Neutronic, thermohydraulic and thermomechanical coupling for the modeling of criticality accidents in nuclear systems

Juan Antonio Blanco

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITE GRENOBLE ALPES

Spécialité : **MEP : Mécanique des fluides Energétique, Procédés**

Arrêté ministériel : 25 mai 2016

Présentée par

Juan Antonio BLANCO

Thèse dirigée par **Pablo RUBIOLO, Professeur, Université Grenoble Alpes**,
et codirigée par **Eric DUMONTEIL, Ingénieur-Chercheur, Université Paris-Saclay**

préparée au sein du **Laboratoire de Physique Subatomique et Cosmologie**
dans **l'École Doctorale I-MEP2 - Ingénierie - Matériaux, Mécanique, Environnement, Energétique, Procédés, Production**

Couplage neutronique, thermohydraulique et thermomécanique pour la modélisation des accidents de criticité dans des systèmes nucléaires

Neutronic, thermohydraulic and thermomechanical coupling for the modeling of criticality accidents in nuclear systems

Thèse soutenue publiquement le **11 décembre 2020**,
devant le jury composé de :

Monsieur PABLO RUBIOLO
Professeur, Grenoble INP, Directeur de thèse
Monsieur ERIC DUMONTEIL
Ingénieur-Chercheur HDR, CEA, Co-directeur de thèse
Madame SANDRA DULLA
Professeur, Politécnico di Torino, Rapportrice
Monsieur ANDREA ZOIA
Ingénieur-Chercheur HDR, CEA, Rapporteur
Monsieur SYLVAIN DAVID
Directeur de Recherche, CNRS, Président
Monsieur FAUSTO FRANCESCHINI
Docteur-Ingénieur, Westinghouse Mangiarotti, Examinateur
Monsieur BENJAMIN DECHENAUX
Docteur-Ingénieur, IRSN, Invité
Monsieur CARLO FIORINA
Docteur-Ingénieur, EPFL, Invité

Abstract

This thesis was developed in the framework of the multi-scale and multi-physics modeling for the simulation of criticality accidents, carried jointly between the CNRS and the IRSN. A multi-physics and multi-scale approach aims to produce a numerical model taking into account all the relevant physical phenomena existing in nuclear systems as well as their coupling. This approach makes possible to improve the predictive capacities of the single physics models and to numerically study the behavior of a nuclear system under conditions that would be difficult to achieve or reproduce by experiments. The multi-scale / multi-physics approach is, therefore, particularly useful for the study of nuclear reactor criticality accidents, or more generally, for all nuclear systems where a tight coupling exists between neutronics, mechanics (of solids and fluids) and heat transfers.

The objectives of the thesis were, firstly, to develop a new numerical scheme for the coupling between the neutronic code Serpent 2 (Monte Carlo code) and the Computational Fluid Dynamics (CFD) code OpenFOAM. Secondly, to develop the physical models that allow greater flexibility for criticality accidents studies in terms of type of transients, systems and phenomena considered. Among the various physical models developed during the work, it can be mentioned the transient neutronic models based on a quasi-static Monte Carlo approach and on the deterministic SP_1 and SP_3 methods. A porous medium model was also developed during the work to allow performing studies on nuclear systems containing a solid nuclear fuel cooled by a fluid. The numerical implementation of the multi-physics coupling was performed in C/C++ in the OpenFOAM code. This code is very well suited to numerically solve continuous mechanics problems using a finite volume method. It also provides very large library of CFD algorithms (RANS, LES and DNS). The thesis work specially focused on the study of the strategy to be followed to implement the quasi-static method numerically with a Monte Carlo type code in the same platform through internal coupling.

The performances of the coupling and the developed models were studied for different scenarios and nuclear systems: the transient Godiva experiments, an international benchmark for multi-physics codes for Molten Salt Reactor and the case of a hypothetical criticality accident in a Boiling Water Reactor (BWR) Spent Fuel Pool. These diverse scenarios and systems were selected because they are characterized by presenting a multitude of highly coupled physical phenomena which required a very careful modeling. One can mention: the Doppler and fuel density effects, the thermal expansion and thermomechanical stresses, the presence of laminar or turbulent flows in the coolant or liquid fuel, the delayed neutrons precursors convection, and the energy and mass transfers and the phase change in porous media. The comparison between the multi-physics tool and the available data shows a very good agreement and confirms that the selected approach is pertinent for the study of criticality accidents and allows obtaining very good precision and flexibility while maintaining satisfactory computational costs.

Résumé

Cette thèse s'inscrit dans le cadre de travaux portant sur la mise au point de modèles multi-échelle et multi-physiques pour la simulation des accidents de criticité, menés conjointement par le CNRS et l'IRSN. L'approche multi-physique et multi-échelle a comme objectif de produire un modèle numérique prenant en compte tous les phénomènes physiques importants dans les systèmes nucléaires ainsi que leur couplage. Cette approche permet d'améliorer les capacités prédictives des modèles et d'étudier de manière numérique le comportement des composants d'un système nucléaire dans des conditions difficilement réalisables/reproductibles par des expériences. Elle est donc particulièrement utile pour l'étude des accidents de criticité, ou plus généralement, pour tous les systèmes nucléaires où de très forts couplages existent entre la neutronique, la mécanique (des solides et des fluides) et la thermique.

