

A Coarse-mesh Methodology for the Analysis of One and Two-phase Nuclear Reactor Thermal-hydraulics in a Multi-physics Context

Présentée le 8 juillet 2021

Faculté des sciences de base

Laboratoire de physique des réacteurs et de comportement des systèmes

Programme doctoral en énergie

pour l'obtention du grade de Docteur ès Sciences

par

Stefan RADMAN

Acceptée sur proposition du jury

Prof. S. Haussener, présidente du jury

Prof. A. Pautz, C. Fiorina, directeurs de thèse

Dr S. Kelm, rapporteur

Dr P. Rubiolo, rapporteur

Prof. H.-M. Prasser, rapporteur

“I have quadrupled my flip power!” – Anakin Skywalker to Obi-Wan Kenobi

Acknowledgements

I would like to express my sincerest gratitude to my supervisor Prof. Dr. Andreas Pautz and co-supervisor Dr. Carlo Fiorina for their invaluable patience, support and guidance which were fundamental to shaping this PhD project.

My gratitude further extends to all the researchers at the Laboratory for Reactor Physics and Systems Behaviour (LRS), Dr. Mathieu Hursin, Dr. Vincent Lamirand and Dr. Pavel Frajtag as well as to all my friends and colleagues with whom I shared the path of doctoral studies: Oskari Pakari, Alessandro Scolaro, Fanny Vitullo and Daniel Siefman. These people greatly enriched each day, both inside and outside university.

Outside of the LRS, I would like to thank all the people and researchers that I have met and had an opportunity to collaborate with along the way. I would particularly like to thank Dr. Konstantin Mikityuk at the Paul Scherrer Institute (PSI), and Chirayu Batra at the International Atomic Energy Agency (IAEA), whose role was crucial, among others, in organizing a coordinated research project that proved beneficial to this thesis work as well.

I would further like to thank all the people that contributed to my academic and personal development in a way that allowed me to start a PhD in the first place.

On the academic side, special recognition is owed to my MSc thesis advisors Dr. Mathieu Hursin and Dr. Gregory Perret, as well all the professors at the EPFL and ETHZ that helped form me as a nuclear engineer, particularly Prof. Dr. Andreas Pautz and Prof. Dr. Horst-Michael Prasser. This recognition extends to my former professors at the Politecnico di Milano during my BSc studies in engineering physics, with special regards to Prof. Dr. Matteo Passoni for his energetic teaching approach that further strengthened my interest in nuclear science and technology. Further back in time, I want to acknowledge the very important role of my former professors at the “Paolo Frisi” high school in Monza, Italy, most importantly my physics professor Raffaella Scappatura and history and philosophy professor Enrico Colombo.

On the personal side, I could not be more grateful for all the friends I made along the way, with special regard to those in Italy who are still around in spite of years of doctoral hardships and the Alps in between us.

Last but not least, I am eternally grateful to my parents, Danica and Darko, for raising me to the best of their abilities and never failing to encourage my interest in science and engineering.

Abstract

The analysis of nuclear reactors for performance and safety assessment benefits from the use of computational tools. In this context, this work aims at the development and application of a thermal-hydraulics methodology and related software that respond to emerging needs in the computational field: 1) greater geometric and physics modelling flexibility; 2) streamlined coupling with other single-physics to enable multi-physics capabilities; 3) parallel scalability on High Performance Computing clusters; 4) adoption of modern programming practices.

A coarse-mesh approach is proposed to offer a reasonable balance between computational accuracy, comparable to that of sub-channel codes, and computational burdens. The developed approach can make use of general 3-D geometries with unstructured meshes, which are beneficial to the aforementioned geometric flexibility needs. Additionally, in a multi-physics context, field transfer operations between the different physics are simplified both by the adoption of a coarse-mesh approach and by the use of standardized mesh formats. The employed programming framework consists of the Finite Volume Method-based OpenFOAM library, which offers the desired features of massive parallel scalability and of a modern object-oriented programming paradigm.

The coarse-mesh methodology is presented alongside a thorough theoretical derivation of the governing equations for a generic multi-phase system. Based on this, a computer code is developed for the modelling of one-phase and two-phase flows, with a focus on the simulation of Sodium-cooled Fast Reactors (SFRs), which represent the nearest-term deployable fast reactor technology. These were also chosen as the simulation of phase change in sodium represents a challenging case for the numerical stability of two-phase solution algorithms.

The main achievements of this development effort consist of: 1) a novel solution algorithm for two-phase pressure-velocity coupling that enhances stability and performances compared to existing algorithms; 2) implementation of the code based on object-oriented programming practices, which allow for a seamless implementation of different working fluids and structure models; 3) code verification via an ad-hoc implementation of the Method of Manufactured Solutions; and 4) demonstrated good parallel scaling of the code up to thousands of computer cores. In terms of applications: 1) preliminary validation based on sodium boiling experiments; and 2) detailed investigation of existing and novel features for SFR fuel elements. Furthermore, the multi-physics capabilities of the developed methodology are demonstrated by integrating it within the GeN-Foam multi-physics environment. As a test case, the resulting software is applied to the simulation of a Loss Of Flow Without SCRAM test performed at the Fast Flux Test Facility. This benchmark re-analysis takes place within the framework of a coordinated research project by the International Atomic Energy Agency and is set to provide valuable feedback in terms of code-to-code comparison data.

Résumé

L’analyse des problématiques concernant la sûreté des réacteurs nucléaires nécessite l’utilisation d’outils de calcul. Dans ce contexte, ce travail vise à développer et appliquer une méthodologie thermo-hydraulique qui réponde aux besoins émergents dans le domaine du calcul: 1) une plus grande souplesse de modélisation géométrique et physique; 2) un couplage simplifié avec d’autres physiques pour atteindre des permettre la modélisation de phénomènes multi-physiques; 3) capacité de calcul parallèle sur infrastructures de calcul haute performance; 4) adoption de pratiques de programmation modernes.

Une approche “coarse-mesh” est proposée pour offrir un équilibre entre la précision de calcul, comparable à celle des codes “sub-channel”, et le coût du calcul lui-même. L’approche développée peut utiliser des géométries 3-D avec des maillages non-structurés, ce qui donne une grande flexibilité à l’outil numérique pour le traitement de la géométrie du problème considéré. En outre, dans un contexte multi-physique, les opérations de couplage entre les différentes physiques sont simplifiées par l’adoption de cette approche et par l’utilisation de formats de maillage standardisés. Le cadre de programmation consiste dans le logiciel OpenFOAM, qui offre les caractéristiques souhaitées en terme de programmation orientée objet.

La méthodologie “coarse-mesh” est présentée, ainsi qu’une dérivation des équations qui gouvernent un système générique à plusieurs phases. Sur cette base, un algorithme est développé pour la modélisation des systèmes mono-phasiques et di-phasiques, en particulier pour la simulation des Réacteurs à Neutrons Rapides à caloporteur sodium (RNR-Na), qui représentent une technologie de réacteur rapide déployable à court terme. Ce type de problème a également été choisis car le changement de phase du sodium représente un cas de modélisation difficile en terme de stabilité numérique des algorithmes.

Les principales réalisations de cet effort de développement sont: 1) un nouvel algorithme de solution pour le couplage pression-vitesse à deux phases qui améliore la stabilité et les performances par rapport aux algorithmes existants; 2) adoption de pratiques de programmation orientée objet qui permettent une mise en œuvre simplifié de différents fluides caloporteurs et modèles de structure; 3) vérification du code par une mise en œuvre ad-hoc de la Méthode des Solutions Manufacturées; 4) démonstration des bonnes capacités de calcul parallèle jusqu’à des milliers de processeurs. En termes d’applications: 1) le schéma de calcul est validé par comparaison avec des données expérimentales d’ébullition au sodium; 2) des caractéristiques proposées pour les éléments combustibles RNR-Na sont étudiées à titre de démonstration. En outre, la méthodologie est intégrée dans l’environnement multi-physique du code GeN-Foam et appliquée à la simulation d’un essai “Loss Of Flow Without SCRAM” effectué au Fast Flux Test Facility. Cette ré-analyse a lieu dans le cadre d’un activité de recherche coordonnée par l’Agence Internationale de l’Énergie Atomique et est destinée à fournir un retour d’information précieux en termes de données concernant la comparaison code à code.

Abstract

L’analisi dei reattori nucleari per la valutazione di performance e margini di sicurezza beneficia dell’utilizzo di strumenti computazionali. In tale contesto, questo lavoro punta allo sviluppo ed applicazione di una metodologia termoidraulica e un relativo codice che rispondano a bisogni emergenti nel campo computazionale: 1) maggior flessibilità di modellistica geometrica e fisica; accoppiamento semplificato con altre fisiche per l’abilitazione di capacità multi-fisiche; 3) capacità di calcolo parallela su infrastrutture “High Performance Computing”; 4) utilizzo di paradigmi di programmazione moderni.

Un approccio “coarse-mesh” cerca di offrire un bilancio ragionevole fra accuratezza di calcolo, paragonabile a quella dei codici “sub-channel”, e intensità computazionale. Questo approccio può trarre vantaggio dall’utilizzo di geometrie 3-D con mesh non-strutturate, il che aiuta a garantire una maggior flessibilità di modellistica geometrica. In aggiunta a ciò, in un contesto multi-fisico l’accoppiamento fra le varie fisiche è semplificato dall’utilizzo di un approccio “coarse-mesh” e dall’utilizzo di formati di mesh standardizzati. Il framework computazionale che è stato adottato in questo lavoro consiste nella libreria computazionale OpenFOAM basata sul Metodo a Volumi Finiti, la quale offre alcuni degli aspetti desiderati, ovvero capacità di calcolo parallele e un paradigma di programmazione moderno e orientato ad oggetti.

La metodologia “coarse-mesh” è presentata insieme ad una derivazione teorica delle equazioni di bilancio per un generico sistema multi-fase. Sulla base di ciò, è stato sviluppato un codice per la simulazione di flussi mono-fase e bi-fase, principalmente rivolto ai reattori veloci raffreddati a sodio (SFRs), che rappresentano la tipologia di reattori veloci dispiegabile nel più breve termine. Questa tipologia di reattori è stata scelta anche in virtù delle problematiche numeriche associate alla simulazione dei cambi di fase nel sodio, utili per la valutazione della stabilità numerica degli algoritmi di soluzione.

I principali risultati di questo sviluppo consistono in: 1) un nuovo algoritmo di soluzione bi-fase per l’accoppiamento pressione-velocità che migliora stabilità e prestazioni rispetto ad algoritmi esistenti; 2) l’adozione di pratiche di programmazione moderne ed orientate ad oggetti che permettono la implementazione di diversi fluidi termovettori; 3) la verifica del codice tramite un’implementazione ad-hoc del metodo delle “Manufactured Solutions”; 4) il raggiungimento di una buona scalabilità parallela fino a migliaia di core computazionali. In termini di applicazioni: 1) alcuni esperimenti di ebollizione del sodio sono stati riprodotti per ragioni di confronto e convalidazione del codice; 4) alcune caratteristiche di design degli elementi di combustibile degli SFR sono state investigate come caso dimostrativo. Inoltre, la metodologia è stata integrata nel codice multi-fisico GeN-Foam e applicata alla simulazione di un test “Loss Of Flow Without SCRAM” effettuato presso la Fast Flux Test Facility. Questa analisi di benchmark si svolge nell’ambito di un progetto di ricerca coordinato dell’Agenzia Internazionale per l’Energia Atomica ed è destinata a fornire preziosi feedback in termini di confronto fra codici.

Апстракт

Анализа перформанси и сигурности нуклеарних реактора се ослања на коришћење рачунарских метода. У том контексту, овај рад има за циљ развој и примену термо-хидрауличке методологије и софтвера који одговарају новим потребама у рачунарском домуену: 1) већа флексибилност геометријског и физичког моделирања; 2) једноставније повезивање са другим физичким процесима унутар реактора, ради омогућења „мулти-физичког“ приступа; 3) паралелне обраде на „High Performance Computing“ кластерима; 4) усвајање савремених метода у програмирању.

„Coarse-mesh“ приступ нуди разумну равнотежу између рачунске тачности, упоредиве са тачношћу „sub-channel“ приступима и са друге стране оптерећења рачунарских ресурса. Развијен приступ може да користи описане 3-D геометрије са неструктурисаним мрежама, које су корисне за горе наведене захтеве за геометријском флексибилношћу. Поред тога, у контексту „мулти-физичког“ приступа, операције повезивања између различитих физичких процеса унутар реактора поједностављене су усвајањем приступа „coarse-mesh“ и употребом стандардизованих мрежних формата. Употребљени програмски оквир састоји се од софтверског пакета OpenFOAM, који нуди жељене карактеристике, то јест, потенцијал за обимне паралелне обраде и модерне парадигме објектно-оријентисаног програмирања.

Методологија „coarse-mesh“ је представљена заједно са теоријским извођењем главних једначина за општи вишебројни систем. На основу тога, развијен је рачунарски програм за моделирање једнобројних и двобројних токова, са фокусом на симулацију брзих реактора хлађени натријумом (БНРи), који представљају најразвијенију брзо-реакторску технологију. Они су такође изабрани пошто симулација промене фазе у натријуму представља изазован случај за процену нумеричке стабилности алгоритама за решења двобројних токова.

Главна достигнућа овог рада огледају се у следећем: 1) нови алгоритам за двобројну спрегу притиска и брзине који побољшава стабилност и перформансе у поређењу са постојећим алгоритмима; 2) усвајање савремених објектно-оријентисаних метода програмирања које омогућавају поједностављену примену различитих расхладних флуида; 3) верификација програма путем „Manufactured Solutions“ метода; 4) доказане добре перформансе паралелних обрада до вишебројада рачунарских језгара. У смислу примене истог: 1) прелиминарна потврда програма на основу изведеног експеримента кључања натријума; 2) постојеће и предложене нове карактеристике горивих елемената БНРа истражују се као демонстративни случај. Методологија је интегрисана у „вишебројничко“ окружење програма GeN-Foam и примењена на симулацију „Loss Of Flow Without SCRAM“ теста изведеног на Fast Flux Test Facility. Ова анализа се одвија у оквиру координисаног истраживачког пројекта Међународне Агенције за Атомску Енергију и очекује се да ће пружити драгоцене информације у оквиру поређења резултата овог програма у односу на друге програме.

Contents

1	Introduction	1
1.1	Background	1
1.1.1	Role of nuclear reactor computer modelling	1
1.1.2	Overview of modelling approaches	2
1.2	Dissertation scope	4
1.2.1	Motivation	4
1.2.2	Objectives	6
1.2.3	Outline	7
2	Modelling framework	9
2.1	Introduction	9
2.2	The Finite Volume Method	9
2.2.1	Introduction	9
2.2.2	Domain discretization	10
2.2.3	Equation discretization	10
2.2.4	The OpenFOAM programming library	14
2.3	The Coarse-mesh Methodology	15
2.3.1	Volume averaging techniques	18
2.3.2	Volume average of a transport equation	27
2.3.3	Volume average of the Navier-Stokes equations	29
2.3.4	Volume averaged enthalpy transport equation	31
2.4	Clarifications and conclusive remarks	33
2.4.1	Remarks on the relationship between the coarse-mesh approach and an Euler-Euler approach	33
2.4.2	Remarks on the relationship between the coarse-mesh approach and the FVM	33
3	Algorithms and implementation	36
3.1	Introduction	36
3.2	Modelling	37
3.2.1	Flow regime map	38
3.2.2	Momentum transfer closure	40
3.2.3	Heat and mass transfer closure	47
3.2.4	Structure thermal modelling	50
3.2.5	Turbulence modelling and tortuosity remarks	52
3.3	Solution algorithm	55

3.3.1	Solution of the Navier-Stokes equations	55
3.3.2	Solution of the enthalpy equation	66
3.3.3	Coupling algorithm	67
3.4	Algorithm verification	70
3.4.1	The Method of Manufactured Solutions	70
3.4.2	Results	71
3.5	Pressure-velocity coupling algorithm performance	74
3.6	Parallel performance	78
4	Validation against sodium boiling experiments	80
4.1	Introduction	80
4.2	Steady state validation	80
4.2.1	Description of the experiments	81
4.2.2	Computational modelling	81
4.2.3	Results	84
4.3	Transient validation	86
4.3.1	Description of the experiment	86
4.3.2	Computational modelling	88
4.3.3	Results	93
4.4	Conclusions	100
5	Application to flow blockage investigation in SFRs	102
5.1	Introduction	102
5.2	Development of dedicated internal boundary conditions	103
5.2.1	Wrapper heat transfer modelling	103
5.2.2	Pressure baffle	105
5.3	Computational modelling	105
5.3.1	Geometric modelling	105
5.3.2	Investigation procedure	110
5.4	Mesh convergence study	111
5.4.1	Theoretical remarks	111
5.4.2	Results	113
5.5	Assessment of inter-assembly gap, wrapper window behaviour	117
5.5.1	Full power case	117
5.5.2	Low power case	120
5.6	Conclusions	122
6	Multi-physics investigation of FFTF LOFWOS Test 13	125
6.1	Introduction	125
6.2	Multi-physics framework	126
6.2.1	Single-physics capabilities and coupling fields	126
6.2.2	Multi-physics coupling algorithm	128
6.3	Description of the Fast Flux Test Facility and LOFWOS Test 13	130
6.3.1	The Fast Flux Test Facility	130
6.3.2	LOFWOS Test 13 and safety features demonstration	133

6.4	Computational modelling	134
6.4.1	Thermal-hydraulics modelling	134
6.4.2	Neutronics modelling	143
6.5	Results	146
6.5.1	Steady state results	146
6.5.2	Transient results	148
6.6	Conclusions	154
7	Conclusions	156
7.1	Summary	156
7.2	Perspectives and future work	159
Appendix A	1-D nuclear fuel pin thermal model	161
A.1	Modelling and discretization	161
A.1.1	General remarks and approximations	161
A.1.2	Domain discretization	163
A.1.3	Equation discretization	164
A.1.4	Boundary conditions and fuel-cladding coupling	167
A.1.5	Discretization summary and final remarks	171
A.2	Coarse-mesh integration	173

List of Figures

2.3.1	Definition of possible REVs over a bundle of pins	16
2.3.2	Averaging volumes over a two-phase system for the calculation of a spatial derivative	21
3.2.1	Overview of the thermal-hydraulic closure modelling	38
3.2.2	Pressure drop dependence on mass flow rate for two-phase systems	42
3.3.1	Flowchart of the developed thermal-hydraulics solution algorithm	68
3.4.1	Discretization errors versus mesh cell size for the verification of the novel pressure-velocity coupling algorithm with a cell-centered velocity approach	73
3.4.2	Discretization errors versus mesh cell size for the verification of the novel pressure-velocity coupling algorithm with a face-centered velocity approach	73

3.4.3	Discretization errors versus mesh cell size for the verification of the novel pressure-velocity coupling algorithm with a cell-centered velocity approach and larger fluid viscosity compared to previous examples	74
3.5.1	Pressure equation initial residual versus pressure iterations compared between the novel two-phase pressure-velocity coupling approach and existing approaches in OpenFOAM for a variety of purely advective 1-D cases	76
3.5.2	Pressure equation initial residual versus pressure iterations compared between the novel two-phase pressure-velocity coupling approach and existing approaches in OpenFOAM for a simplified 1-D boiling transient	77
3.5.3	Pressure equation initial residual versus pressure iterations compared between the novel two-phase pressure-velocity coupling approach and existing approaches in OpenFOAM for a standard OpenFOAM 2-D test case, <code>bubbleColumn</code>	77
3.6.1	Parallel scaling of the latest implementation of the global solution algorithm and its sub-components as defined in Figure 3.3.1 up to 4096 threads	79
4.2.1	Details of the tubular test section of the JRC Ispra sodium-boiling loop	82
4.2.2	Comparison between calculated and experimental results concerning two-phase pressure drops at the JRC Ispra sodium boiling facility	85
4.3.1	Schematic of the KNS test loop used for sodium boiling investigation	87
4.3.2	View of the KNS-37 assembly geometry with highlights on coarse-mesh averaging REVs	89
4.3.3	Calculated and experimental radial and axial temperature distributions within the KNS-37 bundle during test L22, both at steady-state and before boiling inception	94
4.3.4	Calculated results for the evolution of inlet mass flow rate, pressure, total vapour volume and vapour axial extent in the 2-D axial-symmetric model of the KNS-37 pin bundle obtained by a certain set of correlations, model I	95
4.3.5	Calculated results for the evolution of inlet mass flow rate, pressure, total vapour volume and vapour axial extent in the 2-D axial-symmetric model of the KNS-37 pin bundle obtained by a certain set of correlations, model II	96
4.3.6	Vapour phase fraction distribution in the 2-D axial-symmetric model of the KNS-37 pin bundle at different times during test L22	97

4.3.7	Calculated results for the evolution of inlet mass flow rate, pressure, total vapour volume and vapour axial extent in the 2-D axial-symmetric model of the KNS-37 pin bundle obtained by a certain set of correlations, model III	98
4.3.8	Calculated results for the evolution of inlet mass flow rate, pressure, total vapour volume and vapour axial extent in the 2-D axial-symmetric model of the KNS-37 pin bundle obtained by a certain set of correlations, model IV	99
5.2.1	Schematic view of computational cells at opposing sides of a thermally conductive baffle	104
5.3.1	Various details of the computational domain representative of a bundle of seven SFR assemblies inclusive of the inter-assembly gap	107
5.3.2	Computational domain over SFR assembly wrappers inclusive of assembly windows	108
5.4.1	Examples of different meshes used for the assessment of mesh-convergence of the results obtained from the analysis of flow blockage in a bundle of SFR assemblies	114
5.4.2	Superficial velocity magnitude distribution around SFR wrapper windows in a certain flow blockage condition with different mesh resolution	116
5.5.1	Example temperature distribution in a bundle of windowed SFR where an assembly undergoes flow blockage	117
5.5.2	Outlet bulk temperature of an assembly undergoing flow blockage in a bundle of seven ESFR assemblies for different wrapper configuration, varying inter-assembly gap flow magnitudes and assembly blockage levels at reference power	118
5.5.3	Maximum temperature within an assembly undergoing flow blockage in a bundle of seven ESFR assemblies for different wrapper configuration, varying inter-assembly gap flow magnitudes and assembly blockage levels at reference power	119
5.5.4	Superficial velocity streamlines in around the wrapper windows region of three neighbouring assemblies for varying inter-assembly sodium gap flow	121
5.5.5	Outlet bulk temperature of an assembly undergoing flow blockage in a bundle of seven ESFR assemblies for different wrapper configuration, varying inter-assembly gap flow magnitudes and assembly blockage levels at a decay power level	122
5.5.6	Maximum temperature within an assembly undergoing flow blockage in a bundle of seven ESFR assemblies for different wrapper configuration, varying inter-assembly gap flow magnitudes and assembly blockage levels at a decay power level	123

5.5.7 Time-to-boil of the sodium flow in individual assembly undergoing flow blockage in a bundle of seven ESFR assemblies for different wrapper configuration, varying inter-assembly gap flow magnitudes and assembly blockage levels at a decay power level	124
6.2.1 Relationship between the single-physics sub-solvers of GeN-Foam and overview of coupling fields.	128
6.2.2 Multi-physics coupling algorithm of the GeN-Foam code.	129
6.3.1 Overview of the FFTF components and loops	130
6.3.2 Overview of the FFTF reactor vessel	132
6.3.3 Overview of the operating principle of GEM passive safety feature in the FFTF	133
6.4.1 Representation of the 2-D computational domain of the FFTF thermal-hydraulics	135
6.4.2 Modelling of an IHX via the coarse-mesh approach	139
6.4.3 Representation of the 2-D computational domain of the FFTF neutronics	144
6.5.1 Calculated neutron flux distributions in the FFTF core for different energy groups at steady state with the diffusion neutronics sub-solver	147
6.5.2 Calculated temperature distribution and representative velocity streamlines in the FFTF at steady-state	148
6.5.3 Calculated results for the evolution of the individual reactivity contributions obtained by the reference model of FFTF LOFWOS Test 13 compared against experimental data	149
6.5.4 Calculated results for the evolution of total reactor power, inner core outlet temperature, primary mass flow rate, total reactivity obtained by the reference model of FFTF LOFWOS Test 13 compared against experimental data	150
6.5.5 Calculated temperature distributions and representative velocity streamlines in the FFTF at two different points in time during the transient	151
6.5.6 Comparison of calculated results for the evolution of inner core outlet temperature and reactivity obtained by different models of FFTF LOFWOS Test 13	153
6.5.7 Calculated inner core outlet temperature results for the FFTF LOFWOS Test 13 compared against results obtained by the SAS4A/SASSYS-1 system code and experimental results	154
A.1.1 Representation of an axial slice of a nuclear fuel pin with the definition of a 1-D radial mesh for a FVM-based discretization of the heat diffusion equation	163

List of Tables

4.3.1	KNS-37 pin bundle specifications	87
4.3.2	Data of the test L22 performed at the KNS-37 experimental facility	88
4.3.3	Coarse-mesh properties of the 2-D axial-symmetric model representing the KNS-37 heated bundle by coarse-mesh region. . .	89
4.3.4	Comparison of employed physical correlations for a variety of test cases for the simulation of the KNS-37 test L22 transient.	95
5.3.1	Reference design and performance data of the ESFR core . .	106
5.3.2	Geometric and coarse-mesh modelling data for a bundle of seven SFR assemblies	109
5.4.1	Mesh convergence of the results obtained from the analysis of flow blockage in a bundle of SFR assemblies at low power . .	114
5.4.2	Mesh convergence of the results obtained from the analysis of flow blockage in a bundle of SFR assemblies at high power . .	115
6.3.1	Reference FFTF data at nominal reactor conditions, together with assembly, fuel data for LOFWOS Test 13	131
6.4.1	FFTF fuel modelling	137
6.4.2	Properties of the coarse-mesh model of the FFTF thermal-hydraulics	138
6.5.1	Comparison of calculated and experimental steady-state FFTF conditions before the LOFWOS Test 13 transient	147

List of Acronyms

ANL	Argonne National Laboratory
CFD	Computational Fluid Dynamics
CFL	Courant–Friedrichs–Lowy condition
CRP	Coordinated Reasearch Project
DNS	Direct Numerical Simulation
DHX	Dump Heat Exchanger
ESFR	European Sodium-cooled Fast Reactor
FCT	Flux Corrected Transport
FFTF	Fast Flux Test Facility
FVM	Finite Volume Method
GCI	Grid Convergence Index
GEM	Gas Expansion Module
HPC	High Performance Computing
IAEA	International Atomic Energy Agency
IHX	Intermediate Heat Exchanger
JRC	Joint Research Centre
KNS	Kompakter Natriumsiede Kreislauf
LES	Large-Eddy Simulation
LHS	Left-hand Side
LOFWOS	Loss Of Flow Without SCRAM
LWR	Light Water Reactor
MMS	Method of Manufactured Solutions
MSR	Molten Salt Reactor
MULES	Multidimensional Universal Limiter with Explicit Solution
PDE	Partial Differential Equation
PIMPLE	Merged PISO-SIMPLE
PISO	Pressure Implicit with Splitting of Operators

RV	Representative Volume
RANS	Reynolds-Averaged Navier Stokes
REV	Representative Elementary Volume
RIA	Reactivity Insertion Accident
RHS	Right-hand Side
SIMPLE	Semi-Implicit Method for Pressure Linked Equations
SFR	Sodium-cooled Fast Reactor
S_N	Discrete Ordinates of order N
ULOF	Unprotected Loss Of Flow

List of Symbols

Bold symbols represent tensor quantities of order equal to or greater than one. The symbol – is used to signify that the corresponding quantity is dimensionless, while the symbol • is used to signify that the corresponding quantity has dimensions that depend on the context.

Greek letters - quantities

α	Volumetric phase fraction (–)
α^N	Structure-normalized volumetric phase fraction (–)
β	Delayed neutron fraction (–)
β_i	Delayed neutron fraction of group i precursors (–)
ϵ	Relative surface roughness (–)
ε	Specific turbulence kinetic energy dissipation rate (m^2/s^3)
Γ	Volumetric mass source/sink ($kg/m^3/s$)
κ	Thermal conductivity ($W/m/K$)
Λ	Prompt neutron generation time (s)
λ	MULES limiter (–)
λ_i	Decay constant of group i precursors ($1/s$)
μ	Molecular viscosity ($Pa\ s$)
ν	Kinematic diffusivity (m^2/s)
ρ	Density (kg/m^3) or reactivity (–) depending on context
θ	Backward-forward Euler weight factor (–)
ϕ	Generic quantity (•) or superficial volumetric fluid flux (m^3/s) depending on context
ϕ_α	Volumetric fluid flux (m^3/s)

ϕ^2	Two-phase drag multiplier (-)
ψ	Compressibility ($kg/m^3/Pa$)
ξ	Continuity error ($kg/m^3/s$)
ξ_k	Head loss coefficient (-)
ω	Turbulence convergence time scale in porous media ($1/s$)

Roman letters - quantities

a	Thermal diffusivity (m^2/s)
A'''	Volumetric surface area density (m^2/m^3)
C_i	Concentration of group i precursors ($1/m^3$)
c_p	Specific heat capacity ($J/Kg/K$)
D_h	Hydraulic diameter or phase characteristic dimension (m)
D_p	Pin diameter (m)
D_w	Wire diameter (m)
E	Absolute error (\bullet)
e	Relative error (-)
\mathbf{e}	Basis vector (-)
e_P	Parallel efficiency (-)
$f_{A''_s}$	Fluid-structure contact fraction (-)
f_d	Drag coefficient (-)
f_F	Nucleate boiling flow enhancement factor (-)
f_S	Nucleate boiling suppression factor (-)
f_{vm}	Virtual mass coefficient (-)
\mathbf{g}	Gravitational acceleration (m/s^2)
h	Specific enthalpy (J/kg) or characteristic mesh dimension (-) depending on context
H	Heat transfer coefficient ($W/m^2/K$)
I_t	Turbulence intensity (-)

k	Specific turbulence kinetic energy (m^2/s^2)
k_{eff}	Effective neutron multiplication factor (-)
K, \mathbf{K}	Dimensional drag factor, scalar for fluid-fluid drag, tensor for fluid-structure drag ($kg/m^3/s$)
L_w	Pin-wrapping-wire lead length (m)
L_t	Turbulence length scale (m)
\mathbf{M}	Momentum source ($kg/m^2/s^2$)
n	Neutron density ($1/m^3$)
\mathbf{n}	Surface normal vector (-)
P_p	Pin pitch (m)
p	Pressure (Pa)
$P_t = D_p + 1.0444 D_w$	Wire-adjusted pin pitch (m)
q	Volumetric power density or heat transfer term (W/m^3) or order-of-accuracy (-) depending on context
r	Radius (m)
S	Surface area (m^2)
T	Temperature (K)
\mathbf{T}	Tortuosity tensor (-)
\mathbf{u}	Velocity (m/s)
V	Volume (m^3)
W_p	Wet pin bundle perimeter (m)
W_w	Wet wrapper perimeter (m)
\dot{x}	Flow quality (-)
X^2	Lockhart-Martinelli parameter (-)

Greek letters - labels & subscripts

γ, ζ	Generic phase
∂	Phase interface in a two-phase system
Ω	Spatial domain

Roman letters - labels & subscripts

c	Mesh cell
e	Extrapolated value
f	Mesh cell face
i, j	Generic fluid phase (unless differently specified)
p	Fuel pin
s	Solid structure phase
sup	Superficial quantity
w	Fuel pin wrapping wire

Chapter 1

Introduction

1.1 Background

1.1.1 Role of nuclear reactor computer modelling

The process of designing and predicting the behaviour of nuclear reactors in a variety of circumstances, from steady-state operation to safety-relevant accidents requires modelling of a variety of phenomena, that can be essentially grouped in a number of *physics*, chiefly among which (yet not exclusively) are:

- *thermal-hydraulics*, namely the analysis of momentum and heat transfer processes involving the coolant and/or moderator flow inside the reactor vessel and other circuit components;
- *neutronics*, namely the analysis of neutron transport and reaction processes through the media that constitute the reactor core with the aim of predicting reactor power and fuel composition evolution;
- *fuel behaviour*, namely the analysis of the physical and chemical properties of the fuel and their evolution in response to varying reactor conditions;
- *structural-mechanics*, namely the analysis of the physical and thermal properties of the structures of interest that can range from the core to the containment building as a whole in response to varying conditions.

These physics are fundamentally non-linear and highly coupled, which thus characterizes nuclear reactors as *multi-physics* systems. As a further layer of complexity, the temporal and spatial scales that characterize phenomena of interest can span several orders of magnitude in both time and space. Temporally, these vary from fractions of a second (e.g. the simulation of Reactivity Insertion Accidents (RIAs)) to years (e.g. the evolution of fuel and structural material properties due to long-term irradiation). Spatially, these vary from millimeters (e.g. fluid flow and heat transfer in the fuel pin lattice) to tens of meters (e.g. fluid flow and heat transfer between the reactor vessel and heat

exchangers). While experimental approaches for the investigation of relevant reactor phenomena provide the ultimate connection between theory and reality, the resulting theoretical validation as well as the plethora of data produced by experiments enable the utilization of computers for a number of purposes.

Over the decades, with progress in computer hardware as well as increased knowledge of nuclear reactor dynamics, computer codes have become an important support or integral element for activities ranging from reactor design and development to safety analysis, decision-making for risk-informed activities, reactor licensing, and resolution of technical issues in general [1]. These computer codes range in scope from the investigation of purely single-physics phenomena to e.g. the modelling of the multi-physics coupling between a number of single-physics of interest, chiefly among which is the coupling between thermal-hydraulics and neutronics [2].

To better understand the context of this work, a review of modelling approaches adopted by computer codes for the treatment of both single-physics (with specific regards to thermal-hydraulics) and multi-physics phenomena is presented. The objectives of this work are then discussed from the perspective of how advancements in different key areas of scientific computing and programming are taken advantage of in order to address some of the shortcomings of existing (often referred to as “legacy”) computer codes and their modelling approaches.

1.1.2 Overview of modelling approaches

Single-physics modelling To this day, the development of nuclear computer codes capable of resolving all of the spatial scales that partake in defining the domain of even a single-physics remains a challenging task. With regards to these single-physics codes, different phenomena of interest spurred the development of different codes with different applicability ranges [3]. In the field of thermal-hydraulics, computer codes are generally distinguished based on their target spatial resolution.

System analysis codes, also referred to as *system codes*, take advantage of a limited spatial resolution to describe all the relevant components (reactor vessel, core, circuit piping, heat exchangers, etc.) as a collection of 0-D or 1-D elements. Advances in the hardware and the codes themselves are nonetheless leading to 2-D and 3-D approaches for the modelling of the most critical components, such as the core, albeit at a similarly limited spatial resolution scale. Notable examples of system codes include ATHLET [4], RELAP5/-3D [5] and TRACE [6].

Another category of thermal-hydraulics computer codes is that of *sub-channel codes* [7], that operate at length scales that are intermediate between that of a fuel assembly, also known as channel, and a fuel pin. The modelling approaches are most commonly 2-D and 3-D, and while generally limited to individual fuel elements or bundles of fuel elements, applications for full-core simulations do exist, as testified e.g. by the CTF code [8] within the VERA

simulation environment [9]. Boundary conditions for such analyses are generally provided from the broader-scale analyses performed via system codes [7]. In addition to CTF, further notable sub-channel codes consist of COBRA-TF [10], SABENA [11], SABRE [12], SubChanFlow [13].

At a further level of spatial resolution lie what are commonly referred to as Computational Fluid Dynamics (CFD) approaches, which are aimed at resolving fluid flow and heat transfer dynamics at an intra-pin scale, typically with 3-D approaches. While CFD approaches have no theoretical maximum length scale applicability range, this is practically constrained by the availability of computational power, as larger geometries will take more computational resources to treat at the envisioned scales of detail. An important aspect should be nonetheless clarified. While the term CFD should *by definition* collectively denote all computational methods for the treatment of reactor thermal-hydraulics, in the nuclear community it is almost exclusively used to represent computational approaches with the following characteristics: 1) computational domain represented in 2-D or 3-D, possibly with the use of unstructured meshes; 2) characteristic mesh length scales small enough to fully resolve the geometry of interest (e.g. the fluid in between fuel pins); 3) usage of Reynolds-Averaged Navier Stokes (RANS)-based models for the treatment of turbulence.

While higher-fidelity approaches exist, e.g. Large-Eddy Simulation (LES) turbulence modelling or Direct Numerical Simulation (DNS) approaches, these are fundamentally constrained by their computational cost. An example is provided by Bieder et al. whose LES analysis of a single fuel assembly of a Sodium-cooled Fast Reactor (SFR) entailed a computational domain of over $260 \cdot 10^6$ cells [14], which would otherwise largely suffice for the analysis of an entire reactor core with RANS-based CFD approaches. The main value of LES and DNS generally consists in providing benchmarks for gauging the effectiveness of physical models used in “standard” (i.e. in the sense defined earlier) RANS-based CFD approaches [15], in addition to experimental data.

Important consequences follow from the choice of a particular approach. When moving up the spatial resolution scale from the finest level of engineering interest to the coarsest level:

- the treatment of various physical phenomena needs to increasingly rely on experimental data in the form of correlations to account for the loss of geometric detail;
- the computational domains are reduced in both size and possibly dimensionality (e.g. 3-D to 1-D), which significantly alleviates computational burdens.

At one end of the scale, ideally, a DNS approach would exclusively require a detailed definition of the domain geometry and the thermo-physical properties of the fluid of interest in order to resolve all heat transfer, momentum transfer phenomena for any possible reactor type or reactor geometry. On the other

hand, system codes were generally designed for specific reactor types, reactor geometries and/or working fluids so that specific experimental correlations could be employed to account for the lack of spatial resolution. When one talks of *hard-coding*, they refer precisely to the degree to which case-dependent data (geometries of interest, correlations, etc.) is embedded in the code itself, rather than being an input that can be selected and/or defined independently by the user at code run time. In simpler terms, hard-coding ultimately manifests itself in the practice of developing case-specific simulation codes (e.g. system codes for the exclusive analysis of a number of Light Water Reactor (LWR) designs or sub-channel codes for the exclusive analysis of sodium-cooled hexagonal-lattice fuel elements).

Multi-physics modelling Due to the original limitations imposed by hardware in the development of computer codes for reactor analyses, what became well-established codes generally arose in the context of the analysis of a single specific physics. With the increased availability of computational power came the interest to achieve various degrees of coupling between the individual physics for multi-physics simulations to analyze more complex phenomena. In order to capitalize on the efforts that had been dedicated to the development of single-physics codes, multi-physics approaches initially focused on what is referred to as *external coupling* of existing single-physics codes. This consists in developing program interfaces that exchange data between the individual codes in the form of input-output files that are written to or read from disk whenever necessary. In spite of the overheads associated with disk read-write operations at every simulation step, this approach retained and still retains a significant appeal. This is also due to the fact that it allows to couple codes written with either different programming languages or, if using the same programming language, that rely on different data management approaches for the numerical representation of the coupling physical fields and quantities. With time, where allowed by the use of the same programming language and/or similar data management approaches, *internal coupling* strategies arose to further improve performance. In such a strategy, coupling fields are passed directly via memory rather than disk read-write operations. To this day, the external and internal coupling of mature, independent single-physics codes represent the vast majority of multi-physics approaches in the nuclear field, as testified by the extensive review by Wang et al. [2].

1.2 Dissertation scope

1.2.1 Motivation

From a general perspective, “design-by-simulation” represents a trend of increasing importance in the nuclear field, which calls for higher-accuracy approaches in the modelling of the different reactor physics and their coupling.

This is even more-so relevant for the investigation and analysis of advanced reactor concepts, which are enjoying a gradual resurgence of interest as testified by the Generation IV International Forum (GIF) [16] and for which: 1) “design-by-experiment” resources are lacking compared to what LWR technology had enjoyed over past decades; 2) legacy codes could be unsuitable for their analysis due to the fact that advanced reactor technologies such as SFRs, Molten Salt Reactors (MSRs), Lead-cooled Fast Reactors (LFRs), Gas-cooled Fast Reactors (GFRs), Supercritical Water-cooled Reactors (SCWRs), Very High Temperature Reactors (VHTRs) adopt typically unconventional geometries and/or working fluids.

From the perspective of single-physics thermal-hydraulics, state-of-the-art analysis with legacy codes is typically performed at a multi-scale level with the use of multiple codes with different spatial resolution [7][17]. Practically, this approach employs system codes for the simulation of reactor circuits, and the results obtained from this scale of analysis are employed as boundary conditions for the simulation of more spatially-limited domains (e.g. the reactor core only) with a higher resolution. These are in turn treated via e.g. sub-channel codes, and possibly CFD codes for the analysis of core plena or higher resolution of the fuel assemblies. However, the coupling of these thermal-hydraulics codes for the purpose of achieving a multi-scale resolution entails a number of disadvantages. The same disadvantages are encountered in the frame of multi-physics coupling between different single-physics codes.

As a matter of fact, a code that consist of coupled legacy single-physics codes (regardless of the scope, whether multi-physics codes or multi-scale single-physics codes) is fundamentally characterized by two drawbacks, regardless of whether the coupling is external or internal. On one hand, the maintenance of such a code is essentially split among the maintenance of the two (or more) single-physics codes and the maintenance of the data-passing interface itself. This can be a cumbersome task to sustain over time owing to possibly different developing institutions as well as code-to-code implementation differences. On the other hand, possible inconsistencies and differences in the numerical methods (discretization schemes, linear solver solution algorithms, intra-physics coupling schemes to resolve single-physics non-linearities) used by each of the coupled single-physics codes can in principle affect simulation results to a significant degree, and thus the achievable accuracy.

In the context of multi-physics codes, an emergent trend consists instead of internally coupled multi-physics codes that rely on single-physics codes developed *ex-novo* yet within a unified framework. The driving factors behind these efforts are summarized by:

- the benefits of multiple physics being treated in a single and coherent *programming framework*;
- the possibility to take advantage of modern hardware advances, ranging from parallel calculations on workstations to massive parallel scalability on High Performance Computings (HPCs);

- the possibility to take advantage of modern object-oriented paradigms offered by programming languages such as C++.

The idea of such programming frameworks is to provide tools for the management of fundamental yet by no means trivial nor necessarily nuclear-specific aspects that lie at the basis of the vast majority of computer simulation codes, such as the representation of the computational domain, fields, partial differential equations (PDEs) and linear system solvers. Due to their generality, these frameworks are oriented towards 3-D approaches and provide native support for parallel capabilities, to fully take advantage of modern hardware. Notable examples of this are represented by the Finite Element Method (FEM)-based MOOSE framework [18] and the Finite Volume Method (FVM)-based OpenFOAM framework [19]. With specific regards to the latter, while it was developed as a general framework for CFD simulations (in the general, non-nuclear sense), it was taken advantage of in multiple occasions for nuclear specific applications, and represents the overarching framework for the present dissertation.

1.2.2 Objectives

The purpose of this work lies in the development and application of a reactor thermal-hydraulics modelling methodology and a related computer code for the treatment of one-phase and two-phase flows, in a manner that addresses some of the issues of existing computational approaches as illustrated so far. This is achieved via focusing on the following aspects: multi-scale capabilities, code modelling and development flexibility, internal integration in a multi-physics environment within a unified programming framework.

The methodology consists of a *coarse-mesh* approach, which sets out to achieve a degree of accuracy comparable to sub-channel approaches while achieving multi-scale potential, i.e. the possibility to model all reactor scales of interest in a unified computational environment. The methodology thus offers the potential to act in a manner analogous to system codes in reactor circuit regions, sub-channel codes in reactor core regions (and will thus generally require correlations for heat, momentum transfer phenomena), CFD codes in core plena.

The instantiation of the methodology in the form of a computer code takes advantage of the aforementioned OpenFOAM computational environment, which simplifies the adoption of general, possibly unstructured meshes for the modelling of geometries of interest (conducive to the investigation, among others, of advanced reactor concepts), enhances code flexibility for the implementation of possibly ad-hoc features and streamlines the multi-physics coupling process via certain field manipulation and projection algorithms.

While there are no limitations in the reactor geometries and types that can be modelled via a coarse-mesh approach, the work presented in this dissertation is heavily focused on modelling phenomena in SFRs. This is due to

a number of reasons. In the first place, SFRs represent one of the, if not the, nearest-term deployable fast reactor technology for which dedicated simulation tools are lacking when compared to the e.g. LWRs. In the second place, two-phase flows in SFRs resulting from coolant boiling pose greater numerical challenges when compared against boiling water flows in LWRs. It should be stressed that in spite of this SFR-oriented work, the developed methodology and computer code is fundamentally applicable to any reactor concept of interest, insofar as experimental correlations for the treatment of system-specific momentum, heat and mass transfer phenomena are available.

On the multi-physics side, the developed single-physics thermal-hydraulics code is integrated within a broader multi-physics code represented by the OpenFOAM-based GeN-Foam code, which was also subject to further development. GeN-Foam was originally jointly developed at the Paul Scherrer Institute and at the Laboratory for Reactor Physics and Systems Behaviour, EPFL, as a multi-physics code for the coupled 3-D simulation of reactor thermal-hydraulics, neutronics and core structure-mechanics [20].

1.2.3 Outline

The remainder of this chapter outlines the structure of the rest of this document. Chapter 2 presents the overall numerical and modelling framework that is the foundation for the present thermal-hydraulics approach. In particular, it covers fundamentals of the numerical discretization approach, the FVM as well as an overview of the OpenFOAM environment. A rigorous derivation of the mathematical tools that enable a coarse-mesh fluid flow description, as well as their application for the derivation of governing equations for a generic multi-phase system is presented. Chapter 3 presents in greater detail how the tools developed in chapter 2 are used to construct and provide the necessary closure of the governing thermal-hydraulics equations in a coarse-mesh context. It further discusses the solution algorithms for the numerical treatment of the governing equation, namely the treatment of pressure-velocity coupling and the coupling of fluid-mechanics with enthalpy transport. This is inclusive of the discussion of further developments in the area of two-phase pressure-velocity coupling that bring a number of advantages compared to other approaches found in existing OpenFOAM two-phase solvers. A verification of the solution approaches via the Method of Manufactured solution is provided for further clarity. The multi-core scaling performance of the two-phase algorithm up to 4096 cores concludes the chapter. Chapter 4 presents the validation of the methodology and the implemented models by comparison against experimental data, with focus on SFR technology. This consists of the modelling of a number of sodium-boiling scenarios in both quasi-steady-state and transient conditions, with the introduction of sodium-specific models and correlations. Chapter 5 discusses an application of the one-phase thermal-hydraulics methodology for the investigation of both existing and proposed novel design features for SFR fuel assemblies. Furthermore, this one-phase

investigation is also aimed at establishing guidelines for the assessment of the validity of the obtained coarse-mesh results, specifically in terms of mesh-independence. Chapter 6 is dedicated to the application of the methodology to the multi-physics, multi-scale simulation of one-phase thermal-hydraulics and neutronics for a transient experiment conducted at the Fast Flux Test Facility (FFTF), a former experimental SFR. Chapter 7 concludes this dissertation by providing a summary of the investigated topics and achieved results as well as outlining possible future research directions.

Chapter 2

Modelling framework

2.1 Introduction

This chapter discusses the foundational notions required for contextualizing and understanding the developments presented in this work. The degree of detail with which each topic is introduced and treated is limited to the extent to which it is instrumental for further discussion.

Section 2.2 presents the numerical approach employed to represent and solve the governing equations of the physics of interest, namely the Finite Volume Method (FVM). The numerical representation and solution framework was not developed *ex novo*, and consists instead of the OpenFOAM C++ programming library, introduced in section 2.2.4. This enables to take advantage of a modern, object-oriented programming paradigm. Section 2.3 presents the coarse-mesh methodology and how it takes advantage of volume averaging techniques to produce a coarse-mesh representation of a generic system of coupled partial differential equations (PDEs), as well as an application to the governing equations of interest in thermal-hydraulics, namely the Navier-Stokes equations and the enthalpy transport equation. Section 2.4 presents some conclusive clarifications regarding the role of a coarse-mesh methodology compared to other concepts explored in this chapter.

2.2 The Finite Volume Method

2.2.1 Introduction

The process of obtaining a numerical approximation to the solution of a PDE or to a system of coupled PDEs defined over a set of domains and bounded by a set of boundary and initial conditions fundamentally relies on discretization practices. These can be distinguished between:

- the discretization of the PDE domains, which in the broader field of Computational Fluid Dynamics (CFD) generally consist of a spatial domain (from 1-D to 3-D) and a temporal domain;

- the discretization of the PDEs themselves. This allows to represent the initial problem posed by the PDEs in terms of a set of algebraic relationships which can be numerically evaluated.

The FVM is one of the possible discretization approaches and the goal of the present sub-sections is to introduce the most relevant aspects concerning it.

2.2.2 Domain discretization

The discretization of the domains over which the PDEs are defined results in the computational domain, which is a collection of discrete elementary sub-domains. In the field of CFD, these individual sub-domains are generally referred to as *control volumes* or *cells* when related to the spatial domain, and *time steps* or *time levels* when related to the temporal domain. The collection of all the control volumes is also referred to as computational *grid* or *mesh*. While there are no mathematical limitations on the shape of the boundary $\partial\Omega_c$ of each cell Ω_c other than to avoid cell overlapping, polyhedral cells are the most reasonable choice for convenience of a simpler numerical representation. Any continuous field of interest $\phi^* = \phi^*(\mathbf{x}, t)$ is then represented on the discretized domain in terms of a piecewise constant function $\phi = \phi(\mathbf{x}, t)$, so that over each cell Ω_c that has a volume V_c :

$$\phi_c \equiv \phi(\mathbf{x}_p, t) = \frac{1}{V_c} \int_{\Omega_c} \phi^*(\mathbf{x}, t) dV \quad (2.2.1)$$

in which \mathbf{x}_p is the position of any point $p \in \Omega_c$. If the field ϕ^* consists of the solution to the PDEs of interest over said domain, the sought discrete field ϕ will need to be obtained from the solution of the discretized set of PDEs.

2.2.3 Equation discretization

The Finite Volume discretization of the PDEs consists in their integration over each control volume Ω_c , time step $t, t + \Delta t$, and the application of the Stokes theorem, where necessary, to relate the initial continuous domain problem to a collection of algebraic relations between variables in the discrete domain. To better illustrate the concept, the conservation equation of a generic quantity ϕ^* (which could be a scalar or a vector) advected by a field \mathbf{u}^* is used as an example:

$$\frac{\partial}{\partial t} \phi^* + \nabla \cdot (\phi^* \mathbf{u}^*) = 0 \quad (2.2.2)$$

Let us focus on a generic mesh cell Ω_c of the discretized spatial domain and on a generic time step of the discretized temporal domain bounded by t and $t + \Delta t$. By integrating equation 2.2.2 over those domains, the following holds:

$$\int_t^{t+\Delta t} \int_{\Omega_c} \frac{\partial}{\partial t} \phi^* dV dt + \int_t^{t+\Delta t} \int_{\Omega_c} \nabla \cdot (\phi^* \mathbf{u}^*) dV dt = 0 \quad (2.2.3)$$

Let us proceed with the further discretization of the two terms independently. With regards to the temporal derivative term, i.e. the first term on the left-hand side (LHS) of 2.2.3, due to the independence of the spatial and temporal coordinates and by recalling equation 2.2.1 that defines the discretized field ϕ , it is shown that:

$$\int_t^{t+\Delta t} \int_{\Omega_c} \frac{\partial}{\partial t} \phi^* dV dt = \int_t^{t+\Delta t} \frac{\partial}{\partial t} \int_{\Omega_c} \phi^* dV dt = V_c \int_t^{t+\Delta t} \frac{\partial}{\partial t} \phi_c dt \quad (2.2.4)$$

with $\phi_c = \phi(\mathbf{x}_p, t) \forall p \in \Omega_c$ and it was further assumed that the cell volume V_c does not change in time. The time derivative itself can be discretized in a variety of ways to algebraically relate it to the new time step field value $\phi_c^n = \phi(\mathbf{x}_p, t + \Delta t)$ and the old time step value $\phi_c^o = \phi(\mathbf{x}_p, t)$ or even older time steps $\phi_c^{oo} = \phi(\mathbf{x}_p, t - \Delta t)$, $\phi_c^{ooo} = \phi(\mathbf{x}_p, t - 2\Delta t)$, etc. depending on the time discretization scheme. If we consider e.g. the Euler discretization scheme:

$$\frac{\partial}{\partial t} \phi_c = \frac{\phi_c^n - \phi_c^o}{\Delta t} \quad (2.2.5)$$

By considering that ϕ is piecewise constant in time as well, then the space and time-integrated time derivative term becomes:

$$\int_t^{t+\Delta t} \int_{\Omega_c} \frac{\partial}{\partial t} \phi^* dV dt = V_c (\phi_c^n - \phi_c^o) \quad (2.2.6)$$

Let us now focus on the discretization of the advective term in equation 2.2.3, i.e. the second term on the LHS. By taking advantage of the Stokes theorem it is possible to equate the integral of $\nabla \cdot (\phi^* \mathbf{u}^*)$ over the cell Ω_c to the integral of $\phi^* \mathbf{u}^*$ over the boundary $\partial\Omega_c$ of cell Ω_c :

$$\int_t^{t+\Delta t} \int_{\Omega_c} \nabla \cdot (\phi^* \mathbf{u}^*) dV dt = \int_t^{t+\Delta t} \int_{\partial\Omega_c} \phi^* \mathbf{u}^* \cdot d\mathbf{S} dt \quad (2.2.7)$$

in which $d\mathbf{S} = \mathbf{n} dS$ is the infinitesimal surface area vector on $\partial\Omega_c$. Since we assume that the mesh is composed exclusively of polyhedra, the integral over the cell boundary can be decomposed into a sum of integrals over the boundary faces:

$$\int_{\partial\Omega_c} \phi^* \mathbf{u}^* \cdot d\mathbf{S}_c = \sum_{f \in \partial\Omega_c} \int_f \phi^* \mathbf{u}^* \cdot d\mathbf{S} = \sum_{f \in \partial\Omega_c} \left(\int_f \phi^* \mathbf{u}^* dS \right) \cdot \mathbf{n}_f \quad (2.2.8)$$

where \mathbf{n}_f represents the face area normal. Just as equation 2.2.1 relates the continuous-domain fields to their discretized counterparts over each cell Ω_c , it is possible to define a discretized field ϕ_f that is piecewise constant over each cell face f so that:

$$\phi_f \equiv \phi(\mathbf{x}_f, t) = \frac{1}{S_f} \int_f \phi^*(\mathbf{x}, t) dS \quad (2.2.9)$$

in which \mathbf{x}_f is the position of any point $p \in f \in \partial\Omega_c$. By defining a new continuous quantity $\mathbf{F}^* = \phi^* \mathbf{u}^*$ and taking advantage of the discrete face integral defined in equation 2.2.9, equation 2.2.8 can be rewritten as:

$$\sum_{f \in \partial\Omega_c} \left(\int_f \mathbf{F}^* dS \right) \cdot \mathbf{n}_f = \sum_{f \in \partial\Omega_c} S_f \mathbf{F}_f \cdot \mathbf{n}_f = \sum_{f \in \partial\Omega_c} \mathbf{F}_f \cdot \mathbf{S}_f \quad (2.2.10)$$

We recall that the objective of this process is to reformulate the advective term in terms of the discrete fields ϕ , \mathbf{u} defined by equation 2.2.1. So far, we reformulated it in terms of a discrete field \mathbf{F}_f that exists on the mesh faces. Recalling the definition of its continuous counterpart, $\mathbf{F}^* = \phi^* \mathbf{u}^*$, we approximate \mathbf{F}_f with the discrete fields ϕ , \mathbf{u} as:

$$\mathbf{F}_f \approx (\phi \mathbf{u})|_f \quad (2.2.11)$$

where $\bullet|_f$ denotes an interpolation of \bullet to the face f . While not explicitly stated so far, the discrete field values of any field have been assumed to be representative of the continuous field values at the cell centers, so that the generic interpolation $\bullet|_f$ typically involves (at least) the values of the discrete field \bullet at the cell centers of the two cells that share face f . As an example, for a generic variable ϕ advected by a field \mathbf{u} , assuming that the face f is shared between cells $c0, c1$ and has a normal \mathbf{n}_f that points from $c0$ to $c1$, the *upwind* interpolation scheme prescribes:

$$\phi|_f = \begin{cases} \phi_{c0} & \text{if } \mathbf{u}_{c0} \cdot \mathbf{n}_f > 0 \\ \phi_{c1} & \text{if } \mathbf{u}_{c1} \cdot \mathbf{n}_f > 0 \end{cases} \quad (2.2.12)$$

Depending on the chosen interpolation scheme, the resulting computational stencil might be larger. The ultimate point at issue is that $(\phi \mathbf{u})|_f$ can be expressed as a linear combination of the values ϕ_c , \mathbf{u}_c that exist in some neighbourhood of face f . Thus, the volume and time integrated advective term becomes:

$$\int_t^{t+\Delta t} \int_{\Omega_c} \nabla \cdot (\phi^* \mathbf{u}^*) dV dt \approx \int_t^{t+\Delta t} \sum_{f \in \partial\Omega_c} (\phi \mathbf{u})|_f \cdot \mathbf{S}_f dt \quad (2.2.13)$$

The remaining time integral can be treated in a variety of ways. As all the discretized fields are assumed to be piecewise constant in time just as they are in space and by assuming that the cell faces do not change shape nor orientation in time, the following is valid:

$$\int_t^{t+\Delta t} \sum_{f \in \partial\Omega_c} (\phi \mathbf{u})|_f \cdot \mathbf{S}_f dt = \sum_{f \in \partial\Omega_c} (\theta^o (\phi \mathbf{u})^o|_f + \theta^n (\phi \mathbf{u})^n|_f) \cdot \mathbf{S}_f \Delta t \quad (2.2.14)$$

in which θ^o , θ^n with $\theta^o + \theta^n = 1$ are coefficients that weight the old and new time step fields contributions, and $(\phi \mathbf{u})^o|_f$, $(\phi \mathbf{u})^n|_f$ are the face values of $\phi \mathbf{u}$

obtained via interpolation of either the old time step fields ϕ^o , \mathbf{u}^o or the new (i.e. current) time step fields ϕ^n , \mathbf{u}^n . In simpler terms, what equation 2.2.14 means is that a good approximation to $\int_t^{t+\Delta t} (\phi\mathbf{u})|_f dt$ is represented by: 1) $(\phi\mathbf{u})^o|_f \Delta t$ if $\theta^o = 1$; 2) $(\phi\mathbf{u})^n|_f \Delta t$ if $\theta^n = 1$; 3) a blending of the two for intermediate values of θ^o , θ^n .

By combining the results obtained by equations 2.2.6, 2.2.14 and dividing by the time step Δt , the time and space-integrated initial PDE which was formulated in terms of the continuous variables ϕ^* , \mathbf{u}^* . can be re-written in terms of the discrete cell and time step values ϕ , \mathbf{u} as:

$$\frac{V_c (\phi_c^n - \phi_c^o)}{\Delta t} + \sum_{f \in \partial\Omega_c} (\theta^o (\phi\mathbf{u})^o|_f + \theta^n (\phi\mathbf{u})^n|_f) \cdot \mathbf{S}_f = 0 \quad (2.2.15)$$

This equation is valid within each cell Ω_c , and its form emphasizes that the original continuous PDE problem is represented in each cell via a single algebraic relationship involving the old time step values $\phi_{c,nbr}^o$ and new time step values $\phi_{c,nbr}^n$, wherein by c , nbr we mean the cell Ω_c as well as its neighbouring cells, depending on the stencil used for face interpolations. If $\theta^o = 1$, the equation is said to be *explicit* as the unknown new time step value ϕ_c^n is algebraically related exclusively to the known old time step values $\phi_{c,nbr}^o$. Conversely, if $\theta^n = 1$, the problem is said to be *implicit* as ϕ_c^n now depends also on the (unknown) new time step values in the neighbouring cells $\phi_{c,nbr}^n$. The problem can be thus described as a linear system and can be solved accordingly via adequate techniques.

As it is clear by know, there is a considerable amount of aspects of different nature that partake in defining the process of numerically solving a set of PDEs, ranging from the construction and representation of the computational mesh, the representation of the fields and quantities of interest over said mesh, the schemes employed for the discretization in time and space of the PDEs, the algorithms for the solution of the linear systems represented by the individual discretized PDEs and practices for the evaluation of the quality of the obtained solutions. With regards to this last point, the analysis of the errors that are introduced by the domain and temporal discretization practices is of fundamental importance [21][22]. This is especially true in a coarse-mesh context, as it is discussed in chapter 5.

In order to deal with this level of baseline complexity related to the field of CFD and allow ourselves to shift the focus towards higher-level modelling, it was decided to rely on the extensive framework for Finite Volume modelling provided by the OpenFOAM C++ programming library. By higher-level modelling we mean: 1) the creation of the necessary numerical infrastructure required to represent specific phenomena in one and two-phase flows modelled in a coarse-mesh environment; 2) the solution algorithms to address the coupled nature of the governing equations (namely the Navier-Stokes equations and enthalpy transport equations); 3) further solution algorithms to address the coupling between the thermal-hydraulics and the other physics of interest when modelling nuclear systems, namely neutronics and thermal-mechanics.

2.2.4 The OpenFOAM programming library

The OpenFOAM (namely Open Field Operation And Manipulation) library is an open source C++ library and package of applications for the numerical modelling and solution of a variety of problems in physics via the FVM [19][23]. While it was originally designed for CFD calculations, the features of the library allowed it to be extended for the treatment of a variety of problems in physics ranging from e.g. solid-mechanics to magnetohydrodynamics. What the OpenFOAM library provides can be fundamentally classified as:

- a collection of *solvers* and *utilities*, both of which are fundamentally programs that act on user provided *cases*;
- a highly hierarchical and structured library of C++ custom *object types* (also known as *classes*) and functions for the numerical representation of the computational domain, mesh, fields, discretized PDEs, discretization and interpolation schemes, linear solvers. This is referred to as the OpenFOAM *framework*.

Solvers are programs that solve a certain implemented PDE or set of coupled PDEs to compute solution fields, while utilities are programs with a number of applications. These range from pre- and post-processing of fields and variables of interest to geometry and mesh generation and manipulation, as well as mesh conversion from popular mesh formats into OpenFOAM readable ones.

An important mechanism provided by the OpenFOAM framework provides is that of *run time selection*. This allows to specify and set a number of parameters at run time instead of compilation time, most importantly: term-by-term PDE discretization and interpolation schemes as well as the choice of linear solvers to be used for the solution of each discretized PDE. One of the greatest advantages of OpenFOAM is that said mechanism is embedded in the high-level syntax that can be used in the development of novel solvers. As an example, the solution of the time-dependent continuity equation for a generic quantity ϕ advected by a time-independent field \mathbf{u} presented in equation 2.2.2 can be implemented in a few lines of code as:

```
surfaceScalarField UfSf(fvc::flux(U));
while (runTime.run())
{
    runTime++;
    fvScalarMatrix contEqn
    (
        fvm::ddt(phi) + fvm::div(UfSf, phi) == 0
    );
    contEqn.solve();
    phi.correctBoundaryConditions();
    runTime.write();
}
```

The portions of code highlighted in red are custom object types in the OpenFOAM library that are meant to represent e.g. fields of scalars that exist on mesh faces (`surfaceScalarField`), sparse matrices that result from the discretization of a scalar PDE (`fvScalarMatrix`). The portions of the code highlighted in blue represent custom functions in the OpenFOAM library, e.g. for the explicit evaluation of certain quantities (i.e. the `fvc` *namespace*) or for the construction of the matrix coefficients representative of individual PDE terms (i.e. the `fvm` namespace). The portions of code highlighted in cyan represent *object methods*, namely functions that belong to certain objects¹ and that can operate on their member variables, e.g. the `solve()` method of an `fvScalarMatrix` type object, such as `contEqn`, proceeds to solve the linear system that said object represents.

In addition to a few further lines of code, not reported, for the inclusion of header files that contain the required object type and function declarations and definitions, as well as some lines for the initialization of the `runTime`, `U` and `phi` variables, the snippet provided above is all it takes to implement a solver capable of solving the equation of interest. This example showcases how the use of the OpenFOAM framework allows to abstract the physics of interest both from the computational domain over which the problem is to be solved as well as from the specifics of its numerical implementation, as the discretization, interpolation, linear system solution operations are conveniently encapsulated (in a broader and non-strictly programming-oriented sense) in a variety of object types, their methods and functions. All of those schemes are ultimately specified by the user at run time. Furthermore, as the library is open source, all of those object types, methods and functions are accessible by the user to further their understanding of the underlying mechanics and/or to modify those according to their needs.

While the overall learning curve in both utilizing existing OpenFOAM solvers as well as using the library to create new ones is rather steep, the advantages in terms of code flexibility, ease of maintenance and reliance on a modern programming paradigm were deemed highly desirable in the context of the present work.

2.3 The Coarse-mesh Methodology

The coarse-mesh methodology is an approach for modelling complex engineering systems with the purpose of being computationally lightweight while maintaining an acceptable degree of accuracy. It does so by borrowing from the mathematical tools commonly found in the domain of porous body theory [24], namely volume averaging techniques [25]. For this reason, the coarse-mesh approach is sometimes referred to as porous medium-based coarse-mesh approach, or simply porous body approach [26].

¹By *object* we mean a *variable* of a custom data type that does not belong to standard C++.

To illustrate the fundamental idea behind the approach, let us task ourselves with the simulation of fluid flow through a component such as the one represented in a) in Figure 2.3.1. It portrays a possible structure of interest, e.g. a bundle of pins wrapped by an outer can that can accommodate fluid flow along the pin axis direction.

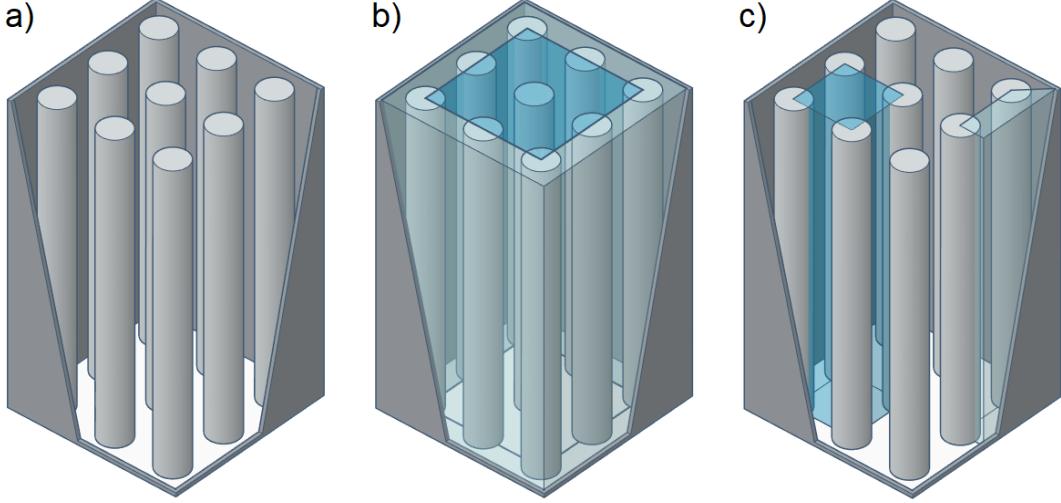


Figure 2.3.1: a) a generic structure of interest, hereby represented by a bundle of square-lattice pins wrapped by a can; b) definition of a REV in the bundle bulk (dark blue) and bundle edge (light blue); c) representation of the bundle bulk (blue) and bundle edge (light blue) regions whose averaged properties can be fully described by those computed over the corresponding representative elementary volume (REV)s defined in b).

In order to investigate the flow in such a geometry via a generic fine-mesh CFD approach or a sub-channel approach, the computational domain and mesh would extend exclusively over the volume occupied by the fluid, in between the pins, with the degree of geometric complexity and mesh density that this entails. A coarse-mesh approach would instead describe the system as a collection of representative volumes (RVs), namely volumes over which a number of geometric properties of the modelled structure are reasonably constant. While some of those quantities are discussed later in greater detail, these consist of:

- the *volume fraction* α_s of the structure, namely the ratio of structure volume encompassed by the RV to the total RV volume; its complementary $1 - \alpha_s$ is also known as the structure *porosity*;
- the surface area density A''_s , also known as structure *volumetric area*, namely the ratio between the total structure surface area in the RV and the total RV volume;
- the structure tortuosity tensor \mathbf{T} , namely a tensor quantifying the lengthening of diffusion lines.

Examples of possible RVs are highlighted in b) in Figure 2.3.1. For the specific example at hand, it is reasonable to assume that these or some of these properties will in principle differ between the bundle bulk and the bulk edge, which have been highlighted in blue and light blue respectively. In order to compute those quantities it is generally convenient to operate on the smallest possible portion of each RV that is still representative of the RV as a whole.

