \tilde{\rho}_k \tilde{h}_k \rangle}{\partial t} + \langle \tilde{\rho}_k \tilde{h}_k \vec{v}^i \cdot \vec{n}^i_k \delta^i \rangle \\
\langle \chi_k \nabla \cdot \left(\tilde{\rho}_k \tilde{h}_k \tilde{\vec{v}}_k \right) \rangle &= \nabla \cdot \langle \chi_k \tilde{\rho}_k \tilde{h}_k \tilde{\vec{v}}_k \rangle - \langle \tilde{\rho}_k \tilde{h}_k \tilde{\vec{v}}_k \vec{n}^i_k \delta^i \rangle \\
\langle \chi_k \frac{d}{dt} \tilde{P}_k \rangle &= \frac{d}{dt} \langle \chi_k \tilde{P}_k \rangle + \langle \tilde{P}_k \tilde{\vec{v}}_i \vec{n}^i_k \delta^i \rangle \\
\langle \chi_k \nabla \cdot \tilde{\bar{\bar{\tau}}}_{\mathbf{k}} \tilde{\vec{v}}_k \rangle &= \nabla \cdot \langle \chi_k \tilde{\bar{\bar{\tau}}}_{\mathbf{k}} \tilde{\vec{v}}_k \rangle - \langle \tilde{\vec{v}}_k \cdot \left(\tilde{\bar{\bar{\tau}}}_{\mathbf{k}} \cdot \vec{n}^i_k \right) \delta^i \rangle \\
\langle \chi_k \nabla \cdot (-\tilde{k}_k \nabla \tilde{T}_k) \rangle &= \nabla \cdot \langle \chi_k (-\tilde{k}_k \nabla \tilde{T}_k) \rangle - \langle (-\tilde{k}_k \nabla \tilde{T}_k) \nabla \chi_k \rangle = \nabla \cdot \langle \chi_k (-\tilde{k}_k \nabla \tilde{T}_k) \rangle \\
&\quad - \langle (-\tilde{k}_k \nabla \tilde{T}_k) . \vec{n}^i_k \delta^i \rangle - \langle (-\tilde{k}_k \nabla \tilde{T}_k) . \vec{n}^w_k \delta^w \rangle
\end{aligned}
\tag{3.9}
$$

Finally:

$$
\begin{aligned}
&\frac{\partial \langle \chi_k \tilde{\rho}_k \rangle}{\partial t} + \nabla \cdot \langle \chi_k \tilde{\rho}_k \tilde{\vec{v}}_k \rangle = \langle \tilde{\rho}_k (\tilde{\vec{v}}_k - \vec{v}^i) . \vec{n}^i_k \delta^i \rangle \\
&\frac{\partial \langle \chi_k \tilde{\rho}_k \tilde{\vec{v}}_k \rangle}{\partial t} + \nabla \cdot \langle \chi_k \tilde{\rho}_k \tilde{\vec{v}}_k \otimes \tilde{\vec{v}}_k \rangle = -\nabla \langle \chi_k \tilde{P}_k \rangle + \langle \chi_k \tilde{\rho}_k \rangle \vec{g} + \nabla \cdot \langle \chi_k \tilde{\bar{\bar{\tau}}}_{\mathbf{k}} \rangle + \langle (\tilde{\rho}_k \tilde{\vec{v}}_k (\tilde{\vec{v}}_k - \vec{v}^i)) . \vec{n}^i_k \delta^i \rangle \\
&+ \langle \tilde{P}_k . \vec{n}^i_k \delta^i \rangle - \langle \tilde{\bar{\bar{\tau}}}_{\mathbf{k}} \vec{n}^i_k \delta^i \rangle + \langle \tilde{P}_k . \vec{n}^w_k \delta^w \rangle - \langle \tilde{\bar{\bar{\tau}}}_{\mathbf{k}} \vec{n}^w_k \delta^w \rangle \\
&\frac{\partial \langle \chi_k \tilde{\rho}_k \tilde{h}_k \rangle}{\partial t} + \langle \tilde{\rho}_k \tilde{h}_k \vec{v}^i . \vec{n}^i_k \delta^i \rangle + \nabla \cdot \langle \chi_k \tilde{\rho}_k \tilde{h}_k \tilde{\vec{v}}_k \rangle - \langle \tilde{\rho}_k \tilde{h}_k \tilde{\vec{v}}_k \vec{n}^i_k \delta^i \rangle = \frac{d \langle \chi_k \tilde{P}_k \rangle}{dt} + \langle \tilde{P}_k \tilde{\vec{v}}_i \vec{n}^i_k \delta^i \rangle \\
&+ \nabla \cdot \langle \chi_k \tilde{\bar{\bar{\tau}}}_{\mathbf{k}} \tilde{\vec{v}}_k \rangle - \langle \tilde{\vec{v}}_k \cdot \left(\tilde{\bar{\bar{\tau}}}_{\mathbf{k}} \cdot \vec{n}^i_k \right) \delta^i_k \rangle + \nabla \cdot \langle \chi_k (-\tilde{k}_k \nabla \tilde{T}_k) \rangle + \langle (\tilde{k}_k \nabla \tilde{T}_k) . \vec{n}^i_k \delta^i \rangle + \langle (\tilde{k}_k \nabla \tilde{T}_k) . \vec{n}^w_k \delta^w \rangle
\end{aligned}
\tag{3.10}
$$

For boundary conditions at the gas-liquid interfaces. the transfer relations are written:

Mass transfer:

$$
[\![\langle \tilde{\rho}_k (\tilde{\vec{v}}_k - \tilde{\vec{v}}^i) . \vec{n}^i_k \delta^i \rangle]\!] = 0 \tag{3.11}
$$

Momentum transfer:

$$[\![-\langle \tilde{P}_k . \vec{n_k^i} \delta_k^i \rangle + \langle \tilde{\bar{\bar{\tau}}}_{\mathbf{k}} \vec{n_k^i} \delta^i \rangle - \langle \tilde{\rho}_k \tilde{\vec{v_k}} (\tilde{\vec{v_k}} - \vec{v^i}) . \vec{n_k^i} \delta_k^i \rangle]\!] = \langle \gamma C_i \vec{n_\ell^i} \rangle \tag{3.12}$$

Energy transfer:

$$\left[\!\!\left[\langle \tilde{P}_k \tilde{\vec{v_k}} \cdot \vec{n_k^i} \delta_k^i \rangle - \langle \tilde{\vec{v_k}} \cdot \left(\tilde{\bar{\bar{\tau}}}_{\mathbf{k}} \cdot \vec{n_k^i} \right) \delta_k^i \rangle - \langle \tilde{k}_k \nabla \tilde{T}_k \cdot \vec{n_k^i} \delta_k^i \rangle - \langle \tilde{h}_k \tilde{\rho}_k (\tilde{\vec{v_k}} - \tilde{\vec{v^i}}) \cdot \vec{n_k^i} \tilde{\delta}_k^i \rangle \right]\!\!\right] = [\![q_k^{i\,\prime\prime}]\!] = 0 \tag{3.13}$$

Introducing:

$$\begin{array}{lcll}
\Phi & = & 1 - \langle \chi_s \rangle & \\
\Phi\varepsilon_k & = & \langle \chi_k \rangle & \text{for } k = g \text{ or } \ell \\
\rho_k & = & \langle \chi_k \tilde{\rho}_k \rangle / \Phi\varepsilon_k & \text{for } k = g \text{ or } \ell \\
h_k & = & \langle \chi_k \tilde{\rho}_k \tilde{h}_k \rangle / \Phi\varepsilon_k \rho_k & \text{for } k = g \text{ or } \ell \\
v_k & = & \langle \chi_k \tilde{\rho}_k \tilde{v}_k \rangle / \Phi\varepsilon_k \rho_k & \text{for } k = g \text{ or } \ell \\
\Phi\varepsilon_k \tau_{\mathbf{k}}^{\tilde{\bar{\bar{\mathbf{Re}}}}} & = & \rho_k \vec{v}_k \otimes \vec{v}_k - \langle \chi_k \tilde{\rho}_k \tilde{\vec{v_k}} \otimes \tilde{\vec{v_k}} \rangle & \text{for } k = g \text{ or } \ell \\
P_k & = & \langle \chi_k \tilde{P}_k \rangle / \Phi\varepsilon_k & \text{for } k = g \text{ or } \ell \\
\bar{\bar{\tau}}_{\mathbf{k}} & = & \langle \chi_k \tilde{\bar{\bar{\tau}}}_{\mathbf{k}} \rangle / \Phi\varepsilon_k & \text{for } k = g \text{ or } \ell \\
\vec{M}_k^i & = & \langle (\tilde{\rho}_k \tilde{\vec{v_k}} (\tilde{\vec{v_k}} - \ \tilde{\vec{v}}^i)) . \vec{n_k^i} \delta^i \rangle + \langle (\tilde{P}_k - P_k) \vec{n_k^i} \delta^i \rangle - \langle \tilde{\bar{\bar{\tau}}}_{\mathbf{k}} . \vec{n_k^i} \delta^i \rangle & \text{for } k = g \text{ or } \ell \\
\vec{M}_k^w & = & \langle (\tilde{P}_k - P_k) \vec{n_k^w} \delta^w \rangle - \langle \tilde{\bar{\bar{\tau}}}_{\mathbf{k}} . \vec{n_k^w} \delta^w \rangle & \text{for } k = g \text{ or } \ell \\
\Gamma_k & = & \langle \tilde{\rho}_k (\tilde{\vec{v_k}} - \tilde{\vec{v^i}}) . \vec{n_k^i} \delta^i \rangle & \text{for } k = g \text{ or } \ell \\
\vec{\kappa} & = & \langle \gamma C_i \vec{n_\ell^i} \rangle & \\
q_k^{w\prime\prime} & = & \langle (\tilde{k}_k \nabla \tilde{T}_k) . \vec{n_k^w} \delta^w \rangle & \text{for } k = g \text{ or } \ell \\
q_k^{i\,\prime\prime} & = & \langle \tilde{P}_k \tilde{\vec{v_k}} \cdot \vec{n_k^i} \delta_k^i \rangle - \langle \tilde{\vec{v_k}} \cdot \left(\tilde{\bar{\bar{\tau}}}_{\mathbf{k}} \cdot \vec{n_k^i} \right) \delta_k^i \rangle - \langle \tilde{k}_k \nabla \tilde{T}_k \cdot \vec{n_k^i} \delta_k^i \rangle - \langle \tilde{h}_k \tilde{\rho}_k (\quad \tilde{\vec{v_k}} - \tilde{\vec{v^i}}) \cdot \vec{n_k^i} \delta_k^i \rangle &
\end{array} \tag{3.14}$$

It should be noted that the previous list contains averaged values rather than real values. We do not have a zero mean velocity at the wall because the control volume contains areas with non-zero velocities.

As $\tilde{\rho}_k$ and $\tilde{\mu}_k$ usually have very small variation inside a control volume, a usual simplification is :

$$\begin{array}{lcll}
\langle \chi_k \tilde{\rho}_k \tilde{\vec{v_k}} \rangle & \simeq & \rho_k \langle \chi_k \tilde{\vec{v_k}} \rangle = \rho_k \Phi\varepsilon_k \vec{v_k} & \text{for } k = g \text{ or } \ell \\
\langle \chi_k \tilde{\mu}_k \tilde{\vec{v_k}} \rangle & \simeq & \mu_k \langle \chi_k \tilde{\tilde{\vec{v_k}}} \rangle = \mu_k \Phi\varepsilon_k \vec{v_k} & \text{for } k = g \text{ or } \ell
\end{array} \tag{3.15}$$

If a bubble (or solid, respectively) is completely inside the control volume, the gradient of the void fraction (or porosity, respectively) does not change with the displacement of the control volume.

Noting that :

$$\nabla(\Phi\varepsilon P) = \Phi\varepsilon\nabla P + P\varepsilon\nabla\Phi + P\Phi\nabla\varepsilon = \Phi\varepsilon\nabla P + P_k\vec{n_k^w} + P_k\vec{n_k^i} \tag{3.16}$$

With $P_k\vec{n_k^w}+P_k\vec{n_k^i}$ nullify on average as vapour and solid presents close surface. In accordance with the definition, so $\nabla(\Phi\varepsilon P) \approx \Phi\varepsilon\nabla P$. This approximation will be wrong on stratified flow. By the simple definition of the indicator function, we have:

$$\varepsilon_g + \varepsilon_\ell = 1 \tag{3.17}$$

Then, space averaged Navier-Stokes equations becomes :

$$\begin{array}{rcl}
\dfrac{\partial\Phi\varepsilon_k\rho_k}{\partial t} + \nabla\cdot(\Phi\varepsilon_k\rho_k\vec{v_k}) & = & \Gamma_k \\
\dfrac{\partial}{\partial t}(\Phi\varepsilon_k\rho_k\vec{v_k}) + \nabla\cdot(\Phi\varepsilon_k\rho_k\vec{v_k}\otimes\vec{v_k}) & = & -\Phi\varepsilon_k\nabla P_k + \Phi\varepsilon_k\rho_k\vec{g} + \nabla\cdot(\Phi\varepsilon_k(\bar{\bar{\tau}}_{\mathbf{k}} + \tau_{\mathbf{k}}^{\bar{\bar{\mathbf{Re}}}})) \\
 & & +\vec{M}_k^i + \vec{M}_k^w \\
\dfrac{d}{dt}(\Phi\varepsilon_k\rho_k h_k) + \nabla\cdot(\rho_k\varepsilon_k h_k\vec{v_k}) & = & \dfrac{\partial}{\partial t}(\Phi\varepsilon_k P_k) + \nabla\cdot\Phi\varepsilon_k\bar{\bar{\tau}}_{\mathbf{k}}\vec{v_k} + \nabla\cdot\Phi\varepsilon_k(-k_k\nabla T_k) \\
 & & +q_k^{w\prime\prime} + q_k^{i\prime\prime}
\end{array} \tag{3.18}$$

For boundary conditions between the two phases,

$$\begin{array}{l}
\llbracket\Gamma_k\rrbracket = 0 \\
\llbracket\vec{M}_k^i\rrbracket = -\vec{\kappa} \\
\llbracket\vec{q}_k^{w\prime\prime} + q_k^{i\prime\prime}\rrbracket = q_m^{w\prime\prime}
\end{array} \tag{3.19}$$

3.1.2 Mixture equation

If we introduce the boundary conditions into the equations, we have :

$$
\begin{aligned}
\frac{\partial \Phi\varepsilon\rho_g}{\partial t} + \nabla\cdot(\Phi\varepsilon\rho_g\vec{v}_g) &= \Gamma_g \\
\frac{\partial \Phi(1-\varepsilon)\rho_\ell}{\partial t} + \nabla\cdot(\Phi(1-\varepsilon)\rho_\ell\vec{v}_\ell) &= -\Gamma_g \\
\\
\frac{\partial}{\partial t}(\Phi\varepsilon\rho_g\vec{v}_g) + \nabla\cdot(\Phi\varepsilon\rho_g\vec{v}_g\otimes\vec{v}_g) &= -\Phi\varepsilon\nabla P_g + \Phi\varepsilon\rho_g\vec{g} + \nabla\cdot(\Phi\varepsilon(\bar{\bar{\tau}}_{\mathbf{g}} + \tau_{\mathbf{g}}^{\bar{\bar{\mathbf{Re}}}})) \\
&\quad +\vec{M}_g^i + \vec{M}_g^w + \vec{\kappa} \\
\frac{\partial}{\partial t}(\Phi(1-\varepsilon)\rho_\ell\vec{v}_\ell) + \nabla\cdot(\Phi(1-\varepsilon)\rho_\ell\vec{v}_\ell\otimes\vec{v}_\ell) &= -\Phi(1-\varepsilon)\nabla P_\ell + \Phi(1-\varepsilon)\rho_\ell\vec{g} \\
&\quad +\nabla\cdot(\Phi(1-\varepsilon)(\bar{\bar{\tau}}_\ell + \tau_\ell^{\bar{\bar{\mathbf{Re}}}})) - \vec{M}_g^i + \vec{M}_\ell^w \\
\frac{\partial}{\partial t}(\Phi\varepsilon\rho_g h_g) + \nabla\cdot(\rho_g\varepsilon h_g\vec{v}_g) &= \frac{d}{dt}(\Phi\varepsilon P_g) + \nabla\cdot\Phi\varepsilon\bar{\bar{\tau}}_{\mathbf{g}}\vec{v}_g \\
&\quad +\nabla\cdot\Phi\varepsilon(-k_g\nabla T_g) + q_g^{w\prime\prime} + q_g^{i\prime\prime} \\
\frac{\partial}{\partial t}(\Phi(1-\varepsilon)\rho_\ell h_\ell) + \nabla\cdot(\rho_\ell(1-\varepsilon)h_\ell\vec{v}_\ell) &= \frac{d}{dt}(\Phi(1-\varepsilon)P_\ell) + \nabla\cdot\Phi(1-\varepsilon)\bar{\bar{\tau}}_\ell\vec{v}_\ell \\
&\quad +\nabla\cdot\Phi(1-\varepsilon)(-k_\ell\nabla T_\ell) + q_\ell^{w\prime\prime} - q_\ell^{i\prime\prime}
\end{aligned}
\tag{3.20}
$$

resulting in 8 equations but many more unknowns $\Phi, \varepsilon, \vec{v}_g, \vec{v}_\ell, P_g, P_\ell, \bar{\bar{\tau}}_{\mathbf{g}}, \tau_{\mathbf{g}}^{\bar{\bar{\mathbf{Re}}}}, \bar{\bar{\tau}}_\ell, \tau_\ell^{\bar{\bar{\mathbf{Re}}}}, \vec{\kappa}$, $q_\ell^{i\prime\prime}$, $q_g^{i\prime\prime}$ and $\vec{M}_k^w$. q_k^i is the heating related to the viscous effect that can be neglected in the context of this work. According to equation 3.19, the sum of the q_k^i is 0. For the rest of the terms, it is therefore necessary to propose additional closure relations.

The so-called mixture equation is the sum of equations for momentum:

$$
\begin{aligned}
\frac{\partial \Phi(\varepsilon_g\rho_g + \varepsilon_\ell\rho_\ell)}{\partial t} &+ \nabla\cdot(\Phi(\varepsilon_g\rho_g\vec{v}_g + \varepsilon_\ell\rho_\ell\vec{v}_\ell)) = 0 \\
\frac{\partial}{\partial t}(\Phi(\varepsilon_g\rho_g\vec{v}_g + \varepsilon_\ell\rho_\ell\vec{v}_\ell)) &+ \nabla\cdot(\Phi(\varepsilon_g\rho_g\vec{v}_g\otimes\vec{v}_g + \varepsilon_\ell\rho_\ell\vec{v}_\ell\otimes\vec{v}_\ell)) = -\Phi(\varepsilon_g + \varepsilon_\ell)\nabla(P_g + P_\ell) + \vec{\kappa} \\
&+ \Phi((\varepsilon_g\rho_g + \varepsilon_\ell\rho_\ell)\vec{g} + \nabla\cdot(\varepsilon_g(\bar{\bar{\tau}}_{\mathbf{g}} + \tau_{\mathbf{g}}^{\bar{\bar{\mathbf{Re}}}})) + \nabla\cdot(\varepsilon_\ell(\bar{\bar{\tau}}_\ell + \tau_\ell^{\bar{\bar{\mathbf{Re}}}})) \\
&+ \vec{M}_g^w + \vec{M}_\ell^w \\
\frac{\partial}{\partial t}(\Phi(\varepsilon_g\rho_g h_g + \varepsilon_\ell\rho_\ell h_\ell)) &+ \nabla\cdot(\Phi(\varepsilon_g\rho_g h_g\vec{v}_g + \varepsilon_\ell\rho_\ell h_\ell\vec{v}_\ell)) = \frac{d}{dt}(\Phi(\varepsilon_g P_g + \varepsilon_\ell P_\ell)) \\
&+ \nabla\cdot\Phi(\varepsilon_g\bar{\bar{\tau}}_{\mathbf{g}}\vec{v}_g + \varepsilon_\ell\bar{\bar{\tau}}_\ell\vec{v}_\ell) + \nabla\cdot\Phi(-\varepsilon_g k_g\nabla T_g - \varepsilon_\ell k_\ell\nabla T_\ell) + q_g^{w\prime\prime} + q_\ell^{w\prime\prime}
\end{aligned}
\tag{3.21}
$$

gives:

$$\begin{cases} \text{- Mass conservation:} \\ \dfrac{\partial \Phi \rho_m}{\partial t} + \nabla \cdot \Phi \rho_m \vec{u}_m = 0 \\ \\ \text{- Momentum conservation :} \\ \dfrac{\partial \Phi \rho_m \vec{u}_m}{\partial t} + \nabla \cdot \Phi \rho_m \vec{u}_m \otimes \vec{u}_m = -\Phi \nabla P_m + \Phi \rho_m \vec{g} + \nabla \cdot \Phi \bar{\bar{\tau}}_{\mathbf{m}} + \vec{\kappa} + \nabla \cdot \Phi \tau_{\mathbf{m}}^{\bar{\bar{\mathbf{Re}}}} + \vec{M}_g^w + \vec{M}_\ell^w \\ \\ \text{- Energy conservation :} \\ \dfrac{\partial}{\partial t}\left(\Phi \rho_m h_m\right) + \nabla \cdot \left(\Phi \rho_m h_m \vec{u}_m\right) = \dfrac{d}{dt}(\Phi P_m) + \nabla \cdot \Phi \bar{\bar{\tau}}_{\mathbf{m}} \vec{u_m} + \nabla \cdot \Phi(-k_m \nabla T_m) + q_m^{w\prime\prime} \end{cases} \tag{3.22}$$

with

$$\begin{array}{rcl} \rho_m & = & = (\varepsilon \rho_g + (1-\varepsilon)\rho_\ell) \\ \vec{u}_m & = & (\varepsilon \rho_g \vec{v}_g + (1-\varepsilon)\rho_\ell \vec{v}_\ell)/\rho_m \\ P_m & = & \varepsilon P_g + (1-\varepsilon) P_\ell \\ h_m & = & (\varepsilon \rho_g h_g + (1-\varepsilon)\rho_\ell h_\ell)/\rho_m = x h_g + (1-x) h_\ell \\ \nabla \cdot (\bar{\bar{\tau}}_{\mathbf{m}} u_m) & = & \nabla \cdot \Phi(\varepsilon_g \bar{\bar{\tau}}_{\mathbf{g}} \vec{v_g} + \varepsilon_\ell \bar{\bar{\tau}}_\ell \vec{v_\ell}) \\ \nabla \cdot \Phi(-k_m \nabla T_m) & = & (\varepsilon_g k_g \nabla T_g) + (\varepsilon_\ell k_\ell \nabla T_\ell) \end{array} \tag{3.23}$$

3.2 Drift flux model

The drift flux model (DFM) is a mathematical method commonly used to model two-phase flows. This model assumes that the mixture can be treated as a single fluid while considering phase slip, i.e., the relative velocity difference between the two constituent fluids.

The drift flux model relies on a set of conservation equations to describe the behaviour of the two-phase flow. These equations are coupled with additional relations to account for phase drift, such as void fraction (fraction of the volume occupied by the gas phase) and drift velocity (relative velocity of the gas phase with respect to the liquid phase).

Averaging for the drift flux model

To use the drift flux model, it is important to average the equation the same way we averaged section 3.1.1. As a reminder, we assume the following notation for quantities ϕ and ψ

$$\begin{aligned}
\langle\tilde{\psi}_k\rangle &= \frac{1}{V}\iiint_V \tilde{\psi}_k dV \\
\phi_k = \frac{1}{V_k}\iiint_{V_k} \tilde{\psi}_k dV &= \frac{1}{\varepsilon_k V}\iiint_V \chi_k\tilde{\psi}_k dV = \langle\chi_k\tilde{\phi}_k\rangle/\langle\chi_k\rangle \\
\varepsilon_k &= \langle\chi_k\rangle
\end{aligned} \tag{3.24}$$

When we measure or define a physical property in the mixture, it will, therefore, be weighted by the local void ratio χ_k of the phase in the flow. The equation 3.24 is a pretty good approximation if we consider equation 3.15, which suggests that the density is nearly constant in a control volume V_c

It should be noted that when we have two different physical properties, $\psi_k(\vec{x}), \phi_k(\vec{x})$ we know that the average of the product is not equal to the product of the averages.

$$\langle\tilde{\psi}_k(\vec{x})\tilde{\phi}_k(\vec{x})\rangle \neq \langle\tilde{\psi}_k(\vec{x})\rangle \cdot \langle\tilde{\phi}_k(\vec{x})\rangle \tag{3.25}$$

Total volumetric flux of the two-phase mixture

There is always a difference in speed between the two phases. Knowing that the vapour density is smaller, the velocity of the dispersed phase tends to be higher due to the conservation of mass.

We can add another variable to our modelisation: the total volumetric flux of the two-phase mixture. It is given by: $j = j_g + j_\ell$. Using the averaging by the void ratio, we can write:

$$\langle\tilde{j}\rangle = (1-\varepsilon)v_\ell + \varepsilon v_g \tag{3.26}$$

Drift velocity

Additionally, we can define $\tilde{V}_{gj} = \tilde{v}_g - j$ the difference of velocity of the gaseous phase relatively to the total apparent surface velocities: **the drift velocity.** It is the fundamental principle of the drift flux model. The velocity of a mixture corresponds to the velocity of the centre of mass in a given arbitrary volume.

The void weighted average of the local drift is given by:

$$V_{gj} = \frac{\frac{1}{A}\int_A \chi_k\tilde{V}_{gj}\ dA}{\frac{1}{A}\int_A \chi_k\ dA} = \frac{\langle\chi_k\tilde{V}_{gj}\rangle}{\langle\chi_k\rangle} \tag{3.27}$$

using the definition of the drift velocity, we can write:

$$V_{gj} = \frac{\frac{1}{A}\int_A \chi_k \tilde{v}_g \, dA - \frac{1}{A}\int_A \chi_k \tilde{j} \, dA}{\frac{1}{A}\int_A \chi_k \, dA} = v_g - \frac{\langle \chi_k \tilde{j} \rangle}{\langle \chi_k \rangle} \tag{3.28}$$

Using this relationship between V_{gj} and v_g we can express the void fraction weighted mean vapour velocity as:

$$v_g = V_{gj} + \frac{\langle \chi_k \tilde{j} \rangle}{\langle \chi_k \rangle} \frac{\langle \tilde{j} \rangle}{\langle \tilde{j} \rangle} = V_{gj} + C_0 \langle \tilde{j} \rangle \tag{3.29}$$

using the definition of C_0 that we can identify in Eq. 3.29 by multipling $\frac{\langle \chi_k \tilde{j} \rangle}{\langle \chi_k \rangle}$ by $\frac{j}{j}$:

$$C_0 = \frac{\langle \chi_k \tilde{j} \rangle}{\langle \chi_k \rangle \langle \tilde{j} \rangle} = \frac{\langle \chi_k \tilde{j} \rangle}{\varepsilon \langle \tilde{j} \rangle} \tag{3.30}$$

C_0 the "distribution parameter" quantifies the impact of radial void and volumetric flux distribution on vapour velocity. The average drift velocity, represented by V_{gj} serves as a measure of local slip and is closely tied to the terminal rise velocity of vapour bubbles through a stationary liquid. C_0 can be found experimentally. We can use probes to determine alpha and measure velocities to verify the validity of this relationship. Experimentally $C_0 \neq 1$ because the y-intercept is not zero.