Les objectifs de la thèse étaient, d'une part, de développer un nouveau schéma numérique de couplage entre le code neutronique Serpent 2 (code du type Monte Carlo) et le code OpenFOAM (code de mécanique des fluides numérique ou CFD) et, d'autre part, de développer des modèles physiques permettant une plus grande flexibilité dans les études des accidents de criticité en termes de type de transitoires, de systèmes et de variété de phénomènes. Parmi les modèles développés dans notre outil multi-physique on peut citer des modèles neutroniques transitoires du type quasi-statique Monte Carlo et déterministes SP_1 et SP_3. Un modèle thermo-hydraulique du type milieu poreux a été aussi mis en place pour les études des systèmes comportant du combustible solide entouré par un caloporteur. L'implémentation numérique du couplage multi-physique a été faite en C/C++ sur la plateforme OpenFOAM qui permet la résolution numérique de modèles de mécanique des fluides (RANS, LES et DNS), et plus généralement, de la mécanique des milieux continus, en utilisant la méthode des volumes finis. Le travail de thèse a porté par ailleurs sur l'étude de la stratégie à suivre pour implémenter numériquement la méthode quasi-statique avec un code de type Monte Carlo dans la même plateforme et à travers d'un couplage interne.

Les performances du couplage et des modèles que nous avons développés dans la thèse ont été testées dans différents scénarios et systèmes nucléaires : expériences transitoires avec l'expérience Godiva, benchmark international entre codes multi-physiques d'un réacteur à sels fondus et scénarios hypothétiques d'accidents de criticité dans des piscines de combustibles des Réacteur à Eau Bouillante (REB). Ces divers scénarios et systèmes ont été choisis pour leurs nombreux phénomènes physiques couplés nécessitant une modélisation très précise : effet Doppler, expansion thermique et contraintes thermomécaniques dans le combustible, présence des écoulements laminaires ou turbulents dans le caloporteur ou le combustible liquide, convection des précurseurs de neutrons retardés et phénomènes de transfert de masse, transfert d'énergie et de changement de phase dans des milieux poreux. La confrontation des prévisions de l'outil multi-physique et des résultats disponibles s'avère très satisfaisante et montre que l'approche adoptée est très pertinente et adaptée aux particularités des études de criticité avec un niveau de précision et une flexibilité adéquate tout en présentant des coûts computationnels raisonnables.

Contents

List of Figures

List of Tables

Acronyms

ADS Accelerator-Driven Systems. 32, 182

BC Boundary Condition. 62, 110, 117, 126, 146, 149, 151, 173

BWR Boiling Water Reactor. iii, xiii, xv, 14–16, 163, 164, 173, 185

CASL Consortium for the Advanced Simulation of Light Water Reactors. 51

CDF Cumulative Distribution Function. 26, 27, 94

CFD Computational Fluid Dynamics. iii, v, 3, 75, 76, 87, 121, 177, 185

CHT Conjugated Heat Transfer. 3, 119, 193, 194, 212

CNRS Centre National de la Recherche Scientifique. iii, v, xiii, xix, 11, 12, 19, 121, 146, 185

CV Control Volume. 40, 63–65, 67, 112

CVs Control Volumes. 64

DNS Direct Numerical Simulation. iii, v, 3, 75, 76

DOE Department of Energy. 51

ENDF Evaluated Nuclear Data File. xix, 123, 128, 138, 139

FA Fuel Assembly. 77, 118, 162, 163, 170, 172, 207

FAs Fuel Assemblies. xv, xvi, xx, 1, 14, 16, 17, 32, 77, 116, 122, 162–166, 168–173, 185, 209

FEM Finite Element Method. 111

FVM Finite Volume Method. 87, 97, 111, 146

I/O Input/Output. 112

IAEA International Atomic Energy Agency. 178

ICNC International Conference on Nuclear Criticality Safety. 162

IMPES IMplicit Pressure Explicit Saturation. 117, 208

IQM Improved Quasi-Static Method. xiii, 47, 97, 100–103, 127, 128

IRSN Institut de Radioprotection et de Sûreté Nucléaire. iii, v

JEFF Joint Evaluated Fission and Fusion File. xix, 12, 138, 139, 146

LANL Los Alamos National Laboratory. xiii, xiv, xix, 8, 10, 122, 125, 126, 129, 130, 137, 139–142

LES Large Eddy Simulation. iii, v, 3, 76, 116

LHS Left Hand Side. 38, 78, 80

LOCA Loss of Coolant Accident. xvii, 16, 168, 185, 209, 211

LOFA Loss of Flow Accident. xvii, 168, 209, 211

LPSC Laboratoire de Physique Subatomique et de Cosmologie. 97, 189, 192, 193, 243

LWR Light Water Reactor. 184

MCNP Monte Carlo N-Particle. 86

MPI Message Passing Interface. 87

MSCB Molten Salt Cavity Benchmark. xix, 8, 14, 97, 121, 145, 146, 158, 179, 180, 185, 193, 201

MSFR Molten Salt Fast Reactor. 8, 11, 19, 161

MSR Molten Salt Reactor. iii, 39, 179, 185, 193

MV Material Volume. 66, 70

MVs Material Volumes. 64

NEA Nuclear Energy Agency. xv, 123, 125, 163, 164

NEP Nuclear Electric Propulsion. 193

NIST National Institute of Standards and Technology. 146

NPP Nuclear Power Plant. 14, 176

NRC Nuclear Regulatory Commission. 166

ODE Ordinary Differential Equation. 40, 50

OECD Organisation for Economic Co-operation and Development. xv, 163, 164

OF-PCQM OpenFOAM PCQM. xiv, xv, xix, xx, 98, 103, 126, 134–136, 138, 141, 143–146, 160, 161, 201