An example of this is provided in c) in Figure 2.3.1, as it is easy to see that, due to the structure regularity, the two volumes are still representative of the RVs of the corresponding color in b). Such volumes are generally known as REVs in virtue of the properties stated above. In simple terms, for each RV, a REV represent a basis polyhedron for a 3-D tessellation of the RV.

After the definition of the RVs and the volume averaging process for the calculation of relevant structure properties, a computational grid can be defined over each RV. In principle, from a modelling perspective, there are no particular limitations on the shape of the cells and as a consequence of the averaging process over the RVs, *they do not need to be conformal to any REV*. Practically, some limitations do nonetheless exist and are related to the numerical solution process. Fundamentally, these are related to:

- mesh convergence-related aspects: a mesh that is too coarse will provide results with unacceptable discretization errors and/or that are not in the *asymptotic convergence range*;
- performance-related aspects: a mesh that is overly fine will inevitably result in longer calculation times as well as additional limitations on the maximum size of the simulation time step due to the well-known Courant–Friedrichs–Lowy (CFL) condition;

Some aspects pertaining to mesh convergence are explored from both a theoretical and practical perspective in chapter 5. A further formal limitation is rather conceptual in nature and comes from the observation that an overly-fine mesh would defy the purpose of a coarse-mesh approach altogether.

After the definition of the RVs, their coarse-mesh properties and computational mesh, the governing equations can be solved over the grid. However, it is in principle not guaranteed that the governing equations that hold true at the fluid-intensive level (i.e. within a neighbourhood of a fluid parcel) will still hold true at the coarse-mesh cell level. This is due to the fact that, as a result of the volume averaging process, the mesh cells can in principle contain multiple phases (regardless of whether they are fluids or a solid structure). Borrowing from porous media theory nomenclature, these two levels are referred to as “microscopic” and “macroscopic” levels respectively. Volume averaging techniques are now discussed and applied to the governing equations, in a manner that is not dissimilar from the Finite Volume approach discussed in section 2.2.

2.3.1 Volume averaging techniques

Let us consider a generic domain Ω with volume V encompassing a portion Ω_γ with volume V_γ of a certain phase γ , along with possibly many other phases. Please consider that for the remainder of this sub-section, by “phase” we refer to any possible phase, i.e. both fluid *and* solid phases, unless otherwise specified. For any given microscopic or *local* (i.e. that is physically defined only over a certain phase) property ϕ_γ^* of the generic phase γ the following averaging operators are defined, namely the *intrinsic* average operator:

$$\langle \phi_\gamma^* \rangle_I = \frac{1}{V_\gamma} \int_{\Omega_\gamma} \phi_\gamma^* dV \quad (2.3.1)$$

and the *superficial* average operator:

$$\langle \phi_\gamma^* \rangle = \frac{1}{V} \int_{\Omega_\gamma} \phi_\gamma^* dV \quad (2.3.2)$$

In general, *extensive* physical quantities² are macroscopically described in terms of their superficial average, while *intensive* physical quantities³ are macroscopically described by their intrinsic average, as is commonly done in porous media theory [27].

The superficial average operator can be used to relate microscopically-defined quantities to their macroscopic equivalent (i.e. that are defined at the coarse-mesh level). In this chapter, the $*$ superscript is used to denote that a quantity is microscopic. Let us consider the indicator function of a given phase χ_γ^* defined as:

$$\chi_\gamma^*(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in \Omega_\gamma \\ 0 & \text{otherwise} \end{cases} \quad (2.3.3)$$

The superficial average of the indicator function yields the volume fraction of the given phase:

$$\alpha_\gamma \equiv \langle \chi_\gamma^* \rangle = \frac{V_\gamma}{V} \quad (2.3.4)$$

This allows to relate the intrinsic and superficial averages as:

$$\langle \phi_\gamma^* \rangle = \alpha_\gamma \langle \phi_\gamma^* \rangle_I \quad (2.3.5)$$

A further quantity that is introduced in the present context is the *deviation* $\tilde{\phi}_\gamma^*$ of the local quantity ϕ_γ^* with respect to the intrinsic average $\langle \phi_\gamma^* \rangle_I$:

$$\tilde{\phi}_\gamma^* = \phi_\gamma^* - \langle \phi_\gamma^* \rangle_I \quad (2.3.6)$$

²Quantities that depend on the amount of matter under investigation, e.g. mass, volume, energy.

³Quantities that do not depend on the amount of matter under investigation, e.g. density, pressure, temperature.

As previously stated, the governing equations at the coarse-mesh level are obtained via a superficial average of the microscopic ones, which are physically known. Due to the additivity of the averaging operator (as it consists of an integral), this equates to a term-by-term superficial average of the governing equations. However, depending on the quantity of interest governed by the equations, it could be expressed either in terms of its superficial or intrinsic average. In the next paragraphs, the main properties of the superficial and intrinsic average are thus explored before proceeding with the description of the fully averaged equations.

Volume average of a constant Assume that ϕ_γ^* is a local quantity of the fluid phase that is constant over Ω_γ . From the definition of intrinsic average in 2.3.1 one has:

$$\langle \phi_\gamma^* \rangle_I = \phi_\gamma^* \quad (2.3.7)$$

Then, it stems from equation 2.3.5 that:

$$\langle \phi_\gamma^* \rangle = \alpha_\gamma \phi_\gamma^* \quad (2.3.8)$$

Volume average of the deviation Since the averaging operator is additive, the superficial average of the deviation $\tilde{\phi}_\gamma$ is:

$$\langle \tilde{\phi}_\gamma \rangle = \langle \phi_\gamma^* \rangle - \langle \langle \phi_\gamma^* \rangle_I \rangle$$

By considering that the intrinsic average is by definition constant over Ω_γ , equation 2.3.7 can be employed to obtain:

$$\langle \tilde{\phi}_\gamma \rangle = \langle \phi_\gamma^* \rangle - \alpha_\gamma \langle \phi_\gamma^* \rangle_I$$

However, the relationship between superficial and intrinsic average (equation 2.3.5) then implies:

$$\langle \tilde{\phi}_\gamma \rangle = 0 \quad (2.3.9)$$

Such result is expected as the deviation from the average value of any quantity should vanish if averaged on the same volume on which the average is defined.

Volume average of a scalar product Let us consider two scalar quantities ϕ_γ^* and ψ_γ^* . By employing equation 2.3.6, the superficial average of the scalar product of the previously introduced ϕ_γ^* and ψ_γ^* can be written as:

$$\langle \phi_\gamma^* \psi_\gamma^* \rangle = \langle \langle \phi_\gamma^* \rangle_I \langle \psi_\gamma^* \rangle_I \rangle + \langle \langle \phi_\gamma^* \rangle_I \tilde{\psi}_\gamma \rangle + \langle \tilde{\phi}_\gamma \langle \psi_\gamma^* \rangle_I \rangle + \langle \tilde{\phi}_\gamma \tilde{\psi}_\gamma \rangle$$

Since the intrinsic averages are constant over Ω_γ , this can be rewritten as:

$$\langle \phi_\gamma^* \psi_\gamma^* \rangle = \alpha_\gamma \langle \phi_\gamma^* \rangle_I \langle \psi_\gamma^* \rangle_I + \langle \langle \phi_\gamma^* \rangle_I \rangle \langle \tilde{\psi}_\gamma \rangle + \langle \tilde{\phi}_\gamma \rangle \langle \langle \psi_\gamma^* \rangle_I \rangle + \langle \tilde{\phi}_\gamma \tilde{\psi}_\gamma \rangle$$

By considering that the superficial average of the deviation vanishes and that the intrinsic average and the superficial average are related by equation 2.3.5, the following is obtained:

$$\langle \phi_\gamma^* \psi_\gamma^* \rangle = \frac{1}{\alpha_\gamma} \langle \phi_\gamma^* \rangle \langle \psi_\gamma^* \rangle + \langle \tilde{\phi}_\gamma \tilde{\psi}_\gamma \rangle$$

The final formulation of the superficial average of a scalar product thus includes the superficial average of quantities which are essentially unknown, that are the deviations $\tilde{\phi}_\gamma$ and $\tilde{\psi}_\gamma$. However, for a sufficiently small choice of Ω it is reasonable to expect that the deviations will be significantly smaller than the superficial averages [27] so that $\langle \tilde{\phi}_\gamma \tilde{\psi}_\gamma \rangle \ll \langle \phi_\gamma^* \rangle \langle \psi_\gamma^* \rangle$. Thus:

$$\langle \phi_\gamma^* \psi_\gamma^* \rangle = \frac{1}{\alpha_\gamma} \langle \phi_\gamma^* \rangle \langle \psi_\gamma^* \rangle \quad (2.3.10)$$

while for the intrinsic average one has (equation 2.3.5):

$$\langle \phi_\gamma^* \psi_\gamma^* \rangle_I = \alpha_\gamma \langle \phi_\gamma^* \rangle_I \langle \psi_\gamma^* \rangle_I \quad (2.3.11)$$

Volume average of an inner or outer product Given two vector quantities ϕ_γ^* and ψ_γ^* , the derivation shown in the previous paragraphs for scalar quantities still holds true, so that:

$$\langle \phi_\gamma^* \cdot \psi_\gamma^* \rangle = \frac{1}{\alpha_\gamma} \langle \phi_\gamma^* \rangle \cdot \langle \psi_\gamma^* \rangle \quad (2.3.12)$$

$$\langle \phi_\gamma^* \cdot \psi_\gamma^* \rangle_I = \alpha_\gamma \langle \phi_\gamma^* \rangle_I \cdot \langle \psi_\gamma^* \rangle_I \quad (2.3.13)$$

$$\langle \phi_\gamma^* \otimes \psi_\gamma^* \rangle = \frac{1}{\alpha_\gamma} \langle \phi_\gamma^* \rangle \otimes \langle \psi_\gamma^* \rangle \quad (2.3.14)$$

$$\langle \phi_\gamma^* \otimes \psi_\gamma^* \rangle_I = \alpha_\gamma \langle \phi_\gamma^* \rangle_I \otimes \langle \psi_\gamma^* \rangle_I \quad (2.3.15)$$

Volume average of a spatial derivative As the final governing equation for the generic quantity ϕ_γ^* will be expressed in terms of either $\langle \phi_\gamma^* \rangle$ or $\langle \phi_\gamma^* \rangle_I$, the goal of the present paragraphs is to reformulate the volume average of a gradient or divergence of ϕ_γ^* into the gradient or divergence of $\langle \phi_\gamma^* \rangle$ or $\langle \phi_\gamma^* \rangle_I$.

The following derivation is conceptually the same in nature as those presented by Whitaker [25], Bear and Bachmat [28], yet it is revisited for completeness. Let us consider $\langle \nabla \phi_\gamma^* \rangle$. The results obtained for this example can be easily related to the other scenarios. By expanding the definition:

$$\langle \nabla \phi_\gamma^* \rangle = \frac{1}{V} \int_{\Omega_\gamma} \nabla \phi_\gamma^* dV \quad (2.3.16)$$

By taking advantage of the Stokes theorem to describe the volume integral of the gradient in terms of a surface integral over the boundary $\partial\Omega_\gamma$ of Ω_γ :

$$\int_{\Omega_\gamma} \nabla \phi_\gamma^* dV = \int_{\partial\Omega_\gamma} \phi_\gamma^* \mathbf{n} dS$$

with \mathbf{n} being the surface normal on $\partial\Omega_\gamma$ that is oriented outward from Ω_γ . It is recalled that Ω_γ is a sub-domain of the larger averaging domain Ω that is within phase γ . The boundary $\partial\Omega_\gamma$ can be decomposed into two parts, the boundary between the phase γ within Ω_γ and the same phase outside Ω_γ , which is referred to as $\partial\Omega_{\gamma\gamma}$, and the boundary between the phase γ within Ω_γ and the other phases outside of Ω_γ . For the time being, the discussion is limited to a system of two phases, which are labelled as γ and ζ , so that the latter boundary is denoted by $\partial\Omega_{\gamma\zeta}$. These phases might be both fluids, or a fluid and a structure phase. This is represented in Figure 2.3.2.

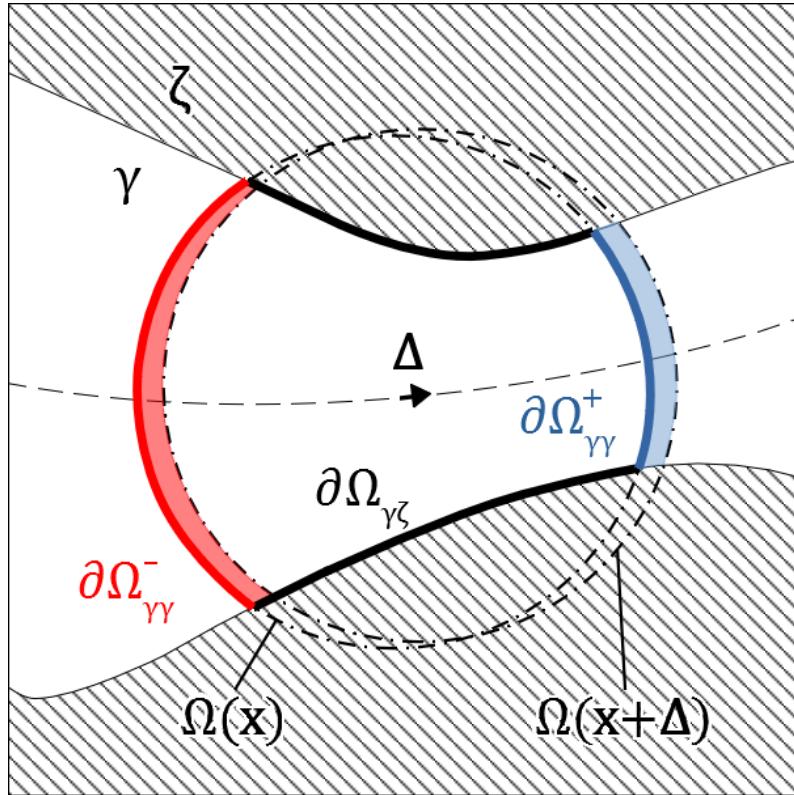


Figure 2.3.2: generic representation of two phases γ and ζ at the local or microscopic level. $\Omega(\mathbf{x})$ represents a portion of the domain centered around \mathbf{x} over which a certain average quantity of the γ phase, $\langle \phi_\gamma^* \rangle$ is computed. The domain $\Omega(\mathbf{x} + \Delta)$ represents an averaging volume translated with respect to $\Omega(\mathbf{x})$ by an amount Δ . This can be used for the calculation of $\nabla \langle \phi_\gamma^* \rangle$. The boundary of $\Omega_\gamma(\mathbf{x}) \subset \Omega(\mathbf{x})$ is decomposed into an intra-phase boundary $\partial\Omega_{\gamma\gamma} = \partial\Omega_{\gamma\gamma}^+ \cup \partial\Omega_{\gamma\gamma}^-$ (the red and blue outlines of $\Omega(\mathbf{x})$) and an inter-phase boundary $\partial\Omega_{\gamma\zeta}$ (the bold black lines within $\Omega(\mathbf{x})$) with $\partial\Omega_\gamma = \partial\Omega_{\gamma\gamma} \cup \partial\Omega_{\gamma\zeta}$.

The boundary integral can be thus expressed as:

$$\int_{\partial\Omega_\gamma} \phi_\gamma^* \mathbf{n} \, dS = \int_{\partial\Omega_{\gamma\gamma}} \phi_\gamma^* \mathbf{n} \, dS + \int_{\partial\Omega_{\gamma\zeta}} \phi_\gamma^* \mathbf{n} \, dS$$

so that equation 2.3.16 becomes:

$$\langle \nabla \phi_\gamma^* \rangle = \frac{1}{V} \left(\int_{\partial\Omega_{\gamma\gamma}} \phi_\gamma^* \mathbf{n} \, dS + \int_{\partial\Omega_{\gamma\zeta}} \phi_\gamma^* \mathbf{n} \, dS \right) \quad (2.3.17)$$

Let us focus our attention on the first surface integral on the right-hand side (RHS) of equation 2.3.17. Particularly, it is now shown that $\nabla \langle \phi_\gamma^* \rangle$, which is hereby derived, is equivalent to it.

Let us start by computing the directional derivative $\nabla_\Delta \langle \phi_\gamma^* \rangle$ in a certain direction represented by the unit vector \mathbf{e}_Δ . From the definition of directional derivative:

$$\mathbf{e}_\Delta \cdot \nabla \langle \phi_\gamma^* \rangle \equiv \nabla_\Delta \langle \phi_\gamma^* \rangle = \lim_{\Delta \rightarrow 0} \frac{\langle \phi_\gamma^* \rangle(\mathbf{x} + \Delta) - \langle \phi_\gamma^* \rangle(\mathbf{x})}{\Delta} \quad (2.3.18)$$

where $\langle \phi_\gamma^* \rangle(\mathbf{x})$ is the average of ϕ_γ^* as defined in equation 2.3.2 computed in a domain $\Omega(\mathbf{x})$ that is centered on \mathbf{x} . Please note that $\langle \phi_\gamma^* \rangle(\mathbf{x})$ is continuous. The vectors $\Delta = \Delta \mathbf{e}_\Delta$ denote the increment vector of magnitude Δ . By taking advantage of the definition of the superficial average in equation 2.3.2, equation 2.3.18 can be expanded as:

$$\begin{aligned} & \lim_{\Delta \rightarrow 0} \frac{\langle \phi_\gamma^* \rangle(\mathbf{x} + \Delta) - \langle \phi_\gamma^* \rangle(\mathbf{x})}{\Delta} \\ &= \lim_{\Delta \rightarrow 0} \frac{1}{\Delta} \left(\int_{\Omega_\gamma(\mathbf{x} + \Delta)} \phi_\gamma^* \, dV - \int_{\Omega_\gamma(\mathbf{x})} \phi_\gamma^* \, dV \right) \end{aligned} \quad (2.3.19)$$

As the integrals in equation 2.3.19 both operate on the same quantity, have opposite signs, and are performed over partially overlapping domains, the net result consists in the integration of said quantity only over certain sub-domains of the respective domains, so that

$$\begin{aligned} & \lim_{\Delta \rightarrow 0} \frac{1}{\Delta} \left(\int_{\Omega_\gamma(\mathbf{x} + \Delta)} \phi_\gamma^* \, dV - \int_{\Omega_\gamma(\mathbf{x})} \phi_\gamma^* \, dV \right) \\ &= \lim_{\Delta \rightarrow 0} \frac{1}{\Delta} \left(\int_{\Omega_\gamma(\mathbf{x} + \Delta) \setminus \Omega_\gamma(\mathbf{x})} \phi_\gamma^* \, dV - \int_{\Omega_\gamma(\mathbf{x}) \setminus \Omega_\gamma(\mathbf{x} + \Delta)} \phi_\gamma^* \, dV \right) \end{aligned} \quad (2.3.20)$$

with $\Omega_\gamma(\mathbf{x} + \Delta) \setminus \Omega_\gamma(\mathbf{x})$ representing the domain resulting from the subtraction of $\Omega_\gamma(\mathbf{x})$ from $\Omega_\gamma(\mathbf{x} + \Delta)$, highlighted in a light shade of blue in Figure 2.3.2, and $\Omega_\gamma(\mathbf{x}) \setminus \Omega_\gamma(\mathbf{x} + \Delta)$ representing the domain resulting from the subtraction of $\Omega_\gamma(\mathbf{x} + \Delta)$ from $\Omega_\gamma(\mathbf{x})$, highlighted in a light shade of red in Figure 2.3.2. In the considered limit of $\Delta \rightarrow 0$ and with reference to the aforementioned

figure, the volume differentials dV over $\Omega_\gamma(\mathbf{x} + \Delta) \setminus \Omega_\gamma(\mathbf{x})$ can be obtained by extruding the infinitesimal surface dS defined over the boundary $\partial\Omega_{\gamma\gamma}^+$ in the direction of Δ . Mathematically, this can be expressed as:

$$\lim_{\Delta \rightarrow 0} \int_{\Omega_\gamma(\mathbf{x} + \Delta) \setminus \Omega_\gamma(\mathbf{x})} \phi_\gamma^* dV = \lim_{\Delta \rightarrow 0} \int_{\partial\Omega_{\gamma\gamma}^+} \phi_\gamma^* \Delta \cdot \mathbf{n} dS \quad (2.3.21)$$

where it is recalled \mathbf{n} is the outward normal of $\partial\Omega_\gamma$ at dS . With regard to $\Omega_\gamma(\mathbf{x}) \setminus \Omega_\gamma(\mathbf{x} + \Delta)$, considering that the surface normal \mathbf{n} on $\partial\Omega_{\gamma\gamma}^-$ is always⁴ such that $\mathbf{n} \cdot \Delta < 0$, one can similarly find:

$$\lim_{\Delta \rightarrow 0} \int_{\Omega_\gamma(\mathbf{x}) \setminus \Omega_\gamma(\mathbf{x} + \Delta)} \phi_\gamma^* dV = - \lim_{\Delta \rightarrow 0} \int_{\partial\Omega_{\gamma\gamma}^-} \phi_\gamma^* \Delta \cdot \mathbf{n} dS \quad (2.3.22)$$

By combining the results of equations 2.3.21, 2.3.22 with 2.3.20 and recalling that $\Delta = \Delta \mathbf{e}_\Delta$:

$$\begin{aligned} & \lim_{\Delta \rightarrow 0} \frac{1}{\Delta} \left(\int_{\Omega_\gamma(\mathbf{x} + \Delta)} \phi_\gamma^* dV - \int_{\Omega_\gamma(\mathbf{x})} \phi_\gamma^* dV \right) \\ &= \lim_{\Delta \rightarrow 0} \frac{1}{\Delta} \left(\int_{\partial\Omega_{\gamma\gamma}^+} \phi_\gamma^* \Delta \cdot \mathbf{n} dS + \int_{\partial\Omega_{\gamma\gamma}^-} \phi_\gamma^* \Delta \cdot \mathbf{n} dS \right) \\ &= \lim_{\Delta \rightarrow 0} \frac{1}{\Delta} \left(\int_{\partial\Omega_{\gamma\gamma}} \phi_\gamma^* \Delta \cdot \mathbf{n} dS \right) \\ &= \lim_{\Delta \rightarrow 0} \frac{\Delta}{\Delta} \mathbf{e}_\Delta \cdot \left(\int_{\partial\Omega_{\gamma\gamma}} \phi_\gamma^* \mathbf{n} dS \right) \\ &= \mathbf{e}_\Delta \cdot \left(\int_{\partial\Omega_{\gamma\gamma}} \phi_\gamma^* \mathbf{n} dS \right) \end{aligned} \quad (2.3.23)$$

where the fact that the sum of the surface integrals of the same quantity over two separate sub-domains $\partial\Omega_{\gamma\gamma}^+$ and $\partial\Omega_{\gamma\gamma}^-$ is equal to a single surface integral of the same quantity of the domain resulting from their union, namely $\partial\Omega_{\gamma\gamma} = \partial\Omega_{\gamma\gamma}^+ \cup \partial\Omega_{\gamma\gamma}^-$, was taken advantage of. By combining the results of equation 2.3.23 with 2.3.19, 2.3.18:

$$\mathbf{e}_\Delta \cdot \nabla \langle \phi_\gamma^* \rangle = \mathbf{e}_\Delta \cdot \left(\frac{1}{V} \int_{\partial\Omega_{\gamma\gamma}} \phi_\gamma^* \mathbf{n} dS \right) \quad (2.3.24)$$

which implies:

$$\nabla \langle \phi_\gamma^* \rangle = \frac{1}{V} \int_{\partial\Omega_{\gamma\gamma}} \phi_\gamma^* \mathbf{n} dS \quad (2.3.25)$$

⁴While not explicitly stated so far, that is precisely how $\partial\Omega_{\gamma\gamma}^+$ and $\partial\Omega_{\gamma\gamma}^-$ are defined, namely $\partial\Omega_{\gamma\gamma}^+$ is that sub-domain of $\partial\Omega_{\gamma\gamma}$ such that $\mathbf{n} \cdot \Delta > 0$ and $\partial\Omega_{\gamma\gamma}^-$ is that sub-domain of $\partial\Omega_{\gamma\gamma}$ such that $\mathbf{n} \cdot \Delta < 0$

Returning back to the original problem of interest, namely defining $\langle \nabla \phi_\gamma^* \rangle$ in relation to $\nabla \langle \phi_\gamma^* \rangle$, by combining equations 2.3.25 and 2.3.17 the final result is obtained:

$$\langle \nabla \phi_\gamma^* \rangle = \nabla \langle \phi_\gamma^* \rangle + \frac{1}{V} \int_{\partial\Omega_{\gamma\zeta}} \phi_\gamma^* \mathbf{n} \, dS \quad (2.3.26)$$

By employing a very similar approach, the average gradient and divergence of a tensor quantity ϕ_γ^* of order greater than zero (e.g. a vector) can be obtained as:

$$\langle \nabla \phi_\gamma^* \rangle = \nabla \langle \phi_\gamma^* \rangle + \frac{1}{V} \int_{\partial\Omega_{\gamma\zeta}} \phi_\gamma^* \otimes \mathbf{n} \, dS \quad (2.3.27)$$

$$\langle \nabla \cdot \phi_\gamma^* \rangle = \nabla \cdot \phi_\gamma^* + \frac{1}{V} \int_{\partial\Omega_{\gamma\zeta}} \phi_\gamma^* \cdot \mathbf{n} \, dS \quad (2.3.28)$$

As stated earlier, while the governing equations at the coarse-mesh level are obtained via a superficial average on a term-by-term basis, the final variables of interest for which the coarse-mesh governing equations are solved will more often than not be represented by the intrinsic averages of the fields of interest rather than their superficial averages. It is thus useful to express e.g. $\langle \nabla \phi_\gamma^* \rangle$ in terms of $\nabla \langle \phi_\gamma^* \rangle_I$. In virtue of equation 2.3.5 which relates a superficial average to the intrinsic average, one has:

$$\langle \nabla \phi_\gamma^* \rangle = \alpha_\gamma \langle \nabla \phi_\gamma^* \rangle_I = \nabla \left(\alpha_\gamma \langle \phi_\gamma^* \rangle_I \right) + \frac{1}{V} \int_{\partial\Omega_{\gamma\zeta}} \phi_\gamma^* \mathbf{n} \, dS \quad (2.3.29)$$

In form, this is also valid for the intrinsic average of equations 2.3.27, 2.3.28, not reported for brevity. It is however clear from the form of these equations that one requires a closure for the last term on the RHS of these relationships. In particular, this term reformulates the local inter-phase interaction (which takes place on the inter-phase boundary $\partial\Omega_{\gamma\zeta}$) into a volumetric interaction term “spread out” over Ω . The physical significance of this term and its modelling is introduced in general terms in subsection 2.3.2 and further expanded in chapter 3.

With specific regard to the average of a gradient, it is nonetheless possible to reformulate equation 2.3.29 in terms of $\nabla \langle \phi_\gamma^* \rangle_I$ with some approximations and with knowledge of the inter-phase boundary geometry $\partial\Omega_{\gamma\zeta}$ (which is known e.g. for a system consisting of a fluid and a solid non-moving phase). This formulation introduces the concept of *tortuosity*, a second order tensor which is a geometrical property resulting exclusively from the distribution of the phase of interest within the representative volume. Essentially, it quantifies the fact that $\langle \nabla \phi_\gamma^* \rangle$ and $\nabla \langle \phi_\gamma^* \rangle_I$ are not necessarily parallel. By considering the example provided by a simple diffusion problem for a quantity ϕ_γ^* over a coarse-mesh, the concept of tortuosity thus quantifies how the spatial arrangement of the phases distorts the lines along which ϕ_γ^* diffuses with respect to a single-phase scenario in which only γ is present in the domain. While this concept is of limited to no practical applicability in a two-phase

system with two fluid phases and no static structure phase (due to the possibly ever-changing configuration of the phases within each averaging volume), it is of significant importance when describing diffusion-dominated phenomena of one-fluid-phase flows through structures treated via a coarse-mesh approach. Remarks on how this can be extended to multi-fluid-phase flows through such structures are presented later in subsection 3.2.5 in chapter 3, as it is shown that this topic can be tied to the broader discussion of turbulence modelling.

Let us begin by calculating the superficial average of the indicator function χ_γ^* for a generic phase γ in a two-phase system. By taking advantage of the definition of phase fraction α_γ provided by equation 2.3.4 and considering that it follows from the definition of χ_γ^* in equation 2.3.3 that $\chi_\gamma^*(\mathbf{x}) = 1 \forall \mathbf{x} \in \partial\Omega_{\gamma\zeta}$, one can use 2.3.26 to derive:

$$\langle \nabla \chi_\gamma^* \rangle = \nabla \alpha_\gamma + \frac{1}{V} \int_{\partial\Omega_{\gamma\zeta}} \mathbf{n} \, dS \quad (2.3.30)$$

However, in virtue of the definition of χ_γ^* , it follows that $\nabla \chi_\gamma^*(\mathbf{x}) = 0 \forall \mathbf{x} \in \Omega_\gamma$, so that $\langle \nabla \chi_\gamma^* \rangle_I = 0$ and consequently $\langle \nabla \chi_\gamma^* \rangle = 0$ due to equation 2.3.29. Equation 2.3.30 can be thus used to relate the gradient of the phase fraction to the spatial configuration of the inter-phase boundary in Ω :

$$\nabla \alpha_\gamma = -\frac{1}{V} \int_{\partial\Omega_{\gamma\zeta}} \mathbf{n} \, dS \quad (2.3.31)$$

Taking advantage of equation 2.3.31 and taking advantage of the fact that $\langle \phi_\gamma^* \rangle_I$ is constant within Ω , 2.3.29 can be expanded as follows:

$$\begin{aligned} \langle \nabla \phi_\gamma^* \rangle &= \nabla \left(\alpha_\gamma \langle \phi_\gamma^* \rangle_I \right) + \frac{1}{V} \int_{\partial\Omega_{\gamma\zeta}} \phi_\gamma^* \mathbf{n} \, dS \\ &= \alpha_\gamma \nabla \langle \phi_\gamma^* \rangle_I + \langle \phi_\gamma^* \rangle_I \nabla \alpha_\gamma + \frac{1}{V} \int_{\partial\Omega_{\gamma\zeta}} \phi_\gamma^* \mathbf{n} \, dS \\ &= \alpha_\gamma \nabla \langle \phi_\gamma^* \rangle_I - \frac{1}{V} \langle \phi_\gamma^* \rangle_I \int_{\partial\Omega_{\gamma\zeta}} \mathbf{n} \, dS + \frac{1}{V} \int_{\partial\Omega_{\gamma\zeta}} \phi_\gamma^* \mathbf{n} \, dS \\ &= \alpha_\gamma \nabla \langle \phi_\gamma^* \rangle_I + \frac{1}{V} \int_{\partial\Omega_{\gamma\zeta}} \tilde{\phi}_\gamma \mathbf{n} \, dS \end{aligned} \quad (2.3.32)$$

It is recalled that $\tilde{\phi}_\gamma = \phi_\gamma^* - \langle \phi_\gamma^* \rangle_I$ is the deviation of $\langle \phi_\gamma^* \rangle_I$ as defined in equation 2.3.6. To further simplify equation 2.3.32, let us expand ϕ_γ^* with a first-order Taylor series around the centroid \mathbf{x}_0 of the averaging volume Ω :

$$\phi_\gamma^*(\mathbf{x}) \approx \phi_\gamma^*(\mathbf{x}_0) + \nabla \phi_\gamma^*|_{\mathbf{x}_0} \cdot (\mathbf{x} - \mathbf{x}_0) \quad (2.3.33)$$

where the centroid is defined geometrically as:

$$\mathbf{x}_0 = \frac{1}{V} \int_{\Omega_\gamma} \mathbf{x} \, dV \quad (2.3.34)$$

Precisely as \mathbf{x}_0 is the cell centroid, it is reasonable to assume that $\langle \phi_\gamma^* \rangle_I$ and $\langle \nabla \phi_\gamma^* \rangle_I = \frac{1}{\alpha_\gamma} \langle \nabla \phi_\gamma^* \rangle$ are representative of $\phi_\gamma^*(\mathbf{x}_0)$ and $\nabla \phi_\gamma^*|_{\mathbf{x}_0}$ respectively. Thus, following from equation 2.3.33, the deviation can be approximated at a first order as:

$$\tilde{\phi}_\gamma \approx \frac{1}{\alpha_\gamma} \langle \nabla \phi_\gamma^* \rangle \cdot (\mathbf{x} - \mathbf{x}_0) \quad (2.3.35)$$

By considering that $\langle \nabla \phi_\gamma^* \rangle$ is a constant within Ω_γ and that the following identity $(\mathbf{a} \cdot \mathbf{b}) \mathbf{c} = (\mathbf{c} \otimes \mathbf{b}) \cdot \mathbf{a}$ holds true for any three vectors $\mathbf{a}, \mathbf{b}, \mathbf{c}$, it is easy to show that 2.3.32 becomes:

$$\begin{aligned} \langle \nabla \phi_\gamma^* \rangle &= \alpha_\gamma \nabla \langle \phi_\gamma^* \rangle_I + \frac{1}{V} \int_{\partial\Omega_{\gamma\zeta}} \frac{1}{\alpha_\gamma} \langle \nabla \phi_\gamma^* \rangle \cdot (\mathbf{x} - \mathbf{x}_0) \mathbf{n} dS \\ &= \alpha_\gamma \nabla \langle \phi_\gamma^* \rangle_I + \left(\frac{1}{V_\gamma} \int_{\partial\Omega_{\gamma\zeta}} \mathbf{n} \otimes (\mathbf{x} - \mathbf{x}_0) dS \right) \cdot \langle \nabla \phi_\gamma^* \rangle \\ \Rightarrow \langle \nabla \phi_\gamma^* \rangle &= \alpha_\gamma \mathbf{T}_\gamma \nabla \langle \phi_\gamma^* \rangle_I \end{aligned} \quad (2.3.36)$$

in which the *tortuosity tensor* \mathbf{T}_γ was defined as:

$$\mathbf{T}_\gamma^{-1} = \mathbb{I} - \frac{1}{V_\gamma} \int_{\partial\Omega_{\gamma\zeta}} \mathbf{n} \otimes (\mathbf{x} - \mathbf{x}_0) dS \quad (2.3.37)$$

Please note that the tortuosity is only specific to a single phase γ and not to a phase pair $\gamma - \zeta$ as the surface integral in equation 2.3.37 might seem to imply. As a matter of fact, the derivation presented hereby holds in multi-phase scenarios as well by considering that $\partial\Omega_{\gamma\zeta}$ consists of the boundary between γ and *all* the other phases other than γ (i.e. by collectively denoting all the other phases as ζ).

The result obtained in 2.3.37 is an extension of the tortuosity concept as presented by Lehner [29], Whitaker [30] and Gray [31] all of whom reach expressions that correspond to 2.3.32 yet without further expanding the deviation by taking advantage of a Taylor series. Due to this, the practical applicability of their formulation is limited [30]. However, further work by Kim and Whitaker employed a similar approach to the one adopted here [32], yet arrived at an arguably more complex formulation by expanding $\phi_\gamma^*(\mathbf{x})$ in terms of $\nabla \langle \phi_\gamma^* \rangle_I$ rather than the more natural $\langle \nabla \phi_\gamma^* \rangle$. Finally, an approximation of the tortuosity is proposed by Bear and Bachmat [28], which is more numerically convenient when actually applying these expressions for calculating the lattice tortuosity:

$$\mathbf{T}_\gamma = 3 \frac{\alpha_\gamma^S}{\alpha_\gamma S_\gamma} \int_{\partial\Omega_{\gamma\gamma}} \mathbf{n} \otimes \mathbf{n} dS \quad (2.3.38)$$

In which α_γ^S is the phase fraction at the boundary (i.e. the surface average of χ_γ over $\partial\Omega$), which might in principle differ from α_γ^S and S_γ is the total surface of the intra-phase boundary $\partial\Omega_{\gamma\gamma}$.

Volume average of a time derivative In a similar way as to that presented for the derivation of the average of spatial derivatives, one can obtain that the average of a time derivative of a quantity ϕ_γ^* as:

$$\langle \frac{\partial}{\partial t} \phi_\gamma^* \rangle = \frac{\partial}{\partial t} \langle \phi_\gamma^* \rangle - \frac{1}{V} \int_{\partial\Omega_{\gamma\zeta}} \phi_\gamma^* \mathbf{u}_\partial^* \cdot \mathbf{n} \, dS \quad (2.3.39)$$

in which \mathbf{u}_∂^* is the velocity of the interface. Similarly to what was discussed in the previous paragraph, this can be expressed in terms of the time derivative of the intrinsic average as:

$$\langle \frac{\partial}{\partial t} \phi_\gamma^* \rangle = \alpha_\gamma \langle \frac{\partial}{\partial t} \phi_\gamma^* \rangle_I = \frac{\partial}{\partial t} (\alpha_\gamma \langle \phi_\gamma^* \rangle_I) - \frac{1}{V} \int_{\partial\Omega_{\gamma\zeta}} \phi_\gamma^* \mathbf{u}_\partial^* \cdot \mathbf{n} \, dS \quad (2.3.40)$$

2.3.2 Volume average of a transport equation

For the sake of generality we now task ourselves with the construction of a macroscopically-valid (i.e. coarse-mesh valid) transport equation for a generic fluid quantity ϕ_γ^* in a multi-phase context. The symbol ζ is now treated as a generic phase index to represent any phase other than γ . No particular assumptions are made on the nature of the individual phases, which might be fluids or solids. It is important to recall that all of the results concerning the volume average of different quantities and operators for a generic phase γ presented in subsection 2.3.1 are equally valid in a multi-phase context. This is easy to see by considering that what was denoted by ζ in that subsection can be seen as the collection of all phases other than γ .

Let us start by considering the local transport equation for ϕ_γ^* over an advective field \mathbf{u}_γ^* , diffused by a flux \mathbf{J}_γ^* and affected by an additional body source term b_γ^* :

$$\frac{\partial}{\partial t} \phi_\gamma^* + \nabla \cdot (\mathbf{u}_\gamma^* \phi_\gamma^*) = \nabla \cdot \mathbf{J}_\gamma^* + b_\gamma^* \quad (2.3.41)$$

If ϕ_γ^* is a tensor quantity of order greater than zero, the advective term is described as $\nabla \cdot (\mathbf{u}_\gamma^* \otimes \phi_\gamma^*)$. Furthermore, \mathbf{J}_γ^* will be a tensor quantity of an order that is greater than that of ϕ_γ^* by one. A macroscopic governing equation is typically formulated in terms of the intrinsic average of ϕ_γ^* , and for notation simplicity it is referred to simply as ϕ , namely:

$$\phi_\gamma \equiv \langle \phi_\gamma^* \rangle_I \quad (2.3.42)$$

The same notation is applied to all other variables. To obtain the macroscopic governing equation in a domain Ω of volume V , the superficial average of 2.3.41 needs to be taken. This amounts to a term-by-term superficial average. By considering all of the results obtained in subsection 2.3.1 and by considering

the notation introduced in 2.3.42:

$$\langle \frac{\partial}{\partial t} \phi_\gamma^* \rangle = \frac{\partial}{\partial t} (\alpha_\gamma \phi_\gamma) - \frac{1}{V} \sum_{\zeta \neq \gamma} \left(\int_{\partial\Omega_{\gamma\zeta}} \phi_\gamma^* \mathbf{u}_\partial^* \cdot \mathbf{n} \, dS \right) \quad (2.3.43)$$

$$\langle \nabla \cdot (\mathbf{u}_\gamma^* \phi_\gamma^*) \rangle = \nabla \cdot (\alpha_\gamma \mathbf{u}_\gamma \phi_\gamma) + \frac{1}{V} \sum_{\zeta \neq \gamma} \left(\int_{\partial\Omega_{\gamma\zeta}} \phi_\gamma^* \mathbf{u}_\gamma^* \cdot \mathbf{n} \, dS \right) \quad (2.3.44)$$

$$\langle \nabla \cdot \mathbf{J}_\gamma^* \rangle = \nabla \cdot (\alpha \mathbf{J}_\gamma) + \frac{1}{V} \sum_{\zeta \neq \gamma} \left(\int_{\partial\Omega_{\gamma\zeta}} \mathbf{J}_\gamma^* \cdot \mathbf{n} \, dS \right) \quad (2.3.45)$$

$$\langle b_\gamma^* \rangle = \alpha_\gamma b_\gamma \quad (2.3.46)$$

where it is recalled that \mathbf{n} is the boundary normal oriented from γ towards ζ . By taking advantage of these averages, the original transport equation can be re-written as:

$$\frac{\partial}{\partial t} (\alpha_\gamma \phi_\gamma) + \nabla \cdot (\alpha_\gamma \mathbf{u}_\gamma \phi_\gamma) = \nabla \cdot (\alpha \mathbf{J}_\gamma) + \alpha_\gamma b_\gamma - \sum_{\zeta \neq \gamma} (b_{\alpha,\gamma \rightarrow \zeta} + b_{\mathbf{J},\gamma \rightarrow \zeta}) \quad (2.3.47)$$

where the additional volumetric source terms are defined as:

$$b_{\alpha,\gamma \rightarrow \zeta} = -\frac{1}{V} \int_{\partial\Omega_{\gamma\zeta}} \phi_\gamma^* (\mathbf{u}_\partial^* - \mathbf{u}_\gamma^*) \cdot \mathbf{n} \, dS \quad (2.3.48)$$

$$b_{\mathbf{J},\gamma \rightarrow \zeta} = -\frac{1}{V} \int_{\partial\Omega_{\gamma\zeta}} \mathbf{J}_\gamma^* \cdot \mathbf{n} \, dS \quad (2.3.49)$$

For completeness, if ϕ_γ is a tensor quantity of an order greater than zero, the governing macroscopic equation becomes:

$$\begin{aligned} \frac{\partial}{\partial t} (\alpha_\gamma \phi_\gamma) + \nabla \cdot (\alpha_\gamma \mathbf{u}_\gamma \otimes \phi_\gamma) &= \nabla \cdot (\alpha \mathbf{J}_\gamma) + \alpha_\gamma b_\gamma + \\ &\quad - \sum_{\zeta \neq \gamma} (\mathbf{b}_{\alpha,\gamma \rightarrow \zeta} + \mathbf{b}_{\mathbf{J},\gamma \rightarrow \zeta}) \end{aligned} \quad (2.3.50)$$

with:

$$\mathbf{b}_{\alpha,\gamma \rightarrow \zeta} = -\frac{1}{V} \int_{\partial\Omega_{\gamma\zeta}} ((\mathbf{u}_\partial^* - \mathbf{u}_\gamma^*) \otimes \phi_\gamma) \mathbf{n} \, dS \quad (2.3.51)$$

$$\mathbf{b}_{\mathbf{J},\gamma \rightarrow \zeta} = -\frac{1}{V} \int_{\partial\Omega_{\gamma\zeta}} \mathbf{J}_\gamma^* \cdot \mathbf{n} \, dS \quad (2.3.52)$$

Formally, the macroscopic governing equation (2.3.47) and the locally governing one (2.3.41) are very similar if not for the presence of the phase fraction α_γ and the additional volumetric source terms $b_{\alpha,\gamma \rightarrow \zeta}$ and $b_{\mathbf{J},\gamma \rightarrow \zeta}$. These quantify the change of ϕ_γ^* due to the interaction of phase γ with the other phases within Ω . In particular, $b_{\alpha,\gamma \rightarrow \zeta}$ represents the variation due to the appearance/disappearance of phase γ (i.e. phase change), while $b_{\mathbf{J},\gamma \rightarrow \zeta}$ is associated with the diffusion of ϕ_γ^* across the interface. These terms always require some form of closure in a coarse-mesh context, as the local domain is never geometrically resolved and knowledge of the microscopic fields is not available.

2.3.3 Volume average of the Navier-Stokes equations

In order to bridge the theoretical framework presented so far with its implementation and application to the analysis of fluid flows, the general form of the volume averaged Navier-Stokes equations for a generic system of a number of fluid phases and a single solid, immovable phase is presented. The fluid phase under consideration is denoted via the i subscript, while the variable fluid phase index is j , and the structure phase index is s .

Mass conservation equation This governing equation, also known as the continuity equation, consists of the transport equation (2.3.41) for the specific mass (i.e. density) ρ_i^* of the fluid under exam, with $\mathbf{J}_i^* = 0$ and $b_i^* = 0$. One might wonder why, in a generic multi-phase scenario, the local density source term b_i^* is zero, but that is justified by the fact that inter-phase mass transfer physically takes place exclusively at phase boundaries, and that the local microscopic balance equations hold true only within a volume where one phase is present. As by definition no inter-phase boundaries are presented in such volume, the local density source term is necessarily zero. However, at a macroscopic level, the mass transfer terms appears as a consequence of the averaging procedure. By substituting the relevant quantities in the volume-averaged transport equation 2.3.47, the macroscopic continuity equation for fluid i becomes:

$$\frac{\partial}{\partial t} (\alpha_i \rho_i) + \nabla \cdot (\alpha_i \mathbf{u}_i \rho_i) = - \sum_{j \neq i} \Gamma_{i \rightarrow j} \quad (2.3.53)$$

In which the inter-phase specific mass transfer $\Gamma_{i \rightarrow j}$ was introduced, equivalent to $b_{\alpha,i \rightarrow j}$ as defined in 2.3.48, namely:

$$\Gamma_{i \rightarrow j} = -\frac{1}{V} \int_{\partial\Omega_{ij}} \rho_i^* (\mathbf{u}_\partial^* - \mathbf{u}_i^*) \cdot \mathbf{n} \, dS \quad (2.3.54)$$

On top of this, we *impose* the additional physically-motivated constraint that $\Gamma_{i \rightarrow j} = -\Gamma_{j \rightarrow i}$. The mass transfer term is typically modelled on the basis of thermal considerations, as it is discussed in chapter 3. Please note that as the phase indices for $b_{\alpha,\gamma \rightarrow \zeta}$ in equation 2.3.47 span all the phases, regardless of them being fluids or solids, the term $\Gamma_{i \rightarrow s}$ should be present as well in the macroscopic continuity equation. However, as long as the fluid and the solid do not belong to the same chemical species and no phase change is possible, the no-slip condition always holds at a fluid-solid boundary (within the limits of the continuum hypothesis, which is assumed valid throughout this work), so that $\mathbf{u}_\partial = \mathbf{u}_i$ at the $\partial\Omega_{is}$ boundary and $\Gamma_{i \rightarrow s} = 0$.

Momentum conservation equation This equation is equivalent to the transport of the specific momentum of a phase. Its microscopic formulation follows from equation 2.3.41 with $\phi_i^* = \rho_i^* \mathbf{u}_i^*$ the volumetric momentum, $\mathbf{J}_i^* =$

$\boldsymbol{\sigma}_i^*$ the fluid stress tensor, and $b_i^* = \rho_i^* \mathbf{g}$ the gravitational acceleration. Only Newtonian fluids are considered in this work, and by the Stokes hypothesis the local stress tensor is expressed as:

$$\boldsymbol{\sigma}_i^* = -p^* \mathbb{I} + \boldsymbol{\sigma}_{d,i}^* \quad (2.3.55)$$

$$\boldsymbol{\sigma}_{d,i}^* = \mu_i^* \left(\nabla \mathbf{u}_i^* + (\nabla \mathbf{u}_i^*)^T - \frac{2}{3} (\nabla \cdot \mathbf{u}_i^*) \mathbb{I} \right) \quad (2.3.56)$$

where $\boldsymbol{\sigma}_{d,i}^*$ is the deviatoric component of the stress tensor and μ_i^* is the molecular viscosity of the fluid. The formulation of the stress tensor provided above assumes that all the phases in the system share a single pressure p , which is a widespread assumption when dealing with *dispersed* multiphase flows. Under this assumption it is also reasonable to assume that $p \equiv \langle p^* \rangle_I \approx p^*$. Then, it can be shown by employing the tools developed in 2.3.1 that the volume average of the divergence of the stress tensor is:

$$\langle \nabla \cdot \boldsymbol{\sigma}_i^* \rangle = -\alpha_i \nabla p + \nabla \cdot (\alpha_i \boldsymbol{\sigma}_{d,i}) + \frac{1}{V} \sum_{j \neq i} \left(\int_{\partial \Omega_{ij}} \boldsymbol{\sigma}_{d,i} \cdot \mathbf{n} \, dS \right) \quad (2.3.57)$$

$$\boldsymbol{\sigma}_{d,i} \equiv \langle \boldsymbol{\sigma}_{d,i}^* \rangle_I = \mu_i^* \mathbf{T}_\gamma \cdot \left(\nabla \mathbf{u}_i + (\nabla \mathbf{u}_i)^T - \frac{2}{3} (\nabla \cdot \mathbf{u}_i) \mathbb{I} \right) \quad (2.3.58)$$

By substituting the relevant quantities in the general macroscopic transport equation 2.3.41 and taking advantage of the formulations of the volume averaged stress tensor presented in 2.3.57 the momentum transport equation can be formulated as:

$$\begin{aligned} \frac{\partial}{\partial t} (\alpha_i \rho_i \mathbf{u}_i) + \nabla \cdot (\alpha_i \rho_i \mathbf{u}_i \otimes \mathbf{u}_i) &= -\alpha_i \nabla p + \nabla \cdot (\alpha_i \boldsymbol{\sigma}_{d,i}) + \alpha_i \rho_i \mathbf{g} + \\ &\quad - \sum_{j \neq i} (\mathbf{M}_{\alpha,i \rightarrow j} + \mathbf{M}_{\boldsymbol{\sigma}_d,i \rightarrow j}) + \\ &\quad - \mathbf{M}_{\alpha,i \rightarrow s} - \mathbf{M}_{\boldsymbol{\sigma}_d,i \rightarrow s} \end{aligned} \quad (2.3.59)$$

The additional source terms \mathbf{M} are related to the inter-phase momentum transfer. This can occur either due to phase change (represented by the \mathbf{M}_α terms) or due to momentum diffusion across the interface (represented by the $\mathbf{M}_{\boldsymbol{\sigma}_d}$ terms). The contributions indexed via $i \rightarrow j$ denote the exchange exclusively from the fluid phase to another fluid phase, while $i \rightarrow s$ denotes the exchange from the fluid phase to the solid phase (i.e. the structure).

The term $\mathbf{M}_{\alpha,i \rightarrow s}$ is equivalent to 2.3.51 with $\phi_i^* = \rho_i^* \mathbf{u}_i^*$. However, due to the same considerations regarding the local no-slip condition between fluid and structure made when discussing the continuity equation, this term is null, so that $\mathbf{M}_{\alpha,i \rightarrow s} = 0$. The term $\mathbf{M}_{\alpha,i \rightarrow j}$ can be further simplified by assuming that $\mathbf{u}_i \equiv \langle \mathbf{u}_i^* \rangle_I \approx \mathbf{u}_i^*$. Recalling the definition of the mass transfer term in 2.3.54, it can be shown that:

$$\mathbf{M}_{\alpha,i \rightarrow j} \approx \Gamma_{i \rightarrow j}^+ \mathbf{u}_i - \Gamma_{i \rightarrow j}^- \mathbf{u}_j \quad (2.3.60)$$

with the $+$, $-$ superscripts denoting the positive, negative parts of $\Gamma_{i \rightarrow j}$ respectively. It is clear from this formulation that the $\mathbf{M}_{\alpha,i \rightarrow j}$ terms account for the momentum transfer from the i -th to the j -th phase due to mass transfer, which is positive if phase i transfers net momentum to phase j .

The term $\mathbf{M}_{\sigma_d,i \rightarrow s}$ quantifies the diffusive transport of momentum from the fluid to the structure, which more informally can be described as the friction between the fluid and the structure and is defined as in 2.3.52 with $\mathbf{J}_i^* = \boldsymbol{\sigma}_{d,i}^*$. This term needs to be treated via experimental correlations depending on the fluid and case geometry under exam, and is further discussed in chapter 3.

The term $\mathbf{M}_{\sigma_d,i \rightarrow j}$ is associated with the diffusive momentum transport across a fluid-fluid interface, yet it cannot in principle be described exclusively in terms of friction between phases, due to the fact that said interfacial boundary, unlike that between a fluid and the structure, is movable. In the most general case, this term can be interpreted as the collection of the experimentally observed forces that affect a generic fluid phase in a multi-phase flow, ranging from friction-proper to the lift force (which arises from velocity gradients within the continuous phase and is thus tied to the continuous fluid deviatoric stress tensor, which $\mathbf{M}_{\sigma_d,i \rightarrow j}$ depends on), surface tension, virtual mass et cetera. For simplicity, as all the phase change-related momentum transfer terms have been simplified, the subscript σ_d is removed from the remaining diffusion related terms $\mathbf{M}_{\sigma_d,i \rightarrow j}$ for notation simplicity so that $\mathbf{M}_{i \rightarrow j} \equiv \mathbf{M}_{\sigma_d,i \rightarrow j}$. The coupled system of the macroscopic Navier-Stokes equations can be thus summarized as:

$$\left\{ \begin{array}{l} \frac{\partial}{\partial t} (\alpha_i \rho_i) + \nabla \cdot (\alpha_i \mathbf{u}_i \rho_i) = - \sum_{j \neq i} \Gamma_{i \rightarrow j} \\ \frac{\partial}{\partial t} (\alpha_i \rho_i \mathbf{u}_i) + \nabla \cdot (\alpha_i \rho_i \mathbf{u}_i \otimes \mathbf{u}_i) = \\ \quad - \alpha_i \nabla p + \nabla \cdot (\alpha_i \boldsymbol{\sigma}_{d,i}) + \alpha_i \rho_i \mathbf{g} + \\ \quad - \sum_{j \neq i} (\Gamma_{i \rightarrow j}^+ \mathbf{u}_i - \Gamma_{i \rightarrow j}^- \mathbf{u}_j + \mathbf{M}_{i \rightarrow j}) - \mathbf{M}_{i \rightarrow s} \end{array} \right. \quad (2.3.61)$$

Further discussion on the way that $\mathbf{M}_{i \rightarrow s}$, $\mathbf{M}_{i \rightarrow j}$ are modelled is presented in chapter 3.

2.3.4 Volume averaged enthalpy transport equation

To fully describe a fluid flow from a thermal-hydraulics perspective it is also necessary to simulate the evolution and transport of fluid enthalpy, represented by its specific enthalpy h_i^* . Its microscopic formulation follows from equation 2.3.41 with $\phi_i^* = \rho_i^* h_i^*$ the volumetric enthalpy, $\mathbf{J}_i^* = \kappa_i^* \nabla T_i^*$ the diffusion heat flux with T_i^* the fluid temperature, and $b_i^* = \frac{\partial}{\partial t} p^* + \rho_i^* \mathbf{u}_i^* \cdot \mathbf{g} + q_{int,i}^*$ the volumetric heat source due to the pressure work term, the gravitational work term an a possible volumetric heat source intrinsic to the fluid e.g. molten salt fuel in Molten Salt Reactors (MSRs).

By taking advantage of the coarse-mesh formulation of the general transport equation in 2.3.47 and considering the notation introduced in 2.3.42, the macroscopic enthalpy transport equation is obtained as:

$$\begin{aligned} \frac{\partial}{\partial t} (\alpha_i \rho_i h_i) + \nabla \cdot (\alpha_i \mathbf{u}_i \rho_i h_i) &= \nabla \cdot (\alpha_i \kappa_i \mathbf{T}_i \nabla T_i) + \alpha_i \frac{\partial}{\partial t} p + \alpha_i \rho_i \mathbf{u}_i \cdot \mathbf{g} + \\ &+ \alpha_i q_{int,i} - \sum_{j \neq i} (q_{\alpha,i \rightarrow j} + q_{\kappa,i \rightarrow j}) - q_{\alpha,i \rightarrow s} + \\ &- q_{\kappa,i \rightarrow s} - \frac{1}{V} \int_{\partial \Omega_{ij}} p^* \mathbf{u}_\partial^* \cdot \mathbf{n} dS \end{aligned} \quad (2.3.62)$$

The last term on the RHS comes from the volume average of the pressure work term $\frac{\partial}{\partial t} p^*$ and can be typically neglected owing to the fact that, for the vast majority of fluids of interest for nuclear applications, at nuclear operating pressures (i.e. in the 1 – 160 atm range) one has $p^* \ll \rho_i^* h_i^*$.

As seen when discussing the momentum equation, the terms q_α , q_κ represent the inter-phase heat transfer terms associated with phase change and diffusive heat conduction respectively. The fluid-structure heat transfer term associated with phase change $q_{\alpha,i \rightarrow s}$ is null for the same reasons discussed in the previous subsection. The phase change-related enthalpy source term $q_{\alpha,i \rightarrow j}$ is equivalent to $b_{\alpha,i \rightarrow j}$ in 2.3.48 with $\phi_i^* = \rho_i^* h_i^*$. Somewhat similarly to, but not exactly the same as the momentum transfer term $\mathbf{M}_{\sigma_d,i \rightarrow j}$, this term can be modelled as:

$$q_{\alpha,i \rightarrow j} = \Gamma_{i \rightarrow j} h_i \quad (2.3.63)$$

Please note that this formulation entails $q_{\alpha,i \rightarrow j} \neq -q_{\alpha,j \rightarrow i}$. What is conserved instead is the overall heat transfer due to *both* phase change *and* diffusive processes between the phases, namely $q_{\alpha,i \rightarrow j} + q_{\kappa,i \rightarrow j} = -(q_{\alpha,j \rightarrow i} + q_{\kappa,j \rightarrow i})$. This is subject to further discussion in section 3.2.3 in chapter 3. For clarity, if phases i and j do not belong to the same chemical species no phase transfer is possible, $\Gamma_{i \rightarrow j} = 0$ by definition and the phase change-related enthalpy transfer term is null.

With regard to the diffusive enthalpy transfer terms $q_{\kappa,i \rightarrow s}$, $q_{\kappa,i \rightarrow j}$, those are defined as in equation 2.3.49 with $\mathbf{J}_i^* = \kappa_i^* \nabla T_i^*$. These terms are modelled via experimental correlations that are discussed in chapter 3. As all the phase change-related enthalpy transfer terms have been simplified, the subscript κ is omitted from the diffusion-related inter-phase transfer terms, so that $q_{i \rightarrow j} \equiv q_{\kappa,i \rightarrow j}$. The macroscopic enthalpy conservation equation for the i -th phase can be thus written as:

$$\begin{aligned} \frac{\partial}{\partial t} (\alpha_i \rho_i h_i) + \nabla \cdot (\alpha_i \mathbf{u}_i \rho_i h_i) &= \nabla \cdot (\alpha_i \kappa_i \mathbf{T}_i \cdot \nabla T_i) + \alpha_i \frac{\partial}{\partial t} p + \alpha_i \rho_i \mathbf{u}_i \cdot \mathbf{g} + \\ &+ \alpha_i q_{int,i} - \sum_{j \neq i} (\Gamma_{i \rightarrow j} h_i + q_{i \rightarrow j}) - q_{i \rightarrow s} \end{aligned} \quad (2.3.64)$$

Further modelling aspects related to the treatment of the phase-change dependent terms and the diffusive inter-phase transfer terms $q_{i \rightarrow j}$, $q_{i \rightarrow s}$ are discussed in chapter 3.

2.4 Clarifications and conclusive remarks

2.4.1 Remarks on the relationship between the coarse-mesh approach and an Euler-Euler approach

It is recognized that equations 2.3.61 and 2.3.64 are formally equivalent to those formulated by an Euler-Euler approach for the modelling of multi-phase flows. This formal equivalence stems from the utilization of volume averaging techniques, which are at the core of both methods. The key difference lies however at the geometric scale of interest.

A coarse-mesh approach is fundamentally intended for the investigation of one-fluid-phase or possibly multi-fluid-phase flows through a complex structure of interest *whose geometric details lie below the geometric scale of interest*, so that volume averaging techniques are applied over RVs (and thus their constituent mesh cells) that are significantly coarser than said details.

An Euler-Euler approach, in its general formulation, is obtained via the application of the same volume averaging techniques over any possible computational domain of interest (i.e. cell-by-cell) with no specific limitations on the size of the mesh cells and thus no limitation on the details that can be resolved. For this reason, a coarse-mesh approach is equivalent to a *dispersed* Euler-Euler approach, wherein by dispersed one means that the details of the phase geometry and phase interfaces (both fluid-fluid and fluid-structure) are not explicitly geometrically resolved, rather, treated with other approaches (e.g. interfacial area transport methods in combination with other models for the treatment of characteristic phase dimensions).

Ultimately, the difference in how the formally identical equations are used in e.g. a coarse-mesh context against e.g. a fine-mesh Euler-Euler context lies entirely in how the inter-phase mass $\Gamma_{j \rightarrow i}$, momentum $\mathbf{M}_{j \rightarrow i}$ and heat transfer $q_{j \rightarrow i}$ terms are modelled.

2.4.2 Remarks on the relationship between the coarse-mesh approach and the FVM

This section has discussed how the coarse-mesh approach is used in the context of a generic multiphase system to transform microscopic quantities, mathematical operators and equations, i.e. that hold within an infinitesimal volume of a single-phase, into their macroscopic equivalent, i.e. into a volume-averaged equivalent that holds over a certain sub-domain Ω of the domain of interest that might encompass a variety of phases. In particular, starting from a generic microscopic transport equation, equation 2.3.41, the averaging tools

have been used to derive the macroscopic Navier-Stokes and enthalpy conservation equations governing fluid-flow, namely equations 2.3.61, 2.3.64. Extensive discussion of the modelling for the closure of mass, momentum and heat inter-phase transfer terms is made in chapter 3.

As both the FVM and the coarse-mesh approach have been discussed from a general perspective in this chapter, it is convenient to reassert the fundamental differences between these two methods. This is not necessarily trivial as both approaches involve volume averages to manipulate a set of governing equations, to the point where the reader might be left wondering what is the purpose of each method compared to the other.

The coarse-mesh approach is used to construct a *mathematical model* that is representative of some fundamental governing equations in a multi-phase context at a spatial scale that is coarser than that at which the microscopic governing equations actually hold true. The FVM is instead used to construct a *numerical representation* of a given mathematical model of interest (e.g. the set of PDEs representing the coarse-mesh Navier-Stokes and enthalpy conservation equations), namely something that can be numerically treated and solved.

As a final example for clarification purposes, the operations that are required to obtain a numerical representation of e.g. the enthalpy equation that governs the thermal behaviour of a fluid in a complex engineering structure are presented:

- definition of the relevant RVs over the geometry for the calculation of the structure macroscopic quantities (i.e. volume fractions, interfacial areas, tortuosities);
- construction of the computational domain over the RVs by subdividing them into computational cells (i.e. *meshing*);
- application of the coarse-mesh methodology tools to obtain the macroscopic enthalpy equation (for this example, this consists of 2.3.64) that is valid *within each RV* (and by extension, within the mesh cells within each RV);
- application of the FVM to the coarse-mesh enthalpy equation (i.e. the actual governing equation over the geometry at the scale of interest) within each mesh cell of each RV to obtain a linear system representative of the governing equation, which can then be solved with adequate techniques, not presently discussed due to the vastness of the topic [33].

Please note that all the macroscopic enthalpy equations that hold within each RV are identically in form to one another, except for the coarse-mesh parameters that overtly figure in 2.3.64, namely the fluid phase fraction α and the tortuosity tensor \mathbf{T} . Thus, a single coarse-mesh enthalpy equation holds over the entirety of the computational domain as long as these additional coarse-mesh parameters are modelled as fields whose values vary cell-by-cell based

on which RV the cell belongs to. Additional coarse-mesh parameters, such as the structure volumetric surface area density (and hydraulic diameter, which can generally be related to the volumetric surface area density) generally figure in the specific correlations for the closure of the momentum and heat transfer terms introduced in the previous subsection, that are discussed more thoroughly in chapter 3.

Chapter 3

Algorithms and implementation

3.1 Introduction

The previous chapter has discussed how the coarse-mesh methodology is used to obtain the macroscopic set of equations governing some aspects of fluid flow for each fluid phase in a multi-phase system. The coarse-mesh Navier-Stokes and the enthalpy conservation equations were derived, which are recalled for clarity:

$$\frac{\partial}{\partial t} (\alpha_i \rho_i) + \nabla \cdot (\alpha_i \mathbf{u}_i \rho_i) = - \sum_{j \neq i} \Gamma_{i \rightarrow j} \quad (3.1.1)$$

$$\begin{aligned} \frac{\partial}{\partial t} (\alpha_i \rho_i \mathbf{u}_i) + \nabla \cdot (\alpha_i \rho_i \mathbf{u}_i \otimes \mathbf{u}_i) = \\ - \alpha_i \nabla p + \nabla \cdot (\alpha_i \boldsymbol{\sigma}_{d,i}) + \alpha_i \rho_i \mathbf{g} + \end{aligned} \quad (3.1.2)$$

$$\begin{aligned} \frac{\partial}{\partial t} (\alpha_i \rho_i h_i) + \nabla \cdot (\alpha_i \mathbf{u}_i \rho_i h_i) = \\ \nabla \cdot (\alpha_i \kappa_i \mathbf{T}_i \cdot \nabla T_i) + \alpha_i \frac{\partial}{\partial t} p + \alpha_i \rho_i \mathbf{u}_i \cdot \mathbf{g} + \alpha_i q_{int,i} + \\ - \sum_{j \neq i} (\Gamma_{i \rightarrow j} h_i + q_{i \rightarrow j}) - q_{i \rightarrow s} \end{aligned} \quad (3.1.3)$$

This chapter is dedicated to the description of the modelling approaches and numerical solution algorithms that were developed and employed in this work for the treatment of the coupled system represented by equations 3.1.1, 3.1.2, 3.1.3. In practice, discussing the present thermal-hydraulic approach amounts to discussing the computer code that has been developed which implements said approach.

The computer code is based on the OpenFOAM numerical library and is capable of modelling one-fluid-phase and two-fluid-phase flows through struc-

tures via the coarse-mesh approach as discussed previously. It consists of two independent sub-solvers, one for one-fluid-phase flows and one for two-fluid phase flows. This was done exclusively for computational efficiency as the two-phase sub-solver contains additional elements that represent unnecessary overheads for the simulation of one-phase flows. Nonetheless, the two sub-solvers share a single implementation of the same modelling approach, as it will be discussed. The code has been ultimately integrated in the GeN-Foam multi-physics code replacing its original thermal-hydraulics approach, and will be occasionally referred to via its internal name, *FFSEulerFoam*, with FFS standing for “Fluid-Fluid-Structure”, referring to its coarse-mesh, generally two-fluid-phase vocation.

Section 3.2 covers the modelling framework, more specifically regarding the closure terms introduced so far as well as some further important modelling aspects, ranging from the modelling of the structure thermal behaviour to turbulence modelling. Section 3.3 discusses the solution algorithms for the system of governing equations represented by 3.1.1, 3.1.2, 3.1.3 for the one and two-phase approaches. This is inclusive of a presentation of further algorithm improvements with respect to standard two-phase thermal-hydraulic approaches in OpenFOAM. Sections 3.4, 3.5 present the verification and assessment of the performance of a novel approach for treating pressure-velocity coupling in two-fluid-phase flows, compared against existing approaches found in other OpenFOAM-based solvers. Section 3.6 presents the overall parallel scalability performance of the solution algorithm.

3.2 Modelling

The present section expands upon the modelling elements required for the closure of the governing thermal-hydraulic equations. As the modelling framework is shared between the one-phase and two-phase sub-solvers (i.e. all the one-phase models are a special case of the two-phase ones), the discussion will focus on the broader case of two fluid phases. These will be labelled as i and j for consistency with the previous chapter, yet it should be noted that all the summations terms of the inter-phase transfer terms in the governing equations 3.1.1, 3.1.2, 3.1.3 give way to a single inter-phase transfer term (for the only possible fluid-pair that exists in a two-phase system).

An overview of the modelling framework is provided in Figure 3.2.1, inclusive of the variables that are set/controlled by each model.