Centre of mass velocity

If we consider a given volume whose centre of mass moves, its velocity corresponds to the velocity of propagation of a plane through which no net mass flux passes. The quantity of liquid entering is given by $\rho_\ell(1-\varepsilon)v_\ell$ and the quantity of gas leaving is given by $\rho_g \varepsilon v_g$. With a 1D mass balance considering collinear velocities (no vector equation), we rewrite the equation 3.23 and notice that it is equal to the mixture velocity:

$$\frac{\rho_\ell(1-\varepsilon)v_\ell + \rho_g \varepsilon v_g}{\rho_\ell(1-\varepsilon) + \rho_g \varepsilon} = u_m \tag{3.31}$$

and

$$\rho_m u_m = \rho_\ell(1-\varepsilon)v_\ell + \rho_g \varepsilon v_g \tag{3.32}$$

Center of volume velocity

To determine the velocity of the center of the volume we consider a control volume from where gas and liquid enter and leave. We can make a volume balance, the mass bilan is gives in fact the definition of the surperficial velocity:

$$(1-\varepsilon)v_\ell + \varepsilon v_g = \langle j_g\rangle + \langle j_\ell\rangle = \langle\tilde{j}\rangle \tag{3.33}$$

Therefore, we can assume that the total volumetric flux of the two-phase mixture corresponds to its center-of-volume velocity. Thanks to this relation, if we know the center-of-volume velocity, we can find the superficial velocities.

Relationship Between Phase Velocities and Center of Mass and Center of Volume Velocities

Averaging procedures are the same as the section 3.1.1. By multiplying the center of volume velocity 3.31 by ρ_ℓ/ρ_m and subtracting by the center of mass velocity 3.33:

$$u_m - \frac{\rho_\ell}{\rho_m}\langle\tilde{j}\rangle = \frac{\rho_\ell(1-\varepsilon)v_\ell + \rho_g\varepsilon\rho_g}{\rho_m} - \frac{\rho_\ell}{\rho_m}\cdot(1-\varepsilon)v_\ell + \varepsilon v_g = \frac{\rho_g v_g\varepsilon}{\rho_m} - \frac{\rho_\ell v_g\varepsilon}{\rho_m} \tag{3.34}$$

With the definition of the mixture density, $\rho_m = \rho_\ell(1-\varepsilon) + \rho_g\varepsilon$ we can write $\rho_g\varepsilon = \rho_m - \rho_\ell(1-\varepsilon)$. Substituting $\rho_g\varepsilon$ yields to:

$$v_g = u_m + \frac{\rho_\ell}{\rho_m}(v_g - \langle\tilde{j}\rangle) \tag{3.35}$$

This leads to the key velocity definitions:

$$v_g = u_m + \frac{\rho_\ell}{\rho_m}\cdot(V_{gj} + (c_0-1)\langle\tilde{j}\rangle) \tag{3.36}$$

and

$$v_\ell = u_m + (\frac{\varepsilon}{1-\varepsilon})\cdot\frac{\rho_\ell}{\rho_m}\cdot(V_{gj} + (c_0-1)\langle\tilde{j}\rangle) \tag{3.37}$$

using the following abbreviation:

$$V'_{gj} = V_{gj} + (c_0-1)\langle\tilde{j}\rangle \tag{3.38}$$

The phase velocity 3.36 and 3.37 become:

$$v_g = u_m + \frac{\rho_\ell}{\rho_m} \cdot V'_{gj} \tag{3.39}$$

and

$$v_\ell = u_m - (\frac{\varepsilon}{1-\varepsilon}) \cdot \frac{\rho_g}{\rho_m} \cdot V'_{gj} \tag{3.40}$$

Relationship between center of mass and center of volume velocity

Multiplying equation 3.40 by $(1 - \varepsilon)$ and 3.39 by ε we can write:

$$\langle \tilde{j} \rangle = u_m + \varepsilon \frac{\rho_\ell - \rho_g}{\rho_m} V'_{gj} \tag{3.41}$$

We can now rewrite the mass, momentum and energy equations 3.21 of both phases in the case of the drift flux model. Assuming that $V = Adz$, rewriting the equation assuming that the cross-section is constant along the channel for each phase and assuming a 1D set of equations, summing for each phase:

Mass conservation equation

$$\frac{\partial}{\partial t} A \left[\varepsilon \rho_g + (1-\varepsilon)\rho_\ell \right] + \frac{\partial}{\partial z} A \left[\varepsilon \rho_g v_g + (1-\varepsilon)\rho_\ell v_\ell \right] = 0 \tag{3.42}$$

using the definition of mixture density 3.23 and the center of mass velocity 3.31 we have

$$\frac{\partial}{\partial t} A\rho_m + \frac{\partial}{\partial z} A\rho_m u_m = 0 \tag{3.43}$$

Momentum conservation equation

Using equations 3.39 and 3.40 of the phase velocity using the drift flux model, summing the phase momentum conservation equation, taking into account the momentum interface condition, assuming that the pressure is constant across the flow section, the mixture momentum conservation equation is obtained as:

$$\begin{split}\frac{\partial}{\partial t}(\Phi A(\varepsilon_g \rho_g v_g + \varepsilon_\ell \rho_\ell v_\ell)) + \nabla \cdot (\Phi A(\varepsilon_g \rho_g v_g^2 + \varepsilon_\ell \rho_\ell v_\ell^2)) = -\nabla(\Phi A((\varepsilon_g P_g + \varepsilon_\ell P_\ell)) + \kappa \\ +\Phi A((\varepsilon_g \rho_g + \varepsilon_\ell \rho_\ell) g + \nabla \cdot (\varepsilon_g(\bar{\bar{\tau}}_{\mathbf{g}} + \tau_{\mathbf{g}}^{\bar{\bar{\mathbf{Re}}}})) + \nabla \cdot (\varepsilon_\ell(\bar{\bar{\tau}}_\ell + \tau_\ell^{\bar{\bar{\mathbf{Re}}}})) + M_g^w + M_\ell^w\end{split} \tag{3.44}$$

using the definition of the center of mass 3.31 we can rewrite the first term:

$$\frac{\partial}{\partial t}\Phi A[\varepsilon\rho_g\, v_g\, + (1-\varepsilon)\rho_\ell\, v_\ell\,] = \frac{\partial\Phi A\rho_m u_m}{\partial t} \tag{3.45}$$

Using the equation 3.39 and 3.40 of the drift flux model respectively v_g and v_ℓ, the second term becomes:

$$\begin{aligned}\tfrac{\partial}{\partial z}\Phi A\left[\varepsilon\rho_g\, v_g^2\, + (1-\varepsilon)\rho_\ell\, v_\ell^2\right] &= \frac{\partial}{\partial z}\Phi A\left[\varepsilon\rho_g + (1-\varepsilon)\rho_\ell\right]u_m^2 + \frac{\partial}{\partial z}\Phi A\frac{\varepsilon}{\rho_m^2}\rho_g\rho_\ell V_{gj}^{\prime 2}(\rho_\ell + \frac{\varepsilon}{1-\varepsilon}\rho_g) \\ &= \frac{\partial}{\partial z}\Phi A\left[\rho_m u_m^2 + \frac{\varepsilon}{1-\varepsilon}\frac{\rho_g\rho_\ell}{\rho_m}V_{gj}^{\prime 2}\right]\end{aligned} \tag{3.46}$$

The term $\nabla\cdot(\varepsilon_g\tau_{\mathbf{g}}^{\bar{\bar{\mathbf{Re}}}} + \varepsilon_\ell\tau_\ell^{\bar{\bar{\mathbf{Re}}}})$ is already modelled in the drift flux. Sources showing that this is already modelled are shown in appendix E. In fact, the drift flux model represents this turbulence effect in 1D flow as :

$$\nabla\cdot(\varepsilon_g\tau_{\mathbf{g}}^{\bar{\bar{\mathbf{Re}}}} + \varepsilon_\ell\tau_\ell^{\bar{\bar{\mathbf{Re}}}}) = -\frac{\partial}{\partial z}\left[\frac{\varepsilon_g}{\varepsilon_\ell}\frac{\rho_g\rho_\ell}{\rho_m}V_{gj}^{\prime 2}\right] \tag{3.47}$$

and the viscous terms $\nabla\cdot(\varepsilon_g\bar{\bar{\tau}}_{\mathbf{g}} + \varepsilon_\ell\bar{\bar{\tau}}_\ell)$ in 1D flow :

$$\nabla\cdot(\varepsilon_g\bar{\bar{\tau}}_{\mathbf{g}} + \varepsilon_\ell\bar{\bar{\tau}}_\ell) = \varepsilon_g\mu_g\frac{\partial^2\tilde{v}_g}{\partial z^2} + \varepsilon_\ell\mu_\ell\frac{\partial^2\tilde{v}_\ell}{\partial z^2} \approx 0 \tag{3.48}$$

$\vec{\kappa}$ the surface tension effect. In the case of the 1D pipe in the reactor, we can neglect it. The demonstration to neglect it is demonstrated in appendix D

The fourth term:

$$\Phi A[\varepsilon\rho_g + (1-\varepsilon)\rho_\ell]g = \Phi A\rho_m g \tag{3.49}$$

The wall-fluid friction terms can be written using a wall friction factor f and a two-phase multiplier Φ_m:

$$M_g^w + M_\ell^w = -u_m^2\rho_m\frac{f\phi_m}{D_h}\Delta V \tag{3.50}$$

For a vertical tube of constant flow section and constant phase densities in the radial direction, whitout prosity changes, the momentum equation takes the following form:

$$\frac{\partial}{\partial t}A\rho_m u_m + \frac{\partial}{\partial z}\left[A\rho_m u_m^2 + A\frac{\varepsilon}{1-\varepsilon}\,\frac{\rho_g\,\rho_\ell}{\rho_m}V_{gj}^{\prime 2}\right] + \frac{\partial}{\partial z}\,AP_m + A\rho_m g = -u_m^2\rho_m\frac{f\phi_m}{D_h}\Delta V \tag{3.51}$$

In practice, we can notice that

Energy conservation equation

By summing up the phase energy equation, taking into account interface energy condition and assuming that:

1. the axial gradient of the viscous work is negligible
2. axial heat conduction is negligible
3. there are no internal heat sources
4. kinetic and potential energies are negligible compared to internal energy
5. pressure is constant across the flow section

The energy equation can be written as:

$$\begin{aligned}\frac{\partial}{\partial t}(\Phi A(\varepsilon_g\rho_g h_g + \varepsilon_\ell\rho_\ell h_\ell)) + \nabla\cdot(\Phi A(\varepsilon_g\rho_g h_g\vec{v}_g + \varepsilon_\ell\rho_\ell h_\ell\vec{v}_\ell)) = \frac{d}{dt}(\Phi A(\varepsilon_g P_g + \varepsilon_\ell P_\ell))\\ +\nabla\cdot\Phi A(\varepsilon_g\bar{\bar{\tau}}_{\mathbf{g}}\vec{v_g} + \varepsilon_\ell\bar{\bar{\tau}}_\ell\vec{v_\ell}) + \nabla\cdot\Phi A(-\varepsilon_g k_g\nabla T_g - \varepsilon_\ell k_\ell\nabla T_\ell) + q_g^{w\prime\prime} + q_\ell^{w\prime\prime}\end{aligned} \tag{3.52}$$

The first term of the energy equation can be written using the definition of the mixture enthalpy:

$$\frac{\partial}{\partial t}\Phi A\left[\varepsilon\rho_g\,h_g\, + (1-\varepsilon)\rho_\ell\,h_\ell\right] = \frac{\partial\Phi A\rho_m h_m}{\partial t} \tag{3.53}$$

With the equations 3.39 and 3.40 the second term becomes:

$$\begin{aligned}\frac{\partial}{\partial z}\Phi A\left[\varepsilon\rho_g v_g h_g + (1-\varepsilon)\rho_\ell v_\ell h_\ell\right] &= \frac{\partial}{\partial z}\Phi A\left[\varepsilon\rho_g(u_m + \frac{\rho_\ell}{\rho_m}V_{gj}^{\prime})h_m + (1-\varepsilon)\rho_\ell(u_m - \frac{\varepsilon}{1-\varepsilon}\frac{\rho_g}{\rho_m}V_{gj}^{\prime})h_\ell\right]\\ &= \frac{\partial}{\partial z}\Phi A\rho_m u_m h_m + \frac{\partial}{\partial z}\Phi A\left[\varepsilon\frac{\rho_g\rho_\ell}{\rho_m}V_{gj}^{\prime}(h_g - h_\ell)\right]\end{aligned} \tag{3.54}$$

The next term is the wall heat transfer to the mixture and can be written as:

$$\nabla \cdot \Phi A(\varepsilon_g \bar{\bar{\tau}}_{\mathbf{g}} \vec{v_g} + \varepsilon_\ell \bar{\bar{\tau}}_\ell \vec{v_\ell}) + \nabla \cdot \Phi A(-\varepsilon_g k_g \nabla T_g - \varepsilon_\ell k_\ell \nabla T_\ell) + q_g^{w\prime\prime} + q_\ell^{w\prime\prime} = A_w \vec{n}_w \vec{q_w}^{\prime\prime} \tag{3.55}$$

Using the definition of the material derivative, the pressure term becomes:

$$\frac{d}{dt}\Phi A(\varepsilon_g P_g + \varepsilon_\ell P_\ell) = \frac{\partial \Phi A P_m}{\partial t} + j \cdot \nabla \Phi A P_m = \frac{\partial \Phi A P_m}{\partial t} + \left[\varepsilon \frac{\rho_\ell - \rho_g}{\rho_m} V'_{gj}\right] \cdot \Phi A P_m \tag{3.56}$$

Using the definition of enthalpy, knowing that the tube flow section is independent of time and assuming that specific masses of vapour and liquid are constant across the flow section we can write the simplified version of the energy conservation equation in the case of the 1D drift flux model:

$$\begin{aligned}\frac{\partial}{\partial t}\Phi A \rho_m \, h_s - \frac{\partial}{\partial t}\, \Phi A P_m + \frac{\partial}{\partial z}\Phi A \rho_m u_m h_s + \frac{\partial}{\partial z}\Phi A \varepsilon \frac{\rho_v \rho_\ell}{\rho_m} V'_{gj}(h_v - h_\ell) \\ + (u_m + \frac{\varepsilon(\rho_\ell - \rho_g)}{\rho_m} V'_{gj}) \cdot \frac{\partial}{\partial z}\Phi A P_m = \Phi A_w \, \vec{n}_w \cdot \vec{q}''_w.\end{aligned} \tag{3.57}$$

To learn more about the conservation equation using the drift flux model, as well as the assumptions used in the development of those equations, we refer interested readers to the book by Ishii and Hibiki [28].

CHAPTER 4 Developement of the new thermohydraulic solver

The THM: module is a simplified thermal-hydraulic solver within the Version5 environment that models reactors as independent parallel channels without inter-channel flow. 1D convection equations along its length characterize each channel for the fluid modelisation and 1D cylindrical equations for every Fourier conduction equation in each pin cell. It employs a two-fluid homogeneous model and is integrated with FreeSteam, an open-source implementation of IAPWS-IF97 steam tables for light water.

The simplified model utilized in THM: is built upon the contributions of M. F. Fehri for CANDU clusters [29] and P. Gallet [30] for PWR assemblies. This module operates in steady-state conditions, incorporating a subcooled flow boiling model based on either the Bowring correlation [31] or the Saha-Zuber correlation [32] for determining temperature subcooling at the onset of fully developed boiling (OFDB). Within each channel, the 1D thermal-hydraulics equations are solved, using two prescribed inlet conditions for coolant velocity and temperature, along with a fixed outlet condition for pressure.

The primary objective of the module modification is to be able to obtain all the thermal-hydraulic parameters that influence neutronics calculations. The fields of interest are, therefore, the void fraction, the coolant density, the fuel temperature and the water temperature, both inside and on the surface of the cladding. However, it will also be possible to obtain velocity, pressure and enthalpy fields along the z-axis after modification. The thermomechanics for neutronics is calculated by determining the radial temperature distribution from the centre of the fuel to the coolant as a function of the different physical properties of the material and the coolant at a specific z-level.

The model employs these key simplifying assumptions:

- For the conduction in the solid parts, heat transfer in axial and angular directions is negligible

- The fluid will only flow in direction z, we consider a 1D problem for the convection in the coolant.

- Heat transfer by coolant conduction is negligible.

4.1 Initial mathematical and numerical modelization in `THM`:

The initial modelling has been written in this documentation by Alain Hébert [7]. The conduction problem already developed will not be changed but the numerical modelisation of the convection will be updated.

4.1.1 Thermal conduction equation in the fuel, the gap and the clad

We are considering a radial symmetry in the fuel rod, using a constant surface discretization of the fuel rod $A(r) = \frac{r^2}{2}$. We define I equal-size volumes in the fuel, a gap, and I_c equal-size volumes in the clad. Fuel rod interfaces are located at $A_{i+1/2}$ with $1 \leq i \leq I$ and$A_{1/2} = 0$. The conduction discretisation equation is made using mesh centered finite difference as shown 4.1.

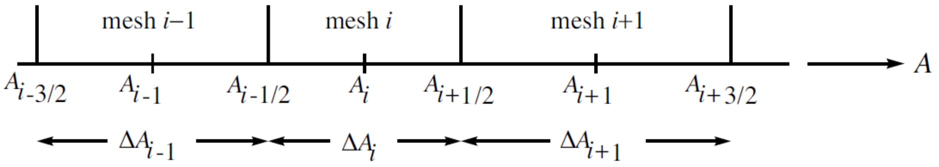


Figure 4.1 Mesh-Centered Finite Difference

The mesh size $\Delta A_i = A_{i+1/2} - A_{i-1/2}$. We define I meshes in the fuel, one mesh for the gap, and I_c meshes in the cladding. At the center of the fuel, at $r = 0$, we have $A_{1/2} = 0$. Note that the heat flux density is continuous at each $A_{i-1/2}$.

The temperature distribution in the fuel rod is given by Fourier's law of heat conduction:

$$\nabla \cdot [k(T)\nabla T(r,t)] + Q(r,t) = \frac{\partial}{\partial t}\left[\rho(T)C(T)\,T(r,t)\right] \tag{4.1}$$

As said before this equation is discretized using the MCFD. The $I + 1$ position corresponds to the gap center and $I + I_c + 3/2$ is at the outer surface of the cladding.

$$\begin{aligned} 4\left[A_{i+1/2}\,\frac{T_{i+1}(t) - T_i(t)}{\Delta A_i} - A_{i-1/2}\,\frac{T_i(t) - T_{i-1}(t)}{\frac{\Delta A_{i-1}}{k_i(t)} + \frac{\Delta A_i}{k_i(t)}}\right] & \\ + \Delta A_i Q_i(t) \;=\; \Delta A_i \frac{\partial}{\partial t}\left[\rho_i(t)C_i(t)\,T_i(t)\right] & \end{aligned} \tag{4.2}$$

where

$T(r), t) =$ temperature distribution in the fuel rod, including the clad (K)

$Q(r), t) =$ fission power distribution in fuel (W/m^3)

$k(T) =$ thermal conductivity of the fuel and clad (W/m/K)

$\rho(T) =$ density of the fuel and clad (kg/m^3)

$C(T) =$ specific heat capacity of the fuel and clad (J/K/kg).

which leads to the following tridiagonal system:

$$\begin{aligned} \Delta A_i \frac{\partial}{\partial t}\left[\rho_i(t) C_i(t)\, T_i(t)\right] - D_{i-1/2}(t)\, T_{i-1}(t) \\ + \Big[D_{i-1/2}(t) + D_{i+1/2}(t)\Big]\, T_i(t) - D_{i+1/2}(t)\, T_{i+1}(t) = \Delta A_i Q_i(t). \end{aligned} \tag{4.3}$$

At the fuel surface temperature we can write:

$$\begin{aligned} \Delta A_I \frac{\partial}{\partial t}\left[\rho_I(t) C_I(t)\, T_I(t)\right] - D_{I-1/2}(t)\, T_{I-1}(t) \\ + \Big[D_{I-1/2}(t) + E_{I+1/2}(t)\Big]\, T_I(t) - E_{I+1/2}(t)\, T_{I+1/2}(t) = \Delta A_I \mathcal{Q}_I(t). \end{aligned} \tag{4.4}$$

The equation at the interface gap/clad is the same, using a symmetric transformation. Using the H_{gap} correlation and the continuity of the heat flux we can write the equation in the gap:

$$\begin{aligned} -E_{I+1/2}(t)\, T_I(t) + \Big[E_{I+1/2}(t) + G_{I+1}(t)\Big]\, T_{I+1/2}(t) - G_{I+1}(t)\, T_{I+3/2}(t) = 0 \\ G_{I+1}(t)\, T_{I+1/2}(t) + \Big[G_{I+1}(t) + F_{I+3/2}(t)\Big]\, T_{I+3/2}(t) - F_{I+3/2}(t)\, T_{I+2}(t) = 0. \end{aligned} \tag{4.5}$$

and the tridiagonal relation for clad surface temperature:

$$\begin{aligned} \Delta A_{I+I_c+1} \frac{\partial}{\partial t}\Big[\rho_{I+I_{c+1}}(t) C_{I+I_{c+1}}(t)\, T_{I+I_{c+1}}(t)\Big] - D_{I+I_c+1}(t)\, T_{I+I_c}(t) \\ + \Big[D_{I+I_c+1/2}(t) + E_{I+I_c+3/2}(t)\Big]\, T_{I+I_c+1} = E_{I+I_c+3/2}(t)\, T_{\text{surf}}(t) = E_{I+I_c+1} Q_{I+I_e+1}(t). \end{aligned} \tag{4.6}$$

which leads to the following tridiagonal linear system of order $I + I_c + 2$ with a source term function of the external clad surface temperature T_{surf} for each z step:

$$\left[\frac{\partial}{\partial t}\,\mathrm{C}\,(t) + \mathrm{A}(t)\right] T(t) = S_1(t) + T_{\mathrm{surf}}(t) S_2(t) \tag{4.7}$$

To learn more about the details to find the tridiagonal linear system, as well as the assumptions used in the development of those equations, we refer interested readers to the notice by Alain Hébert [7].

4.1.2 Initial convection modelisation in the coolant

In the initial code of `THM:` we are assuming a single phase and forced convection flow regime. The mass flow equation in the coolant is based on the following balance relation:

$$\frac{\partial}{\partial t}\rho(z,t) + \frac{\partial}{\partial z}\dot{m}''(z,t) = 0 \tag{4.8}$$

ρ is given by the IAPWS-IF97 water tables as a function of temperature and pressure.

The pressure change along the channel is determined by the momentum conservation equation, which factors in the gradient of mass flow rate, gravitational influence, and coolant friction. This equation can be expressed as::

$$\frac{\partial}{\partial t}\dot{m}''(z,t) + \frac{\partial}{\partial z}\left[\frac{\dot{m}''^2(z,t)}{\rho(z,t)} + p(z,t)\right] + \left[\frac{f\,\dot{m}''^2(z,t)}{2\rho(z,t)\,D_H} + \rho(z,t)\,g\right] = 0 \tag{4.9}$$

where
f = friction parameter as given by the Müller-Steinhagen correlation.

The thermal convection equation in the coolant is based on a one-phase energy conservation relation in energy. In steady-state conditions, the mass flow rate per unit of area $\dot{m}''$ is constant along the channel and the `THM:` documentation is giving the increase in enthalpy over each axial mesh:

$$h_j = h_{j-1} + \frac{q''_{\mathrm{fuid}}}{\dot{m}''\,A_j}\,\Delta z_j = h_{j-1} + \frac{P_{\mathrm{h}}\varphi}{\dot{m}''\,A_j}\,\Delta z_j \tag{4.10}$$

where
P_h = heated perimeter (m)
φ = heat flux delivered to fluid (W/m2)
$\dot{m}''$ = mass flow rate per unit of area (kg/m2/s)
A_j = coolant cross section area (m2)

Δz_j = axial mesh width (m)

4.1.3 Limit of the thermohydraulic module

Conduction in the cladding, gap and fuel is not intended to be modified. In fact, the `THM:` module was developed to model presurized water reactors. Apart from the materials used, there are no significant differences in fuel rod conduction between these reactors and boiling water reactors. The main difference between the old version of `THM:` and the new one `THM`$_p$ is the addition of the two-phase flow and the conservation equations for mass and momentum. This makes it possible to obtain the pressure and velocity distribution along the channel.

Some tests were carried out to determine the limitations of the single-phase version of `THM:`. We performed various tests on a single fuel rod case, varying fission power, inlet temperature, inlet flow velocity and pressure along the channel.

4.2 New modelisation of the convection

4.2.1 Mathematical modelisation

The mathematical modelization is based on the two-fluid model developed by Ishii and Hibiki [28] in chapter 14 and explained in section 3.2. Noting that $A = \Phi A_0 = \Phi A(z = 0)$, we can rewrite the drift flux model equation 3.43, 3.51 and 3.57:

The mixture mass conservation equation is written:

$$\frac{\partial}{\partial t} A\rho_m + \frac{\partial}{\partial z} A\rho_m u_m = 0 \tag{4.11}$$

The mixture momentum conservation equation:

$$\frac{\partial}{\partial t} A\rho_m u_m + \frac{\partial}{\partial z}\left[A\rho_m u_m^2 + A\frac{\varepsilon}{1-\varepsilon}\frac{\rho_g\,\rho_\ell}{\rho_m}V_{gj}'^2\right] + \frac{\partial}{\partial z}\,Ap + A\rho_m g = -u_m^2\rho_m\frac{f\phi_m}{D_h} \tag{4.12}$$

The mixture enthalpy conservation equation:

$$\begin{aligned}\frac{\partial}{\partial t} A\rho_m\,h_s - \frac{\partial}{\partial t}\,p + \frac{\partial}{\partial z} A\rho_m u_m h_s + \frac{\partial}{\partial z} A\varepsilon\frac{\rho_v\rho_\ell}{\rho_m}V_{gj}'(h_v - h_\ell)&\\ +(u_m + \frac{\varepsilon(\rho_\ell-\rho_g)}{\rho_m}V_{gj}')\cdot\frac{\partial}{\partial z}Ap = A_w\,\vec{n}_w\cdot\vec{q}_w''.&\end{aligned} \tag{4.13}$$

Solving a drift flux model always begins with the three fundamental equations of mass, momentum, and enthalpy written before. These equations describe the behaviour of the two-phase mixture: the continuous phase (usually the liquid phase) and the dispersed phase (usually the vapour phase). However, these three equations taken alone are not sufficient to close the problem. This is because there are more unknowns than equations, leading to an underdetermined system.

The unknowns are: the velocity u_m, the pressure P_m, the enthalpy h_m, the density ρ_ℓ and ρ_v and ρ_m of the two phases and the mixture, the void fraction ε and the two drift velocities V_{gj} and $V_{gj}^{'}$. This leads to 9 unknowns but a lot of equations to describe this system.

To close the system and make it resolvable, additional equations are required. One common method to model the void fraction is to introduce a new conservation equation for the momentum of the dispersed phase. This new equation involves formulating a new momentum conservation bilan considering the interaction between the dispersed and the continuous phase. For example, it will be needed to model the drag force, the lift force, and interfacial momentum transfer. This approach can provide interesting insight into the phase dynamics, but it increases a lot the complexity of the model and the computational effort needed.

An alternative method, which is often more practical, is to use a correlation for the void fraction. We can use an empirical or semi-empirical void fraction correlation. The correlation is often related to measurable quantities such as mass flux, flow velocity, and fluid properties. Using a void fraction correlation simplifies the model and makes it more powerful for analytical and numerical solutions.

In addition to the three conservation equations and the void fraction correlation, we can add the equation 3.23 of the density of the mixture, the equation 3.38 from the drift flux model for the two drift velocity and a correlation discussed in the section 4.2.2. Knowing that the liquid phase is water and the dispersed phase is vapour, their densities can be found using the IAPWS95 vapour tables.

4.2.2 Constitutive equations

Several constitutive relations are needed to specify void fraction, drift velocity, density, friction losses, heat transfer coefficients, and equations of state that specify the steam/water thermodynamic properties.