OMP Open Multi-Processing. 87

ONCORE Open-source Nuclear Code for Reactor Analysis. 178

OpenFOAM Open-source Field Operation and Manipulation. iii, v, xiv, xv, xvii, xx, 3–5, 19, 20, 33, 50, 56, 63, 76, 83, 86–88, 90, 91, 93, 94, 97, 101–103, 106, 110–113, 116–119, 121, 128, 136, 144, 146–157, 159–161, 165, 176–181, 184–186, 189, 195, 202, 203, 207, 211–213

OQM Original Quasi-Static Method. 47, 50

PCQM Predictor-Corrector Quasi-Static Method. xiii, 97–99, 101–103

PDE Partial Differential Equation. 87, 146

PDF Probability Density Function. 26, 27, 94

PIRT Phenomena Identification and Ranking Table. 7, 177

PISO Pressure-Implicit with Splitting of Operators. xiii, 113–116

PK Point Kinetics. 126, 135

PWR Pressurized Water Reactor. 9, 109, 116

QSM Quasi-Static Method. xiv, xv, xix, xx, 3, 4, 19, 22, 25, 30, 34, 35, 39, 41, 46, 47, 50, 83, 85, 86, 90, 94, 96, 97, 101, 105, 106, 122, 123, 126–129, 134, 139–144, 159–161, 178, 179, 181, 182, 186, 187

RANS Reynolds-averaged Navier–Stokes equations. iii, v, 3, 76, 83, 116, 185, 189, 190

REB Réacteur à Eau Bouillante. v

REV Representative Elementary Volume. 77, 78, 80–82, 207, 213

RHS Right Hand Side. 38, 43, 56, 66, 79, 80

RIA Reactivity-Initiated Accident. 9

RTT Reynolds Transport Theorem. 64–66, 70, 112

SAMOFAR Safety Assessment of the Molten Salt Fast Reactor. 8, 14, 146, 179, 185

SFP Spent Fuel Pool. iii, xiii, xvi, xvii, 1, 3, 8, 14–17, 20, 77, 78, 82, 83, 112, 116–119, 121, 122, 162, 163, 165–168, 170, 172, 173, 175, 176, 179, 185–187, 207–209, 211, 212

SIMPLE Semi-Implicit Method for Pressure Linked Equations. xiii, 113–117

SP_1 Simplified P_1. iii, v, xv, xx, 3, 4, 19, 22, 51, 55, 83, 85, 107, 123, 126, 137, 146, 148–150, 152, 156, 160, 161, 184

SP_3 Simplified P_3. iii, v, xiv, xv, xix, xx, 3, 4, 19, 55, 56, 83, 85, 107, 110, 123, 126, 127, 137–146, 148–157, 159–161, 184, 201, 235

SP$_\mathbf{N}$ Simplified P$_\mathrm{N}$. 3, 5, 19, 22, 28, 50–54, 56, 83, 85, 86, 90, 100, 106, 107, 109, 122, 129, 144, 179, 184, 187, 235

TMS Target Motion Sampling. 95, 105

VOF Volume of Fluid. 186, 187, 211, 212

Chapter 1

Introduction

The French regulation concerning the safety rules related to the use and storage of nuclear materials relies on separate decrees for operating nuclear reactors (where self-sustained fission chain reactions or **criticality** are part of the nominal functioning of the reactor) and for safety-criticality installations which encompass all other uses of nuclear materials (as for instance fuel storage, fuel reprocessing facilities, Spent Fuel Pool (SFP), Fuel Assemblies (FAs), transport of radioactive materials, loading of nuclear reactors, or nuclear medicine just to name a few). In these latter installations, criticality conditions are prohibited so that operators have to demonstrate that all industrial operations involving fissile materials stay at a certain "distance" of criticality. Any flaw in the design of such an industrial facility, or in the criticality-safety demonstration regulating its use, can therefore lend itself to persisting neutron chains, eventually culminating in a critical or super-critical state of the system. Such uncontrolled events, called **criticality accidents**, can result in the release of high radiation doses and cause possible dispersion of radioisotopes, while possibly exposing the workers and the public. Less than fifty known criticality accidents with sanitary consequences occurred worldwide in the past, most of them being hardly documented, the others being reviewed in [1]. The last known accident occurred in the Mayak Russian military complex in 2017 causing the release of ruthenium 106 detected at different places in Europe. Another example is the accident that took place in the Fuel Conversion Test Building at the JCO company site in Tokai-mura, Japan. An 18% enriched uranyl nitrate solution and an unfavorable geometry vessel used during the processing of this solution, along with protocol violations, led to the loss of the lives of two operators and to the exposure of the surrounding population to measurable airborne fission products activity. This criticality accident lasted more than twenty hours, so that questions were raised afterwards concerning eventual mitigation strategies that could be used during the happening of such events.

Understanding the progression of criticality accidents is however a prerequisite before elaborating any such strategy, and this motivated different criticality programs and experiments in the past. But bridging the gap between such experiments and their associated modeling still constitutes a challenge: the diversity of the physics phenomena implied in these experiments ranges from the ability to simulate thermal expansion and compressibility effects (such as shock waves) to the modeling of chemical processes such as radiolysis, and to the fine calculation of precursors advection. Also, the quick variations found during criticality accidents calls for relying on state-of-the-art methods in each sub-field: for instance, the simulation of neutron transport under such conditions requires to precisely calculate fission source distributions (to predict precise radiation dose), or to be able to homogenize and condensate macroscopic cross sections on tightly coupled system with fast variations of spatial and spectral distributions, among others. To study these phenomena in different nuclear systems, the criticality experiments usually employ homogeneous or heterogeneous nuclear fuel in very diverse configurations, geometries and size scales. In addition, the time scales characterizing the physics of such experiments involve transients ranging from some microseconds to many hours or even days.