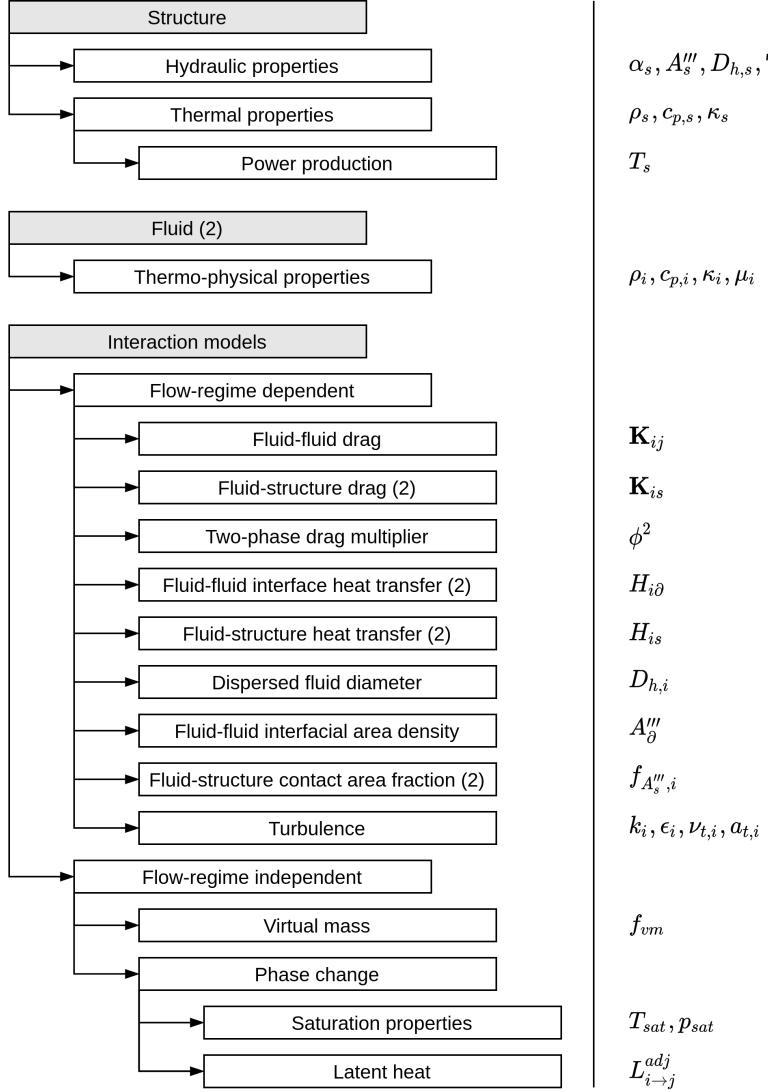


Figure 3.2.1: Logical representation of the modelling components of the code and their hierarchical dependency. Black labels represent the main logical subdivisions of the modelling: 1) structure-related models; 2) fluid-related models; 3) Phase interaction-related models. The main variables that are handled by each model are reported to the right of each entry. Numbers in brackets mean the number of models of that kind required to describe the two-phase system.

3.2.1 Flow regime map

In principle, all of the transfer terms $\mathbf{M}_{i \rightarrow j,s}$, $q_{i \rightarrow j,s}$ are to be formulated in terms of quantities that can be modelled via experimental correlations. However, the generic form of these correlations will typically vary for various *flow configurations* or *flow regimes*, both in one-phase and two-phase conditions.

Let us consider an example for one-phase flows, where different correlations for e.g. the laminar and turbulent flow regimes exist. Let us consider the Darcy friction factor f_d (i.e. drag coefficient) and the heat transfer coefficient

H for a fluid flow in a circular pipe. In the laminar regime, the friction factor can be analytically shown to be $f_d = 64/Re$, with Re being the Reynolds number of the flow. In turbulent regimes, a good approximation is represented by the Blasius correlation $f_d = 0.316/Re^{0.25}$. With regards to the heat transfer coefficient, it can be computed starting from the Nusselt number. In developed laminar pipe flows, it is best represented by a constant value. In developed turbulent pipe flows, it can be modelled e.g. via the Dittus-Boelter correlation, $Nu = 0.023 Re^{0.8} Pr^{0.4}$, with Pr being the Prandtl number of the fluid. Clearly, to be able to compute the value of the drag and heat transfer coefficients in each cell, one first needs to be able to identify the predominance of one or the other flow regime in each mesh cell via certain criteria, so to ascertain which correlation to use. For the specific example at hand, this criterion is represented by the value of the Reynolds number. For each cell, if the Reynolds value is below a certain threshold, a laminar regime will be assumed to dominate, while above another threshold (not necessarily coinciding with the first threshold), a turbulent regime will be assumed to dominate. Thus, via a cell-by-cell knowledge of the dominant regime type, the correlation for each quantity of interest can be chosen accordingly for each cell.

These quantities are typically set (not only in the present computer code but more generally in the domain of computer codes for nuclear applications [6]) via computer subroutines which we will refer to as *models*, each responsible for the management of a different quantity (drag, heat transfer coefficient, etc.).

There are fundamentally two ways to treat the regime dependence. The first and simplest approach would be to implement the flow regime map within each model individually. The principal drawback of this approach is the added computational cost associated with the re-evaluation of the same flow regime map if multiple models happen to depend on the same one, such as in the aforementioned example. The second approach would be to implement a flow regime map as a separate model that determines the flow regime information for each cell once per iteration or time-step and that acts as a “container” for all the flow-regime dependent models, with these models implementing no flow regime-dependence themselves. In this case, it would be the flow regime map itself that ultimately constructs the fields representative of the models it manages (e.g. drag, heat transfer coefficient, etc.) based on the cell-by-cell flow regime information. This approach is potentially more computationally efficient than the first one, as it only needs to evaluate the flow regime map once per iteration/time-step. However, it is generally more complex to implement and has an advantage only if multiple models rely on the same flow regime map, which is not necessarily always the case. An example of this is represented by e.g. the heat transfer coefficient between a liquid and a pipe wall in boiling scenarios, for which different correlations will need to be used depending on more than a single parameter (e.g. wall and fluid temperature, vapour phase mass flux, etc.).

Scenarios of in which different models share the same flow regime map were

deemed frequent enough that support for such a feature was implemented in the code. The maps are currently parametrizable with respect to a single code variable of choice (e.g. the volume fraction of one of the two phases, the Reynolds number, etc.). Interpolation regimes are created automatically if the parameter bounds provided for each flow regime mismatch. For clarity, an example is made. Let us consider a flow regime map parametrized with respect to a generic code variable x , and let us consider two regimes, labelled 1 and 2, with bounds $x_{1,lo}$, $x_{1,hi}$ and $x_{2,lo}$, $x_{2,hi}$ respectively. For a given variable of interest y that is to be modelled in a regime-dependent way (e.g. a heat transfer coefficient), so that flow regime 1 would evaluate y as y_1 and flow regime 2 as y_2 , y will be set on a cell-by-cell basis as:

$$y = \begin{cases} y_1 & x_{1,lo} \leq x < x_{1,hi} \\ y_i & x_{1,hi} \leq x < x_{2,lo} \\ y_2 & x_{2,lo} \leq x < x_{2,hi} \end{cases}$$

with y_i being either a linearly or quadratically interpolated value between the two values predicted by the models prescribed for each of the two flow regimes.

The possibility to hard-code a flow regime map dependence in each model still exists, and the presented flow regime map mechanism is to be viewed as an added benefit that can be taken advantage of in certain situations. Figure 3.2.1 highlights which models are treatable via flow regime maps and which are not.

3.2.2 Momentum transfer closure

Fluid-structure momentum transfer It is recalled that the momentum transfer term $\mathbf{M}_{i \rightarrow s}$ in 3.1.2 consists microscopically of the diffusive transfer of momentum across the fluid-structure interface, which is macroscopically interpreted as a frictional pressure loss of the fluid, so that:

$$\mathbf{M}_{i \rightarrow s} \equiv -\nabla p \Big|_{fric}^i \quad (3.2.1)$$

For the i -th fluid, this is modelled as [6]:

$$\mathbf{M}_{i \rightarrow s} = \mathbf{K}_{is} \cdot \mathbf{u}_i \quad (3.2.2)$$

In which \mathbf{K}_{is} is the fluid-structure *drag factor*, a dimensioned order-2 tensor not to be confused with the dimensionless fluid-structure *drag coefficient* $f_{d,is}$. To simplify the discussion, the relationship between these two quantities is introduced first in the context of a one-phase flow in an *isotropic* structure. By isotropic it is meant precisely that, regardless of the direction in which a pressure gradient is applied at the boundaries of the system containing the isotropic structure, the resulting fluid flow through the structure and the pressure gradient are always anti-parallel. The phase label i is replaced with $1p$

to indicate that these relations are valid in single-phase only. The frictional pressure drop is modelled via the Darcy-Weisbach equation [34]:

$$\nabla p \Big|_{fric}^{1p} = -(1 - \alpha_s) \frac{1}{2D_h} \rho_{1p} |\mathbf{u}_{1p}| f_{d,s}(Re_{1p,s}) \mathbf{u}_{1p} \quad (3.2.3)$$

where $\alpha_s = 1 - \alpha_{1p}$ is the structure phase fraction, D_h is the hydraulic diameter of the structure, $f_{d,s}$ is the one-phase Darcy-Weisbach drag coefficient, which will be referred to just as the drag coefficient, and it generally depends on the one-phase Reynolds number $Re_{1p,s}$. As a clarification, \mathbf{u}_{1p} is the real fluid velocity, not the superficial one. The additional term $(1 - \alpha_s)$ re-scales the frictional pressure gradient (which is dimensionally equal to a momentum loss per unit time, per unit volume) exclusively over the fraction of the cell volume effectively occupied by the fluid. With regards to the Reynolds number for the fluid-structure system, this is defined as:

$$Re_{1p,s} = \frac{D_h |\mathbf{u}_{1p}|}{\nu_{1p}} \quad (3.2.4)$$

with ν_{1p} being the kinematic viscosity of the fluid. The drag coefficient is case-dependent and should be chosen accordingly, depending on the geometry of interest. Specific correlations for its modelling will be discussed alongside the application of the present methodology, in chapters 4, 5. By considering equations 3.2.1 through 3.2.3, in a one-phase scenario the drag factor thus reduces to:

$$\mathbf{K}_{1p,s} \equiv \mathbb{I} K_{1p,s} = \mathbb{I} (1 - \alpha_s) \frac{1}{2D_h} \rho_{1p} |\mathbf{u}_{1p}| f_{d,s}(Re_{1p,s}) \quad (3.2.5)$$

with \mathbb{I} the identity tensor. One might wonder, at this point, what is the necessity to describe this quantity with a second-order tensor, given its scalar nature. The reason is that, while this example considered specifically an isotropic structure for the sake of introducing the fundamental connection between the mathematical model behind the momentum source $\mathbf{M}_{s \rightarrow 1p}$ and the experimental approach based on the Darcy-Weisbach equation, for the case of a generally anisotropic structure (e.g. a pin bundle) the frictional pressure drop experienced by the flow will generally depend on the flow direction.

The general approach for modelling this phenomenon is to construct a drag tensor $\mathbf{K}_{1p,s}$ with different correlations for the drag coefficient for different tensor components. This is fundamentally the same as the approach commonly used in sub-channel codes [7], where the fluid-structure drag is typically separated in a component along e.g. the pin bundle and a component transverse to it, with the use of different correlations for each of the so-called *principal directions*. As it is generally always possible to define a local reference frame in which $\mathbf{K}_{1p,s}$ is diagonal, this amounts to:

$$\mathbf{K}_{1p,s} = \sum_k (1 - \alpha_s) \frac{1}{2D_h} \rho_{1p} u_{1p,k} f_{d,s,k}(Re_{1p,s}) \mathbf{e}_k \otimes \mathbf{e}_k \quad (3.2.6)$$

in which \mathbf{e}_k is the k -th basis vector for the k -th direction, $u_{1p,k}$ is the k -th velocity component and $f_{d,s,k}$ is the drag coefficient for said direction.

The approach presented in 3.2.6 is only valid for one-phase flows, as in a two-phase scenario the frictional pressure drop experienced by each of the two fluids due to the structure cannot be computed solely via this approach. Let us thus tackle the two-phase pressure drop modelling starting from some specific phenomenological considerations that will be generalized.

Let us consider the frictional pressure drop experienced by fluid flow in e.g. a pipe of a certain length L . The one-phase fluid velocity is proportional to the mass flow rate as $\dot{m}_{1p} \propto \rho_{1p} |\mathbf{u}_{1p}|$. For the specific case at hand, the Blasius friction factor for turbulent pipe flows would have $f_{1p,s} \propto Re^{-0.25}$, so that by considering 3.2.3, 3.2.4, $\Delta p_{fric} = |\nabla p_{fric}|L \propto |\mathbf{u}_{1p}|^{1.75}$. In general, without specifications on the flow type or geometry, one can generally expect $\Delta p_{fric} \propto |\mathbf{u}_{1p}|^{1-2}$.

An example of the dependence of frictional pressure drop on mass flow rate is presented in Figure 3.2.2.

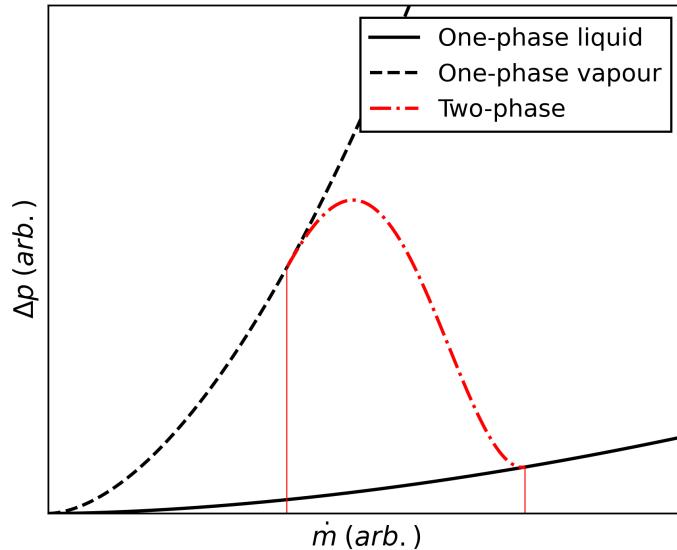


Figure 3.2.2: General dependence of the pressure drop over e.g. a tube for varying levels of inlet mass flow rate ($[\dot{m}] = \text{kg/s}$) and different states of matter of the flowing chemical species under consideration. The pressure drop for a purely liquid phase of certain chemical species is typically much smaller than the pressure drop for the vapour phase, owing to the much larger velocity of the vapour phase at the same mass flow rate.

If the liquid and vapour phases of a chemical species of interest are taken into consideration, for any given mass flow rate the resulting pressure drop for a one-phase liquid flow would be much smaller than the corresponding pressure drop for a one-phase vapour flow, owing to the much smaller vapour density and thus higher vapour velocity. For any particular circumstance that would cause the one-phase flow to transition to a two-phase flow (e.g. boiling or

condensation), the resulting *total* frictional pressure drop of the flow mixture will be somewhere in between the one-phase pressure drops of the individual phase flows. Figure 3.2.2 represents this via the specific example of a fluid flow through a heated tube such that the wall heat flux is constant. By starting at a sufficiently high mass flow rate, the flow stays liquid throughout the length of the tube. For decreasing mass flow rates, the enthalpy removal will not be sufficient to prevent the liquid from reaching saturation, and below a certain mass flow rate boiling occurs. Conversely, for sufficiently small inlet mass flows, all of the liquid undergoes phase change and the pressure drop-mass flow rate relationship is that of one-phase vapour flow. Thus, in the two-phase regime, the pressure drop-mass flow rate relationship must be such to link the two one-phase profiles in some way. From an experimental perspective, the global mixture two-phase pressure drop can be measured directly. The way in which this is modelled and connected to the individual phase drag factors \mathbf{K}_{is} is discussed.

A generic two-phase flow through a structure of interest is now considered. The overall frictional pressure gradient experienced by the two-phase mixture can be expressed in terms of a *superficial* frictional pressure gradient experienced by one phase, adjusted by a *two-phase pressure drop multiplier* ϕ_i^2 , so that:

$$\nabla p \Big|_{fric}^{tot} = \phi_i^2 \nabla p \Big|_{fric,sup}^i \quad (3.2.7)$$

While the theoretical framework of two-phase multipliers was developed for 1-D flow analyses, it is here applied to all components of the frictional pressure gradient in a generic 3-D flow configuration, for lack of a more meaningful alternative, as highlighted by other computer codes with a similar domain of applicability (e.g. the sub-channel code SABENA [11]).

The superficial frictional pressure gradient $\nabla p \Big|_{fric,sup}^i$ can be defined in two different ways. With reference to a 1-D flow for simplicity, it is the frictional pressure gradient that the i -th phase would experience if it occupied the entirety of the flow-available cross-sectional area either with: 1) a mass flux equal to its real mass flux G_i in the two-phase scenario; or 2) a mass flux equal to the total mixture mass flux G_{tot} . These two different definitions of the superficial frictional pressure gradient lead to different definitions of the two-phase multiplier. Within the scope of this computer code, the first definition is used, as it is the one used by many established two-phase pressure drop multiplier correlations [35] (e.g. Lockhart-Martinelli). As a clarification:

$$G_i = \dot{m}_i S = |\alpha_i \rho_i \mathbf{u}_i| \quad (3.2.8)$$

$$G_{tot} = G_1 + G_2 \quad (3.2.9)$$

The phase fraction term α_i quantifies the actual fluid volume whose flow contributes to the mass flux in each computational cell. Note that this volume fraction takes into consideration the presence of the structure within

each cell (e.g. if the two-phase flow is taking place in a pin bundle), so that $\alpha_1 + \alpha_2 + \alpha_s = 1$. For the sake of convenience, a normalized fluid phase fraction is introduced as:

$$\alpha_i^N = \frac{\alpha_i}{1 - \alpha_s} \quad (3.2.10)$$

A disambiguation on the term “superficial” could prove useful to solve possible misunderstandings of upcoming discussions. In the coarse-mesh methodology said term describes a fluid quantity that has been averaged over the entire averaging volume (i.e. inclusive of the structure volume within the averaging volume), rather than only the fluid volume within the averaging volume (e.g. recall the definition 2.3.2). In that sense, a superficial velocity is related to the intrinsic velocity as $\mathbf{u}_{i,sup} = \alpha_i^N \mathbf{u}_i$. In this context however, the term “superficial” is used to describe a fluid quantity that has been averaged over the volume that is *available for fluid flow* within the averaging volume, i.e. the total volume occupied by all fluid phases, which is complementary to the total volume occupied by the structure. Thus:

$$\mathbf{u}_{i,sup} = \alpha_i^N \mathbf{u}_i \quad (3.2.11)$$

Returning onto the problem at hand, the frictional pressure drop can be calculated as (recall 3.2.1, 3.2.2):

$$\nabla p \Big|_{fric,sup}^i = \mathbf{K}_{is,sup} \cdot (\alpha_i^N \mathbf{u}_i) \quad (3.2.12)$$

with $\alpha_i^N \mathbf{u}_i$ being the superficial velocity, $\mathbf{K}_{is,sup}$ being a drag factor that we will refer to as superficial drag factor, which can be defined in the same way as for the one-phase scenario in 3.2.6, except that the superficial velocity is used in place of the intrinsic one and the friction factor is calculated with the superficial Reynolds defined as:

$$Re_{is,sup} = \frac{D_h |\alpha_i^N \mathbf{u}_i|}{\nu_i} \quad (3.2.13)$$

so that:

$$\mathbf{K}_{is,sup} = \sum_k (1 - \alpha_s) \frac{1}{2D_h} \rho_i \alpha_i^N u_{i,k} f_{d,s,k}(Re_{is,sup}) \mathbf{e}_k \otimes \mathbf{e}_k \quad (3.2.14)$$

Returning to the point at issue, it is recalled that the actual frictional pressure gradient experienced by the i -th phase is (equations 3.2.1, 3.2.2):

$$\nabla p \Big|_{fric}^i = \mathbf{K}_{is} \cdot \mathbf{u}_i \quad (3.2.15)$$

The two-phase multiplier approach is connected to the frictional pressure gradient in the form of $\mathbf{K}_{is} \cdot \mathbf{u}_i$ by relating it to the total mixture pressure gradient. Assuming that the total pressure gradient is partitioned between the

two phases according to a *fluid-structure contact fraction* $f_{A''_s,i}$, the following holds:

$$\mathbf{K}_{is} \cdot \mathbf{u}_i = f_{A''_s,i} \left. \nabla p \right|_{fric}^{tot} \quad (3.2.16)$$

The contact fraction $f_{A''_s,i}$ quantifies the fraction the structure surface area that is contacted by fluid i . If the fluid that was used as the basis for the calculation of the total pressure drop as described in 3.2.7 is labelled by M , by combining 3.2.16 with 3.2.7 and by combining the resulting equation with 3.2.12, \mathbf{K}_{is} can finally be calculated as:

$$\mathbf{K}_{is} = f_{A''_s,i} \alpha_M^N \phi_M^2 \mathbf{K}_{Ms,sup} \frac{|\mathbf{u}_M|}{|\mathbf{u}_i|} \quad (3.2.17)$$

It is important to state that in the derivation of 3.2.17, it was assumed that \mathbf{u}_i and \mathbf{u}_M are parallel, in line with the 1-D nature of the underlying two-phase multiplier approach.

As an example, for a liquid-vapour mixture with phases labelled l and v , if the total pressure drop is calculated via a liquid-based two-phase pressure drop multiplier (commonly done in e.g. boiling scenarios), one would have:

$$\begin{cases} \mathbf{K}_{ls} = f_{A''_s,l} \alpha_l^N \phi_l^2 \mathbf{K}_{ls,sup} \\ \mathbf{K}_{vs} = f_{A''_s,v} \alpha_l^N \phi_l^2 \mathbf{K}_{ls,sup} \frac{|\mathbf{u}_l|}{|\mathbf{u}_v|} \end{cases} \quad (3.2.18)$$

which is how the drag factors are calculated. Note that since the total pressure drop was calculated on the basis of the liquid phase superficial frictional pressure gradient (i.e. $M = l$), the assumption that \mathbf{u}_l is parallel to \mathbf{u}_M is tautologically true. However, this is not necessarily true for \mathbf{u}_v . In short, the distribution of the total mixture-structure frictional pressure gradient between the liquid and gas phases is calculated assuming that both the liquid-structure and gas-structure frictional pressure gradients are parallel. This limitation is however deemed acceptable as, as highlighted earlier, this is a 3-D adaptation of an otherwise 1-D model.

To summarize, in a generic two-phase context, the fluid-structure momentum exchange term for the i -th phase is modelled as:

$$\mathbf{M}_{i \rightarrow s} = \left(f_{A''_s,i} \alpha_M^N \phi_M^2 \mathbf{K}_{Ms,sup} \frac{|\mathbf{u}_M|}{|\mathbf{u}_i|} \right) \cdot \mathbf{u}_i \quad (3.2.19)$$

with $\mathbf{K}_{Ms,sup}$ defined as of 3.2.14, in which M is a label representing the phase that was chosen as basis for the calculation of the mixture frictional pressure gradient. For boiling flows, this generally consist of the liquid phase, while for condensing flows, this generally consists of the vapour phase. For one-phase flows instead, $f_{A''_s,i} = f_{A''_s,1p} = 1$, $\alpha_M^N = \alpha_{1p}^N = 1$ and $\phi_M^2 = 1$ by definition, so that:

$$\mathbf{M}_{1p \rightarrow s} = \mathbf{K}_{1p,s} \cdot \mathbf{u}_{1p} \quad (3.2.20)$$

where it is recalled that, in virtue of equations 3.2.6, 3.2.14, one has that $\mathbf{K}_{1p,s} = \mathbf{K}_{1p,s,sup}$ in one-phase flows. Thus, regardless of the number of fluid phases, the drag factor is always computed according to 3.2.14.

Fluid-fluid momentum transfer The fluid-fluid momentum transfer term $\mathbf{M}_{i \rightarrow j}$ is only modelled in two-phase flows, for obvious reasons. It is recalled that it consists of the momentum transfer across the fluid-fluid interface due to diffusive processes. While there is a number of phenomena that contribute to this at a microscopic level, such as friction, virtual mass, surface tension, lift, turbulent dispersion etc., in this context only the friction and virtual mass effects are modelled. This is fundamentally due to the intended spatial resolution scale of the present approach, compatible with sub-channel approaches, which also neglect the description of said forces in an explicit manner. Conversely, the effects of these unresolved forces should be somehow captured via an enhanced momentum and thermal diffusivity. This will be briefly covered in sub-section 3.2.5 concerning turbulence modelling. Thus:

$$\mathbf{M}_{i \rightarrow j} = \mathbf{M}_{d,i \rightarrow j} + \mathbf{M}_{vm,i \rightarrow j} \quad (3.2.21)$$

with $\mathbf{M}_{d,i \rightarrow j}$ and $\mathbf{M}_{vm,i \rightarrow j}$ representing the drag and virtual mass forces respectively. The drag force is modelled in a manner that is fundamentally analogously to fluid-structure drag, namely:

$$\mathbf{M}_{d,i \rightarrow j} = -K_{ij}(\mathbf{u}_j - \mathbf{u}_i) \quad (3.2.22)$$

with the key difference that the fluid-fluid drag force acting on phase i is inherently parallel to the slip velocity $(\mathbf{u}_j - \mathbf{u}_i)$, so that the drag factor for this process consists of a scalar rather than second-order tensor. Similarly to the fluid-structure drag factor seen in 3.2.14, the generic form of K_{ij} is:

$$K_{ij} = (1 - \alpha_s) \frac{\rho_d}{2D_{h,D}} |\mathbf{u}_i - \mathbf{u}_j| f_{d,ij}(Re_{ij}) \quad (3.2.23)$$

with the C, D subscripts denoting either the *continuous* or *dispersed* phase properties. The term $f_{d,ij}$ is the Darcy friction factor. Generally, but not necessarily, this factor is modelled as a function of the fluid-fluid interfacial Reynolds number Re_{ij} , which is calculated as:

$$Re_{ij} = \frac{D_{h,D} |\mathbf{u}_i - \mathbf{u}_j|}{\nu_C} \quad (3.2.24)$$

The virtual mass force is modelled as:

$$\mathbf{M}_{vm,i \rightarrow j} = -f_{vm}\rho_c \left(\frac{\partial}{\partial t} \mathbf{u}_i + \mathbf{u}_i \cdot \nabla \mathbf{u}_i - \frac{\partial}{\partial t} \mathbf{u}_j \right) \quad (3.2.25)$$

with f_{vm} the virtual mass coefficient. Unlike numerous other geometric and non-geometric flow-regime dependent parameters (e.g. characteristic phase dimensions $D_{h,i}$, fluid-structure contact fractions $f_{A''',i}$, volumetric interfacial area densities A'''_i , etc.), the virtual mass is currently modelled as a constant for all flow regimes for simplicity. While it is not uncommon for sub-channel codes to neglect virtual mass effects, these are considered in the present model as they have been shown to improve the numerical stability of the iteratively solved pressure-velocity coupled system [36] while not significantly affecting overall results [37]. This was also observed in this work, resulting in significant simulation run time reduction, as it is discussed in chapter 4.

3.2.3 Heat and mass transfer closure

Fluid-structure heat transfer The fluid-structure heat transfer term $q_{i \rightarrow s}$ is modelled as:

$$q_{i \rightarrow s} = f_{A''',i} A''' H_{is} (T_i - T_s) \quad (3.2.26)$$

with $f_{A''',i}$ the fluid-structure contact area fraction, A''' 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. For clarity, $[H] = W/m^2/K$ and $[q] = W/m^3$. Please note that due to the coarse-mesh modelling, all of the reported variables represent their corresponding quantities within each computational cell, i.e. T_s is the representative surface temperature structure in a certain cell while T_i is the representative fluid temperature of the fluid in that same cell.

For one-phase flows, the heat transfer coefficients is purely convective in nature and is generally calculated starting from the Nusselt number Nu_{is} for the flow-structure configuration as:

$$H_{is,conv} = \frac{\kappa_i \, Nu_{is}}{D_{h,i}} \quad (3.2.27)$$

with κ_i the fluid thermal conductivity and $D_{h,i}$ the characteristic fluid dimension, which in one phase flows equals the structure hydraulic diameter. The Nu_{is} can be calculated via user-provided correlations that are typically in the form [35]:

$$Nu_{is} = c_0 + c_1 \, Re_{is,sup}^{c_2} \, Pr_i^{c_3} \quad (3.2.28)$$

with c_0 through c_3 being fluid, geometry dependent coefficients [38], $Re_{is,sup}$ the fluid-structure Reynolds defined as of 3.2.13¹ and Pr_i the fluid Prandtl number.

For the formulation of the two-phase fluid-structure heat transfer coefficient, the only models currently implemented consist in models for the treatment of nucleate boiling. As it will be explained in chapter 4, since Sodium-cooled Fast Reactors (SFRs) constituted the main application of interest for the present work, due to the peculiar aspects of sodium boiling (suppressed or absent sub-cooled boiling regimes due to the low sodium Prandtl number [39], convective heat-transfer typically dominating up to critical heat flux [40]) this treatment was deemed sufficient. Nonetheless, the specifics of the modelling framework were programmed is such a way to enable the future inclusion of a more complex treatment of the heat transfer coefficient, to allow treatment of fluids e.g. for which significantly more complete (and complex) models for the wall-fluid heat transfer coefficient in various heat transfer regimes (e.g. sub-cooled boiling, pre and post-critical heat flux boiling, etc.) exist, such as models used in e.g. the TRACE code [6].

¹We recall that in one-phase flows $\alpha_i^N = 1$, so that $Re_{is,sup}$ coincides with the traditional Reynolds number definition as of equation 3.2.4.

The current formulation of the two-phase fluid-structure heat transfer coefficient is based on the nucleate boiling formulation by Chen [41], which constructs the total heat transfer coefficient as a combination of a convective heat transfer coefficient $H_{is,conv}$ and a pool-boiling heat transfer coefficient $H_{is,pb}$ as:

$$H_{is} = f_{F,i} H_{is,conv} + f_{S,i} H_{is,pb} \quad (3.2.29)$$

where $f_{F,i}$ and $f_{S,i}$ are a flow enhancement factor and a suppression factor respectively. Physically, the flow factor represents the enhancement of the convective component due to the added near-wall turbulence caused by departing bubbles, which cannot be resolved without a fine-mesh Computational Fluid Dynamics (CFD) treatment. Conversely, the suppression factor exists to blend pool boiling between the nucleate boiling regime to the transition boiling regime, in which the pool boiling contribution decreases significantly.

While the original model by Chen already prescribes models for each of those four quantities, user-selectable models can be provided independently of each other. Some of these are introduced in chapter 4.

Fluid-fluid heat transfer The interfacial heat transfer between the two fluids is modelled via a two-resistance approach, meaning that $q_{i \rightarrow j}$ consists of the heat flux (per unit volume, for each mesh cell) between the bulk of fluid i and the unresolved interface between fluids i and j within each mesh cell. The term $q_{i \rightarrow j}$ is thus denoted as $q_{i \rightarrow \partial}$. Then:

$$q_{i \rightarrow \partial} = A''_\partial H_{i\partial}(T_i - T_\partial) \quad (3.2.30)$$

with A''_∂ being the fluid-fluid volumetric interfacial area density, $H_{i\partial}$ the heat transfer coefficient for the i -th side of the interface and T_∂ the interfacial temperature. The interfacial volumetric area density is generally denoted as A'''_{ij} , but since this instance of the methodology is being applied for two-phase flows at most, it is denoted as A''_∂ .

For the case of two-phase flows in which both phases belong to the same chemical species (e.g. flows where mass transfer between the phases is possible via phase change), the interfacial temperature is set to the saturation temperature according to user-selected saturation models. Possible imbalances in the heat flux on each side of the interface (which cannot store energy) are what drives phase change and the basis of how the mass transfer term Γ is calculated. Otherwise, if the two phases belong to chemically different species and no thermally-driven phase change is to be modelled, the interfacial temperature is calculated so that the heat flux on both sides of the interface is conserved.

Mass transfer The modelling of mass transfer is based on energy conservation considerations. Let us consider the sum of all inter-phase transfer terms as of their first introduction in sub-section 2.3.4 in a two-phase scenario for a

given phase i :

$$q_{\alpha,i \rightarrow j} + q_{\kappa,i \rightarrow j} = \Gamma_{i \rightarrow j} h_i + q_{i \rightarrow \partial} \quad (3.2.31)$$

where it is recalled that $q_{\kappa,i \rightarrow j} \equiv q_{\kappa,i \rightarrow \partial} \equiv q_{i \rightarrow \partial}$ (and similarly $q_{\kappa,j \rightarrow i} \equiv q_{j \rightarrow \partial}$), where \equiv means that these relations exclusively represent a change in notation. By imposing energy conservation across the interface:

$$\begin{aligned} q_{\alpha,i \rightarrow j} + q_{\kappa,i \rightarrow j} &= -(q_{\alpha,j \rightarrow i} + q_{\kappa,j \rightarrow i}) \Rightarrow \\ \Gamma_{i \rightarrow j} h_i + q_{i \rightarrow \partial} &= -\Gamma_{j \rightarrow i} h_j + q_{j \rightarrow \partial} \end{aligned} \quad (3.2.32)$$

The latent heat $L_{i \rightarrow j}$ associated with phase change is introduced starting from its physical definition as:

$$L_{i \rightarrow j}(p, T) = h_j(p, T) - h_i(p, T) \quad (3.2.33)$$

which might be negative or positive depending on which phase is the liquid and which is the vapour, and whose general dependence on temperature and pressure was highlighted. Please note that while physically a latent heat is always positive, the numerical definition provided by 3.2.33 will result in a mass transfer term that is sensitive to label inversion, as it should be, i.e. that results in $\Gamma_{i \rightarrow j} = -\Gamma_{j \rightarrow i}$. By combining 3.2.33 with 3.2.32 it is easy to show that:

$$\Gamma_{i \rightarrow j} = \frac{q_{i \rightarrow \partial} + q_{j \rightarrow \partial}}{L_{i \rightarrow j}} \quad (3.2.34)$$

This formulation for the mass transfer is known as a *heat conduction limited* model, which is commonly employed in sub-channel and system codes in which inter-phase heat transfer phenomena are modelled via a two-resistance approach. In particular, a heat-conduction limited approach ensures that energy is conserved at the inter-phase interface (i.e. the interface cannot store any energy), so that any imbalance between the heat fluxes at each side of the interface is “absorbed” in the volumetric mass transfer rate $\Gamma_{i \rightarrow j}$.

This formulation always requires a certain degree of super-heating or sub-cooling of the phase bulk to drive phase change. Because of this, some adjustments to the model can prove beneficial, as discussed with the following example involving a liquid phase i and a vapour phase j . If the vapour is at saturation and the liquid is super-heated, then $q_{i \rightarrow \partial} > 0$ and $\Gamma_{i \rightarrow j} = q_{i \rightarrow \partial}/L_{i \rightarrow j}$. The heat contribution that figures on the right-hand side (RHS) of the liquid enthalpy equation is $-q_{l\delta}$, which by definition of how the mass transfer term is computed, results in $q_{i \rightarrow \partial} = \Gamma_{i \rightarrow j} L_{i \rightarrow j}$. Conversely, as the solver operates in terms of sensible enthalpy (rather than absolute), the heat contribution on the vapour side is $q_{j \rightarrow \partial} = 0$ as there is (physically) no change in vapour temperature during boiling. The same logic applies in the opposite scenario with a sub-cooled vapour i and a liquid j at saturation. Thus, the interfacial heat transfer terms $q_{i \rightarrow \partial}$ that figures in the enthalpy equations constitute, during phase change, the latent heat contribution of the phase change process. Let us now provide some physical intuition for the intrinsic enthalpy change contribution $q_{\alpha,i \rightarrow j} = \Gamma_{i \rightarrow j} h_i$ instead.

Let us focus on a fluid mass dm . The enthalpy contribution associated with said mass is $dm h_i$. Thus, when such a mass of liquid is being removed e.g. due to phase change, in addition to the latent heat $L_{i \rightarrow j}$, the intrinsic contribution h_l is accounted by $q_{\alpha,i \rightarrow j}$. Physically, this contribution should be evaluated at saturation enthalpy, i.e. $h_{sat,i}$, as it is at saturation at which the fluid is “disappearing” or “reappearing”. However, as introduced earlier, a super-heat or sub-cooling of the fluid is required for phase change to occur, so that $h_i > h_{sat,i}$ if phase i is a boiling liquid or $h_i < h_{sat,i}$ if it is a condensing vapour. If one were to remove a boiling (condensing) fluid from a cell at an enthalpy lower (higher) than the saturation one, the remaining fluid will inevitably remain at a higher (lower) specific enthalpy and thus, at a higher (lower) temperature (as it is also discussed in the TRACE system code guide [6], pp. 12 – 13). Thus, to prevent a so-called “thermal run-away”, the intrinsic enthalpy change contribution is always evaluated as follows:

$$q_{\alpha,i \rightarrow j} = \begin{cases} \Gamma_{i \rightarrow j} h_i & \Gamma_{i \rightarrow j} \geq 0 \\ \Gamma_{i \rightarrow j} h_{sat,i} & \Gamma_{i \rightarrow j} < 0 \end{cases} \quad (3.2.35)$$

In order to be consistent with this approach of managing the intrinsic enthalpy change contribution, the mass transfer is calculated with an adjusted latent heat:

$$\Gamma_{i \rightarrow j} = \frac{q_{i \rightarrow \partial} + q_{j \rightarrow \partial}}{L_{i \rightarrow j}^{adj}} \quad (3.2.36)$$

with the adjusted latent heat being calculated as [6]:

$$L_{i \rightarrow j}^{adj} = \begin{cases} L_{i \rightarrow j} - |h_i - h_{i,sat}| & \Gamma_{i \rightarrow j} \geq 0 \\ L_{i \rightarrow j} - |h_j - h_{j,sat}| & \Gamma_{i \rightarrow j} < 0 \end{cases} \quad (3.2.37)$$

with $L_{i \rightarrow j}$ being the latent heat computed by the user-selected latent heat model.

3.2.4 Structure thermal modelling

Modelling the structure in a coarse-mesh context consists in a hydraulic modelling aspect and a thermal modelling aspect. On the hydraulic side, relevant quantities consist of the structure volume fraction, hydraulic diameter, surface area density and tortuosity, whose nature and derivation has been discussed in chapter 2. The focus hereby shifts on the description of the modelling of the thermal component.

By thermal structure modelling we mean the representation of some possibly power-producing underlying structures with specific geometries (e.g. nuclear fuel pins), whose internal temperature profiles resolution can be of interest. Some of these models thus rely on a secondary sub-mesh, which exists within each global mesh cell, to resolve these profiles. For this reason, the

energy conservation equations associated with these structures are generally solved on the global mesh on a cell-by-cell basis.

From the perspective of thermally coupling the structures to the fluid, regardless of the specifics of the structure thermal model, while each phase can transfer heat to and from the structure individually via the $q_{i \rightarrow s}$ term, as seen earlier, the structure temperature/enthalpy is updated by assuming it exchanges heat with the two-phase mixture m as a whole, namely:

$$q_{m \rightarrow s} = A_s''' H_{ms} (T_m - T_s) \quad (3.2.38)$$

where H_{ms} is a certain mixture-structure heat transfer coefficient and T_m is a certain mixture temperature. These quantities are calculated so to assure energy conservation, meaning $q_{m \rightarrow s} = q_{i \rightarrow s} + q_{j \rightarrow s}$. However, this provides a single constraint for the determination of two variables, H_{ms} and T_m , so that one of the two should be modelled a priori in some way. A reasonable choice is to model H_{ms} as:

$$H_{ms} = f_{A_s''',i} H_{is} + f_{A_s''',j} H_{js} \quad (3.2.39)$$

which physically consists in assuming that the heat transfer between each fluid phase and the structure happens in parallel over the contact fraction $f_{A_s''',i}$, $f_{A_s''',j}$ of each phase with the available total structure volumetric area. From this choice of the mixture heat transfer coefficient, it follows from energy conservation consideration that the mixture temperature must be modelled as:

$$T_m = \frac{f_i H_{is} T_i + f_j H_{js} T_j}{f_i H_{is} + f_j H_{js}} \quad (3.2.40)$$

Equations 3.2.38 through 3.2.40 provide a way to define the structure-mixture heat flux that is ultimately used as a boundary condition for the structure-specific power models that are responsible for updating the structure surface temperature. The possible models are now introduced.

0-D structure Within each mesh cell where it exists, this model represents a structure that can be characterized uniquely by an average temperature value T_s , density ρ_s and specific heat capacity $c_{p,s}$, and possibly an internal power density source term $q_{int,s}$. Thus, this model does not rely on a sub-mesh to update the structure temperature. In particular, this model assumes that the surface temperature is representative of the average structure temperature T_s , so that the structure energy governing equation is modelled as:

$$\rho_s c_{p,s} \frac{\partial}{\partial t} T_s = q_{int,s} + q_{m \rightarrow s} \quad (3.2.41)$$

Heat conduction between adjacent cells in the global mesh is neglected. Please note that $q_{m \rightarrow s}$ is treated semi-implicitly as it entails a dependence on T_s , which is the solution variable.

1-D uniform pin structure Within each mesh cell where it exists, this model represents a cylindrical pin of outer radius r_O , inner radius r_I (which can be 0), density ρ_s and heat capacity $c_{p,s}$, and possibly an internal power density source term $q_{int,s}$. This model relies on a sub-mesh within each global mesh cell, with the sub-mesh representing the radial domain of the pin, and the fuel pin temperature profile $T = T(r)$ being defined on such sub-mesh. An energy equation is solved in the following form:

$$\rho_s c_{p,s} \frac{\partial}{\partial t} T - \kappa_s \nabla \cdot (\nabla T) = q_{int,s} \quad (3.2.42)$$

A convective (i.e. Robin) boundary condition on the outer pin surface (i.e. at $r = r_O$ with an outward surface normal \mathbf{n}) describes the thermal coupling with the fluid:

$$-\kappa_s \nabla T|_{r_O} \cdot \mathbf{n} = q_{m \rightarrow s} \quad (3.2.43)$$

Conversely, a zero-gradient boundary conditions is applied at $r = r_I$. The solution of 3.2.42 with respect to the sub-mesh radial temperature field T within each global mesh cell allows to update the structure surface temperature $T_s = T(r_O)$. The energy equation 3.2.42 is solved via a Finite Volume discretization within each global mesh cell. A more detailed explanation of how this is achieved is presented in appendix A. As for the 0-D model discussed earlier, heat conduction between different sub-meshes in adjacent global mesh cells is neglected.

1-D nuclear fuel pin Within each mesh cell where it exists, this model represents a nuclear fuel pin consisting of two layers: fuel and cladding. Each layer is characterised by an inner and outer radius, density and heat capacity, and an optional power density source term in the fuel. A gap layer separates the fuel from the cladding as is characterised uniquely by a certain gap conductance value. The modelling via a radial sub-mesh and the coupling to the fluid is the same as discussed for the 1-D uniform pin. A detailed derivation of the Finite Volume Method (FVM)-discretized governing equations and a more detailed description of the integration of this approach in a coarse-mesh context is provided and discussed in appendix A.

3.2.5 Turbulence modelling and tortuosity remarks

The macroscopic governing equations for mass, momentum and enthalpy conservation, 3.1.1 3.1.2, 3.1.3 were derived starting from their fundamental counterparts that hold at microscopic level in a neighbourhood of a fluid phase. If the volume averaging process was instead performed on the microscopic Reynolds-Averaged Navier Stokes (RANS) equations, it can be shown that the same formulations for the momentum and enthalpy equations presented in 3.1.2, 3.1.3 would have been obtained, with the addition of the volume average of the Reynolds stress tensor, which is treated differently depending on the turbulence modelling approach.

The present approach relies on an eddy-viscosity-based model for the treatment of the Reynolds stress tensor, namely the standard $k - \varepsilon$ model [42]. The task of deriving the volume-averaged $k - \varepsilon$ equations would fundamentally follow the same procedure outlined in chapter 2, and would lead to additional source terms in both equations, as seen for the volume average of a generic transport equation, which requires closure. While efforts in this direction do exist, such as those by Nakayama et al. [43], Chandersis et al. [44], who investigated the same problem, namely modelling turbulence in engineering structures treated as porous media, the formulation adopted in this code was inherited from the work of Fiorina [20] in the development of the GeN-Foam code. This approach avoids solving the complete volume-averaged $k - \varepsilon$ equations within the porous structure, rather, it forces the turbulence kinetic energy k_i and dissipation rate ε_i to converge to equilibrium values $k_{\infty,i}$, $\varepsilon_{\infty,i}$, namely values for a fully developed flow, so that the equations that are solved in the mesh cells that contain an averaged structure (i.e. where $\alpha_s \neq 0$) are:

$$\frac{\partial}{\partial t} (\alpha_i \rho_i k_i) + \nabla \cdot (\alpha_i \rho_i \mathbf{u}_i k_i) = \alpha_i \rho_i \omega (k_{\infty,i} - k_i) \quad (3.2.44)$$

$$\frac{\partial}{\partial t} (\alpha_i \rho_i \varepsilon_i) + \nabla \cdot (\alpha_i \rho_i \mathbf{u}_i \varepsilon_i) = \alpha_i \rho_i \omega (\varepsilon_{\infty,i} - \varepsilon_i) \quad (3.2.45)$$

These equilibrium values can be computed via correlations for the turbulence intensity $I_{t,i}$ and turbulence length scale $L_{t,i}$:

$$k_{\infty,i} = \frac{3}{2} (|\mathbf{u}_i| I_{t,i})^2 \quad (3.2.46)$$

$$\varepsilon_{\infty,i} = C_{\mu}^{\frac{3}{4}} \frac{k_i^{\frac{3}{2}}}{L_{t,i}} \quad (3.2.47)$$

with those correlations generally in the following form [45] (for pipe or channel flows):

$$I_{t,i} = c_{I,0} Re_{is}^{c_{I,1}} \quad (3.2.48)$$

$$L_{t,i} = c_L D_h \quad (3.2.49)$$

with $c_{I,0}$, $c_{I,1}$, c_L experimentally-determined coefficients.

It is clear that this methodology presently only applies to single-phase flows, as the investigation of two-phase flows with a $k - \varepsilon$ turbulence modelling will result in additional source terms for the turbulent kinetic energy and dissipation equations due to the interaction between fluid phases. While some approaches to model these terms are presently included in the code (such as the approach by Lahey [46]), these are valid exclusively outside of porous regions as the modelling of two-phase turbulence in porous media was not deemed of central importance for this work. This was mainly due to the strongly advective two-phase flow scenarios of interest to this work. Given that eddy-viscosity turbulence models, such as any $k - \varepsilon$ model, have the final effect of

modifying the effective kinematic and thermal diffusivities $\nu_{\text{eff},i}$, $a_{\text{eff},i}$ used in the conservation equations, this was deemed acceptable.

However, an important clarification should be made relative to the role of the $k - \varepsilon$ model when used for predicting turbulence parameters in the fuel region. It is recalled that, in such region, a coarse-mesh model is intended to operate at the same level of spatial resolution as a sub-channel code, by employing the same correlations for modelling momentum and heat transfer phenomena. These correlations are typically already inclusive of diffusive effects arising from turbulence phenomena, so that employing the effective momentum ν_{eff} and thermal a_{eff} diffusivities in the governing equations (in such regions) instead of the molecular values can lead to an over-estimation of diffusive effects. For this reason, the code offers the choice to specify which regions should employ the effective diffusivities in place of the molecular ones. It is in fact recalled that while the code is intended to operate at sub-channel resolution in fuel/core regions, the mathematical formulation of the governing equations would remain the same if said regions were modelled via a fully-geometrically resolved mesh, thus having the code operate like a standard CFD one (the coarse-mesh reverts to a fine-mesh approach). In this scenario, correlations are no longer required and the effective thermal diffusivities (coupled with a standard $k - \varepsilon$ model) are required. A last clarification is made then on the usefulness of the equilibrium $k - \varepsilon$ model when analyzing a core region with a coarse-mesh approach, as the effective diffusivities are not to be used. The principal reason lies in enabling the prediction of meaningful k , ε values at the outlet of such regions, where they connect to core plena (if an entire reactor vessel is being modelled), as in plena regions (i.e. “clear-fluid” regions) the standard $k - \varepsilon$ model equations are being solved and these require meaningful values at plenum inlet.

On a similar note lies the treatment of tortuosity \mathbf{T}_i , which is a term that is only present in the diffusive terms in equations 3.1.2, 3.1.3 as a diffusivity multiplier. In practice it can be thought of as transforming the diffusivities into tensor quantities. From the definition of tortuosity in equation 2.3.37, is it clear that its treatment in two-phase flows is significantly more complex than in one-phase. In one-phase scenarios, the domain over which the integral that defines the tortuosity is computed is constant in time (assuming an immovable structure), as it consists of the fluid-structure interface. Thus, this tortuosity can be computed once and set as a constant for the rest of the simulation. In two-phase scenarios instead, the integration domain will span both the fluid-structure interface *and* the fluid-fluid interface, whose geometry evolves in time and is fundamentally unknown in a coarse-mesh context. This unknown fluctuating tortuosity component can be interpreted as an additional turbulence-related diffusivity multiplier. This would require some closure that, for the reasons illustrated for two-phase turbulence modelling, was not pursued in the present work. Thus, all the tortuosity tensors used in this work in two-phase scenarios consist of the same tensor that would be obtained in a one-phase scenario, which depends exclusively on the one-fluid-structure interface

geometry, i.e. the structure geometry alone.

3.3 Solution algorithm

The previous section has outlined how to provide adequate closure and modelling for all the terms that required so in equations 3.1.1, 3.1.2, 3.1.3. The present section describes how this coupled system of equations is solved. In virtue of the underlying OpenFOAM framework and the significantly larger ease of implementation, the solution algorithm for the various equations is *segregated*, meaning that the equations are discretized and solved individually and the process iterated (in some manner, to be discussed) until convergence. This is contrasted to *matrix-coupled* approaches, in which all equations are discretized so to result into a single matrix that is solved at once. While the coupled approach still requires iterations to resolve non-linearities of the individual equations, it has potential to significantly improve convergence, albeit matrix preconditioning can become an issue and the method becomes increasingly more computationally expensive for larger computational domains [47]. To this day, the vast majority of computer codes for the simulation of the Navier-Stokes equations specifically rely on segregated methods.

The discussion will be structured in the following way. Section 3.3.1 presents the numerical implementation and solution steps for the fluid mechanical behaviour, namely the Navier-Stokes equations. Section 3.3.2 presents the enthalpy solution algorithm. This is inclusive of the presentation and discussion of the novel pressure-velocity coupling solution algorithm. Section 3.3.3 presents the global solution algorithm that couples the Navier-Stokes and the enthalpy equations, together with the overall code solution procedure.

3.3.1 Solution of the Navier-Stokes equations

The system of equations governing fluid flow without thermal considerations is constituted by the Navier-Stokes equations, namely 3.1.1, 3.1.2 in a coarse-mesh context. The fundamental problematics of solving the Navier-Stokes equations (in any CFD context) comes from the fact that, at its core, while there is an equation for density (ρ in a one-phase context, $\alpha_i \rho_i$ in a generic multi-phase context) and an equation for velocity \mathbf{u} , there is no equation for pressure p . This is known as the *pressure-velocity coupling* problem and can be tackled with two fundamentally different approaches, namely a *density-based* approach or a *pressure-based* approach, each of which have resulted in a plethora of specific solution algorithms.

Density-based algorithms are predominantly employed for one-phase, highly compressible flows commonly found e.g. in aeronautic applications. These algorithms take advantage of an additional equation of state to relate the fluid pressure to its density. The solution process would generally consist of the following steps: 1) solve the continuity equation to obtain the fluid density with

the latest available values for fluid velocity; 2) employ the equation of state to analytically compute the pressure field; 3) solve the momentum equation to obtain the fluid velocity; 4) iterate these steps in some way until convergence is achieved for the time step under consideration. This approach is inapplicable for the simulation of incompressible flows (due to the independence of pressure and fluid density) or for the simulation of liquids, which are typically characterized by very small values of compressibility $\psi \equiv \frac{1}{\rho} \frac{\partial \rho}{\partial p}$, which means that small variations in density will lead to very large variations in pressure, hindering the stability of the method.

For these circumstances, pressure-based algorithms are preferred. As the equation of state that relates pressure and density is either too density-sensitive or is not formulated in terms of a density-pressure dependence at all (e.g. equations of state that relate fluid density to its temperature, which are extensively used in a plethora of engineering applications for modelling liquid flows), a different approach to computing a pressure field is employed. In particular, pressure-based methods combine the momentum equation and the continuity equation to construct an equation for pressure. Thus, the pressure field is computed so that it ensures mass continuity.

Pressure-based approaches were first developed by Spalding, Patankar [48] with the Semi-Implicit Method for Pressure Linked Equations (SIMPLE) for the treatment of steady-state flows and by Issa [49] who developed the Pressure Implicit with Splitting of Operators (PISO) algorithm for the treatment of transient flows. While the two algorithms fundamentally differ in how certain steps of the solution procedure are iterated, the underlying basis is the same, namely the construction and solution of a pressure equation that ensures continuity. The key steps of pressure-based algorithms can be summarized as follows:

1. construct and solve the momentum equation based on an explicit evaluation of the pressure gradient. This yields a new velocity field that does not necessarily ensure mass conservation. This step is referred to as *velocity predictor* step;
2. construct and solve a pressure equation from the momentum and continuity equations. This and the next step are referred to as *pressure correction* step;
3. correct the velocities with the newly estimated pressure gradient to ensure mass conservation.

To “construct” means to discretize according to the overarching numerical framework, represented by the FVM in our case, so to obtain a matrix representation of the linearized equation of interest. The specifics of how these steps are iterated is what fundamentally differentiates the various pressure-based solution algorithms (e.g. SIMPLE, SIMPLER[50], PISO, etc.). This work employs the merged PISO-SIMPLE (PIMPLE) algorithm, which fundamentally retains

the transient nature of the PISO algorithm while gaining some of the advantages of the SIMPLE algorithm, chiefly among which is the possibility to overcome numerical limitations imposed by the Courant–Friedrichs–Lewy (CFL) condition, which otherwise poses constraints on the maximum time step size depending on the mesh size and flow velocity for numerical stability reasons [51].

So far, this discussion has been oblivious to the fact that the system of interest is potentially a two-fluid system, and this was done for the sake of introducing some fundamental concepts in tackling the problem of pressure-velocity coupling. Under the assumption that all phases share a single pressure field, which is the default approach for the simulation of multi-phase flows across a wide range of computer codes, from general-purpose CFD software such as ANSYS [45] to nuclear system-codes [6], the key steps outlined earlier remain the same, as it will be shown in greater detail. However, a further difficulty is introduced by two aspects: the presence of further variables, namely the phase fractions α_i ; and the coupling between the two fluid momentum equations via the transfer terms $\mathbf{M}_{j \rightarrow i}$, which are a function of both phase velocities as seen in equation 3.2.22.

The phase fractions are obtained by solving the continuity equations, as it will be shown in the next paragraphs. The treatment of the momentum coupling is discussed afterwards, as it resulted in an improved implementation of the original Partial Elimination Algorithm (PEA) by Spalding [52].

3.3.1.1 Solution of the continuity equation

The continuity equation is solved to yield the new time step phase fractions α_i provided knowledge of all the other fields, ρ_i , \mathbf{u}_i , $\Gamma_{i \rightarrow j}$. This step is performed at the beginning of each new PIMPLE iteration and/or time step, before any of the other steps that concern the treatment of pressure-velocity coupling. It is recalled the that the continuity equation for the i -th phase in a two-phase system is:

$$\frac{\partial}{\partial t} (\alpha_i \rho_i) + \nabla \cdot (\alpha_i \mathbf{u}_i \rho_i) = -\Gamma_{i \rightarrow j} \quad (3.3.1)$$

Compressibility effects due to density changes, as well as phase change proper, can be isolated on the left-hand side (LHS) by expanding the derivative terms:

$$\frac{\partial}{\partial t} \alpha_i + \nabla \cdot (\alpha_i \mathbf{u}_i) = S_{\alpha,i} \quad (3.3.2)$$

$$S_{\alpha,i} = \frac{1}{\rho_i} \left(\alpha_i \frac{\partial}{\partial t} \rho_i + \alpha_i \mathbf{u}_i \cdot \nabla \rho_i - \Gamma_{i \rightarrow j} \right) \quad (3.3.3)$$

Please note that in a two-phase system, it is sufficient to solve only one phase fraction equation. The phase fraction source term $S_{\alpha,i}$ is evaluated explicitly and the phase fraction equation is solved via the Multidimensional Universal Limiter with Explicit Solution (MULES) algorithm [53], which is a particular instance of a Flux Corrected Transport (FCT) technique [54].

FCT was originally devised as a method to solve the continuity equations using a hybrid discretization of the advective term: less diffusive than a low-order scheme (e.g. upwind) yet not resulting in a potentially unbounded phase fraction that a higher-order scheme might cause. This is particularly relevant close to steep phase fraction gradients, thus to interfaces between representative volumes (RVs) that contain structures and RVs that do not in a coarse-mesh context.

Let us consider a Finite Volume discretization of the phase fraction equation 3.3.2. Subsequent to volume integration and the application of the Gauss-Green theorem, as discussed in section 2.2, it can be written in each mesh cell c as:

$$\frac{\partial}{\partial t} \alpha_{i,c} + \frac{1}{V} \sum_{f \in c} \phi_{\alpha,i,f} = S_{\alpha,i,c} \quad (3.3.4)$$

where the second term on the LHS is the advection contribution, which can be explicitly evaluated as a sum over cell faces f of the volumetric flux $\phi_{\alpha,i}$, which is a field defined over mesh cell faces. The core idea of FCT is to construct $\phi_{\alpha,i}$ as a combination of a flux obtained via a low-order interpolation scheme $\phi_{\alpha,i,LO} = (\alpha_i \mathbf{u}_i)|_{f,LO} \cdot \mathbf{S}_f$ and a flux obtained via a high-order interpolation scheme $\phi_{\alpha,i,HO} = (\alpha_i \mathbf{u}_i)|_{f,HO} \cdot \mathbf{S}_f$:

$$\phi_{\alpha,i} = \phi_{\alpha,i,LO} + \lambda (\phi_{\alpha,i,HO} - \phi_{\alpha,i,LO}) \quad (3.3.5)$$

in which λ are limiting factors that quantify the maximum possible contribution from a high-order flux that guarantees boundedness.

All the different implementations of a FCT technique consist in variations on how λ is computed. The MULES algorithm is already implemented in several OpenFOAM solvers and was adopted in this work. The general idea behind MULES is to compute λ from mass balance and allowable phase fraction considerations on a cell-by-cell basis by considering that, physically, the diffusion process alone cannot create new maxima or minima in a transported scalar quantity. The detailed procedure at the heart of the MULES algorithm is described in [53], [55] and is conceptually similar to the limiters devised by Zalesak [56].

3.3.1.2 Implementation of the momentum equation

Before discussing the solution procedure itself, some remarks on the numerical implementations of the momentum equations, namely 3.1.2 are made. For the i -th phase, this implementation is as follows: The general form of the momentum conservation equation for the i -th phase is:

$$\begin{aligned} \frac{\partial}{\partial t} (\alpha_i \rho_i \mathbf{u}_i) + \nabla \cdot (\alpha_i \rho_i \mathbf{u}_i \otimes \mathbf{u}_i) = \\ - \alpha_i \nabla p + \nabla \cdot (\alpha_i \rho_i \nu_{\text{eff},i} \mathbf{T}_i \cdot (\nabla \mathbf{u}_i + (\nabla \mathbf{u}_i)^T - \frac{2}{3} (\nabla \cdot \mathbf{u}_i) \mathbf{I})) + \mathbf{S}_{\mathbf{u},i} \end{aligned} \quad (3.3.6)$$

where the source terms have been grouped in $\mathbf{S}_{\mathbf{u},i}$:

$$\begin{aligned}\mathbf{S}_{\mathbf{u},i} = & K_{ij}(\mathbf{u}_j - \mathbf{u}_i) - \mathbf{K}_{is} \cdot \mathbf{u}_i + f_{vm}\rho_c \left(\frac{\partial}{\partial t} \mathbf{u}_i + \mathbf{u}_i \cdot \nabla \mathbf{u}_i - \frac{\partial}{\partial t} \mathbf{u}_j \right) + \\ & + \Gamma_{j \rightarrow i} \mathbf{u}_j - \Gamma_{i \rightarrow j} \mathbf{u}_i + \alpha_i \rho_i \mathbf{g} - \xi_i \mathbf{u}_i\end{aligned}\quad (3.3.7)$$

This formulation is identical to the one presented in 3.1.2, yet expanded to showcase the formulation of the diffusive and transfer terms, and except for the last term added in $\mathbf{S}_{\mathbf{u},i}$, namely $\xi_i \mathbf{u}_i$. This is a purely numerical correction that accounts for continuity errors. The continuity error ξ is an explicit numerical evaluation of the continuity equation so that:

$$\xi_i = \frac{\partial}{\partial t} (\alpha_i \rho_i) + \nabla \cdot (\alpha_i \rho_i \mathbf{u}_i) + \Gamma_{i \rightarrow j} \quad (3.3.8)$$

The continuity errors should be physically null at all times. In practice, these quantify how well the pressure-velocity coupling was resolved (as the pressure comes from the solution of an equation that is derived from the continuity equation). These can be also thought of as spurious mass sources for phase i (when $\xi_i > 0$) or sinks (when $\xi_i < 0$) that arise from a non-conservative momentum field, consequence of the segregated nature of the algorithm.

In the case of the momentum equation, the spurious momentum sources associated with the continuity error is $\xi_i \mathbf{u}_i$, which is thus accounted for in the momentum equation. As it will be seen, this is accounted in the enthalpy equation as well. The continuity equation already takes this correction term into account as the phase fraction source term $S_{\alpha,i}$ defined in 3.3.3 is evaluated explicitly for the solution of the continuity equation. By expanding the derivatives in the definition of the continuity error 3.3.8:

$$\alpha_i \frac{\partial}{\partial t} \rho_i + \alpha_i \mathbf{u}_i \cdot \nabla \rho_i = \xi_i - \rho_i \left(\frac{\partial}{\partial t} \alpha_i + \nabla \cdot (\alpha_i \mathbf{u}_i) \right) - \Gamma_{i \rightarrow j} \quad (3.3.9)$$

Taking advantage of this result, it is easy to see that explicitly evaluating the phase fraction source term $S_{\alpha,i}$ via 3.3.3 is *equivalent* to evaluating it as follows, which highlights the fact that continuity errors are accounted indeed:

$$S_{\alpha,i} = \frac{1}{\rho_i} (\xi_i - 2\Gamma_{i \rightarrow j}) - \frac{\partial}{\partial t} \alpha_i + \nabla \cdot (\alpha_i \mathbf{u}_i) \quad (3.3.10)$$

The general discretization procedure of the various terms of a momentum equation in the form of 3.3.6 are achieved by the high-level functions provided by the OpenFOAM environment [23].

In order to enhance the convergence properties of the discretized matrix (e.g. by increasing diagonal dominance), a further manipulation is performed on the fluid-structure drag factor. As discussed earlier, this is a second-order tensor to account for the potential anisotropy of the structure. This term is treated semi-implicitly:

$$\mathbf{K}_{is} \cdot \mathbf{u}_i = \frac{1}{3} \text{tr}(\mathbf{K}_{is}) \mathbf{u}_i + (\mathbf{K}_{is} - \frac{1}{3} \text{tr}(\mathbf{K}_{is}) \mathbb{I}) \cdot \mathbf{u}_i$$

where $tr(\cdot)$ denotes the matrix trace operator. The diagonal contribution of \mathbf{K}_{is} is thus treated implicitly², while the off-diagonal is treated explicitly, as there is no other possible approach in segregated methods (recall that the momentum equation for each phase fundamentally consists of three momentum equations for each scalar component of the phase velocity).

3.3.1.3 Two-phase pressure-velocity coupling

A key element of equations 3.3.6 is the inter-phase momentum coupling term arising from drag, i.e. $K_{ij}(\mathbf{u}_j - \mathbf{u}_i)$, which can benefit from specific treatments to enhance the stability of the algorithm. This paragraph presents how the pressure-velocity coupling is resolved in the developed algorithm, inclusive of novel developments for the treatment of the inter-phase momentum coupling.

For notation clarity and simplicity, the two phases will be labelled as 1 and 2, while the fluid-fluid drag factor will be indicated without any subscripts as K , and referred to as coupling coefficient.

Assume the momentum equations can be linearized and discretized according to a chosen set of discretization schemes, so that the momentum equations 3.3.6 (one for each of the two phases) can be expressed as:

$$\begin{aligned}\hat{M}_1 \mathbf{u}_1 &= \mathbf{b}_1(p) + K(\mathbf{u}_2 - \mathbf{u}_1) \\ \hat{M}_2 \mathbf{u}_2 &= \mathbf{b}_2(p) + K(\mathbf{u}_1 - \mathbf{u}_2)\end{aligned}\quad (3.3.11)$$

in which \hat{M}_i is the discretized momentum transport operator, $\mathbf{b}(p)_i$ is the source term inclusive of the pressure gradient, and where the fluid-fluid drag term is treated separately.

The first option to treat the coupling term is an explicit representation where the latest available values for the phase velocities are employed. This treatment is simple but potentially unstable for large values of the coupling coefficient.

The second option is to treat it semi-implicitly by modifying the operator diagonal as follows:

$$\hat{M}_i^* = \hat{M}_i + K\hat{I} \quad (3.3.12)$$

as this increases the diagonal dominance the discretized transport operator. Nonetheless, an explicit evaluation of the remaining $K\mathbf{u}_j$ on the left hand side is still required. This approach is generally referred to as the Partially Implicit treatment [57].

The third option results from noticing that 3.3.11 is a system of two equations in two variables³, so that there could be a way to formulate each equation in terms of a single variable, thus eliminating the need for explicit coupling

²As \mathbf{K}_{is} is itself a function of velocity, as seen in 3.2.17, 3.2.14, it is the explicit evaluation of \mathbf{K}_{is} that is treated implicitly, so that formally this treatment is semi-implicit.

³Technically, if the computational domain consists of N mesh cells, the actual problem consists of of $6N$ algebraic relations and $6N$ variables, namely 3 velocity vector components for each of the 2 phases for each cell.

evaluations. This idea for the treatment of the momentum equations is known as Partial Elimination [52], and there are a number of ways to implement. The implementation that was derived and adopted is novel in the domain of OpenFOAM solvers and will be introduced in this sub-section.

Let us return to equations 3.3.11. It is possible to algebraically manipulate them to remove the coupling, in line with Partial Elimination ideas:

$$\begin{aligned} (\hat{M}_1 + K(\hat{M}_2^{-1}\hat{M}_1 + \hat{I}))\mathbf{u}_1 &= \mathbf{b}_1(p) + K\hat{M}_2^{-1}(\mathbf{b}_1(p) + \mathbf{b}_2(p)) \\ (\hat{M}_2 + K(\hat{M}_1^{-1}\hat{M}_2 + \hat{I}))\mathbf{u}_2 &= \mathbf{b}_2(p) + K\hat{M}_1^{-1}(\mathbf{b}_1(p) + \mathbf{b}_2(p)) \end{aligned} \quad (3.3.13)$$

It is clear from the structure of system 3.3.13 that the coupling between the phases is now treated in a fully implicit manner. However, the numerical assembly of the system is computationally more intensive due to the additional matrix inversion operations. The idea is then to not solve the momentum equations, but to take advantage of the discretized momentum matrices to assemble the pressure equation. The velocities can then be algebraically reconstructed from the pressure gradient as it is generally done in a traditional pressure correction step.

The pressure equation can be assembled via a number of approaches. In a shared phase pressure framework, it is obtained from the sum of the continuity equations 3.3.1 for the two phases:

$$\nabla \cdot (\alpha_1 \mathbf{u}_1 + \alpha_2 \mathbf{u}_2) = S_\alpha(p) \quad (3.3.14)$$

in which $S_\alpha(p) = S_{\alpha,1}(p) + S_{\alpha,2}(p)$ is the sum of phase fraction source terms (i.e. due to phase transfer and/or compressibility effects), which are possibly pressure-dependent in some way (as $S_{\alpha,i}$ depends on ρ_i , which can be pressure-dependent depending on the fluid equation of state). The time derivative of the sum of phase fractions is null assuming that the volume fraction of the structure is constant in time. In order to obtain an equation for pressure from 3.3.14, an algebraic relationship between the phase velocities and pressure is required. To this end, let us focus our attention back to the momentum equations.

The momentum transport operator \hat{M}_i introduced earlier can be decomposed in a diagonal part $\hat{M}_{D,i}$ and an off-diagonal part $\hat{M}_{OD,i}$. Since the diagonal coefficients are the same for all velocity components of a phase, these will be represented with a scalar field $A_i = \hat{M}_{D,i}$. By recalling that the source term $\mathbf{b}_i(p) = \mathbf{b}_i^* - \alpha_i \nabla p$ is inclusive of the pressure gradient and by defining:

$$\mathbf{H}_i = -\hat{M}_{OD,i}\mathbf{u}_i + \mathbf{b}_i^* \quad (3.3.15)$$

as the explicit evaluation of the off-diagonal velocity contributions inclusive of other source terms, an algebraic relationship between each velocity field and the pressure gradient can be obtained from either 3.3.11 or 3.3.13 [21][58]. Standard OpenFOAM approaches⁴ derive it from equation 3.3.11, which still

⁴As of OpenFOAM v2006.

entails an explicit evaluation of the coupling term, while the developed approach derives it from 3.3.13, as it is now discussed.