Void fraction

Void fraction is determined using the standard drift flux formulation based on the flow quality. This equation implied empirical parameters mentioned before in the section 3.2 such as the distribution parameters C_0 and the drift velocity V_{gj}. These two parameters depend on the chosen correlation to model two-phase flows [33].

$$\varepsilon = \frac{x}{C_0\left(x + \frac{\rho_v}{\rho_\ell}(1-x)\right) + \frac{\rho_v V_{gj}}{\rho_m u}} \tag{4.14}$$

These two parameters depend on the characteristics of the flow:

$$\begin{aligned} C_0 &= f(\varepsilon, P, h, D_h, u) \\ V_{gj} &= f(\varepsilon, P, h, D_h, u) \end{aligned} \tag{4.15}$$

To model C_0 and V_{gj}, a variety of void fraction models exist and are described in appendix A:

- Electric Power Research Insititute (EPRI) void model
- GE-Ramp
- Modified Bestion

Flow quality

To model the flow quality used in the equation 4.14 we use a simple formulation:

$$\begin{cases} x &= 0 & \text{if } h_m < h_{l,sat} \\ x &= \dfrac{h_m - h_{l,sat}}{h_{g,sat} - h_{l,sat}} & \text{if } h_{g,sat} > h_m > h_{l,sat} \\ x &= 1 & \text{if } h_m > h_{g,sat} \end{cases} \tag{4.16}$$

To model the flow quality more efficiently, the EPRI subcooled model is often used in nuclear thermohydraulic simulation such as in the code [1]. The model allows the numerical modelling to consider the flow regime to predict how the vapour phase begins to form and distribute within the subcooled liquid.

In the subcooled region, the quality is negative or zero, showing no vapour phase. When the flow heats up, the model predicts the beginning of the boiling and adjusts the thermohydraulic properties, such as the void fraction, depending on the specific flow regime. This correlation can be implemented for future work.

4.2.3 Friction and two-phase factor correlations

To correctly model the two-phase flow pressure drop, we must carefully select the correlation that efficiently captures the influence of interphase and wall interactions. Precise parameters such as the friction factor and the two-phase multiplier coefficient must be used to estimate the pressure drop along the channel accurately. Here, we review the correlation employed in our model.

Friction factor

We use the **Churchill correlation** [34], which is a semi-empirical formula developed to determine the friction factor f in pipes with various levels of roughness. This correlation is applicable across all flow regimes. It is expressed as follows:

$$f = 8\left(\left(\frac{8.0}{\mathrm{Re}}\right)^{12} + \left(\left(2.475\ln\left(\left(\frac{7}{\mathrm{Re}}\right)^{0.9} + 0.27R\right)\right)^{16} + \left(\frac{37530}{\mathrm{Re}}\right)^{16}\right)^{-1.5}\right)^{\frac{1}{12}}$$

with R, the adimentional roughness.

We can also use the **blasius correlation** [35] which is used specifically for turbulent flows in smooth pipes. This simple formula can be used when the Reynold number is in a moderate range between 3000 and 10^5 :

$$f = 0.3164 \cdot \mathrm{Re}^{-0.25} \tag{4.17}$$

Two-phase multiplier

The two-phase multiplier represents the impact of both gas and liquid phases on the pressure drop across a flow system. Various models have been developed to account for the complex behaviour of two-phase systems, and we employ different multipliers to capture the variations in pressure drop based on specific flow conditions.

Homogeneous Equilibrium Model (HEM) variants 1 & 2 [33] provide a simplified estimation of pressure losses by assuming no significant frictional or slip effects between phases. It implies that the two velocities of the phases are the same, that The two velocities are uniform within the area, and finally, the two phases are in thermodynamic equilibrium. HEM2 is better than HEM1 because it incorporates minor slip effects:

- **HEM1:**
$$\phi_m^2 = \left(\frac{v_\ell}{v_g}\right) x_{\text{th}} + 1$$

- **HEM2:**
$$\phi_m^2 = \left[\left(\frac{v_\ell}{v_g}\right) x_{\text{th}} + 1\right]\left[1 + x_{th}\left(\frac{\mu_g}{\mu_\ell} - 1\right)\right]$$

The **Lockhart-Martinelli** [36] correlation is a classic method for determining the two-phase multiplier, particularly in low-pressure and gas-liquid flows. The correlation is based on the Martinelli parameter X, representing the ratio of pressure drops for hypothetical single-phase flows of the liquid and gas components. Considering the same friction factor for both phases, we can write the Lockhart-Martinelli parameter:

$$X^2 = \left(\frac{\mu_f}{\mu_g}\right)^{0.25}\left(\frac{1 - x_{th}}{x_{th}}\right)^{1.75}\left(\frac{\rho_g}{\rho_\ell}\right) \tag{4.18}$$

And then the two-phase multiplier is:

$$\phi_m^2 = 1 + \frac{C}{X} + \frac{C}{X^2} \tag{4.19}$$

The Lockhart-Martinelli correlation covers a wide range of flows. Depending on the phase velocities and viscosities (through the calculation of the Reynold number), the C coefficient can take different values:

Flow Regime	C **coefficient**
Turbulent-Turbulent flow	20
Viscous-Turbulent flow	12
Turbulent-Viscous flow	10
Viscous-Viscous flow	5

Table 4.1 Coefficient values for different flow regimes

4.2.4 1D numerical discretisation

The rational approach to obtaining a one-dimensional drift-flux model is to integrate the three-dimensional drift-flux model over a cross-sectional area and then introduce proper mean values. For the rest of the master's thesis, we will study steady-state equations.

To determine the discretized equation, the finite volume technique is utilized, adopting a face-based approach for the equations. The face-based approach is motivated by the use of the Dirichlet condition at the inlet and at the outlet. So for N volume, we have $N+1$ faces and $N+1$ values of P, v, h to solve in our matricial resolution. By defining the control volume to encompass the entire cross-sectional area within the channel, the equations are simplified into a one-dimensional flow, incorporating heat fluxes and stress terms from the boundary. The domain is divided into control volumes of height Δz and area A. The center of each numerical control volume is located at a height $z_{i+\frac{1}{2}}$ and corresponds to the field value $\psi(z_{i+\frac{1}{2}}) = \psi_{i+\frac{1}{2}}$.

The Finite Volume discretization of partial differential equation consists of their integration over each numerical control volume V_n and the application of the Stokes theorem. Note that $V_c \neq V_n$. V_c is used to establish the averaged equation, and V_n is related to the numerical method. Consider the following general conservation equation:

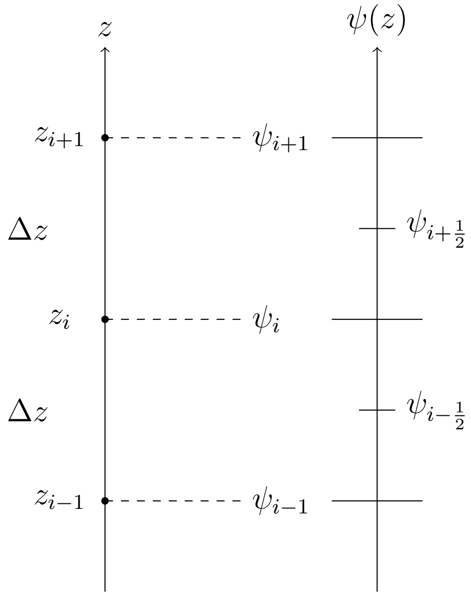
$$\frac{d}{dx}\left(F(\phi)\right) + S(\phi) = 0, \tag{4.20}$$

where $\psi(\phi)$ represents the flux associated with the quantity ϕ et $S(\phi)$ is the source term.

As the problem is one-dimensional along the z, we can write for the volume $i + \frac{1}{2}$, bounded by z_{i+1} and z_i:

$$\int_{V_n} \psi dV = \int_{z_i}^{z_{i+1}} \psi A dz = A \int_{z_i}^{z_{i+1}} \psi dz \tag{4.21}$$

The schema associated is:



In this type of scheme we solve the equations for the ψ_i variables. Gradients are computed using the values at the nodes but we still need the intermediate $\psi_{i+\frac{1}{2}}$ values for example in the momentum conservation equation. To determine this term we compute a sort of interpolation which is the mean values between the two at the nodes:

$$\psi_{i+\frac{1}{2}} = \frac{\psi_i + \psi_{i+1}}{2} \tag{4.22}$$

We integrate the equation (4.20) :

$$\int_{V_n} \frac{d}{dx}\left(F(\phi)\right)\, dx + \int_{V_n} S(\phi)\, dx = 0. \tag{4.23}$$

By applying an infinitesimal approximation, the first integral simplifies to:

$$\int_{V_n} \frac{d}{dx}\left(F(\phi)\right)\, dx = F_{i+1} - F_i, \tag{4.24}$$

and, assuming the source term is constant over the volume, the second integral becomes $S_{i+\frac{1}{2}}\,\Delta x$. Then, the discrete formulation at $z_{i+\frac{1}{2}}$ is expressed as

$$\psi_{i+1} - \psi_i + S_{i+\frac{1}{2}}\,\Delta x = 0. \tag{4.25}$$

We obtain the following equation using the equations from section 4 and discretizing them using the finite volume method. The asterisk represents the variable at the previous iteration for the state variables. To linearise the non-linear terms such as u^2 we use the fixed-point iteration method. In this approach to determine u^2, we use the values from the previous iteration (e.g., u_i^*) to linearize the term in the current iteration. This means that instead of solving directly for the nonlinear term, you use the approximation:

$$u_i^2 \approx u_i \cdot u_i^*$$

To de-couple the energy equation with the pressure-velocity system $u_m h_m$, we incorporate the result of the pressure-velocity resolution at the iteration k into the numerical resolution of the energy equation at the iteration k. To simplify the notation, the discretized mixture density and velocity will be noted ρ and u instead of ρ_m and u_m.

The mixture mass conservation equation is written:

$$\int_{V_n} \frac{\partial}{\partial z} \rho_m u_m = (\rho^* u A)_{i+1} - (\rho^* u A)_i = 0 \tag{4.26}$$

The mixture momentum equation:

$$\int_{V_n} \frac{\partial}{\partial z} \left[\rho_m u_m^2 + \frac{\varepsilon}{1-\varepsilon} \frac{\rho_\nu \, \rho_\ell}{\rho_m} V_{gj}'^2 \right] dV + \int_{V_n} \frac{\partial}{\partial z}\, p \; dV = -\int_{V_n} \rho_m g \; dV - \int_{V_n} u_m^2 \rho_m \frac{f \phi_m}{D_h} dV \tag{4.27}$$

became:

$$\begin{aligned} (\rho^* u^* A)_{i+1} u_{i+1} - (\rho^* u^* A)_i u_i \quad &+ \quad A_{i+1} P_{i+1} - A_i P_i \\ &= \quad \left(\frac{\varepsilon^*}{1-\varepsilon^*} \frac{\rho_\ell^* \rho_g^*}{\rho^*} (V_{gj}^*)^2 A \right)_i - \left(\frac{\varepsilon^*}{1-\varepsilon^*} \frac{\rho_\ell^* \rho_g^*}{\rho^*} (V_{gj}^*)'^2 A \right)_{i+1} \\ &- \quad \left(\frac{\rho_{i+1} \Delta z_{i+1} A_{i+1} + \rho_i A \Delta z_i}{2} \right) g \\ &- \quad \left(\frac{A \Delta z \rho u^* f \phi_m}{2 D_h} \right)_{i+1} u_{i+1} + \left(\frac{A \Delta z \rho u^* f \phi_m}{2 D_h} \right)_i u_i \end{aligned} \tag{4.28}$$

if we note:

$$\hat{A}_i^* = \left(A + \frac{\Phi_m^*}{2} \frac{f^*}{D_h} A \Delta z \right)_i \tag{4.29}$$

we can rewrite the equation 4.28:

$$
\begin{aligned}
(\rho^* u^* \hat{A}^*)_{i+1} u_{i+1} - (\rho^* u^* \hat{A}^*)_i u_i + A_{i+1} P_{i+1} - A_i P_i = &\left(\frac{\varepsilon^*}{1-\varepsilon^*} \frac{\rho_\ell^* \rho_g^*}{\rho^*} \left(V_{gj}'^* \right)^2 A \right)_i \\
&- \left(\frac{\varepsilon^*}{1-\varepsilon^*} \frac{\rho_\ell^* \rho_g^*}{\rho^*} \left(V_{gj}'^* \right)^2 A \right)_{i+1} - \left(\frac{\rho_{i+1} \Delta z_{i+1} A_{i+1} + \rho_i \Delta z_i A_i}{2} \right) g
\end{aligned}
\tag{4.30}
$$

The mixture enthalpy-energy equation:

$$
\begin{aligned}
(\rho^* u^* A)_{i+1} h_{i+1} - (\rho^* u^* A)_i h_i = q''' A \Delta z &+ \left(\varepsilon^* \frac{\rho_\ell^* \rho_g^*}{\rho^*} \Delta h_{fg} V_{gj}'^* A \right)_i - \left(\varepsilon^* \frac{\rho_\ell^* \rho_g^*}{\rho^*} \Delta h_{fg} V_{gj}'^* A \right)_{i+1} \\
&+ \frac{\left(u^* + \frac{\varepsilon^* (\rho_\ell^* - \rho_g^*)}{\rho^*} V_{gj}'^* \right)_i + \left(u^* + \frac{\varepsilon^* (\rho_\ell^* - \rho_g^*)}{\rho^*} V_{gj}'^* \right)_{i+1}}{2} \cdot \left(P_i^* A_i - P_{i+1}^* A_{i+1} \right)
\end{aligned}
\tag{4.31}
$$

with the use of equation 4.22

$$
(u_m + \frac{\varepsilon(\rho_\ell - \rho_g)}{\rho_m} V_{gj})_{i+\frac{1}{2}} = \frac{\left(u^* + \frac{\varepsilon^* (\rho_\ell^* - \rho_g^*)}{\rho^*} V_{gj}^* \right)_i + \left(u^* + \frac{\varepsilon^* (\rho_\ell^* - \rho_g^*)}{\rho^*} V_{gj}^* \right)_{i+1}}{2} \tag{4.32}
$$

At the outlet of each channel, the outlet pressure P_{out} is provided by the user. At the inlet of each channel, the inlet mass flow rate and temperature are provided by the user. For each iteration the inlet pressure calculated is used to compute the boundary condition h_{inlet} and v_{inlet}.

To go further, the discretization to consider the transient part is given in the appendix C. For the rest of this master thesis we will model only the steady state aspect.

4.3 Numerical simulation methodology

4.3.1 Methodology

Developing a Python solver from scratch for fluid dynamic problems involving complex phenomena like compressible flow, pressure drop, and two-phase flow is always an iterative process. The methodology is incrementally building up the solver's capability. I started from the simplest case and gradually added complexity.

Starting from an incompressible flow without pressure drop and without creating a void fraction, neglecting the frictional losses and elevation changes, allows us to validate the base structure algorithm and the initial numerical resolution algorithm.

The pressure drop was then implemented using the Darcy-Weisbach equation to calculate the pressure drop due to friction in the pipe. After a validation test case, the density change (using the IAPWS97 table) was implemented to introduce more realistic water and steam properties. At this stage, the fluid is treated as incompressible, allowing density to change. This logically leads to the addition of void fraction and drift flux model correlation.

Refactoring and classifying the code is a key point to improve structure and maintainability. This involves, for example, separating the numerical methods, physical models and water properties models. The code is made up 6 class. The classes are described below:

- **Version_5_THM_prototype()** is the main driver for setting up and solving thermal-hydraulic cases in a nuclear reactor. It solves the one-dimensional heat convection in the water using the class *DFMclass()*. It determined the convective heat transfer between the coolant (water) and the surface of the clad. This class also solve the radial heat conduction in the clad, gap and fuel rod using the finite difference method in another class. It also computes the temperature distribution in the centre of the fuel rod. This class combines conduction and convection procedures and allows the user to visualize and compare different models.

- **FDM_HeatConductionInFuelPin()** simulates radial heat conduction in a nuclear fuel pin using the finite difference method (FDM). The code models the temperature distribution across the fuel pin, considering the fuel, gap, and cladding regions based on the work done by A. Hebert [7].

- **DFMclass()** is a two-phase solver of convection in a heated tube. It sets up the physical and geometrical parameters and constructs the system of equations for solving steady states or transient problems using the FVM. Then, it uses the class *numercialResolution()* to solve the linear system. At each iteration, we solve a system, then use the class *waterProp()* to determine the flow properties and then calculate residuals.

- **numericalResolution()** class implements several numerical methods for solving linear systems of the form $Ax = b$. It supports different solvers and preconditioning techniques to improve convergence and it is designed to handle large systems.

- **FVM()** class is used by the *DFMclass()* class to fill in the system of equation.

- **waterProp()** contains the functions to create fields and give the values of the variables in the drift flux model resolution file. This class models the thermohydraulic behaviour of a fluid flow in a system divided into multiple cells. Each cell is characterised by properties such as velocity (U), pressure (P), enthalpy (H), void fraction, and various geometric parameters. For each cell, the void fraction and thermohydraulic properties can be computed depending on the correlation chosen. This class performs iterative updates to simulate two-phase flow dynamics.

- **plotting()** class is a Python class designed to handle the post-processing and visualisation of simulation results from multiple cases at one time of thermal-hydraulic analyses. It can plot comparisons, compute errors and plot cases depending on the attribute given by the user.

4.3.2 Algorithm

The algorithm is illustrated in figure 4.2. Initially, the values of all state variables are set. This step is crucial as it establishes the starting point for the following iterative process.

The boundary conditions in the linear system being solved are the outlet pressure, the inlet temperature, and the fluid velocity at the inlet (or alternatively, the mass flow rate). The outlet pressure is a key parameter that influences the pressure distribution throughout the system, while the inlet enthalpy directly affects the thermal state of the fluid as it enters the system.

Next, the set of mass and momentum conservation equations is solved simultaneously over all nodes for the velocity (u) and pressure (P) terms that do not include asterisks. By solving these equations, the model ensures a consistent and accurate calculation of these variables across the entire domain. In a second time, the energy conservation equation is solved using the new velocity (u) and pressure fields (P).

As the pressure at the inlet (P_{in}) is determined during each iteration and the flow rate is a constant of the problem, the density and the velocity at the inlet are adjusted at each step. This adjustment is critical as it ensures that the pressure drop across all channels remains consistent throughout each iteration. Maintaining this balance is essential for the stability and accuracy of the model.

With the updated values of velocity (u), pressure (P), and enthalpy (h), these are then used to calculate the terms with asterisks in the mass and momentum conservation equations. These terms require the updated state variables for accurate computation.

Finally, this iterative process is repeated until the error tolerance is reached. The tolerance error is calculated using the state variables. The repetition ensures that the model converges to an accurate and stable solution, providing reliable results for the modelled system. A time loop has been represented if a future work involves modelling the system's transient behaviour.

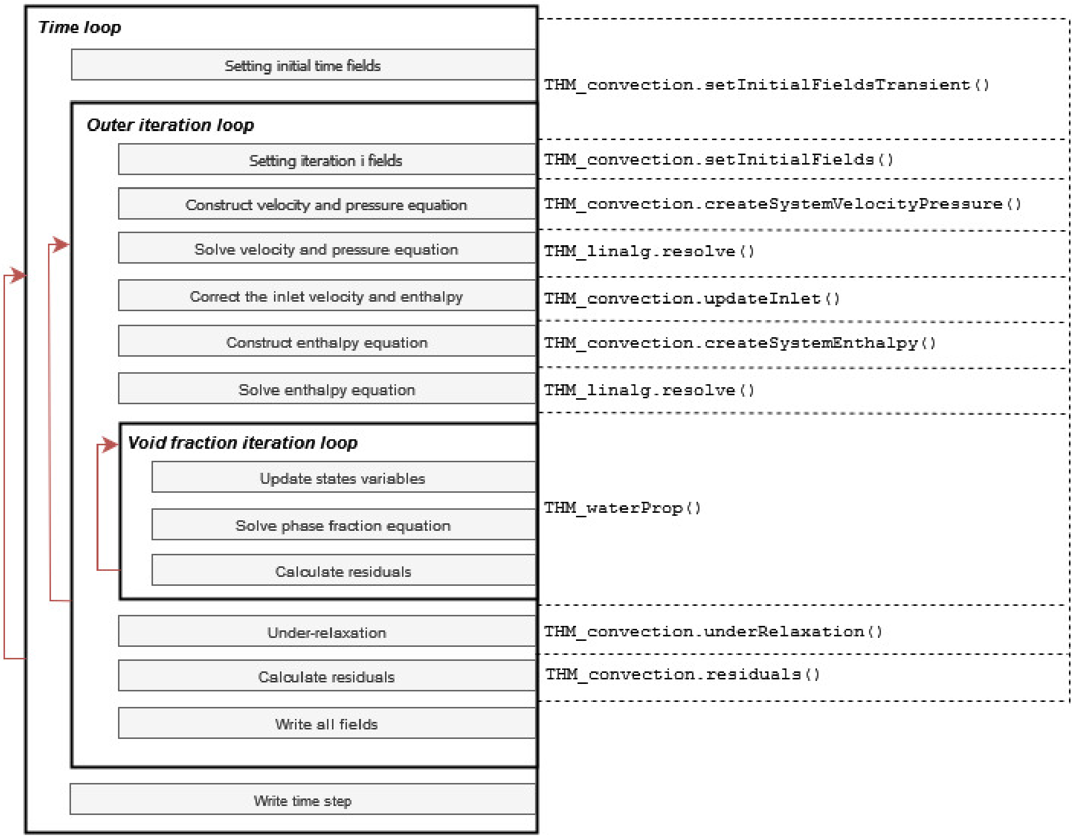

Time loop
Setting initial time fields
THM_convection.setInitialFieldsTransient()
Outer iteration loop
Setting iteration i fields
THM_convection.setInitialFields()
Construct velocity and pressure equation
THM_convection.createSystemVelocityPressure()
Solve velocity and pressure equation
THM_linalg.resolve()
Correct the inlet velocity and enthalpy
THM_convection.updateInlet()
Construct enthalpy equation
THM_convection.createSystemEnthalpy()
Solve enthalpy equation
THM_linalg.resolve()
Void fraction iteration loop
Update states variables
Solve phase fraction equation
Calculate residuals
THM_waterProp()
Under-relaxation
THM_convection.underRelaxation()
Calculate residuals
THM_convection.residuals()
Write all fields
Write time step

Figure 4.2 Algorithm flowchart and link with the oriented-object code

4.3.3 Numerical resolution

In the first approach, we used a simple resolution algorithm. The system of linear equations $\mathbf{Ax} = \mathbf{b}$ (x corresponding to the concatenation of u and P or for the set of h) was solved by inverting the matrix $\mathbf{A}$. This method involves calculating the inverse matrix $\mathbf{A}^{-1}$ and then obtaining the solution $\mathbf{x}$ by multiplying this inverse by the vector of independent terms $\mathbf{b}$:

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \tag{4.33}$$

Although matrix inversion is a direct and conceptually simple technique, it can be susceptible to numerical instability, particularly when the matrix $\mathbf{A}$ is poorly conditioned. In a preliminary study, it was observed that the algorithm required does not converge efficiently, and residuals oscillate for some complex problems.

To mitigate these issues an under-relaxation step was incorporated into the solution process. Under-relaxation is a technique used to improve the convergence of iterative methods, especially when the system is challenging to solve directly due to numerical instability. After obtaining an initial solution $\mathbf{x}^{(m-1)}$ through matrix inversion, under-relaxation is applied to gradually adjust this solution. This is done by updating the solution at each iteration m according to the following formula:

$$\mathbf{x}^{(m+1)} = \mathbf{x}^{(m)} + \omega \left(\mathbf{x}^{(m-1)} - \mathbf{x}^{(m)} \right)$$

where ω is the under-relaxation factor, chosen between 0 and 1 to control the convergence rate. This factor is crucial for stabilising the solution and avoiding oscillations or divergence, particularly in cases where the matrix $\mathbf{A}$ is poorly conditioned.

The under-relaxation step showed very good performance in terms of numerical stability but was pretty slow. Three other numerical methods were implemented and can be chosen depending on the problem. The Gauss-Siedel method was implemented, but the convergence of this method is only guaranteed if the $\mathbf{A}$ matrix is symmetric, positive-definite or diagonally dominant. The Conjugate gradient method as been then implemented. It is the conjugate version of the gradient method, an iterative algorithm that can be viewed as an optimisation technique.

CHAPTER 5 Presentation of the validation case and GeN-Foam

5.1 Presentation of the validation case

We will develop a solver to solve a specific case: the thermohydraulics of a vertical pincell. Nuclear fuel pin cell thermohydraulics presents coupled challenges in heat transfer and fluid dynamics that require robust numerical solutions. We develop $\texttt{THM}_p$, a modular solver redesigning `THM:`'s convection capabilities while preserving its established interface for backwards compatibility. It's important to remember that the solver needs to be developed with a modular approach so that it can be improved in the future. Moreover, the solver's new inputs should be as close as possible to those of the existing `THM:` solver. In the following, we will refer to `THM:` or $\texttt{THM:}_f$ as the old solver and $\texttt{THM:}_p$ or $\texttt{THM:}_{\text{prototype}}$ as the new solver.

As mentioned before, the case studied is a pincell composed of (Fig. 5.1):

- Fuel: UO_2 pellet (r_f) with volumetric heat generation
- Pellet-to-cladding gap filed by a gaz i.e Helium, generally modelled with a coefficient $H_{gap} = 10000\ W/m^2K$
- Clad: Zircaloy-4 sheath ($r_{c,out}$) with gap conductance model
- Coolant: Light water flow ($\dot{m}$ in $[kg/s]$) in subcooled-to-boiling regimes

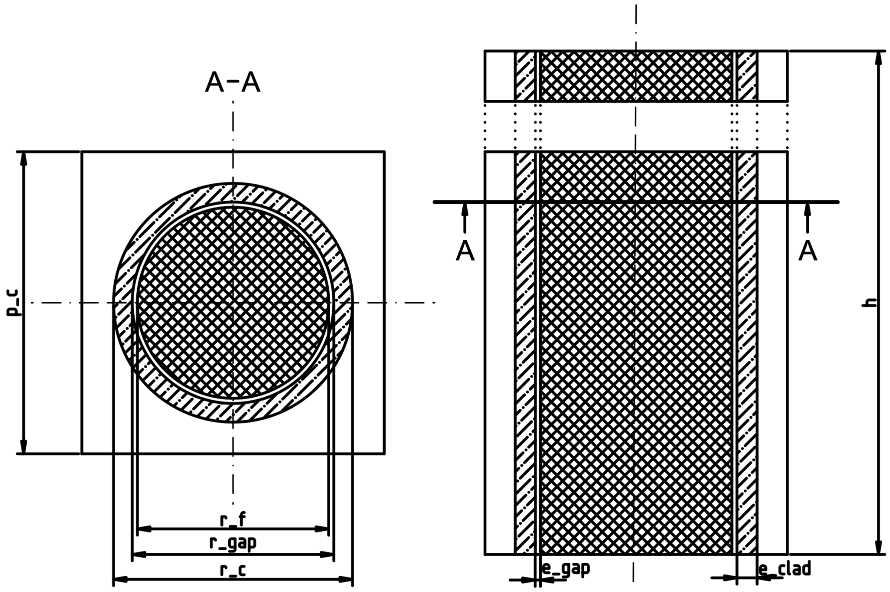


Figure 5.1 Schem of the geometry studied

In every benchmark, design and code reviewed in the section 2.4, the fixed parameters are the material properties, the geometry, and the thermohydraulic boundaries. The thermohydraulic boundaries are the inlet mass flow rate and temperature and the exit pressure. With the code update, the intention is to completely redesign the convection part of the `THM:`, keeping as much of the same notation and inputs as possible in the code. The new model needs the following inputs (the new input in $\texttt{THM}_p$ is colored in red):

Table 5.1 Model Parameters and Properties

Category	**Parameters**
Geometrical properties	Canal type
	Outer clad radius (Pin cell radius)
	Inner clad radius (or the thickness of the clad)
	Fuel radius
	Length of the pin cell
	Height of the pin cell
Thermohydraulic properties	Outlet pressure
	Inlet temperature
	Inlet mass flow rate
Mechanical properties	Thermal conductivity of the fuel
	Thermal conductivity of the clad
	Heat transfer coefficient through gap
Numerical solution parameters	Number of radial mesh in the fuel
	Number of radial mesh in the clad
	Number of axial mesh in the coolant
	Name of the void fraction correlation to use
	Name of the friction factor correlation to use
	Name of the two-phase multiplier factor correlation to use
	Name of the numerical method to use
	Max number of outer iterations
	Tolerance on the residuals

Table 5.2 Input of the actual `THM:` module and proposition for new

The new variable purpose are:

Parameter	**Purpose**
Correlation selection	Model flexibility for different application
Tolerance (ϵ_{tol})	Adaptive convergence control
Max iterations	Computational resource management

Table 5.3 Purpose of new input of the updated `THM:`

5.2 Two-phase simulation using OpenFOAM and the nuclear sub-solver GeN-Foam

The solver GeN-Foam is based on OpenFOAM, an open-source C++ library for solving partial differential equations using finite-volume discretisation. GeN-FOAM is a comprehensive solver designed for reactor analysis across multiple physical domains, offering capabilities to solve:

- Neutronics, encompassing point kinetics, transient and eigenvalue diffusion, eigenvalue adjoint diffusion, transient and eigenvalue SP3, and eigenvalue discrete ordinates models.
- One-phase thermal-hydraulics, employing RANS-CFD and porous-medium coarse-mesh methodologies, which can be utilised independently or in combination within the same mesh.
- Two-phase porous-medium thermal-hydraulics, employing an Euler-Euler model with specific models and correlations tailored for sodium and water.
- Sub-scale solid structure temperatures within porous-medium regions, with options for user-selectable models including 1-D fuel, fixed temperature, fixed power, fuel pebbles, heated rods, as well as a generic lumped-parameters model based on electric equivalence principles.
- Thermal-mechanical analysis based on linear thermo-elasticity, facilitating the evaluation of deformations and temperatures within solid structures. These deformations can then be employed to adapt meshes for thermal-hydraulics and neutronics simulations.