In the past, different ad-hoc numerical modeling tools have been developed to support criticality experiments but their use was very often restricted to a particular application: for example, specific codes were developed to model the criticality of wetted powders while other codes would support only the modeling of specific geometries containing liquid solutions.

This PhD thesis therefore targets the development of a more general, multi-scale and multi-physics numerical tool, that can be used to study in detail the large variety of physics phenomena occurring during criticality accidents and allowing the *best estimate* of target quantities and the propagation of uncertainties. A particular emphasis will be devoted to neutronics models, as well as their coupling with thermal-hydraulics and thermal-mechanics phenomena. However, in regards of such a versatile tool to model complex geometries and phenomena in an environment rich in numerical methods suited to deal with convergence problems, one has to consider the cost of relying on enlarged input data (materials properties, geometry descriptions, initial and boundary conditions, ...) and on the intensive computational resources that are required by such codes.

Therefore, to properly handle a wide range of physics phenomena, geometries and time scales, it was decided that the code could be developed with the aid of the C++ Library Open-source Field Operation and Manipulation (OpenFOAM) [2][3] for continuum mechanics and, specially, Computational Fluid Dynamics (CFD). This library provides a syntax-friendly finite volumes framework with state-of-the-art CFD algorithms (RANS, LES and DNS) and other continuum mechanics models (e.g. Conjugated Heat Transfer), and a toolkit of numerical schemes making this library well suited for coupling with other codes.

However, a certain degree of adaptation of current tools and implementation of new models under the same solver was needed, especially concerning neutronics. Indeed, while the use of brute-force approaches relying on the Monte Carlo solving of the critical transport equations -making very few hypotheses- is not always practically possible (due to the calculation costs), using deterministic solvers (such as diffusion) might also constitute a challenge as these methods are often fine-tuned for specific problems (for example, dealing with discontinuity factors [4] at assembly interfaces when simulating a Spent Fuel Pool). Both complementary approaches were therefore developed to perform transient calculations: a stochastic method based on the Monte Carlo code Serpent 2 [5][6] and a deterministic method based on a Simplified P_N (SP_N) method. The Monte Carlo method is indeed well suited for the modeling of criticality accidents since it can readily handle a large variety of geometries and materials, it uses continuum-energy cross sections data, and has the possibility to perform uncertainties propagation (for stand-alone neutronics calculations at least). Nevertheless, an important drawback of these methods when coupled to multi-physics codes is the prohibitive computational cost needed by a Monte Carlo simulation, especially by the so-called **Dynamic Monte Carlo** [7] that simulates each time steps and takes into account feedbacks reactions between time steps. To overcome this issue, we explore the use of a Quasi-Static Method (QSM) to perform transient neutronics simulations with a Monte Carlo code. As an alternative neutronic method, a Simplified P_1 (SP_1) and a Simplified P_3 (SP_3) models were also developed and implemented in the multi-physics tool. This latter deterministic approach will serve as a reference calculation to compare computational time and results accuracy with the ones obtain with the stochastic methodologies.

The objectives of this thesis work include:

- The development of a multi-scales, multi-physics numerical tool internally coupling Serpent 2 and OpenFOAM including a triple coupling between neutronics, thermal-mechanics and thermal-hydraulics under the same solver.
- The analysis of the performances of a Quasi-Static Method when modeling fast -but still realistic- transients. Other works have extensively studied in the past the limits of the Quasi-Static Method but using hypothetical cases with a very large reactivity insertion (and thus not realistic) intended for the method to fail as we will see in section 5.1.
- The investigation of the performances of the tool to study tightly coupled small nuclear systems where the diffusion hypothesis fails.
- The investigation of the performance of the tool to model liquid fuel.
- The evaluation of the respective computational cost when using stochastic or deterministic approaches, in an effort to check the feasibility of the first when performing a normally expensive transient calculation.
- The precise modeling of thermal expansion in systems where leakage is very dependent on the changes in the cross sections due to density modifications and, thus, it provides a strong correlation to thermal volume expansion.
- The implementation of a porous medium model to study criticality accidents in a heterogeneous medium.
- The study of the convergence of neutron sources (in magnitude and spatially). In nuclear safety studies, having a good estimation of the neutron source term is critical for the accuracy of the analysis. To this extent, it can be noticed that the use of a Monte Carlo code does not lead to over-prediction of the source term like in more conservative methods.

In summary, this work presents the mathematical modeling and numerical implementation of the diverse phenomena taking place in nuclear systems subject to potentially accidental conditions. The numerical implementation of the thermal-mechanics and thermal-hydraulics models was performed using OpenFOAM [2][3]. The neutronics simulations were carried-out using a Quasi-Static Method applied to the Serpent 2 Monte Carlo code [5][6] and, a SP_1 and SP_3 stand-alone methods. This PhD manuscript is organized as follows: the nuclear systems used to test the coupled tool are described in Chapter 2, then, a detailed explanation of the neutronics, thermal-mechanical and thermal-hydraulics models is given in Chapter 3. The methodology covering the Serpent-OpenFOAM coupling, the implementation of the $\mathrm{SP_N}$ equations, the density and Doppler broadening feedback effects, and algorithms used in the simulation of the transients are discussed in Chapter 4. Then, results on the deterministic and stochastic respective performances during the transient simulations are presented in Chapter 5 along with a discussion on the main features of the methods and their computational costs. In addition, a porous medium model applied to a heterogeneous medium is studied. Finally, the conclusions on the overall performances of the methodologies adopted are discussed in Chapter 6.