By decomposing the momentum transport operator \hat{M}_i in a diagonal part A_i and an explicit off-diagonal contribution \mathbf{H}_i as described before and defining:

$$A_i^{**} = A_i + K \left(1 + \frac{A_i}{A_j} \right) \quad (3.3.16)$$

the following relationship can be obtained from 3.3.13:

$$\begin{aligned} \mathbf{u}_1 &= \frac{1}{A_1^{**}} \left(\left(1 + \frac{K}{A_2} \right) \mathbf{H}_1 + \frac{K}{A_2} \mathbf{H}_2 - \left(\left(1 + \frac{K}{A_2} \right) \alpha_1 + \frac{K}{A_2} \alpha_2 \right) \nabla p \right) \\ \mathbf{u}_2 &= \frac{1}{A_2^{**}} \left(\left(1 + \frac{K}{A_1} \right) \mathbf{H}_2 + \frac{K}{A_1} \mathbf{H}_1 - \left(\left(1 + \frac{K}{A_1} \right) \alpha_2 + \frac{K}{A_1} \alpha_1 \right) \nabla p \right) \end{aligned} \quad (3.3.17)$$

The relationship established by 3.3.17 thus accounts for Partial Elimination (as there is no coupling between the velocities) and provides the desired algebraic relationship between pressure and velocity. Thus, by taking advantage of equations 3.3.14 and 3.3.17, an equation for pressure is obtained:

$$\begin{aligned} &\nabla \cdot \left(\left(\frac{\alpha_1}{A_1^{**}} \left(1 + \frac{K}{A_2} \right) + \frac{\alpha_2 K}{A_2^{**} A_1} \right) \mathbf{H}_1 + \left(\frac{\alpha_2}{A_2^{**}} \left(1 + \frac{K}{A_1} \right) + \frac{\alpha_1 K}{A_1^{**} A_2} \right) \mathbf{H}_2 \right) + \\ &- \nabla \cdot \left(\left(\frac{\alpha_1}{A_1^{**}} \left(\alpha_1 + \frac{K}{A_2} (\alpha_1 + \alpha_2) \right) + \frac{\alpha_2}{A_2^{**}} \left(\alpha_2 + \frac{K}{A_1} (\alpha_1 + \alpha_2) \right) \right) \nabla p \right) = \\ &= S_\alpha(p) \end{aligned} \quad (3.3.18)$$

Let us thus summarize the developed approach. First, the discretized momentum transport operators \hat{M}_i in form 3.3.11 are constructed. This enables the evaluation of the modified diagonal coefficients field A_i^{**} via 3.3.16 and the explicit evaluation of off-diagonal and source terms (exclusive of pressure) \mathbf{H}_i as of 3.3.15. These are then used to assemble and solve a pressure equation in form 3.3.18. After this solution and the evaluation of a new pressure gradient, phase velocities are updated according to equations 3.3.17.

For the sake of comparison, it is worth noting that in existing OpenFOAM implementations the pressure-velocity relationship is derived from equation 3.3.11 instead with an implicit treatment of $-K\mathbf{u}_i$, achieved by decomposing the momentum transport operator \hat{M}_i in a diagonal part A_i which is modified as:

$$A_i^* = A_i + K \quad (3.3.19)$$

and an explicit off-diagonal contribution \mathbf{H}_i still evaluated of 3.3.15. The following algebraic relationship between pressure and velocity is thus obtained:

$$\begin{aligned} \mathbf{u}_1 &= \frac{1}{A_1^*} (-\alpha_1 \nabla p + \mathbf{H}_1 + K \mathbf{u}_2) \\ \mathbf{u}_2 &= \frac{1}{A_2^*} (-\alpha_2 \nabla p + \mathbf{H}_2 + K \mathbf{u}_1) \end{aligned} \quad (3.3.20)$$

The relationship established by 3.3.20 is not based on Partial Elimination, as the coupling is clearly not implicit. Nonetheless, it is used to construct and solve a pressure equation from 3.3.14, which consequently also evaluates the coupling contributions explicitly:

$$\nabla \cdot \left(\frac{\alpha_1}{A_1^*} (\mathbf{H}_1 + K \mathbf{u}_2) + \frac{\alpha_2}{A_2^*} (\mathbf{H}_2 + K \mathbf{u}_1) \right) - \nabla \cdot \left(\left(\frac{\alpha_1^2}{A_1^*} + \frac{\alpha_2^2}{A_2^*} \right) \nabla p \right) = S_\alpha(p) \quad (3.3.21)$$

Standard OpenFOAM approach take advantage of Partial Elimination only at the velocity correction step. Unlike the approach that was adopted, velocities are not computed from the same pressure-velocity relationship used to construct the pressure equation. Instead, coupling variables are eliminated from 3.3.20 to obtain:

$$\begin{aligned} \mathbf{u}_1 &= \frac{1}{A_1^* - \frac{K^2}{A_2^*}} \left(-\alpha_1 \nabla p + \mathbf{H}_1 + \frac{K}{A_2^*} (\mathbf{H}_2 - \alpha_2 \nabla p) \right) \\ \mathbf{u}_2 &= \frac{1}{A_2^* - \frac{K^2}{A_1^*}} \left(-\alpha_2 \nabla p + \mathbf{H}_2 + \frac{K}{A_1^*} (\mathbf{H}_1 - \alpha_1 \nabla p) \right) \end{aligned} \quad (3.3.22)$$

Let us thus summarize the exiting OpenFOAM approach. First, the discretized momentum transport operators \hat{M}_i in form 3.3.11 are constructed. This enables the evaluation of the modified diagonal coefficients field A_i^* via 3.3.19 and the explicit evaluation of off-diagonal and source terms (exclusive of pressure) \mathbf{H}_i as of 3.3.15. These are then used to assemble and solve pressure equation in form 3.3.21. After this solution, phase velocities are updated according to equations 3.3.20.

The fundamental difference between the developed and standard OpenFOAM approach thus lies in the fact that the OpenFOAM approach only takes advantage of Partial Elimination in the velocity reconstruction step, as the coupling is treated explicitly in constructing the pressure equations. Conversely, the developed approach applies Partial Elimination in the construction of the pressure equation itself, which thus conditions the convergence properties of its solution, as it is clear from the larger coefficient associated to the pressure gradient in in equation 3.3.18 when compared to 3.3.21. The verification of the developed algorithm via the Method of Manufactured Solutions (MMS) is discussed in section 3.4, while its performance gains compared to the standard OpenFOAM algorithm are discussed for a number of scenarios in section 3.5.

3.3.1.4 One-phase pressure-velocity coupling

In a one-phase scenario with an immovable structure, the Navier-Stokes equations reduce to:

$$\nabla \cdot (\alpha \mathbf{u}) = S_{\alpha,1p} \quad (3.3.23)$$

$$\begin{aligned} \frac{\partial}{\partial t} (\alpha \rho \mathbf{u}) + \nabla \cdot (\alpha \rho \mathbf{u} \otimes \mathbf{u}) &= -\alpha \nabla p + \mathbf{S}_{\mathbf{u},1p} + \\ &+ \nabla \cdot \left(\alpha \rho \nu_{\text{eff}} \mathbf{T} \cdot (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) - \frac{2}{3} (\nabla \cdot \mathbf{u}) \mathbb{I} \right) \end{aligned} \quad (3.3.24)$$

with the source terms $S_{\alpha,1p}$, $\mathbf{S}_{\mathbf{u},1p}$ defined similarly to 3.3.3, 3.3.7 yet without the terms that pertain to mass or momentum transfer with another fluid phase:

$$S_{\alpha,1p} = \frac{1}{\rho} \left(\alpha \frac{\partial}{\partial t} \rho + \alpha \mathbf{u} \cdot \nabla \rho \right) \quad (3.3.25)$$

$$\mathbf{S}_{\mathbf{u},1p} = -\mathbf{K}_{1p,s} \cdot \mathbf{u} + \alpha \rho \mathbf{g} - \xi \mathbf{u} \quad (3.3.26)$$

Given that the structure is immovable, the time derivative term is absent from the continuity equation and the phase fractions (of fluid and structure) are constant in time. Thus, this equation does not need to be solved. However, it used to construct the pressure equation as seen for the two-phase algorithm. By discretizing the momentum matrix in the same way as previously discussed:

$$\begin{aligned} \hat{M} \mathbf{u} &= \mathbf{b}(p) \Rightarrow \\ \Rightarrow \hat{M}_D \mathbf{u} &= -\hat{M}_{OD} \mathbf{u} - \alpha \nabla p + \mathbf{b}^* \Rightarrow \\ \Rightarrow \mathbf{u} &= \frac{1}{A} (\mathbf{H} - \alpha \nabla p) \end{aligned} \quad (3.3.27)$$

with $A \equiv \hat{M}_D$ the field of diagonal coefficient of M , $\mathbf{H} = -\hat{M}_{OD} \mathbf{u} + \mathbf{b}^*$ the sum of the explicit evaluation of the off-diagonal terms with the latest available \mathbf{u} and of the source terms \mathbf{b}^* , as seen before. The pressure equation is constructed by substituting 3.3.27 in the continuity equation 3.3.23:

$$\nabla \cdot \left(\frac{\alpha}{A} \mathbf{H} - \frac{\alpha^2}{A} \nabla p \right) = S_{\alpha,1p} \quad (3.3.28)$$

The fundamental steps of the solution algorithm follow the same logic of the two-phase approach, namely: 1) discretize the momentum equation to obtain \hat{M} , \mathbf{b}^* ; 2) solve the pressure equation 3.3.28; 3) update the velocity with the now available pressure gradient via 3.3.27. The actual iteration procedure, as for the two-phase algorithm, is clearly outlined in subsection 3.3.3. Note that, out of consistency with the two-phase approach, no velocity predictor is ever calculated (i.e. the momentum equation is never solved on its own).

3.3.1.5 Remarks on a cell-centered and face-centered momentum treatment

The developed algorithm employs a standard co-located variable treatment, meaning that all variables are stored at cell centers. Thus, during the update of the velocity at the pressure-correction step via equation 3.3.17 (or 3.3.20 for the standard OpenFOAM algorithm), pressure gradients need to be evaluated at cell-centers. However, this process is known to be sensitive to the employed interpolation schemes, as numerical artifacts such as field “checkerboarding” [59] can arise. Part of the reason for this lies in the fact that the most natural definition of a gradient over a mesh is at cell-faces, rather than at cell centers. This is one of the reasons why most early CFD approaches relied on *staggered* grids, wherein the velocity field, which is reconstructed from pressure gradients, was stored at cell faces, rather than cell centers. A staggered grid approach solves the issue of potential field staggering in its entirety, but has the drawback of a more cumbersome implementation and less flexibility when it comes to unstructured meshes, so co-located variable treatments generally prevailed in the CFD domain. To address this issue, a variant of a Rhie-Chow interpolation technique [60] is employed by OpenFOAM based algorithms, inclusive of the presently developed one.

However, given that multi-phase simulations prove to be further sensitive to the issue of field checkerboarding in certain occasions, Weller proposed a different approach for velocity reconstruction [61], which resembles a staggered grid approach in “spirit”.

The core idea is to reconstruct the velocity fields from superficial phase fluxes ϕ_i . In principle these fluxes represent the face-interpolated velocities dotted with the surface area vectors, namely $\phi_i = \mathbf{u}|_f \cdot \mathbf{S}_f$. By interpolating both sides of equation 3.3.17 and dotting them with \mathbf{S}_f we obtain an expression that relates ϕ_i to the *surface-normal* pressure gradient at the cell faces, which is available directly after the solution of the pressure equation with no additional cell-center interpolation operations. Weller thus proposes to reconstruct the cell-center velocity fields from the face-centered superficial volumetric fluxes ϕ_i , which are obtained from the face-interpolated, face-surface-area-dotted pressure-velocity relationship, namely equation 3.3.17 in our case. This reconstruction to cell-centers can be achieved via [19]:

$$\mathbf{u} = \left(\sum_f (\mathbf{n}_f \otimes \mathbf{S}_f) \right)^{-1} \cdot \left(\sum_f \mathbf{n}_f (\mathbf{u}|_f \cdot \mathbf{S}_f) \right) \quad (3.3.29)$$

with \mathbf{n}_f being the face normal vector. This approach is implemented in some multi-phase OpenFOAM solvers and is referred to as *face-momentum* approach, as it tries to mimic some of the advantages of a staggered grid approach. The face-momentum approach was found to reduce phase fraction checkerboarding effects in most circumstances where it would otherwise occur and enhance stability [61]. For this reason, a face-centered variant of the

algorithm has also been developed. It should however be noted that, as the superficial fluxes are obtained from a face-interpolated version of the pressure-velocity relationship, a certain degree of loss of accuracy is expected. For this reason, both a traditional and a face-momentum-based versions of the developed pressure-velocity coupling algorithm were investigated during the verification process, as discussed in section 3.4.

3.3.2 Solution of the enthalpy equation

Let us start by recalling the formulation presented in 3.1.3 for a two-phase flow:

$$\begin{aligned} \frac{\partial}{\partial t} (\alpha_i \rho_i h_i) + \nabla \cdot (\alpha_i \mathbf{u}_i \rho_i h_i) = \\ \nabla \cdot (\alpha_i \kappa_{\text{eff},i} \mathbf{T}_i \nabla T_i) + \alpha_i \frac{\partial}{\partial t} p + \alpha_i \rho_i \mathbf{u}_i \cdot \mathbf{g} + \\ + \alpha_i q_{int,i} - \Gamma_{i \rightarrow j} h_i - q_{i \rightarrow j} - q_{i \rightarrow s} \end{aligned} \quad (3.3.30)$$

with the phase transfer terms defined as in equations 3.2.26, 3.2.32,:;

$$q_{i \rightarrow j} \equiv q_{i \rightarrow \partial} = A_{\partial}''' H_{i\partial} (T_i - T_{\partial}) \quad (3.3.31)$$

$$q_{i \rightarrow s} = f_i A_s''' H_{is} (T_i - T_s) \quad (3.3.32)$$

After the solution of the continuity equation and pressure-velocity coupling for the prediction of the new phase fractions α_i , velocities \mathbf{u}_i , pressures p and densities ρ_i resulting from the application of a certain equation of state, an equation in the form of 3.3.30 can be solved on its own. However, while the solution variable consists of the enthalpy h_i , the fluid temperature T_i figures as well. While these terms can in principle be evaluated explicitly, an implicit formulation of these terms (especially for the diffusive term) would prove significantly more beneficial for the convergence properties of the linear system representing the enthalpy equation when solved via iterative means. For this reason, by recalling the definition of heat capacity $c_{p,i} = \partial h_i / \partial T_i$ and neglecting its spatial variations, the diffusive heat flux term is approximated as:

$$\alpha_i \kappa_{\text{eff},i} \nabla T_i \approx \alpha_i \frac{a_{\text{eff},i}}{\rho_i} \nabla h_i \quad (3.3.33)$$

in which $a_{\text{eff},i} \equiv \kappa_{\text{eff},i} / (\rho_i c_{p,i})$ is the effective thermal diffusivity of the fluid. This formulation allows for an implicit treatment of the diffusive term. A similar approach is used for the treatment of the heat source terms 3.3.31, 3.3.32.

A Taylor expansion of the enthalpy around the latest available fluid temperature T_0 yields:

$$\begin{aligned} h(p, T) \approx h(p, T_0) + c_p(p, T_0) (T - T_0) \Rightarrow \\ \Rightarrow T \approx T_0 + \frac{1}{c_p(p, T_0)} (h(p, T) - h(p, T_0)) \end{aligned} \quad (3.3.34)$$

By taking advantage of this result, the heat source terms can be implemented as:

$$q_{i \rightarrow \partial} = A''_{\partial} H_{i\partial} \left(T_i + \frac{1}{c_{p,i}} (h_i - h_i(p, T_i)) - T_{\partial} \right) \quad (3.3.35)$$

$$q_{i \rightarrow s} = f_{A''_s, i} A''_s H_{is} \left(T_i + \frac{1}{c_{p,i}} (h_i - h_i(p, T_i)) - T_s \right) \quad (3.3.36)$$

in which h_i is the solution variable for which the enthalpy equation is discretized, while $h_i(p, T_i)$ is the value of the specific enthalpy evaluated at T_i , namely the latest available enthalpy value. Given that these source terms figure with a negative sign on the RHS of the enthalpy equation and given that the coefficients that multiply h_i , namely $A''_{\partial} H_{i\partial}/c_{p,i}$ and $f_i A''_s H_{is}/c_{p,i}$ are always positive (or null at worst), this semi-implicit treatment will always beneficially contribute to the discretized enthalpy matrix diagonal, improving convergence properties.

On a last note, as seen for the momentum equation, the spurious heat source resulting from possible continuity errors $\xi_i h_i$ is accounted as well. The final formulation of the implemented enthalpy equation thus is:

$$\begin{aligned} \frac{\partial}{\partial t} (\alpha_i \rho_i h_i) + \nabla \cdot (\alpha_i \mathbf{u}_i \rho_i h_i) = \\ \nabla \cdot \left(\alpha_i \frac{a_{\text{eff},i}}{\rho_i} \nabla h_i \right) + \alpha_i \frac{\partial}{\partial t} p + \alpha_i \rho_i \mathbf{u}_i \cdot \mathbf{g} + \\ + \alpha_i q_{int,i} - \Gamma_{i \rightarrow j} h_i - q_{i \rightarrow \partial} - q_{i \rightarrow s} - \xi_i h_i \end{aligned} \quad (3.3.37)$$

3.3.3 Coupling algorithm

The overall solution strategy that was developed is presented in Figure 3.3.1. It is based on a merged PISO-SIMPLE [48][49] algorithm (PIMPLE).

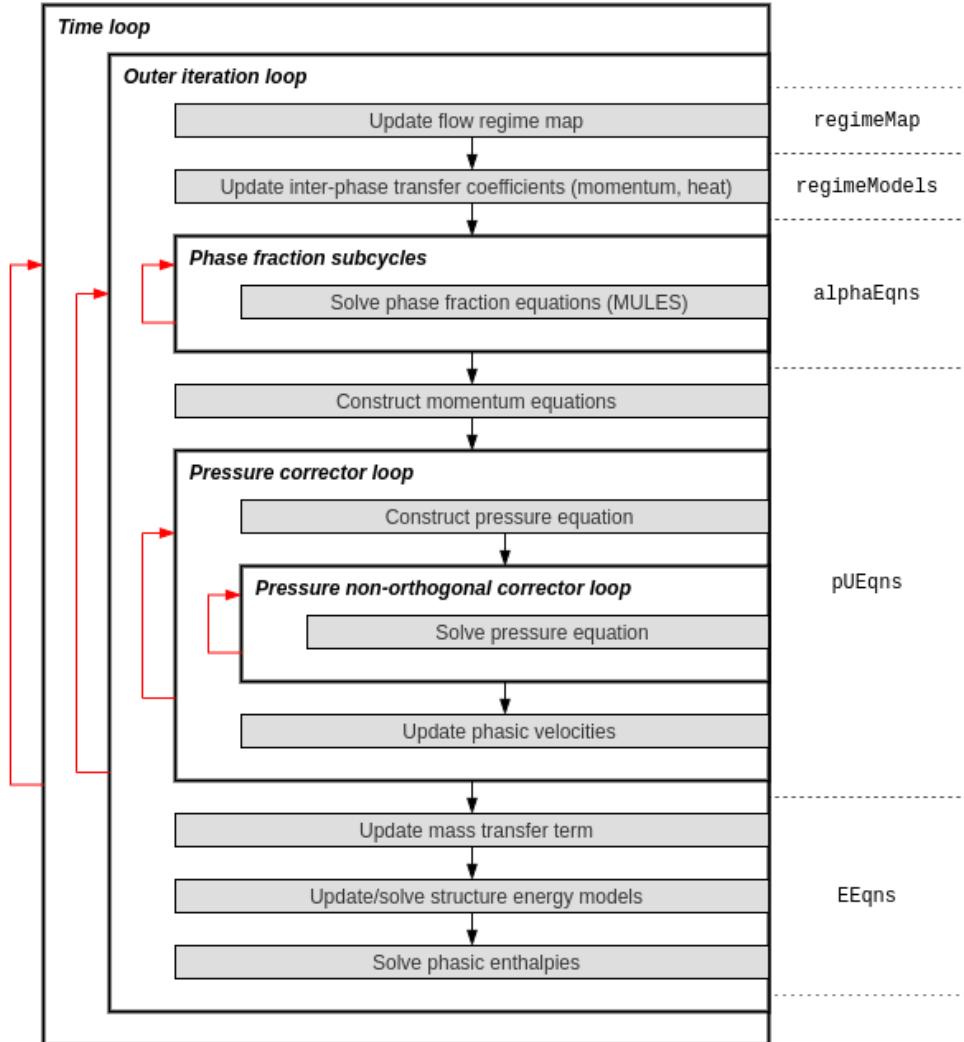


Figure 3.3.1: flowchart of the solution algorithm. Red arrows indicate that a section (outlined in bold black) consists of a loop (governed by certain conditions). On the rightmost side, the different flowchart steps are logically grouped into what we refer to as “components”. These were subject to scalability analyses, as discussed in section 3.6.

Within each time step, a certain number of outer iterations (logically equivalent to SIMPLE iterations, even though they are more generally referred to by the global algorithm name, i.e. PIMPLE iterations) can be performed to resolve the coupling between velocity, pressure and energy. In particular, each outer iteration consists of the following steps:

1. **flow regime map update.** In this context, each flow regime acts as a place-holder for different models used to compute the inter-phase transfer coefficients. At this step, the spatial extent of every regime is assessed based on the flow regime map;
2. **models update.** These consist of: fluid-fluid drag factor K_{ij} , fluid-

structure drag factors \mathbf{K}_{is} , fluid bulk-interface heat transfer coefficients $H_{i\partial}$, fluid-structure heat transfer coefficients H_{is} , fluid-fluid interfacial area density A''_∂ , fluid-structure contact fractions $f_{A''_s,i}$ and characteristic fluid dimensions $D_{h,i}$. This update is performed based on the updated spatial extent of the regimes;

3. **solve phase fractions.** the MULES algorithm is used to compute the new phase fractions α_i and volumetric fluxes $\phi_{\alpha,i}$ by solving equation 3.3.4. This is done starting from the existing face-centered superficial fluxes ϕ_i and the latest available evaluation of the phase fraction source terms $S_{\alpha,i}$. The algorithm can be iterated multiple times within each outer iteration to allow for *sub-cycling*. Sub-cycling consists in solving the phase fraction equation a number N of times over a time step N times smaller than the global time step, which allows to ease the time step limitations imposed by the CFL condition;
4. **construct momentum equations.** This consists in obtaining the coefficient matrices \hat{M}_i and source terms \mathbf{b}_i in 3.3.11. The matrix diagonal coefficients A_i and the modified coefficients A_i^{**} obtained via 3.3.16 as used in equations 3.3.17, 3.3.18 are thus obtained as well;
5. **pressure corrector loop.** This is logically equivalent to a PISO loop and is performed a number of times that can be either user-selected or determined at run time by the convergence properties of the solution. It fundamentally serves the purpose of solving the pressure equation by iterating on the non-linearities of the explicitly evaluated off-diagonal contributions and source terms \mathbf{H}_i . Please note that the modified diagonal coefficients A_i^{**} are never updated in this loop as the momentum equations are discretized once per outer iteration (i.e. the non-linearities of the diagonal coefficients are resolved by the outer iteration loop);
 - 5.1. **construct pressure equation** This consists in assembling the pressure equation as of 3.3.18. It requires the evaluation of \mathbf{H}_i via equations 3.3.15 for each phase (to be evaluated explicitly with the latest available phase velocities). This also requires the evaluation of the phase fraction source terms $S_{\alpha,i}$ computed as of 3.3.3;
 - 5.2. **pressure non-orthogonal corrector loop.** This is performed to resolve the effects of possible mesh non-orthogonality, i.e. the degree to which, for each mesh face, the face normal is not parallel to the segment connecting the cell centers of the cells that share the face. This was extensively covered by Jasak [21] and is not discussed here. The number of non-orthogonal iterations is user-selected;
 - 5.2.1. **solve pressure equation;**
 - 5.3. **update phase velocities.** Once the pressure is computed and the pressure gradient available, use equations 3.3.17 to calculate the new phase velocities \mathbf{u}_i and superficial fluxes ϕ_i ;

6. **update mass transfer term.** The new mass transfer term $\Gamma_{i \rightarrow j}$ is calculated via 3.2.36, which requires the explicit estimation of the interfacial heat fluxes $q_{i \rightarrow \partial}$ as well as the adjusted latent heat L_{ij}^{adj} ;
7. **update structure energy models.** The models used to describe the structure enthalpy are updated/solved as discussed in sub-section 3.2.4 to yield the new structure surface temperature T_s ;
8. **solve phase enthalpies.** Construct and discretize the enthalpy equations 3.3.37 and solve them to update the fluid enthalpies h_i . Update the fluid thermo-physical models to update fluid density, viscosity, heat capacity, thermal conductivity, etc.;

3.4 Algorithm verification

While the verification should in principle cover both fluid-dynamics and energy, the thermal part of the solver relies on well established OpenFOAM matrix assembly and solution operations. Thus, the verification covers only the fluid-dynamics aspect of the solver, with particular regard to the developed pressure-velocity algorithm.

The chosen verification approach consists in the Method of Manufactured Solutions (MMS), which has enjoyed some utilization in the field of multi-phase solvers [62] and is considered to be among the most flexible verification approaches. A detailed explanation of the MMS is out of scope of the present work, as it is a standard, albeit complex verification approach that has been extensively covered in other works [63]. Nonetheless, the general idea behind such approach is introduced in subsection 3.4.1 while the results of the verification effort are summarized in subsection 3.4.2 for a variety of cases.

3.4.1 The Method of Manufactured Solutions

Let us consider a computer code implementation that solves a problem for an unknown x governed by an equation represented in an operator form as $\hat{M}x = b$. The MMS can be summarized as follows:

1. select/construct an analytical target solution x_0 that the code is supposed to approximate;
2. derive the analytical source term s that satisfies $s = \hat{M}x_0 - b$, which is possible since the analytical form of \hat{M} is known;
3. implement the analytical source term s in the code;
4. impose a set of boundary conditions that reflect the behaviour of x_0 at the domain boundaries;

5. compare the solution x provided by the code with the target solution x_0 and draw conclusions based on a set of verification acceptance or rejection criteria.

The choice of x_0 is in principle arbitrary and not bound by the physical properties that the equation is supposed to model. Nonetheless, following the purpose of a verification effort, any x_0 should satisfy a certain number of properties, most importantly: 1) be composed of smooth analytic functions to ensure that the theoretical order-of-accuracy can be attained (which ties into verification criteria discussed later); 2) be general enough and have a non-trivial number of derivatives so to exercise all terms of the governing equation; 3) do not compromise code robustness by predicting values outside the intended solution range (e.g. unbounded phase fraction values). Furthermore, since boundary conditions are an integral part of the solution procedure, the verification should be repeated for different sets of boundary conditions.

With regards to verification acceptance criteria, the one employed in this work consists in the assessment of the *order-of-accuracy* of the numerical solution. This translates into evaluating the solution with increasingly refined meshes, and to assess the order-of-accuracy q , which quantifies how numerical errors ϵ scale as a function of a characteristic mesh dimension h , so that $\epsilon = O(h^q)$. The verification criterion thus consists in assessing whether or not the observed order-of-accuracy reflects the theoretical order-of-accuracy, which depends on the employed set of equation discretization and interpolation schemes. In particular, the errors considered in this work consist of the L^2 norm error:

$$\epsilon_{L^2} = \sqrt{\int_{\Omega} (x - x_0)^2 d\Omega} \quad (3.4.1)$$

and the L^∞ norm error:

$$\epsilon_{L^\infty} = \sup\{|x - x_0|\} \quad (3.4.2)$$

3.4.2 Results

The investigated domain consists of a 2-D square domain of side $L = 0.1\text{ m}$ with a uniform square meshing. The following expressions have been selected

for phase fractions α_i , phase velocities \mathbf{u}_i and pressure p :

$$\begin{aligned}
\alpha_s &= x + y; \\
\alpha_1 &= 0.5 + 2.5x \cos(\pi(10x - 4t)) \\
\alpha_2 &= 1.0 - \alpha_1 - \alpha_s \\
u_{1,x} &= 0.25 \sin^2(10\pi y) \\
u_{1,y} &= 0.1 \sin^2(10\pi y) \sin^2(10\pi x) \\
u_{2,x} &= u_{1,x} \\
u_{2,y} &= -u_{1,y} \\
p &= 1 \cdot 10^5 + (x - 0.1) \cdot 10^4
\end{aligned} \tag{3.4.3}$$

This choice exercises all the terms of the governing equations, including all derivative terms.

With regards to boundary conditions, a number of combinations were investigated. The results presented in this section were obtained via Neumann boundary conditions for pressure at $y = 0$, $y = L$, $x = 0$ and for $\alpha_{1,2}$, $\mathbf{u}_{1,2}$ at $y = 0$, $y = L$, $x = L$, while Dirichlet boundary conditions were imposed for pressure at $x = L$ and for $\alpha_{1,2}$, $\mathbf{u}_{1,2}$ at $x = 0$. Results comparable to the ones that are presented were obtained for other combinations of boundary conditions.

Fluid thermo-physical properties and inter-phase coupling coefficients were set so that the contributions of all terms in the momentum equations are comparable in magnitude. The resulting analytical source terms for the phase fraction equations, momentum equations and the pressure equations are not reported for conciseness, given the non-trivial amount of derivative operators to be evaluated.

Time-dependent simulations were run from 0 s to 6 s and the L^2 and L^∞ norms for all the involved fields were averaged over said time duration.

Progressively refined meshes from 16 to 128 cells per side were investigated. In particular, the performance of the cell-centered variant versus the face-centered variant of the developed algorithm were compared.

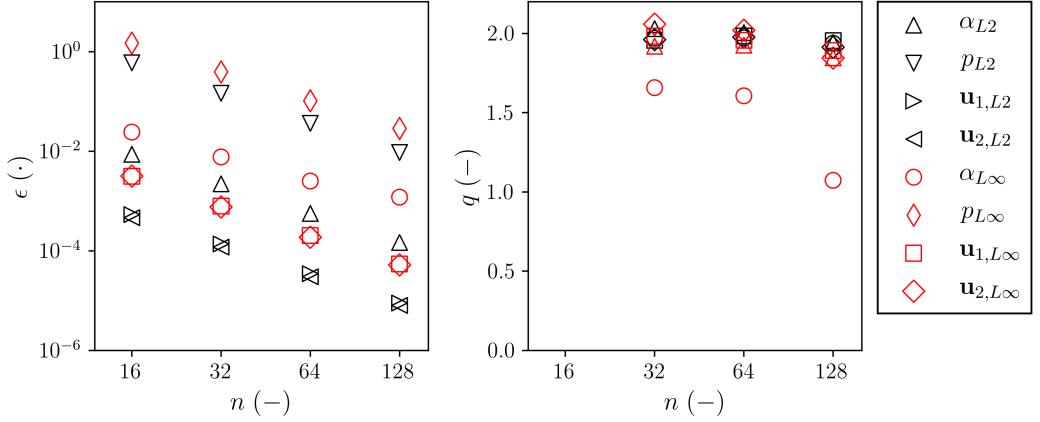


Figure 3.4.1: errors ϵ (left) and associated order-of-accuracy q (right) computed by the L_2 (black) and L_∞ (red) norms for the various quantities of interest with the cell-centered algorithm. Errors are absolute and dimensional, namely: α (–), p (Pa), $\mathbf{u}_1, \mathbf{u}_2$ ($\frac{m}{s}$)

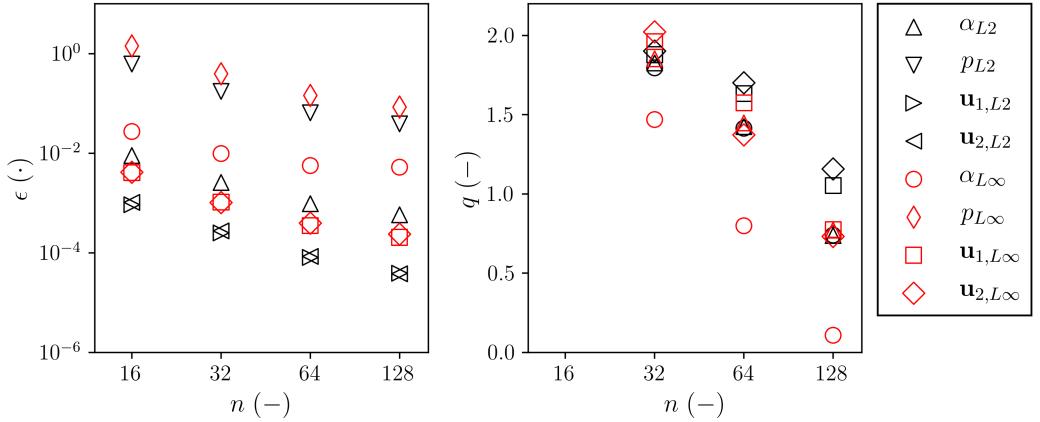


Figure 3.4.2: errors ϵ (left) and associated order-of-accuracy q (right) computed by the L_2 (black) and L_∞ (red) norms for the various quantities of interest with the face-centered algorithm. Errors are absolute and dimensional, namely: α (–), p (Pa), $\mathbf{u}_1, \mathbf{u}_2$ ($\frac{m}{s}$)

Since a linear scheme was employed for the discretization of derivatives and for variable face-interpolation in the momentum and pressure equations, the theoretical order of accuracy for pressure and velocity should be 2. However, this is expected to be affected by the coupled nature of the equations with other equations discretized with lower schemes. In particular, the theoretical order of accuracy for the phase fraction equations cannot be exactly determined due to the iterative nature of the flux correction technique employed by the MULES algorithm. Nonetheless, the order of accuracy should lie between 1 due to the upwind first order scheme used for the bounded fluxes estimation, and 2 due to the scheme by Van Leer [64] used for the high-order correction fluxes.

Based on these considerations, the verification of the cell-centered version of the algorithm is deemed satisfactory, also in the light of the small magni-

tudes of the errors. This holds true for both the L^2 and L^∞ norms. However, the same does not appear to hold for the face-momentum algorithm, as the order-of-accuracy deteriorates for fine meshes. This is consistent with a larger diffusive contribution of the interpolation of velocity coefficients and off-diagonal velocity source terms at cell faces.

As a matter of fact, the diffusivity of the face-momentum algorithm is exacerbated by having set fluid thermo-physical properties so to result in comparable contributions from the different terms of the momentum equation. This results in a rather large choice for the viscosity of the fluids of $\mu_{1,2} \simeq 0.1 \text{ Pa s}$, in the range of e.g. motor oils. A new set of less diffusive simulations was then performed with $\mu_{1,2} \simeq 1 \cdot 10^{-3} \text{ Pa s}$, more compatible with the applications envisioned for the algorithm presented here. The results are presented in Figure 3.4.3 and show that the order of accuracy significantly improves for the face-momentum algorithm. It should also be noticed that in all cases, degradation of the order of accuracy happens for error magnitudes below practical relevance.

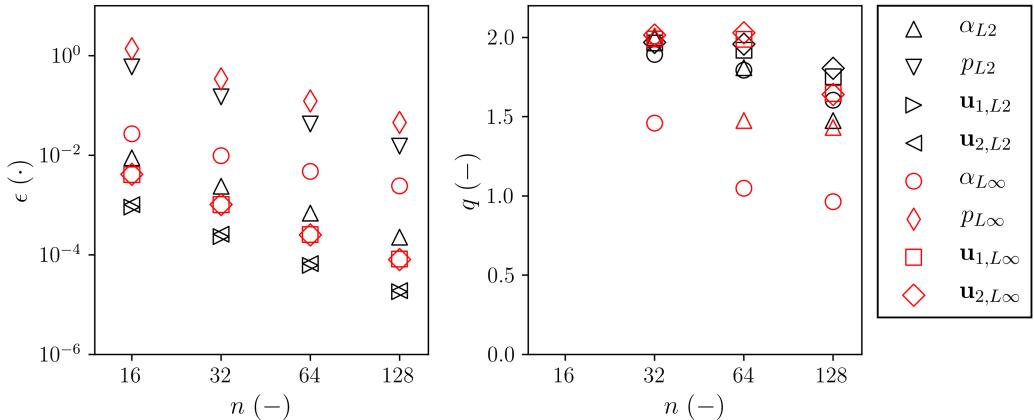


Figure 3.4.3: errors ϵ (left) and associated order-of-accuracy q (right) computed by the L_2 (black) and L_∞ norms for the various quantities of interest with the face-centered algorithm for a lower fluid viscosity. Errors are absolute and dimensional, namely: α ($-$), p (Pa), $\mathbf{u}_1, \mathbf{u}_2$ ($\frac{\text{m}}{\text{s}}$)

3.5 Pressure-velocity coupling algorithm performance

To assess the performance of the developed pressure-velocity coupling algorithm, the existing standard OpenFOAM algorithm has been used as a reference. As a first test case, the 1-D motion of a dispersed fluid via advection by a carrier fluid through a porous structure was investigated.

The initial and boundary conditions for the variables of interest in the

domain of length $L = 2\text{ m}$ are:

$$\alpha_s(x, 0) = \begin{cases} 0 & x < 0.5 \vee x > 0.5 \\ 0.5 & 0.5 \leq x \leq 0.5 \end{cases}$$

$$\alpha_1(x, 0) = \begin{cases} 1 & x < 0.5 \vee x > 0.5 \\ 0.25 & 0.5 \leq x \leq 0.5 \end{cases}$$

$$\alpha_2(x, 0) = \begin{cases} 0 & x < 0.5 \vee x > 0.5 \\ 0.25 & 0.5 \leq x \leq 0.5 \end{cases}$$

$$\mathbf{u}_1(x, 0) = 0.5 \frac{m}{s} \quad (3.5.1)$$

$$\mathbf{u}_2(x, 0) = 0 \frac{m}{s} \quad (3.5.2)$$

$$\mathbf{u}_1(0, t) = 0.5 \frac{m}{s} \quad (3.5.3)$$

$$p(2, t) = 1 \cdot 10^5 \text{ Pa} \quad (3.5.4)$$

with all other variables having a null gradient at the boundaries.

This simplified test allows to establish the general trends to be expected as certain system properties are varied. Results are reported in Figure 3.5.1. The left plots reports the last initial pressure residual at the last outer iteration for each time step, which is used as a measure of how well the pressure-velocity coupling is resolved. The right plots represents the total number of pressure linear solver iterations performed within the time-step. It is recalled that, as velocity is reconstructed from pressure, the pressure equation residual is the only indicator of convergence of the pressure-velocity coupling. All simulations were performed with 3 outer iterations per time step and 3 pressure correctors per outer iteration. The Generalized Algebraic Multi-Grid (GAMG) linear solver was used for the cases reported here, yet the same trends were observed with the Preconditioned Bi-conjugate Gradient Stabilized (PBiCGStab) linear solver. The same linear solver convergence criteria were used in both cases.

As the developed algorithm is based on Partial Elimination ideas, which are designed to improve performance in scenarios involving large inter-phase drag coefficients, the following cases were investigated. The only force acting on the fluids (excluding pressure gradients) consists of an inter-phase drag modelled as $K_{12} = 1 \cdot 10^4 \frac{\text{kg}}{\text{m}^3\text{s}}$ and $K_{12} = 1 \cdot 10^5 \frac{\text{kg}}{\text{m}^3\text{s}}$ for the first and second row in Figure 3.5.1, respectively. Since the developed algorithm alters both the pressure equation coefficients and source terms, the last row investigates the effect of larger source terms due to e.g. fluid-structure drag, treated explicitly for this test. In particular, a diagonal fluid-structure drag factor with magnitude $|\mathbf{K}_{1s}| = 5 \cdot 10^3 \frac{\text{kg}}{\text{m}^3\text{s}}$ to be comparable in magnitude with the fluid-fluid drag was used.

Overall, the developed algorithm appears to perform consistently better in these scenarios in terms of pressure equation residuals and of number of necessary iterations when compared to the standard OpenFOAM algorithm

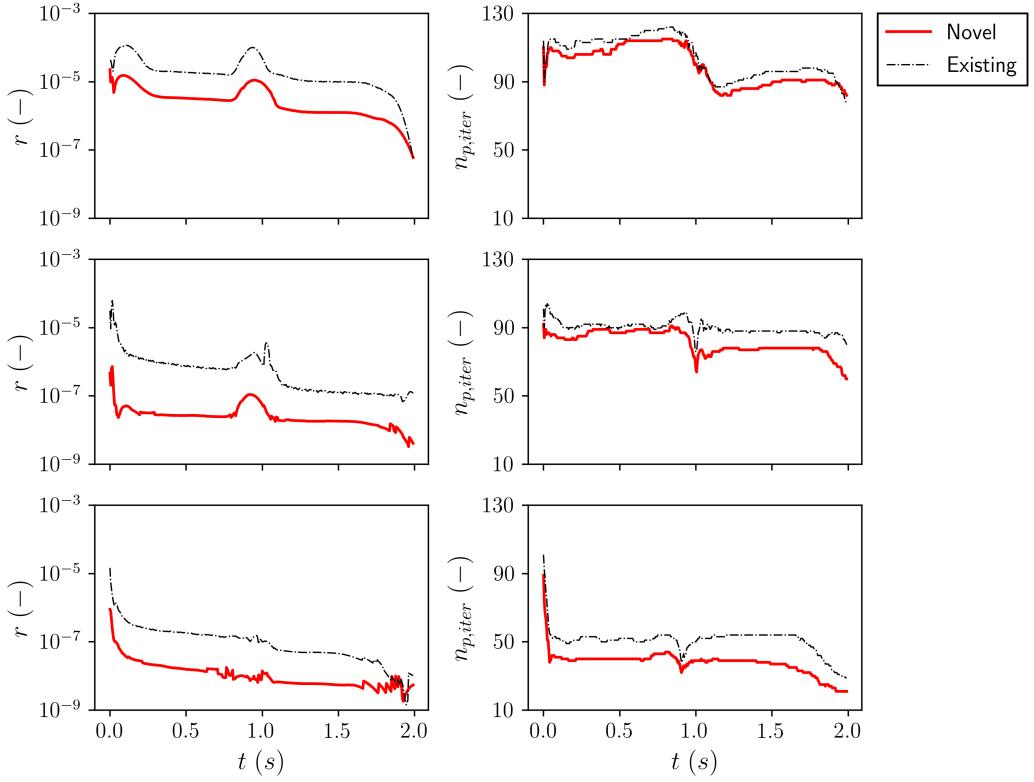


Figure 3.5.1: evolution of pressure initial residuals r at the last outer iteration and last pressure corrector (left column) and number of pressure iterations $n_{p,iter}$ (right column) within each simulation time step. These are compared between the novel and existing PEA implementations for three different cases: small drag factor $K_{12} = 1 \cdot 10^4 \frac{kg}{m^3 s}$ with no other forces acting on the fluids (top row); large drag factor $K_{12} = 1 \cdot 10^5 \frac{kg}{m^3 s}$ with no other forces acting on the fluids (middle row); large drag factor with the addition of explicit virtual mass forces due to a virtual mass coefficient of 0.5 and large fluid-structure drag $K_{1s} = 5 \cdot 10^3 \frac{kg}{m^3 s}$ (bottom row).

(which is also based on Partial Elimination, yet implemented differently, as thoroughly discussed in sub-sub-section 3.3.1.3).

The performance of the algorithm has then been assessed for two more realistic scenarios. The first consists in a 1-D liquid sodium boiling transient in a bundle of electrically heated pins. The modelling details and correlation choices are specific to liquid sodium are the same as those used for the modelling of validation experiments that covered in chapter 4 (with the difference that the model used for these tests is 1-D rather than 2-D). Results are presented in Figure 3.5.2 for both the developed and the standard OpenFOAM pressure-velocity coupling algorithms. The start and end of the boiling can be inferred from the increase in pressure equation residuals and total number of pressure equation linear solver iterations in time.

The overall results observed before still apply, with a significant reduction in both residuals and linear solver iterations. For this particular case, a $\sim 5\%$ reduction in simulation time was observed as a consequence of the reduced number of pressure equation iterations.

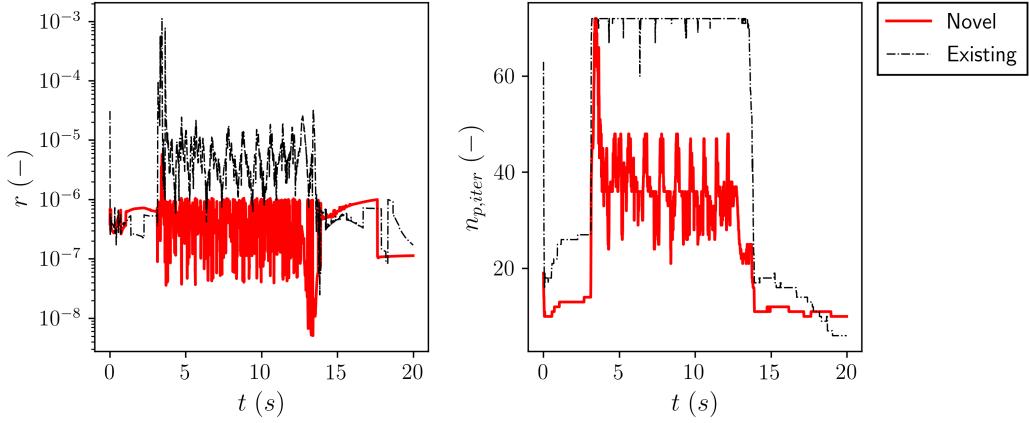


Figure 3.5.2: evolution of pressure initial residuals r at the last outer iteration and last pressure corrector (left) and number of pressure iterations $n_{p,iter}$ (right) within each simulation time step for a 1-D sodium boiling transient.

The second realistic scenario consists in a modified version of a standard 2-D OpenFOAM test case `bubbleColumn`, consisting of air bubbles injected at the bottom of a column of stationary water. Results for the two algorithms are presented in Figure 3.5.3 for the first $2s$ of simulation. Due to the considerably lower magnitude of the coupling between the phases, partly due to the lower slip velocity, there are no noticeable differences between the two algorithms. This case serves as an example to show that in those scenarios where interphase drag does not play a significant role between the phases, the developed algorithm performs similarly to the standard OpenFOAM one.

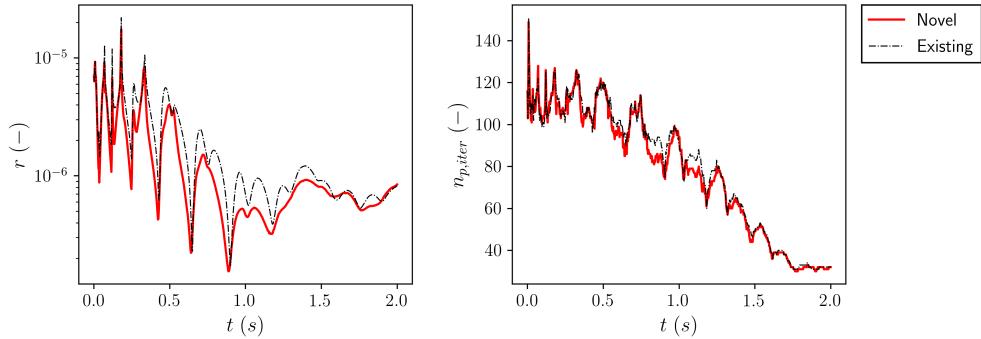


Figure 3.5.3: evolution of pressure initial residuals r at the last outer iteration and last pressure corrector (left) and number of pressure iterations $n_{p,iter}$ (right) within each simulation time step for a modified 2-D `bubbleColumn` OpenFOAM test case.

3.6 Parallel performance

While it is true that one of the primary goals of a coarse-mesh approach is to reduce computational burdens, the modelling of large engineering systems can still result, depending on the degree of detail, in several tens of millions of mesh cells.

For parallelization, the code employs Message Passing Interface (MPI)-based domain decomposition [65] enabled by the OpenFOAM framework. The scaling performance provided by this framework depends on the equations that are solved, and on the steps that are taken to solve them. This is why scaling performances have been investigated. In particular, the strong scaling characteristics of the solver and of the different algorithm components (with reference to Figure 3.3.1) were assessed. As a reminder, strong scaling consists in evaluating the performance of a code for a fixed problem (i.e. mesh) size while varying the number of computer cores used for the parallel calculation.

A test case that could stress all of the different components of the solver was devised. It consists of a 3-D cubic domain with one face acting as an inlet and the opposite face as an outlet. Two structures modelled as isotropic porous media are defined in the two cube halves contacting the inlet and outlet. These are referred to as “upwind” and “downwind” structures respectively. The surface temperatures of both structures are set to constant values so that the upwind half is above a prescribed fluid saturation temperature and the downwind one below said temperature. The thermo-physical properties and correlations that are chosen for modelling drag and heat transfer were that of liquid sodium employed for boiling investigations discussed in chapter 4. A fictitious regime map consisting of five identical regimes (i.e. prescribing the same correlations for drag and heat transfer) depending on the vapour phase volume fraction is provided to test the intensiveness of the cell-by-cell assessment of the flow regime map. This test case was overall devised and adjusted so to lead to a numerically steady-state boiling scenario.

Strong scaling performance was assessed with a mesh of $128^3 \simeq 2 \cdot 10^6$ cells decomposed on a varying number of cores, ranging from 32 to 1024. A further set of simulations on a mesh of $256^3 \simeq 16 \cdot 10^6$ up to 4096 cores was also performed. The scalability results were in line with those established on the smaller mesh, so that only the former are presented. Simulations were performed on a Cray XC-50 supercomputer operated by the Swiss National Supercomputing Centre (CSCS). The results are summarized in Figure 3.6.1 and report the total run time as well as the run time of each code section, referred to as "component" as reported in Figure 3.3.1, as the number of threads is varied. For a better understanding of the trends, the parallel efficiency of the algorithm is also reported. In this context, the parallel efficiency for a given case i is a measure of the simulation speed-up with respect to a case j performed on a different number of threads:

$$e_{P,i} = \frac{t_i}{t_j} \frac{N_{T,i}}{N_{T,j}} \quad (3.6.1)$$

with t being the total simulation run time, N_T the number of threads used for the simulation. Parallel efficiency should ideally be of the order of 100% or greater, meaning that e.g. a doubling of the number of threads results in halving of the total run time.

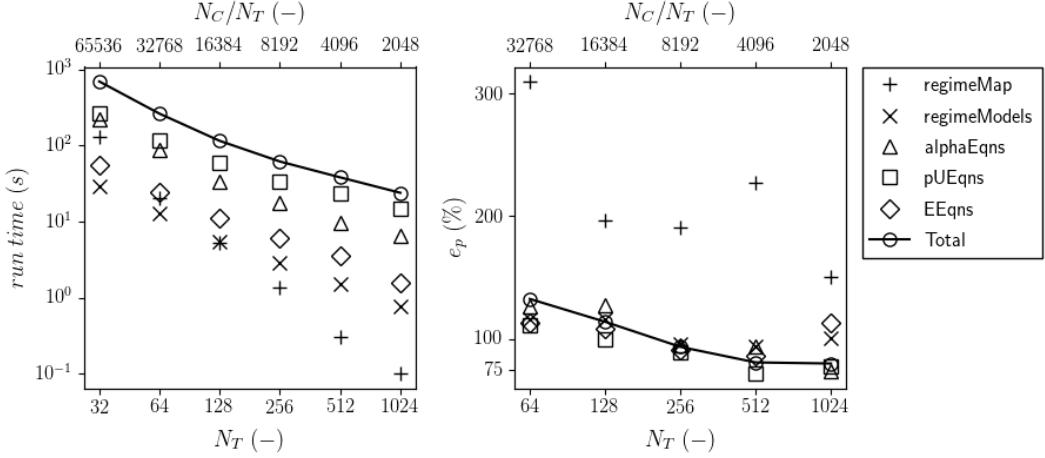


Figure 3.6.1: strong scaling results for a mesh of 128^3 cells. These consist of the component run times (right) and parallel efficiencies (left) in relationship to the amount of threads N_T used for the simulation. Additional axes in terms of cells per thread $\frac{N_C}{N_T}$ are provided for ease of discussion. By "component" we mean a specific block of code with reference to what is defined in Figure 3.3.1.

It can be observed that the algorithm as a whole as well as its individual components scale well up to 8192 cells per core, after which the parallel efficiency degrades, especially for the pressure-velocity coupling and MULES algorithms. This is expected for OpenFOAM-based solvers, which are known to scale poorly below $\simeq 10000$ cells per thread.

When it comes to the impact of individual solver components, a clear trend emerges. The most computationally demanding components are, in descending order, the pressure-velocity coupling algorithm, the MULES algorithm, the energy equation solution, and the update of momentum and heat transfer coefficients based on the regime map. However, the update of the regime map itself scales significantly more than linearly, as observed by the values of the parallel efficiency above unity. This is attributed to the fact that the regime map update relies on a considerable amount of operations to be performed on a cell-by-cell basis. This is suspected to cause a relatively large number of cache misses, which decrease as the amount of cells per thread (and thus cache requirements) is reduced. This type of "superlinear" scaling was similarly observed in some parallel applications in other computational domains and found to be partly attributable to cache-misses [66]. Nonetheless, even for more substantial counts of cells per thread, the overall cost associated with the regime map update is acceptable.

Chapter 4

Validation against sodium boiling experiments

4.1 Introduction

The present chapter covers the application of the code and algorithms presented in the previous chapters for the simulation of a number of sodium boiling scenarios, and their validation against experimental data.

Section 4.2 discusses the simulation and comparison against experimental results of quasi-steady-state boiling experiments performed at the Joint Research Centre (JRC) in Ispra, Italy [67] in heated tubular test sections. Because of the quasi-steady-state nature of these experiments, the outcome of calculation results is mostly dependent on just a handful of physical models, namely two-phase pressure drop multipliers, latent heat, and saturation temperature models. In section 4.3, the results obtained by the code are compared against a transient boiling experiment carried out at the KNS experimental facility at the Kernforschungszentrum Karlsruhe [68], which allows for a more thorough code validation. The comparison also allowed to perform a preliminary qualitative sensitivity study of the calculation results against a variety of both geometric and physical modelling choices. Conclusions regarding the results presented in this paper, along side future perspective work directions are presented in section 4.4.

4.2 Steady state validation

The present section discusses the validation efforts performed against quasi-steady-state boiling scenarios. In particular, the experiments under consideration are those performed at the Joint Research Centre in Ispra, Italy, in the 1980s, concerning sodium boiling [67]. These were carried in the wider then-existing European framework of Sodium-cooled Fast Reactor (SFR) development (a notable example being the German SNR-300 SFR), yet also to further knowledge concerning a fluid, sodium, whose use as a coolant was

enjoying increasing popularity outside of the nuclear domain.

4.2.1 Description of the experiments

The experimental setup at JRC, Ispra consisted of a closed loop with interchangeable heated test sections so that boiling in different flow geometries could be investigated. The overall description of the test loop can be found in [67]. A number of different test sections were experimented with, ranging from tubular and annular ones, to hexagonal wire-wrapped pin bundles.

With regards to the set of experiments that were used for this validation stage, these consist of quasi-steady-state boiling in the tubular test-sections. These were performed for a number power levels, and were used to assess the dependence of velocity on the total two-phase pressure drops. These tests are referred to as quasi-steady-state due to the fact that while boiling is an inherently transient phenomenon, the experimental conditions were such to allow a meaningful collection of *time-averaged* data.

These experiments represents a great opportunity for validation purposes as most two-phase multiplier models depend exclusively on the liquid and fluid vapour flow qualities, which, physically speaking, depend almost exclusively on the thermo-physical properties of the fluid, on the liquid inlet properties, and on the input power.

The geometry of the test section is illustrated in Figure 4.2.1. In particular, it consisted of a tube of 6 mm in inner diameter and 29 mm in outer diameter. Different input power levels resulting in (inner) wall heat flux densities ranging from 106 W/cm² to 250 W/cm² were investigated. For each power level, different inlet mass flow rates were prescribed by adjusting the inlet pressure, and, for each velocity, the overall pressure drop over the test assembly was measured. Data points span both one-phase and two-phase flow regimes.

4.2.2 Computational modelling

The system was represented as a 1-D model spanning the entire geometry as reported in Fig. 4.2.1. With regards to the condenser, it was modelled via a tapered pyramid of progressively larger volumes towards the outlet, yet its meshing is still 1-D, in the axial direction. Boundary conditions for each case consisted of fixed outlet pressure, fixed liquid inlet temperature and velocity, and inlet pressure gradient prescribed to be consistent with the inlet velocity (i.e. `fixedFluxPressure` in OpenFOAM). All other quantities were prescribed to have a null gradient at system boundaries. The modelling of the condenser is necessary as, if the model stopped short at the tube outlet (where two-phase flow exists), it would be inaccurate to prescribe a fixed pressure at such location. With regards to measured quantities, the overall pressure drop (exclusive of the hydrostatic component) between points *P11* and *P14* in Figure 4.2.1 was monitored.

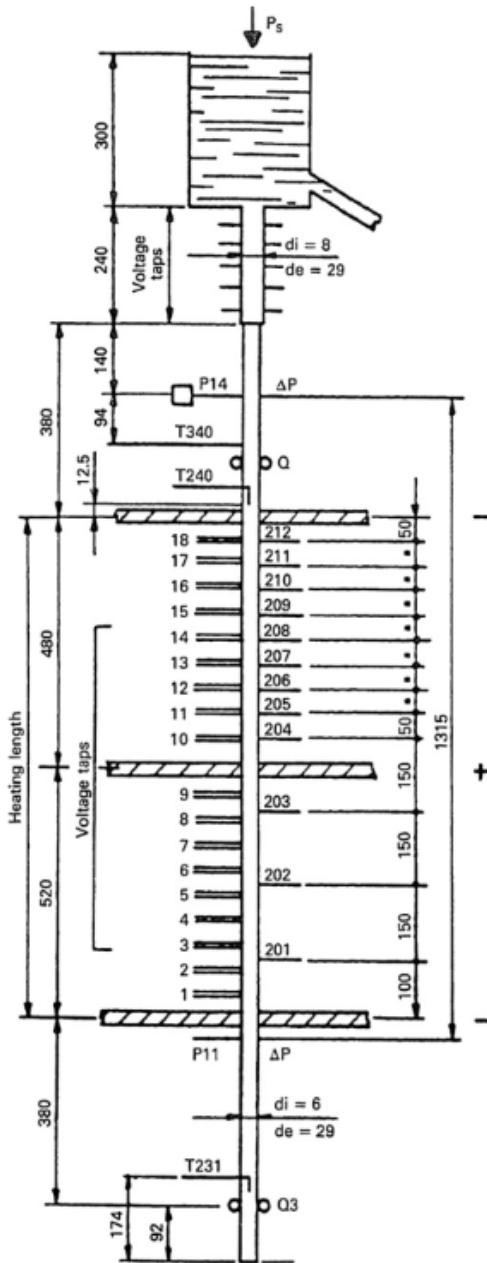


Figure 4.2.1: details of the tubular test section of the JRC Ispra experimental loop.

As mentioned earlier, the advantage of a quasi-steady-state scenario is that the final results are expected to be independent of a number of parameters, as the final vapour quality would depend exclusively on an enthalpy balance. As a result, heat transfer coefficients and inter-phase drag coefficients play little role in shaping the final results. The focus is thus shifted on few important parameters such as: liquid-structure drag coefficient $f_{d,ls}$; two-phase drag multiplier for the liquid phase ϕ_l^2 ; and the thermo-physical properties of the fluid, inclusive of its pressure-temperature saturation curve and the latent heat of

evaporation $L_{l \rightarrow v}$. The subscripts l and v will be used to refer to the liquid and vapour phase respectively, unless discussing general models, for which i and j will be used.

With regards to the liquid-structure drag coefficient $f_{d,ls}$, due to the tubular geometry of the system, the Blasius correlation was used:

$$f_{d,ls} = \frac{0.316}{Re_{ls}^{0.25}}$$

For the liquid two-phase pressure drop multiplier ϕ_l^2 , the following five models were investigated:

- Lockhart-Martinelli [69], a well-known correlation which relates the two-phase multiplier of the i -th phase ϕ_i^2 to a dimensionless parameters X_i^2 , called the Lockhart-Martinelli parameter, defined as:

$$X_i^2 = \left(\frac{\mu_i}{\mu_j} \right)^{0.2} \left(\frac{1 - \dot{x}_j}{\dot{x}_j} \right)^{1.8} \frac{\rho_j}{\rho_i} \quad (4.2.1)$$

and:

$$\phi_i^2 = 1 + \frac{C}{X_i} + \frac{1}{X_i^2}$$

with C being a parameter that ranges between 5 and 20 depending on the postulated flow regime of each phase (whether turbulent or laminar, allowing for 4 possible combinations). Generally $C = 20$ is used for most applications.

- Lottes-Flinn, which predicts:

$$\phi_i^2 = \frac{1}{(1 - \alpha_j^N)^2}$$

This was proposed for small values of the gas phase flow quality. With specific regards to sodium boiling however, Nguyen proposed the following relationship between α_j^N and the Lockhart-Martinelli parameter of the i -th phase to extend it to high-phase fraction boiling flows [70]:

$$\alpha_j^N = (1 + X_i^{0.8})^{-0.378}$$

- Kaiser [71], a correlation developed for sodium-boiling in pin bundle geometries that relies on the Lockhart-Martinelli parameter. This was developed for specific application to the liquid phase, so the l subscript is used in place of i :

$$\ln \phi_l = 1.48 - 1.05 \ln \sqrt{X_l} + 0.09 \left(\ln \sqrt{X_l} \right)^2$$

- Kottowski-Savatteri [67], a correlation developed for Sodium boiling in pin bundle geometries that relies on the Lockhart-Martinelli parameter. Similarly to the work by Kaiser et al., this was developed for specific application to the liquid phase l :

$$\log_{10} \phi_l = 0.6252 - 0.5098 \log_{10} X_l + 0.1046 (\log_{10} X_l)^2$$

- Chenk-Kalish [72], a correlation developed for potassium boiling that relies on the Lockhart-Martinelli parameter, developed for the liquid phase l :

$$\ln \phi_l = 1.59 - 0.518 \ln X_l + 0.0867 (\ln X_l)^2$$

With regards to the thermophysical properties of the liquid and vapour sodium, the proposed data compiled by Fink and Leibowitz [73] was used for density, heat capacity, molecular viscosity, latent heat and the pressure-temperature saturation curve. In particular, the latent heat of evaporation is evaluated as:

$$L_{l \rightarrow v}(p) = 393370 \left(1 - \frac{T_{sat}(p)}{T_c}\right) + 4398600 \left(1 - \frac{T_{sat}(p)}{T_c}\right)^{0.29302}$$

with $Q_{L0, ij}$ in J/kg , temperatures in K , $T_c = 2503.7K$ the critical temperature of sodium and T_{sat} its saturation temperature at the current system pressure. The pressure-temperature saturation curve was modelled as [73]:

$$\ln p = 11.9463 - \frac{12633.7}{T} - 0.4672 \ln T$$

with T_{sat} in K , p in MPa .

4.2.3 Results

The simulations involved the reproduction of the experiments performed at four different input power levels, resulting in wall heat fluxes of $138 W/cm^2$, $159 W/cm^2$, $175 W/cm^2$, $250W/cm^2$, each with varying levels of inlet liquid flow velocities. Each set of simulations performed at a different power level will be referred to as “case”.

The results are summarized in Figure 4.2.2. For all cases, the flow at the highest simulated inlet velocity results in a one-phase flow, and thus the overall pressure drop decreases as the velocity is reduced. However, below a certain threshold that depends on the input power level, the liquid velocity is insufficient to prevent boiling, and the pressure drop increases due to the two-phase flow circumstances. In particular, the lower the velocity, the higher the vapour quality. Physically though, the pressure drop is due to reach a maximum as the pressure-velocity curve approaches that of a one-phase vapour flow. This can be clearly observed in the cases with wall heat fluxes of $158 W/cm^2$ and $175 W/cm^2$, as they were run also for particularly small inlet velocities.

With regards to the actual comparison between the experimental and the calculated results, a good agreement can be observed, with the two-phase pressure drop from Kottowski and Savatteri most accurately reproducing the results. This is not surprising, as this correlation was developed specifically starting from the data of these experiments [67]. The second best performing correlation is the one developed by Kaiser, also based on sodium boiling experiments in tubular and pin bundle geometries. The correlations by Lockhart and Martinelli, Lottes and Flinn also prove to yield very comparable results. With regards to the latter, however, the Nguyen correlation for the vapour phase fraction was used. It was in fact observed that using the original correlation by Lottes and Flinn tends to overestimate the two-phase pressure drop at high vapour volume fractions, while yielding results in line with the other correlations at lower phase fractions. The Nguyen correlation was devised to adjust the Lottes-Flinn model specifically for sodium boiling scenarios. Similar findings were observed by Ninokata with the SABENA code [11]. On the other end of the spectrum, the correlation by Chen and Kalish was developed for experimental data regarding potassium alloys, and proves to overestimate the overall pressure drop in all circumstances.

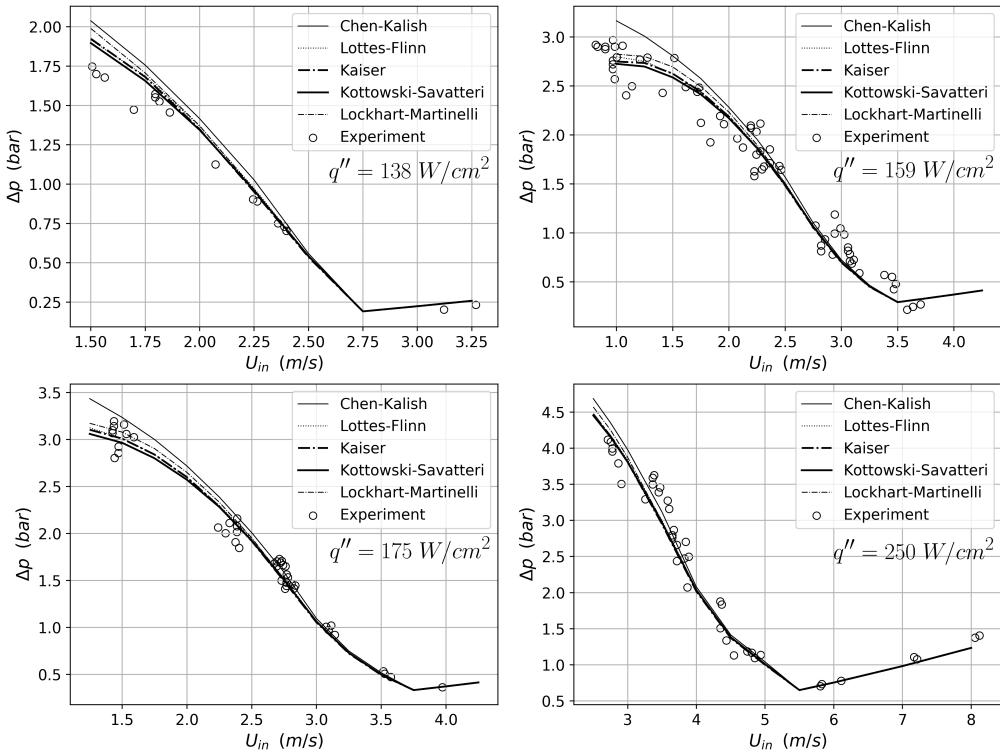


Figure 4.2.2: Comparison between the calculation results obtained by the various pressure drop multiplier correlations against experimental results for each of the four different wall heat fluxes. For each wall heat flux, calculations were performed by varying the inlet velocity in steps of 0.25 m/s.

The latent heat and saturation pressure-temperature relations were found to be important quantities in shaping the calculated curves. The higher the pressure,

the higher the corresponding saturation temperature, which in turn reduces the overall vapour production, its mass flow rate, and ultimately the overall pressure drop. Conversely, the evaporation latent heat model predicts smaller values of the latent heat for higher saturation temperatures (consistently with the fact that e.g. at the critical point, the latent heat is null by definition), and thus, for higher system pressures. A smaller latent heat will result in a larger vapour production, yet the latent heat effects on the overall resulting pressure drop were found to be significantly smaller than those associated with the saturation pressure-temperature dependence.