The current version of GeN-Foam is based on OpenFOAM, ESI/OpenCFD distribution, currently v2412. The source code can be found at: GeN-Foam gitlab and version 2412 of OpenFOAM.

GeN-Foam was created to solve various problems in multiphysics reactor modelling. Older coupled codes do not account for non-linearities, don't support parallel computing (which limits problem size) and are challenging to modify. In GeN-Foam three distinct meshes are employed for conducting thermal-hydraulics, thermal-mechanics, and neutron diffusion analyses. Within the fuel zones of the thermal-hydraulics mesh, the sub-scale fuel model is resolved individually within each mesh cell.

In this study we will focus on the thermohydraulic subsolver of GeN-Foam developed by Stephan Radman [37]. It is an advanced tool designed for thermohydraulic simulation in particular sodium-cooled fast reactors (SFRs). It utilises a coarse mesh methodology. This chapter details the mathematical formulations, physical models, and numerical methods that were implemented.

5.2.1 Thermohydraulic mathematical and numerical modelling in GeN-Foam

Governing equation

The coarse mesh methodology employs volume averaging. It is sometimes referred to as the porous body approach or porous medium-based coarse mesh. Where RANS focuses on simplifying turbulent flow simulation by averaging small-scale fluctuations, the coarse mesh methodology aims to simplify the entire simulation of large-scale systems by averaging over large spatial regions. It reduces the complexity and computational requirements of the simulation. The coarse mesh methodology involves spatial averaging over much larger volumes, leading to an equation representing the system at a much coarser resolution. On the contrary, the RANS involves time or spatial averaging at a much finer scale, leading to mean flow equations with still relevant details.

The core of GeN-Foam's fluid dynamics modelling is governed by the Navier-Stokes equations in a coarse mesh. These equations describe the conservation of mass, momentum, and energy within the reactor's coolant and moderator systems.

Mass Conservation Equation for phase k or j in a multi-phase system:

$$\frac{\partial}{\partial t}\left(\varepsilon_k \rho_k\right) + \nabla \cdot \left(\varepsilon_k \vec{v}_k \rho_k\right) = -\sum_{j \neq i} \Gamma_{i \to j} \tag{5.1}$$

with $\Gamma_{i\to j}$ the inter-phase specific mass transfer.

Momentum conservation equation for phase k or j in a multi-phase system:

$$\frac{\partial}{\partial t}(\varepsilon_k \rho_k v_k) + \nabla\cdot(\varepsilon_k \rho_k \vec{v}_k \otimes \vec{v}_k) = -\varepsilon_k \nabla P + \nabla\cdot(\varepsilon_k \sigma_{d,k}) + \varepsilon_k \rho_k g - \sum_{j\neq k}(\Gamma^{+}_{k\to j} v_k - \Gamma^{k}_{k\to j} u_j + M_{k\to j}) - M_{k\to s} \tag{5.2}$$

with $\sigma_{d,k}$ the deviatoric component of the stress tensor. The $\Gamma^{+}_{k\to j} v_k, \Gamma^{k}_{k\to j} v_j, M_{k\to j}, M_{k\to s}$ are the momentum source terms.

Energy conservation equation for phase k or j in a multi-phase system:

$$\begin{aligned}\frac{\partial}{\partial t}(\varepsilon_k\rho_k h_k) + \nabla\cdot(\varepsilon_k\rho_k\vec{v}_k h_k) \quad = \quad & -\nabla\cdot(\varepsilon_k\vec{T}_k v_k\cdot\nabla T_k) + \varepsilon_k\frac{\partial}{\partial t}P + \varepsilon_k\rho_k\vec{v}_k\cdot\vec{g} + \varepsilon_k q_{int,k} \\ & - \textstyle\sum_{j\neq k}(\Gamma^i_{k\to j}h_k + q_{k\to j}) - q_{k\to s}\end{aligned} \tag{5.3}$$

Modeling

Flow regime map

Instead of embedding the flow regime determination directly within each model (e.g., drag model, heat transfer model), Gen-Foam uses a centralised flow regime map that evaluates the flow regime for each cell once per iteration or time step. The user gives the flow regime map. The flow regime map is based on key dimensionless parameters characterising each cell's flow. Common parameters include Reynolds Number (Re), Weber Number (We, represents the relative importance of inertia compared to surface tension) and void fraction (ε). Once the flow regime is identified, the appropriate set of physical models and empirical correlations is applied to that cell.

Momentum transfer closure

The momentum transfer closure in Gen-Foam, as described in Stephan Radman's thesis is modeled n the term $M_{i\to j}$. The momentum transfer at the fluid-structure interface is modelled as a frictional pressure loss. For a single-phase fluid interacting with an isotropic structure, the momentum transfer can be expressed as: $\mathbf{M}_{i\to s} = -\nabla P\Big|_{i\in\mathbb{K}} = K_{is}\cdot\mathbf{u}_k$. Where K_is is the fluid-structure drag factor. The frictional pressure drop is modelled using the Darcy-Weisbach equation.

Heat and mass transfer closure

The fluid-structure heat transfer term $q_{i\to s}$ represents the heat flux from the fluid phase i to the solid structure s. This term is modeled using the following equation: $q_{i\to s} = f_{A''_c,i}\,A''_s H_{is}\,(T_k - T_s)$ with $f_{A'''_s,i}$ the fluid-structure contact area fraction, A'''_s the structure volumetric surface area density, H_{is} the fluid-structure heat transfer coefficient, T_s and T_i the structure surface temperature and the fluid temperature respectively. The code using also Fluid-Fluid heat transfer closure equation and mass transfer and phase change.

Turbulence modelling

Turbulence modelling in Gen-Foam is based on the standard $k-\epsilon$ turbulence model, which is widely used in engineering applications due to its balance between accuracy and computational efficiency. The model focuses on resolving the average flow characteristics while accounting for the effects of turbulence through additional transport equations for the turbulent kinetic energy (k) and its dissipation rate (ϵ).

Numerical Solution Methods

The Finite Volume Method (FVM) is the primary discretisation approach used in Gen-Foam to solve the governing Partial Differential Equations. This is the main discretisation method implemented natively in OpenFOAM. For spatial discretisation, the solver uses the Gauss-Green theorem to convert volume integrals into surface integrals, which are then discretised over the faces of each control volume. For temporal discretisation, Gen-Foam uses implicit time integration schemes.

Gen-Foam employs a pressure-based approach, specifically the PIMPLE algorithm, which is a hybrid of the PISO (Pressure Implicit with Splitting of Operators) and SIMPLE (Semi-Implicit Method for Pressure Linked Equations) algorithms. This algorithm maintains the transient resolution used in the PISO method and incorporates features from the SIMPLE algorithm that allow for more flexible time-stepping. The PIMPLE algorithm enables a larger time step than other resolution algorithms.

5.2.2 Validation and application - GeN-Foam test cases

OECD/NRC PWR PSBT benchmark test case for a single pin (Exercise 1, Case 12223) has been used for the numerical modelling and validation. This benchmark is a renowned reference for validating calculation codes in nuclear engineering, developed from the vast NUPEC database and updated by the OECD.

The two phases of the benchmark are interesting in the case of this master's thesis. Known as the "Void Distribution Benchmark," the first phase aims to provide comprehensive data for validating void distribution models in pressurised water reactors (PWRs). It comprises four exercises. The first exercise focuses on a single sub-channel steady-state scenario, ideal for improving and validating numerical models related to vacuum generation and distribution within a sub-channel. There is also a steady-state exercise for an assembly and two transient assembly exercises.

The second phase is the "departure from nucleate boiling" phase. This benchmark aims to develop truly mechanistic models for DNB prediction. It comprises three exercises. The first will be the most useful in our case, as it is intended as a reference for steady-state fluid temperatures. Model 2 is a model for testing predictive DNB simulations. Finally, the last model is a transient DNB benchmark.

The test facility associated with this benchmark is capable to test high-pressure, high-temperature conditions typical of PWRs. It uses a bundle of electrically heated rods to reproduce the thermo-hydraulic environment inside a PWR fuel assembly, making it a robust tool for testing and refining fluid flow and heat transfer models. The use of this benchmark aim to ensure that the numerical models accurately reflect the complex fluid dynamics that occur within PWR fuel assemblies under various operational conditions.

Although the OECD/NRC PWR PSBT benchmark is made explicitly for PWR, it can provide information that is highly relevant to BWR as well. The benchmark's focus on void distribution and two-phase flow behaviour within fuel assemblies is critical for understanding similar phenomena in BWR. Many models and hypotheses developed for PWRs, such as those related to void generation, heat transfer, and fluid dynamics, can be adapted or extended to BWRs.

The fundamentals governing the interaction between liquid and vapour phases, the influence of geometric factors on flow behaviour, and the impact of boundary conditions on the entire system are relevant across both reactor types. Therefore, the PSBT benchmark validates PWR models and serves as a foundation for refining and validating BWR thermal-hydraulic models.

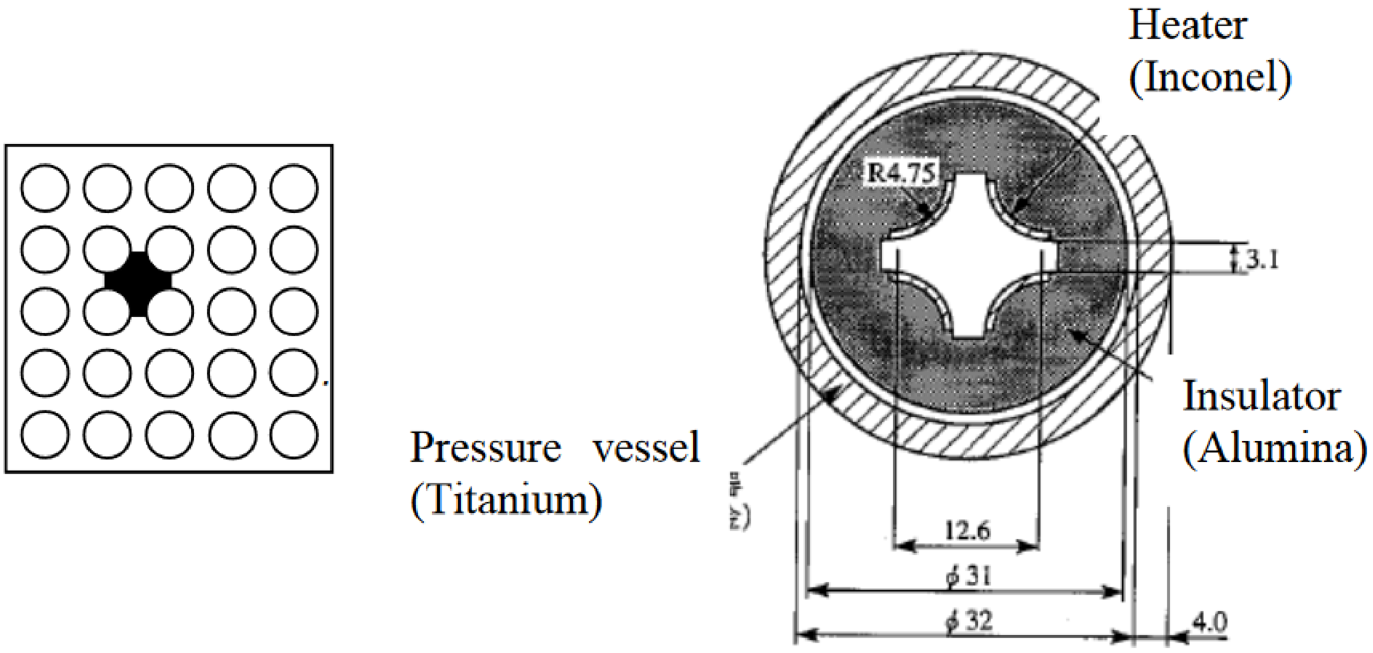


Figure 5.2 Single subchannel geometry used to validate the boiling capacities of GeN-Foam

An electrically heated rod bundle was used to simulate a partial section and the full length of a PWR fuel assembly. The effective heated length is 3,658 mm, divided into three sections, with coolant flowing from the bottom of the pressure vessel to the top of the assembly under test. A centered pincell is studied, as shown in figure 5.2. The equivalent pincell is a water channel surrounded by a heating rod, itself surrounded by an insulator.

Vacuum fraction measurements were carried out using a gamma-ray transmission method, which enabled the determination of the flow density and vacuum fraction in the two-phase gas-liquid flow.

For the rod-beam tests, void fraction data for 36 subchannels were obtained using multi-beam systems, with measurements taken at three axial elevations. The relationships between the mean void fractions of the subchannels and the string were used to adjust and correct the void fraction values.

Geometrical characteristics are given in the following table.

Item	**Data**
Flow area (mm^2)	107.098
Heated perimeter (mm)	29.845
Wetted perimeter (mm)	54.645
Heater outer radius (mm)	4.75
Heater thickness (mm)	0.85
Insulator outer radius (mm)	31
Axial heated lenght (mm)	1555
Axial power shape	Uniform

Table 5.4 Geometrical and heated characteristics of sub-channel

More information about this experimental benchmark are detailed section 2.3.1 and in the publication from the OECD [15]

CHAPTER 6 Results and code to code validation

This section presents the results of the one-dimensional steady-state compressible drift flux model developed for simulating turbulent flow inside a pincell. The improvement of the solver capabilities can be separated into two cases. Firstly, the solver is now able to model the pressure drop. Secondly, the solver performance to model the void fraction and all related thermohydraulic properties (velocity, temperature, etc.) is now more accurate. Both of these improvements have been made on a Python solver. For the moment, only the first improvements have been ported to Donjon5.

To discuss the result, we will first study the improvement on Donjon5 and compare it to an analytical solution.

In the second part, we will analyse the Python solver's performance. First, we will perform a validation test case on a simple problem and compare it to a reference case and a benchmark. Then, we will validate models and correlations used to characterise the flow. We will then discuss the solver's computational performance and analyse the results.

6.1 `THM:` pressure drop implementation

One of the significant contributions from this master thesis is the implementation of the pressure drop in the case of a monophasic and two-phase flow subchannel analysis. This achievement addresses a long-standing industrial request to resolve a persistent issue where the code failed to converge when the pressure drop calculation was active. In this section, we will present the results obtained from **only** the implementation and validation of pressure drop calculations. This correction represents a significant milestone in developing the thermohydraulic solver `THM:` of the core calculation code `Donjon5` [9].

The updated solver introduces several critical changes contributing to the algorithm's convergence. First, where there was no pressure-velocity coupling in the previous solver, a coupling was implemented in the updated one. This explicit coupling solves the mass and momentum conservation equations simultaneously, instead of updating the velocity, and then use the updated velocity in the momentum conservation equation. The previous `THM:` code can be found in appendix B. The numerical scheme is now a linear system of equations, where the previous one was an incremental velocity update at each axial step. This coupling enhances numerical stability and helps the numerical algorithm to converge. The last difference is the use of correlation. Where the previous version of `THM:` use the Müller-Steinhagen [38] to model the friction factor the `THM`$_p$ use more reliable correlation such as Blasius [35] and Churchill [34] correlations.

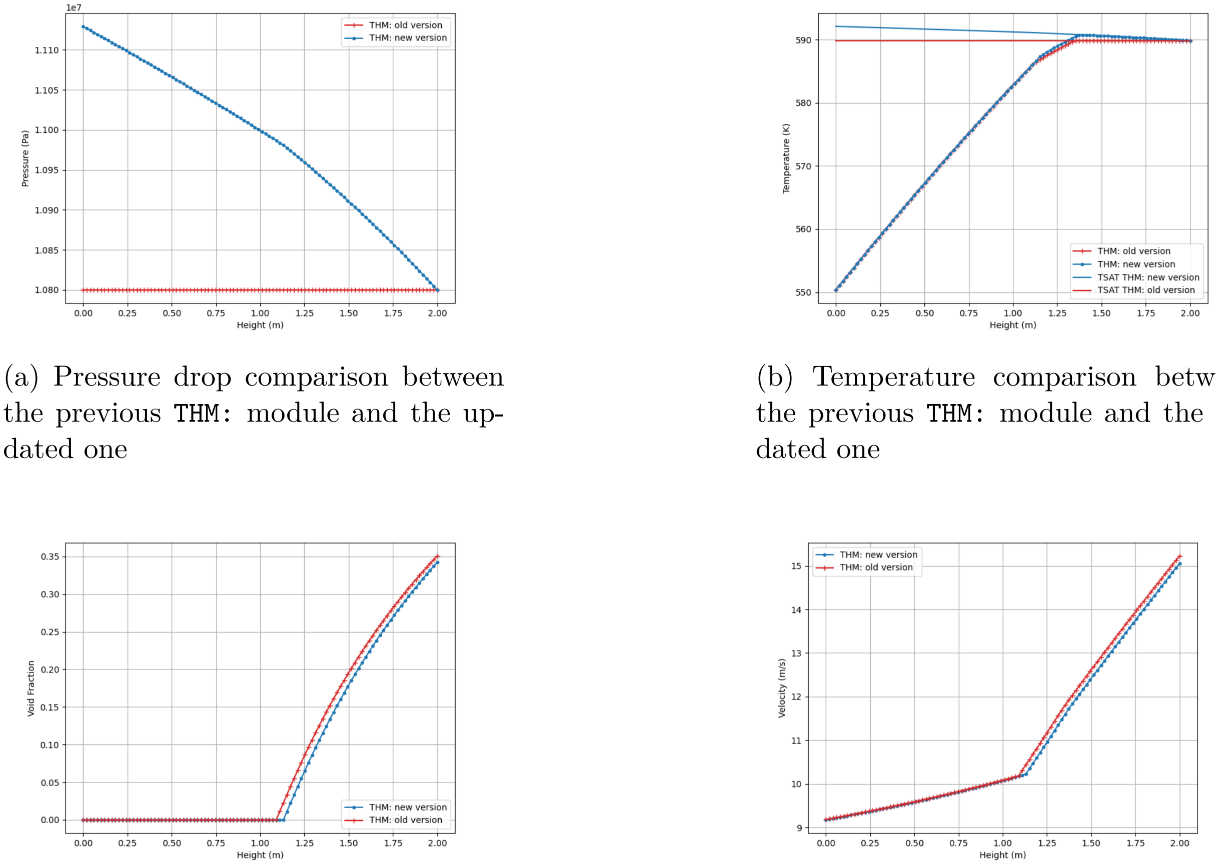
To go further in the implementation of `THM`$_p$ in `THM:`, it could be essential to implement the developed drift flux model. Using the drift flux model in the framework helps provide a more accurate representation of phase distribution along the channel. In the conservation equation of the momentum, the drift flux model term is the following expression:

$$\frac{\partial}{\partial z}\frac{\varepsilon}{1-\varepsilon}\,\frac{\rho_g\,\rho_\ell}{\rho_m}V_{gj}^2 \tag{6.1}$$

The drift flux model implementation helps particularly to model the creation of void fractions and capture segregation phenomena and slip between phases.

6.1.1 New results compared to the previous code

The following graphs illustrate the improvement of the updated `THM:` module. It now incorporates a pressure drop calculation along the channel: a significant advancement over the previous version, in which pressure was treated as constant. This modification directly influences key thermodynamic and hydrodynamic parameters, including density, saturation temperature, boiling behaviour and velocity.



(a) Pressure drop comparison between the previous `THM:` module and the updated one

(b) Temperature comparison between the previous `THM:` module and the updated one

(c) Void fraction comparison between the previous `THM:` module and the updated one

(d) Liquid velocity comparison between the previous `THM:` module and the updated one

Figure 6.1 Pressure, temperature, void fraction and liquid velocity comparison between the previous `THM:` module and the updated one

The pressure profile highlights the core improvement of the new `THM:` module. Unlike the old version (constant pressure), the updated model shows a clear pressure drop along the channel, decreasing from ≈ 11.1 MPa at the base to ≈ 10.8 MPa at the top (2.00 m). This gradient of 30 kPa is physically realistic and essential for modelling systems with significant elevation changes or flow resistance.

The observed changes in velocity, temperature and void fraction are linked to the updated pressure calculation. The pressure drop along the channel changes the fluid density (via the FREESTEAM tables), which affects the velocity profile through the mass conservation equation. The temperature profile shifts are more significant during nucleate boiling. Knowing that the saturation temperature T_{SAT} is no longer constant but decreases with pressure, the boiling departure occurs slightly later in the new model. It appends because it requires a marginally higher mixture temperature to initiate phase change. The pressure-dependent T_{SAT} consequently change the void fraction profile. The void fraction evolution differs because the temperature reaches T_{SAT} later. Since steam is created later, density changes abruptly later and thus velocity too.

6.1.2 Verification of the code

To make the verification of the implementation of mass and momentum conservation equations in the updated THM$_p$ module, an analytical solution was found. The density was artificially prescribed as $\rho_m = \dfrac{1}{a \cdot z + b}$ (with a and b real constants to fix the inlet and outlet density) to simulate boiling effects while enabling an exact solution for this kind of simplified simulation. The friction factor was set to 0.001 (same friction factor along the channel), and the two-phase multiplier was set to 1 (no boiling effect on the pressure drop). The numerical results were compared to the derived analytical solutions for both pressure and velocity fields. The condition $\dfrac{\partial}{\partial z}\rho_m u_m = 0$ leads to a linear velocity profile:

$$u_m(z) = K(a \cdot z + b) \tag{6.2}$$

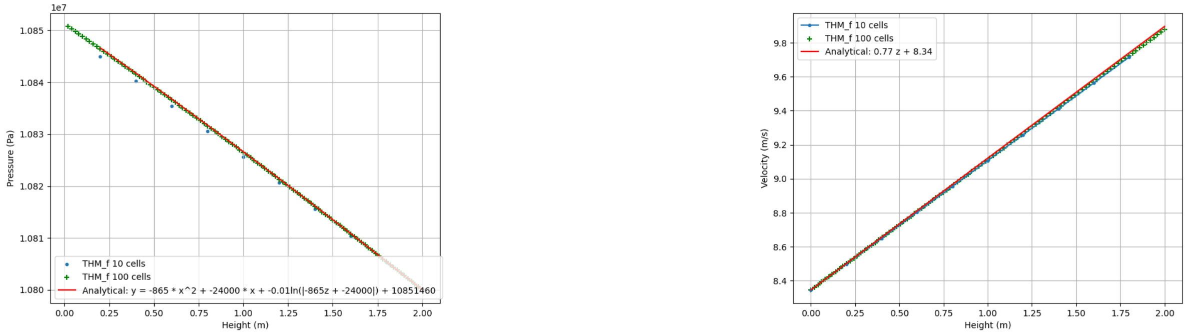
with K a real constant depending on the inlet mass flow rate. To find the pressure profile, the momentum conservation equations needs to be solve with the hypothesis presented before. We solve the simplified conservation equation:

$$\rho_m \frac{\partial u_m^2}{\partial z} + \frac{\partial P_m}{\partial z} = -\rho_m u_m^2 \frac{\phi_m f}{D_h} - \rho_m g \tag{6.3}$$

Using the velocity profile already determined, the pressure profile is:

$$P(z) = -K^2(1 + \frac{\phi_m f}{D_h})(\frac{1}{2}az^2 + bz) + \frac{g}{a} ln(|az + b|) + C \tag{6.4}$$

with C a real constant depending on the pressure boundary condition. Plotting the numerical solution for 10 and 100 volumes versus the analytical solution leads to the following graphs:



(a) Numerical solution of the pressure profile of $\texttt{THM}_f$ for 10 and 100 volumes compared to the analytical solution

(b) Numerical solution of the velocity profile of $\texttt{THM}_f$ for 10 and 100 volumes compared to the analytical solution

Figure 6.2 Numerical solution of $\texttt{THM}_f$ for 10 and 100 volumes compared to the analytical solution

The numerical solutions for 10 and 100 cells converge toward the analytical profile, with the 100-cell case showing excellent agreement. Minor deviations in the coarse mesh (10 cells) are attributed to discretisation errors. The numerical pressure profiles exhibit strong consistency with the analytical solution, particularly for the 100-cell mesh. The quadratic shape confirms the correct implementation of the momentum equation, including friction and gravitational terms.

The numerical solution was further validated through a mesh convergence analysis, comparing the results against the derived analytical solution as a function of the number of control volumes. Results are shown in the following plot:

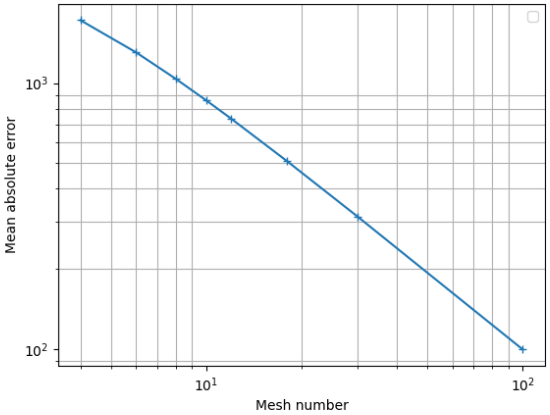


Figure 6.3 Convergence of the numerical solution compared to the analytic solution

The study demonstrates that the numerical implementation achieves an order of convergence of 0.9, indicating near-linear convergence as the mesh is refined. The near-linear convergence suggests that the numerical model correctly resolves the dominant physical processes. Minor deviations (0.1) from ideal first-order behaviour (1) may be explained by the treatment of nonlinear terms in the momentum equation. This provides confidence in the model's ability to simulate the pressure drop accurately.