Chapter 2

Criticality Accidents

As stated in the previous chapter, the precise study of criticality accidents in nuclear systems has a major impact on the design of facilities and protocols used to deal with nuclear materials and constitute an operational experience that can be used during criticality safety assessments. The physics phenomena involved in criticality accidents are however diverse. A Phenomena Identification and Ranking Table (PIRT) has been done to identify as much of the phenomena present during a criticality accident, the following list is not exhaustive but covers a significant part of them:

- Transient neutron transport including super-prompt criticality
- Doppler broadening
- Radiolysis and bubbles formation within a fluid
- Precursors transport in solid and liquid medium
- Buoyancy
- Conjugated heat transfer
- Radiative heat transfer
- Incompressible laminar and turbulent flow
- Compressible flow and shock waves
- Multi-phase flow
- Stress and strain analysis
- Elasticity and plasticity
- Thermal expansion
- Geometrical and material heterogeneity
- Different time and space scales

- Anisotropic medium
- Phase change and mass transfer (e.g boiling and condensation, solidification)
- Fluid free surface (sloshing)
- Multi-phase multi-component flow in porous medium

Hence, the purpose of this work was to develop a tool versatile enough to tackle as many as possible of these complex phenomena while being capable of robust prediction. For this reason, three systems, or cases of application, were selected in an attempt to include a selection of the aforementioned phenomena.

The first consists in a **solid homogeneous medium** known as the Godiva Experiment, characterized by its fast neutronic transient ($\sim \mu$s) and thermal expansion as it will be introduced in Section 2.1. The second is a **liquid homogeneous medium** based on a simplification of the Molten Salt Fast Reactor (MSFR) developed as a benchmark by the participants of the European Project SAMOFAR. It introduces convective phenomena including buoyancy and precursors transport in a slower transient (∼min). This Molten Salt Cavity Benchmark (MSCB) is described in Section 2.2. Finally, the recriticality possibility of a Spent Fuel Pool assembly is explored with a slow transient (∼hours) in a **solid-liquid heterogeneous medium** with a hypothetical model based on the Fukushima nuclear accident. An overview of the progression of such an accident is discussed in Section 2.3.

2.1 LANL Godiva Experiment

The Godiva device was built by Los Alamos National Laboratory (LANL) and has been in operation since August, 1951. The device is made from Oralloy: uranium enriched to a nominal U-235 isotopic abundance of 93.5% . It has three main sections that could be assembled into a nearly spherical shape as can be seen in Figure 2.1. In practice, a shim of Oralloy had to be added for achieving criticality giving to the system a slightly elongated sphere shape. In addition, to fine tune the experience reactivity, the device had: i) two different mass adjustment Oralloy plugs on the surface, ii) two Oralloy rods could be inserted in the static central section for continuously variable reactivity control and iii) a third Oralloy rod that could be propelled by an explosive charge for a rapid reactivity insertion. A more detailed description of the device is given in reference [8].

Other identified interesting experiments including more phenomena were the SILENE

Reactor [9] (includes radiolysis and shock waves) and the CABRI Facility (Reactivity-Initiated Accident which must be considered in the safety analysis of PWRs). For lack of time, these were not modeled but should be studied in the future.

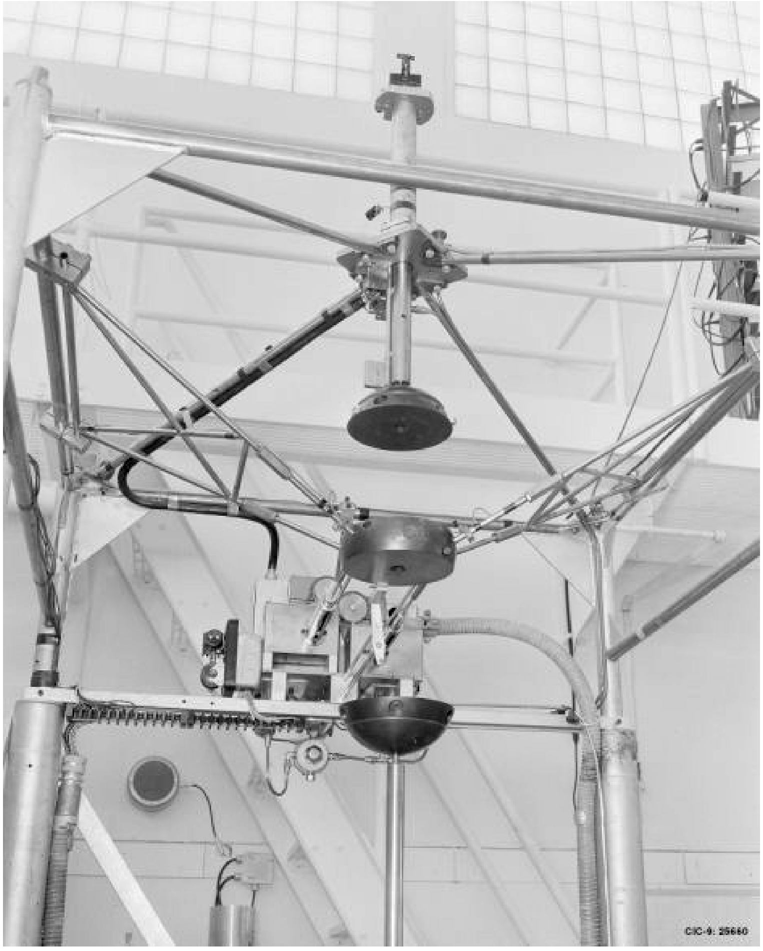


FIGURE 2.1 – Godiva device in scrammed position [10].