4.3 Transient validation

In order to assess a wider range of functionalities, the focus is now shifted on the modelling of a sodium boiling experiment that was performed at the Kompakter Natriumsiede Kreislauf (KNS) experimental facility at the Kernforschungszentrum Karlsruhe (now the Karlsruhe Institute of Technology), Germany, in the 1980s.

The experimental setup and details pertaining the test itself is presented in subsection 4.3.1. The computational modelling choices in terms of model geometry and physics model choices are presented in subsection 4.3.2. The computational results will be presented alongside the experimental ones in subsection 4.3.3

4.3.1 Description of the experiment

The KNS was originally set up within the broader framework of fast reactor development, specifically for the German SNR-300 Fast Breeder Reactor development program. The KNS was a sodium loop that allowed for the investigation of boiling in a variety of assembly geometries under Unprotected Loss Of Flow (ULOF) conditions. A schematic of the facility in its KNS-37 configuration is presented in Figure 4.3.1. This configuration was dedicated to the investigation of boiling phenomena in a electrically-heated mock-up 37-pin bundle.

Among the tests that were performed on the KNS facility in this configuration, the one chosen for this analysis consists of test L22. This test was about the investigation of sodium boiling subsequent to a fast pump trip transient representative of possible ULOF conditions. Qualitatively, the test proceeded as follows. For a certain axial input power profile with no radial tilt and inlet conditions, a steady state single-phase flow was reached. The pump was then tripped resulting in a rapid loss of flow, eventually inducing the onset of boiling. Due to the appreciable difference in the power-to-flow ratio between the assembly bulk and its periphery, the void region was initially limited to the bulk of the assembly and only minor inlet mass flow oscillations were observed. As the boiling progressed and the void region extended towards the assembly

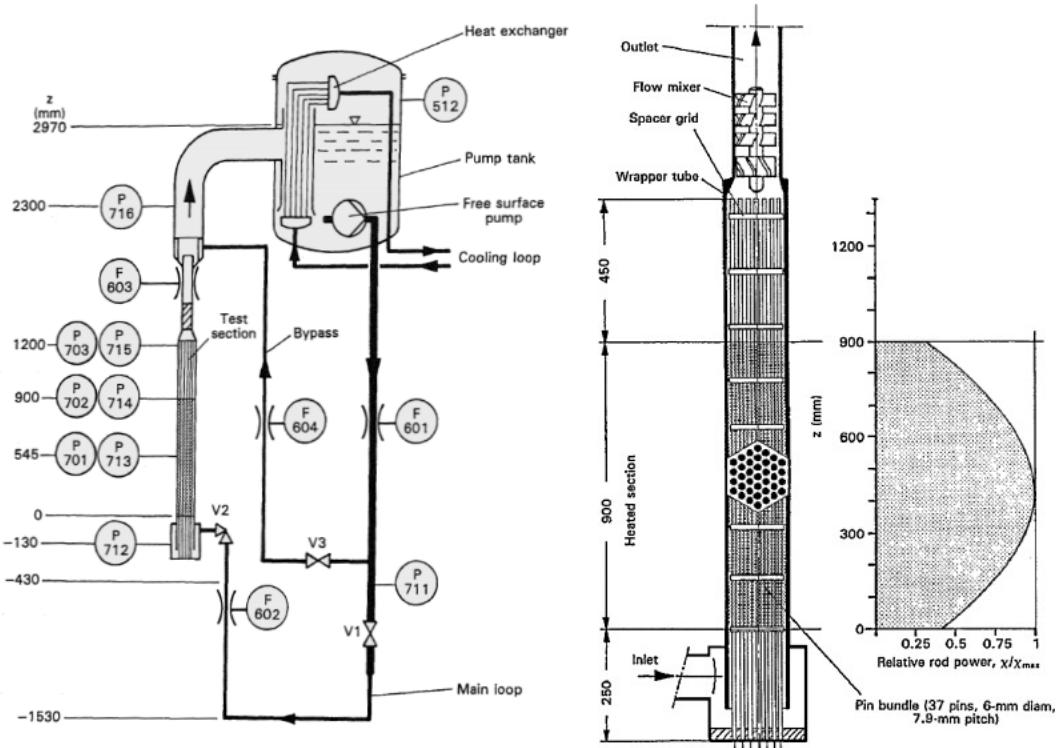


Figure 4.3.1: Schematic of the KNS test loop facility (right) and assembly details (left).

periphery, the increase in pressure drop results in a flow excursion. Among different quantities that were monitored at some locations throughout the test, pin surface cladding temperatures proved of great importance. In particular, a sharp rise in said temperatures was taken as indicative of dryout conditions, which was used as a criterion to power off the assembly. After power off, the transient proceeded until flow restoration and eventual end of boiling. The quantitative technical and transient details are presented in Tables 4.3.1 and 4.3.2.

Assembly data	
Number of pins	37
Pin diameter (mm)	6
Pin pitch (mm)	7.6
Flow area (cm^2)	11.7
Heated pin length (mm)	900
Unheated pin length (mm)	
- top	450
- bottom	200
Maximum linear power (W/cm)	320

Table 4.3.1: Pin bundle specifications.

test L22 data	
Steady state conditions	
Average linear power (W/cm)	215.4
Normalized axial power profile	$0.38 + 0.62 \sin(\pi \frac{z+25}{900})$
Bundle pressure (bar)	
- inlet	2.241
- outlet	1.045
Bulk coolant temperatures (K)	
- inlet	653
- outlet	812
Inlet mass flow (kg/s)	3.41
Boiling dynamics	
Normalized inlet mass flow ramp down until boiling	$(1 + 0.3003 t)^{-1.297}$
Time elapsed until (s)	
- onset of boiling	6.11
- onset of flow excursion	8.30
- onset of dryout	9.25
- power off	9.45
- end of boiling	12.31

Table 4.3.2: test L22 data. The variables z and t represent the axial distance from the bottom of the heated pin section in mm and the elapsed time since pump trip in s respectively.

4.3.2 Computational modelling

As mentioned earlier, the evolution of the transient is initially shaped by the radial vapour expansion. Because of this, and as it has already been highlighted in other works [40], flow excursions would be significantly anticipated in a 1-D model. This is due to the fact that in a 1-D model, right after boiling onset, the entirety of the available flow cross section is already occupied by some fraction of vapour. A 2-D axial-symmetric wedge model was then adopted, as it allows to refine the radial domain with a much smaller computational footprint when compared to a full 3-D model.

4.3.2.1 Geometrical modelling and volume averaging

The model consists of a 2-D axial symmetric wedge of 2 degrees of aperture. The pin bundle was treated with a coarse-mesh approach. Because of differences between the bulk of the pin bundle and its edge, different properties were obtained for these two regions. With reference to Figure 4.3.2, the inner bundle volume fraction and hydraulic diameter were estimated on the basis of the innermost triangular subchannel, in blue. Conversely, properties for the bundle edges were estimated on the basis of the highlighted trapezoidal edge region, in red.

Axially, the model extends from the bottom of the lower unheated pin region to 50 cm above the top of the upper unheated pin section. The condenser was not modelled as, experimentally, the vapour volume remained confined within the bundle, condensing while traversing the cooler upper unheated pin

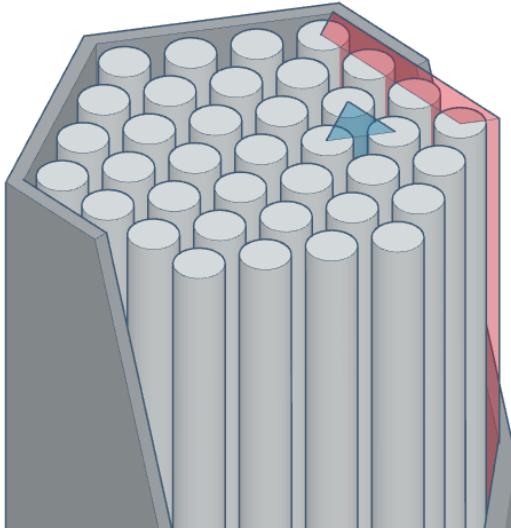


Figure 4.3.2: View of the geometric detail of the KNS-37 assembly, with the wrapper partially cut for ease of view. Inner bundle porous properties were calculated based on the REV highlighted in blue. Outer bundle porous properties were calculated based on the REV highlighted in red.

bundle region. The 2-D axial-symmetric wedge model is thus composed of seven regions: 1) two regions (inner and outer, i.e. edge, bundle) for the lower unheated pins; 2) two regions (inner and outer bundle) for the heated pins); 3) two regions (inner and outer bundle) for the upper unheated pin; 4) the region in between the outlet and the bundle top, which will be referred to as pipe region. The volume averaged coarse-mesh properties of these regions are summarized in Table 4.3.3.

Region	Dimensions				Coarse-mesh properties		
	Physical (mm)		Mesh (-)		α_s (-)	D_h (mm)	\mathbf{T} (-)
	Axial	Radial	Axial	Radial			
Inner lower bundle	200	21.553	10	5	0.523	5.469	0.441
Outer lower bundle	200	5.007	10	2	0.374	12.305	0.562
Inner heated bundle	900	21.553	23	5	0.523	5.469	0.441
Outer heated bundle	900	5.007	23	2	0.374	12.305	0.562
Inner upper bundle	450	21.553	12	5	0.523	5.469	0.441
Outer upper bundle	450	5.007	12	2	0.374	12.305	0.562
Pipe	500	26.560	13	7	0	53.120	1

Table 4.3.3: 2-D axial-symmetric wedge model data with regions listed in order, inwards from outwards, bottom to top. With regard to the porous properties columns, α_s refers to the structure volume fraction, D_h to the hydraulic diameter, \mathbf{T} to the transverse (i.e. radial) tortuosity component (the axial tortuosity is unity due to the fact that a pin bundle can be obtained by extrusion in the axial direction).

With regard to the thermal modelling, the pins were treated via a 1-D cylindrical model and coupled to the fluid as discussed in section 3.2.4, with a power density source prescribed according to the experimental axial power distribution in both the inner and the outer heated bundle regions. The interfacial

area density A_s''' of the pin cladding was derived from the volume averaging process to be related to the pin volume fraction and pin radius r_o as:

$$A_s''' = \frac{2\alpha_s}{r_o}$$

4.3.2.2 Constitutive relations and closure

In place of the i and j subscripts to represent the two fluids, the subscripts l and v will be here used to denote the liquid and vapour phase respectively.

Fluid-fluid drag With respect to equation 3.2.23, the following drag coefficient $f_{d,lv}$ models were investigated :

- Autruffe et al. [74], a correlation for two-phase sodium flows:

$$f_{d,lv} = 4.31 ((1 - \alpha_g^N)(1 + 75 (1 - \alpha_g^N)))^{0.95}$$

- Wallis, a correlation for two-phase sodium flows used in the SABENA code [11]:

$$f_{d,lv} = 0.02 \sqrt{\alpha_g^N} \left(1 + 150 (1 - \sqrt{\alpha_g^N}) \right)$$

Fluid-structure drag As discussed in section 3.2.2, the fluid-structure drag depends on the fluid-structure superficial drag coefficients, the fluid-structure contact fractions and the two phase pressure drop multiplier. For all flow regimes, the liquid was used as the basis fluid for the calculation of the drag coefficients via the two phase multiplier method, i.e. $M = l$ with reference to equation 3.2.17. With respect to said equation, the following drag coefficient $f_{d,is}$ models were investigated:

- Blasius correlation in the form:

$$f_{d,ls} = \frac{0.316}{Re_{ls}^{0.25}}$$

which was applied to both (i.e. axial and radial) flow directions in the pipe region of the model.

- Rehme [75] correlation for the friction factor for flows in wire-wrapped pin bundles with an hexagonal lattice:

$$f_{d,ls} = \left(\frac{64}{Re_{ls}} F^{0.5} + \frac{0.0816}{Re_{ls}^{0.133}} F^{0.9335} \right) \frac{W_p}{W_p + W_w}$$

$$F = \sqrt{\frac{P_t}{D_p}} + \left(7.6 \frac{D_p + D_w}{L_w} \left(\frac{P_t}{L_w} \right)^2 \right)^{2.16}$$

which was applied to the axial flow direction in all the pin bundle regions. It should be noted that the KNS-37 pins are spaced by a spacer grid rather than a wrapping wire. However, within a coarse-mesh context, the frictional pressure loss effect of said grids was captured by adjusting the Rehme model wire lead length (L_w , i.e. the axial pitch of the wounding) in order to obtain the measured steady-state pressure drop.

- Gunter-Shaw correlation in the form:

$$f_{d,ls} = \frac{0.96}{Re_{ls}}$$

which was applied to the transverse flow direction in all the pin bundle regions.

All of the two-phase pressure drop multiplier models that were presented in the Ispra experimental analysis section were investigated as well.

Fluid-fluid interface heat transfer coefficients The heat transfer coefficients between fluid bulk and fluid-fluid interface $H_{i,\partial}$ that govern boiling-related energy and mass transfer as per equations 3.3.35, 3.2.36 were calculated from a prescribed, constant Nusselt. The Nusselt numbers used both for the liquid and vapour side of the heat transfer coefficients were set to 10, as per the recommendations of Schor [76].

Fluid-structure heat transfer coefficients The heat transfer coefficient between liquid and structure was computed via the superposition model as reported in equation 3.2.29, thus relying on four sub-models for the convective heat transfer component, pool boiling heat transfer component, a flow enhancement factor and a flow suppression factor. In particular:

- for the convective component, a correlation in the form of equation 3.2.27 was used with the Nusselt computed as:

$$Nu = 7.485 + 0.030 Re_{ls}^{0.77} Pr_l^{0.77}$$

which was proposed by Mikityuk after a review of pin bundle to liquid sodium heat transfer coefficient data [38];

- for the pool boiling component, the correlation by Shah [77] for liquid metals was used:

$$H_{ls, PB} = C q_{ls}^{0.7} p_r^m$$

$$C = \begin{cases} 13.7 & p_r < 0.001 \\ 6.8 & p_r \geq 0.001 \end{cases}$$

$$m = \begin{cases} 0.22 & p_r < 0.001 \\ 0.12 & p_r \geq 0.001 \end{cases}$$

with q_{ls} being the fluid-structure heat flux and and p_r the reduced pressure (i.e. the pressure divided by the critical pressure of the liquid in exam. Among other metals, this correlations was verified for Sodium in a wide range of pressures. The numerical inconvenience of pool boiling correlations is that they generally related the heat transfer coefficient to the heat flux, which is what the heat transfer coefficients is needed for in the first place. Within the scope of this code, this dependence is treated explicitly;

- for the flow enhancement and suppression factor models, the well known correlations by Chen [41] were employed.

With regard to the structure-vapour heat transfer coefficient, while it can nonetheless be modelled, it was deemed of secondary importance due to the fact that vapour and structure are expected to come into contact only for a very short period of time around dryout. The heat transfer was calculated assuming a constant Nusselt of unity. As for the fluid-structure drag coefficients, the final heat transfer effect depends on the fluid-structure contact fractions $f_{A''_s,i}$ between liquid and structure, vapour and structure, now discussed.

Fluid geometry models By fluid geometry models we mean any correlations that set the liquid-vapour interfacial area A''_∂ , dispersed phase characteristic dimension $D_{h,d}$ and the structure contact fractions $f_{A''_s,l}$, $f_{A''_s,v}$ that quantify what proportion of the structure (in this case, the pins) is wetted by the liquid and vapour respectively. With regard to the fluid-fluid interfacial area, both the correlations by Schor [76] and by No and Kazimi [78] were investigated.

For the remaining quantities, these were set via a regime map with two flow regimes, which we will refer to as annular flow regime and mist flow regime. This regime map was parametrized with respect to vapour volume fraction (normalized with respect to the structure volume fraction) α_v^N . Each of these regimes exists for certain values of the vapour volume fraction. In this case, the regime map can be entirely described by two parameters, $\alpha_{v,an}^N$ and $\alpha_{v,ms}^N$ with $\alpha_{v,an}^N < \alpha_{v,ms}^N$, namely the vapour volume fraction below which the annular regime is assumed to exist and the vapour volume fraction above which the mist regime is assumed to exist. For vapour volume fractions in between the two, all the parameters and the other models discussed so far are quadratically interpolated. Different regime bounds were investigated, and these will be discussed alongside the results.

For each regime, the fluid-structure contact fractions and characteristic dimensions were treated as constants. With regard to the contact fractions, these were set to $f_{A''_s,l} = 1.0$ and $f_{A''_s,v} = 0.0$ in the annular flow regime, assuming that the liquid is at all times in contact with the structure. Conversely, for the mist flow regime we assumed $f_{A''_s,l} = 0.0$ and $f_{A''_s,v} = 1.0$. Furthermore, the vapour was treated as the dispersed phase in the annular regime, and the

liquid was treated as the dispersed phase in the mist regime. Practically, this affects all quantities that rely on tracking which phase is dispersed and which is continuous in any given cell. Chief among these are continuous phase density and dispersed phase volume fraction.

4.3.2.3 Boundary conditions and simulation remarks

This transient necessitates the prescription of a pressure field at the inlet and outlet in order to resolve the pressure-driven flow oscillations and excursion. At the outlet, a pressure equal to the experimentally measured pressure of 1bar was imposed throughout the simulation. At the inlet, a time-dependent pressure profile is imposed in order to reproduce the experimentally reported pump mass flow ramp, as reported in Table 4.3.2.

All other fields have trivial boundary conditions: fixed values for the liquid temperature at the inlet and null gradient boundary conditions for all other quantities at both the inlet and outlet. On the wedge faces, a null gradient is imposed for all quantities in the direction normal to the faces. The power is turned off at the same time as the one reported experimentally.

4.3.3 Results

4.3.3.1 Single-phase results

The calculated and experimental results concerning the axial and radial temperature distribution at steady state and at boiling onset are reported in Figure 4.3.3. The overall agreement between the calculated and experimental values is good, yet the calculated radial temperature profile is somewhat under-estimated at the bundle-edges at steady-state.

The main issue lies in the fact that, at this stage of modelling, we relied on the same drag coefficient correlation, the one by Rehme, both for the inner and outer pin bundles. However, the outer bundle hydraulic diameter had to be adjusted to obtain the right flow redistribution between inner and outer bundle, which is paramount for the prediction of the radial temperature distribution. Ideally, one would require a different model for the drag coefficient in the outer bundle regions. While a number of models where experimented with, with the geometrically calculated outer bundle hydraulic diameter, these efforts were met with little success, so that it was preferred to use a single drag coefficient model and vary the outer hydraulic diameter until the radial temperature distribution (prioritizing agreement with the boiling-inception one over the steady-state one) could be reproduced. This is of fundamental importance for the transient evolution, as the radial temperature gradient governs the radial temperature vapour expansion rate, and thus the flow excursion dynamics.

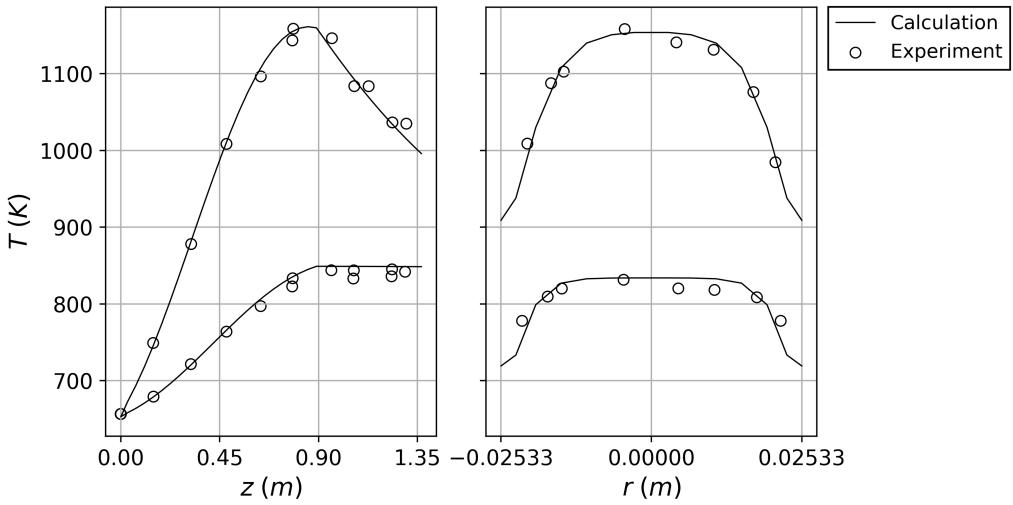


Figure 4.3.3: Experimental measurements and calculated results for the axial (left) and radial (right) temperature profiles at steady state (smaller values) and at the onset of boiling (larger values). The axial temperatures are evaluated along the bundle centerline, while the radial temperature profiles are evaluated at an axial elevation of $z = 779\text{ mm}$ from the bottom of the heated section of the pin bundle. A continuous line was used to report the calculated values for the sake of clarity.

4.3.3.2 Two-phase results

A number of different correlations for different quantities and parameter combinations were investigated. The results obtained by the application of four different sets of correlations and relevant parameters are presented and compared against the experimental ones, while the effects of changing other parameters are discussed qualitatively.

It should be noted that the overall outcome of modifying any given model or model parameters is itself dependent on the choice for correlations governing other quantities, yet a thorough sensitivity and co-variance analysis of all the models was out of scope for this investigation. For clarity, the four presented sets of results will be referred to as models I through IV. All of the presented results relied on the Schor interfacial area correlation, a virtual mass coefficient of $f_{vm} = 0.1$, an upper annular regime normalized vapour fraction threshold $\alpha_{v,an}^N = 0.94$ and a lower mist regime normalized vapour fraction threshold $\alpha_{v,ms}^N = 0.99$. The differences between the four models are summarized in Table 4.3.4

Figure 4.3.4 presents the results obtained with what we will refer to as model I. In particular, the Autruffe correlation for the liquid-vapour interfacial friction and the Kottowski-Savatteri two-phase multiplier were used.

Figures 4.3.5 and 4.3.7 present the results obtained by model II and III respectively, which used the same correlations as model I, except model II relied on the Lockhart-Martinelli two-phase pressure drop multiplier and model

Model	$\phi^2 (-)$	$f_{d,lv} (-)$
I	Kottowski-Savatteri	Autruffe
II	Lockhart-Martinelli	Autruffe
III	Chen-Kalish	Autruffe
IV	Kottowski-Savatteri	Wallis

Table 4.3.4: Summary of the main differences between models I through IV, which differ in the choice of the two-phase pressure drop multiplier ϕ^2 and the liquid-vapour drag coefficient $f_{d,lv}$.

III on the Chen-Kalish one. Figure 4.3.4 finally presents the results obtained by model IV, which used the Wallis liquid-vapour interfacial friction correlation alongside the Kottowski-Savatteri two-phase multiplier model.

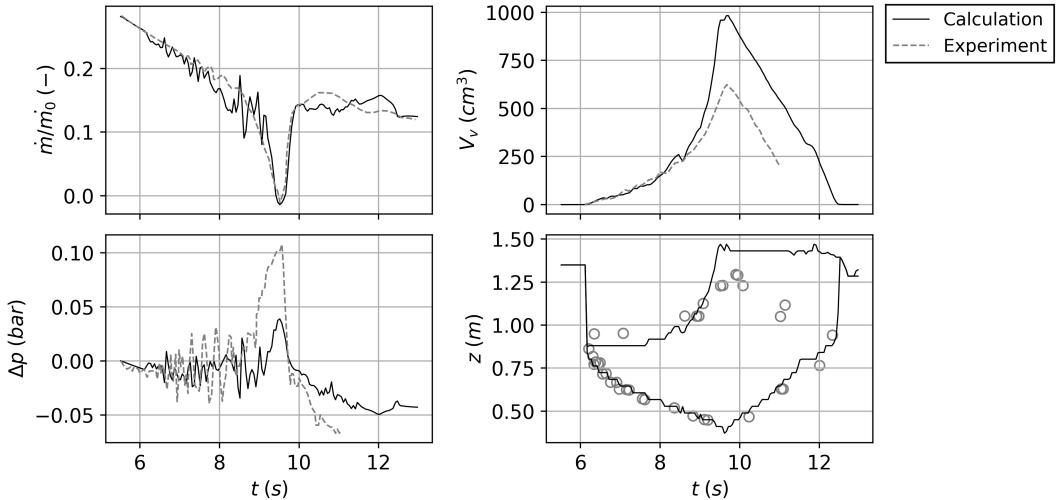


Figure 4.3.4: Model I, calculated results in time for: normalized inlet mass flow (top left); total vapour volume (top right); absolute pressure change at the axial location $z = 779 \text{ mm}$ (bottom left); maximum and minimum axial vapour extent (defined as $\alpha_v^N \geq 0.01$, bottom right). The time is measured with respect to the pump trip time at $t = 0 \text{ s}$.

All models predict the same time for the onset of boiling, which is a consequence of the inlet mass flow ramp down and pin power which were the same in all cases. In particular, the boiling onset was predicted at $t = 6.14 \text{ s}$ after pump trip, comparable to the experimental $t = 6.11 \text{ s}$. In the time window between boiling onset and the start of flow excursion, all models yield equivalent results in terms of total vapour volume evolution (top right plot of each figure), and axial vapour expansion (bottom right plot of each figure), and they compare well against the experimental data.

With regard to the inlet mass flow rate (top left plot of each figure) and pressure (always evaluated at $z = 779 \text{ mm}$, bottom left plot of each figure), the average trend of all models except for model III are comparable between each other and comparable to experimental results. The main difference results in the magnitude of the high-frequency (in the $3-6 \text{ Hz}$ range) oscillations, which was found to be higher for model II, that employs the Wallis correlation. This

is primarily due to the fact that the Wallis model predicts a lower value of the drag coefficient across all vapour void fraction ranges, so that the decreased momentum coupling between the fluid decreases oscillation dampening.

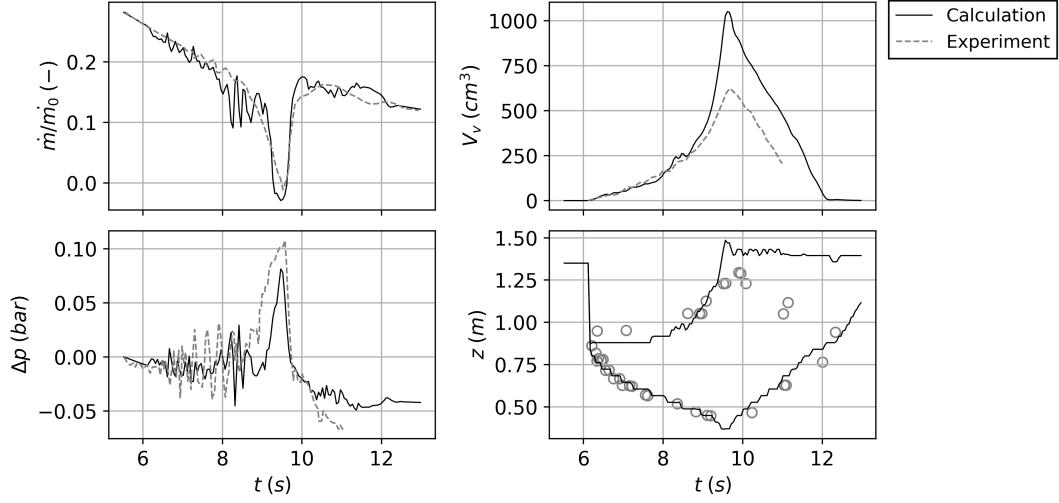


Figure 4.3.5: Model II, calculated results in time for: normalized inlet mass flow (top left); total vapour volume (top right); absolute pressure change at the axial location $z = 779 \text{ mm}$ (bottom left); maximum and minimum axial vapour extent (defined as $\alpha_v^N \geq 0.01$, bottom right). The time is measured with respect to the pump trip time at $t = 0 \text{ s}$.

Another important aspect that was later on found to reduce the inlet mass flow oscillations, regardless of the models, was to extend the overall simulation domain, e.g. lengthening the assembly. This effectively allows to model the inertia of the rest of liquid sodium in the loop, which is otherwise neglected if limiting the analysis to the assembly domain and imposing von Neumann boundary conditions for the velocity at the inlet. Regardless, the reason why model III predicts an overall faster inlet mass flow reduction is due to the fact that, as it was observed for the steady-state Ispra experiments, it predicts a larger two phase drag multiplier factor.

With regard to the vapour axial and radial expansion in this initial transient phase (bottom right plot of each figure), all models agree well both among each other and with experimental results, especially with regard to the downward vapour expansion. In particular, boiling starts in the innermost cells at the axial location between $z = 860 \text{ mm}$ and $z = 900 \text{ s}$, that is, at the top of the heated bundle. The large axial temperature gradient in the upper unheated bundle region combined with the much smaller axial temperature gradient within the top of the heated bundle itself, below the unheated part, results in an overall downward vapour axial expansion in the initial phase of the transient. As the upper bundle is heated by the condensing vapour, the upper front starts rising as well. Radially, while the results are not shown, the vapour reached the outer bundle in the $8.25 - 8.35 \text{ s}$ range for all models except model III, which anticipates it at 7.30 s . The start of the flow ex-

cursion follows shortly after for all models and can be recognized clearly from the sharp increase in pressure and decrease in inlet mass flow. All models except for model III predicted a somewhat delayed start of the flow excursion at 8.90 s, compared to the experimentally measured 8.30 s. As an example, the radial vapour distribution predicted by model I at three different instances in time is reported in Figure 4.3.6.

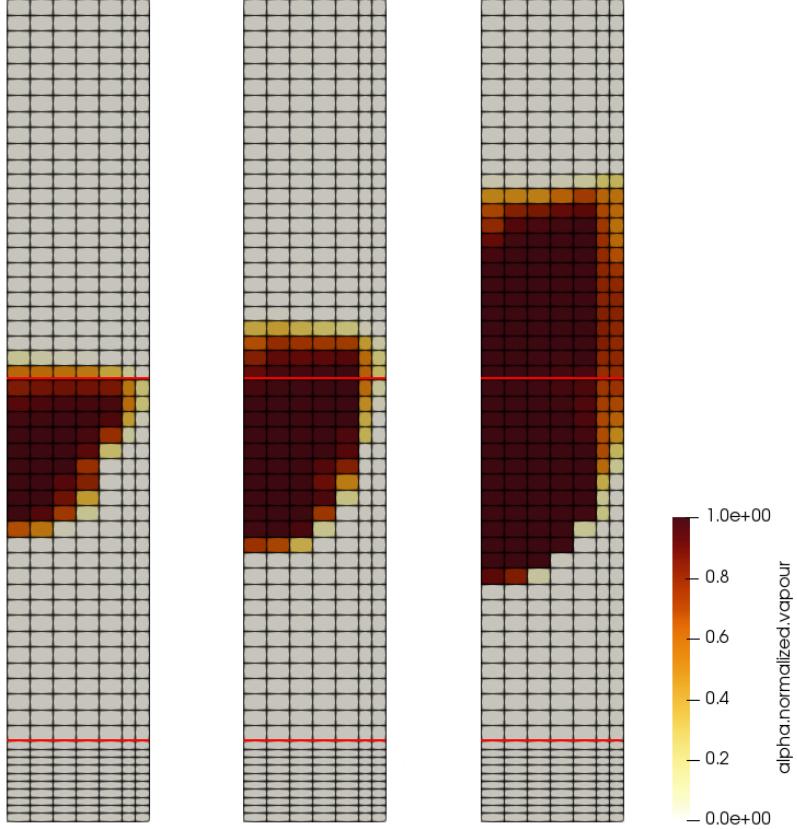


Figure 4.3.6: Distribution of the normalized vapour phase fraction α_v^N at three different moments of the transient, when (from left to right): at $t = 8.30\text{s}$, the vapour reaches outer bundle region; at $t = 8.90\text{s}$, the flow excursion starts; at $t = 9.45$, the power is turned off. Please recall that the mesh domain consists of a 2-D axial-symmetric wedge, with the wedge axis being the leftmost edge of the domain in each of the three images. The heated bundle section upper and lower boundaries are highlighted with red lines. The mesh and domain are axially scaled by a factor 0.075 for compactness.

From the perspective of inlet mass flow rate evolution however, the somewhat faster rate at which the flow excursion takes place makes it so that the overall inlet flow profile is well resolved by all models except model III. As it was highlighted earlier, the two-phase drag multiplier predicted by the Chen-Kalish model is higher than that predicted by other models, so that the flow excursion is much more severe, thus resulting in a larger vapour volume production and no flow recovery within the time bounds of the transient. For this reason, model III will not be referred to any further in the discussion.

Concerning the pressure profiles, while all of the three models under exam

under-predict the pressure increase, model I predicts the smallest increase among them. When comparing models I and II, this result can appear to be unexpected since, as it was observed in the Ispra experiments re-analysis, the Kottowski-Savatteri and the Lockhart-Martinelli correlations for the two-phase drag multiplier yielded comparable results in all investigated scenarios. The main difference lies in the fact that during the KNS transient, much larger vapour flow qualities were attained in some mesh cells, resulting in much smaller values of the liquid Lockhart-Martinelli parameter X_l , on which both the Kottowski-Savatteri and the Lockhart-Martinelli multipliers depend. In particular, at very low values of X_l , the Lockhart-Martinelli two-phase multiplier is larger than the Kottowski-Savatteri one, which explains the more pronounced pressure increase observed in model II compared to model I. Model IV also uses the Kottowski-Savatteri correlation, yet the use of the Wallis interfacial drag model results in a higher pressure increased compared to model I itself. The reason for this is not fully clear, yet a reasonable assumption is that this is due to the increase in the vapour flow quality due to the lower drag, on which the two-phase drag models depend.

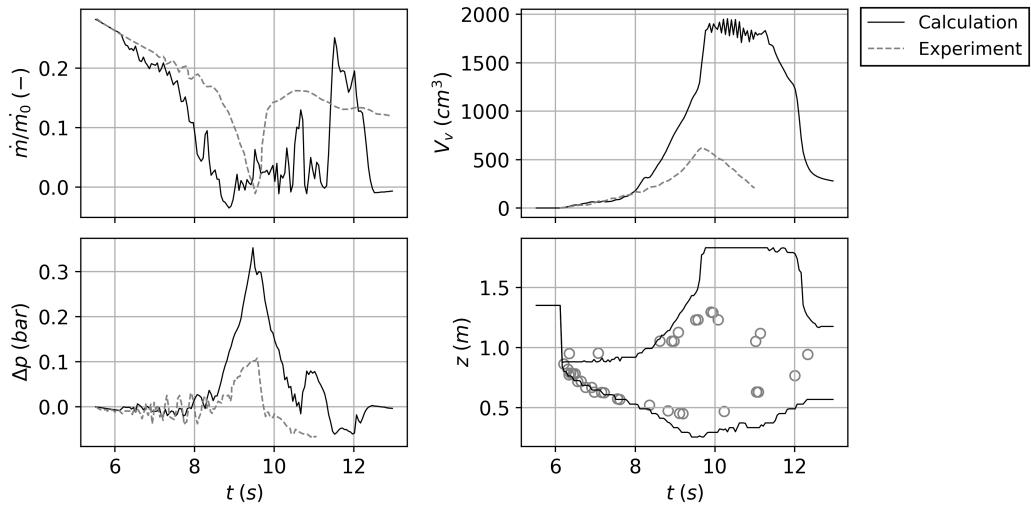


Figure 4.3.7: Model III, calculated results in time for: normalized inlet mass flow (top left); total vapour volume (top right); absolute pressure change at the axial location $z = 779 \text{ mm}$ (bottom left); maximum and minimum axial vapour extent (defined as $\alpha_v^N \geq 0.01$, bottom right). The time is measured with respect to the pump trip time at $t = 0 \text{ s}$.

Shortly after flow excursion, at 9.45 s , the power is turned off in all models. While experimentally this was performed in response to a dryout detection at 9.25 s following a rapid rise in cladding temperatures (of 90 K/s as measured at $z = 870 \text{ mm}$), no such reliable dryout criterion could be identified by the simulations. In fact, while all models predicted a surface pin temperature rise between the onset of boiling and the onset of flow excursion, they predicted a relatively slow rise of pin surface temperature of 50 K/s . For this reason, we preferred to impose a power-off time to be compatible with the experimental

ones.

With regard to the total vapour volume evolution (top right plot of each figure), regardless of the models, the same total vapour production is observed to the point where vapour reaches the outer bundle region. After that, behaviour varies significantly between the models, with model IV predicting a vapour volume evolution that is the closest to the experimentally measured one (which was measured indirectly via a time-integrated difference between inlet and outlet liquid mass flows). In this sense, model I and II predict that the upper axial vapour front reaches the outlet of the bundle, while model IV predicts an upper axial vapour front evolution that resembles experimental data points more closely.

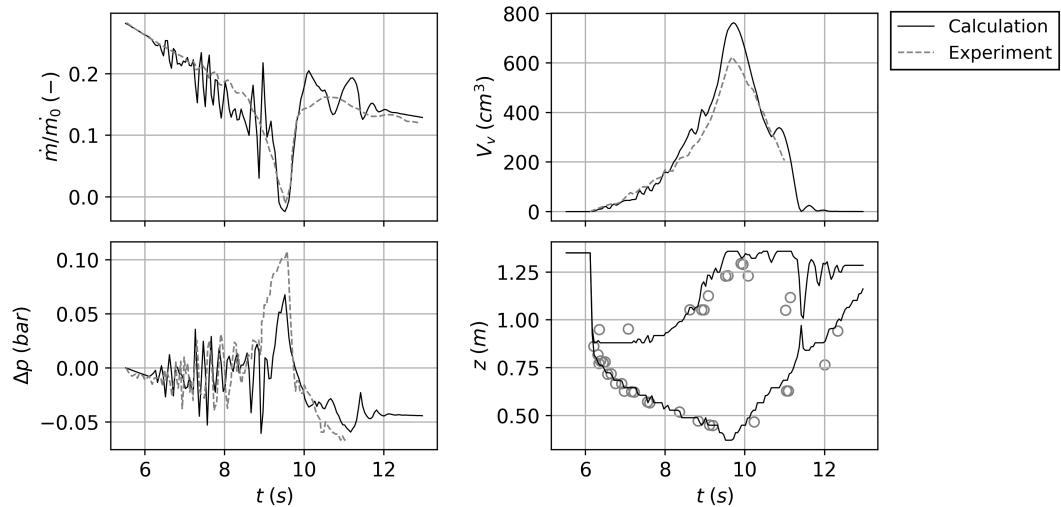


Figure 4.3.8: Model IV, calculated results in time for: normalized inlet mass flow (top left); total vapour volume (top right); absolute pressure change at the axial location $z = 779 \text{ mm}$ (bottom left); maximum and minimum axial vapour extent (defined as $\alpha_v^N \geq 0.01$, bottom right). The time is measured with respect to the pump trip time at $t = 0 \text{ s}$.

With regard to the inlet mass flow re-establishment, all the three models agree well with experimental results. Concerning the end of boiling, it varied between 12 s and 13 s for all models, compared to the experimentally determined 12.30 s . The overall rates at which the vapour condenses compare well between all models and the experiment, in spite of starting from different maxima, which explains the different end-of-boiling times.

With regard to other models and parameters, a qualitative note of interest is made regarding virtual mass effects. As expected, the virtual mass coefficient was found to significantly reduce maximum liquid and vapour flow velocities in the boiling region, without significantly affecting overall results (as most parameters of relevance, e.g. the vapour flow quality, are defined in relative and not absolute terms). However, the reduction in flow velocities had the overall effect of reducing inlet mass flow oscillations and, most importantly, calculation times. In particular, simulations performed without virtual mass

effects were found to take up to 4 times as long as calculations with a virtual mass coefficient as low as 0.1. While this is also due to the aforementioned fluid velocity reduction, better convergence of the numerical solution of the pressure-velocity system was also observed. These findings appear to be in line with extensive numerical analyses performed by Lahey on virtual mass effects in accelerating two-phase flows [37].

4.4 Conclusions

This chapter focused on a preliminary validation effort against steady-state and transient sodium boiling experiments, as well as the one-phase heat-up dynamics that lead to boiling.

The steady-state experiments, performed at JRC, Ispra (Italy), consisted in sodium boiling in heated tube geometries at constant inlet flow velocities and inlet power levels until a quasi-steady state is reached. This proved an excellent ground for the validation of some of the models of code, as the sensitivity of quasi-steady-state boiling scenarios is small with respect to a variety of models that are otherwise relevant in transient scenarios. The results yielded by this investigation chiefly consisted of total frictional pressure drop magnitudes for varying magnitudes of inlet mass flow rate, and the experimental results could be reproduced satisfactorily. Practically, the use of the Kottowski-Savatteri [67] correlation for the two-phase pressure drop multiplier yielded the closest results to the experimental ones, though the overall spread resulting from the use of the other investigated correlations is within experimental uncertainties, with the exception of the Chen-Kalish two-phase pressure drop multiplier, which was devised from Potassium boiling data. With regards to the thermo-physical properties of sodium, the adoption of a pressure-dependent saturation temperature model (based on the data compiled by Fink and Leibowitz [73]) proved of great importance to capturing the boiling hydraulic characteristic of the system under investigation.

Transient boiling experiments, more specifically consisting of test L22 performed at the KNS-37 facility at Kernforschungszentrum Karlsruhe (Germany) were modelled and investigated to assess the transient capabilities of the solver. Different correlations for a variety of quantities were investigated. While an appreciable sensitivity of the overall calculation results to the selection of models for the different involved parameters was highlighted, as expected from transient two-phase simulations, the obtained results were deemed overall satisfactory when compared to the experimental ones. These results chiefly consisted of the prediction of inlet mass flow rates, total vapour volume, axial distribution of the vapour and total pressure at specific assembly locations. Unlike the steady-state results, due the significantly more “integral” nature of these simulations (i.e. its degree of sensitivity to virtually all parameters which require correlations), the validation cannot be said to pertain to a specific model for a specific parameter, rather, it pertains to the overall choice of a

number of models for a variety of parameters. While a full sensitivity analysis was out-of-scope for this work, some of the parameters to which simulation results were more sensitive to consist of the liquid-vapour drag coefficient and the two-phase pressure drop multiplier. For these parameters, the correlations that yielded the closest agreement with experimental data consist of the Kottowski-Savatteri correlation for the two-phase pressure drop multiplier and the Wallis [11] correlation for the liquid-vapour drag coefficient.

Chapter 5

Application to flow blockage investigation in SFRs

5.1 Introduction

The present chapter discusses an instance of the application of the one-phase thermal-hydraulic methodology to the analysis of design features in Sodium-cooled Fast Reactors (SFRs) and their impact on assembly flow blockage scenarios. These features consist of the inter-assembly gap flow, which is a consequence of wrapped fuel elements, as well as a novel design feature proposed for SFRs and consisting of *wrapper windows*.

Fuel assemblies in SFRs are typically wrapped by an outer can, referred to also as wrapper. This design feature allows to control the power-to-flow ratio in individual assemblies throughout the reactor core. Inter-assembly gap flow is thus a standard feature in SFRs and it can constitute a non-negligible heat sink given the excellent thermal properties of liquid sodium ($> 60 \text{ W/m/K}$ of thermal conductivity) and the favourable heat transfer properties of the assembly steel wrapper. While the role of the inter-assembly gap used to be neglected or strongly approximated in the past, the need for a more detailed representation of the inter-assembly flow has recently been recognized in some investigations [79][80][81].

Assembly wrapper windows [82] are instead an innovative design feature that consists of openings in the wrappers below the end of the active fuel region. At first, this feature might seem to defy the purpose of the wrappers themselves. However, while the resulting flow coupling between neighbouring assemblies might be undesirable in steady-state operation, these windows are intended to limit the negative consequences that might arise from sodium boiling in transient scenarios. Sodium boiling can in fact lead to a significant flow reduction, which in turn can exacerbate the consequences of an Unprotected Loss Of Flow (ULOF) scenario [83][67], as it was illustrated in the previous chapter. The flow excursion is due to the considerable increase of the friction factor in two-phase flow conditions, which facilitates the downward propaga-

tion of the sodium void, potentially to core regions characterized by a local high positive void reactivity coefficient. Assembly windows can then be used to help restricting the presence of vapour to the upper core region in such scenarios.

A detailed simulation of the inter-assembly flow and of the assembly windows can provide important information in various reactor conditions. In this work, the assembly flow blockage was chosen as a test case. It represents both a severe accident initiator in SFRs, as seen in the partial fuel meltdown in the Unit 1 of the E. Fermi Atomic Power Plant (Michigan, U.S.A.) [84] and a suitable case to highlight the impact of 3-D effects due to the large radial thermal gradients and sodium flow through the windows. The investigation was however limited to one-phase scenarios as the one-phase solution algorithm and modelling capabilities were developed far before the two-phase ones, and the overall time frame of the doctoral work did not allow for a re-analysis with the two-phase algorithm. Regardless, this application makes use of important developments required for the coarse-mesh modelling of heat transfer through the wrappers and of the sodium flow through the assembly windows. It also discusses the important aspect of mesh-convergence related to the coarse-mesh methodology.

Sub-section 5.2 presents the developed wrapper heat transfer model and the wrapper window pressure drop model. Sub-section 5.3 presents the modelling of the assembly bundle and the blockage investigation procedure, which was performed on a representative assembly of the latest European Sodium-cooled Fast Reactor (ESFR) design [85]. Sub-section 5.4 presents the results of the mesh convergence study. Sub-section 5.5 presents the results of the actual parametric analysis of the thermal-hydraulic impact of the features under investigation in flow-blockage conditions. Relevant conclusions are drawn in sub-section 5.6.

5.2 Development of dedicated internal boundary conditions

5.2.1 Wrapper heat transfer modelling

In SFRs, the fuel assemblies are encased in so-called wrappers and are separated by a sodium gap a few millimeters in thickness. These can be modelled in OpenFOAM via 2-D internal boundaries (called baffles) that physically separate the flow in the assemblies from that in the gap, and that take into account the thermal resistance of the wrapper. To understand how this heat transfer through the wrappers can be modelled, let us consider a baffle with a physical thickness δ_b , a thermal conductivity k_b and let us denote the two sides of the baffle with + and - superscripts. The temperatures of two opposing baffle parcels will be denoted with T_b^+ and T_b^- , while the cell center temperatures of the cells adjacent to these baffle parcels will be denoted with T_c^+ and T_c^- . A

representation of this is provided in Figure 5.2.1.

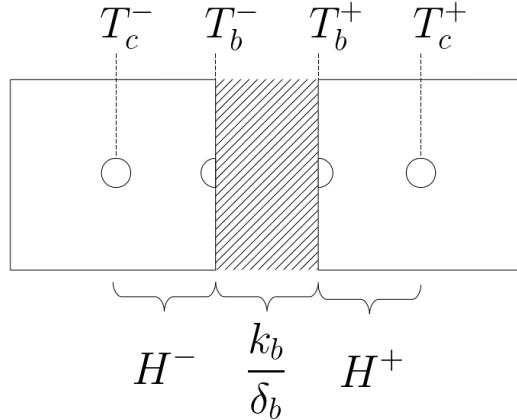


Figure 5.2.1: Schematic of two computational cells at the opposing sides of a thermally conductive baffle. Cell centers are marked with full circles while face centers at cell boundaries are marked with half circles. The heat transfer coefficients between relevant positions in the geometry are reported. From a geometric modelling perspective, the baffle has no thickness but the heat transfer is treated as if it had a thickness δ_b .

It should be stressed that the baffle thickness is not explicitly resolved in the computational geometry. For this reason, the two sides of the baffle lie on the same plane. By neglecting the thermal inertia of the baffle (a reasonable assumption if the absolute heat capacity of the wrapper is small compared to that of the sodium in the domain), the baffle surface temperatures can be computed by enforcing heat flux conservation, namely that the heat per unit area that flows into one side of the baffle is instantaneously the same as the heat per unit area flowing out of the baffle. Thus:

$$\frac{1}{\frac{\delta_b}{k_b} + \frac{1}{H^+}}(T_c^+ - T_b^-) = H^-(T_b^- - T_c^-) \quad (5.2.1)$$

$$\frac{1}{\frac{\delta_b}{k_b} + \frac{1}{H^-}}(T_c^- - T_b^+) = H^+(T_b^+ - T_c^+) \quad (5.2.2)$$

The equations represent an instantaneous heat flux balance, given that: 1) the heat transfer between a cell and the adjacent baffle parcel on the + side is characterised by the heat transfer coefficient H^+ ; 2) the heat transfer between the two opposing baffle parcels on the + and - side respectively by the heat transfer coefficient δ_b/k_b ; 3) the heat transfer between the baffle parcel and the adjacent cell on the - side is characterised by heat transfer coefficient H^- . These heat transfer coefficients are computed according to 3.2.27 with user-specified correlations for the Nusselt number as seen in equation 3.2.28. Due to the code architecture, the boundary conditions cannot be managed via the flow regime map approach internal to the code as discussed in subsection 3.2.1 in chapter 3. This boundary condition thus further implements an internal flow regime map of possibly two flow regimes, namely laminar

and turbulent, whose correlation coefficients and regime bounds can be user-specified, with linear interpolation of the Nusselt in the transition regime. Within the overall algorithm framework presented in Figure 3.3.1 in chapter 3, the baffle temperatures are updated according to 5.2.1, 5.2.2 right after the solution of the phase enthalpy equations within each outer iteration.

The approach was analytically verified in 1-D scenarios.

5.2.2 Pressure baffle

Consider two adjacent regions of the computational domain characterized by some macroscopic averaged parameters. Given a fluid flow between the two regions, an additional irreversible pressure jump is physically expected at the interface in case of a change of the sub scale structure geometry or due to a change of flow characteristics (e.g. direction). For the practical case under investigation, an ESFR assembly, this scenario is identified at the wrapper windows. In particular, the pressure jump is modelled in a similar manner as for orifices and pipe bends, that is:

$$\Delta p = \xi_k \frac{1}{2} \rho |\mathbf{u}|^2 \quad (5.2.3)$$

with ξ_k a head loss coefficient. As a limiting case (i.e. maximum pressure drop), the head loss coefficient for the present application was assumed to be equal to that associated with a double right-angled mitered pipe bend. This would be the pressure loss if the fluid e.g. entering the assembly from the gap were forced to bend at a right angle with respect to the main flow direction. Additional irreversible pressure losses are expected within the assembly itself at the interface between regions characterized by sharp changes in the porosity (e.g. the pin bundle and the intra-assembly plenum). However, these losses were found to be negligible in magnitude. An internal boundary condition for the non-hydrostatic component of the pressure was thus developed and applied to the windows. It consists of an internal boundary conditions that imposes a pressure jump on the boundary on which it is applied (i.e. the wrapper windows), which is calculated according to 5.2.3.

5.3 Computational modelling

5.3.1 Geometric modelling

Three distinct geometric models are investigated, based on the data of the ESFR-SMART core assemblies, with the main design parameters reported in Table 5.3.1. These models consist of: 1) a single assembly with thermally adiabatic wrappers; 2) a bundle of seven assemblies with thermally conductive, window-less wrappers, separated by an inter-assembly gap; 3) a bundle of seven assemblies with thermally conductive wrappers with windows, separated

by an inter-assembly gap. These models will be referred to as “adiabatic”, “windowless” and “windowed” respectively in the remainder of this chapter.

The adiabatic and windowless models span over the axial length of the fuel region within the assemblies. The windowed model extends up to the assembly outlet. This modelling extension is necessary as sodium flow through windows is governed by the pressure gradients between the assemblies and the gap. This in turn depends on the entire axial structure of the assembly, as a common pressure can only be reasonably assumed to be shared at the outlet. An overview of the computational domains of the models is presented in Figures 5.3.1 and 5.3.2.

The models are coarse-mesh based and the assembly properties are obtained via volume averaging as discussed earlier. In particular, the properties of the assemblies were obtained by averaging the pins over the inner assembly volume (i.e. exclusive of the wrapper), while the gap properties were obtained by averaging the wrapper thickness over the volume of the wrappers and the inter-assembly gap volume. Relevant geometric data, coarse-mesh modelling data, friction factor and Nusselt number correlations with the related parameters, when applicable, are reported in Table 5.3.2. In particular, the data for each of the axial regions of the windowed model is reported, while the adiabatic and windowless models share the data of the fuel axial region of the windowed model.

Quantity	Value
Reactor thermal power (<i>MW</i>)	3600
Number of inner fuel assemblies	216
Number of outer fuel assemblies	288
Number of pins per fuel assembly	271
Core outlet temperature ($^{\circ}\text{C}$)	395
Core inlet temperature ($^{\circ}\text{C}$)	545
Active core mass flow rate (kg/s)	19000
Fuel	MOX
Pu enrichment (% <i>wt.</i>)	17.99
Cladding and wrapper material	EM10 steel
Active fuel height, inner core (mm)	750.00
Active fuel height, outer core (mm)	950.00
Assembly lattice pitch (mm)	209.85
Wrapper outer flat-to-flat(mm)	205.35
Wrapper thickness (mm)	4.50
Inter-assembly gap width (mm)	4.50
Fuel pellet inner diameter (mm)	1.56
Fuel pellet outer diameter (mm)	4.68
Cladding inner diameter (mm)	4.84
Cladding outer diameter (mm)	5.34

Table 5.3.1: Reference design and performance data of the ESFR core

With reference to Table 5.3.2, a few remarks should be made. The gap height was not reported as it corresponds to the total model height, which varies between the three models. Two values of the hydraulic diameter for the fuel

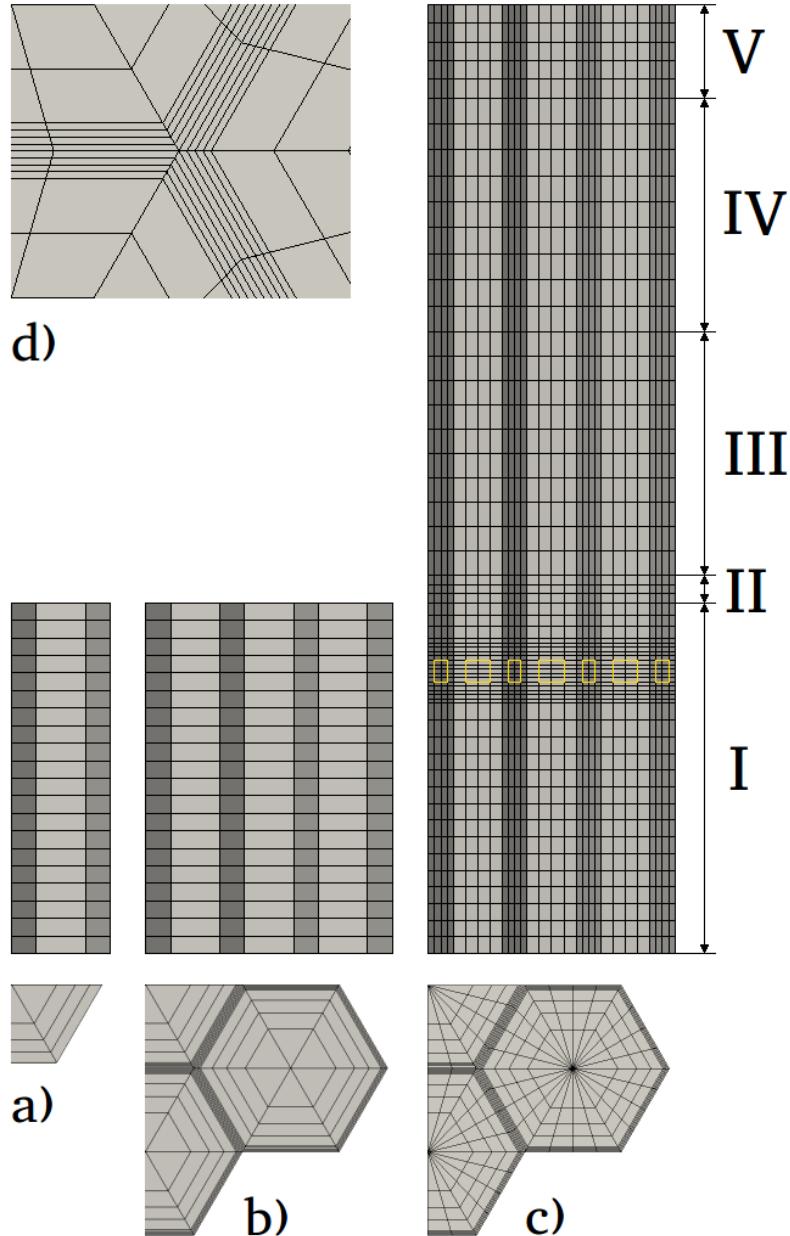


Figure 5.3.1: side view and a quarter of top view for: a) the single assembly model (i.e adiabatic); b) the seven assembly bundle with thermally conductive wrappers without windows (i.e windowless); c) the seven assembly bundle with thermally conductive wrappers with windows (i.e windowed); please notice that the windows cannot be seen from the side view as they are shrouded by the half-gap around the assemblies. The location of the windows behind the half-gap mesh was highlighted in yellow. The windowed model consists of five axial regions: I) fuel; II) plugs; III) intra-assembly plenum; IV) upper shield; V) assembly head. d) Detail of the inter-assembly gap meshing.

region are reported. The bulk value is employed for the computation of the friction factors and pin-sodium heat transfer coefficients, while the edge value is used for sodium-wrapper heat transfer coefficient calculations. With regard

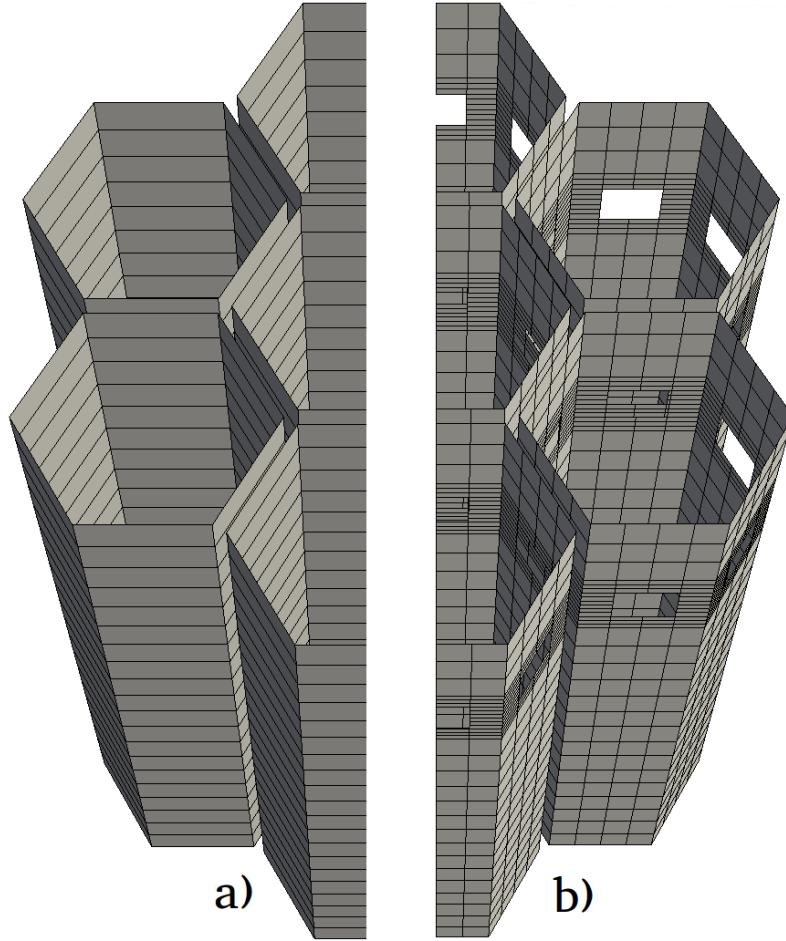


Figure 5.3.2: detail of the assembly wrappers: a) without windows; b) with windows. Please notice that the wrappers in the windowed model extend up to the assembly outlet. However, for a clearer view of the windows, the wrappers have been cut at the end of the fuel region.

to friction factor correlations, a Blasius correlation is used to model friction in the intra-assembly plena, heads and in the inter-assembly gap. For the pin-occupied regions, a Rehme correlation [75] with adequate geometric parameters was employed for the calculation of the momentum sink parallel to the bundle direction, while a Gunter-Shaw correlation [86] was employed for the momentum sink due to cross-flow in the direction perpendicular to the pins.

Two sets of parameters for the Nusselt calculations are employed throughout the model, one for the laminar regime (i.e. for $Re < Re_{lam}$) and one for the turbulent regime (i.e. for $Re > Re_{turb}$). The coefficients were obtained from the proposed correlation by Mikityuk [38] with respect to the present fuel bundle geometry for the turbulent regime. For the laminar regime, the Nusselt number is constant. For the transition regime, the Nusselt number is obtained via linear interpolation between the two Nusselt numbers computed at the transition regime boundaries. While the use the same correlation (devised for pin bundles) for the Nusselt number for both the pin-fluid heat transfer and

Quantity	Value in region					
	Fuel	Plugs	Plenum	Shield	Head	Gap
Height (mm)	864.29	68.00	600.00	576.00	230.00	-
Width (mm)	196.35	196.35	196.35	196.35	196.35	13.50
α_s (-)	0.268	0.268	1.000	0.280	1.000	0.362
T (-)	0.673	0.673	1.000	1.000	1.000	1.000
D_h (mm)	3.35 (bulk) 4.36 (edge)	3.35	113.36	3.37	113.36	9.10
A''_s (m^2/m^3)	272.08	0	0	0	0	0
Friction factor correlation	Rehme, Gunter-Shaw N (-) D_p (mm) P_p (mm) D_w (mm) H_w (mm)	Rehme, Gunter-Shaw N (-) D_p (mm) P_p (mm) D_w (mm) H_w (mm)	Rehme, Gunter-Shaw N (-) D_p (mm) P_p (mm) D_w (mm) H_w (mm)	Blasius Rehme, Gunter-Shaw N (-) D_p (mm) P_p (mm) D_w (mm) H_w (mm)	Blasius Blasius Blasius	Blasius
Value						
Nusselt correlation	Laminar regime					
	A_{Nu}	3.521				
	B_{Nu}	0				
	C_{Nu}	0				
	Turbulent regime					
	A_{Nu}	3.521				
	B_{Nu}	0.014				
	C_{Nu}	0.770				
	Re_{lam} (-)	1000				
	Re_{turb} (-)	2300				
	ξ_k (-)	1.17				

Table 5.3.2: Geometric and porous medium modelling data for the axial regions of the windowed model. The adiabatic and windowless models share the same data of the fuel region of the windowed model.

the wrapper wall-fluid heat transfer is debatable, it should be noted that this has little impact on the results of the present work. As a matter of fact, the overall heat transfer coefficient between the intra-assembly sodium and the sodium in the assembly gap will be limited by the thermal resistance of the wrapper wall (i.e. 4.5 mm of steel, $\kappa_w \approx 20 \text{ W/m/K}$). Even the worst case estimates for the fluid-wrapper Nusselt (e.g. for a $Nu \simeq 3$) result in a heat transfer coefficient that is one order of magnitude larger than the heat transfer coefficient of the wrapper itself. Furthermore, fine-tuning the turbulence model to fit currently unavailable experimental data was out of scope for the current work.

The tortuosity that is reported in Table 5.3.2 is the scalar value of the tortuosity in the two transverse directions, as the tortuosity in the axial direction is unity (due to pin bundles fundamentally being axially-extruded geometries).

With regard to wrapper windows, these are opening with a hydraulic diameter of 60 mm whose center lies at 168 mm below the top of the fuel. It is recalled that these opening are located close to the top of the fuel so to prevent, in boiling scenarios, downward axial vapour propagation towards core regions characterized by a locally positive void reactivity coefficient. While the windows are circular in shape, for modelling simplicity these were represented as square openings characterized by the same hydraulic diameter as the actual window geometry. A head loss coefficient of $\xi_k = 1.17$ was assumed for the pressure drop across the windows. In particular, this value was obtained

by averaging the Reynolds-dependent correlation for head loss coefficients of right angle miter bends [87] over the range of Reynolds numbers of the investigated cases. The magnitude of the pressure drop is then computed according to 5.2.3, with appropriate directionality taken into account.

For the simulation of the different investigated scenarios, fixed value boundary conditions are imposed at the inlets of the central assembly, neighbouring assemblies and gap for the velocity and a common pressure is set at all the outlets. An axially cosine shaped, radially uniform fuel power profile is set across the fuel region in each assembly to model internal power production. The fuel pins are thermally modelled in each mesh cell of the fuel region via the 1-D nuclear fuel pin model described in sub-section 3.2.4, and in greater detail in appendix A.

5.3.2 Investigation procedure

Each simulated case is defined by four parameters, namely: 1) reactor power expressed as a fraction f_P of the power at steady state operation; 2) total sodium velocity in the neighbouring assemblies expressed as a fraction f_N of the reference sodium velocity at steady state operation; 3) sodium velocity in the central assembly expressed as a fraction f_C of the sodium velocity in the surrounding assemblies that do not undergo blockage; 4) sodium velocity in the inter-assembly gap expressed as a fraction f_G of the sodium velocity in the surrounding unblocked assemblies. A parametric approach with respect to this quantity is justified by the lack of experimental data regarding inter-assembly gap velocity magnitudes.

For every combination of these values, for each of the three models (i.e. adiabatic, windowless, windowed), simulations are performed in the following way. First, a simulation with no blockage is performed until (and if) a steady-state is reached, given the other parameters for reactor power, unblocked assembly sodium velocity and gap sodium velocity fractions (the gap is not modelled in the adiabatic case for obvious reasons). Afterwards, an instantaneous blockage is modelled in the form of a sudden sodium inlet velocity excursion in the central assembly, and the simulation is performed until either a new steady state or boiling temperatures are reached. If a new steady state is reached, the outlet bulk temperatures of the gap and central assembly are estimated. If boiling temperature is reached, only the time elapsed until the onset of boiling is registered and simulations are terminated, as the investigation was limited to one-phase scenarios.

The investigated parameters consist of: 1) 100 %, 5 % reactor power; 2) 100 %, 5 % sodium velocity in surrounding assemblies; 3) sodium velocity in central assembly from 0 % (i.e total blockage) to 60 % in steps of 10 % of the neighbouring assemblies sodium velocity; 4) sodium velocity in gap from 0 % to 100 % in steps of 20 % of the neighbouring assemblies sodium velocity. In particular, the investigated power levels were selected so that results characteristic of varying degrees of blockages at both full power ($f_P = 100 \%$) and at

power levels compatible with core status right after shutdown ($f_P = 5\%$, e.g. a protected transient following blockage detection) could be obtained. Nominal assembly parameters correspond to a power of 7.143 MW and a flow rate of 44.7 l/s . This corresponds to a sodium velocity of 5.00 m/s through the active core pin region.

5.4 Mesh convergence study

5.4.1 Theoretical remarks

A mesh convergence analysis was motivated by the specific challenges that coarse meshes pose. When selecting a mesh in a coarse-mesh context, it should be recalled that: 1) an overly fine mesh would defy the purpose of a coarse-mesh approach; 2) an overly coarse mesh might lead to unacceptable discretization errors in the obtained solution. To avoid large discretization errors one should be able to make an estimate on the error bounds obtained on the quantities of interest. For this estimate to be possible, however, the results need to be in what is referred to as *asymptotic convergence range* [22]. For a result of interest to be in the asymptotic convergence range, it needs to monotonically approach a definite value as the mesh is further refined. This behaviour enables a meaningful extrapolation of the results at an ideally infinitesimal mesh size, such as the approach of generalized Richardson extrapolation [88], which in turn allows for error bounds to be estimated.

Let us consider a certain output quantity of interest ϕ . If ϕ_e is the extrapolated value at an ideally null mesh spacing and ϕ_h is the value obtained at a certain mesh spacing h , the error E_h can be estimated as:

$$E_h = \phi_h - \phi_e = C h^q + O(q' > q) \quad (5.4.1)$$

wherein q is a certain order of convergence of the error. In particular, the obtained result ϕ_h is in the asymptotic convergence range if the error E_h can be expressed as a polynomial of the grid spacing whose lowest order (i.e. of order q) coefficient C does not depend on h . The big O notation is used to represent higher order terms that become negligible as the mesh spacing approaches infinitesimal values.

If the results are in the asymptotic convergence range, given three output values $\phi_i, \phi_{i+1}, \phi_{i+2}$ obtained respectively on progressively refined meshes by a factor $r = h_i/h_{i+1} = h_{i+1}/h_{i+2} > 1$, the convergence order q and the extrapolated value ϕ can be obtained as (via Richardson extrapolation):

$$q = \frac{1}{\ln(r)} \ln \left(\frac{\phi_i - \phi_{i+1}}{\phi_{i+1} - \phi_{i+2}} \right) \quad (5.4.2)$$

$$\phi_e \simeq \phi_i + (\phi_{i+1} - \phi_i) \frac{r^q}{r^q - 1} \quad (5.4.3)$$

Various estimators for the error on the coarse mesh solution ϕ_i can be defined:

$$e'_i = \left| \frac{\phi_i - \phi_{i+1}}{\phi_i} \right| \quad (5.4.4)$$

$$e_i = \left| \frac{\phi_i - \phi_e}{\phi_i} \right| = e'_i \frac{r^q}{r^q - 1} \quad (5.4.5)$$

Neither of these should be interpreted as error bounds on the coarse solution ϕ_i , rather as error ‘‘bands’’, i.e. a tolerance on the accuracy of the solutions, that may as well be exceeded.