6.2 Complete steady states model validation

To validate the $\texttt{THM}_p$ model we will use the CFD code GeN-Foam described in section 5. To compute the errors, we will consider different types of errors depending on the variable studied. The relative error, absolute error, RMS error and MAX error will be used. To investigate the error on the pressure P, temperature T, and velocity u_m computed by the $\texttt{THM}_p$ model, the scaled relative error will be used. The subscript $_p$ will be for the $\texttt{THM}_p$ model and $_{GF}$ for the GeN-Foam results. With P_i, the variable at volume i, for P:

$$\Delta_{rel,P_i} = \frac{|P_{i,p} - P_{i,GF}|}{P_{i,GF,max} - P_{i,GF,min}} \tag{6.5}$$

$$\Delta_{abs,P_i} = |P_{i,p} - P_{i,GF}| \tag{6.6}$$

and $\Delta_{rel,P} = \dfrac{\Sigma_{i=1}^{N} \Delta_{rel,P,i}}{N} \cdot 100$

Because the void fraction is already a percentage, we will perform an absolute difference:

$$\Delta_{abs,\varepsilon,i} = |\varepsilon_{P,i} - \varepsilon_{GF,i}| \tag{6.7}$$

and $\Delta_{avg,\varepsilon} = \dfrac{\Sigma_{i=1}^{N} \Delta_{avg,\varepsilon,i}}{N}$

The RMS errors are defined as:

$$\Delta_{RMS,P} = \sqrt{\frac{1}{N} \Sigma_{i=1}^{N} \left(\Delta_{rel,P_i}\right)^2} \tag{6.8}$$

and

$$\Delta_{RMS,\varepsilon} = \sqrt{\frac{1}{N} \Sigma_{i=1}^{N} \left(\Delta_{abs,\varepsilon,i}\right)^2} \tag{6.9}$$

6.2.1 Simple validation against reference code

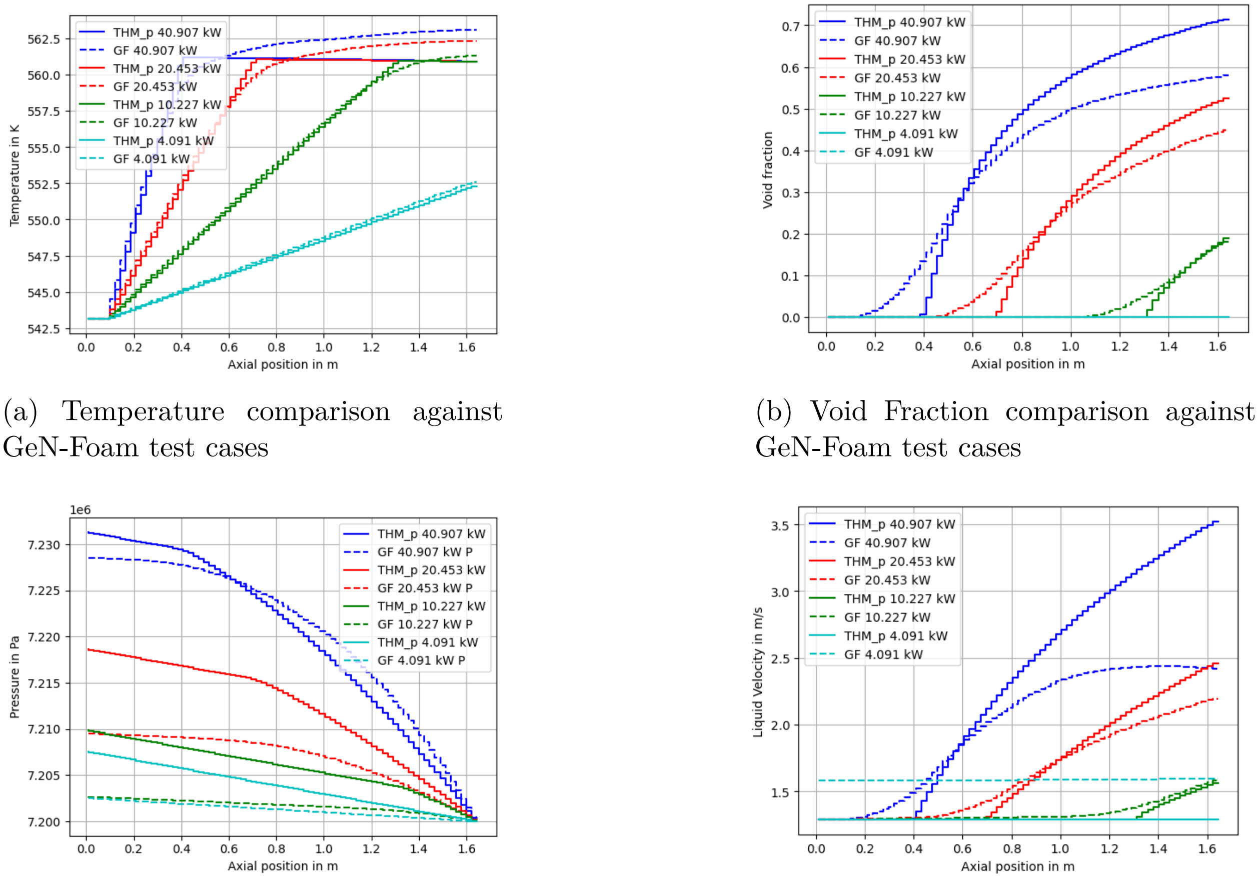
To verify the accuracy of the developed one-dimensional steady-steady states drift flux model, we performed a simple simulation based on the ATRIUM 10 benchmark [16] and compared it to the reference code Gen-Foam.

The simulation parameters from the benchmark are summarised in the table 6.1

Geometric Parameters		**Thermohydraulic Parameters**	
Parameter	**Value**	**Parameter**	**Value**
Axial mesh points (I_z)	70	Power	4.091 - 40.907 kW
Fuel length (L_f)	1.655 m	Mass flux rate (q_{flow})	0.08407 kg/s
Fuel radius (r_f)	0.4435 cm	Outlet pressure (p_{out})	7.2 MPa
Clad radius (r_{clad})	0.5140 cm	Inlet temperature (T_{in})	543.15 K
Channel pitch (r_w)	1.295 cm	Inlet velocity (u_{in})	1.29 m/s

Table 6.1 Geometric and Thermohydraulic Parameters for the simple test

Results are presented in the following plots:



(a) Temperature comparison against GeN-Foam test cases

(b) Void Fraction comparison against GeN-Foam test cases

(c) Pressure comparison against GeN-Foam test cases

(d) Liquid velocity comparison against GeN-Foam test cases

Figure 6.4 Temperature, void fraction, pressure and liquid velocity comparison against GeN-Foam test case for 10.227, 20.453 and 40.907 kW fuel power

Table 6.2 shows the RMS (Δ_{RMS}), the average (Δ_{avg}) and maximum (Δ_{max}) deviations obtained when comparing $\texttt{THM}_p$ and GeN-Foam distributions obtained for ε and p and T_m. The errors on $\texttt{THM}_p$ the predicted pressure drops are about $5kPa$ for the maximum deviation.

Quantity	Δ_{RMS} (%)			Δ_{max}(%)			Δ_{avg}(SI)		
Power (kW)	10.227	20.453	40.97	10.227	20.453	40.97	10.227	20.453	40.907
Temperature	1.27	4.05	6.60	3.05	7.49	10.96	0.20	0.63	1.12
Void fraction	1.19	4.06	8.18	4.92	7.87	13.42	0.005	0.030	0.069
Pressure	174.86	64.19	6.61	273.53	98.05	9.89	1.60	0.57	0.060
Velocity	2.10	5.31	20.41	6.40	12.01	45.47	0.021	0.072	0.349

Table 6.2 Absolute RMS, maximum, and average `THM prototype`-GeNFoam deviations on pressure, void fraction and coolant temperature at power levels of 10.227, 20.453 and 40.907 kW

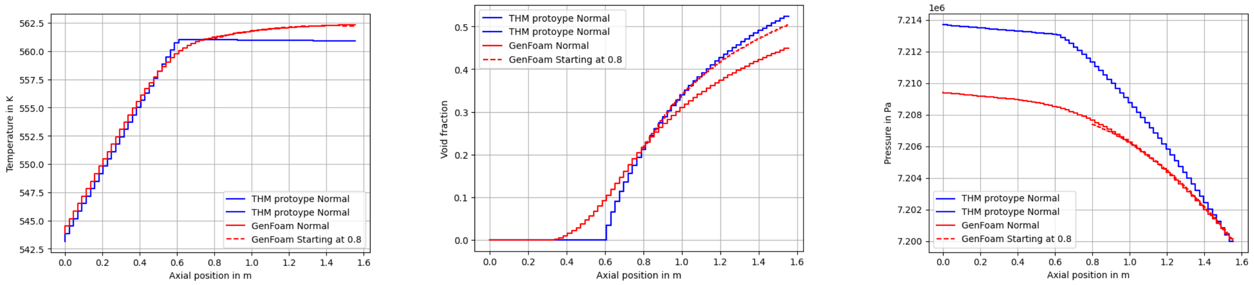
The comparison between the two codes showed a strong correlation, with deviations within acceptable engineering limits. As depicted in Figure F.1a, the temperature profiles along the conduit from both models align closely, indicating that the drift flux model accurately captures the heat transfer processes modelled by the energy conservation equation. Similarly, the pressure drop predictions, illustrated in Figure F.1c, demonstrate good agreement, validating the implementation of the friction factor and two-phase flow correlations in our model. However, it is still challenging to predict the pressure drop, mostly in two-phase flow and in low-velocity two-phase flow. Some researchers [39] showed that existing correlations are pretty reliable but limited, especially for low velocity two-phase flow.

The main result of this master's thesis is predicting the void fraction distribution along the channel. We can note 2 principal effects:

- **There is an earlier boiling onset in GeN-Foam**. The subcooled boiling model in GeN-Foam may enable earlier initiation of vapour generation. It can be due to the explicit tracking of bubble formation at nucleation sites. This explains the higher void fraction in GeN-Foam comparing to `THM`$_p$ at lower axial position.
- **Saturation effect at high void fractions**. As the void fraction increases, the system approaches a limit where further vapour generation can be constrained by some physical effect such as: the reduced heat transfer efficiency, the increased interfacial drag between phases, which can suppress bubble growth and the fact that pressure changes can limit the phase change rates.

These physical limitations align with the trends that both models exhibit similar behaviour at higher axial positions despite differing initial dynamics. The absence of a subcooled boiling model in the current model could delay vapour generation initiation. Implementing a similar mechanistic model (e.g., wall boiling correlations, bubble population balance) would improve agreement with GeN-Foam while preserving the logical saturation trend observed at higher void fractions.

To continue investigating the new solver, we can compare the two codes with an inlet void fraction. Plots are presented in Figure 6.5. In fact, simulating a channel with an inlet void fraction could allow us to compare the two codes, avoiding the difference in the subcooled model when modelling the start of bubble formation.



(a) Temperature comparison against GeN-Foam

(b) Void Fraction comparison against GeN-Foam

(c) Pressure comparison against GeN-Foam

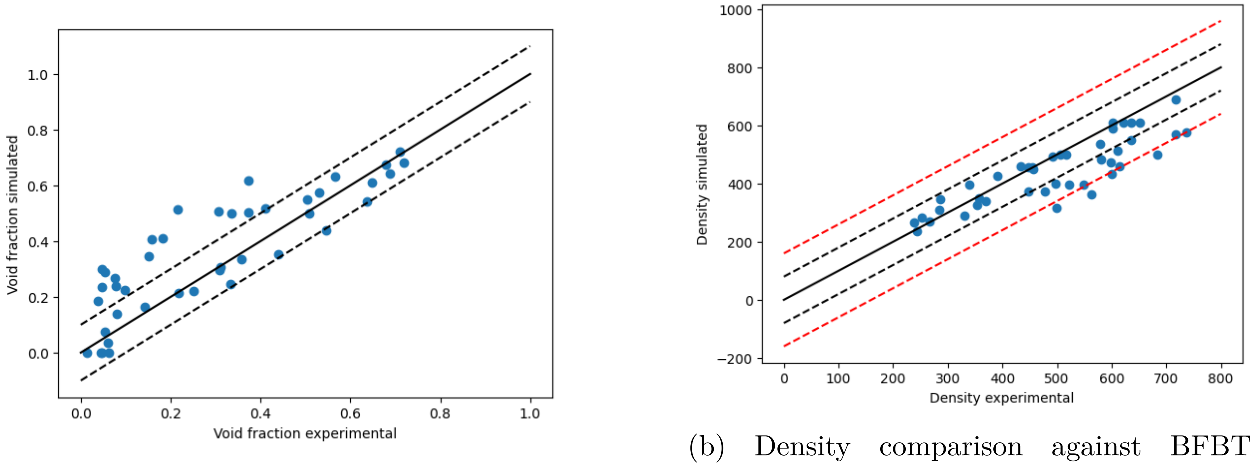
Figure 6.5 Comparison between $\mathtt{THM}_p$ and GeN-Foam with a simulation starting at 0.8 m with inlet void fraction equal to the $\mathtt{THM}_p$ void fraction at 0.8 m

In figure 6.5, red lines are the two GeN-Foam modelisations and the dotted one is with an inlet void fraction and the pipe starting at 0.8m, where some steam has already been generated. It can be pointed out that the new solver seems pretty consistent with the reference code despite the difference and the error created in the models used to model the beginning of the ebullition. Integrating a new ebullition model will improve the solver's ability to simulate the start of boiling and enhance the accuracy at the outlet of the channel.

Moreover, further validation against experimental data may be interesting to isolate model-specific uncertainties in GeN-Foam. The appendix F shows a comparison with the code TWOPORFLOW [40] and reveals outstanding performance.

6.2.2 Validation against the BTBF PWR subchannel benchmark

The following section validates the subchannel model against the OECD/NEA BTBF PWR benchmark (Phase I). It focuses on a single fuel pin configuration. The available experimental data from the OECD for this phase are limited to exit void fraction and density measurements.



(a) Void Fraction comparison against BFBT benchmark. Black dotted lines: ± 10%

(b) Density comparison against BFBT benchmark. Black dotted lines: ± 10%, red dotted lines ± 10 %

Figure 6.6 Terminal void fraction (a) and density (b) comparison against the Nupec OECD BFBT benchmark

The first figure compares the simulated void fraction from $\mathtt{THM}_p$. The black line corresponds to perfect agreement between the simulated values and the experimental data. The simulated values clearly correlate with the experimental data following the general trend. however, several points fall outside the ±10% bounds. The cases with deviations could stem from experimental uncertainties or limitations in the drift flux model. It may highlight some inconsistencies for some test cases due to the boundary conditions. Knowing which cases do not correspond to the experimental values can be interesting. We decided to investigate the problem and analyse whether the error is due to the drift flux model. To deal with this, we can plot the errors on the void fraction depending on the mean energy given to the fluid. To determine the mean energy given to the fluid during his residency in the channel, we can calculate the ratio q'''/u_m which is the same dimension as an energy over a length to the power 4. The results are presented below figure 6.7

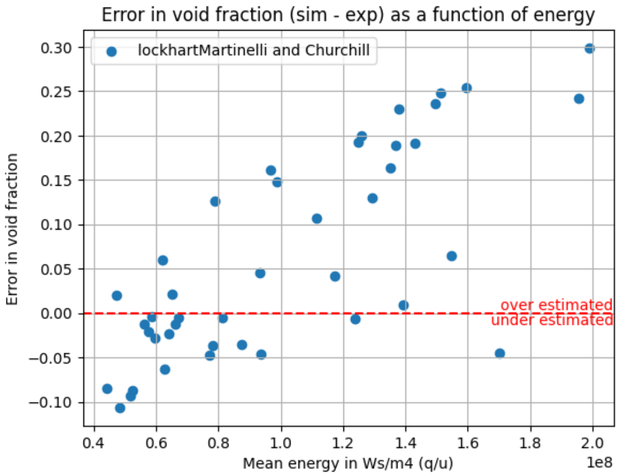


Figure 6.7 Error in void fraction ($\varepsilon_{simulated} - \varepsilon_{experimental}$) as a function of energy.

The numerical simulations show systematic deviations in predicting the exit void fraction: at low energy levels (corresponding to low void fractions), the model underestimates the void fraction, while at high energy levels (high void fractions), it overestimates it. As discussed in Section 6.2.1, this discrepancy can be attributed to the absence of a subcooled boiling model, which is critical for accurately capturing wall-driven nucleate boiling dynamics. Additionally, since the current model is restricted to a one-dimensional axial framework, it lacks radial mesh resolution. It can not resolve flow patterns, such as bubbly flow, that depend on multidimensional phenomena.

6.2.3 Models and correlation comparison

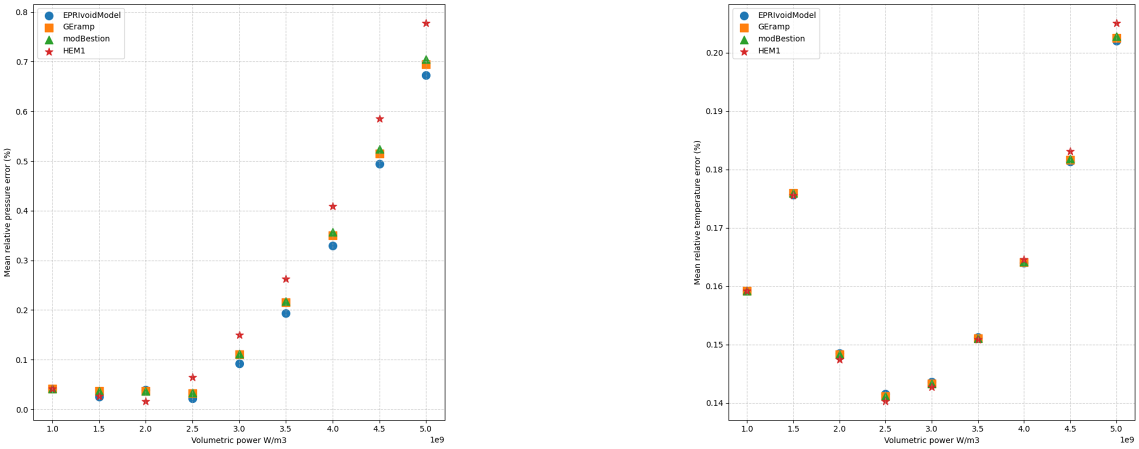
Void Fraction model analysis

The impact of different void fraction correlations on the model's predictions was assessed by implementing several commonly used correlations for the drift flux model, such as the EPRI void correlation [41], the HEM correlation [33], the GE-ramp correlation [42] and the modified Bestion correlation [21] used in TRACE. Simulations were performed for a range of $1GW/m^3$ to $5GW/m^3$. Below those powers, there is no significant error with GeN-Foam. The power distribution is a constant profile using the geometrical and thermohydraulic parameters presented in Table 6.3:

Geometric Parameters		Thermohydraulic Parameters	
Parameter	**Value**	**Parameter**	**Value**
Axial mesh points (I_z)	75	Volumetric power	1 / 5 GW/m^3
Fuel length (L_f)	1.555 m	Mass flux rate (q_{flow})	0.3253 kg/s
Fuel radius (r_f)	0.2812 cm	Outlet pressure (p_{out})	14.7 MPa
Clad radius (r_{clad})	0.4749 cm	Inlet temperature (T_{in})	592.75 K
Channel pitch (r_w)	1.334 cm	Inlet velocity (u_{in})	1.29 m/s

Table 6.3 Geometric and Thermohydraulic Parameters for the void fraction model correlation

The main parameters affected by the correlation are shown below.



(a) Relative mean pressure error

(b) Relative mean temperature error

Figure 6.8 Pressure and temperature mean relative error compared to GeN-Foam test cases for a range of $1GW/m^3$ to $5GW/m^3$.

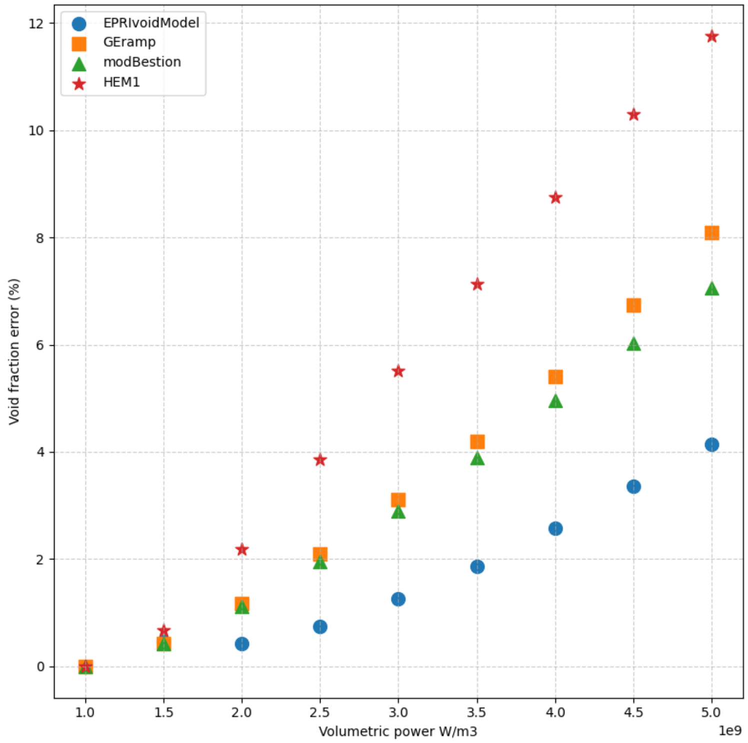


Figure 6.9 Void fraction mean error compared to GeN-Foam test cases for $1GW/m^3$ to $5GW/m^3$.

We can compute and summarise the max and mean RMS error calculated using every power amplitude to know which fits best in most situations. The results are shown in the following table:

Void Fraction correlation	**Root-Mean-Square Error**	**Maximum error**
EPRI	2.14 %	4.15%
HEM	4.39%	8.09%
GERamp	3.94 %	7.06 %
Modified Bestion	6.85%	11.76%

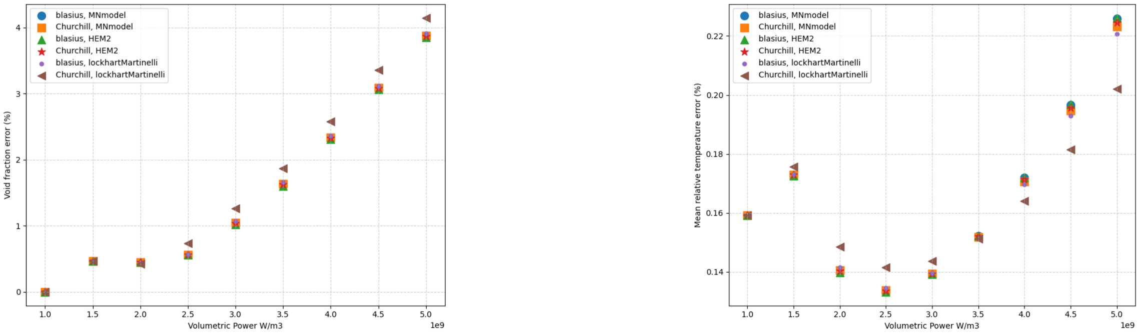
Table 6.4 RMS and maximum error on void fraction for the different void fraction correlations

The analysis revealed that the void fraction correlation chosen significantly influenced the simulated void fraction distribution. This difference increases in regions where there is a lot of steam. The EPRI void correlation is the one that fits our case best. It is one of the best maximum errors and mean errors. It is consistent with the results, which can be found in the PATHS publication [1].

Friction factor correlation effect on pressure drop

We investigated the effect of different friction factor models on the calculated pressure drops. The friction factor f and the two-phase multiplier ϕ_m can influence the pressure drop and other field variables. The previous section showed that the EPRI void model is the more accurate correlation in our cases. For the next simulations, we will use the EPRI void model.

The friction factor uses the Blasius [35] and Churchill [34] correlation, where the two-phase multiplier correlation can be chosen between the Lockhart Martinelli [36], HEM1 & HEM2 [33], and MNmodel [43] correlation. The figure 6.11 compares the pressure drop along the channel for each couple of correlations cited previously to a wide range of power amplitudes.



(a) Absolute void fraction error

(b) Relative mean temperature error

Figure 6.10 Void fraction and temperature error compared to GeN-Foam test cases for a range of $1GW/m^3$ to $5GW/m^3$.

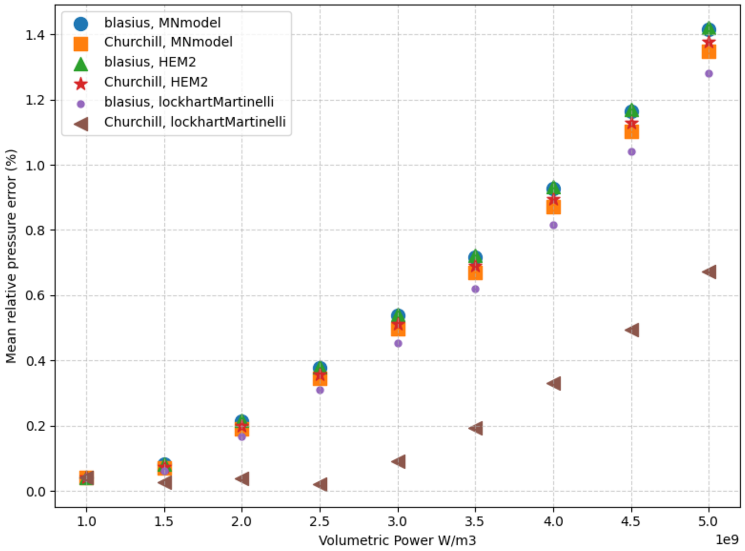


Figure 6.11 Relative mean pressure error compared to GeN-Foam test cases for a range of $1GW/m^3$ to $5GW/m^3$.

The graphs show a strong link between the chosen friction factor correlation and the two-phase multiplier, with the error on the pressure. It also shows that the selection of the correlation has negligible influence on temperature and void fraction predictions. We can compute and summarise the max and mean RMS error on pressure drop calculated using every power amplitude to know which fits best in most situations. The results are shown in the following table:

Friction factor	**Two phase multiplier**	**Root-Mean-Square Error**	**Maximum error**
Blasius	HEM2	0.612%	1.421%
Blasius	Lockhart-Martinelli	0.532%	1.279%
Blasius	Martinelli-Nelson	0.608%	1.415%
Churchill	HEM2	0.586%	1.378%
Churchill	Lockhart-Martinelli	0.212%	0.673%
Churchill	Martinelli-Nelson	0.570%	1.347%

Table 6.5 RMS and maximum error pressure for the different two-phase multiplier and friction factor correlations

The most Gen-Foam-accurate correlation is the Churchill + Lockhart-Martinelli couple correlation. This is especially noticeable at high power levels. However, the literature shows that modelling the pressure drop in two-phase flows is challenging. It is possible that GeN-Foam doesn't compute the pressure drop very well, making the comparison useless.

For future research, comparing an experimental benchmark of a heated vertical pipe with a two-phase flow could be interesting.

6.3 Perfomance

The computational performance of the model was evaluated by measuring computation time under different simulation conditions. The goal is to understand which part of the algorithm is slower and what the effect of the mesh refinement is to propose improvements for this project's future.

As part of the development of this solver, a temporal analysis was performed on the Python program to identify the functions that consume time and are called most frequently. This analysis proposes optimisation strategies by focusing on the most critical points. The `cProfile` tool for Python was used to collect performance statistics. The results were visualised using `SnakeViz` and exported into tables for Python analysis.

The profiling was limited to 6 levels to reduce the graph's and post-processing complexity. Based on the test case mentioned in table 6.1, it has been made. The figure 6.12 represents the time and call repartition between the main functions of the different classes.