As discussed, we are particularly interested in studying the performance of the neutronics models during very fast transient. Such transients were performed in Godiva as a result of a reactivity insertion over prompt critical and were known as the **Godiva prompt-burst program** [10]. These experiments were intended to study various phenomena: firstly, the system behaviour in the verge of a delayed to super-prompt criticality (reactivity ranging from 95 to 110 cents); secondly, the self-termination of the burst due to the assembly thermal expansion effect on leakage; thirdly, whether the mass associated with prompt-criticality prediction was accurately predicted from the estimations at low reactivities with the in-hour equation; and finally, if there were short-period delayed neutron emitters.

The Godiva prompt-burst program experiments were initiated at zero power (low neutron counts) and without external source by a quick reactivity insertion, which caused a sudden exponential increase in the fission rate and, therefore, in the power. Afterwards, the assembly thermal expansion caused a decrease in reactivity causing the power to drop. This bell-shaped transient is known as **burst**. Finally, a lower power level was sustained by the delayed neutrons and it was terminated upon scramming of the assembly (disassembly of the device). The experimental results were compared in [10] with an analytical model based on point kinetics taking into account the thermal expansion feedback as observed in Figure 2.2.

In the same work, experimental results of several power excursions with initial zero power (low neutron counts) without source with different reactivity insertions were reported.

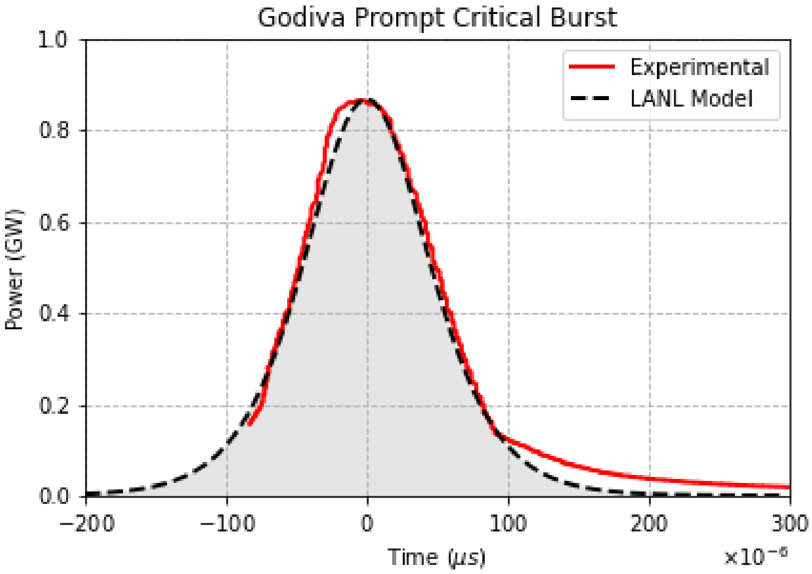


FIGURE 2.2 – LANL Data Experimental and Point Kinetics model.

In this work we focus the analysis in the transient having a 29.5 μs reactor period because it was fast enough not to have a significant precursors concentration but still slow enough to avoid pressure waves. The maximum reactivities were chosen in the experiments to limit this issue, as explained in [11].

The LANL point kinetics model reported in [8] follows equation 2.1 for the fission rate $\dot{F}$:

$$\dot{F} = f 2 \rho_0^2 a_p \frac{\exp \alpha \left(t - t_m\right)}{\left[1 + \exp \alpha \left(t - t_m\right)\right]^2} \tag{2.1}$$

where ρ_0 is the initial super prompt reactivity in dollars, $f = 4.8 \times 10^{16}$ fissions and $a_p =$

$1.03 \times 10^6\ s^{-1}$ are parameters derived from experimental data [8], $\alpha = 1/T_r$ with T_r being the reactor period or e-folding time.

The fission rate peak value and the total fissions can be calculated from equations 2.2 and 2.3 respectively [8].

$$\dot{F}_{Max} = \frac{1}{2}\rho_0^2 a_p \tag{2.2}$$

$$\dot{F}_{Tot} = 2\rho_0 \tag{2.3}$$

From this description, it can be seen that a model covering neutronics - thermomechanical coupling with volume thermal expansion was needed. It was developed and implemented as explained in Chapters 3 and 4 respectively. The results for the Godiva Experiment 29.5 μs prompt-burst are later presented in Section 5.1.

2.2 CNRS Molten Salt Square Cavity Benchmark

The study case used to evaluate the performance of the multi-physics coupling in a liquid homogeneous fissile medium was the multi-physics benchmark proposed by the CNRS to identify error sources between different multi-physics tools [12].

As displayed in Figure 2.3, the system is a critical 2 m x 2 m 2-D square cavity filled with a FLiBe molten fuel salt, under natural and forced convection conditions. The FLiBe molten fuel salt has a composition of ^6Li, ^7Li, ^9Be, ^{19}F and ^{235}U with atomic fractions of 2.11488%, 26.0836%, 14.0992%, 56.3969% and 1.30545% respectively. Both, the salt composition and geometry dimensions were selected to match the MSFR spectral indexes and dominance ratio, to provide a representative simplified case study. All cavity walls are fixed, except the upper one which have a constant velocity U_x to induce forced convection conditions inside the cavity. Non-slip velocity and adiabatic conditions are imposed at all the cavity walls. Gravity and thermal fluid expansion are allowed to create a strong coupling between thermohydraulics and neutronics phenomena.