In the experience of Roache [22], the most commonly reported error estimator in mesh convergence studies is the one represented by equation 5.4.4. However, this estimator contains no information of the actual order of convergence of the solution on its own, which is necessary to assess if the solution is in the asymptotic convergence range. This information is instead present in the relative error estimator e_i . To address this, Roache proposes the use of a so-called Grid Convergence Index (GCI) defined so that:

$$e'_i \frac{r^q}{r^q - 1} = GCI_i \frac{r_0^{q_0}}{r_0^{q_0} - 1} \quad (5.4.6)$$

The GCI thus represents what would be the equivalent error estimator in the form of equation 5.4.4 if the results were obtained by a q_0 -order accurate method for a mesh refinement ratio r_0 . Typical values suggested by Roache for the comparison are $r_0 = 2$ and $q_0 = 2$, thus consisting in a comparison against the uncertainties associated with the convergence of a second order accurate method with mesh halving. From equations 5.4.4, 5.4.5 and 5.4.6 it follows that:

$$GCI_i = \frac{3}{4} e_i = \frac{3}{4} \left| \frac{\phi_i - \phi_{i+1}}{\phi_i} \right| \frac{r^q}{r^q - 1} \quad (5.4.7)$$

It should be noted that this formulation differs by that of Roache by a factor $1/r_0^{q_0}$ as this is done from the perspective of error estimation for the coarser solution ϕ_i and not for the finer solution ϕ_{i+1} , yet the approach is the same.

It can be shown that the output of interest obtained by a coarse mesh ϕ_i is in the asymptotic convergence range if:

$$GCI_i \simeq GCI_{i+1} r^q \quad (5.4.8)$$

This can be computed as long as the output parameters ϕ_{i+1} , ϕ_{i+2} obtained at subsequently refined meshes by a factor r are available. Thus, to assess whether results are in the asymptotic convergence range (ACR), the following ratio is evaluated and checked whether it is within unity:

$$ACR = \frac{GCI_i}{GCI_{i+1} r^q} \quad (5.4.9)$$

To summarize, the mesh convergence effort is aimed at establishing whether or not results are in the asymptotic convergence range via the estimation of the

ACR and the subsequent estimation of representative expected error bands on the obtained results.

In the context of this work, this process is furthermore meant to establish which mesh topology is best suited for the spatial discretization of the domain given the nature of the investigated cases.

5.4.2 Results

Two cases were investigated, namely: 1) $f_P = 100\%$, $f_N = 100\%$, $f_C = 50\%$, $f_G = 100\%$; 2) $f_P = 5\%$, $f_N = 5\%$, $f_C = 50\%$, $f_G = 100\%$. These two cases will be referred to as “full power” and “low power”, respectively.

The quantity for which convergence properties were assessed was the outlet bulk temperature of the central assembly and the gap. In particular, with regard to the results concerning the GCIs, subscripts 1 and 2 will be used to denote the GCIs related to the coarse and fine meshes respectively.

5.4.2.1 Adiabatic and windowless model

For both the full and low-power cases, two different mesh topologies were investigated, reported in Figure 5.4.1. Referring to said figure, these meshes will be referred to as triangular mesh and web mesh.

The reason for investigating these topologies is that while the triangular mesh is among the most popular choices for the investigation of fast reactor assemblies, it might not be the most suitable option due to the added numerical diffusivity in this scenario. This comes from the fact that many cell faces will not be orthogonal to thermal gradients. The web mesh has been investigated to try to address this issue. The radial divisions within an assembly are such that each cell has the same overall volume.

For each mesh topology (Figure 5.4.1), different meshes obtained by varying the number of nodes in the characteristic directions were investigated: 1) n_A number of nodes in the axial direction; 2) n_R number of nodes in the assembly radial direction; 3) n_S number of nodes per assembly side (only for the web mesh); 4) n_G number of nodes in the half-gap radial direction. Any mesh for the windowless model can be described in terms of these four parameters.

The mesh that was found to be suitable for the scope of the subsequent parametric flow blockage investigation is a web mesh characterized by $n_A = 20$, $n_R = 4$, $n_S = 1$, $n_G = 4$. Convergence properties were investigated separately for radial and axial mesh refinements.

It is recalled that in order to assess the convergence properties, the results need to be obtained on two further refined meshes. For the case of radial refinement, this consisted in subsequently doubling both the number of radial nodes in the assembly n_R and half-gap n_G , while maintaining the same number of nodes in other directions. For the case of axial refinement, this consisted in subsequently doubling n_A while maintaining the same number of nodes in the other directions.

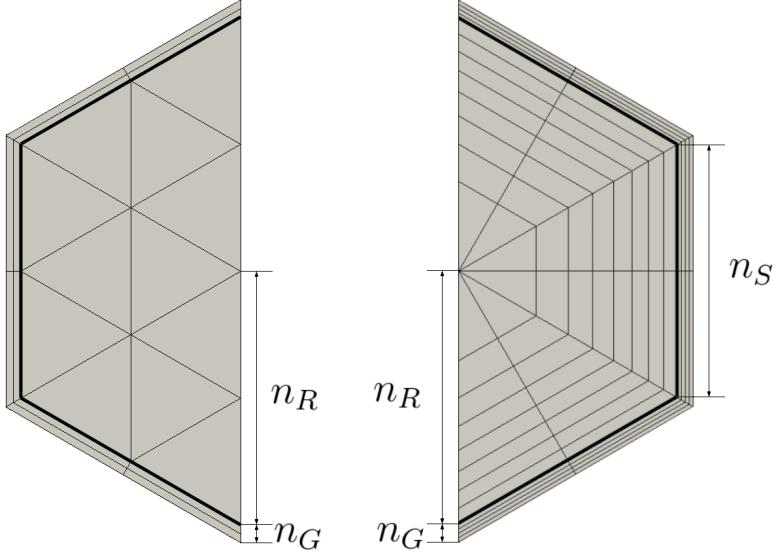


Figure 5.4.1: examples of half cross-sections of the investigated mesh topologies: (left) triangular mesh; (right) web mesh. The key quantities that characterise each mesh are reported. These consists in the number of computational nodes in each reported direction. Each mesh is further characterized by the number of nodes in the axial direction n_A , not reported in the figure.

The mesh convergence properties of the outlet bulk temperatures for the central assembly T_C and gap T_G for the low power case is provided in Table 5.4.1.

	Mesh	Value (K)	Refinement	$q (-)$	$GCI_1 (\%)$	$GCI_2 (\%)$	$ACR (\%)$	$T_e (K)$	$\Delta T (K)$
T_C	Tri	894.17	Radial	1.27	0.524	0.218	100.170	900.44	6.30
			Axial			-			
	Web	894.87	Radial	1.32	0.429	0.172	100.137	900.01	5.15
			Axial			-			
T_G	Tri	771.64	Radial	0.70	1.000	0.614	99.685	761.44	10.12
			Axial	0.67	0.054	0.034	100.017	772.20	0.56
	Web	771.10	Radial	0.71	0.817	0.500	99.741	762.74	8.31
			Axial	0.56	0.051	0.035	100.015	771.63	0.5

Table 5.4.1: mesh convergence results for the outlet bulk temperature of the central assembly T_C and gap T_G for the low power. The results are presented for radial and axial refinements starting from the results obtained by a web mesh and a triangular mesh with the same parameters: $n_R = 4$, $n_S = 1$, $n_G = 4$, $n_A = 20$. The quantity q represents the order of convergence as defined in equation 5.4.2. The convergence results related to the axial refinement of the mesh on T_C are not reported because the obtained values for T_C were found to be already at geometric convergence (i.e. $< 0.1 K$ difference between the coarse and the extrapolated values) for both the web and triangular mesh. The columns T_e and ΔT report the extrapolated temperature values and the error bands respectively.

The results are found to be in the asymptotic convergence range ($ACR \simeq 1$) for both the triangular and web meshes and all the refinement types.

Convergence properties related to central assembly temperature T_C for axial mesh refinement are not reported as these were found to be already at geometric convergence (i.e. $< 0.1 K$ difference between the coarse and the extrapolated values).

For a given set of radial mesh parameters, further axial refinements do not have a significant impact on the solution. This is expected, as the thermal gradients in the axial direction are much smaller than those in the radial direction. Furthermore, the web mesh is characterized by better mesh convergence properties in terms of the estimated error bands on the obtained results. This was also expected due to the previous remarks on the reduced numerical diffusivity that the web mesh provides.

The same trends in the mesh convergence analysis were found in the full power case. In particular, the results were still consistently in the asymptotic convergence range and the web mesh proved to be a more desirable choice with respect to the triangular mesh. The associated uncertainty bands were approximately twice as smaller when compared to the low power scenario.

5.4.2.2 Windowed model

The mesh choice for the windowed model is defined by a larger number of parameters when compared to the windowless model as there are multiple axial regions (Figure 5.3.1 c).

The mesh convergence properties of a windowed mesh with similar mesh parameters as those selected for the windowless model were assessed. The investigated windowed mesh consisted in a web mesh with $n_R = 4$, $n_G = 4$ (i.e. the same as for the windowless model), $n_S = 4$ (the bare minimum required to model windows of the correct size) and a variable number of mesh nodes depending on the axial zone. Separate axial and radial refinement studies were performed on the mesh for the low power and high power cases. Results for the web mesh only (the triangular mesh was not investigated) for the high power case are presented in Table 5.4.2.

	Value (K)	Refinement	$q (-)$	$GCI_1 (\%)$	$GCI_2 (\%)$	$ACR (\%)$	$T_e (K)$	$\Delta T (K)$
T_C	864.56	Radial	4.30	0.053	0.003	99.997	863.94	0.61
		Axial	1.19	0.195	0.086	99.936	862.31	2.25
T_G	712.64	Radial	0.81	0.146	0.083	99.952	711.26	1.38
		Axial	1.39	0.250	0.095	100.079	715.02	2.38

Table 5.4.2: mesh convergence results for the outlet bulk temperature of the central assembly T_C and gap T_G for the high power case for the windowed web mesh. The results are presented for radial and axial refinements starting from the results obtained by a web mesh with: $n_R = 4$, $n_S = 1$, $n_G = 4$ and a variable number of axial nodes depending on the axial region. The columns T_e and ΔT report the extrapolated temperature values and the error bands respectively.

All of the results are in the asymptotic convergence range, and the same was observed for the low power case. The associated uncertainty bands are of the same order of magnitude as for the windowless case. However, the relative importance of axial refinement with respect to radial refinement is increased in this case, as testified by the reported uncertainty bands. The same was found for the low power case. This was expected, as the outlet bulk temperatures

are no longer shaped exclusively by the heat transfer process across the wrapper, but also by the 3-D mixing processes that occur at the windows. As an example, the differences in flow resolution around wrapper windows between the coarsest and the most refined (both axially and radially) case is presented in Figure 5.4.2.

In conclusion, the selected web mesh that is radially identical to and axially comparable to the one used for the windowless model was found to be suitable for the windowed model as well.

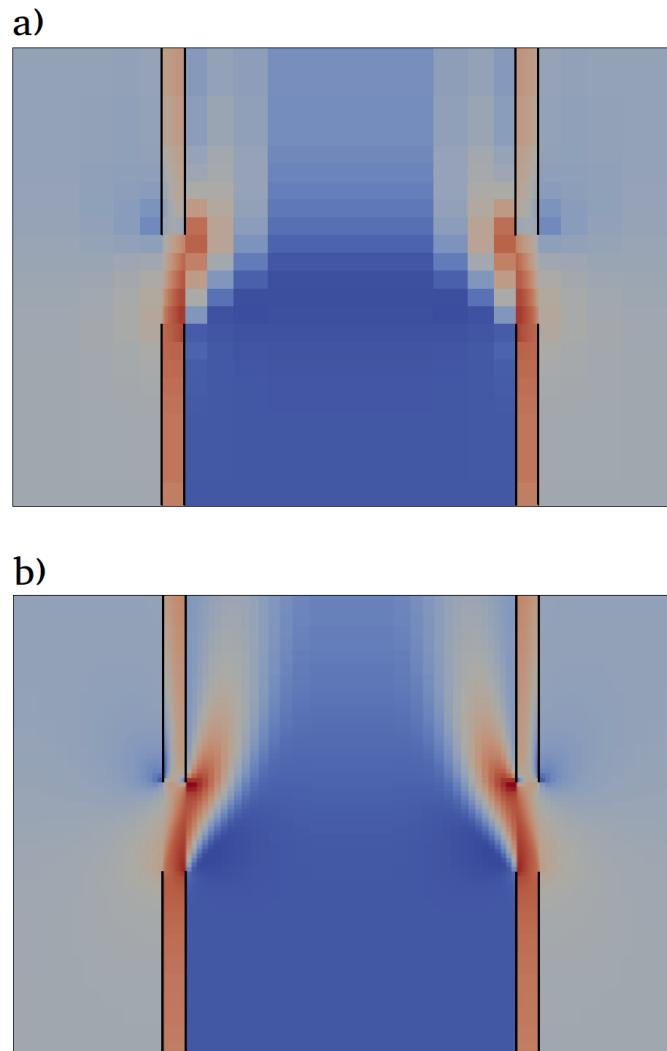


Figure 5.4.2: superficial velocity magnitude distribution at the height of the wrapper windows (black lines represent wrappers) over a 2-D slice parallel to the assembly axis for a high power blockage case: a) for the coarsest mesh; b) for the finest mesh (i.e. number of nodes in all directions increased by a factor 4).

5.5 Assessment of inter-assembly gap, wrapper window behaviour

The investigation was performed parametrically for two scenarios: 1) full power and flow conditions corresponding to nominal operation values ($f_P = 100\%$, $f_N = 100\%$), which will be referred to as “full power” case; 2) low power and flow conditions compatible with core status in natural circulation regime after a protected transient ($f_P = 5\%$, $f_N = 5\%$), which will be referred to as “low power” case. For each of the two scenarios, simulations were performed for different combinations of the central assembly sodium velocity (to model blockage) and inter-assembly gap sodium velocity. Each simulation is performed on all three models, “adiabatic”, “windowless” and “windowed”. To better visualize the thermal-hydraulics of the system, results obtained by the simulation of one of the multiple investigated cases is reported in Figure 5.5.1.

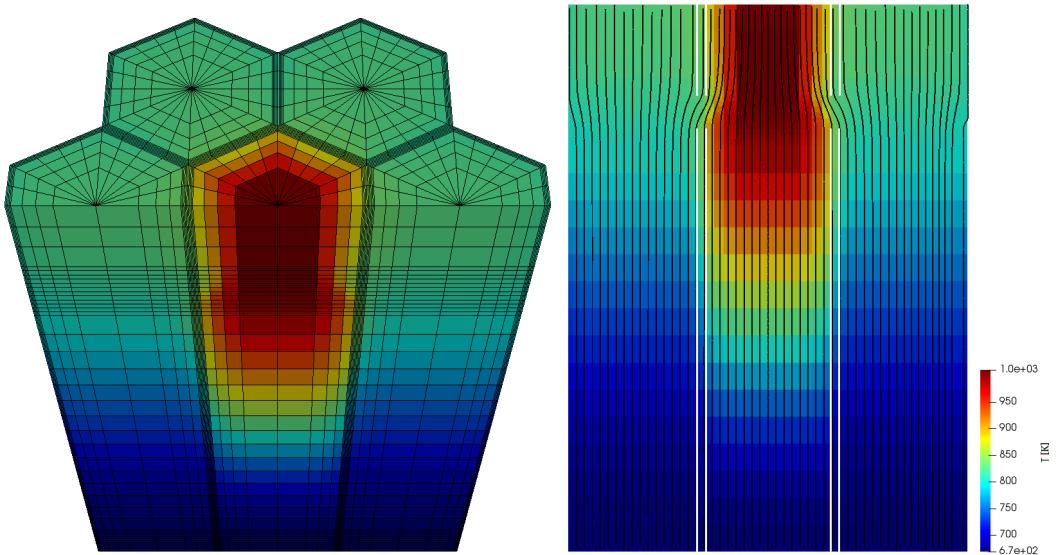


Figure 5.5.1: temperature distribution in the windowed model for $f_P = 5\%$, $f_N = 5\%$, $f_C = 40\%$, $f_G = 20\%$. On the right, the velocity streamlines are reported to highlight sodium leakage from the surrounding assemblies to the gap, and from the gap to the central assembly undergoing partial blockage. The wrappers are highlighted in white on the right slice.

5.5.1 Full power case

A comparison of the central assembly outlet bulk temperatures, for varying magnitudes of the central assembly sodium velocity and gap velocity are presented in Figure 5.5.2. These temperature profiles are reported only for those cases that did not reach boiling.

It can be observed that an increase in the inter-assembly gap sodium velocity leads to an overall decrease in the central assembly outlet bulk temperature, as expected. This dependence is considerably more prominent in the

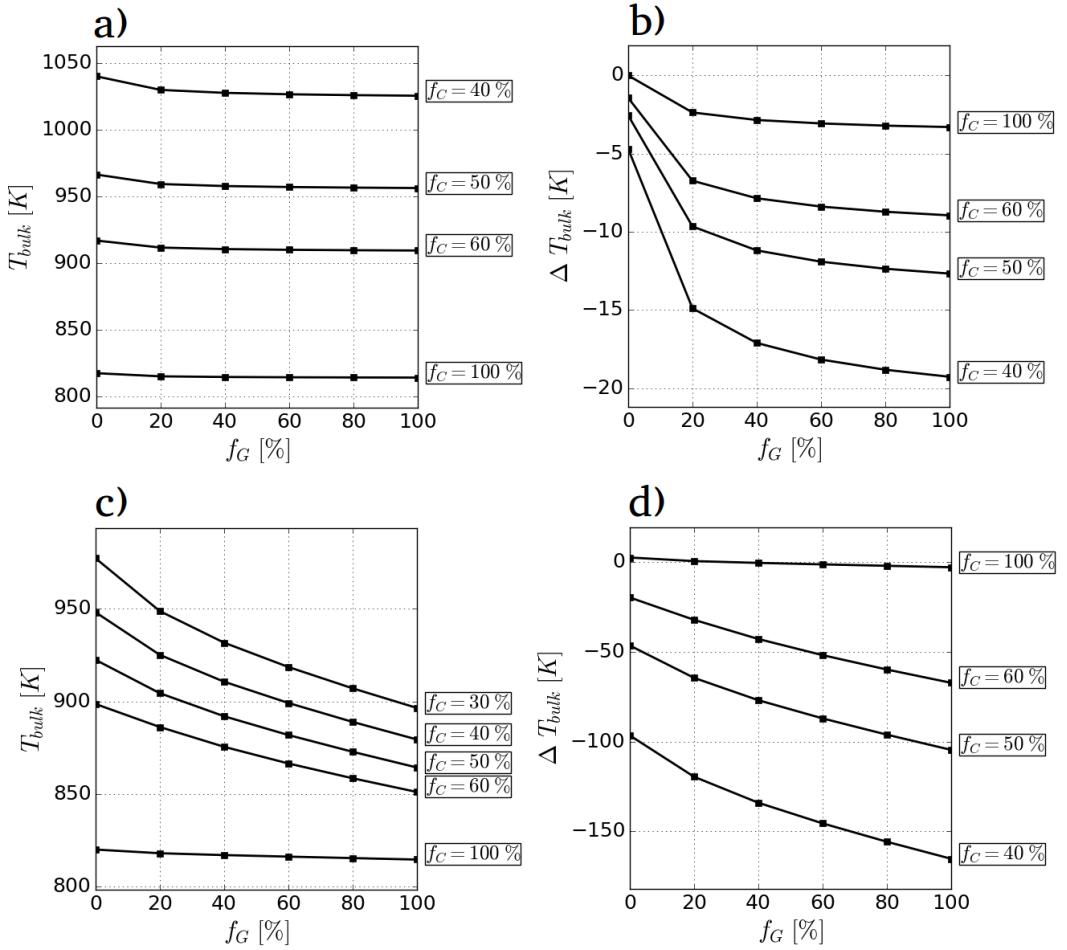


Figure 5.5.2: high power, high flow case: outlet bulk temperature of the central assembly for the: a) windowless model; c) windowed model. Difference between the central assembly temperatures between: b) the windowless and adiabatic models; d) the windowed and adiabatic model. The temperatures are reported against the gap sodium velocity fraction f_G for different central assembly sodium velocities f_C .

windowed (Figure 5.5.2 c) rather than the windowless (Figure 5.5.2 a) model, due to the mixing with the cooler sodium in the inter-assembly gap provided by the windows. This is further strengthened by comparing the differences in outlet bulk temperatures between the windowless and adiabatic (Figure 5.5.2 b) and windowed and adiabatic (Figure 5.5.2 d) models. While these differences are relatively small at nominal operation (i.e. $f_C = 100\%$), they grow progressively larger for lower sodium velocities in the central assembly. This is consistent with the fact that, for a sodium velocity fraction $f_C \leq 60\%$, sodium inflow in the central assembly from the windows is observed. This is observed even for a null inter-assembly gap sodium velocity, meaning that cooler sodium from the neighbouring assemblies leaks into central assembly undergoing blockage via the windows and inter-assembly gap. In particular, it was found that a central assembly sodium velocity fraction $f_C \leq 30\%$ (i.e.

flow blockage $\geq 70\%$) leads to boiling in the windowless model, but not in the windowed model due to the additional cooling windows provide.

The dependence of the central assembly outlet bulk temperature on flow parameters does not necessarily represent the behaviour of the maximum temperature in the central assembly. As a matter of fact, it was found that the maximum central assembly temperature in the windowless model is essentially unaffected by gap flow (i.e. differences $< 0.1 K$). This is not the case for the windowed model, as reported in Figure 5.5.3.

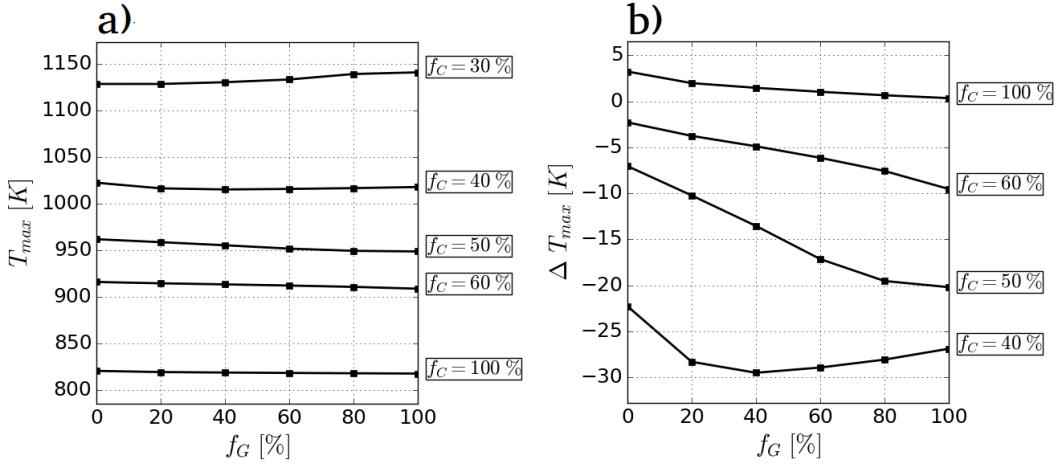


Figure 5.5.3: high power, high flow case: a) maximum temperature in the central assembly of the windowed model; b) difference between the maximum temperatures obtained by the windowed and adiabatic models. These are presented against inter-assembly gap sodium velocity fraction f_G at various central assembly sodium velocity fractions f_C .

While the dependence of the maximum central assembly temperature on gap sodium velocity is weaker than for the outlet bulk temperature (Figure 5.5.3 a), two observations can be made when comparing it against the adiabatic model (Figure 5.5.3 b): 1) at nominal operation ($f_C = 100\%$) the maximum temperature in the windowed model is marginally higher than in the adiabatic model; 2) the behaviour of the maximum temperature with respect to gap velocity is not consistently monotonic.

With regard to the first observation, this is due to the fact that at nominal operation the pressure difference between the gap and assemblies is negative, meaning that sodium outflows from the assemblies into the gap. With increasing gap velocity, the pressure drop over the axial length of the gap increases. The magnitude of the pressure difference between the gap and assemblies at the height of the windows thus decreases, limiting the outflow. If the pressure difference between gap and assembly was to be further increased and become positive by e.g. a decrease in the central assembly velocity, sodium inflow in the assembly is observed instead, leading to a reduction of the maximum temperatures with respect to the adiabatic model. With regard to the second observation, the non-monotonic behaviour of the maximum temperature of the windowed model for $f_C \leq 40\%$ is a consequence of the 3-D nature of

the sodium jet flowing into the central assembly from the windows. For fixed sodium velocity magnitudes at the inlet of the central assembly and the neighbouring assemblies, increasing the gap sodium velocity results in two effects: 1) the pressure difference between gap and central assembly increases, favouring sodium inflow into the central assembly; 2) the pressure in the vicinity of the windows in the central assembly decreases due to the increase of the velocity of the inflowing sodium jet.

With regard to the second observation, this can be clearly observed in Figure 5.5.4, as for high sodium gap velocity (Figure 5.5.3 a), the velocity profile in the bulk of the central assembly deviates towards the windows, while this effect is considerably less prominent at low gap sodium velocity (Figure 5.5.3 b). From the perspective of variations in the maximum temperature in the assembly, these two effects are opposing. Sodium inflow in the assembly will favour the decrease in the maximum temperature. Conversely, a flow widening at the height of the window region will result in a lower axial sodium velocity. Given that the maximum temperature is found to be on the assembly centerline at the height of the windows, this results in an increase of the maximum sodium temperature. This effect becomes more prominent at lower sodium velocities because of the increased velocity difference (and thus pressure difference) between the inflowing sodium jet from the windows and the central assembly sodium flow.

With regard to the time-to-boil, it was found that the gap flow had no effect in appreciably delaying its onset in no case that reached boiling temperatures. This is due to the fact that at full power the heat up process subsequent to a severe blockage (i.e. $f_C \leq 30\%$ in the adiabatic and windowless models, $f_C \leq 20\%$ in the windowed model) is so rapid to be essentially adiabatic and in the order of magnitude of a few seconds. Thus, no results of interest are reported in this regard.

5.5.2 Low power case

A comparison of the central assembly outlet bulk temperatures, for varying magnitudes of central assembly sodium velocity and gap velocity are presented in Figure 5.5.5. As before, these temperature profiles are reported only for those cases that did not reach boiling.

The overall trends are the same as those observed for the full power case in Figure 5.5.2. There is an overall increased in magnitude of the additional cooling power that the windowed (Figure 5.5.5 c) and windowless model (Figure 5.5.5 d) features provide with respect to the adiabatic case. The slope changes observed in the bulk temperature profiles for the windowless case (Figure 5.5.5 a and Figure 5.5.5 b) are associated with the laminar-turbulent transition of the fluid flow in the inter-assembly gap. Outlet bulk temperatures are not reported for $f_C \leq 30\%$ because boiling is reached for both the windowless and windowed models. For the adiabatic model, boiling is reached for $f_C \leq 40\%$. With regard to the maximum temperatures in the system and

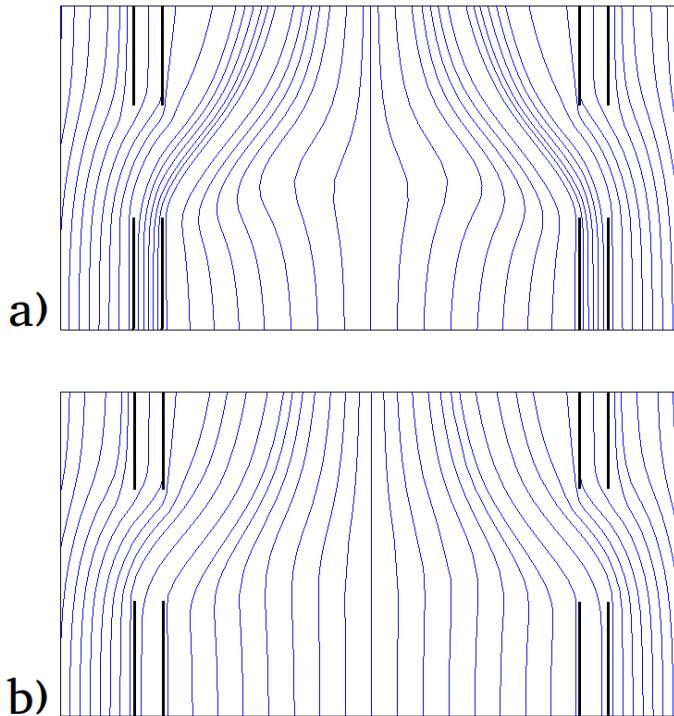


Figure 5.5.4: comparison between the full power, full flow superficial velocity streamlines at $f_C = 30\%$ in the windowed model for two different inter-assembly gap velocities: a) $f_G = 100\%$, b) $f_G = 20\%$. The view consists of a 2-D slice parallel to the main flow direction that encompasses the central assembly, the inter-assembly gap and part of the neighbouring assemblies. The position of the assembly wrappers and thus the windows is highlighted in black.

their dependence on flow parameters, results are presented in Figure 5.5.6. At these low power and flow conditions, the inter-assembly gap is capable of sorting an effect on the maximum temperatures, unlike what was observed for the full power and flow case. This is due to the diffusive effects becoming more prominent as the importance of advective effects on heat transfer decreases at lower sodium velocities.

With regard to the time-to-boil, both the windowless and windowed models were found to perform comparably in term of delaying the onset of boiling with respect to the adiabatic case. For this reason, only time-to-boil results relative to the windowed model and its comparison against the adiabatic model are presented in Figure 5.5.7. The overall boiling delay effects grow smaller in magnitude for lower central assembly sodium velocities (i.e. higher blockages) as expected. On the other hand, inter-assembly gap flow can provide an additional margin to boiling delay.

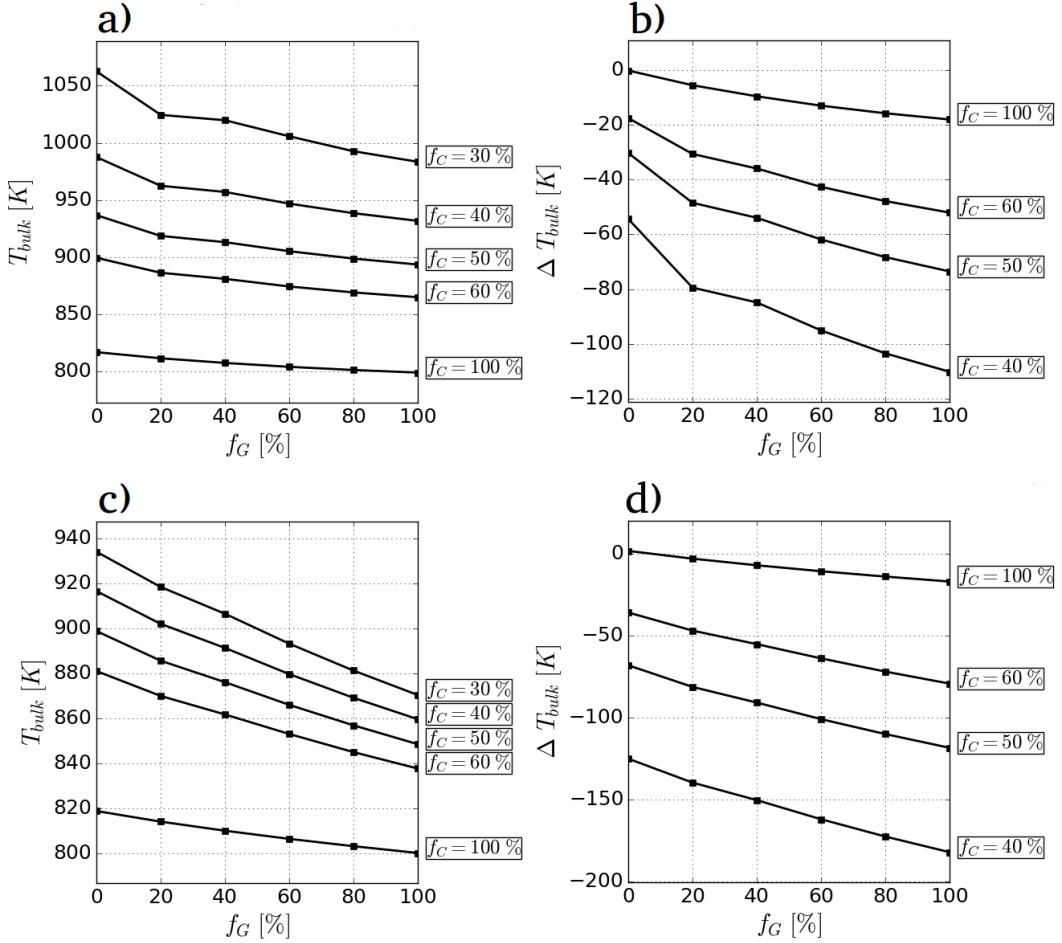


Figure 5.5.5: low power, low flow case: outlet bulk temperature of the central assembly for: a) windowless model; c) windowed model. Difference between the central assembly temperatures between: b) the windowless and adiabatic models; d) the windowed and adiabatic model. The temperatures are reported against the gap sodium velocity fraction f_G for different central assembly sodium velocities f_C .

5.6 Conclusions

This chapter dealt with the further expansion of the methodology and with its application to characterize some features of interest in the recent ESFR core design, namely: the inter-assembly gap and novel assembly wrapper windows. A wrapper heat transfer model was developed and verified against analytical solutions. The developed heat transfer model neglects the thermal inertia of the wrapper, which is acceptable for thin baffles made of a relatively low specific heat capacity material. The localized pressure losses at the windows were modelled as those that would be expected from a 90-degree miter bend of the same hydraulic diameter.

The parametric investigation at two different power and flow levels was performed on three different models: 1) a single assembly (referred to as “adiabatic” model); 2) seven assemblies with thermally conductive wrappers (re-

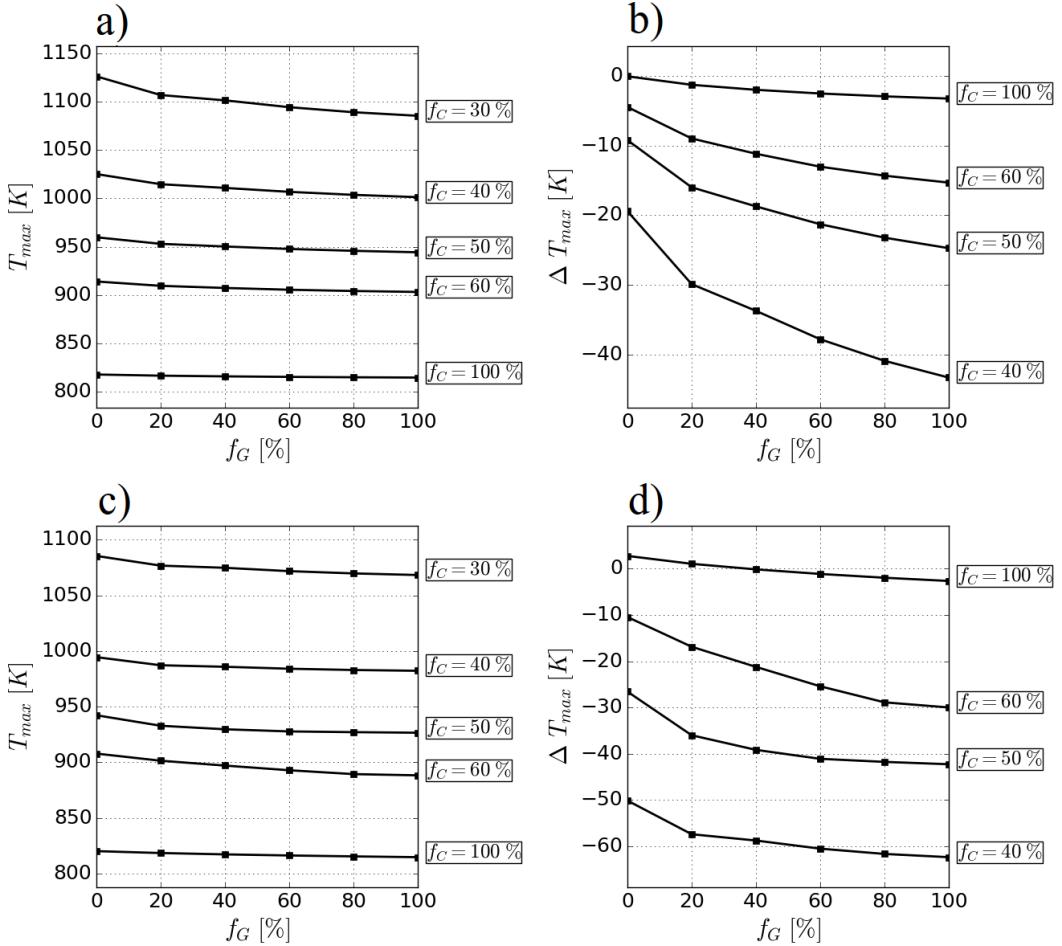


Figure 5.5.6: low power, low flow case: maximum temperature in the central assembly for the: a) windowless model; c) windowed model. Differences of the maximum temperature between: b) the windowless and the adiabatic models; d) the windowed and the adiabatic models. These are presented against inter-assembly gap sodium velocity fraction f_G at various central assembly sodium velocity fractions f_C .

ferred to as “windowless” model); 3) seven assemblies with thermally conductive wrappers and windows (referred to as “windowed” model).

The most important results are hereby summarised. 1) At reference power and flow conditions, sodium outflow from the assemblies through the windows is observed. This results in a small assembly outlet bulk temperature increase $\leq 5 K$ depending on the inter-assembly gap sodium velocity. 2) For any investigated central assembly sodium velocity below nominal values, sodium inflow in the assembly through the wrapper windows is observed. This was always found to reduce both the assembly outlet temperatures and the maximum assembly temperature by appreciable quantities with respect to the thermally insulated, windowless wrappers. 3) The change in maximum assembly coolant temperature with respect to inter-assembly gap sodium velocity is not always monotonic in the presence of wrapper windows. This was found to be due

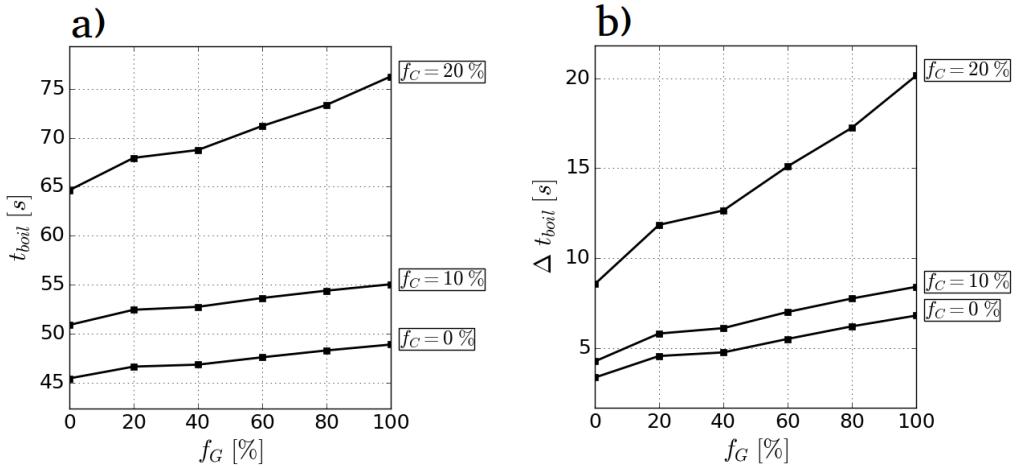


Figure 5.5.7: low power, low flow case: a) time-to-boil of the windowed model; b) difference in time-to-boil between the windowed and adiabatic models. These are presented against the inter-assembly gap sodium velocity fraction f_G for various central assembly sodium velocity fractions f_C .

specifically to 3-D effects of the inflowing sodium jet from the wrapper windows. 4) Inter-assembly gap flow in the absences of wrapper windows was found to be less effective than the windowed wrapper in lowering outlet bulk temperatures for varying magnitudes of the inter-assembly gap sodium velocity. Furthermore, it was found to have no effect at all on the maximum assembly temperatures at nominal flow conditions due to the dominance of advective effects compared to diffusive effects. 5) It was found that inter-assembly gap flow, with or without windows, is capable of preventing boiling for some flow conditions at nominal power levels when compared to the thermally insulated wrapper case.

This chapter also tackled the issue of mesh convergence in coarse-mesh applications. A coarse-mesh has limited room for improvement in terms of mesh independence of the results when compared to traditional fine mesh approaches. For this reason, a suitable mesh convergence methodology was proposed to assess the most suitable coarse mesh for the investigated cases and to establish the expected uncertainty bands. The chosen meshes satisfied the requirements of providing outlet bulk temperatures in the asymptotic convergence range. The estimated uncertainty bands on the outlet bulk temperatures varied in the $1 K - 8 K$ range depending on the case.

Chapter 6

Multi-physics investigation of FFTF LOFWOS Test 13

6.1 Introduction

The thermal-hydraulics methodology and its implementation that have been presented so far have been integrated in the GeN-Foam multi-physics code. This integration process also saw further developments in the overall coupling algorithm with the other physics, namely neutronics and thermal-mechanics. It is recalled that a coarse-mesh approach carries the advantage of a straightforward coupling with other physics owing to the possibility of employing standardized field manipulation and mesh projection algorithms.

An excellent opportunity to test the multi-physics capabilities of the code was identified in the International Atomic Energy Agency (IAEA) coordinated research project (CRP) on the benchmark re-analysis of Loss Of Flow Without SCRAM (LOFWOS) Test 13 [89] performed at the Fast Flux Test Facility (FFTF), a former experimental Sodium-cooled Fast Reactor (SFR). The test consisted of a rapid Unprotected Loss Of Flow (ULOF) transient to assess the effectiveness of a novel passive safety feature, namely the so-called Gas Expansion Modules (GEMs) [90]. Although this transient did not result in sodium boiling, it allowed to test the multi-physics capabilities of the code. First, the steady-state reactor conditions were obtained via a one-way coupling between diffusion-based neutronics and one-phase thermal-hydraulics while the transient simulation relied on a tight coupling between the one-phase thermal-hydraulics and a point kinetics model, with the flux shape resulting from the diffusion calculation.

Section 6.2 discusses the integration of the developed thermal-hydraulic methodology within the multi-physics framework of GeN-Foam. It discusses the global coupling algorithm and introduces the capabilities of the two other sub-solvers for neutronics and thermal-mechanics. Section 6.3 provides an overview of the FFTF reactor and of the transient under investigation. Section 6.4 discusses how the system of interest has been modelled and simulated

with a coarse-mesh approach, covering in particular two aspects: 1) thermal-hydraulics modelling with emphasis of novel developments such as intermediate heat exchangers (IHXs) modelling; and 2) neutronics modelling for both the steady-state diffusion-based calculations as well as the point kinetics model inclusive of the effects of the GEMs. Section 6.5 presents the results of the multi-physics investigation and compares them against available experimental data, with an additional parametric investigation for parameters of relevance. Section 6.6 draws the main conclusions of the chapter.

6.2 Multi-physics framework

The integration of the developed thermal-hydraulic solver in GeN-Foam was simplified by the underlying shared OpenFOAM framework and the way in which sub-solvers are organized, namely encapsulated in individual objects with the coupling fields managed by the top-level algorithm.

The present section will discuss in greater detail the multi-physics capabilities as a whole. Sub-section 6.2.1 presents the scope of each individual physics and its inter-dependency on the other physics from the perspective of the coupling fields. Sub-section 6.2.2 presents the overall solution algorithm.

6.2.1 Single-physics capabilities and coupling fields

Each of the three individual physics, namely, thermal-hydraulics, neutronics and thermal-mechanics, is treated by a sub-solver that relies on the run-time selection mechanism of OpenFOAM. This allows to select the desired treatment for each physics without code re-compilation. This also enables to individually control which single-physics should be solved and which should not, enabling a variety of investigations: purely thermal-hydraulics, purely neutronics, coupled thermal-hydraulics and neutronics, et cetera.

From a thermal-hydraulics perspective, the available treatments consist of the already extensively discussed one-phase and two-phase approaches. From a neutronics perspective, these consist of diffusion [91], discrete ordinates (S_N) [92], SP₃ [93] which can be used in either a transient or eigenvalue (i.e. steady-state) mode. Within the scope of the present work, point kinetics was developed and integrated as well, for reasons discussed later. From a thermal-mechanics perspective, the only treatment presently available consists of a linear-elastic model for core structure mechanics at a coarse scale, combined with a model for fuel axial expansion [20].

Each of the individual physics can operate on an individual computational mesh to resolve different levels of geometric detail or to restrict the investigation of each single-physics to specifics portions of the overall domain of interest [20]. The passing of coupling fields between the physics is enabled by the mesh-to-mesh projection operations that are supported by the underlying OpenFOAM framework. The single-physics sub-solvers of GeN-Foam and

their inter-dependencies are presented in Figure 6.2.1. Each of these single-physics is now discussed from the perspective of its interaction with the other physics.

Thermal-hydraulics The thermal-hydraulics sub-solver consists of the methodology and software that have been extensively discussed in the present work. In a multi-physics context, it can be viewed as a sub-solver that operates on a certain computational domain Ω_{TH} . Given an input volumetric power density q , the thermal-hydraulics sub-solver is tasked with predicting the resulting fluid temperature T , density ρ and velocity \mathbf{u} fields, as well as relevant structure temperature fields, collectively denoted with T_s , for coupling purposes. For a two-phase treatment, the fields ρ , T , \mathbf{u} consist of mass-weighted mixture values. The velocity field \mathbf{u} is used only when simulating Molten Salt Reactors (MSRs) to advect the precursors. The field q is the volumetric fuel power density and it can pertain either to the structure (i.e. $q_{int,s}$ in equations 3.2.41, 3.2.42 depending on the structure thermal model, e.g. the volumetric fuel power density of a fuel pin) or the fluid (i.e. the internal liquid fuel power density $q_{int,i}$ when simulating MSRs), depending on the system under investigation. The field T_s collectively denotes the temperature fields of the structure, which can range from the fuel and cladding temperatures of a nuclear fuel pin or an average structure temperature representing e.g. control rod drivelines, wrappers, the diagrid, et cetera. This entirely depends on what the selected structure thermal models are supposed to represent in the mesh cells where they have been defined, as discussed in sub-section 3.2.4.

Neutronics The neutronics sub-solver operates on a computational domain Ω_N and is tasked with predicting the volumetric fuel power density q for varying coupling fields. Not all of these fields are always used, depending on the selected type of neutronics treatment. In general terms, the diffusion, S_N , SP_3 treatments are capable of modelling reactivity feedbacks from: coolant temperature T and density ρ , average fuel and cladding temperatures collectively denoted with T_s , fuel axial displacement and core radial displacement collectively denoted as \mathbf{d} . As long as a parametrization of the macroscopic cross sections against these quantities is provided (typically from Monte Carlo calculations), these feedbacks can be resolved. The feedbacks of the point kinetics solver will be discussed separately in the modelling section 3.4, as they have been developed in the frame of this thesis work.

Thermal-mechanics The thermal-mechanics sub-solver operates on a computational domain Ω_{TM} and is tasked with predicting an overall displacement field that is used to deform all the computational domains Ω_{TH} , Ω_N , Ω_{TM} . The displacement is the one that arises due to thermal gradients of the various structure temperature fields T_s of interest. The displacement field is further decomposed into a fuel axial displacement and core radial displacement col-

lectively denoted as d , which are passed to the neutronics to model feedbacks from possible parametrizations.

While the thermal-mechanics sub-solver has enjoyed a range of applications up to the most recent preliminary modelling of inter-assembly contact and its effect on core thermal-hydraulics in SFRs [94], it was not employed in the present analysis.

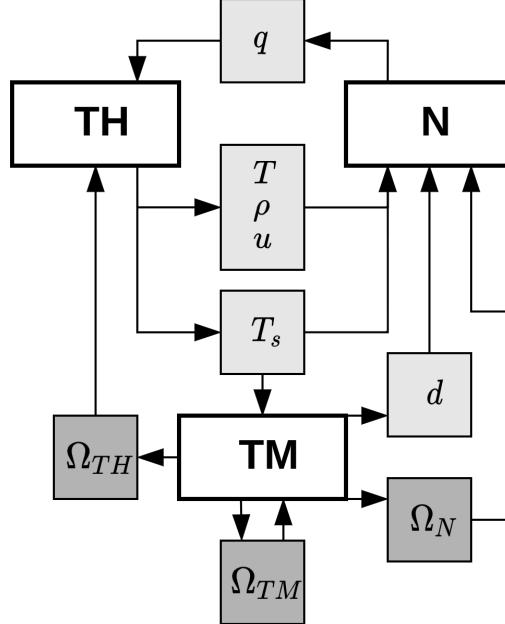


Figure 6.2.1: relationship between the single-physics sub-solvers of GeN-Foam, namely thermal-hydraulics (TH), neutronics (N) and thermal-mechanics (TM).

6.2.2 Multi-physics coupling algorithm

The overall coupling algorithm is presented in Figure 6.2.2. Within each time step:

1. **Outer iteration loop.** This loop is used to couple the single-physics solution steps it encompasses. It can be performed a user-selected number of times or controlled by the convergence of the residuals of the slowest-converging physics;
 - 1.1. **solve fluid-mechanics.** This step consists of the collection of steps `regimeMap` through `pUEqns` as discussed for the single-physics thermal-hydraulics algorithm presented in Figure 3.3.1;
 - 1.2. **solve fluid and structure enthalpy.** This step consists of the collection of steps `EEqns` as discussed for the single-physics thermal-hydraulics algorithm presented in Figure 3.3.1;

- 1.3. **interpolate thermal-hydraulics fields to thermal-mechanics mesh.** This step projects the coupling fields T_s discussed earlier to the thermal-mechanics mesh;
- 1.4. **solve thermal-mechanics.** Solve structure expansion and predict displacement fields \mathbf{d} , also used to deform the computational meshes;
- 1.5. **interpolate displacement field to all meshes and deform meshes;**
- 1.6. **interpolate thermal hydraulic-fields to neutronics mesh.** This step projects the coupling fields ρ , T , \mathbf{u} discussed earlier to the thermal-mechanics mesh;
- 1.7. **solve neutronics.** This steps solves the equations representative of the selected neutronics model. From a coupling perspective, the new fuel power density q is obtained;
- 1.8. **interpolate fuel power density to thermal-hydraulics mesh;**

This formally consists of a Picard iteration approach [2]. Please note that steps pertaining to any single-physics are performed depending on the coupling to be resolved. For the investigation discussed in this chapter, the effects of thermal-mechanics on core neutronics were captured via simplified models built into the point kinetics neutronics model.

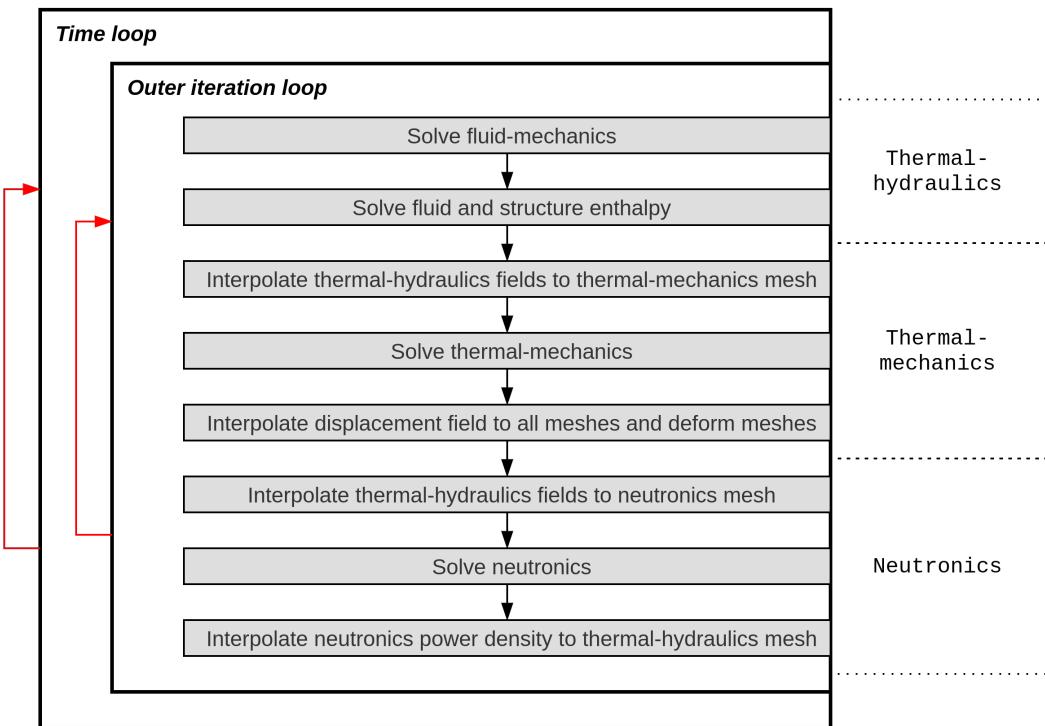


Figure 6.2.2: multi-physics coupling algorithm of the GeN-Foam code.

6.3 Description of the Fast Flux Test Facility and LOFWOS Test 13

6.3.1 The Fast Flux Test Facility

The FFTF was a 400 MW_{th} experimental SFR located at the Hanford Site, Washington, USA. It was operated between 1982 and 1992 to test advanced nuclear fuels, materials and components to support the future operation of then-planned commercial fast reactors, with particular emphasis on the investigation of passive safety features, some of which innovative [90][95].

The reactor design consists of a hybrid pool-loop type, wherein the core was accommodated in a pool within the reactor vessel while the IHXs and pumps are located outside the pool. The plant is schematically represented in Figure 6.3.1 while the vessel is represented in Figure 6.3.2. The plant did not produce electricity and its thermal output was ultimately discharged to the atmosphere via so-called dump heat exchangers (DHXs) on the secondary loop. The most relevant reactor parameters are reported in Table 6.3.1. A comprehensive technical description of the FFTF circuits, vessel, core, assemblies and fuel can be found in [96] and is not reported for brevity.

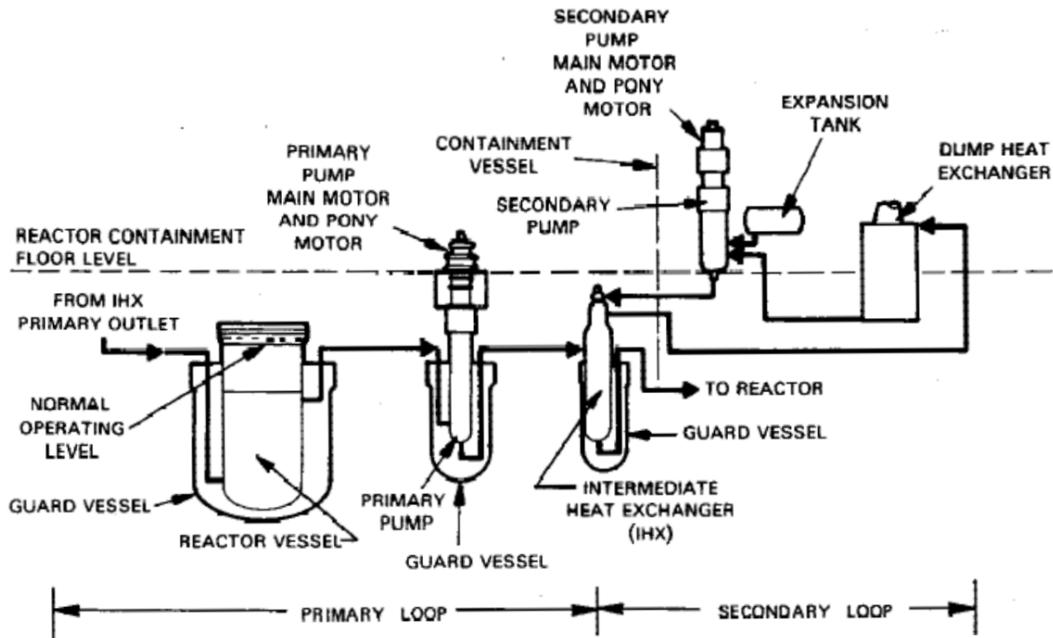


Figure 6.3.1: overview of the FFTF components and loops. Courtesy of Cabell [96].

Quantity	Value	Quantity	Value
Thermal power (MW)	400	No. of fuel assemblies (-)	
Core volumetric flow rate (m^3/s)	2.74	- inner core	32
Core outlet temperature (K)	799.9	- outer core	48
Prim. cold leg temperature (K)	633.2	No. f pins per assembly (-)	217
Prim. mass flow per loop (kg/s)	730.5	Assembly pitch (cm)	12.014
No. of primary loops (-)	3	Wrapper material	SS-316
Sec. hot leg temperature (K)	729.3	Wrapper inner flat-to-flat (cm)	11.621
Sec. cold leg temperature (K)	586.0	Wrapper thickness (mm)	3.050
Sec. mass flow per loop (kg/s)	730.5	Fuel	MOX
No. of secondary loops (-)	3	Pu content (%), <i>wt.</i>)	
		- inner core	24.06
		- outer core	28.52
		Pellet diameter (mm)	4.940
		Pellet column height (cm)	91.44
		Cladding material	SS-316
		Cladding inner diameter (mm)	5.080
		Cladding outer diameter (mm)	5.842
		Spacer wire material	SS-316
		Spacer wire diameter (mm)	1.445
		Sparer wire lead length (cm)	30.2

Table 6.3.1: Reference FFTF data at nominal reactor conditions (left), together with assembly, fuel data for LOFWOS Test 13 (right).

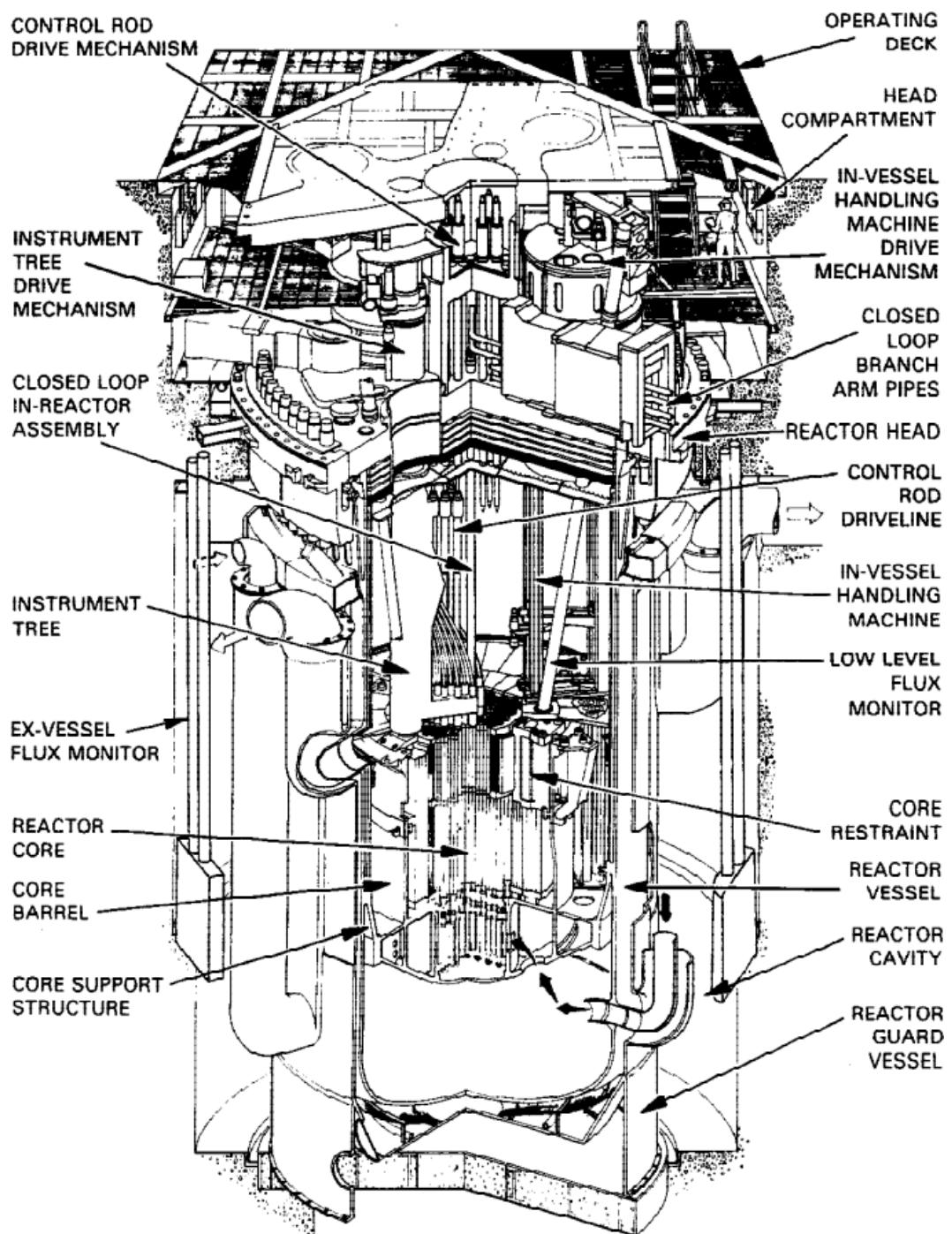


Figure 6.3.2: overview of the FFTF reactor vessel. Courtesy of Cabell [96].

6.3.2 LOFWOS Test 13 and safety features demonstration

The LOFWOS series of test consisted in ULOF transients aimed at determining safety margins, demonstrate the benefits of certain design features and obtain data for the validation of computational tools.

Among the investigated safety features, of particular interest were the GEMs. These consisted of empty wrapper cans (i.e. containing no fuel pins) closed at the top and filled with argon gas, placed in the periphery of the active core. A simplified representation of their intended operational principle is reported in Figure 6.3.3.

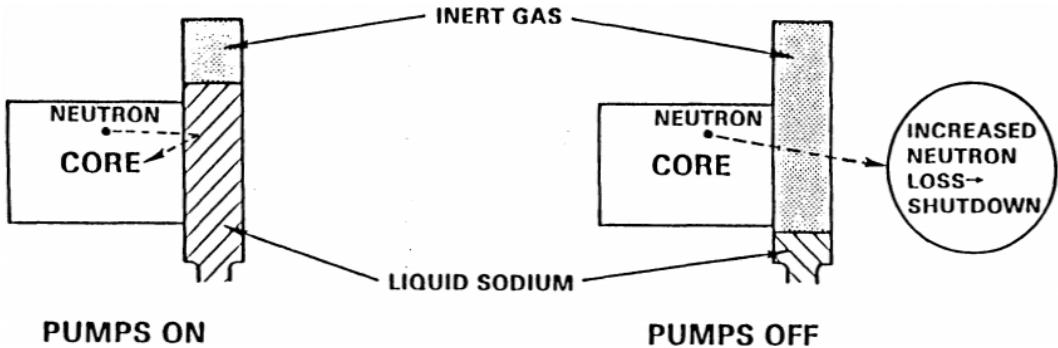


Figure 6.3.3: Overview of the operating principle of the GEMs. Courtesy of Cabell [96].

In regular operation, the pressure head provided by the pumps compressed the argon gas within the GEMs, so that the sodium free surface (within the GEMs) was found above top of the active fuel region, reducing radial neutron leakage. If this pressure head was lost for any reason, e.g. in an ULOF transient, the subsequent downward expansion of the argon gas would lower the sodium free surface level below the bottom of the active fuel. From a neutronics perspective, this “uncovers” the core radially, increasing neutron leakage thus providing a large amount of negative reactivity. This safety feature proved particularly effective (up to $-1.41\text{ \$}$ of reactivity worth for the core configuration of LOFWOS Test 13, for a total of 9 GEMs) due to the small radial dimension of the FFTF active core, $\approx 1.2\text{ m}$ in diameter.

Test 13 proceeded as follows. Starting from a steady-state at half nominal thermal power and full nominal primary and secondary mass flows, the primary pumps were tripped, halving the primary mass flow rates within the first 7.2 s and coasting down still in 90 s since the start of the transient. The evolution of relevant core temperatures and mass flow was fundamentally governed by the interplay of neutronics feedbacks with the natural circulation regime that was established as the pumps were coasting down. These variables of interest were measured at a variety of locations throughout the transient, and represent the data against which the results obtained by the code are compared, in section 6.5.

6.4 Computational modelling

The transient was investigated from the perspective of thermal-hydraulics and neutronics coupling, and required the modelling of the steady-state preceding the transient as well.

The steady-state was investigated in two separate steps: 1) an eigenvalue diffusion-based neutronics calculation to predict the neutron fluxes and the fuel power density distribution in the core; 2) a steady-state thermal-hydraulics calculation to predict the initial conditions for the transient stemming from the fuel power density distribution. The macroscopic cross sections for the neutronics were obtained by the Serpent2 Monte Carlo code [97] from previous purely neutronics investigations of the FFTF core [98]. However, the parametrizations of the cross section with respect to the feedback fields of interest were not obtained. This, coupled with availability of point kinetics feedback coefficients (provided by the Argonne National Laboratory (ANL)) and the difficulty of resolving the GEMs feedback with a diffusion-based approach, led to employing a point kinetics model for the simulation of the transient.

Sub-section 6.4.1 presents the thermal-hydraulic modelling, ranging from geometric modelling and choice of correlations for the closure of momentum and heat transfer as well as the modelling of the thermal coupling between primary and secondary circuits via the IHXs. Sub-section 6.4.2 presents the neutronics modelling and the implementation of the point kinetics model.

6.4.1 Thermal-hydraulics modelling

6.4.1.1 Geometry and structure modelling

The FFTF was modelled via a 2-D hybrid computational domain. The hybrid nature comes from the use of a wedge for the modelling of the vessel, and parallelopipeds for the modelling of the primary and secondary pipes, IHXs and pumps. The thermal-hydraulics computational domain is presented in Figure 6.4.1, with its coarse-mesh properties reported in Table 6.4.2. Note that the primary and secondary loops are geometrically disconnected, but the heat exchange between the two is modelled via the thermally coupled IHX regions, as it will be discussed later.

With reference to said Figure and Table, the most important modelling choices and approximations that were made are hereby introduced for each component of interest.