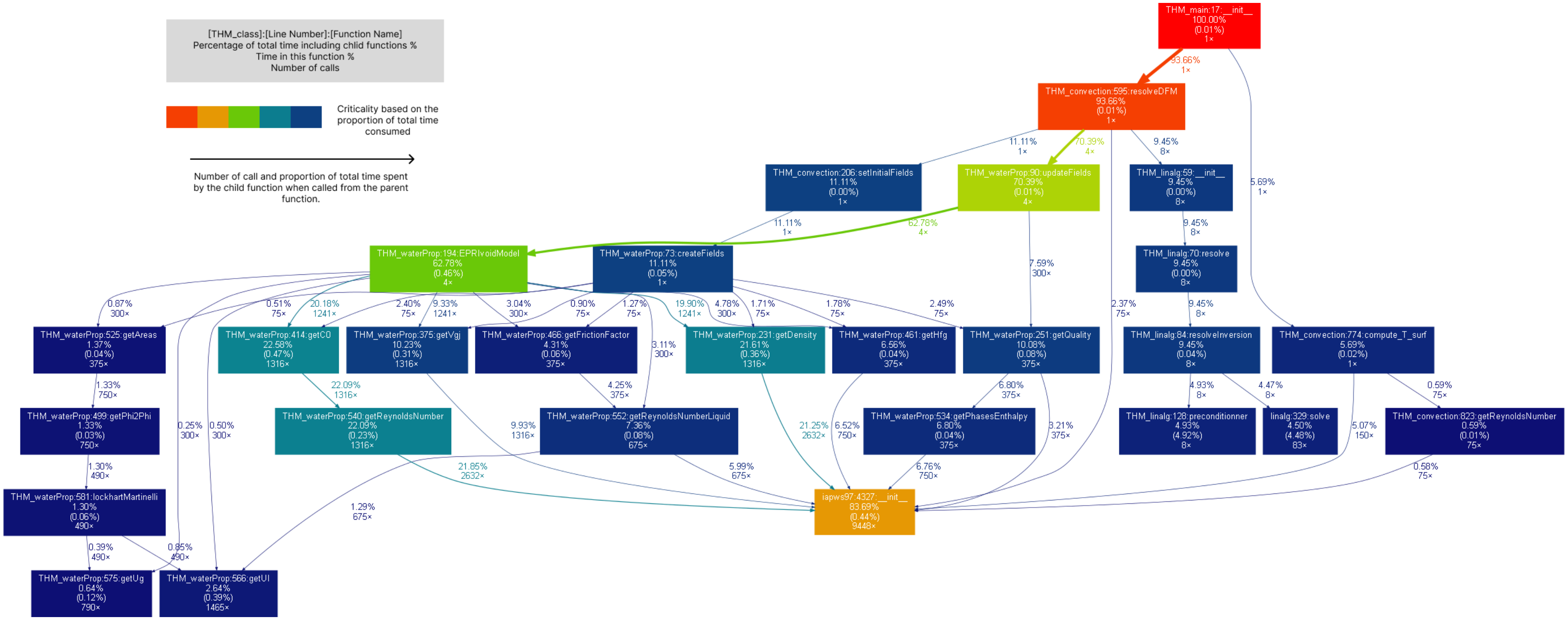

[THM_class]:[Line Number]:[Function Name]
Percentage of total time including child functions %
Time in this function %
Number of calls
Criticality based on the proportion of total time consumed
Number of call and proportion of total time spent by the child function when called from the parent function.
THM_main:17:__init__ 100.00% (0.01%) 1×
93.66% 1×
THM_convection:595:resolveDFM 93.66% (0.01%) 1×
11.11% 1×
70.39% 4×
9.45% 8×
5.69% 1×
THM_convection:206:setInitialFields 11.11% (0.00%) 1×
THM_waterProp:90:updateFields 70.39% (0.01%) 4×
THM_linalg:59:__init__ 9.45% (0.00%) 8×
11.11% 1×
62.78% 4×
9.45% 8×
THM_waterProp:194:EPRIvoidModel 62.78% (0.46%) 4×
THM_waterProp:73:createFields 11.11% (0.05%) 1×
7.59% 300×
THM_linalg:70:resolve 9.45% (0.00%) 8×
0.87% 300×
0.51% 75×
20.18% 1241×
2.40% 75×
9.33% 1241×
3.04% 300×
0.90% 75×
1.27% 75×
19.90% 1241×
4.78% 300×
1.71% 75×
1.78% 75×
2.49% 75×
2.37% 75×
9.45% 8×
THM_waterProp:525:getAreas 1.37% (0.04%) 375×
THM_waterProp:414:getC0 22.58% (0.47%) 1316×
THM_waterProp:375:getVgj 10.23% (0.31%) 1316×
THM_waterProp:466:getFrictionFactor 4.31% (0.06%) 375×
3.11% 300×
THM_waterProp:231:getDensity 21.61% (0.36%) 1316×
THM_waterProp:461:getHfg 6.56% (0.04%) 375×
THM_waterProp:251:getQuality 10.08% (0.08%) 375×
THM_linalg:84:resolveInversion 9.45% (0.04%) 8×
THM_convection:774:compute_T_surf 5.69% (0.02%) 1×
1.33% 750×
22.09% 1316×
4.25% 375×
6.80% 375×
4.93% 8×
4.47% 8×
0.59% 75×
THM_waterProp:499:getPhi2Phi 1.33% (0.03%) 750×
0.25% 300×
0.50% 300×
THM_waterProp:540:getReynoldsNumber 22.09% (0.23%) 1316×
9.93% 1316×
THM_waterProp:552:getReynoldsNumberLiquid 7.36% (0.08%) 675×
21.25% 2632×
6.52% 750×
THM_waterProp:534:getPhasesEnthalpy 6.80% (0.04%) 375×
3.21% 375×
THM_linalg:128:preconditionner 4.93% (4.92%) 8×
linalg:329:solve 4.50% (4.48%) 83×
5.07% 150×
THM_convection:823:getReynoldsNumber 0.59% (0.01%) 75×
1.30% 490×
21.85% 2632×
5.99% 675×
6.76% 750×
0.58% 75×
THM_waterProp:581:lockhartMartinelli 1.30% (0.06%) 490×
1.29% 675×
iapws97:4327:__init__ 83.69% (0.44%) 9448×
0.39% 490×
0.85% 490×
THM_waterProp:575:getUg 0.64% (0.12%) 790×
THM_waterProp:566:getUl 2.64% (0.39%) 1465×

Figure 6.12 $\mathtt{THM}_p$ profiling

The computational analysis of the solver highlighted the key processes that consume most of the execution time and are summarised in Table 6.6. The total execution time measured is **6.56 seconds**.

Process	**Time (s)**	**Proportion of total time**
Total	6.56	100%
IAPWS97	5.49	83.69 %
Linear Algebra (linalg)	0.619	9.45 %
Presure drop (getAreas)	0.0899	1.37 %
Velocities (getUl, getUg)	0.215	0.033 %

Table 6.6 Main computational bottleneck

The most significant bottleneck is the IAPWS97 module, which enables the solver to find the thermodynamic properties of the fluid and vapour phases. It is responsible for **83.69%** of the runtime. Optimising this module could lead to a strong improvement in performance.

The second largest contributor is the linear algebra module developed for the solver, which accounts for **9.45%** of the total time. This module uses numpy to invert and manipulate the matrix. It can also perform LU decomposition for the preconditioning (4.93 % of the total time) and resolution algorithms (4.50 % of the total time) such as the BICG, Gauss-Siedel or BICGStab. It is called at each iteration in the `resolveDFM` function. The main improvement cannot be at this level, even though milliseconds can be saved. The preconditioning step can be switched off for very stabilised and simple cases.

To determine the pressure drop modelled in the computation of the area, the program calls the `getAreas` function. It consumes **1.37%** of the runtime.

6.4 Discussion

This part synthesises the key findings of this study, evaluates the validity of the implemented model, and discusses the implications of the main hypothesis.

The key achievements are in two forms: the solver's capabilities to model essential phenomena with a relatively good accuracy have been ameliorated, and the solver is now capable of computing a pressure drop along the channel. These achievements are rare in the open-source community. Even if some strong hypotheses have been made, the solver capabilities showed significant improvement compared to the old one and excellent results against experimental benchmark and numerical simulation.

However, the validity of the results is constrained by several assumptions that deserve to be discussed:

- **The drift flux model** uses a homogeneous approach, which can be problematic. The slip ratio is empirically modelled. It neglects local interfacial momentum, mass and energy transfer. Every field is averaged in a section, ignoring the radial variability of the fields in a section, such as in fuel bundles or in a reactor core.
- **The absence of a subcooling boiling model** delays the vapour generation initiation compared to GeN-Foam. A simplified boiling model has been implemented in `THM:` and needs to be improved.
- **The equal pressure** for the liquid and the vapour phases neglects various phase transitions, cavitation, capillarity and rapid transients effects.
- **Kinetic and potential energy** terms are neglected, which is reasonable for low-Mach numbers.
- **Pressure drop** modelled by the friction factor and the two-phase multiplier is challenging due to the lack of an experimental or numerical model to forecast the two-phase friction on the wall along the channel. The Churchill [34] and Lockhart-Martinelli [36] correlation are a good beginning.

Concerning the bubble size modelling discussed in section 2.2.3, our results allow us to question the approximations made. To model the drift velocity through correlation, we assumed that the flow was average in the slug regime. This approximation was used to model drift velocity as a function of hydraulic diameter. This avoids the need to calculate bubble size for each control volume and each iteration. It makes the model less complex because the bubble size calculation can be very computationally demanding, depending on the flow's nature. Bubble size can be required not only for the drift velocity, but also for the pressure drop modelling and the heat transfer at the wall.

CHAPTER 7 Multiphysics modelisation

This following section is an article written by Raphael Guasch, Alain Hébert, Cédric Beguin, Guy Marleau and myself. The title is **Coupled Neutronics and Thermal Hydraulics Simulations of a BWR Fuel Channel in the Version5 Environment** and published for the M&C 2025 conference.

The abstract is: Modeling Boiling Water Reactors (BWRs) in deterministic neutron transport codes presents significant challenges due to the substantial void fraction introduced in the coolant. This strong interdependence between neutron transport and thermal-hydraulics motivates the development of coupled multi-physics calculation schemes. This work presents progress made at Ecole Polytechnique de Montréal (EPM) in the modeling of BWRs in the Version5 environment. The DONJON5 full core solver is used to obtain an axial power distribution. The latter is then provided to our open source thermal-hydraulics (TH) solver. A single BWR fuel rod, both used as an equivalent full core and fuel channel, is selected as a simple comparison test case. Neutron transport and thermal-hydraulics calculations are compared to Serpent2 and GeN-Foam respectively. Work done towards numerically validating the newly implemented TH solver and the multi-physics iterative procedure is presented.

7.1 Introduction

The strong multi-physics coupling observed in Boiling Water Reactors (BWRs) requires calculation routes capable of describing coupled thermal-hydraulics and neutron transport phenomena. Indeed, the axial voiding of the cooling water induces significant changes in the neutron transport properties [44]. Most notably, the coolant density decreases, causing a lack of thermalization of the neutron flux. This induces a loss of reactivity in the upper part of the fuel channel. The axial variation in the fuel effective temperature impacts the resonant absorption of neutrons due to Doppler broadening [45], also leading to a loss of reactivity in the upper section of the core.

The current `THM:` [46] module of DONJON5 [9] has historically been developed to model CANDU and PWR designs. It is therefore not well suited to model two-phase flows encountered in BWRs. In this work, we present our new `THM prototype`, aimed at extending the capabilities of DONJON5 in terms of BWR thermal-hydraulics modeling. Our `THM prototype` solver consists of a stand-alone open-source Python3 program that can be found on the PolyBWR GitHub page. The open-source character of our `THM prototype` makes it a good candidate for future implementation in the DONJON5 full core code.

The presented solution is based on a drift flux model, which allows for the treatment of the convective heat transfer along a fuel channel. Additionally, the radial temperature distribution in the fuel and clad is solved for, providing us with an effective fuel temperature to be used in the neutronics solution.

The thermal-hydraulics properties obtained with `THM prototype` can in turn be used to interpolate microscopic cross sections, used in the full core DONJON5 code to iteratively refine our estimate for the axial power profile.

Code-to-code comparisons are performed in order to assess the performances of our proposed coupled calculations route. The resulting axial power density of DONJON5 is compared to a reference Serpent2 [47] solution. The DONJON5 power distributions are exported, allowing for a `THM prototype`-GeN-Foam [48] comparison. The numerical stability and convergence of the method are assessed by performing a spatial convergence study on the axial meshing parameters.

7.2 Theory

The `THM prototype` is composed of two sub-modules, the first is responsible for solving the steady-state convective heat transfer problem at the coolant/clad interface. This includes solving for velocity, pressure, coolant temperature and void fraction along the channel. The second sub-module is used to solve for the thermal conduction in the cladding and the fuel. This provides a fuel temperature distribution, which can be used to estimate the effective fuel temperature for neutron transport applications.

7.2.1 Convection in the coolant

This new version of the module implements a homogeneous fluid, three-field representation (mixture velocity u_m, channel pressure p, and mixture enthalpy h_m) of two-phase flows. This enables it to model boiling and vapor formation in BWR channels. The `THM prototype` allows for the simulation of a 1D square or cylindrical channel flow. Each channel is characterized by 1D convection equations. The equations of mass, momentum and energy conservation are solved using an homogeneous drift flux model [28].

The mixture continuity, momentum and enthalpy-energy equations can be written as the following system of equations:

$$\frac{\partial}{\partial z}\left(\rho_m u_m\right) = 0 \tag{7.1}$$

$$\frac{\partial}{\partial z}\left[\rho_m u_m^2 + \frac{\varepsilon}{1-\varepsilon}\,\frac{\rho_\nu\,\rho_\ell}{\rho_m} V_{\nu j}^2\right] + \frac{\partial}{\partial z}\,p + \rho_m g = -\frac{4\tau_w}{D_h} \tag{7.2}$$

$$\frac{\partial}{\partial z}\left[\rho_m u_m h_m\right] + \frac{\partial}{\partial z}\left[\varepsilon \frac{\rho_v \rho_\ell}{\rho_m} V_{vj}(h_v - h_\ell)\right] = \frac{\partial}{\partial z} p \cdot \left[u_m + \frac{\varepsilon(\rho_l - \rho_g)}{\rho_m} V_{vj}\right] + \vec{n}_w \cdot \vec{q}_w^{\,\prime\prime} \tag{7.3}$$

To close the system of equations, constitutive relations are needed to model the thermal behavior of the flow. The `THM prototype` implements correlations for void fraction, friction losses, drift velocity and equation of states to determine the density and other steam/water properties. The void fraction ε is determined using the standard drift flux model equation based on the flow quality:

$$\varepsilon = \frac{x}{C_0\Big(x + \frac{\rho_g}{\rho_\ell}(1-x)\Big) + \frac{\rho_g V_{vj}}{\rho_m u_m}} \tag{7.4}$$

The void distribution parameter and the drift velocity are computed using the Electric Power Research Institute (EPRI) void model used in the PATHS/PARCS code [1].

To solve equations 7.1, 7.2 and 7.3, a finite difference method is used and the equations are cast into a face-based scheme which is motivated by the use of the inlet and outlet Dirichlet conditions. This reduces the problem to a 1D flow with fixed inlet mass flow rate, inlet temperature and outlet pressure. First, the mass and momentum equation are solved simultaneously across all nodes using the preconditioned Bi-conjugate Gradient (BiCG) method. Then, the energy conservation equation is solved using the updated velocity and pressure fields. As the inlet pressure is determined at each iteration and the flow rate and the temperature remains constant, the inlet velocity and inlet enthalpy are adjusted. This step ensures a consistent pressure drop along the channel. The new velocity, pressure and enthalpy fields are then used to compute state variables and drift flux parameters. An iterative process is used due to the coupling between the void fraction correlation and the density of the mixture.

7.2.2 Conduction in the fuel pin

The Fourier law of heat conduction models the temperature distribution in the fuel rod. A radial symmetry of the temperature distribution is assumed. The fuel rod is meshed according to a *constant surface* discretization. The heat conduction equation is solved using a mesh-centered finite difference method (MCFD). Boundary conditions at the fuel surface and gap/cladding surface ensure continuity of the heat flux. Due to the coupling between the surface heat exchange coefficient and the coolant temperature, the latter is iteratively updated until convergence is reached. Further details of the method are referenced in Héberts's documentation [46]. As stated earlier, the conduction solver implemented in our `THM prototype` is limited to constant, temperature independent thermal transfer properties presented in Table 7.1. Once the temperature distribution has been obtained, the effective fuel temperature T_F is computed using Rowland's formula [46].

7.3 Methodology

Homogenized and condensed cross sections are obtained using the DRAGON5 lattice code. The DONJON5 finite core code is used to perform diffusion calculations on a single fuel channel. The neutron transport properties are assumed to be homogeneous in each axial node. As the coolant void fraction, density and fuel effective temperature vary, a *linear* interpolation of microscopic cross sections is performed, associating new material properties to each fuel bundle. The temperature and density values used in the diffusion solution are obtained via our `THM prototype`. The coupling between neutronics and TH fields is ensured through a *PyGan* script described in section 7.3.4. DONJON5 to Serpent2 and `THM prototype` to GeN-Foam comparisons are performed. Numerical validation criteria are set.

7.3.1 Problem specifications

The geometry of the problem is that of a single ATRIUM-10 pin cell [16]. At the lattice calculation step, the fuel pellet is radially divided into four radial regions to account for spatial self-shielding effects. Each of these represents 50, 30, 15 and 5 % of the total fuel volume. The fuel composition is UOX at 2.4% enrichment.

BWR core representative data has been obtained from [49]. The selected user input geometrical, thermal hydraulics and heat transfer properties are shown in Table 7.1. The total rod height is taken from a GeN-Foam test case [25]. It is worth noting that the thermal conductivity values for the fuel, clad and the convective heat transfer coefficient at the gap are set. The tests are carried out for total powers of $P_{tot} = 10$ and $35kW$.

Table 7.1 Specifications of the case studied : geometric features, thermal-hydraulic data, and heat transfer properties.

<table>
<tr><th colspan="4">Specifications of the studied case</th></tr>
<tr><th>Thermal-hydraulic properties</th><th>Value</th><th>Geometric data</th><th>Value</th></tr>
<tr><td>Inlet temperature (K)</td><td>$T_{in} = 543.15$</td><td>Channel pitch (cm)</td><td>$p_c = 1.295$</td></tr>
<tr><td>Outlet pressure (MPa)</td><td>$P_{out} = 7.2$</td><td>Fuel radius (cm)</td><td>$r_f = 0.4435$</td></tr>
<tr><td>Mass flow rate (kg/s)</td><td>$\dot{m} = 8.407 \times 10^{-2}$</td><td>Cladding radius (cm)</td><td>$r_c = 0.514$</td></tr>
<tr><td>Hydraulic diameter (cm)</td><td>$D_h = 1.049$</td><td>Channel length (m)</td><td>$h = 1.555$</td></tr>
<tr><th colspan="4">Heat transfer properties</th></tr>
<tr><td colspan="4">Fuel thermal conductivity (W/mK): $k_{fuel} = 4.18$
Gap heat exchange coefficient (W/m^2K): $H_{gap} = 10000$
Cladding thermal conductivity (W/mK): $k_{clad} = 21.5$</td></tr>
</table>

7.3.2 Generating microscopic cross sections : the DRAGON5 lattice code

A two-steps lattice scheme is used to obtain spatially homogenized and group condensed microscopic cross sections using the SHEM [50] 295 energy groups based on the JEFF3.1.1 library. The 2 groups condensed and homogenized cross sections are tabulated as a function of three parameters: effective fuel temperature T_F, coolant temperature T_m and coolant density ρ_m. For every tabulation point, a self-shielding calculation is performed using the newly implemented RSE method [51]. The main flux calculation is performed using the method of characteristics (MOC). A specular reflective boundary condition is enforced. It is worth noting that at this stage, no leakage or reflector models have been introduced. Table 7.2 shows the T_F, T_m and ρ_m values for which microscopic cross sections were obtained and indexed in the `MULTICOMPO` object.

Table 7.2 Tabulation points for T_F, T_m, ρ_m, used in lattice calculations.

Parameter	List of tabulation points
Effective fuel temperature T_F (K)	500, 700, 900, 1100, 1300, 1500, 1800
Coolant temperature T_m (K)	500, 540, 550, 560, 570
Coolant density ρ_m (kg/m^3)	10, 100, 200, 300, 400, 500, 600, 700, 800

7.3.3 Full core diffusion calculations in DONJON5

The equivalent "full core" geometry is defined in the DONJON5 code. Reflective boundary conditions are imposed at the $\pm X$ and $\pm Y$ limits. Void boundary conditions are enforced at the top and bottom Z surfaces. The TRIVAC5 [14] finite elements solver is used to perform a two-groups flux calculation in the diffusion theory. Cross sections for each axial fuel slice are obtained by linearly interpolating from the `MULTICOMPO` database produced by DRAGON5.

7.3.4 Steady state coupled neutronics-TH loop

Our algorithm aims at computing fields representative of the coupled neutronics-TH system. For each axial node, it computes the local power, fuel effective temperature, coolant temperature and density. The void fractions and power densities are also retrieved in each node along the axial dimension.

The implementation makes use of the Python3 *LCM* and *CLE-2000* APIs available through GANLIB Version5 [52]. The latter allows for the manipulation of DONJON5 procedures, *LCM* data structures as well as our `THM prototype` from a unique Python3 procedure.

The proposed coupling procedure is schematized as shown in Figure 7.1.

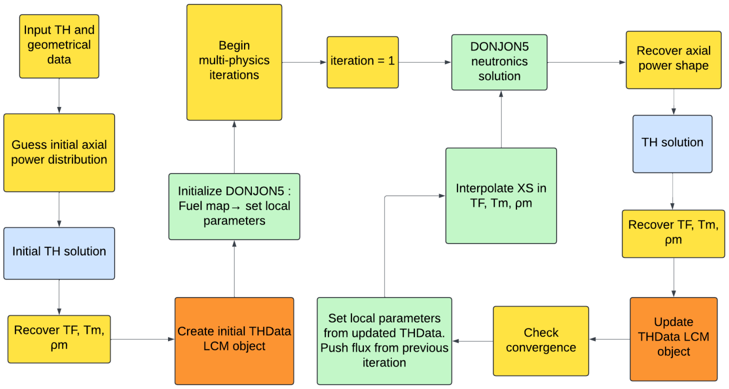

Input TH and geometrical data
Guess initial axial power distribution
Initial TH solution
Recover TF, Tm, ρm
Create initial THData LCM object
Initialize DONJON5 : Fuel map→ set local parameters
Begin multi-physics iterations
iteration = 1
DONJON5 neutronics solution
Recover axial power shape
TH solution
Recover TF, Tm, ρm
Update THData LCM object
Check convergence
Set local parameters from updated THData. Push flux from previous iteration
Interpolate XS in TF, Tm, ρm

Figure 7.1 Multi-physics coupling loop : data flow in *PyGan* procedure. Steps performed in the main Python3 procedure, calls to our `THM prototype`, to the *Python3 LCM* API and to procedures through the *Python3 CLE-2000* API are respectively indicated by yellow, blue, orange and green boxes.

As an initial guess, a sinusoidal power distribution is assumed along the axial dimension. This allows us to initialize TH fields which can in turn be used to initialize the neutronics solver. Microscopic cross sections stored in the `MULTICOMPO` are interpolated in T_F, T_m and ρ_m. Each axial node has an associated material *mixture* with unique neutron transport properties. A two-groups diffusion theory flux calculation is then performed by the FEM solver TRIVAC5 [14]. The computed power distribution is normalized to P_{tot}. Feeding this power distribution back into our `THM prototype` allows us to update the values of TH fields. This process is repeated iteratively until all convergence criteria on nodal power, T_F, T_m, ρ_m and k_{eff} are met. The convergence criteria for the coupled procedure are presented in Table 7.3.

Table 7.3 Convergence criteria set for multi-physics iterations

Quantity	**Convergence criteria**
Nodal power	0.01 %
k_{eff}	0.1 pcm
Nodal T_F	0.001 K
Nodal T_m	0.001 K
Nodal ρ_m	0.001 kg/m^3

7.3.5 Numerical verification and validation methodology

A first verification step is performed, checking the proper spatial convergence of our coupled calculation. Then, a code-to-code comparison is performed in an attempt to validate both our neutronics and thermal-hydraulics solutions. Converged fields are exported and used to run independent reference neutronics and thermal-hydraulics calculations.

The converged T_F, T_m and ρ_m fields are used to set-up an exact geometry 3D Serpent2 [47] model of the pincell. Detector scores for total heat production (MT=301) are extracted from each axial slice. In order to ensure statistical convergence, 10^4 batches of 5×10^4 neutrons are simulated, discarding the first 500 generations. To allow for a direct DONJON5-Serpent2 comparison of power distributions, the latter are normalized to the same total power used in the DONJON5 solution.

The converged power density distribution is also exported to set up an equivalent two-phase flow problem in GeN-Foam [48]. GeN-Foam is a computational tool based on OpenFoam, developed for reactor analysis. It specializes in the modeling of thermal-hydraulic processes using a two-fluid, three field representation of the two-phase flow.

The water temperature T_m, pressure P and void fraction ε are compared to GeN-Foam reference field using Root-Mean-Squared (RMS) deviations to determine their associated errors. Since GeN-Foam treats liquid and vapor phases distinctly, the equivalent mixture temperature is computed using : $T_m = \varepsilon T_v + (1 - \varepsilon T_l)$.

For this prototypal coupled calculation scheme to be considered valid we set the following criteria :

1. DONJON5 succeeds at providing a power density distribution with a RMS relative deviation (D5-S2) below 3%.

2. DONJON5 computes a k_{eff} value within a $\pm 150 pcm$ range of Serpent2.

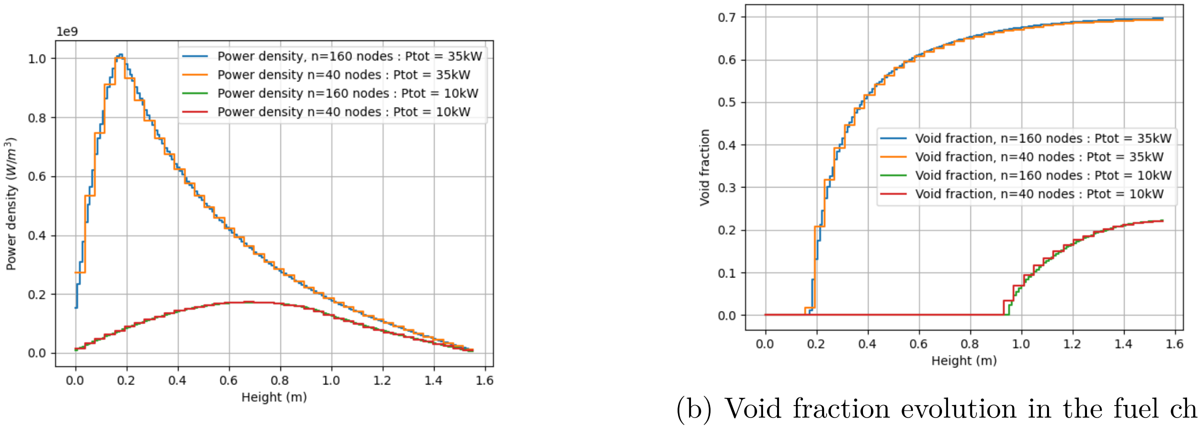
3. DONJON5 nodal errors on power distributions stay withing a ±5% interval from Serpent2.

4. `THM prototype` successfully provides void fraction and coolant temperature distributions with RMS (THM-GeNFoam) relative deviations below 5%.

7.4 Numerical results and analysis

7.4.1 Coupled neutronics-TH loop verification

Coupled neutronics-TH calculations are performed using the EPRI void fraction correlation [41], the Churchill [35] correlation for the friction factor and the Lockhart-Martinelli [36] correlation for the pressure drop. Total power normalizations to $10kW$ and $35kW$ are compared. The resulting power density, fuel temperature, coolant temperature, coolant density and void fraction distributions are studied for varying axial meshes. Figures 7.2a and 7.2b respectively show power density and void fraction distributions along the fuel channel. computed on meshes with 40 and 160 nodes.



(a) Axial power density distribution.

(b) Void fraction evolution in the fuel channel.

Figure 7.2 Power density (a), void fraction (b) evolution along the axial direction. Spatial convergence analysis for increasingly fine meshes. Power normalizations $10kW$ and $35kW$.

In order to assess the spatial convergence of our solution, TH fields computed on the coarser meshes are compared to those obtained on the finest 160 nodes mesh. Field values are associated to the mid points of control volumes, reference values are obtained by linearly interpolating between the 160 reference values. The RMS deviations on nodal error distributions are computed for fuel and coolant temperatures, coolant densities and void fractions. Table 7.4 displays the RMS deviations (Δ_{RMS}) evolution as the number of axial mesh elements increases. The spatial convergence of our solution acts as a verification of the self-consistency of the method.

Table 7.4 Dependence of Δ_{RMS} on axial mesh discretisation for 10 kW and 35 kW power normalizations.