To avoid uncertainties arising from turbulence modeling, the benchmark uses laminar, incompressible formulation of the Navier-Stokes equations along with Boussinesq approximation to determine the impact of natural circulation, fuel salt thermal expansion feedback and delayed neutron precursors drift in a simple way (modeled in Chapter 3). The choice of the viscosity, Prandtl (Pr) and Schmidt (Sc) numbers were carefully optimized in order to highlight important physical phenomena and reduce, at the same time, the impact of numerical issues whose evaluation was beyond the scope of the present benchmark. The heat is removed from the cavity via a uniform artificial volumetric heat removal coefficient:

$$q^{'''}(\vec{r}) = h_{sink}\left(T(\vec{r}) - T_{ref}\right) \tag{2.4}$$

where $h_{sink} = 10^6\ WK^{-1}m^{-2}$ and $T_{ref} = 900\ K$.

The neutron data library JEFF-3.1 was used with an 8 groups effective delayed neutron precursors. Doppler effect correction is not made to avoid introducing uncertainties related to the Doppler treatment. However, the macroscopic cross sections of the molten fuel salt are modified accordingly to the salt temperature and the thermal expansion coefficient, i.e.:

$$\Sigma(T) = \frac{\rho(T)}{\rho(T_{ref})}\Sigma(T_{ref}) \tag{2.5}$$

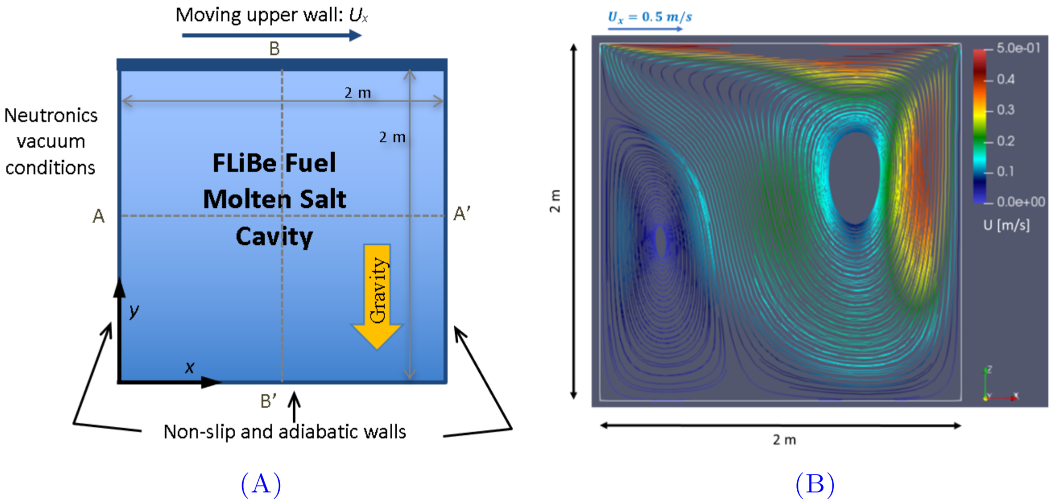


FIGURE 2.3 – CNRS multi-physics benchmark: (A) Lid-driven Cavity Geometry (B) Example of velocity field under forced and natural convection conditions [13].

The benchmark is divided in three phases: **Phase 0** (single physics and one-way coupling verification), **Phase 1** (steady-state benchmark), and **Phase 2** (transient benchmark).

Three steps of the **Phase 0** were used in this work:

- **Step 0.1** aims to accurate simulate the steady-state incompressible flow of the molten salt setting the top-lid velocity of the cavity to 0.5 m s^{-1}. The stand-alone hydraulics is tested.

- **Step 0.2** focus on the neutronics criticality eigenvalue problem in the presence of a static fuel, 900 K uniform temperature distribution and normalized power to 1 GW.
- **Step 0.3** tests the energy equation stand-alone resolution in the presence of the velocity field calculated in Step 0.1 and the heat source distribution from Step 0.2. The heat removal follows equation 2.4.

For **Phase 1**, four of the steps present in the benchmark relabelled as in reference [14] were simulated:

- **Step 1.1** adds the effect of the circulating fuel to the neutronics criticality eigenvalue problem. The fixed velocity field from Step 0.1 is used to compute the precursors transport and quantify its effect on the neutronics. Neutron flux is normalized to obtain an integrated power of 1 GW. Temperature field is not calculated and set to 900 K uniformly.
- **Step 1.2** the two-way coupling between the neutronics and the temperature distribution in the presence of the velocity field from Step 0.1 is observed. The effect of the non-uniform fuel temperature field on the neutronics is quantified. Power is normalized to 1 GW.
- **Step 1.3** uses the full neutronics-thermal-hydraulics coupling. Velocity at the top lid is set to 0 m s^{-1}. Buoyancy effects are presented at this step. Power is normalized to 1 GW.
- **Step 1.4** once again uses the full coupling but with a fixed velocity at the top lid. In the benchmark velocities from 0 to 0.5 m s^{-1} with a step of 0.1 m s^{-1} and power between 0.2 and 1 GW with a step of 0.2 GW are analysed. In this work, we treat only the case of 0.5 m s^{-1} and 1 GW.

In **Phase 2**, to compare the multi-physics model predictions in transient conditions, the benchmark adopted a different approach to what is usually used in transient analysis of nuclear systems. Rather than studying the maximum (or minimum) values taken by key parameters (for example the temperature), the benchmark studies the frequency domain response (i.e. the transfer function of the tool) of the multi-physics tool for a small perturbation: a ±10% amplitude sinusoidal fluctuation of the heat exchange coefficient h_{sink} was introduced during a transient simulation. By performing several transients with different period length of the perturbation T_p, the frequency response of the tool can be obtained. Since the system response is almost linear for such small perturbation, the predictions of different multi-physics tools can then be compared using a Bode plot containing the phase and the amplitude amplification at different frequencies. Note that changes on the heat coefficient will modify the natural circulation conditions and thus the delayed neutron precursors distribution. This method allows then to test the coupling of different equations. Tools having similar Bode diagram will have similar transient response for small perturbations. Phase 2 of the benchmark is particularly well adapted to study the numerical performance of the different transient neutronics models used in our multi-physics tool.