Reactor vessel and core The vessel consists of regions 1 through 15. It is modelled as a 2-D wedge of two degrees of aperture. The reactor core consists of regions 1 through 11, and it was obtained by averaging the radial heterogeneity of the actual 3-D core layout in a manner that preserves the overall volume of each region. The coarse-mesh properties of regions such as the bypass (region 13), reflector (regions 8, 9, 10), radial shield (region 11), GEM

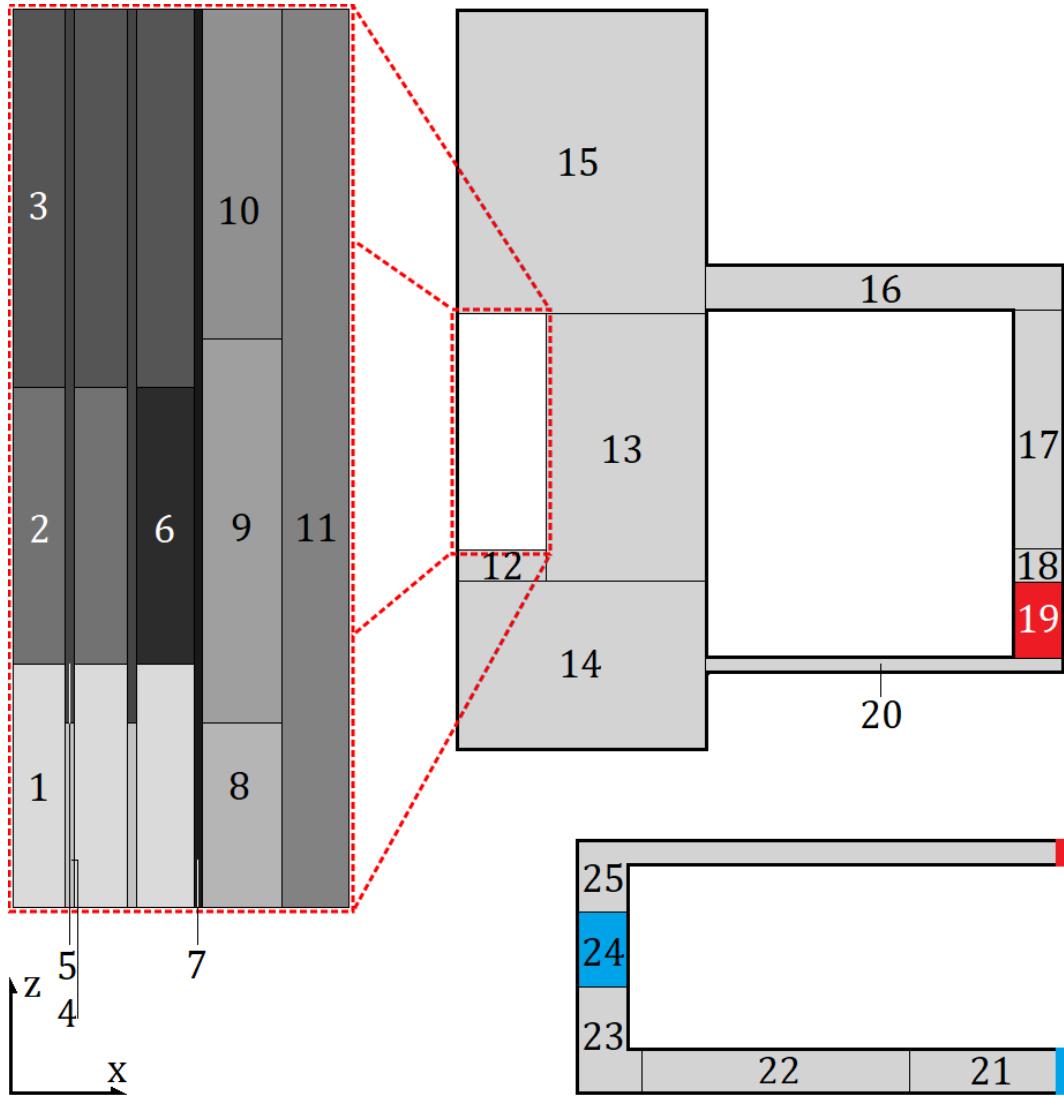


Figure 6.4.1: computational domain of the FFTF thermal-hydraulics. It consists of a 2-D hybrid wedge-parallelopipedal model. Regions 1 through 20 represent the primary, while regions 21 through 25 represent the secondary. The primary and secondary loops are geometrically disconnected domains that are thermally coupled via the IHX. The primary side of the IHX consists of region 19 (red) while the secondary side consists region 24 (blue). Open system boundaries consist of the inlet (blue segment) and outlet (red segment) of the secondary loop. Coarse-mesh region properties are reported in table 6.4.2.

(region 7) were adjusted so to ensure the correct flow redistribution through the core. This is justified by the significant geometric complexity of these components (some of which consist e.g. of alternating layers of plate orifices of varying geometries), so that some coarse-mesh properties, most importantly the hydraulic diameter D_h (on which friction factors depend), became a model parameter in those regions rather than a consequence of the volume-averaging process itself.

From a thermal perspective, the behaviour of the unresolved structures is

modelled only in the inner and outer fuel (regions 2, 6), upper and lower fuel shields (regions 1, 3) and the two IHX sides, (regions 19 and 24). The IHX modelling is discussed later in this sub-section.

The thermal modelling of those core regions is also meant to provide the point kinetics with temperatures to compute feedback effects, as it will be discussed later. With regard to Table 6.4.2, two thermal models are specified in both the inner and outer fuel regions, namely a 0-D structure thermal model and a 1-D nuclear fuel pin model. Within each inner or outer fuel region mesh cell, the 0-D model represents the thermal behaviour of the non-power-producing structures (i.e. spacer wire and assembly wrappers) according to equation 3.2.41, while the 1-D nuclear fuel pin model represents the thermal behaviour of the fuel and cladding according to what was discussed in subsection 3.2.4.

It should be noted that each thermal model t can be prescribed its own volumetric surface area density $A''_{s,t}$, which governs heat transfer with the fluid, and phase fraction $\alpha_{s,t}$, which quantifies what is the actual volume fraction of the mesh cell that is occupied by that specific thermal structure compared to the overall structure in said mesh cell. As an example, Table 6.4.2 reports that the overall sub-scale structure (i.e. the collection of pins, spacer wires, wrappers) occupies a fraction $\alpha_s = 0.590$ of the inner and outer fuel regions, while the spacer wires and wrappers themselves only occupy a fraction $\alpha_{s,t} = 0.036$ of it. This is relevant as $\alpha_{s,t}$ is the actual α_s (if specified) that figures in the 0-D model energy equation 3.2.41. Conversely, the governing 1-D nuclear fuel pin equation 3.2.42 does not contain any information on the structure volume fraction α_s , and the energy balance depends uniquely on the pin surface (i.e. cladding) area density $A''_{s,t}$. For further details, the topic of a 1-D nuclear fuel pin model integration is further expanded in appendix A.

The thermo-physical properties of the wrappers and spacer wires required for solving 3.2.41 consist of that of stainless steel SS-316, as that was the material used for said components. The properties of the 1-D nuclear fuel pin model consist of the data presented in Table 6.3.1 and were the same between inner and outer fuel. Additional thermo-physical properties for the modelling are reported in Table 6.4.1 for completeness.

The effects of changing linear fuel power (i.e. pellet heating or cooling) on the gap conductance (due to the change in gap width) were modelled via the correlation by Lavarenne [99], developed for MOX SFR fuel. The correlation also depends on fuel burnup. While the exact irradiation and shuffling history of the fuel prior to LOFWOS Test 13 was not available, the estimated average burnup was $\approx 50 \text{ GWd}/t_{HM}$. Nonetheless, the effect of varying fuel burnup on the gap conductance correlation was found significant enough on the steady state and transient results to motivate a parametric analysis.

All the different core regions, namely regions 1 through 11 are physically separated from other radially adjacent regions via vertical baffles, as discussed in sub-section 5.3.2.

Quantity	Value
Fuel heat capacity ($J/kg/K$)	250
Fuel thermal conductivity ($W/m/K$)	3
Cladding heat capacity ($J/kg/K$)	500
Cladding thermal conductivity ($W/m/K$)	20
Gap conductance ($W/m^2/K$)	$(300 + 1 \cdot 10^{-1} q)(1 + \tanh(2 \cdot 10^{-2} B))$

Table 6.4.1: FFTF fuel modelling data. The gap conductance model is the one proposed for SFR MOX fuel by Lavarenne [99], wherein q is the pin linear power in W/m and B is the fuel burn-up in GWd/t_{HM} .

Please note that no specific tortuosity modelling was performed, i.e. the tortuosity tensor is an identity tensor in the entirety of the domain. Due to the axially-extruded geometry of the pins, the tortuosity would only affect heat and momentum diffusive processes in the radial direction within the core. However, these were deemed of little importance for the investigation of the transient at hand.

Primary loop The primary loop consists of regions 16 through 20, modelled as 2-D parallelopipes. While the FFTF had three primary loops, these were modelled as a single loop with the same total volume, scaled by the vessel wedge size. In practical terms, the ratio of the 2-D primary loop volume to the sum of the real volumes of the three primary loops is 2/360, which is the same as the ratio of the 2-D vessel wedge (two degrees in aperture) model volume to the real vessel volume.

While preserving piping volume preserves fluid circulation time through the piping, the same does not necessarily hold for the vessel, as the overall primary circuitry re-circulation time will depend on fluid stagnation zones, which significantly approximated in 2-D. Another consideration that needs to be made is that the primary loop shape, length and elevation changes were significantly simplified, which prevents a meaningful estimation of the pressure drops in the primary piping. However, as most of the system frictional pressure drops take place in the reactor core, this was deemed acceptable, also in light of the way the primary pump (region 17) is modelled.

An additional time-dependent momentum source field \mathbf{M}_{pump} is in fact defined exclusively over said region and is added to the source term \mathbf{S}_u of the momentum equation 3.3.6, as seen in chapter 3. This momentum source was obtained iteratively so to reproduce the experimental mass flow rate ramp down in the primary loop for the modelling of the transient. For this reason, an accurate modelling of the primary loop pressure drops was deemed of secondary importance. An additional primary loop component of relevance is the IHX, more particularly the primary side of the IHX (region 19), which allows to thermally couple the two otherwise geometrically disconnected loops.

Secondary loop The three secondary loops were modelled based on the same considerations made for the primary loops. The secondary loops are not

however modelled in their entirety. These extend from the outlet of the DHX, which constitute the inlet of the modelled secondary loop (blue segment in Figure 6.4.1) to the inlet of the DHX (red segment in Figure 6.4.1). This was done as boundary conditions for the secondary loop were provided at these locations based on experimental data.

The secondary pump was modelled for completeness (to preserve circulation time within the modelled portion of the secondary loop) but no momentum source was defined there, as the mass flow through the secondary is imposed via velocity boundary conditions at the loop inlet. The secondary loop also contains the secondary side of the IHX (region 24), which allows to thermally couple the two otherwise geometrically disconnected loops.

Region		α_s (-)	D_h (m)	Thermal structure modelling		
No.	Name			Type	$\alpha_{s,t}$ (-)	$A''_{s,t}$ ($\frac{m^2}{m^3}$)
1	Lower fuel shield	0.867	0.0112	0-D	0.867	27.56
2	Inner fuel	0.590	0.0032	0-D	0.036	100.93
3	Upper fuel shield	0.590	0.0032	1-D NFP	N/A	379.30
4	Lower CR shield	0.950	0.0019	0-D	0.036	100.93
5	Control rod (CR)	0.630	0.0051			
6	Outer fuel	0.590	0.0032	0-D	0.036	100.93
7	GEM	0.200	$1 \cdot 10^{-6}$	1-D NFP	N/A	379.30
8	Lower ref. shield	0.971	0.0002			
9	Reflector	0.928	0.0031			
10	Upper ref. shield	0.928	0.0005			
11	Radial shield	0.945	0.0002			
12	Diagrid	0.590	0.005			
13	Bypass	0.996	$6 \cdot 10^{-4}$			
14	Lower plenum	0	6.29			
15	Upper plenum	0	6.29			
16	Primary hot leg	0	0.6922			
17	Primary pump	0	0.0032			
18	Primary junction	0	0.3874			
19	IHX (primary side)	0	0.0011	IHX	N/A	106.67
20	Primary cold leg	0	0.3874			
21	Secondary cold leg	0	0.3874			
22	Secondary pump	0	0.3874			
23	Secondary junction	0	0.3874			
24	IHX (secondary side)	0	0.0011	IHX	N/A	106.67
25	Secondary hot leg	0	0.3874			

Table 6.4.2: Coarse-mesh properties of the thermal-hydraulics 2-D domain of the FFTF as represented in Figure 6.4.1.

IHX modelling The three IHXs (one per primary loop) were modelled as a single IHX whose sides are split between the primary loop (region 19) and secondary loop (region 24).

From a thermal perspective, a novel structure model was developed which

allows to treat the fluid heat transfer between the two loops, not dissimilar from the wrapper heat transfer modelling discussed in sub-section 5.2.1. While the modelling presented hereby is fundamentally applicable to any IHX design, let us consider the FFTF IHX design, namely a shell and tube heat exchanger. In such a design, the primary circuit coolant flows through a shell that hosts a tube bundle, while the secondary circuitry coolant flows through said tube bundle, with the heat transfer taking place through the tube walls. A volume Ω is superimposed over a possible detailed geometry of such a configuration in the center of Figure 6.4.2.

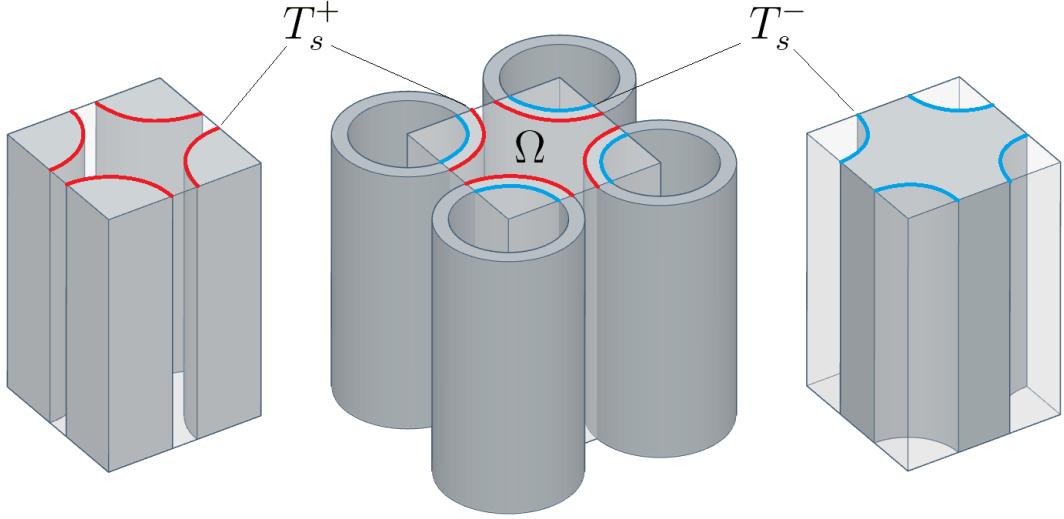


Figure 6.4.2: middle: representation of the unresolved detailed geometry of an IHX, which is partly covered by a tentative computational mesh cell Ω . Tube surface temperatures are referred to as T_s^+ on the outer tube side (corresponding e.g. to the FFTF primary side) and T_s^- on the tube inner side (corresponding e.g. to the FFTF secondary side). Left: coarse-mesh representation of the IHX primary side (i.e. a cell of region 19 in Figure 6.4.1). Right: coarse-mesh representation of the IHX secondary side (i.e. a cell of region 24 in Figure 6.4.1).

Within said volume Ω : 1) the average fluid temperature is denoted with T^+ on the primary side (i.e. outside the tube bundle), and with T^- on the secondary side (i.e. inside the tube bundle); 2) the average tube surface temperature is denoted with T_s^+ on the primary side (on the red tube surfaces), and with T_s^- on the secondary side (on the blue tube surfaces); 3) the average fluid-tube surface heat transfer coefficients are denoted with H^+ on the primary side and H^- on the secondary side, with the tube wall heat transfer coefficient denoted with H_w .

Assuming that: 1) the tube wall is thin enough so that the surface area of the inner and outer tube wall is the same within the considered volume; 2) the absolute heat capacity of the wall compared to the fluid is small enough to be neglected; then an instantaneous heat flux balance holds between the primary and secondary loops, so that the same relationship between the inner and outer tube surface temperatures and fluid temperatures that were obtained for the

wrapper heat transfer model hold, namely equations 5.2.1, 5.2.2:

$$\frac{1}{\frac{1}{H_w} + \frac{1}{H^+}}(T^+ - T_s^-) = H^-(T_s^- - T^-) \quad (6.4.1)$$

$$\frac{1}{\frac{1}{H_w} + \frac{1}{H^-}}(T^- - T_s^+) = H^+(T_s^+ - T^+) \quad (6.4.2)$$

The idea is to model the heat transfer through the IHX tube walls by calculating the wall surface temperatures T_s^+ , T_s^- as described by equations 6.4.1, 6.4.2. This approach fits well with the existing modelling framework for the description of fluid-structure heat transfer, treated via source or sink terms in the form of equation 3.2.26 discussed in subsection 3.2.3, which only requires knowledge of a structure surface temperature, fluid-structure heat transfer coefficient and structure interfacial area density. With regard to the former, due to the modelling choices, an average between the inner and outer tube surface area is used for the calculation of the tube surface area density. The wall heat transfer coefficient H_w is user-provided.

Compared to the wrapper heat transfer model, which updates the wrapper surface temperature in a segregated manner due to the OpenFOAM boundary condition management framework, the system of the two equations 6.4.1, 6.4.2 can be re-arranged implicitly for the two variables T_s^+ , T_s^- , so that:

$$T_s^+ = \frac{B^+ A^- + B^-}{A^+ A^- - 1} \quad (6.4.3)$$

$$T_s^- = \frac{B^- A^+ + B^+}{A^- A^+ - 1} \quad (6.4.4)$$

with:

$$A^\pm = \frac{H^\pm + H_w}{H_w} \quad (6.4.5)$$

$$B^\pm = \frac{H^\pm T^\pm}{H_w} \quad (6.4.6)$$

Equations 6.4.3, 6.4.4 are thus used to update the structure (i.e. IHX wall) surface temperatures similarly to how equations 3.2.41, 3.2.42 are used to update the structure temperatures of a generic 0-D structure thermal model or a 1-D pin model.

An important aspect related to the implementation of these equations needs to be discussed. Assuming that the volume Ω is a mesh cell, the representation of the IHX model shown in center of Figure 6.4.2 would require two sets of fields modelling the coolant temperature T and the coolant-wall heat transfer coefficient H coexisting in each mesh cell, one for the primary side and one for the secondary side. The most effective way to tackle this was to split the IHX computational domain into two fluid-mechanically separate domains, the primary side (i.e. region 19 in Figure 6.4.1, with a computational mesh cell

represented on left side in Figure 6.4.2) and the secondary side (i.e. region 24 in Figure 6.4.1, with a computational mesh cell represented on right side in Figure 6.4.2). Thus, with respect to Figure 6.4.1, the coolant temperature T in cells belonging to region 19 corresponds to T^+ and the tube wall structure temperature T_s corresponds to T_s^+ , with the same holding for T^- and T_s^- on the secondary side in region 24. However, as it can be seen from equations 6.4.3, 6.4.4, the update of the tube wall temperature on any side of the tube requires knowledge of the coolant fields on the other side as well. To obtain these values, OpenFOAM-enabled mesh-to-mesh projection operations are used to interpolate the coolant and tube surface temperature fields from one region to another. The mesh-to-mesh projection operations are in fact also possible between different cell regions belonging to the same single-physics mesh, such as is the case of the thermal-hydraulics. This projection operation is meaningful as long as both mesh regions modelling the primary and secondary side of the IHX have the same overall geometric shape and volume, so that a one-to-one correspondence between different mesh cells of the regions is maintained. While it is not required that the two mesh over the two regions are conformal for the projection operation to work, conformal meshes ensure the highest degree of energy conservation.

By denoting the mesh-to-mesh projection operation of a quantity \bullet from region + to region - with $\bullet|^{+ \rightarrow -}$, and vice-versa for $\bullet|^{- \rightarrow +}$, it follows from 6.4.3 that on the primary side one has:

$$T_s = \frac{B(A|^{- \rightarrow +}) + (B|^{- \rightarrow +})}{A(A|^{- \rightarrow +}) - 1} \quad (6.4.7)$$

while it follows from 6.4.4 that on the secondary side one has:

$$T_s = \frac{B(A|^{+ \rightarrow -}) + (B|^{+ \rightarrow -})}{A(A|^{+ \rightarrow -}) - 1} \quad (6.4.8)$$

with A, B defined as of 6.4.5, 6.4.6. Note that all the superscripts have been dropped as these fields A, B are now computed from the only existing values of the fluid temperature and fluid-structure heat transfer coefficient existing in each mesh cell, as the primary and secondary domains have been physically separated.

If the coolant on any (or both) sides of the IHX consists of a two-phase mixture (for any reason), the model presented is still applicable as long as the fluid temperature T and fluid-coolant heat transfer coefficient H are substituted by their mixture values as defined in equations 3.2.40 and 3.2.39 respectively, as discussed in sub-section 3.2.4.

As a final clarification, similarly to other structure thermal models, equations 6.4.7, 6.4.8 are employed at step seven of the thermal-hydraulic coupling algorithm discussed in sub-section 3.3.3, right before the solution of the fluid enthalpy equations.

6.4.1.2 Closure modelling

The closure models for momentum and heat transfer between fluid and structure are hereby introduced with respect to the various regions defined in Figure 6.4.1.

Flow regime map Two flow regimes have been considered for this one-phase scenario, parametrized with respect to the Reynolds number, namely a laminar regime for $Re < 1000$ and a turbulent regime for $Re > 2300$, as seen for the investigations of chapter 5. A quadratic interpolation of the friction factors, Nusselt numbers for the modelling of momentum and heat transfer in the transition flow regime was used.

Momentum transfer The momentum transfer due to the pump momentum source in the primary pump (region 17) was introduced earlier, and it was set so to reproduce the experimental mass flow rate ramp down for the transient. The momentum transfer due to frictional pressure losses was modelled as follows. The Rehme correlation [75] for the friction factor in wire-wrapped fuel pin bundles was used for the turbulent flow regime in all pin bundle regions, namely regions 2, 3, 5, 6, 10, 19, 24 for the turbulent flow regime. The Churchill correlation [100] for the friction factor was used for the turbulent flow regime in all structure regions that did not consist of a pin bundle, namely regions 1, 7, 8, 9, 10, 11, 12, 17, 22 with an assumed relative surface roughness of $1 \cdot 10^{-6}$. The Blasius correlation for the friction factor was used for the turbulent flow regime in the pipe regions 16, 17, 18, 20, 21, 23, 25. No friction factor correlations (i.e. no frictional pressure drops) were modelled in the pool plena. For the laminar flow regime, the well known analytic friction factor $f_d = 64/Re$ was used for all regions (except the plena). While this factor is derived specifically for circular ducts, it was deemed conservative enough for the task at hand.

Heat transfer Heat transfer closure relied on correlations for the Nusselt number in the form of equation 3.2.28. Said correlations were specified only in those regions where heat transfer phenomena between fluid and structure are intended to take place, namely regions 1, 2, 3, 6, 19, 24. For the laminar flow regime, a constant Nusselt of 3.66 was employed, which is the value for fully developed pipe laminar flow for a constant pipe temperature [101]. While not all components can be characterized simply as pipes, and while the surface temperature of structure was certainly not constant, it was deemed a more conservative choice compared to a Nusselt of 4.36, used in laminar flow regimes for constant wall heat flux scenarios instead. Nonetheless, the impact of the laminar correlations was investigated. For the turbulent flow regimes, the Nusselt number correlation by Mikityuk was employed [38] for all regions, with region-specific coefficients that depend on the pitch-diameter ratio of the bundle in each region. While region 1 does not properly consist of a pin bundle, the same correlation as that used for region 2, 3, 6 was used for simplicity.

6.4.1.3 Fluid thermo-physical properties

The thermo-physical properties of the coolant were those of sodium as compiled by Fink and Leibowitz [73] evaluated at a constant temperature of 633.2 K , namely the primary cold leg temperature. This was done to deal with the fact that the primary circuitry is modelled as a closed domain which offers no room for an expanding liquid with temperature changes. While this can be mitigated by modelling a small aperture within the core to allow for inflow-outflow (e.g. representative of the interface between the sodium free surface and the argon cover layer in the upper vessel plenum), this approach was deemed sufficient for the case at hand. Buoyancy effects were treated via an additional momentum source modelled with the Boussinesq approximation, which was of fundamental importance to treat natural convection established after pump trip.

6.4.2 Neutronics modelling

6.4.2.1 Geometric modelling

The neutronics mesh consists of a 2-D wedge domain as presented in Figure 6.4.3. It is defined exclusively for the reactor core, corresponding to regions 1 through 11 of the thermal-hydraulics mesh presented in Figure 6.4.1. While all of the boundaries of the thermal-hydraulics core regions match the boundaries of the different regions of the neutronics mesh, the latter is subdivided in a larger number of regions. This is done to account for the spatial variability of the macroscopic cross sections.

6.4.2.2 Role of the geometric model

The geometric model was used for the steady-state diffusion neutronics calculations to obtain the volumetric fuel power density field q that then acts as the nuclear fuel pin power source term $q_{int,s}$ as seen in equation 3.2.42, once interpolated to the thermal-hydraulics mesh.

The macroscopic cross sections that were used for the different regions were generated by Nikitin [98] in a previous collaboration for a neutronics-only analysis of the FFTF control rods worth with various computer codes, inclusive of GeN-Foam, with a 24-energy group binning [102]. As mentioned earlier, the exact fuel history for Test 13 was not available in detail. However, it was deemed reasonable enough to re-use the obtained macroscopic cross sections. Nonetheless, it was of interest to assess the impact of this choice, so that transient calculations were performed both with a flat power density distribution as well as the one obtained by the diffusion neutronics model.

With regard to the transient, which relies on a point kinetics model (which is spatially 0-D by definition), the geometric model was used exclusively for the adjoint flux-weighted averaging of the fields required for the calculation of reactivity feedbacks [103]. While the adjoint flux distribution was not directly

calculated, the sum of all the multi-group fluxes obtained by the steady-state diffusion calculation was deemed a reasonable approximation of the flux that would be obtained by a one-energy-group diffusion calculation, which is self-adjoint [104].

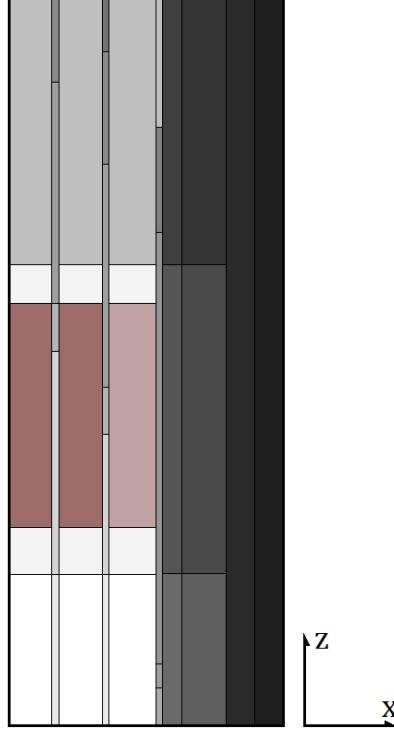


Figure 6.4.3: computational domain of the FFTF neutronics. It consists of a 2-D wedge model.

6.4.2.3 Point kinetics modelling

The point kinetics equations [104] were implemented as an additional possible neutronics treatment in virtue of their widespread use. These can be derived from the neutron transport equation and are formulated as:

$$\frac{d}{dt}n = \frac{\rho - \beta}{\Lambda}n + \sum_i \lambda_i C_i \quad (6.4.9)$$

$$\frac{d}{dt}C_i = \frac{\beta_i}{\Lambda}n - \lambda_i C_i \quad (6.4.10)$$

with n the neutron population density, ρ the system reactivity, β the total delayed neutron fraction, Λ the prompt neutron generation time, and β_i , λ_i , C_i , the delayed neutron fraction, decay constant and precursor concentration for the i -th delayed group respectively.

The system of equations 6.4.9, 6.4.10 was implemented for any number of delayed groups N_d (to be specified by the user), and results in a system of

$N_d + 1$ equations whose discretization is hard-coded with a backward Euler scheme.

Given knowledge of the reactor kinetic parameters β_i , Λ and the reactivity ρ , the time-evolution of n and C_i can be predicted. Due to the proportionality between neutron density and power, the time-dependent evolution of the nuclear fuel power density field can be modelled as $q(\mathbf{x}, t) = q(\mathbf{x}, t_0) n(t)/n(t_0)$, with $q(\mathbf{x}, t_0)$ and $n(t_0)$ being the fuel power density field and the neutron density at the initial simulation time t_0 . The initial values $n(t_0)$ and $C_i(t_0)$ can be calculated from the point kinetics equations by assuming that a steady state holds at t_0 . The reactivity ρ is calculated from the sum of the individual feedback reactivities of thermal-hydraulic and thermal-mechanical phenomena of interest.

These reactivities are typically computed based on the adjoint flux weighted value of thermal-hydraulic field projected to the neutronics mesh. For any thermal-hydraulics field ϕ , the combination of these two operations (adjoint-weighting the projected thermal-hydraulics field) is denoted as $\bar{\phi}$. The reactivity contributions that were considered for the case of the FFTF are hereby presented:

- fast neutron fuel Doppler reactivity contribution in the form $\rho_{f,D} = -c_{\rho,D} \ln(\bar{T}_f/\bar{T}_{f,0})$, with $c_{\rho,D}$ the Doppler coefficient, T_f the average fuel temperature and $T_{f,0}$ the average fuel temperature at steady-state (i.e. at the start of the simulation);
- fuel axial expansion reactivity contribution in the form $\rho_{f,ax} = c_{f,ax}(\bar{T}_f - \bar{T}_{f,0})$;
- coolant density reactivity contribution in the form $\rho_{c,\rho} = c_{c,\rho(T)}(\bar{T} - \bar{T}_0)$, with T being the coolant temperature, wherein the changes in density are modelled via changes in average fuel coolant temperature. Due to the Boussinesq approximation, the coolant density field is constant, and the density change that *would* correspond to the equivalent temperature change needs thus to be modelled via an adequate manipulation of the feedback coefficient $c_{c,\rho(T)}$.
- structure density reactivity contribution in the form $\rho_{s,\rho} = c_{s,\rho(T)}(\bar{T}_s - \bar{T}_{s,0})$ with T_s being the structure temperature of the 0-D thermal model in those regions where it is defined, namely temperature that is taken as representative of the average wrapper, wrapper wire, diagrid temperatures;
- core radial expansion reactivity contribution in the form $\rho_{exp} = c_{exp}(\bar{T}_s - \bar{T}_{s,0})$;
- control rod driveline expansion reactivity contribution ρ_{CR} . This requires: 1) a control rod worth ρ_{CR} vs. control rod insertion Δ_{CR} map; 2) a measure of the average absolute axial expansion Δ_{CR} of the control

rod drivelines. This expansion is approximated as $\Delta_{CR} = c_\Delta(\bar{T}_s - \bar{T}_{s,0})$, where the same field T_s used for the wrapper and wire temperature within the core is taken to be representative of the driveline. This is a very crude approximation but it was a consequence of the fact that the upper plenum was not modelled in the neutronics mesh. Ideally, the control rod driveline temperature could be modelled via a 0-D thermal model of a porous structure in the upper plenum of the thermal-hydraulics mesh and with an extended neutronics mesh to the upper plenum to enable the projection of the driveline temperature field;

- GEM reactivity contribution ρ_{GEM} . This was modelled based on a similar approach to the control rod expansion. A GEM reactivity worth vs. sodium free surface level height l map was provided by the ANL and determined via Monte Carlo approaches. Furthermore, ANL provided a correlation between the core mass flow rate and the sodium free surface level l . The GEM reactivity contribution is thus determined from the sodium velocity field \mathbf{u} (at the core region inlets of the thermal-hydraulics mesh) interpolated to the neutronics mesh.

While the FFTF geometric, structural details presented so far are available in the literature, kinetic parameters and feedback coefficients were provided by ANL for the benchmark and could not be disclosed at the present time.

6.5 Results

6.5.1 Steady state results

The steady-state results of the FFTF neutronics and thermal-hydraulics models were obtained to provide initial conditions for the subsequent transient simulation.

It is recalled that the employed reference model uses a fuel power density distribution coming from the diffusion-based neutronics calculation. The volumetric fuel power distribution is thus proportional, in those regions containing fuel, to the total neutron flux reported on the left-most part of Figure 6.5.1. These flux distributions span exclusively over the core domain (i.e. regions 1 through 11). As the total flux is calculated from sum of the individual group fluxes, some representative flux distributions (normalized for clarity) are reported as well for decreasing neutron energies, left-to-right.

The overall temperature distribution and some representative velocity streamlines are reported in Figure 6.5.2, which also highlights the thermal coupling between the primary and secondary circuit in the IHX regions. The cold and hot leg of the secondary are not shown in their entirety for figure size convenience.

The steady-state values for some representative system quantities are instead reported in Table 6.5.1. With regards to the outlet temperatures of the

Quantity	Value	
	Calculated	Experimental
Total power (MW)	199.2	199.2
Primary mass flow rate (kg/s)	2205.5	2204.2
Inner core mass flow rate (kg/s)	809.6	810.2
Outer core mass flow rate (kg/s)	1068.9	1060.5
Bypass flow(kg/s)	156.1	157.5
Secondary mass flow rate (kg/s)	2215.6	2202.2
Core inlet temperature (K)	602.1	<i>N/A</i>
Inner core outlet temperature (K)	696.2 ^a 706.3 ^b	702.8
Outer core outlet temperature (K)	691.5 ^a 680.6 ^b	679.15

Table 6.5.1: Comparison of calculated and experimental results at steady-state prior to the start of LOFWOS Test 13. The a and b superscripts denote result obtained by the diffusion-based fuel power density the piecewise uniform fuel power density respectively.

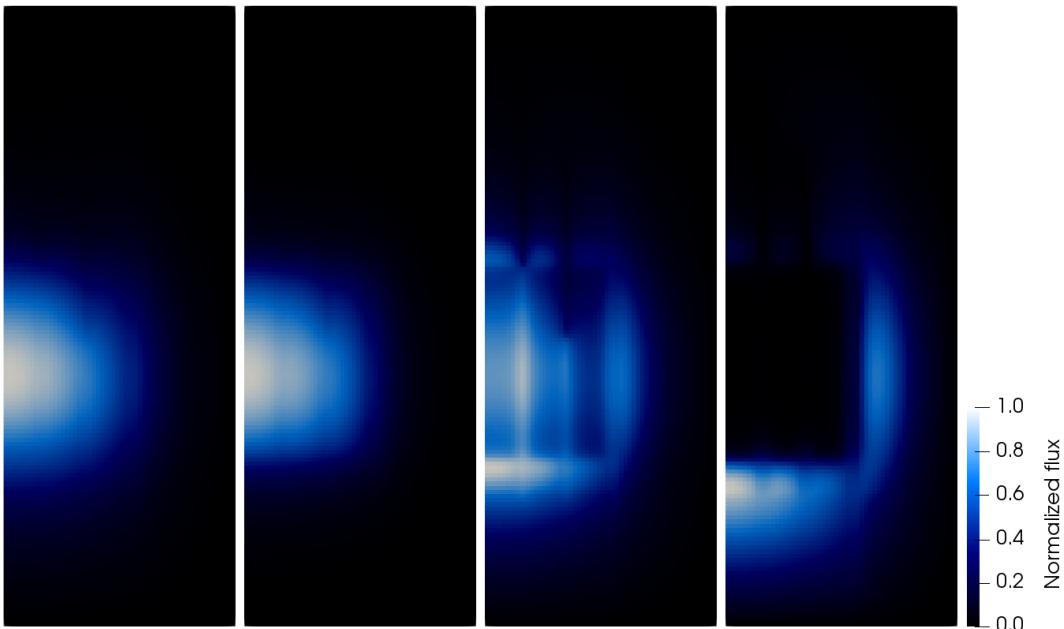


Figure 6.5.1: calculated neutron flux distributions in the FFTF core for different energy groups at steady state with the diffusion neutronics sub-solver. From left to right (peak flux values in brackets): one-group-flux ($3.4 \cdot 10^{15} n/cm^2/s$), flux of group 7 ($3.7 \cdot 10^{14} n/cm^2/s$), flux of group 19 ($6.4 \cdot 10^{13} n/cm^2/s$), flux of group 24 ($1.5 \cdot 10^{14} n/cm^2/s$).

inner and outer core regions, two different sets of calculated results are presented as an assessment of the impact of using the diffusion neutronics fuel power density against a piecewise uniform fuel power density, namely two uniform power density distributions, one for the inner core fuel and one for the outer core fuel. The latter was provided by ANL. As discussed earlier, the macroscopic cross sections that were used for the calculation of the fluxes were obtained from a previous investigation that might not be representative of the (unknown) core fuel burnup at the time of LOFWOS Test 13. The piecewise

uniform fuel power density distributions proved to result in the best agreement with the experimentally measured temperatures at steady-state. Nonetheless, both these models are further investigated.

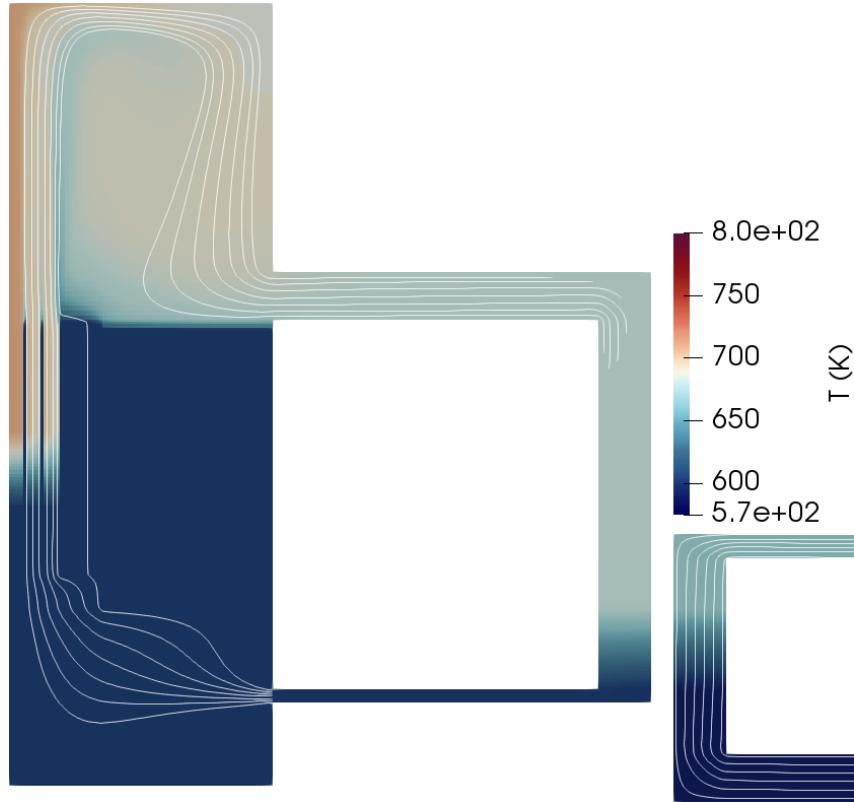


Figure 6.5.2: calculated temperature distribution and representative velocity streamlines in the FFTF at steady-state. The domain of the secondary loop was cut for visualization convenience. The heat transfer between the IHX regions results in the observed thermal gradients in said regions.

6.5.2 Transient results

The transient consisted in a unprotected pump trip and the monitoring of the subsequent evolution of the system to assess the effectiveness of: 1) the GEMs, which are supposed to provide enough negative reactivity to effectively shut down the core; 2) natural circulation to remove the decay heat. The conditions on the secondary side of the circuitry were maintained constant throughout the transient.

6.5.2.1 Reference model results

The evolution of a number of integral system parameters are reported in Figure 6.5.4 and compared against experimental results.

The total reactor power evolution (top-left) is first characterized by a rapid decrease, primarily governed by the GEM reactivity feedback. This is observed

both in the evolution of the total reactivity (bottom-right) and in the evolution of the individual reactivity components, presented in Figure 6.5.3.

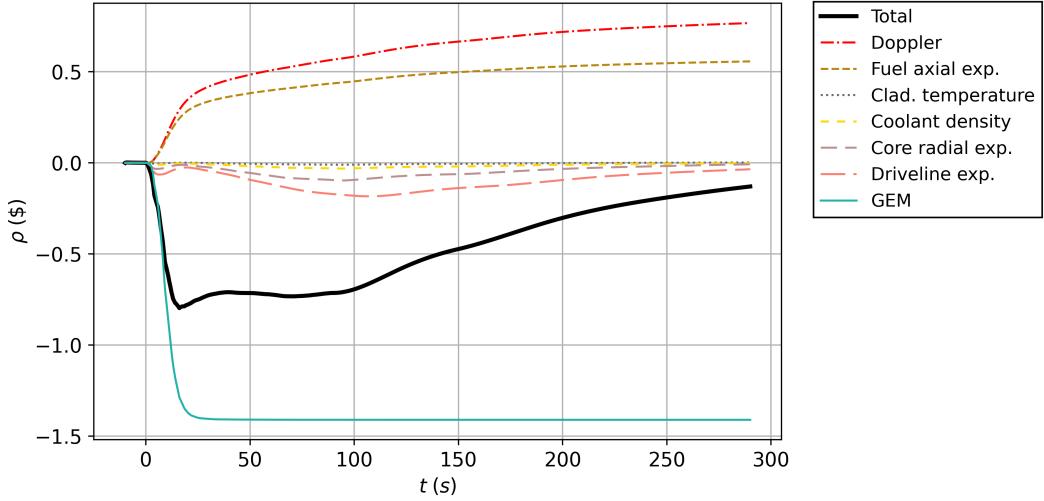


Figure 6.5.3: calculated results for the evolution of the various reactivity contributions obtained by the reference computational model of FFTF LOFWOS Test 13.

The GEM reactivity contribution reaches its maximum magnitude $\simeq 25\text{ s}$ after pump trip, and the evolution of the rest of the transient is governed by the other reactivity feedbacks. In particular, the agreement between the calculated and experimental power is reasonable up to $\simeq 200\text{ s}$, after which disagreements grow larger. This is a consequence of the calculated total reactivity evolution, which consistently under-estimates the experimental one after $\simeq 100\text{ s}$. Possible explanations for this will be discussed later in greater detail when presenting the sensitivity of the results to some model parameters. It is recalled that the evolution of the GEM reactivity feedback is exclusively a function of the total core mass flow rate, which depends on both the primary pump momentum source and the chosen frictional pressure drop models in the various vessel regions that govern the flow distribution.

With regards to the primary mass flow rate, the momentum source was set to reproduce the ramp down for the first 90 s, after which it is turned off to allow for the establishment of a natural circulation regime. The agreement between the calculated and measured total primary mass flow rates is good throughout the transient.

With regards to the coolant temperature, the reported results compare the calculated inner core outlet temperature evolution against the experimentally measured outlet temperature of a specific assembly (PIOTA-2 [90]) in the inner core region. It should be stated that, due to the appreciable degree of thermal-hydraulic heterogeneity of the FFTF core, the temperature values obtained by the code for the inner core region might not be entirely representative of the data measured at specific inner core reactor assemblies, as an appreciable discrepancy exists between calculated and experimental results.

This is also addressed later when comparing calculation results obtained with different model parameters, as well as available data in the literature obtained by other codes. Nonetheless, the main features of the temperature evolution were reproduced satisfactorily. At the start of the transient, the mass flow rate decreases faster than the total power, resulting in a rapid outlet temperature increase. In the subsequent $\simeq 20$ s however, as the power decays almost exponentially while the mass flow rate decays less than exponentially, the resulting decrease in temperature shapes the first temperature peak. As the GEM reactivity worth is fully deployed in the first $\simeq 25$ s, the trend in reactivity reduction is reduced and the power-to-flow ratio starts increasing again up to the establishment of natural circulation after $\simeq 130$ s. After said point, while the reactor power continues to decay, the primary flow rate remains reasonably constant resulting in the overall reduction of the outlet temperature.

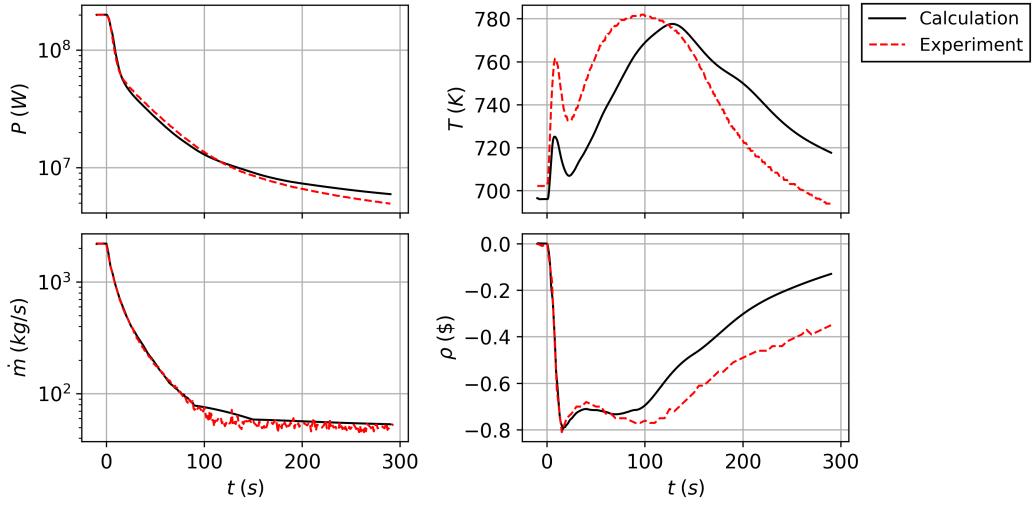


Figure 6.5.4: calculated results for the evolution of total reactor power (top-left), inner core outlet temperature (top-right), total primary mass flow rate (bottom-left), total reactivity (bottom-right) obtained by the reference computational model of FFTF LOFWOS Test 13 compared against experimental data.

For clarity, calculated temperature and velocity streamlines distributions obtained by the reference model are presented in Figure 6.5.5 at: 90 s after pump trip (when the largest outlet temperatures are attained); 300 s after the transient, as the natural circulation regime is established.

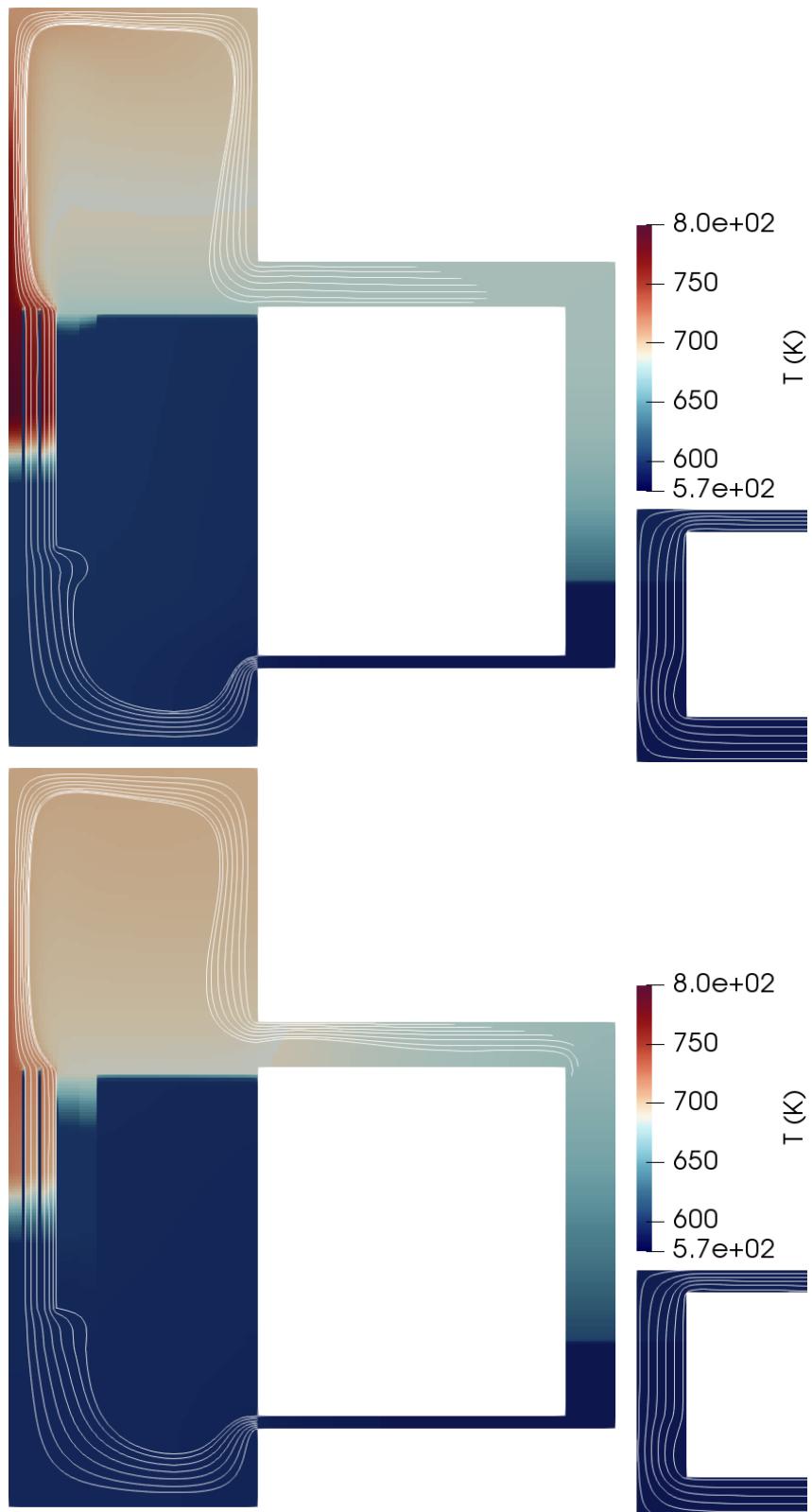


Figure 6.5.5: calculated temperature distributions and representative velocity streamlines in the FFTF at 90 s after pump trip (top) and 300 s after pump trip (bottom).

6.5.2.2 Parametric investigation

The effects of a number of system parameters and models were investigated to assess the nature of the discrepancies between the calculated results and the experimental data previously discussed. While a thorough analysis might focus on a large collection of system parameters, including geometric modelling choices, this investigation only focused on some thermal-dynamics aspects: 1) impact of coolant-structure heat transfer correlations; 2) impact of using the diffusion-based volumetric fuel power density with adjoint flux-weighted feedback fields versus the piecewise uniform fuel power density and unweighted feedback fields; 3) impact of the pin gap conductance versus pin linear power map, as it is a function of burnup, which was not known with certainty. This focus on thermal models results from the observation that the most important calculated reactivity contributions came from the fuel Doppler effect, fuel axial expansion and control rod driveline expansion, as previously seen in Figure 6.5.3, which are all governed by the dynamics of the heating and cooling of different reactor components, as well as how the related feedback fields are weighted.

In the first place, the heat transfer correlations were investigated. It is recalled that the reference model assumes a heat transfer coefficient calculated from a constant Nusselt of 3.66 in the laminar regime (for $Re < 1000$), which gains increasing importance in the core as the flow transitions to natural circulation. As an opposite extreme case, the same heat transfer correlations for the Nusselt number used in the turbulent regime were used in the laminar regime, and the steady-state and transient calculations performed again. The differences with respect to the reference case were fundamentally found to be irrelevant, as the largest thermal resistances in the heat transfer between fuel and coolant consist of the fuel itself, which has a low thermal conductivity, and the gap conductance, which dominate the heat transfer process at low power values (due to the broadening of the gap resulting from fuel cooling and thus a reduction in gap conductance to small values $\simeq 500 \text{ W/m}^2/\text{K}$, depending on the burnup).

In the second place, the impact of using the diffusion-based fuel power density with adjoint flux-weighting of feedback fields versus the use of a piecewise uniform fuel power density field and without feedback field weighting was assessed. The results concerning outlet inner core temperature and total reactivity evolution obtained by the former are presented on the left column of Figure 6.5.6, while those obtained by the latter are presented on the right column of Figure 6.5.6. Before discussing the specific sub-cases that each column shows, the general trend is that the latter model results in a larger temperature excursion, that however more closely reproduces the experimental data up to the maximum outlet temperature, also with regards to the initial local maximum temperature attained at $\simeq 10 \text{ s}$. The experimental total reactivity is appreciably better resolved as well, in spite of the persistence of the discrepancies. This is understandable in light of what was already discussed, specifically

pertaining to the fact that diffusion-based calculations and the obtained fuel power density relied on macroscopic cross sections obtained from previous investigations, while the piecewise uniform power density was provided by the benchmark data compilers at ANL.

In the third place, the effect of employing different gap-conductances (derived from the correlation by Lavarenne [99]) for different fuel burnup values was investigated. It is recalled that the reference model previously discussed used a map obtained for a burnup value of $50 \text{ GW}_d/t_{HM}$, which was deemed a reasonable assumption given the available knowledge of the core. The temperature and reactivity evolution obtained by the use of different maps is reported in Figure 6.5.6 for both the diffusion-based and piecewise uniform fuel power density fields for the limiting cases of fresh fuel on one hand and fuel with a burnup of $100 \text{ GW}_d/t_{HM}$. While such a large burnup is prohibitively large for fuels used in traditional reactor designs, fuel burnups in excess of said value were attained in the FFTF [105].

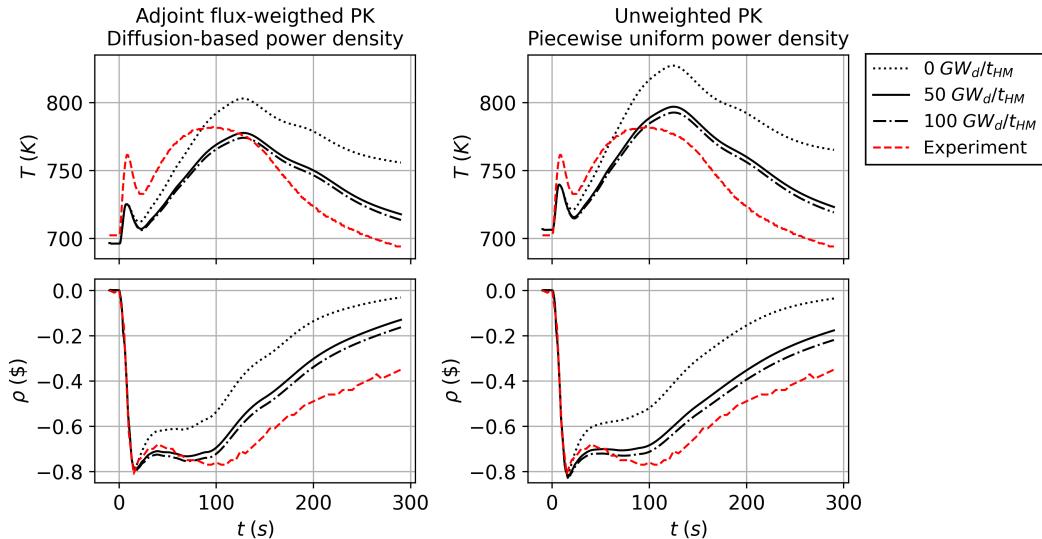


Figure 6.5.6: comparison of calculated obtained by different models of FFTF LOFWOS Test 13. Left column: results obtained with adjoint-flux weighting of reactivity feedback fields and the fuel power density distribution obtained by the diffusion-based neutronics. Right column: results obtained by the unweighted averaging of reactivity feedback fields and a piecewise constant power density distribution. Top row of each column: inner core outlet temperature profiles. Bottom row: total reactivity.

A particular trend can be consistently observed: for larger values of burnup and thus larger values of the gap conductance (see Table 6.4.1), both the overall temperature and reactivity magnitudes decrease. There are two counter-acting effects at play. On one hand, the larger gap thermal conductivity means that the fuel cools down at a faster rate, which in turn means that the positive reactivity insertion from fuel axial contraction and Doppler is inserted at a faster rate as well. However, this effect does not change the final total amount

of reactivity that can be inserted from fuel cooling. On the other hand, a larger value of the gap thermal conductivity also results in a smaller value of the fuel temperature at steady-state, so that the overall maximum reactivity insertion from cooling fuel is reduced. This second effect is thus the most relevant one, and the reduction of the total reactivity brings values closer to the experimental ones.

On the temperature side, a smaller fuel temperature due to a larger gap conductance will inevitably result in larger peak coolant temperatures. This effect is significant when comparing fresh fuel to fuel of $50 \text{ GW}_d/t_{HM}$ of burnup, and its importance is significantly reduced when comparing the $50 \text{ GW}_d/t_{HM}$ burnup fuel to the $100 \text{ GW}_d/t_{HM}$ burnup fuel. A similar effect is thus expected from varying values of the fuel thermal conductivity (which was assumed at 3 W/m/K), yet this was not presently investigated.

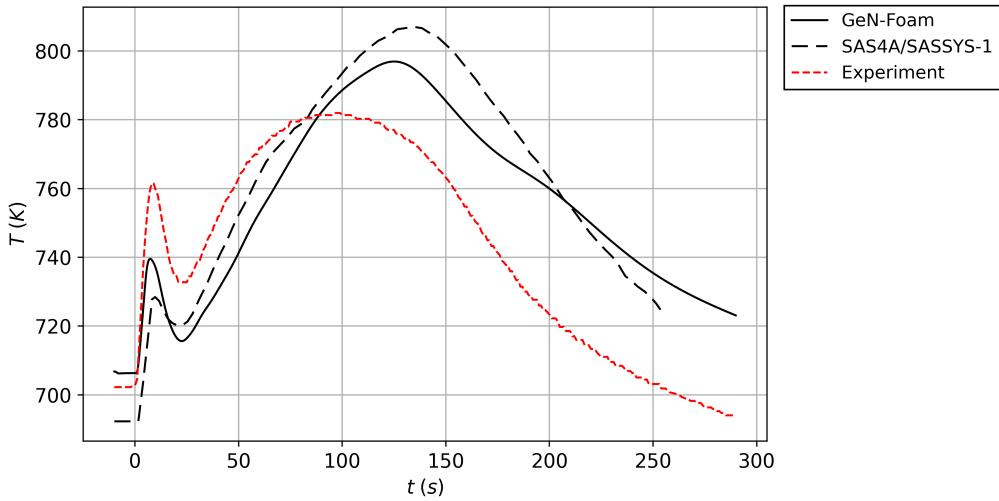


Figure 6.5.7: Calculated inner core outlet temperature results for the FFTF LOFWOS Test 13 obtained compared against results obtained by the SAS4A/SASSYS-1 system code and experimental results.

On a final note, the results obtained by the code are compared against results obtained by other codes, namely the SAS4A/SASSYS-1 system code available in the open literature [90] for LOFWOS Test 13. These are reported in Figure 6.5.7, are compared against the results obtained by the piecewise uniform fuel power density model, unweighted feedback fields and gap conductance versus linear pin power map evaluated at $50 \text{ GW}_d/t_{HM}$. The agreement between the two codes is significant, in spite of the discussed discrepancies with the experimental results.

6.6 Conclusions

This chapter detailed the implementation and application of the thermal-hydraulics methodology and resulting code into the broader GeN-Foam multi-

physics environment, which was further expanded with the addition of a point kinetics model for the treatment of neutronics. The resulting code was applied to the analysis of the LOFWOS Test 13, which was performed at the FFTF SFR to investigate the effectiveness of novel passive safety features, namely the GEMs. This analysis takes place in the broader context on an IAEA CRP benchmark re-analysis effort. The analysis of the transient was performed via a coupled one-phase thermal-hydraulics and neutronics approach, both relying on a 2-D representation of the reactor.

The thermal-hydraulics model represented the entire primary loop and part of the secondary loop with a coarse-mesh approach. A novel model to treat the heat exchange between primary and secondary loops via the IHXs was developed relying on the same mesh-to-mesh field projection operations that enable the coupling between different physics operating on different meshes.

The neutronics model extended exclusively over the reactor core and was used for the calculation of the steady-state fuel power density distribution and the neutron fluxes, which were used for the weighting of the fields used to compute feedbacks for the point kinetics model. For the sake of comparison, the results obtained by using piecewise uniform fuel power distributions were investigated as well.

The overall steady-state results agree well with experimental data, with the best agreement that comes from the use of piecewise uniform fuel power distributions instead of the diffusion-neutronics results. This is understandable, as the macroscopic cross sections used for the diffusion calculation were obtained for a different FFTF core configuration due to the unavailability of precise fuel burnup data preceding LOFWOS Test 13. Conversely, piece-wise uniform power density distribution were provided by benchmark data compilers at ANL.

With regard to the transient, the evolution of various integral parameters was compared against experimental data. The agreement could be deemed reasonable, and good when compared to results obtained by previous investigations with other codes. While a parametric analysis was initiated in the scope of the present investigation to assess some general aspects of the sensitivity of the computational model to different parameters, a broader investigation will be performed within the context of the IAEA CRP benchmark to further assess the capabilities of the model.

Chapter 7

Conclusions

7.1 Summary

This work was centered on the investigation, development and application of a coarse-mesh methodology for both one-phase and two-phase flow scenarios in nuclear reactors. While the coarse-mesh approach aims at a degree of accuracy comparable to that of existing sub-channel codes, the key elements that motivated this thesis work lie in the broader context of recent trends and needs in the nuclear field, namely: 1) greater geometric and physics modelling flexibility, conducive to the investigation of non-traditional or complex reactor concepts and safety features; 2) seamless integration into broader multi-physics frameworks thanks to a streamlined coupling with other single-physics; 3) capability to take advantage of the massive parallel scalability potential that modern High Performance Computing (HPC) clusters offers; 4) use of modern programming paradigms to improve code performance, maintainability and to simplify the implementation of novel solution algorithms, coupling schemes and physical models.

A coarse-mesh methodology based on volume averaging techniques allowed to address points 1 and 2, thanks to the possibility to use general computer-aided design (CAD) geometries, unstructured meshes, and mesh-to-mesh projection algorithms. Points 3 and 4 were instead addressed via the adoption of the Finite Volume Method (FVM)-based OpenFOAM C++ numerical library as a development framework.

The principal investigated aspects, developments and findings of this work are hereby summarized.

- The governing equations for mass, momentum and energy conservation for a generic multi-phase system were rigorously derived in a coarse-mesh context by making use of the mathematical tools of volume averaging. The key steps in constructing a coarse-mesh representation of a system of interest were derived and discussed. The theoretical nature of the equation closure terms that arise as a natural consequence of the volume averaging approach was presented. In a multi-phase context, the

formal equivalence of the governing equations obtained by a coarse-mesh approach and a dispersed Euler-Euler approach was discussed.

- The methodology was implemented in a computer code for the analysis of one-phase and two-phase flows in nuclear reactors by taking advantage of the OpenFOAM numerical library. The discussion of the implementation was centered on two fundamental aspects: the first, related to providing an experimentally-informed closure to the governing equations as well as specific numerical treatments for each closure term; the second, related to the algorithms for the solution of the coupled system of equations for mass, momentum and energy conservation.
- Experimentally-informed closure relations were implemented based on available experimental correlations for the treatment of momentum and heat transfer. Great emphasis was put on the modularity of the modelling approach, which allows for a streamlined implementation of new correlations for specific systems and working fluids.
- The developed thermal-hydraulics solution algorithm for the coupled governing equations stems from the well known merged PISO-SIMPLE (PIMPLE) solution algorithm, with further developments. The evaluation of velocity predictors was found to be inconsequential on final calculation results for the one-phase solution algorithm. With regards to the two-phase pressure-velocity coupling algorithm, the ideas of Partial Elimination were implemented in a novel, numerically implicit manner. This was found to significantly improve convergence properties of two-phase calculations involving tight momentum coupling between the two phases when compared to existing approaches employed e.g. in standard OpenFOAM-based two-phase solvers.
- The overall fluid-mechanics solution algorithm, with specific regard to the phase fraction equation solution based on the Multidimensional Universal Limiter with Explicit Solution (MULES) algorithm and on the developed pressure-velocity coupling algorithm, has been verified for a number of cases via the Method of Manufactured Solutions (MMS). This also involved the verification of two different approaches for managing the velocity field for the solution of the pressure-velocity coupling. While the treatment of variables in OpenFOAM is co-located with respect to the computational mesh (i.e. cell-centered), an existing approach that mimics a staggered grid (i.e. face-centered) approach for the velocity treatment was investigated as well. While the overall mesh convergence properties of the cell-centered algorithm were found to be better in flows dominated by diffusive effects, the stability advantages offered by the face-centered treatment were deemed of greater importance for two-phase investigations.

- The parallel performance of the developed computer code was investigated up to 4096 computer cores via a strong scaling test. Linear scaling was observed down to ≈ 10000 cells per domain, which is a well known limit for OpenFOAM-based programs and confirms that no particular bottlenecks have been introduced in the developed code.
- The main test case for this work consisted of Sodium-cooled Fast Reactors (SFRs)-related applications, as these are among the most technologically mature and nearest-term deployable advanced reactor concept. Interest in sodium as a working fluid was also tied to numerical considerations, as the simulation of phase change in sodium flows poses greater numerical stability issues when compared to water, owing to the much larger phase density difference between the two fluid states at their respective operating pressures.
- The developed code was applied to the investigation of a number of sodium boiling experiments for comparison, validation purposes. These consisted of quasi-steady state boiling at the Joint Research Centre (JRC) facilities in Ispra, Italy, and transient boiling in electrically heated mock-up SFR fuel elements at the Kompakter Natriumsiede Kreislauf (KNS) facility at the Kernforschungszentrum Karlsruhe, Germany. Very good agreement between calculated and experimental results, consisting of total two-phase frictional pressure drop versus inlet mass flow rate, was obtained for the quasi-steady state analysis with a range of models primarily related to the modelling of two-phase pressure drop multipliers. Good agreements were obtained as well for the transient boiling scenarios in terms of the evolution of the calculated inlet mass flow rate, pressure at specific locations, vapour phase total volume and axial spatial extent. However, a much larger sensitivity of these results to specific models was observed.
- The developed code was applied to the investigation of specific SFR fuel assembly features, namely the inter-assembly gap and novel safety features such as assembly wrapper windows. The investigation was specifically aimed at assessing the impact of said features on the flow and temperature distributions during assembly flow blockage, and it was limited to one-phase flow scenarios. This investigation tentatively indicated that: 1) thermal effects of these features are not negligible and may result in boiling delay or prevention in a number of circumstances; and 2) wrapper windows can give rise to 3-D effects that result in a non-monotonic behaviour of assembly temperatures with respect to the inter-assembly gap flow. The investigation was inclusive of a representative study for assessing the mesh convergence of some integral results, and thus assessing the limits of validity of the coarse-mesh approach. An approach was proposed where results were deemed acceptable as long as they lied in

the so-called asymptotic convergence range, and as long as error bands could be estimated.

- The developed thermal-hydraulics code was integrated in the GeN-Foam multi-physics solver. The resulting code was applied for the investigation of the Loss Of Flow Without SCRAM (LOFWOS) Test 13 performed at the Fast Flux Test Facility (FFTF), within the broader context of an International Atomic Energy Agency (IAEA) coordinated research project (CRP) for the benchmark re-analysis of said test for validation and code-to-code comparison purposes. The re-analysis was performed from the standpoint of one-phase thermal-hydraulics and neutronics coupling, involving the use of diffusion-based neutronics for steady-state calculations and a newly developed point kinetics model for the transient calculations. The overall agreement between calculated and transient results is acceptable, yet the computational model will require further refinement as the CRP progresses.
- The thermal-hydraulic investigations also demonstrated the flexibility of the underlying OpenFOAM framework, which enables the implementation of ad-hoc models for treating phenomena such as: 1) the heat transfer through assembly wrappers; and 2) the heat transfer between physically disconnected portions of the computational domain to model an intermediate heat exchanger (IHX).

7.2 Perspectives and future work

Future research work can be subdivided into methodological-related developments and the expansion of the application range to further systems and working fluids of interest.

On the methodological side, the principal aspect that can be identified lies in the investigation of the effectiveness of a first-order implicit method with a purely upwind (i.e. first order) discretization of the advective terms for the solution of the phase fraction equation for the two-phase algorithm. With respect to the MULES, a first order-implicit upwind approach is expected to: 1) reduce calculation times associated with the phase fraction equation solution, as the MULES algorithm tends to be rather computationally expensive; 2) lessen stability limitations coming from the Courant–Friedrichs–Lewy (CFL) condition due to its implicit nature. On the downside, the first-order nature of the approach is expected to make the approach more numerically diffusive.

While the breadth of applications is large given the current code capabilities and framework flexibility, the most important extensions that can be envisioned for this work are hereby summarized.

- Expansion of the modelling capabilities to the treatment of water as a working fluid, specifically for one-phase Pressurized Water Reactor

(PWR) and two-phase Boiling Water Reactor (BWR) analyses. This collectively denotes the process of implementing the required correlations for heat and mass transfer and their validation against experimental data. This process is simplified by the developed modelling framework, yet it is a project of a possibly very large breadth due to the maturity of the Light Water Reactor (LWR) technology.

- Re-analysis of the impact of the investigated European Sodium-cooled Fast Reactor (ESFR) features, namely inter-assembly gap flow and wrapper windows, during flow boiling.
- Within the context of the IAEA CRP, parametric analysis of the impact of further physical models on the calculated LOFWOS Test 13 transient with the FFTF model and/or geometric improvements of the model itself.
- Investigations of Reactivity Insertion Accidents (RIAs) experiments and benchmark calculations for the further validation of the multi-physics approach and code-to-code comparison.
- Extension of the developed point kinetics model to Molten Salt Reactors (MSRs).

Appendix A

1-D nuclear fuel pin thermal model

This appendix intends to clarify the implementation of a 1-D nuclear fuel pin model via a Finite Volume Method (FVM)-based discretization of the domain and governing equations. A discussion of the discretization process in the most general terms is presented in A.1, while the integration of the approach in a coarse-mesh context is discussed in section A.2.

A.1 Modelling and discretization

A.1.1 General remarks and approximations

A nuclear fuel pin generally consists of a vertically stacked column of cylindrical nuclear fuel pellets, which might or might not have a central hole, and a metallic cladding, which encloses the fuel. By design, a thin gap is intended to exist between the fuel pellets and the cladding. The cladding is generally filled with an inert gas such as helium.