Axial mesh	Δ_{RMS} T_F (K)		Δ_{RMS} T_m (K)		Δ_{RMS} ρ_m (kg/m^3)		Δ_{RMS} ε (%)	
	10 kW	35 kW	10 kW	35 kW	10 kW	35 kW	10 kW	35 kW
10	1.12	13.6	0.59	1.56	6.15	38.1	8.80	5.44
20	1.21	6.48	0.41	0.38	4.13	8.36	5.91	1.19
40	0.71	3.60	0.21	0.19	3.02	3.01	4.33	0.43
70	0.32	2.37	0.10	0.09	1.26	1.59	1.79	0.23
80	0.23	1.98	0.07	0.07	0.85	1.79	1.22	0.25

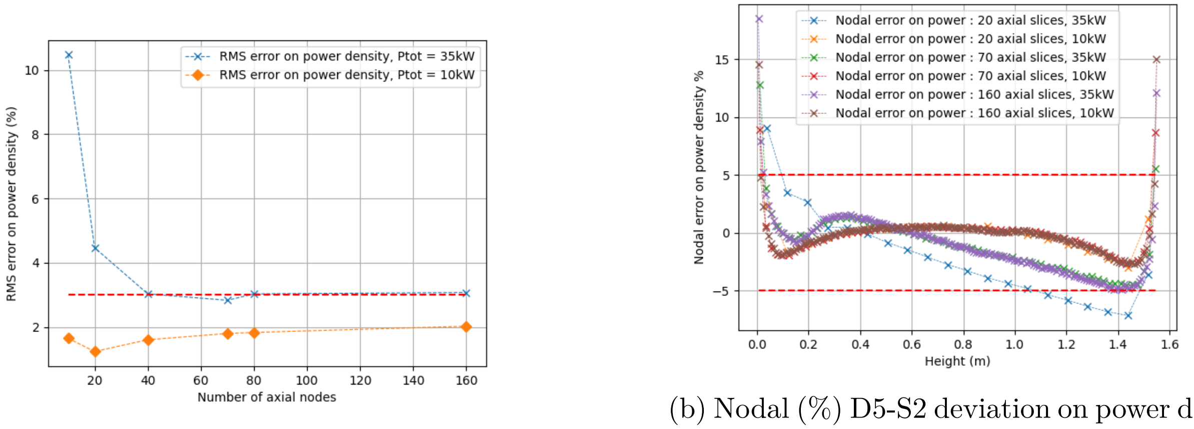
7.4.2 Validation of the neutronics solution

Table 7.5 Evolution of Δk_{eff} (pcm) with increasing number of axial slices for power levels 10kW and 35kW.

Number of Axial Nodes	Δk_{eff} **(pcm) at 10kW**	Δk_{eff} **(pcm) at 35kW**
10	97	406
20	79	148
40	69	87
70	75	71
80	71	73
160	68	71

The D5-S2 deviations on k_{eff} are shown in Table 7.5. RMS and nodal D5-S2 deviations on normalized power distributions are plotted in Figures 7.3a and 7.3b. The horizontal red lines indicate criteria set for a DONJON5 calculation to be validated. As seen in Table 7.4, the agreement of $\pm 150pcm$ on Δk_{eff} is reached for all meshes for $P_{tot} = 10kW$ and for all meshes with $n \geq 20$ for $P_{tot} = 35kW$. It can be seen from 7.3a that cases with $n > 40$ axial nodes successfully meet the 3% requirement set for the Δ_{RMS} on the power density distribution. Furthermore, Figure 7.3b shows that except for nodes close to the $\pm Z$ boundaries, the nodal errors for axial meshes $n = 70$ and $n = 160$ meet the $\pm 5\%$ criteria on D5-S2 deviations. The over estimation of D5 rates at $\pm Z$ boundaries can be explained by the choice of void boundary conditions. The latter induce a steep gradient in the neutron flux, which introduces numerical approximations in the FEM implementation of the "zero re-entrant angular flux" [53] condition. Additionally, the low neutron flux in such regions leads to more important variances associated with S2 scores tallied close to the boundary, which contributes to the increase in nodal errors close to the $\pm Z$ boundaries. It is hypothesized that this effect could be mitigated by introducing an axial reflector model. It should be noted that the error distributions for the $P_{tot} = 35kW$ case depend on z. This is explained by neutron axial streaming effects due to the more important void fractions in the upper sections of the channel.

The D5-S2 comparison shows that the presence of important void fractions requires a fine axial mesh to properly represent axial power variations. The $n = 70$ axial mesh is therefore selected for the `THM prototype` - GeN-Foam comparison.



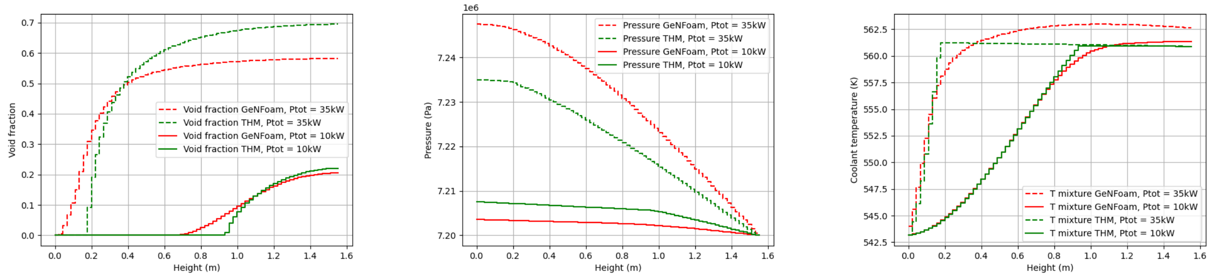
(a) D5-S2 Δ_{RMS} (%) on power density. Cases for $n = 10, 20, 40, 50, 70, 80$ and 160 axial nodes are presented.

(b) Nodal (%) D5-S2 deviation on power density distributions. Cases for $n = 20, 70, 160$ axial nodes and $P_{tot} = 10kW$, $35kW$ are presented.

Figure 7.3 D5-S2 RMS deviation between power density distributions (a) and axial evolution of nodal relative deviation on power densities (b).

7.4.3 Comparison of `THM prototype` solution to GeN-Foam

Figure 7.4 presents comparisons of our `THM prototype` to the GeN-Foam solution for cases with $P_{tot} = 10kW$ and $35kW$. The axial evolutions of the void fractions, pressures, and coolant temperatures are compared. It is important to note that GeN-Foam offers a two-phase description of the flow that inherently introduces differences with respect to our equivalent `THM prototype` mixture.



(a) Void fraction. (b) Pressure. (c) Coolant temperature.

Figure 7.4 Void fraction (a), pressure (b), and coolant temperature (c) evolution along the axial dimension. Full and dashed lines respectively correspond to the $P_{tot} = 10kW$ and $P_{tot} = 35kW$ cases. The green and red colors are associated to the `THM prototype` and to the GeN-Foam solutions respectively.

Table F.2 shows the RMS (Δ_{RMS}), the average (Δ_{avg}) and maximum (Δ_{max}) deviations obtained when comparing `THM prototype` and GeN-Foam distributions obtained for ε and p and T_m. The errors on the predicted pressure drops are $4kPa$ and $-11kPa$ for the $10kW$ and $35kW$ cases respectively.

Table 7.6 Absolute RMS, maximum, and average `THM prototype`-GeNFoam deviations on pressure, void fraction and coolant temperature at power levels 10kW and 35kW.

Physical quantity	Δ_{RMS}		Δ_{max}		Δ_{avg}	
	10kW	35kW	10kW	35kW	10kW	35kW
Pressure (kPa)	3.07	9.55	4.00	12.5	2.85	8.77
Void fraction	0.024	0.101	0.063	0.264	0.009	0.088
Coolant temperature (K)	0.341	1.63	1.13	3.15	0.225	1.52

`THM`$_{\texttt{prototype}}$ successfully predicts the coolant temperature in the $P_{tot} = 10$ and $35kW$ cases as depicted in Figure 7.4c. The axial evolution of the void fraction is also fairly satisfactory, especially when considering low power levels. It can be noted from Figure 7.4a that `THM prototype` consistently underestimates the void fraction in the lower parts of the core. On the other hand, the void fraction is over estimated in the upper section of the channel. This hints at the necessity to implement a sub-cooled boiling model to treat the initial bubble formation. Looking at Figure 7.4b it can be seen that the `THM prototype` fails to accurately predict the axial evolution of pressure, specifically in the $P_{tot} = 10kW$ case. Indeed, `THM prototype` pressure drop is over-estimated by a factor of 2. The cause of these discrepancies regarding pressure evolution is currently being investigated.

7.5 Conclusions

This work presents developments made at EPM towards the modeling of BWRs in the Version5 environment. The newly implemented **open-source** `THM prototype`, based on a drift flux model, is presented. It is coupled to the full core code DONJON5 through the *PyGan* interface. A single BWR pincell is modeled and studied as both an equivalent full core and coolant channel. The impact of the axial discretization and power normalization is assessed. An attempt at numerical validation is made by comparing to Serpent2 and GeN-Foam. The neutronics calculation is successfully validated with respect to Serpent2. However, the comparison between our `THM prototype` and GeN-Foam reveals that our drift-flux model fails to accurately predict axial variations in pressure. It also appears that our ebullition model could be improved, as the void fraction evolution is not entirely satisfactory when compared to GeN-Foam. This could be explained by the important differences between our homogeneous representation of the two-phase flow and the more elaborate CFD description implemented in GeN-Foam. Further work is needed to validate the `THM prototype` before it can be implemented in the open-source full core code DONJON5. Ongoing work involves validating `THM prototype` by studying the PSBT OECD/NEA benchmark, in addition to the ATRIUM-10-like assembly introduced in this work.

CHAPTER 8 Conclusion

8.1 Summary of Works

Most of the work made possible an explicit coupling between mass and momentum conservation equations (pressure - velocity coupling). It replaced the previous incremental velocity calculation using a matrix calculation, enhancing numerical stability and precision. New friction factor correlations, such as the Blasius and Churchill correlations, are provided. The solver showed excellent agreement with analytical solutions 6.2 and numerical solutions with the CFD code GeN-Foam (figure F.1c) with pressure drop errors within 5 kPa in most cases (see table 6.2).

The implementation of the drift flux model makes improvements to the capabilities of the solver to capture essential thermohydraulic phenomena. It especially focuses on fields important for neutronics simulation, such as void fraction, coolant temperature and density, for example, with void fraction deviations < 15 % (see table 6.2). The solver can also model the temperature at the centre of the fuel using the Fourier equation. The pressure drop decreases also the saturation temperature along the channel and delays boiling onset compared to constant pressure models. The evolution of the void fraction is more accurate using the drift flux model and makes the solver able to recover the phase's velocities.

8.2 Future Research

Key advancements should be pursued to improve further the solver's accuracy and applicability to real nuclear systems, such as fuel bundles and reactor cores. Future research aims to address current limitations and expand the model's capabilities to handle new geometry and new use cases.

As mentioned in the section 6, a critical improvement can be achieved by integrating a subcooled boiling model. Coupled with a wall boiling model, this improvement can model with high fidelity heated two-phase flow and be more accurate compared to CFD codes such as GeN-Foam. This model is particularly important for low-power conditions, where the actual model tends to underestimate the void fraction, and for high-power conditions, where the actual model tends to overestimate the void fraction.

The development of transient simulation capabilities could ameliorate the solver's ability to study dynamic phenomena such as flow instabilities, rapid transients, and multiphysics transient simulations. These capabilities would be very valuable for safety analyses and operational studies.

Finally, bridging the gap between the current 1D system code and the 2D or full 3D system code could be very interesting. The solver could also incorporate porosity-based approaches that enable coarse 2D/3D modelling of entire fuel assemblies or reactor cores. It would involve a discussion about keeping the drift flux modelling or switching the modelling to a two-fluid system.

REFERENCES

[1] A. Ward, A. Manera, T. Downar, Y. Xu, J. March-Leuba, C. Thurston, N. Hudson, A. Ireland, and A. Wysocki, “The modeling of advanced BWR fuel designs with the NRC fuel depletion codes PARCS/PATHS,” *Nuclear Technology*, vol. 190, no. 3, pp. 323–335, 6 2015.

[2] A. Tapucu, *The Thermal-Hydraulics of Two-Phase Systems.* Montréal: École Polytechnique de Montréal, 1 2009.

[3] Y. Taitel, D. Bornea, and A. E. Dukler, “Modelling flow pattern transitions for steady upward gas-liquid flow in vertical tubes,” *AIChE Journal*, vol. 26, no. 3, pp. 345–354, May 1980.

[4] “What is the role of nuclear in the energy mix and in reducing greenhouse gas emissions?” 12 2022, grantham Research Institute on Climate Change and the Environment. [Online]. Available: https://www.lse.ac.uk/granthaminstitute/explainers/role-nuclear-power-energy-mix-reducing-greenhouse-gas-emissions/

[5] R. de l’énergie du Canada, “Production d’Électricité nucléaire du canada,” 6 2023. [Online]. Available: https://www.rec-cer.gc.ca/fr/donnees-analyse/produits-base-energetiques/electricite/rapport/archive/energie-nucleaire-2018/production-delectricite-nucleaire-canada.html

[6] A. Ghione, “Development and validation of a two-phase CFD model using OpenFOAM,” 2012.

[7] A. Hébert, “Revisiting the simplified thermo-hydraulics module THM: in DONJON5 code,” École Polytechnique de Montréal, Technical Report, 2018.

[8] ——, “DRAGON5 and DONJON5, the contribution of École Polytechnique de Montréal to the SALOME platform,” *Annals of Nuclear Energy*, vol. 87, pp. 12–20, 2016, special Issue of The 3rd International Conference on Physics and Technology of Reactors and Application.

[9] A. Hébert, D. Sekki, and R. Chambon, “A user guide for DONJON version5,” École Polytechnique de Montréal, Technical Report IGE-344, 2023.

[10] International Atomic Energy Agency, "Development and application of open-source modelling and simulation tools for nuclear reactor analysis," IAEA, Technical Report, 2024.

[11] G. e. a. Giudicelli, "MOOSE: Enabling massively parallel multiphysics simulations," *SoftwareX*, vol. 26, p. 101690, 2024.

[12] Open Cascade SAS, *SALOME Platform User Guide*, 2023. [Online]. Available: https://www.salome-platform.org

[13] OpenFOAM Foundation, "OpenFOAM: Open source CFD software," 2024. [Online]. Available: https://openfoam.org/

[14] A. Hébert, "A user guide for TRIVAC version5," École Polytechnique de Montréal, Technical Report IGE-369, 2023.

[15] "OECD NEA/NSC: Working party on scientific issues of reactor systems (WPRS)." [Online]. Available: https://www.oecd-nea.org/science/wprs/egrsltb/BFBT/

[16] OECD Nuclear Energy Agency, *Physics of Plutonium Recycling.* OECD Publishing, 2003.

[17] R. Guash and C. Huet, "Coupled Neutronics and Thermal Hydraulics Simulations of a BWR Fuel Channel in the Version5 Environment," in *Mathematics and Computational Methods Applied to Nuclear Science and Engineering*, 2025.

[18] M. Hursin, P. Mala, A. Vasiliev, H. Ferroukhi, Y. Liu, S. Choi, and B. Kochunas, "Comparison of CASMO-5, MPACT and Serpent 2 for the modeling of advanced BWR lattices," in *Proceedings of the International Conference on Physics of Reactors (PHYSOR 2022)*, 2022.

[19] C. A. Gosdin, "Boiling water reactor simulations, models, and benchmarking using the thermal hydraulics sub-channel code CTF," Ph.D. dissertation, Pennsylvania State University, 2015. [Online]. Available: https://etda.libraries.psu.edu/catalog/27319

[20] M. Avramova, A. Velazquez-Lozada, and A. Rubin, "Comparative analysis of CTF and TRACE thermal-hydraulic codes using OECD/NRC PSBT benchmark void distribution database." [Online]. Available: https://inis.iaea.org/search/search.aspx?orig_q=RN:47071105

[21] F. Odar *et al.*, "TRACE/RELAP advanced computational engine (TRACE) V5.0 user manual," U.S. Nuclear Regulatory Commission, Technical Report, 2009.

[22] W. Wulff, “Critical review of conservation equations for two-phase flow in the U.S. NRC TRACE code,” *Nuclear Engineering and Design*, vol. 241, no. 10, pp. 4237–4260, 10 2011.

[23] V. J. Chavez, U. Imke, and V. Sanchez-Espinoza, “TWOPORFLOW: A two-phase flow porous media code, main features and validation with BWR-relevant bundle experiments,” *Nuclear Engineering and Design*, vol. 338, pp. 181–188, 8 2018.

[24] CEA, EDF, Framatome, IRSN, “CATHARE: Code for analysis of THermal-hydraulics during an accident of reactor and safety evaluation,” 2023. [Online]. Available: https://www.cathare-ds.com

[25] S. Radman, C. Fiorina, and A. Pautz, “Development of a novel two-phase flow solver for nuclear reactor analysis: Validation against sodium boiling experiments,” *Nuclear Engineering and Design*, vol. 384, p. 111422, 12 2021.

[26] C. Fiorina, I. Clifford, S. Kelm, and S. Lorenzi, “On the development of multi-physics tools for nuclear reactor analysis based on OpenFOAM: State of the art, lessons learned and perspectives,” *Nuclear Engineering and Design*, vol. 387, p. 111604, 2 2022.

[27] J. C. Slattery, “General balance equation for a phase interface,” *Industrial & Engineering Chemistry Fundamentals*, vol. 6, no. 1, pp. 108–115, 1967.

[28] M. Ishii and T. Hibiki, *Thermo-Fluid Dynamics of Two-Phase Flow.* New York: Springer, 2006.

[29] M. Fassi Fehri, “Développement d’un code de thermohydraulique pour le couplage neutronique/thermohydraulique,” Ph.D. dissertation, École Polytechnique de Montréal, 2008.

[30] P. Gallet, “Développement d’un module de thermohydraulique simplifiée dans le code DONJON4 pour l’Étude multiphysique des réacteurs à eau pressurisée,” Ph.D. dissertation, École Polytechnique de Montréal, 2014.

[31] R. W. Bowring, “Physical model, based on bubble detachment, and calculation of steam voidage in the sub-cooled region of a heated channel,” Institutt for Atomenergi (Norway). OECD Halden Reaktor Prosjekt, Technical Report, 1962.

[32] P. Saha and N. Zuber, “Point of net vapor generation and vapor void fraction in subcooled boiling,” in *International Heat Transfer Conference Digital Library.* Begel House Inc., 1974.

[33] N. E. Todreas and M. S. Kazimi, *Nuclear Systems I: Thermal Hydraulic Fundamentals*. New York: Taylor and Francis, 1990.

[34] S. W. Churchill, “Friction-factor equation spans all fluid-flow regimes,” *Chemical Engineering Journal*, 1977.

[35] B. Massey and J. Ward-Smith, *Mechanics of Fluids*, 8th ed. Taylor and Francis, 2006.

[36] R. W. Lockhart and R. C. Martinelli, “Proposed correlation of data for isothermal two-phase two-component flow in pipes,” *Chemical Engineering Progress*, vol. 45, 1949.

[37] S. Radman, “A coarse-mesh methodology for the analysis of one and two-phase nuclear reactor thermal-hydraulics in a multi-physics context,” Ph.D. dissertation, École Polytechnique Fédérale de Lausanne (EPFL), 7 2021.

[38] H. Müller-Steinhagen and K. Heck, “A simple friction pressure drop correlation for two-phase flow in pipes,” *Chemical Engineering and Processing: Process Intensification*, vol. 20, no. 6, pp. 297–308, 1986.

[39] V. N. Blinkov, O. I. Melikhov, V. I. Melikhov, A. V. Kapustin, K. S. Dolganov, D. Y. Tomashchik, V. N. Semenov, A. E. Tarasov, and S. S. Selkin, “Investigation on the interphase drag and wall friction in vertically oriented upward and downward two-phase flows under accident conditions in light water reactors,” *Nuclear Engineering and Design*, vol. 389, p. 111666, 2022.

[40] V. J. Chavez, U. Imke, and V. Sanchez-Espinoza, “TWOPORFLOW: A two-phase flow porous media code, main features and validation with BWR-relevant bundle experiments,” *Nuclear Engineering and Design*, vol. 338, pp. 181–188, 2018. [Online]. Available: https://doi.org/10.1016/j.nucengdes.2018.08.009

[41] B. Chexal and G. S. Lellouche, “A full-range drift-flux correlation for vertical flows,” Electric Power Research Institute, Palo Alto, CA, Technical Report NP-3989-SR, 1986.

[42] A. Manera, H. M. Prasser, and T. H. J. J. van der Hagen, “Suitability of drift-flux models, void fraction evolution and 3D flow pattern visualization during stationary and transient flashing flow in a vertical pipe,” *Nuclear Technology*, vol. 152, no. 1, pp. 38–53, October 2005.

[43] R. C. Martinelli and D. B. Nelson, “The prediction of pressure drop during forced circulation boiling of water,” *Transactions of the ASME*, pp. 695–702, 1948.

[44] H. Okuno, Y. Naito, and Y. Ando, "OECD/NEA burnup credit criticality benchmarks phase IIIA: Criticality calculations of BWR spent fuel assemblies in storage and transport," Japan Atomic Energy Research Institute, Technical Report, 2000.

[45] A. Hébert, *Applied Reactor Physics.* Presses internationales Polytechnique, 2009.

[46] ——, "Revisiting the simplified thermo-hydraulics module THM: in DONJON5 code," École Polytechnique de Montréal, Technical Report IGE-409, 2018.

[47] J. Leppänen, M. Pusa, T. Viitanen, V. Valtavirta, and T. Kaltiaisenaho, "The Serpent Monte Carlo code: Status, development and applications in 2013," *Annals of Nuclear Energy*, vol. 82, pp. 142–150, 2015.

[48] C. Fiorina, I. Clifford, M. Aufiero, and K. Mikityuk, "GeN-Foam: A novel OpenFOAM-based multi-physics solver for 2D/3D transient analysis of nuclear reactors," *Nuclear Engineering and Design*, vol. 294, pp. 24–37, 12 2015.

[49] Advanced Reactors Information System (ARIS), "Status report – BWRX-300 (GE hitachi and hitachi GE nuclear energy) USA," IAEA, Technical Report, 2019.

[50] A. Santamarina and N. Hfaiedh, "The SHEM energy mesh for accurate fuel depletion and BUC calculations," *Proceedings of ICNC2007*, 5 2007.

[51] A. Hébert, "A general formulation of the resonance spectrum expansion self-shielding method," *Nuclear Science and Engineering*, pp. 1–14, 2024.

[52] A. Hébert and R. Roy, "The GANLIB5 kernel guide (64-bit clean version)," École Polytechnique de Montréal, Technical Report IGE-332, 2023.

[53] G. Marleau, A. Hébert, and R. Roy, "A user guide for DRAGON version5," École Polytechnique de Montréal, Technical Report IGE-335, 2024.

[54] M. Lance and J. Bataille, "Turbulence in the liquid phase of a uniform bubbly air-water flow," *Journal of Fluid Mechanics*, vol. 222, pp. 95–118, 1991.

APPENDIX A VOID FRACTION CORRELATION

EPRI void model

The EPRI void correlation [41] is given by:

$$C_0 = \left[\kappa_0 + (1 - \kappa_0)\varepsilon^r \, \frac{1 - e^{-C_1}}{1 - e^{-C_1 \varepsilon}} \right]^{-1} \tag{A.1}$$

and for V_{gj}:

$$V_{gj} = \sqrt{2} \left(\frac{g\sigma \left(\rho_\ell - \rho_g\right)}{\rho_\ell^2} \right)^{1/4} (1 - \varepsilon)^{3/2} \tag{A.2}$$

where:

$$\kappa_1 = \min\left[0.8, \left(1 + e^{-10^{-0}\mathrm{Re}}\right)^{-e}\right] \tag{A.3}$$

$$\kappa_0 = \kappa_1 + (1 - \kappa_1) \left(\frac{\rho_g}{\rho_\ell} \right)^{0.2} \tag{A.4}$$

$$r = \frac{1 + 1.57 \left(\dfrac{\rho_g}{\rho_\ell} \right)}{1 - \kappa_1} \tag{A.5}$$

and

$$C_1 = \frac{4P_c^2}{P(P_c - P)} \tag{A.6}$$

GE-ramp

The GE-ramp [42] correlation is given by:

$$C_0 = \begin{cases} 1.1 & \varepsilon \leq 0.65 \\ 1 + 0.1\dfrac{1 - \varepsilon}{0.35} & \varepsilon > 0.65 \end{cases} \tag{A.7}$$

and

$$V_{gj} = \begin{cases} 2.9V_{gj0} & \varepsilon \leq 0.65 \\ 2.9\dfrac{1-\varepsilon}{0.35}V_{gj0} & \varepsilon > 0.65 \end{cases} \tag{A.8}$$

where

$$V_{gj0} = \left(\frac{g\sigma\left(\rho_\ell - \rho_g\right)}{\rho_\ell^2}\right)^{1/4} \tag{A.9}$$

Modified Bestion

The Modified Bestion correlation used in the same as in TRACE to take into account the interfacial drag [21], is given by:

$$C_0 = 1.2 - 0.2\sqrt{\frac{\rho_g}{\rho_l}} \tag{A.10}$$

and

$$V_{gj} = 0.188\sqrt{\frac{g\Big(\rho_l - \rho_g\Big)D_h}{\rho_g}} \tag{A.11}$$

The HEM model [33] is given by:

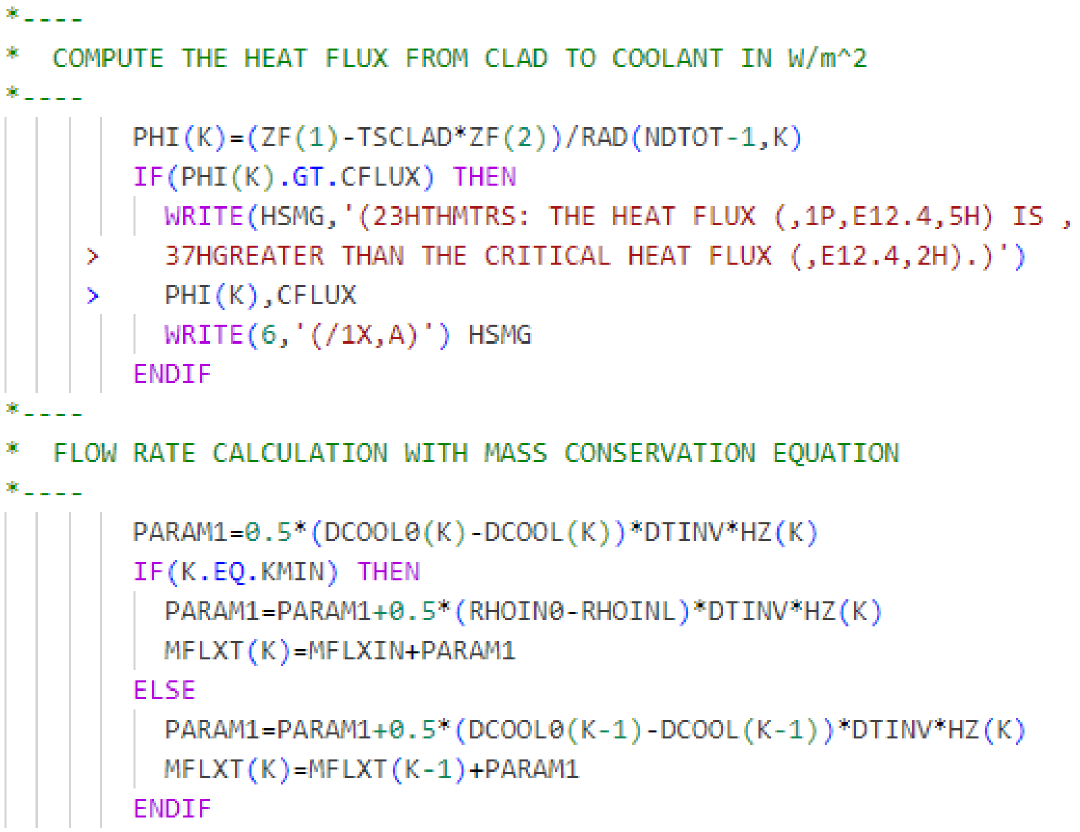
$$C_0 = 1 \tag{A.12}$$

and

$$V_{gj} = 0 \tag{A.13}$$

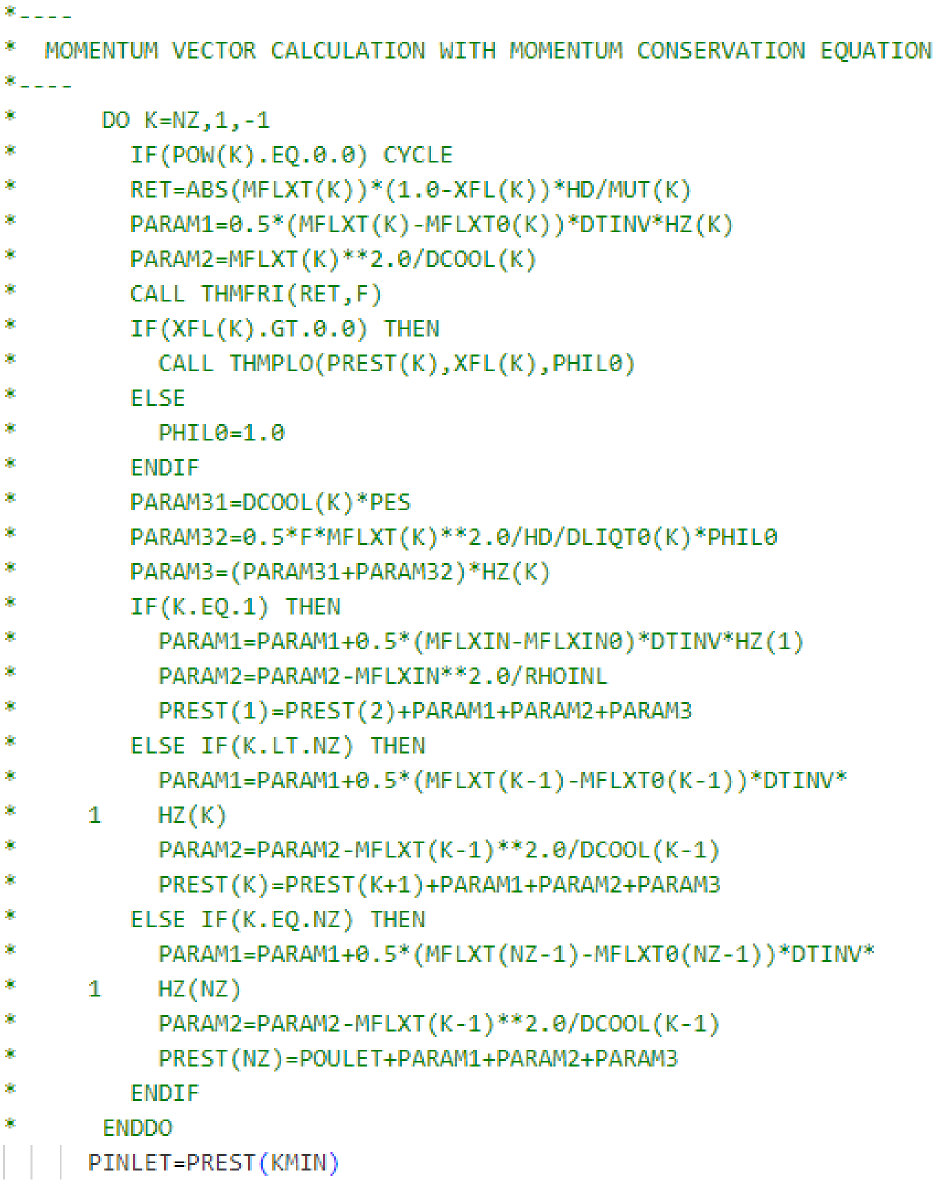
APPENDIX B MASS AND MOMENTUM CONCERVATION EQUATION CODE IN THE PREVIOUS THM: MODEL

The calculation of the momentum vector was initially commented out in the code due to a convergence failure encountered during execution. After further analysis and adjustments, this issue has now been resolved, ensuring accurate results. The correction involved refining the iterative algorithm, coupling the mass-momentum concervation equations and adding the drift flux model to the modelisation. Here is the code mentioned:



```
*----
*  COMPUTE THE HEAT FLUX FROM CLAD TO COOLANT IN W/m^2
*----
        PHI(K)=(ZF(1)-TSCLAD*ZF(2))/RAD(NDTOT-1,K)
        IF(PHI(K).GT.CFLUX) THEN
          WRITE(HSMG,'(23HTHMTRS: THE HEAT FLUX (,1P,E12.4,5H) IS ,
     >    37HGREATER THAN THE CRITICAL HEAT FLUX (,E12.4,2H).)')
     >    PHI(K),CFLUX
          WRITE(6,'(/1X,A)') HSMG
        ENDIF
*----
*  FLOW RATE CALCULATION WITH MASS CONSERVATION EQUATION
*----
        PARAM1=0.5*(DCOOL0(K)-DCOOL(K))*DTINV*HZ(K)
        IF(K.EQ.KMIN) THEN
          PARAM1=PARAM1+0.5*(RHOIN0-RHOINL)*DTINV*HZ(K)
          MFLXT(K)=MFLXIN+PARAM1
        ELSE
          PARAM1=PARAM1+0.5*(DCOOL0(K-1)-DCOOL(K-1))*DTINV*HZ(K)
          MFLXT(K)=MFLXT(K-1)+PARAM1
        ENDIF
```

Figure B.1 Code for mass conservation modelling



```
*----
*  MOMENTUM VECTOR CALCULATION WITH MOMENTUM CONSERVATION EQUATION
*----
*      DO K=NZ,1,-1
*        IF(POW(K).EQ.0.0) CYCLE
*        RET=ABS(MFLXT(K))*(1.0-XFL(K))*HD/MUT(K)
*        PARAM1=0.5*(MFLXT(K)-MFLXT0(K))*DTINV*HZ(K)
*        PARAM2=MFLXT(K)**2.0/DCOOL(K)
*        CALL THMFRI(RET,F)
*        IF(XFL(K).GT.0.0) THEN
*          CALL THMPLO(PREST(K),XFL(K),PHIL0)
*        ELSE
*          PHIL0=1.0
*        ENDIF
*        PARAM31=DCOOL(K)*PES
*        PARAM32=0.5*F*MFLXT(K)**2.0/HD/DLIQT0(K)*PHIL0
*        PARAM3=(PARAM31+PARAM32)*HZ(K)
*        IF(K.EQ.1) THEN
*          PARAM1=PARAM1+0.5*(MFLXIN-MFLXIN0)*DTINV*HZ(1)
*          PARAM2=PARAM2-MFLXIN**2.0/RHOINL
*          PREST(1)=PREST(2)+PARAM1+PARAM2+PARAM3
*        ELSE IF(K.LT.NZ) THEN
*          PARAM1=PARAM1+0.5*(MFLXT(K-1)-MFLXT0(K-1))*DTINV*
*     1    HZ(K)
*          PARAM2=PARAM2-MFLXT(K-1)**2.0/DCOOL(K-1)
*          PREST(K)=PREST(K+1)+PARAM1+PARAM2+PARAM3
*        ELSE IF(K.EQ.NZ) THEN
*          PARAM1=PARAM1+0.5*(MFLXT(NZ-1)-MFLXT0(NZ-1))*DTINV*
*     1    HZ(NZ)
*          PARAM2=PARAM2-MFLXT(K-1)**2.0/DCOOL(K-1)
*          PREST(NZ)=POULET+PARAM1+PARAM2+PARAM3
*        ENDIF
*      ENDDO
      PINLET=PREST(KMIN)
```

Figure B.2 Code for momentum conservation modelisation

APPENDIX C NUMERICAL DISCRETISATION FOR TRANSIENT

The following appendix is not necessary for this master's thesis. Nevertheless, it could prove valuable for future students and researchers seeking to enhance or adapt the model to a transient approach.

The steady-state formulation can be extended to a transient approach by discretising the temporal derivatives. We use a finite difference method based on Taylor series expansions while retaining the finite volume discretisation in space. For N time steps, time is discretized into Δt steps, with solutions computed at times $t^n = n\Delta t$ (where $n = 0, 1, \ldots, N$). We note discredited value of $\phi(z,t)$ as: $\phi(t^n, z_i) = \phi_i^n$. Considering the generalised transient conservation equation:

$$\frac{\partial}{\partial t}\phi + \nabla \cdot (\phi u) = 0 \tag{C.1}$$

the temporal scheme can take 3 forms:

- Explicit Euler (Forward Euler): The simplest scheme approximates the time derivative at time step n using a forward difference:

$$\frac{\partial \phi}{\partial t} \approx \frac{\phi_{i+\frac{1}{2}}^{n+1} - \phi_{i+\frac{1}{2}}^{n}}{\Delta t}, \tag{C.2}$$

  where Δt is the time step. The spatial terms $F(\phi)$ and $S(\phi)$ are evaluated at the current time n. This scheme is conditionally stable (requires a Courant-Friedrichs-Lewy (CFL) condition).

- Implicit Euler (Backward Euler): The time derivative is approximated similarly, but spatial terms are evaluated at the new time $n+1$:

$$\frac{\partial \phi}{\partial t} \approx \frac{\phi_{i+\frac{1}{2}}^{n+1} - \phi_{i+\frac{1}{2}}^{n}}{\Delta t}, \quad F(\phi)^{n+1}, S(\phi)^{n+1}. \tag{C.3}$$

  This scheme is unconditionally stable but requires solving a system of equations at each step.

- Crank-Nicolson (Semi-Implicit): A second-order accurate scheme averages spatial terms between time n and $n+1$:

$$\frac{\partial \phi}{\partial t}\Big|_{i+\frac{1}{2}} \approx \frac{\phi^{n+1}_{i+\frac{1}{2}} - \phi^{n}_{i+\frac{1}{2}}}{\Delta t}, \quad \frac{F(\phi)^n + F(\phi)^{n+1}}{2}. \tag{C.4}$$

Using the explicit Euler difference, the numerical discretization of equations becomes.

Mass concervation equation:

$$\frac{(\rho A)^{n+1}_{i+\frac{1}{2}} - (\rho A)^{n}_{i+\frac{1}{2}}}{\Delta t} + (\rho^* u A)^n_i - (\rho^* u A)^n_{i+1} = 0 \tag{C.5}$$

$$\begin{aligned}
&\frac{(\rho u A)^{n+1}_{i+\frac{1}{2}} - (\rho u A)^{n}_{i+\frac{1}{2}}}{\Delta t} + (\rho^* u^* A)^n_i u^n_i - (\rho^* u^* A)^n_{i+1} u^n_{i+1} + A^n_i P^n_i - A^n_{i+1} P^n_{i+1} \\
&= \left(\frac{\varepsilon^*}{1-\varepsilon^*}\frac{\rho^*_\ell \rho^*_g}{\rho^*}\left(V^*_{gj}\right)^2 A\right)^n_{i+1} - \left(\frac{\varepsilon^*}{1-\varepsilon^*}\frac{\rho^*_\ell \rho^*_g}{\rho^*}\left(V^*_{gj}\right)^2 A\right)^n_{i} \\
&- \left(\frac{\rho^n_{i+1}\Delta V^n_{i+1} + \rho^n_i \Delta V^n_i}{2}\right) g - \left(\frac{\Delta V \rho u^* f \phi_m}{2 D_h}\right)^n_{i+1} u^n_{i+1} + \left(\frac{\Delta V \rho u^* f \phi_m}{2 D_h}\right)_i u^n_i
\end{aligned} \tag{C.6}$$

if we note:

$$\hat{A}^*_i = \left(A + \frac{\Phi^*_m}{2}\frac{f^*}{D_h}\Delta V\right)_i \tag{C.7}$$

we can rewrite the equation C.8:

$$\begin{aligned}
&(\rho^* u^* \hat{A}^*)^n_i u_i - (\rho^* u^* \hat{A}^*)^n_{i+1} u^n_{i+1} + A_i P_i - A^n_{i+1} P^n_{i+1} = \left(\frac{\varepsilon^*}{1-\varepsilon^*}\frac{\rho^*_\ell \rho^*_g}{\rho^*}\left(V^*_{gj}\right)^2 A\right)^n_{i+1} \\
&- \left(\frac{\varepsilon^*}{1-\varepsilon^*}\frac{\rho^*_\ell \rho^*_g}{\rho^*}\left(V^*_{gj}\right)^2 A\right)^n_{i} - \left(\frac{\rho^n_{i+1}\Delta V^n_{i+1} + \rho^n_i \Delta V^n_i}{2}\right) g
\end{aligned} \tag{C.8}$$

The mixture enthalpy-energy equation:

$$\begin{aligned}
\frac{(\rho hA)^{n+1}_{i+\frac{1}{2}} - (\rho hA)^{n}_{i+\frac{1}{2}}}{\Delta t} \quad &+ \quad (\rho^* u^* A)_i h_i - (\rho^* u^* A)_{i+1} h_{i+1} \\
&= \quad q'''\Delta V - \left(\varepsilon^* \frac{\rho^*_\ell \rho^*_g}{\rho^*} \Delta h_{fg} V^*_{gj} A\right)_i + \left(\varepsilon^* \frac{\rho^*_\ell \rho^*_g}{\rho^*} \Delta h_{fg} V^*_{gj} A\right)_{i+1} \\
&+ \quad \frac{\left(u^* + \frac{\varepsilon^*(\rho^*_\ell - \rho^*_g)}{\rho^*} V^*_{gj}\right)_i + \left(u^* + \frac{\varepsilon^*(\rho^*_\ell - \rho^*_g)}{\rho^*} V^*_{gj}\right)_{i+1}}{2} \\
&\quad \cdot \left(P^*_i A_i - P^*_{i+1} A_{i+1}\right)
\end{aligned} \tag{C.9}$$

To calculate the volumetric heat power q''' transferred from the heating rod to the water, the steady-state approximation involves multiplying the rod's volumetric power by its volume and then dividing by the water volume:

$$q'''_{water} = q'''_{fuel} \frac{V_{fuel}}{V_{water}} \tag{C.10}$$

This assumes instantaneous and uniform heat distribution, which is valid in steady-state conditions where temperatures are stable. However, in transient regimes, this method fails because it neglects thermal inertia, dynamic heat transfer delays, and spatial temperature gradients. A transient analysis would instead require solving the time-dependent energy balance equations, accounting for heat capacity and transient conduction/convection effects.

APPENDIX D JUSTIFICATION FOR NEGLECTING THE SURFACE TENSION TERM $\vec{\kappa}$

In this appendix, we demonstrate why the surface tension term

$$\vec{\kappa} = \langle \gamma \, C_i \, \vec{n}_\ell^i \rangle,$$

is negligible in the momentum conservation equation for a one-dimensional (1D) flow in a reactor pipe. Here,

- γ denotes the surface tension coefficient,
- C_i represents the local curvature of the interface,
- $\vec{n}_\ell^i$ is the unit normal vector to the interface.

This term contributes to the pressure jump at the interface, analogous to the Laplace pressure jump. Our analysis focuses on a system operating at a pressure of approximately 10.8 MPa.

Surface Tension and the Laplace Pressure Jump

The pressure difference induced by surface tension is given by the Laplace formula:

$$\Delta P_{\text{capillary}} = \gamma \left(\frac{1}{R_1} + \frac{1}{R_2} \right).$$

In a simplified model where the interface is characterized by a single radius of curvature R (i.e., $C \approx \frac{1}{R}$), this expression can be approximated by

$$\Delta P_{\text{capillary}} \sim \frac{2\gamma}{R}.$$

Order-of-Magnitude Calculation

For our numerical estimation, we use the following typical values:

- Surface tension of water: $\gamma \sim 0.072$ N/m (note that even under high pressure, the order of magnitude remains similar).

- Two representative values for the radius of curvature:

  1. **Case 1:** $R \sim 1$ mm $= 10^{-3}$ m

  $$\Delta P_{\text{capillary}} \sim \frac{2 \times 0.072}{10^{-3}} \approx 144 \text{ Pa}.$$

  2. **Case 2 (worst-case scenario with smaller bubbles):** $R \sim 100$ μm $= 10^{-4}$ m

  $$\Delta P_{\text{capillary}} \sim \frac{2 \times 0.072}{10^{-4}} \approx 1440 \text{ Pa}.$$

Comparison with Operating Pressure

The operating pressure in the reactor is of the order:

$$P \sim 10.8 \text{ MPa} = 10.8 \times 10^6 \text{ Pa}.$$

Even in the worst-case scenario (Case 2), the ratio of the capillary pressure jump to the system pressure is:

$$\frac{\Delta P_{\text{capillary}}}{P} \sim \frac{1440 \text{ Pa}}{10.8 \times 10^6 \text{ Pa}} \approx 1.33 \times 10^{-4}.$$

This indicates that the contribution of the surface tension effect is less than 0.01% of the total pressure, which is negligible in comparison.

Conclusion

The order-of-magnitude analysis clearly shows that, under high operating pressures (approximately 10.8 MPa), the pressure difference induced by the surface tension is several orders of magnitude smaller than the system pressure. Hence, it is justified to neglect the surface tension term $\vec{\kappa}$ in the momentum conservation equation, simplifying the model without compromising the accuracy of the pressure drop prediction.

APPENDIX E TURBULENCE

Context

In the drift flux model used in this study, the effects of turbulence are already incorporated into the model formulation. In particular, the divergence of the Reynolds stress tensors for the gas and liquid phases

$$\nabla \cdot \left(\varepsilon_g \, \tau_g^{Re} + \varepsilon_\ell \, \tau_\ell^{Re}\right)$$

is represented by

$$\nabla \cdot \left(\varepsilon_g \, \tau_g^{Re} + \varepsilon_\ell \, \tau_\ell^{Re}\right) = -\frac{d}{dz}\left[\frac{\varepsilon_g}{\varepsilon_\ell}\frac{\rho_g \rho_\ell}{\rho_m} V_{gj}^2\right], \tag{E.1}$$

where ε_g and ε_ℓ are the gas and liquid volume fractions respectively, ρ_g and ρ_ℓ their corresponding densities, ρ_m is the mixture density, and V_{gj} represents the slip velocity between the phases.

Bubble-Induced Turbulence

The drift flux formulation take into accounts the turbulence by correlating it with the slip velocity between the phases. Especially in bubbly flows when the gas volume fraction exceeds approximately 1%, experimental studies have shown that the turbulence in the liquid phase is strongly influenced by bubble behavior. As demonstrated in the work of Lance and Bataille [54], the turbulence is found to be proportional to the square of the slip velocity.

These observations justify the drift flux model's representation of the turbulence term. Since the turbulence contribution is captured by a term proportional to V_{gj}^2, the expression

$$-\frac{d}{dz}\left[\frac{\varepsilon_g}{\varepsilon_\ell}\frac{\rho_g \rho_\ell}{\rho_m} V_{gj}^2\right]$$

Adequately represents the net effect of the turbulent stresses in a one-dimensional framework.

APPENDIX F VALIDATION USING THE POROUS MEDIA CODE TWOPORFLOW

The validation of THM: can be made using another calculation code called TwoPorFlow (TPF) [40]. TPF is a porous media two-phase flow code. It is based on the coupled Euler equations for both the liquid and vapour phases. It includes fiction with a solid medium. This code is maintained by the Karlsruhe Institute of Technology in Germany. The main capabilities and features of TPF are:

- Handling two phase flow problems in a reactor cores or in RPV
- Six conservation equations (3 for each phase)
- 3 dimension cartésian geometry
- Lots of model to close the system of equations
- Pretty flexible cartesian discretization mesh for different scale such as sub-channel or a fuel assembly
- Incorporate conduction equation resolution as in THM:

The porous approach and the definition of the porosity is similar to the approach mentioned in section 3. TPF use the same steam table as in THM:, and the correlation for the friction losses is pretty close to the one developed in THM:. The correlation used are the Blasius [35] or the Lockhart-Martinelli [36] for the friction factor and the Churchill [34] correlation for the two-phase multiplier.

The main difference is the decoupling between the liquid and the vapour phases, which goes from 3 to 6 equations. This type of modelling involves using new tools to model the transfer between phases. TPF uses wall heat transfer, liquid-vapour interface heat and mass transfer to close to a system of equations. These models are detailed in the theory manual [40].

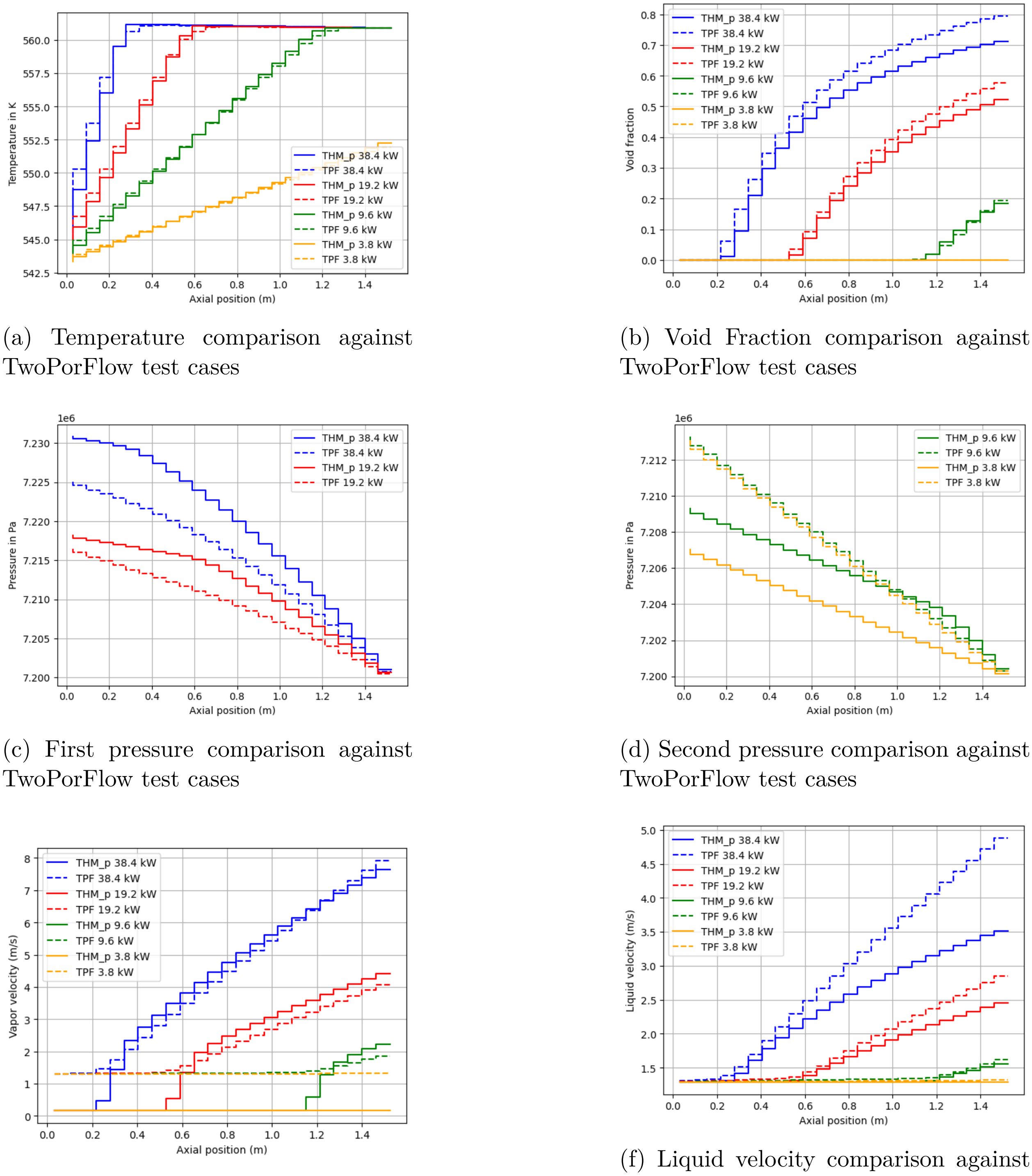
THM: is validated against reference codes mentioned 2.4 and the OECD NUPEC BWR Full-Size Fine-Mesh Bundle Test Benchamrk [15].

In terms of modellisation technique, this code seems closer to THM: than GeN-Foam. To compare both codes, the same geometry and thermohydraulic parameters as in section 6.2.1 and recall hereafter:

Geometric Parameters		Thermohydraulic Parameters	
Parameter	**Value**	**Parameter**	**Value**
Axial mesh points (I_z)	70	Power	3.8 to 38.4 kW
Fuel length (L_f)	1.555 m	Mass flux rate (q_{flow})	0.08407 kg/s
Fuel radius (r_f)	0.4435 cm	Outlet pressure (p_{out})	7.2 MPa
Clad radius (r_{clad})	0.5140 cm	Inlet temperature (T_{in})	543.15 K
Channel pitch (r_w)	1.295 cm	Inlet velocity (u_{in})	1.29 m/s

Table F.1 Geometric and Thermohydraulic Parameters for the TwoPorFlow test case

The results using the Churchill correlation for the friction factor and the Lockhart-Martinelli for the two-phase multiplier are shown below in the figure F.1:



(a) Temperature comparison against TwoPorFlow test cases

(b) Void Fraction comparison against TwoPorFlow test cases

(c) First pressure comparison against TwoPorFlow test cases

(d) Second pressure comparison against TwoPorFlow test cases

(e) Vapour velocity comparison against TwoPorFlow test cases using the drift flux equation to obtain de phases velocity from the mixture: $v_g = u_m + \frac{\rho_\ell}{\rho_m} \cdot V'_{gj}$

(f) Liquid velocity comparison against TwoPorFlow test cases using the drift flux equation to obtain de phases velocity from the mixture: $v_\ell = u_m - (\frac{\varepsilon}{1-\varepsilon}) \cdot \frac{\rho_g}{\rho_m} \cdot V'_{gj}$

Figure F.1 Temperature, void fraction, pressure and liquid velocity comparison against TwoPorFlow test case for 3.8, 9.6, 19.2, 38.4 kW fuel power

The comparison between the new modelisation of `THM:` and TwoPorFlow reveals excellent performance of the modified solver. Errors on void fraction are below 10% for every power as calculated in the following table:

Table F.2 RMS %, maximum % deviations on coolant temperature, void fraction, pressure and velocities at power levels 3.8, 9.6, 19.2, 38.4 kW

Power	Temperature		Void Fraction		Pressure		Liquid Velocity		Vapour Velocity	
	RMS	MAX	RMS	MAX	RMS	MAX	RMS	MAX	RMS	MAX
38.4 kW	0.107	0.312	6.070	8.218	0.067	0.095	16.241	28.065	38.013	87.086
19.2 kW	0.063	0.156	3.149	5.421	0.037	0.055	7.311	13.899	54.539	87.168
9.6 kW	0.040	0.079	0.447	1.418	0.027	0.055	2.430	4.024	76.985	87.216
3.8 kW	0.028	0.055	0.001	0.001	0.049	0.084	1.208	2.059	86.819	86.952

The error on void fraction seems to increase with the power as with GeN-Foam. This increase in the void fraction difference with the power is still lower than with GeN-Foam.

Some differences in the predicted thermal-hydraulic parameters across varying power levels (3.8 kW to 38.4 kW) can be notified. `TPF` generally predicts higher void fractions than `THM:`, especially at higher power levels (e.g., 38.4 kW). This suggests `TPF`'s models for vapour generation (e.g., nucleation, interfacial drag) may be more sensitive to heat flux or less conservative in suppressing vapour formation.

The higher vapour velocities in `TPF` align with this trend, indicating stronger vapour-phase momentum coupling. `THM:` shows lower liquid velocities compared to `TPF`, particularly at mid-to-high power (19.2-38.4 kW). This might reflect differences in wall friction models or turbulent mixing formulations. There is no turbulent mixing formulation in `THM:` contrary to `TPF`

The errors observed in vapour velocities should not be considered as reliable reference values. Both the RMS and maximum percentage deviations are exceptionally high due to an offset from the bottom of the channel between TPF and `THM:` when nucleation occurs (see Figure F.1e). In `THM:`, the vapour velocity is set to zero when no evaporation occurs, whereas TPF maintains a default velocity of 1.2 m/s even in the absence of vapour. This discrepancy in the treatment of zero-vapour conditions creates a significant artificial error in the velocity comparison

Both codes agree closely on coolant temperature trends. The pressure drop errors are of the same magnitude as the GeN-Foam comparison.