While a detailed study of the evolution of such a system from a thermal-mechanical perspective belongs to the domain of fuel behaviour analysis, a number of approximations are made in this context to enable a simplified form of nuclear fuel description and its coupling with the fluid flow. These approximations are:

1. the fuel pellet stack can be treated as a single axial continuum;
2. the fuel pellets and cladding dimensions do not evolve in time from the perspective of the analysis of heat diffusion, and the geometry is axially symmetric;
3. all the fields of interest (i.e. power density, temperature, flow conditions at the outer cladding surface) are axially symmetric;
4. axial temperature gradients are negligibly small compared to radial temperature gradients;
5. the thermo-physical properties of the fuel and cladding, (i.e. density, heat capacity and thermal-conductivity) are constant in time and uniform

- over their respective domains, while the fuel power density profile is uniform over the fuel but can change in time;
6. the absolute heat capacity of the gases in the gap and possibly central fuel hole is negligible;

Approximations 1 through 4 are used to simplify the general 3-D heat diffusion problem to a 1-D problem in the radial direction, as the reference frame that is used consists of a cylindrical frame centered on the pin axis of symmetry. In particular, it is approximation 4, in combination with the typically small fuel thermal conductivity values, that allows to approximate the study of heat conduction in the 3-D pin as a collection of 1-D radial heat conduction problems, one per pin-axial layer. While this neglects the axial coupling between adjacent fuel and cladding axial layers, the generally varying conditions of the fluid flow along the cladding outer surface in the axial direction allow to retain a significant degree of axial coupling.

With regard to a heat diffusion problem in the form:

$$\frac{\partial}{\partial t} (\rho^* c_p^* T^*) - \nabla \cdot (\kappa^* \nabla T^*) = q^* \quad (\text{A.1.1})$$

which is defined over the entirety of the pin, i.e. fuel, gap, cladding, approximations 5 and more specifically 6 allow to decompose it into two diffusion problems, one over the fuel and one over the cladding:

$$\rho_f c_{p,f} \frac{\partial}{\partial t} T^* - \kappa_f \nabla \cdot (\nabla T^*) = q \quad (\text{A.1.2})$$

$$\rho_c c_{p,c} \frac{\partial}{\partial t} T^* - \kappa_c \nabla \cdot (\nabla T^*) = 0 \quad (\text{A.1.3})$$

where the superscripts * denote spatially continuous fields, the subscripts f , c are used to denote fuel and cladding properties respectively, T^* is the radial temperature field and q is the fuel power density. These equations are coupled via heat flow conservation considerations at the fuel-gap and gap-cladding interfaces, based on gap conductance models. This spares the need for meshing the mechanically thin gap and spares numerical discretization problems related to the large difference in thermal conductivity between adjacent materials layers, i.e. between fuel and gap, gap and cladding.

Sub-section A.1.2 discussed the definition and structure of the 1-D radial mesh that is employed in this implementation of a nuclear fuel pin model. Sub-section A.1.3 discusses the discretization of a heat diffusion equation in the form A.1.2 in the bulk of either the fuel or cladding mesh, as the only formal difference lies in the power density source term and its is fundamentally inconsequential on the form of the discretized matrix coefficients. Sub-section A.1.4 closes the problem constructed in sub-section A.1.3 via the application of boundary conditions and coupling conditions to represent and enable the solution of equations A.1.2, A.1.3 as a single system of implicitly coupled equations.

A.1.2 Domain discretization

Let us consider a nuclear fuel pin consisting of a fuel pellet of inner and outer radii r_{fI} , r_{fO} and a cladding of inner and outer radii r_{cI} , r_{cO} , and let us focus on an axial slice of height Δz . This scenario is represented in Figure A.1.1, alongside the 1-D mesh discretization that is adopted in this work.

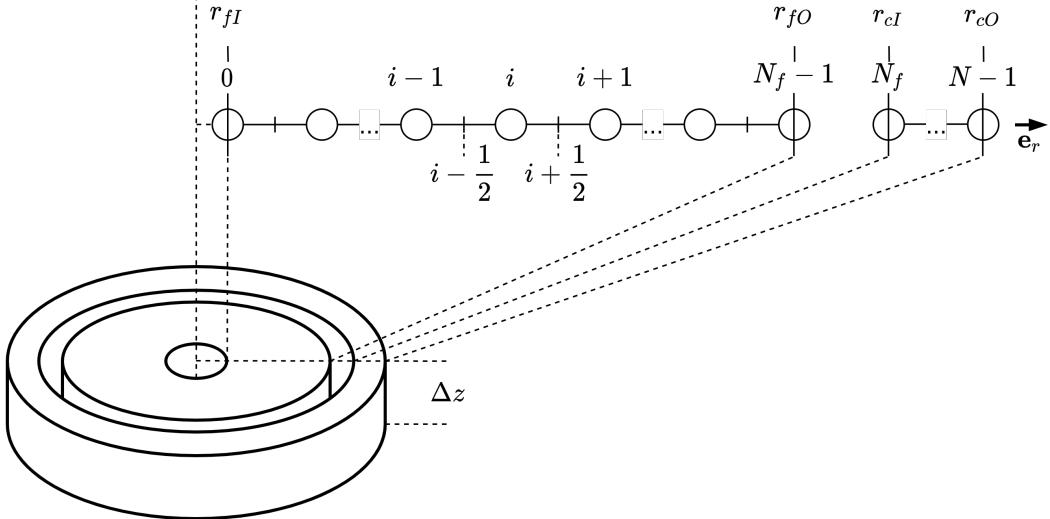


Figure A.1.1: Representation of an axial slice of a nuclear fuel pin with the definition of a 1-D radial mesh. The mesh cell centers are indicated with circles, boundaries with vertical lines. Cells are indexed with integers, boundaries with half-integers. Two additional ghost cells are defined before the first and after the last cell, as they are useful for the imposition of boundary conditions.

The computational domain consists of a mesh over the fuel domain, between r_{fI} and r_{fO} , and one over the cladding domain, between r_{cI} and r_{cO} . While one could define a third mesh over the gap, the negligible heat capacity of the gap combined with its small width renders this effort unnecessary, as it becomes reasonable to thermally couple fuel and cladding by imposing an instantaneous heat flow balance through the gap, as it will be discussed later.

The fuel mesh is composed of N_f elements. Of these, the two boundary elements at $i = 0$, $i = N_f - 1$ have a radial width of $\frac{1}{2}\Delta r_f$, while cells in the fuel bulk have a uniform radial width $\Delta r_f = (r_{fO} - r_{fI}) / (N_f - 1)$. The fuel mesh structure requires $N_f \geq 2$. Similarly, the cladding is composed in N_c elements, with the total number of mesh elements $N = N_f + N_c$. The same approach is used as for the fuel meshing, so that the cladding boundary cells at $i = N_f$ and $i = N - 1$ have a radial width of $\frac{1}{2}\Delta r_c$, while cells in the cladding bulk have a uniform radial width $\Delta r_c = (r_{cO} - r_{cI}) / (N_c - 1)$. The cladding mesh structure thus also requires $N_c \geq 2$. While this is by no means the only possible geometric discretization, having boundary cells centers coinciding with the fuel and cladding boundary surfaces simplifies the application of boundary conditions.

Please note that the cell centers, where field values are stored, are indexed with integer indices i , while cell radial surface boundaries are indexed with half-integer indices $i \pm \frac{1}{2}$. The only exceptions are represented by boundary cells at $i = 0$, $i = N_f - 1$, $i = N_f$, $i = N$, for which cell centers coincide with one of the cell radial boundaries.

Each mesh cell Ω_i , while represented in 1-D, still maintains a well defined 3-D volume in a Finite Volume context. With respect to the mesh defined in Figure A.1.1, integer indices denote cell centers, while half-integers denote the radial surfaces separating adjacent cells, so that a mesh cell centered around r_i is radially bound by two surfaces at $r_{i-\frac{1}{2}}$ and $r_{i+\frac{1}{2}}$. Thus, the volume of each mesh cell Ω_i is obtained as (in cylindrical coordinates):

$$V_i = \int_{\Omega_i} dV = \int_{\Omega_i} r dr d\theta dz = \Delta\theta\Delta z \frac{1}{2} \left(r_{i+\frac{1}{2}}^2 - r_{i-\frac{1}{2}}^2 \right) \quad (\text{A.1.4})$$

where $\Delta\theta$ is the azimuthal width of the cells over the 1-D domain. Similarly, the surface area vectors of the radial mesh boundaries separating cell mesh i from adjacent cells are given by:

$$\mathbf{S}_{i\pm\frac{1}{2}} = \int_{\partial\Omega_{i\pm\frac{1}{2}}} d\mathbf{S} = \int_{\partial\Omega_{i\pm\frac{1}{2}}} \mathbf{n}_{i\pm\frac{1}{2}} r d\theta dz = \pm r_{i\pm\frac{1}{2}} \Delta\theta\Delta z \mathbf{e}_r = \pm S_{i\pm\frac{1}{2}} \mathbf{e}_r \quad (\text{A.1.5})$$

in which we recall that for cell i , the surface normals $\mathbf{n}_{i\pm\frac{1}{2}}$ are always outward-directed from the cell, so that $\mathbf{n}_{i+\frac{1}{2}} = \mathbf{e}_r$ and $\mathbf{n}_{i-\frac{1}{2}} = -\mathbf{e}_r$, with \mathbf{e}_r being the radial basis vector of the radial mesh direction.

A.1.3 Equation discretization

Let us consider a heat diffusion equation in the form introduced in equation A.1.2:

$$\rho c_p \frac{\partial}{\partial t} T^* - \kappa \nabla \cdot (\nabla T^*) = q \quad (\text{A.1.6})$$

where in the $*$ superscript indicates that said quantity represents a continuous physical field, similarly to the notation introduced when discussing the FVM in section 2.2. As this equation is fundamentally the same whether it is solved over the fuel or cladding domain (with $q^* = 0$ over the cladding), all further subscripts pertaining to the fuel or cladding are suppressed in this sub-section.

Let us apply the procedure outlined in section 2.2 to discretize equation A.1.6 via the FVM. For a generic mesh cell domain Ω_i of the mesh introduced in sub-section A.1.2 and a time step between times t and $t + \Delta t$, integrating said equation over those domains equates to a term-by-term integration:

$$\int_t^{t+\Delta t} \int_{\Omega_i} \rho c_p \frac{\partial}{\partial t} T^* dV dt = V_i \rho c_p \int_t^{t+\Delta t} \frac{\partial}{\partial t} T_i dt \quad (\text{A.1.7})$$

in which we have taken advantage of the notation introduced in 2.2.1, namely that for a generic continuous quantity ϕ^* , its volume average over a mesh cell domain Ω_i is represented as:

$$\phi_i = \frac{1}{V_i} \int_{\Omega_i} \phi^*(\mathbf{x}, t) dV \quad (\text{A.1.8})$$

In the context of the implemented model, a backward Euler scheme is used for the temporal discretization, so that:

$$\frac{\partial}{\partial t} T_i = \frac{T_i(t + \Delta t) - T_i(t)}{\Delta t} = \frac{T_i^n - T_i^o}{\Delta t} \quad (\text{A.1.9})$$

where the superscripts n and o are used to denote new and old time step values. By considering equations A.1.9 and A.1.4, the time and volume integrated time derivative term becomes:

$$\int_t^{t+\Delta t} \int_{\Omega_i} \rho c_p \frac{\partial}{\partial t} T^* dV dt = \Delta \theta \Delta z \frac{1}{2} \left(r_{i+\frac{1}{2}}^2 - r_{i-\frac{1}{2}}^2 \right) \rho c_p (T_i^n - T_i^o) \quad (\text{A.1.10})$$

Let us consider the time and volume integrated diffusive term by taking advantage of the Gauss theorem as seen in section 2.2:

$$\int_t^{t+\Delta t} \int_{\Omega_i} \kappa \nabla \cdot (\nabla T^*) dV dt = \kappa \int_t^{t+\Delta t} \int_{\partial \Omega_i} \nabla T^* \cdot \mathbf{dS} dt \quad (\text{A.1.11})$$

Given that our mesh is fundamentally 1-D and thus bound by surfaces only in the radial direction, the surface integral over $\partial \Omega_i$, namely the boundary of cell i centered at r_i reduces to the sum of the surface integrals over boundary at $r_{i+\frac{1}{2}}$, denoted as $\partial \Omega_i^+$, and over the boundary at $r_{i-\frac{1}{2}}$, denoted as $\partial \Omega_i^-$. We define a surface-normal gradient as:

$$\nabla_{\perp} T_{i \pm \frac{1}{2}} = \frac{1}{S_{i \pm \frac{1}{2}}} \int_{\partial \Omega_i^{\pm}} \nabla T^* \cdot \mathbf{dS} \quad (\text{A.1.12})$$

with the surface areas $S_{i \pm \frac{1}{2}}$ defined as of equation A.1.5. To enable the computation of this term, the surface normal is approximated via a central difference scheme between the two cells i and $i \pm 1$ sharing the face $i \pm \frac{1}{2}$. Given the uniformity of the mesh within each of the two domains, fuel and cladding, and given that the backward nature of the time discretization scheme requires all terms to be evaluated at the new time step $t + \Delta t$, the following is obtained:

$$\nabla_{\perp} T_{i+\frac{1}{2}} \approx \frac{T_{i+\frac{1}{2}}^n - T_i^n}{\Delta r} \quad (\text{A.1.13})$$

Please note that the formulation provided by A.1.13, while possibly counter-intuitive for the $i - \frac{1}{2}$ face (as one would be more prone to approximate it as $(T_i^n - T_{i-i}^n)/\Delta r$), is a valid approximation as this is the surface-*normal* gradient,

and the surface normal is anti-parallel to the radial mesh basis vector \mathbf{e}_r on face $i - \frac{1}{2}$. Further more, this approximation is valid at both cell faces *only for cells that are not boundary cells*, i.e. $i = 0$, $i = N_f - 1$, $i = N_f$, $i = N$. The gradient on these faces is provided instead by adequate boundary conditions, that will be discussed in sub-section A.1.4.

By considering equations A.1.11, A.1.12, A.1.13, the time and volume integrated diffusive term can be calculated as:

$$\begin{aligned} & \int_t^{t+\Delta t} \int_{\Omega_i} \kappa \nabla \cdot (\nabla T^*) dV dt = \\ & \approx \Delta t \kappa \left(\nabla_{\perp} T_{i+\frac{1}{2}} S_{i+\frac{1}{2}} + \nabla_{\perp} T_{i-\frac{1}{2}} S_{i-\frac{1}{2}} \right) = \\ & = \frac{\Delta t \kappa}{\Delta r} \left(T_{i+1}^n S_{i+\frac{1}{2}} + T_{i-1}^n S_{i-\frac{1}{2}} - T_i^n \left(S_{i+\frac{1}{2}} + S_{i-\frac{1}{2}} \right) \right) \end{aligned} \quad (\text{A.1.14})$$

To obtain the final formulation, the surface area definitions from A.1.5 are employed in combination with the fact that, over uniform meshes, $r_{i\pm\frac{1}{2}} = r_i \pm \frac{1}{2}\Delta r$ (in either the fuel or cladding domains), so that:

$$\begin{aligned} & \int_t^{t+\Delta t} \int_{\Omega_i} \nabla \cdot (\kappa^* \nabla T^*) dV dt = \\ & \approx \Delta z \Delta \theta \frac{\Delta t \kappa}{\Delta r} \left(T_{i+1}^n r_{i+\frac{1}{2}} + T_{i-1}^n r_{i-\frac{1}{2}} - 2T_i^n r_i \right) \end{aligned} \quad (\text{A.1.15})$$

With regard to the power density term, its time and volume integral is evaluated in virtue of A.1.4 and A.1.8 as:

$$\int_t^{t+\Delta t} \int_{\Omega_i} q dV dt = \Delta \theta \Delta z \frac{1}{2} \left(r_{i+\frac{1}{2}}^2 - r_{i-\frac{1}{2}}^2 \right) q \quad (\text{A.1.16})$$

By taking advantage of the results obtained so far, namely equations A.1.10, A.1.15 and A.1.16, the time and volume integrated heat diffusion equation over a generic mesh cell Ω_i *that is not a boundary cell* can be obtained:

$$\begin{aligned} & \frac{1}{2\Delta t} \left(r_{i+\frac{1}{2}}^2 - r_{i-\frac{1}{2}}^2 \right) \rho c_p (T_i^n - T_i^o) + \\ & - \frac{\kappa}{\Delta r} \left(T_{i+1}^n r_{i+\frac{1}{2}} + T_{i-1}^n r_{i-\frac{1}{2}} - 2T_i^n r_i \right) = \\ & = \frac{1}{2} \left(r_{i+\frac{1}{2}}^2 - r_{i-\frac{1}{2}}^2 \right) q \end{aligned} \quad (\text{A.1.17})$$

in which the Δz and $\Delta \theta$ terms could be cancelled out, as expected from a 1-D model. Please note that this is an algebraic relationship that relates the (unknown) new time step temperature T_i^n within cell i to the (unknown) new time step temperatures $T_{i\pm 1}^n$ in the neighbouring cells. Thus, the heat diffusion

problem is reduced to a set of coupled linear algebraic relationships, that can be represented in matrix form. To this end, equation A.1.17 is re-arranged to clearly highlight the matrix coefficients:

$$\begin{aligned}
& T_{i+1}^n \left(-\kappa \frac{r_{i+\frac{1}{2}}}{\Delta r} \right) + \\
& + T_i^n \left(\frac{\rho c_p}{2\Delta t} \left(r_{i+\frac{1}{2}}^2 - r_{i-\frac{1}{2}}^2 \right) + 2\kappa \frac{r_i}{\Delta r} \right) + \\
& + T_{i-1}^n \left(-\kappa \frac{r_{i-\frac{1}{2}}}{\Delta r} \right) = \\
& = \frac{1}{2} \left(r_{i+\frac{1}{2}}^2 - r_{i-\frac{1}{2}}^2 \right) \left(q + \frac{\rho c_p}{\Delta t} T_i^o \right)
\end{aligned} \tag{A.1.18}$$

The coefficient matrix is thus tri-diagonal, as for each row i there are three elements in the $i-1, i, i+1$ columns. The explicit source terms, inclusive of the old time temperature contribution to the time derivative, have been moved to the right-hand side (RHS). As a sanity check, note that all the temperature coefficients have the same dimensions, namely $W/m/K$. Equation A.1.18 is valid both within the fuel and the cladding (where $q_i = 0$), as long as the adequate material properties for ρ, c_p, κ are used.

As the FVM was here applied without consideration of possible boundary conditions, equation A.1.18 applies everywhere except at the inner, outer fuel surfaces and inner, outer cladding surfaces, namely $i = 0, i = N_f - 1, i = N_f, i = N$.

A.1.4 Boundary conditions and fuel-cladding coupling

To obtain the formulation of equation A.1.18, the FVM is applied at the boundary cells by taking boundary conditions into consideration. A separate discussion for each boundary is made.

Inner fuel surface This is the boundary at $r = r_{fI}$, which coincides with the cell center of cell 0. If $r_{fI} = 0$, (i.e. if the fuel has no central hole), a null Neumann (i.e. zero gradient) boundary condition applies, as $r_{fI} = 0$ geometrically represents the 1-D axial fuel centerline, which cannot store energy due to its dimensionality. If $r_{fI} > 0$ a zero gradient boundary conditions is still valid as approximation 6 introduced in sub-section A.1.1 assumes the absolute gas heat capacities in the hole and gap are negligible. A negligible heat capacity means that negligible amounts of heat flow through the inner fuel surface as a consequence of possible temperature imbalances between the hole gas and the fuel, so that the hole gas can be assumed to be isothermal with the fuel.

Let us apply the FVM to the discretization of the fuel governing equation, namely A.1.2. With regards to the time derivative, the following is obtained:

$$\int_t^{t+\Delta t} \int_{\Omega_0} \rho_f c_{p,f} \frac{\partial}{\partial t} T^* dV dt = \Delta\theta \Delta z \frac{1}{2} \left(r_{\frac{1}{2}}^2 - r_0^2 \right) \rho_f c_{p,f} (T_0^n - T_0^o) \tag{A.1.19}$$

which is equivalent to equation A.1.10 with the difference lying in the value of the cell volume. Note that $r_0 \equiv r_{fI}$ as r_0 denotes the radius of the first cell center, which coincides with the inner fuel surface. The same holds for the integral of the power density term, so that:

$$\int_t^{t+\Delta t} \int_{\Omega_0} q \, dV dt = \Delta\theta \Delta z \frac{1}{2} \left(r_{\frac{1}{2}}^2 - r_0^2 \right) q \quad (\text{A.1.20})$$

With regards to the diffusive term, the application of the FVM leads to the same results outlined by equation A.1.11. By introducing the same surface-normal gradients defined by equation A.1.12 and by substituting left cell face label $-\frac{1}{2}$ with the label 0 (since the left cell face and cell center coincide at $i = 0$), equation A.1.11 is expanded as:

$$\kappa_f \int_t^{t+\Delta t} \int_{\partial\Omega_i} \nabla T^* \cdot \mathbf{dS} dt = \kappa_f \int_t^{t+\Delta t} \left(S_0 \nabla_{\perp} T_0 + S_{\frac{1}{2}} \nabla_{\perp} T_{\frac{1}{2}} \right) dt \quad (\text{A.1.21})$$

While equation A.1.13 remains a valid approximation for $\nabla_{\perp} T_{\frac{1}{2}}$, the temperature gradient at the inner fuel surface $\nabla_{\perp} T_0$ is given by the boundary condition, namely $\nabla_{\perp} T_0 = 0$. By taking advantage of these observations and that $S_{\frac{1}{2}} = \Delta z \Delta\theta r_{\frac{1}{2}}$, the diffusive term can then be thus expressed as:

$$\begin{aligned} & \kappa_f \int_t^{t+\Delta t} \int_{\partial\Omega_i} \nabla T^* \cdot \mathbf{dS} dt \\ & \approx \Delta z \Delta\theta \frac{\Delta t \kappa_f}{\Delta r_f} \left(T_1^n r_{\frac{1}{2}} - T_0^n r_{\frac{1}{2}} \right) \end{aligned} \quad (\text{A.1.22})$$

By proceeding with the same steps outlined in the previous sub-section with the aid of equations A.1.19, A.1.20, A.1.22, it is easy to show that the final formulation of the discretized heat diffusion equation at $i = 0$ is:

$$\begin{aligned} & T_1^n \left(-\kappa_f \frac{r_{\frac{1}{2}}}{\Delta r_f} \right) + \\ & + T_0^n \left(\frac{\rho_f c_{p,f}}{2\Delta t} \left(r_{\frac{1}{2}}^2 - r_0^2 \right) + \kappa_f \frac{r_{\frac{1}{2}}}{\Delta r_f} \right) = \\ & = \frac{1}{2} \left(r_{\frac{1}{2}}^2 - r_0^2 \right) \left(q + \frac{\rho_f c_{p,f}}{\Delta t} T_0^o \right) \end{aligned} \quad (\text{A.1.23})$$

Outer fuel surface This is the boundary at $r = r_{fO}$, which coincides with the cell center of cell $N_f - 1$. In virtue approximation 6 introduced in sub-section A.1.1, the gap has a negligible heat capacity, so that an instantaneous heat flux balance can be assumed at the outer fuel surface. Thus:

$$-\kappa_f \nabla_{\perp} T_{N_f-1} = H_g (T_{N_f-1}^n - T_{N_f}^n) \quad (\text{A.1.24})$$

where H_g is the heat conductance (i.e. heat transfer coefficient) of the gap and we recall that T_{N_f-1} , T_{N_f} and the fuel inner surface and cladding outer surface temperatures respectively.

Let us proceed by applying the FVM as seen before. The integrated temporal derivative term is:

$$\begin{aligned} & \int_t^{t+\Delta t} \int_{\Omega_{N_f-1}} \rho_f c_{p,f} \frac{\partial}{\partial t} T^* dV dt = \\ & = \Delta\theta \Delta z \frac{1}{2} \left(r_{N_f-1}^2 - r_{N_f-1-\frac{1}{2}}^2 \right) \rho_f c_{p,f} \left(T_{N_f-1}^n - T_{N_f-1}^o \right) \end{aligned} \quad (\text{A.1.25})$$

The integrated power density is:

$$\int_t^{t+\Delta t} \int_{\Omega_{N_f-1}} q dV dt = \Delta\theta \Delta z \frac{1}{2} \left(r_{N_f-1}^2 - r_{N_f-1-\frac{1}{2}}^2 \right) q \quad (\text{A.1.26})$$

The integrated diffusive term, similarly to what seen for equation A.1.21 and with the label change of $N_f - 1 + \frac{1}{2}$ to $N_f - 1$ as the position of the right cell boundary coincides with the cell center of $N_f - 1$, is expanded as:

$$\begin{aligned} & \kappa_f \int_t^{t+\Delta t} \int_{\partial\Omega_i} \nabla T^* \cdot \mathbf{dS} dt = \\ & = \kappa_f \int_t^{t+\Delta t} \left(S_{N_f-1-\frac{1}{2}} \nabla_{\perp} T_{N_f-1-\frac{1}{2}} + S_{N_f-1} \nabla_{\perp} T_{N_f-1} \right) dt \end{aligned} \quad (\text{A.1.27})$$

Similarly to what was seen for the inner fuel surface, the surface gradient $\nabla_{\perp} T_{N_f-1-\frac{1}{2}}$ can still be treated via equation A.1.13, while the surface gradient at the fuel outer surface $\nabla_{\perp} T_{N_f-1}$ is to be treated via the boundary condition A.1.24. Thus:

$$\begin{aligned} & \kappa_f \int_t^{t+\Delta t} \int_{\partial\Omega_i} \nabla T^* \cdot \mathbf{dS} dt = \\ & \approx T_{N_f}^n \Delta z \Delta\theta \Delta t H_g r_{N_f-1} + \\ & + T_{N_f-1}^n \Delta z \Delta\theta \Delta t \left(-H_g r_{N_f-1} - \kappa_f \frac{r_{N_f-1-\frac{1}{2}}}{\Delta r_f} \right) + \\ & + T_{N_f-2}^n \Delta z \Delta\theta \Delta t \kappa_f \frac{r_{N_f-1-\frac{1}{2}}}{\Delta r_f} \end{aligned} \quad (\text{A.1.28})$$

By taking advantage of these results, equations A.1.25, A.1.26, A.1.28, the

final discretized heat diffusion equation for cell $N_f - 1$ becomes:

$$\begin{aligned}
& T_{N_f}^n (-H_g r_{N_f-1}) + \\
& + T_{N_f-1}^n \left(\frac{\rho_f c_{p,f}}{2\Delta t} \left(r_{N_f-1}^2 - r_{N_f-1-\frac{1}{2}}^2 \right) + H_g r_{N_f-1} + \kappa_f \frac{r_{N_f-1-\frac{1}{2}}}{\Delta r_f} \right) + \\
& + T_{N_f-2}^n \left(-\kappa_f \frac{r_{N_f-1-\frac{1}{2}}}{\Delta r_f} \right) = \\
& = \frac{1}{2} \left(r_{N_f-1}^2 - r_{N_f-1-\frac{1}{2}}^2 \right) \left(q_{f,N_f-1} + \frac{\rho_f c_{p,f}}{\Delta t} T_{N_f-1}^o \right)
\end{aligned} \tag{A.1.29}$$

where we recall that $r_{N_f-1} \equiv r_{fO}$ and that $r_{N_f-1-\frac{1}{2}} = r_{N_f-1} - \frac{1}{2}\Delta r_f$.

Inner cladding surface This is the boundary at $r = r_{cI}$, which coincides with the cell center of cell N_f . Energy conservation consideration apply, namely that the total heat *flow* from the fuel outer surface must be equal to the total heat *flow* to the cladding inner surface. This discussion must be framed in terms of flow as the inner and outer fuel and cladding surface areas are not equal, and thus equating heat fluxes would not ensure energy conservation. Thus

$$-S_{N_f-1}\kappa_f \nabla_{\perp} T_{N_f-1} = S_{N_f} k_c \nabla_{\perp} T_{N_f} \tag{A.1.30}$$

The sign difference arises from the fact that these are the surface-*normal* gradient, and the surface normals of the outer fuel surface and the inner cladding surface are anti-parallel (surface normals are defined as outward from the domain they enclose). By combining equation A.1.30 with equation A.1.24 to relate the surface temperature gradient to the gap conductance:

$$\kappa_c \nabla_{\perp} T_{N_f} = \frac{S_{N_f-1}}{S_{N_f}} H_g (T_{N_f-1}^n - T_{N_f}^n) = \frac{r_{N_f-1}}{r_{N_f}} H_g (T_{N_f-1}^n - T_{N_f}^n) \tag{A.1.31}$$

By proceeding in the same manner as for the outer fuel surface yet by taking advantage of A.1.31 and considering that $q^* = 0$ in the cladding, it can be shown that the final formulation of the heat diffusion equation at $i = N_f$ is:

$$\begin{aligned}
& T_{N_f+1}^n \left(-\kappa_c \frac{r_{N_f+\frac{1}{2}}}{\Delta r_c} \right) + \\
& + T_{N_f}^n \left(\frac{\rho_c c_{p,c}}{2\Delta t} \left(r_{N_f+\frac{1}{2}}^2 - r_{N_f}^2 \right) + H_g r_{N_f-1} + \kappa_c \frac{r_{N_f+\frac{1}{2}}}{\Delta r_c} \right) + \\
& + T_{N_f-1}^n (-H_g r_{N_f-1}) = \\
& = \frac{1}{2} \left(r_{N_f+\frac{1}{2}}^2 - r_{N_f}^2 \right) \frac{\rho_f c_{p,c}}{\Delta t} T_{N_f}^o
\end{aligned} \tag{A.1.32}$$

Cladding outer surface This is the boundary at $r = r_{cO}$, which coincides with the cell center of cell $N - 1$. As per the assumptions made in sub-section A.1.1, the fluid flow properties outside the cladding are characterized by an average fluid temperature T_m and an average cladding fluid heat transfer coefficient H_m . Please note that this approach is valid even if the fluid consists of a two-phase flows, as long as mixture values for the fluid temperature and heat transfer coefficient are used, as discussed in section 3.2.4. Thus, a convective boundary condition at the outer cladding surface would prescribe:

$$-S_{N-1}\kappa_c \nabla_{\perp} T_{N-1} = H_m (T_{N-1}^n - T_m) \quad (\text{A.1.33})$$

By proceeding in the same way as for the fuel outer surface, the FVM discretization of the cladding heat diffusion equation in cell $N - 1$ is obtained as:

$$\begin{aligned} & T_{N-1}^n \left(\frac{\rho_c c_{p,c}}{2\Delta t} \left(r_{N-1}^2 - r_{N-1-\frac{1}{2}}^2 \right) + H_m r_{N-1} + \kappa_c \frac{r_{N-1-\frac{1}{2}}}{\Delta r_c} \right) + \\ & + T_{N-2}^n \left(-\kappa_c \frac{r_{N-1-\frac{1}{2}}}{\Delta r_c} \right) = \\ & = \frac{1}{2} \left(r_{N-1}^2 - r_{N-1-\frac{1}{2}}^2 \right) \frac{\rho_c c_{p,c}}{\Delta t} T_{N-1}^o + T_m H_m r_{N-1} \end{aligned} \quad (\text{A.1.34})$$

Where it is recalled that $r_{N-1} \equiv r_{cO}$ and $r_{N-1-\frac{1}{2}} = r_{N-1} - \frac{1}{2}\Delta r_c$.

A.1.5 Discretization summary and final remarks

A summary of the discretization process is presented for clarity. The coupled system of equations A.1.2, A.1.3 is discretized over a 1-D mesh represented in Figure A.1.1. This domain consists of two separate domains, one over the fuel discretized with N_f cells, and one over the cladding discretized with N_c cells, though a single mesh indexing is adopted for the whole mesh, thus consisting of $N = N_f + N_c$ cells labelled from 0 to $N - 1$. Four boundaries exist at the inner and outer fuel and cladding surfaces. As cells adjacent to boundaries have their cell centers located at the boundary surfaces, the boundary cells are indicated with indices 0, $N_f - 1$, N_f , $N - 1$.

The coupled system of equation is discretized and linearized so to be represented as:

$$\hat{M}T = b \quad (\text{A.1.35})$$

where \hat{M} is the discretized heat transport operator of size $N \times N$, T is the solution radial temperature field consisting of the new time step values T^n , and b is the source field, both described as a $N \times 1$ arrays. In virtue of the results shown in sub-sections A.1.2, A.1.3, the structure of A.1.35 is discussed row by row:

- row 0 represents the heat transport equation in the cell at the fuel inner surface. The coefficients of columns 0 and 1 of \hat{M} consist of the coefficients of T_0^n , T_1^n respectively as shown in equation A.1.23. The source term b_0 consists of the RHS of equation A.1.23;
- row $i \forall i \in [1, N_f - 2]$ represents the heat transport equation in the i -th cell belonging to the fuel bulk. The coefficients of columns $i - 1$, i , $i + 1$ of \hat{M} consist of the coefficients of T_{i-1}^n , T_i^n , T_{i+1}^n respectively as shown in equation A.1.18 where thermo-physical properties ρ , c_p , κ are that of the fuel. The source term b_i consists of the RHS of equation A.1.18;
- row $N_f - 1$ represents the heat transport equation in the cell at the fuel outer surface. The coefficients of columns $N_f - 2$, $N_f - 1$, N_f of \hat{M} consist of the coefficients of $T_{N_f-2}^n$, $T_{N_f-1}^n$, $T_{N_f}^n$ respectively as shown in equation A.1.29. The source term b_{N_f-1} consists of the RHS of equation A.1.29;
- row N_f represents the heat transport equation in the cell at the cladding inner surface. The coefficients of columns $N_f - 1$, N_f , $N_f + 1$ of \hat{M} consist of the coefficients of $T_{N_f-1}^n$, $T_{N_f}^n$, $T_{N_f+1}^n$ respectively as shown in equation A.1.32. The source term b_{N_f} consists of the RHS of equation A.1.32;
- row $i \forall i \in [N_f + 1, N - 2]$ represents the heat transport equation in the i -th cell belonging to the cladding bulk. The coefficients of columns $i - 1$, i , $i + 1$ of \hat{M} consist of the coefficients of T_{i-1}^n , T_i^n , T_{i+1}^n respectively as shown in equation A.1.18 where thermo-physical properties ρ , c_p , κ are that of the cladding and $q_i = 0$. The source term b_i consists of the RHS of equation A.1.18;
- row $N - 1$ represents the heat transport equation in the cell at the cladding outer surface. The coefficients of columns $N - 2$ and $N - 1$ of \hat{M} consist of the coefficients of T_{N-2}^n , T_{N-1}^n respectively as shown in equation A.1.34. The source term b_{N-1} consists of the RHS of equation A.1.34.

For each time step and/or outer iteration, the solution of the system represented by A.1.35 via any user-selected OpenFOAM iterative matrix solver will update the pin temperature profile.

Note that since the discretization is based on the FVM, it is inherently (energy) conservative regardless of mesh density and regardless of the magnitude of the value of the cladding-fluid heat transfer coefficient H_m . This is important to highlight as a Finite Difference discretization of the problem will lead to a non-conservative formulation instead, whose non-conservation issues worsen as the 1-D mesh is coarsened and/or as the cladding-fluid heat transfer coefficient H_m is reduced (which might happen e.g. in boiling simulations

involving dry-out). This latter aspect is a consequence of the problem being ultimately bound at the inner fuel and outer cladding surfaces by boundary conditions of the first (Neumann, i.e. zero gradient) or second (Robin, i.e. convective) type, which operate exclusively over the gradient of the field and do not provide an absolute value reference for the temperature field.

The FVM-based formulation presented so far is the one ultimately implemented in the code, and represents a significant improvement over the formulation of the original GeN-Foam code thermal-hydraulics. The latter adopted a Finite Difference discretization scheme and solved the two discretized equations over the fuel and cladding separately, with an explicit evaluation of the fuel-cladding heat fluxes for coupling purposes that was performed only once per time step and/or iteration.

A.2 Coarse-mesh integration

The 1-D nuclear fuel pin model presented in section A.1 clearly operates as a stand-alone model, requiring only scalar values for average fluid flow conditions at the cladding. Let us briefly discuss how this model is then treated in a coarse-mesh context.

Let us consider a bundle of pins organized in a certain lattice and whose axes are aligned along an axis z . Let us focus our attention on a representative elementary volume (REV) of volume V of the lattice of height Δz that is centered on a sub-channel, i.e. that contains a single axial slice of a single fuel pin, whose cladding measures r_{cO} in radius. By applying the volume averaging process as discussed in section 2.2, the resulting pin bundle will be characterized, within that REV, by a certain volume fraction obtained as:

$$\alpha_s = \frac{\Delta z \pi r_{cO}^2}{V} \quad (\text{A.2.1})$$

The surface area density of the pin will be:

$$A''_s = \frac{\Delta z 2\pi r_{cO}}{V} \quad (\text{A.2.2})$$

Thus, for a lattice of cylindrical pins, the interfacial area density and pin volume fraction per cell are related as:

$$A''_s = \frac{2\alpha_s}{r_{cO}} \quad (\text{A.2.3})$$

Please note that this result is independent of the geometric definition of the REV (e.g. it does not need to be a pin-centered sub-channel), as long as it is an REV, meaning that its a geometric basis for the lattice. Furthermore, if one were to perform the averaging process over the entire representative volume (RV) over which the lattice exists, one would obtain the same results,

as the RV can be geometrically composed starting from any of the possible REVs.

Let us now use a collection of REVs as a computational mesh for the simulation of fluid flow and heat transfer in said lattice, over its overall RV. Within mesh cell we construct a 1-D nuclear fuel problem in the form of the system A.1.35, which is thus meant to represent the thermal evolution of a representative nuclear fuel pin axial slice in said cell. The average fluid temperature T_m and fluid-cladding heat transfer coefficients H_m that are required for the solution of the 1-D problem in each consist precisely of the fluid temperature and heat transfer coefficient values in the cell under consideration (in a two-phase scenario, these consist of mixture values as discussed in section 3.2.4). Conversely, the structure surface temperature used to define fluid-structure heat fluxes when solving the fluid energy equation(s) consists of the cladding outer temperature $T_{N-1} \equiv T_{cO} \equiv T_s$ for each mesh cell, with T being the 1-D pin temperature profile that is obtained by the solution of A.1.35 *within each mesh cell*.

Two important observations conclude this section. The first concerns energy conservation. The 1-D nuclear fuel pin model calculates a pin temperature profile that ultimately depends on the fluid-cladding heat *flux* at the outer cladding outer surface, which thus depends, among others, on the outer cladding radius r_{cO} . Conversely, the fluid-structure heat source term that figures in the fluid enthalpy equation(s) is represented as a heat flow per unit *volume* of fluid, which thus depends on both the fluid phase fraction (which can be related to the structure phase fraction α_s) as well as the structure (i.e. cladding) surface area density. It is thus clear that these three variables cannot be completely independent if energy conservation is to be ensured, and these variables, namely α_s , A''_s , r_{cO} are related precisely by A.2.3.

The second observation is that due to the averaging process, if the mesh over the lattice RV were to consist of a generic unstructured mesh rather than of a mesh whose cells consist of the lattice REVs, energy-conservation would still be assured in virtue of the FVM, save for discretization errors in the power density field q .

Bibliography

- [1] United States Nuclear Regulatory Commission, Computer codes, <https://www.nrc.gov/about-nrc/regulatory/research/safetycodes.html>.
- [2] J. Wang, Q. Wang, M. Ding, Review on neutronic/thermal-hydraulic coupling simulation methods for nuclear reactor analysis, *Annals of Nuclear Energy* 137 (2020) 107165. doi:<https://doi.org/10.1016/j.anucene.2019.107165>.
URL <http://www.sciencedirect.com/science/article/pii/S0306454919306759>
- [3] A. Mylonakis, M. Varvayanni, N. Catsaros, P. Savva, D. Grigoriadis, Multi-physics and multi-scale methods used in nuclear reactor analysis, *Annals of Nuclear Energy* 72 (2014) 104 – 119. doi:<https://doi.org/10.1016/j.anucene.2014.05.002>.
URL <http://www.sciencedirect.com/science/article/pii/S0306454914002163>
- [4] GRS gGmbH, ATHLET 3.2 Program Overview (2019), https://www.grs.de/sites/default/files/pdf/athlet_3.2_overview.pdf.
- [5] Idaho National Laboratory (Idaho, U.S.), RELAP5-3D, <https://relap53d.inl.gov/SitePages/Home.aspx>.
- [6] United States Nuclear Regulatory Commission, TRACE v5.0 Theory Manual, <https://www.nrc.gov/docs/ML1200/ML120060218.pdf>.
- [7] A. Moorthi, A. Kumar Sharma, K. Velusamy, A review of sub-channel thermal hydraulic codes for nuclear reactor core and future directions, *Nuclear Engineering and Design* 332 (2018) 329 – 344. doi:<https://doi.org/10.1016/j.nucengdes.2018.03.012>.
URL <http://www.sciencedirect.com/science/article/pii/S0029549318302681>
- [8] R. Salko, M. Avramova, A. Wysocki, A. Toptan, J. Hu, N. Porter, T. Blyth, C. Dances, A. Gomez, C. Jernigan, J. Kelly, A. Abarca, Ctf theory manual (2019).
URL https://vera.ornl.gov/wp-content/uploads/2020/07/CASL-U-2019-1886-001_CTF-Theory-Manual.pdf

- [9] CASL, Consortium for Advanced Simulation of Light Water Reactors.
URL <http://www.casl.gov>
- [10] M. Thurgood, Cobra-tf development, in: proceedings of the 8th Water Reactor Safety Information Meeting, 1980.
- [11] H. Ninokata, T. aki Okano, Sabena: Subassembly boiling evolution numerical analysis, Nuclear Engineering and Design 120 (2) (1990) 349 – 367. doi:[https://doi.org/10.1016/0029-5493\(90\)90386-C](https://doi.org/10.1016/0029-5493(90)90386-C).
URL <http://www.sciencedirect.com/science/article/pii/002954939090386C>
- [12] J. Macdougall, J. Lillington, The sabre code for fuel rod cluster thermohydraulics, Nuclear Engineering and Design 82 (2) (1984) 171 – 190. doi:[https://doi.org/10.1016/0029-5493\(84\)90210-3](https://doi.org/10.1016/0029-5493(84)90210-3).
URL <http://www.sciencedirect.com/science/article/pii/0029549384902103>
- [13] U. Imke, V. Sanchez, Validation of the subchannel code subchanflow using the nupec pwr tests (psbt), Science and Technology of Nuclear Installations 2012 (465059) (2012). doi:<https://doi.org/10.1155/2012/465059>.
- [14] U. Bieder, H. Uitslag-Doolaard, B. Mikuž, Investigation of pressure loss and velocity distribution in fuel assemblies with wire-wrapped rods by using rans and les with wall functions, Annals of Nuclear Energy 152 (2021) 108025. doi:<https://doi.org/10.1016/j.anucene.2020.108025>.
URL <http://www.sciencedirect.com/science/article/pii/S0306454920307210>
- [15] G. Grötzbach, M. Wörner, Direct numerical and large eddy simulations in nuclear applications, International Journal of Heat and Fluid Flow 20 (3) (1999) 222 – 240. doi:[https://doi.org/10.1016/S0142-727X\(99\)00012-0](https://doi.org/10.1016/S0142-727X(99)00012-0).
URL <http://www.sciencedirect.com/science/article/pii/S0142727X99000120>
- [16] OECD-NEA, Technology roadmap update for Generation IV nuclear energy systems (2014).
URL <https://www.gen-4.org/gif/upload/docs/application/pdf/2014-03/gif-tru2014.pdf>
- [17] P. Hidalga, A. Abarca, R. Miró, A. Sekrhi, G. Verdú, A multi-scale and multi-physics simulation methodology with the state-of-the-art tools for safety analysis in light water reactors applied to a turbine trip scenario (part i), Nuclear Engineering and Design 350 (2019) 195–204.

doi:<https://doi.org/10.1016/j.nucengdes.2019.05.008>.
URL <https://www.sciencedirect.com/science/article/pii/S0029549319301177>

- [18] C. J. Permann, D. R. Gaston, D. Andrš, R. W. Carlsen, F. Kong, A. D. Lindsay, J. M. Miller, J. W. Peterson, A. E. Slaughter, R. H. Stogner, R. C. Martineau, MOOSE: Enabling massively parallel multiphysics simulation, SoftwareX 11 (2020) 100430. doi:<https://doi.org/10.1016/j.softx.2020.100430>.
URL <http://www.sciencedirect.com/science/article/pii/S2352711019302973>
- [19] OpenCFD Ltd., OpenFOAM.
URL <http://www.openfoam.com>
- [20] C. Fiorina, I. Clifford, M. Aufiero, K. Mikityuk, GeN-Foam: a novel OpenFOAM® based multi-physics solver for 2D/3D transient analysis of nuclear reactors, Nuclear Engineering and Design 294 (2015) 24 – 37. doi:<https://doi.org/10.1016/j.nucengdes.2015.05.035>.
- [21] H. Jasak, Error analysis and estimation for the finite volume method with applications to fluid flows, Ph.D. thesis, Impreial College, University of London (1996).
- [22] P. J. Roache, A method for uniform reporting of grid refinement studies, Journal of Fluids Engineering 116 (1994) 405–413. doi:<https://doi.org/10.1115/1.2910291>.
- [23] H. G. Weller, G. Tabor, H. Jasak, C. Fureby, A tensorial approach to computational continuum mechanics using object-oriented techniques, Computers in Physics 12 (6) (1998) 620–631. arXiv:<https://aip.scitation.org/doi/pdf/10.1063/1.168744>, doi: 10.1063/1.168744.
URL <https://aip.scitation.org/doi/abs/10.1063/1.168744>
- [24] K. Vafai, Handbook of Porous Media, 2nd Edition, CRC Press, 2005.
- [25] S. Whitaker, A simple geometrical derivation of the spatial averaging theorem, Chemical Engineering Education 19 (1) (1985).
- [26] E. Khan, W. Rohsenow, A. Sonin, N. Todreas, A porous body model for predicting temperature distribution in wire-wrapped fuel rod assemblies, Nuclear Engineering and Design 35 (1) (1975) 1 – 12. doi:[https://doi.org/10.1016/0029-5493\(75\)90076-X](https://doi.org/10.1016/0029-5493(75)90076-X).
URL <http://www.sciencedirect.com/science/article/pii/002954937590076X>

- [27] K. Vafai, Handbook of Porous Media, 2nd Edition, Taylor & Francis Group, 2005.
- [28] J. Bear, Y. Bachmat, Introduction to Modelling of Transport Phenomena in Porous Media, 1st Edition, Kluwer Academic Publishers, 1990.
- [29] F. Lehner, On the validity of fick's law for transient diffusion through a porous medium, Chemical Engineering Science 34 (6) (1979) 821 – 825.
doi:[https://doi.org/10.1016/0009-2509\(79\)85137-4](https://doi.org/10.1016/0009-2509(79)85137-4).
URL <http://www.sciencedirect.com/science/article/pii/0009250979851374>
- [30] S. Whitaker, The transport equations for multi-phase systems, Chemical Engineering Science 28 (1) (1973) 139 – 147.
doi:[https://doi.org/10.1016/0009-2509\(73\)85094-8](https://doi.org/10.1016/0009-2509(73)85094-8).
URL <http://www.sciencedirect.com/science/article/pii/0009250973850948>
- [31] W. G. Gray, A derivation of the equations for multi-phase transport, Chemical Engineering Science 30 (2) (1975) 229 – 233.
doi:[https://doi.org/10.1016/0009-2509\(75\)80010-8](https://doi.org/10.1016/0009-2509(75)80010-8).
URL <http://www.sciencedirect.com/science/article/pii/0009250975800108>
- [32] J.-H. Kim, J. A. Ochoa, S. Whitaker, Diffusion in anisotropic porous media, Transport in Porous Media 2 (4) (1987) 327 – 356. doi:<https://doi.org/10.1007/BF00136440>.
- [33] V. Zalizniak, 3 - solving systems of linear equations, in: V. Zalizniak (Ed.), Essentials of Scientific Computing, Woodhead Publishing, 2008, pp. 21 – 46. doi:<https://doi.org/10.1533/9780857099365.21>.
URL <http://www.sciencedirect.com/science/article/pii/B9781904275329500035>
- [34] D. Brkić, Review of explicit approximations to the colebrook relation for flow friction, Journal of Petroleum Science and Engineering 77 (1) (2011) 34 – 48. doi:<https://doi.org/10.1016/j.petrol.2011.02.006>.
URL <http://www.sciencedirect.com/science/article/pii/S092041051100043X>
- [35] H. No, M. Kazimi, Nuclear Systems I: Thermal Hydraulic Fundamentals, 1st Edition, CRC Press, 1989.
- [36] T. Watanabe, Y. Kukita, The effect of the virtual mass term on the stability of the two-fluid model against perturbations, Nuclear Engineering and Design 135 (3) (1992) 327–340. doi:[https://doi.org/10.1016/0029-5493\(92\)90200-F](https://doi.org/10.1016/0029-5493(92)90200-F).

- URL <https://www.sciencedirect.com/science/article/pii/002954939290200F>
- [37] R. Lahey, L. Cheng, D. Drew, J. Flaherty, The effect of virtual mass on the numerical stability of accelerating two-phase flows, International Journal of Multiphase Flow 6 (4) (1980) 281 – 294. doi:[https://doi.org/10.1016/0301-9322\(80\)90021-X](https://doi.org/10.1016/0301-9322(80)90021-X).
 URL <http://www.sciencedirect.com/science/article/pii/030193228090021X>
- [38] K. Mikityuk, Heat transfer to liquid metal: Review of data and correlations for tube bundles, Nuclear Engineering and Design 239 (4) (2009) 680 – 687. doi:<https://doi.org/10.1016/j.nucengdes.2008.12.014>.
- [39] G. Basque, M. Bottone, D. Grand, F. Huber, H. Kottowski-Dümenil, M. Giot, H. Ninokata, W. Peppler, B. Rameau, P. Riboud, K. Schleisiek, J. Seiler, N. Sheriff, D. Wall, R. Webster, J. Macdougall, J. Olive, Liquid Metal Thermal-hydraulics, INFORUM, 1994.
- [40] A. Chenu, K. Mikityuk, R. Chawla, TRACE simulation of sodium boiling in pin bundle experiments under loss-of-flow conditions, Nuclear Engineering and Design 239 (11) (2009) 2417 – 2429. doi:<https://doi.org/10.1016/j.nucengdes.2009.07.015>.
- [41] J. C. Chen, Correlation for boiling heat transfer to saturated fluids in convective flow, Industrial & Engineering Chemistry Process Design and Development 5 (3) (1966) 322–329.
 URL <https://doi.org/10.1021/i260019a023>
- [42] B. E. Launder, B. Sharma, Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc, Letters in heat and mass transfer 1 (2) (1974) 131–137.
- [43] A. Nakayama, F. Kuwahara, A macroscopic turbulence model for flow in a porous medium, Journal of fluids engineering 121 (2) (1999) 427–433. doi:<https://doi.org/10.1115/1.2822227>.
- [44] M. Chandesris, G. Serre, P. Sagaut, A macroscopic turbulence model for flow in porous media suited for channel, pipe and rod bundle flows, International Journal of Heat and Mass Transfer 49 (15) (2006) 2739 – 2750. doi:<https://doi.org/10.1016/j.ijheatmasstransfer.2005.12.013>.
- [45] ANSYS FLUENT 12.0 user manual.
 URL http://www.afs.enea.it/project/neptunius/docs/fluent/html/ug/main_pre.htm
- [46] T. R. Lahey, The simulation of multidimensional multiphase flows, Nuclear Engineering and Design 235 (2005) 1043–1060.

- [47] A. Pascau, C. Pérez, F. J. Serón, A comparison of segregated and coupled methods for the solution of the incompressible navier-stokes equations, Communications in Numerical Methods in Engineering 12 (10) (1996) 617–630. arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/%28SICI%291099-0887%28199610%2912%3A10%3C617%3A%3AAID-CNM10%3E3.0.CO%3B2-J>, doi:[https://doi.org/10.1002/\(SICI\)1099-0887\(199610\)12:10<617::AID-CNM10>3.0.CO;2-J](https://doi.org/10.1002/(SICI)1099-0887(199610)12:10<617::AID-CNM10>3.0.CO;2-J). URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291099-0887%28199610%2912%3A10%3C617%3A%3AAID-CNM10%3E3.0.CO%3B2-J>
- [48] S. Patankar, D. Spalding, A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows, Int. J. Heat Mass Transfer 15 (1972) 1787–1806.
- [49] R. Issa, A. Gosman, A. Watkins, The computation of compressible and incompressible recirculating flows by a non-iterative implicit scheme, Journal of Computational Physics 62 (1) (1986) 66 – 82. doi:[https://doi.org/10.1016/0021-9991\(86\)90100-2](https://doi.org/10.1016/0021-9991(86)90100-2). URL <http://www.sciencedirect.com/science/article/pii/0021999186901002>
- [50] S. Patankar, Numerical heat transfer and fluid flow, CRC Press, 1980.
- [51] R. Courant, K. Friedrichs, H. Lewy, Über die partiellen Differenzengleichungen der mathematischen Physik, Matematische Annalen 100 (1928) 32–74. URL <https://link.springer.com/article/10.1007/BF01448839>
- [52] D. B. Spalding, Numerical computation of multi-phase fluid flow and heat transfer, Vol. 1, 1980, pp. 139–167.
- [53] H. G. Weller, A new approach to vof-based interface capturing methods for incompressible and compressible flow, Tech. Rep. Report, OpenCFD Ltd., London (2008).
- [54] J. P. Boris, D. L. Book, Flux-corrected transport, Journal of Computational Physics 135 (2) (1997) 172 – 186. doi:<https://doi.org/10.1006/jcph.1997.5700>. URL <http://www.sciencedirect.com/science/article/pii/S0021999197957004>
- [55] D. Santiago Marquez, An extended mixture model for the simultaneous treatment of short and long scale interfaces, Ph.D. thesis, Universidad Nacional del Litoral (2013).
- [56] S. T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, Journal of Computational Physics 31 (3) (1979) 335 –

362. doi:[https://doi.org/10.1016/0021-9991\(79\)90051-2](https://doi.org/10.1016/0021-9991(79)90051-2).
URL <http://www.sciencedirect.com/science/article/pii/0021999179900512>
- [57] H. Karem, S. Lo, Efficiency of interphase coupling algorithms in fluidized bed conditions, *Computers & Fluids* 28 (3) (1999) 323 – 360. doi:[https://doi.org/10.1016/S0045-7930\(98\)00028-0](https://doi.org/10.1016/S0045-7930(98)00028-0).
URL <http://www.sciencedirect.com/science/article/pii/S0045793098000280>
- [58] R. Issa, B. Ahmadi-Befrui, K. Beshay, A. Gosman, Solution of the implicitly discretised reacting flow equations by operator-splitting, *Journal of Computational Physics* 93 (2) (1991) 388 – 410. doi:[https://doi.org/10.1016/0021-9991\(91\)90191-M](https://doi.org/10.1016/0021-9991(91)90191-M).
URL <http://www.sciencedirect.com/science/article/pii/002199919190191M>
- [59] A. Date, Fluid dynamical view of pressure checkerboarding problem and smoothing pressure correction on meshes with colocated variables, *International Journal of Heat and Mass Transfer* 46 (25) (2003) 4885–4898. doi:[https://doi.org/10.1016/S0017-9310\(03\)00332-6](https://doi.org/10.1016/S0017-9310(03)00332-6).
URL <https://www.sciencedirect.com/science/article/pii/S0017931003003326>
- [60] C. M. Rhie, W. L. Chow, Numerical study of the turbulent flow past an airfoil with trailing edge separation, *AIAA Journal* 21 (11) (1983) 1525–1532. arXiv:<https://doi.org/10.2514/3.8284>, doi:10.2514/3.8284.
URL <https://doi.org/10.2514/3.8284>
- [61] H. G. Weller, twoPhaseEulerEoam: Added experimental face-based momentum equation formulation (2015), <https://github.com/OpenFOAM/OpenFOAM-dev/commit/16f03f8a393bd4b6e70b548812777bb10a3e1dda>.
- [62] P. Brady, M. Herrmann, J. Lopez, Code verification for finite volume multiphase scalar equations using the method of manufactured solutions, *Journal of Computational Physics* 231 (7) (2012) 2924–2944. doi:10.1016/j.jcp.2011.12.040.
- [63] K. Salari, P. Knupp, Code verification by the method of manufactured solutions, Tech. Rep. SAND2000-1444, Sandia National Laboratories (2000).
URL <https://prod-ng.sandia.gov/techlib-noauth/access-control.cgi/2000/001444.pdf>
- [64] B. van Leer, Towards the ultimate conservative difference scheme. ii. monotonicity and conservation combined in a second-order

- scheme, *Journal of Computational Physics* 14 (4) (1974) 361 – 370. doi:[https://doi.org/10.1016/0021-9991\(74\)90019-9](https://doi.org/10.1016/0021-9991(74)90019-9).
URL <http://www.sciencedirect.com/science/article/pii/0021999174900199>
- [65] D. Padua (Ed.), *Message Passing Interface (MPI)*, Springer US, Boston, MA, 2011, pp. 1116–1116. doi:[10.1007/978-0-387-09766-4\2085](https://doi.org/10.1007/978-0-387-09766-4\2085).
- [66] B. Yan, R. A. Regueiro, Superlinear speedup phenomenon in parallel 3d discrete element method (dem) simulations of complex-shaped particles, *Parallel Computing* 75 (2018) 61 – 87. doi:<https://doi.org/10.1016/j.parco.2018.03.007>.
URL <http://www.sciencedirect.com/science/article/pii/S0167819118300826>
- [67] H. Kottowski, C. Savatteri, Fundamentals of liquid metal boiling thermohydraulics, *Nuclear Engineering and Design* 82 (2) (1984) 281 – 304. doi:[https://doi.org/10.1016/0029-5493\(84\)90216-4](https://doi.org/10.1016/0029-5493(84)90216-4).
- [68] M. Bottoni, B. Dorr, C. Homann, F. Huber, K. Mattes, F. W. Pepperl, D. Struwe, Experimental and numerical investigations of sodium boiling experiments in pin bundle geometry, *Nuclear Technology* 89 (1) (1990) 56–82. arXiv:<https://doi.org/10.13182/NT90-A34359>, doi: [10.13182/NT90-A34359](https://doi.org/10.13182/NT90-A34359).
URL <https://doi.org/10.13182/NT90-A34359>
- [69] R. W. Lockhart, R. C. Martinelli, Proposed correlation of data for isothermal two-phase two-component flow in pipes, *Chemical Engineering Progress* 45 (1) (1949) 39–48.
- [70] H. Nguyen, A finite element method for one-dimensional sodium boiling simulation, in: *Proceedings of the 3rd Int. Topical Meeting on React Thermal-hydraulics*, 1985, p. 17.
- [71] A. Kaiser, F. Huber, M. Bottoni, B. Dorr, Contribution to sodium boiling heat transfer, pressure drop, and void distribution in multi-pin geometry, in: *Proceedings of the 13th Liquid Metal Boiling Working Group (LMBWG)*, 1988, pp. 71–98.
- [72] J. C. Chen, S. Kalish, An experimental investigation of two-phase pressure drop for potassium with and without net vaporization, in: *Proceedings of the 4th Int. Heat Transfer Conference*, Vol. 6, 1988.
- [73] J. Fink, L. Leibowitz, A consistent assessment of the thermophysical properties of sodium, *High Temperature and Materials Science* 35 (1) (2 1996).
URL <https://www.osti.gov/biblio/251097>

- [74] M. Autruffe, G. Wilson, B. Stewart, M. Kazimi, A proposed momentum exchange coefficient for two-phase modeling of sodium boiling., in: Proceedings of the International Meeting on Fast Reactor Safety Technology, 1979, pp. 2515–2521.
- [75] K. Rehme, Pressure drop correlations for fuel element spacers, Nuclear Technology 17 (1973) 12 – 23.
- [76] A. Schor, M. Kazimi, N. Todreas, Advances in two-phase flow modeling for lmfb applications, Nuclear Engineering and Design 82 (2) (1984) 127 – 155. doi:[https://doi.org/10.1016/0029-5493\(84\)90208-5](https://doi.org/10.1016/0029-5493(84)90208-5).
URL <http://www.sciencedirect.com/science/article/pii/0029549384902085>
- [77] M. Mohammed Shah, A survey of experimental heat transfer data for nucleate pool boiling of liquid metals and a new correlation, International Journal of Heat and Fluid Flow 13 (4) (1992) 370 – 379. doi:[https://doi.org/10.1016/0142-727X\(92\)90007-V](https://doi.org/10.1016/0142-727X(92)90007-V).
URL <http://www.sciencedirect.com/science/article/pii/0142727X9290007V>
- [78] H. No, M. Kazimi, Investigation of the physical foundations of two-fluid representation of sodium boiling in the liquid-metal fast breeder reactor., Nuclear Science and Engineering 97 (4) (1987) 327–343, cited By 11. doi:[10.13182/NSE87-A23516](https://doi.org/10.13182/NSE87-A23516).
- [79] N. Yue, D. Zhang, J. Chen, P. Song, X. Wang, S. Wang, S. Qiu, G. Su, Y. Zhang, The development and validation of the inter-wrapper flow model in sodium-cooled fast reactors, Progress in Nuclear Energy 108 (2018) 54 – 65. doi:<https://doi.org/10.1016/j.pnucene.2018.05.007>.
- [80] J.-B. Droin, N. Marie, A. Bachrata, F. Bertrand, E. Merle, J.-M. Seiler, Physical tool for unprotected loss of flow transient simulations in a sodium fast reactor, Annals of Nuclear Energy 106 (2017) 195 – 210. doi:<https://doi.org/10.1016/j.anucene.2017.03.035>.
- [81] R. Hu, Y. Yu, A computationally efficient method for full-core conjugate heat transfer modeling of sodium fast reactors, Nuclear Engineering and Design 308 (2016) 182 – 193. doi:<https://doi.org/10.1016/j.nucengdes.2016.08.018>.
- [82] K. Mikityuk, A. Chenu, K. Sun, European patent: a wrapper tube for a fuel subassembly of a nuclear reactor core and method for protecting fuel against overheating in case of coolant boiling, patent number: WO/2013/098079.
URL <https://patents.google.com/patent/EP2610875A1/en>

- [83] J. Boure, A. Bergles, L. Tong, Review of two-phase flow instability, Nuclear Engineering and Design 25 (2) (1973) 165 – 192. doi:[https://doi.org/10.1016/0029-5493\(73\)90043-5](https://doi.org/10.1016/0029-5493(73)90043-5).
- [84] H. W. Bertini, Descriptions of selected accidents that have occurred at nuclear reactor facilities, Tech. Rep. ORNL/NSIC-176, Oak Ridge National Laboratory (1980). doi:<https://doi.org/10.2172/5591584>.
- [85] K. Mikityuk, E. Girardi, J. Krepel, E. Bubelis, E. Fridman, A. Rineiski, N. Girault, F. Payot, L. Buligins, G. Gerbeth, N. Chauvin, C. Latge, J.-C. Garnier, ESFR-SMART: new Horizon-2020 project on SFR safety, in: Proceedings of International Conference on Fast Reactors and Related Fuel Cycles: Next Generation Nuclear Systems for Sustainable Development, no. IAEA-CN245-450, 2017.
- [86] A. Gunter, W. Shaw, A general correlation of friction factors of various types of surfaces in cross flow, ASME Transactions 67 (1945) 643–660.
- [87] W. Al-Tameemi, P. Ricco, Pressure-loss coefficient of 90 deg sharp-angled miter elbows, Journal of Fluids Engineering 140 (1975) 1 – 12.
- [88] A. Sidi, A complete convergence and stability theory for a generalized richardson extrapolation process, SIAM Journal on Numerical Analysis 34 (5) (1997) 1761–1778. doi:<https://doi.org/10.1137/S0036142994278589>.
- [89] IAEA, Coordinated research project on FFTF LOFWOS tests.
URL <https://www.iaea.org/newscenter/news/new-crp-benchmark-analysis-of-fftf-loss-of-flow-without-scram-test-i32011>
- [90] D. M. Lucoff, Passive safety testing at the fast flux test facility, Tech. Rep. WHC-SA-0046 (September 1987). doi:[10.13182/NT89-A34333](https://doi.org/10.13182/NT89-A34333).
URL <https://www.osti.gov/biblio/6193000>
- [91] C. Fiorina, N. Kerkar, K. Mikityuk, P. Rubiolo, A. Pautz, Development and verification of the neutron diffusion solver for the GeN-Foam multi-physics platform, Annals of Nuclear Energy 96 (2016) 212 – 222. doi:<https://doi.org/10.1016/j.anucene.2016.05.023>.
- [92] C. Fiorina, S. Radman, M.-Z. Koc, A. Pautz, Detailed modelling of the expansion reactivity feedback in fast reactors using openfoam, in: M&C Conference 2019, American Nuclear Society, 2019.
- [93] C. Fiorina, M. Hursin, A. Pautz, Extension of the GeN-Foam neutronic solver to SP3 analysis and application to the CROCUS experimental reactor, Annals of Nuclear Energy 101 (2017) 419 – 428. doi:<https://doi.org/10.1016/j.anucene.2016.11.042>.

- [94] C. Fiorina, S. Radman, A. Pautz, Preliminary application of the gen-foam multiphysics tool to the analysis of the fftf sodium fast reactor: Coupling thermal-hydraulics and core deformations, in: Proceedings of the 18th International Topical Meeting on Nuclear Reactor Thermal Hydraulics (NURETH-18), 2019, pp. 5680–5690.
- [95] D. Wootan, R. Omberg, T. Sofu, C. Grandy, Passive safety testing at the fast flux test facility relevant to new lmr designs, in: IAEA Proceedings of the International Conference on Fast Reactor and Related Fuel Cycles (FR17), 2017.
- [96] C. Cabell, A summary description of the fast flux test facility, Tech. Rep. HEDL-400 (PNNL) (December 1980).
URL <https://community.pnnl.gov/sites/FFTFLibrary/FFT%20Documents/HEDL-400.PDF>
- [97] J. Leppänen, M. Pusa, T. Viitanen, V. Valtavirta, T. Kaltiaisenaho, The serpent monte carlo code: Status, development and applications in 2013, Annals of Nuclear Energy 82 (2015) 142–150, joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo 2013, SNA + MC 2013. Pluri- and Trans-disciplinarity, Towards New Modeling and Numerical Simulation Paradigms. doi:<https://doi.org/10.1016/j.anucene.2014.08.024>.
URL <https://www.sciencedirect.com/science/article/pii/S0306454914004095>
- [98] E. Nikitin, E. Fridman, K. Mikityuk, S. Radman, C. Fiorina, Neutronic modelling of the fftf control rod worth measurements with diffusion codes, in: Proceedings of PHYSOR 2020, 2020.
- [99] J. Lavarenne, E. Bubelis, U. Davies, S. Gicquel, J. Krepel, M. Lainet, B. Lindley, K. Mikityuk, C. Murphy, B. Perrin, W. Pfraun, A. Ponomarev, A. Schubert, E. Shwageraus, P. Van Uffelen, A 2-d correlation to evaluate fuel-cladding gap thermal conductance in mixed oxide fuel elements for sodium-cooled fast reactors, in: Proceedings of Top Fuel 2019, 2019.
- [100] S. Churchill, Friction factor equation spans all fluid flow regimes, Chemical Engineering 84 (1977) 91 – 92.
- [101] G. Sidebotham, Heat Transfer Modeling, Springer, 2015.
URL https://doi.org/10.1007/978-3-319-14514-3_9
- [102] E. Fridman, E. Shwageraus, Modeling of sfr cores with serpent–dyn3d codes sequence, Annals of Nuclear Energy 53 (2013) 354–363. doi:<https://doi.org/10.1016/j.anucene.2012.08.006>.
URL <https://www.sciencedirect.com/science/article/pii/S0306454912002824>

- [103] B. Kiedrowski, Prompt Behaviour of Generalized-eigenvalue Point kinetics Models, in: Proceedings of PHYSOR 2014, 2014.
URL https://mcnp.lanl.gov/pdf_files/la-ur-14-20340.pdf
- [104] G. I. Bell, S. Glasstone, Nuclear Reactor Theory, no. TID-25606, United States Atomic Energy Commission, 1970.
URL <https://www.osti.gov/biblio/4074688>
- [105] Q. L. Baird, R. A. Harris, FFTF operational results: startup to 100 MWd/kg, Nuclear Safety 5 (26).
URL <https://www.osti.gov/biblio/6250714>