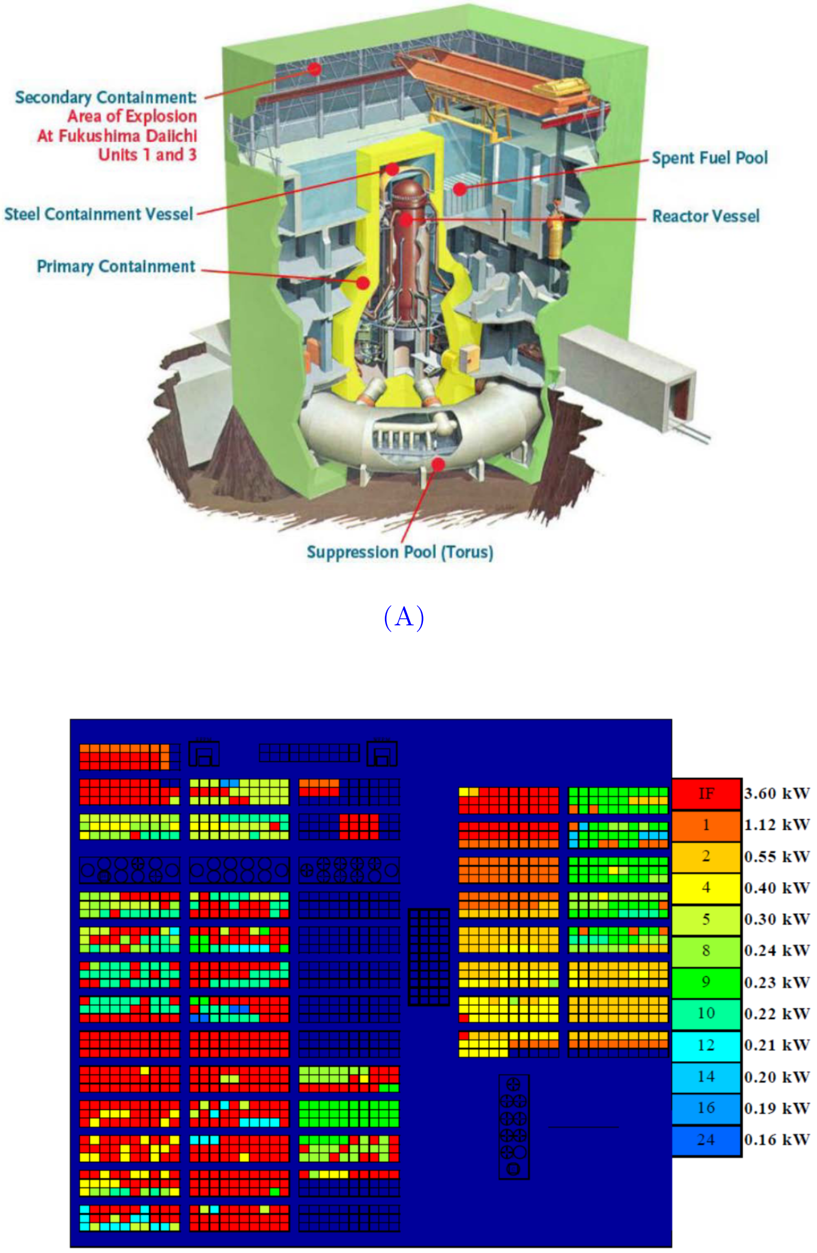
More details on these three phases of the benchmark can be found in [12] while more detailed results on the multi-physics benchmark obtained from various multi-physics codes used by the participants of the SAMOFAR project (and the one treated here) can be found in [14].

The results of the neutronics-thermohydraulics simulations for the Molten Salt Cavity Benchmark are presented in Section 5.2.

2.3 Spent Fuel Pools

In this study, the Spent Fuel Pool (SFP) configuration of the Fukushima Daiichi Nuclear Power Plant (NPP) Station has been used. The Fukushima Daiichi NPP has six General Electric BWRs. Units 1 (BWR3) and Units 2-5 (BWR4) have a Mark I type of containment while the containment of Unit 6 (BWR6) is Mark II type (a schematic of a BWR [15] is displayed in Figure 2.4(A)). The Spent Fuel Pool of each Unit is located in the secondary containment. In March 11, 2011 during the accident caused by the Tohuku earthquake and the subsequent tsunami, Units 1 to 3 were in operation, Unit 4 was defueled for maintenance and Units 5 and 6 were on cold shutdown also for maintenance [15]. To maximize the reactivity, this study is based on the Spent Fuel Pool composition of the Unit 4 since it contained, at the same time, fresh and low irradiated Fuel Assemblies.

The SFP4 consisted in 53 fuel storage racks as shown in Figure 2.4(B). The assemblies were arranged in racks of high-density design in a 3x10 configuration. Cooling water in the pool has natural boron which ensures adequate criticality margin thanks to the very high thermal neutron absorption of boron-10 (3800 barns). The figure also shows on the right side the approximate residual power distribution in the pool at the moment of the accidents: in red the most recently offloaded fuels are shown [16]. As can be seen, fresh



(A)

(B)

Figure 2.4 – (A) Generic cross section of a BWR/4 with a Mark-I containment [15] (B) Thermal loading in the Unit 4 Spent Fuel Pool [16].

Fuel Assemblies (racks in blue) [17] have no heat production. In the case of interruption of the cooling system in the Spent Fuel Pool, the water level will decrease as result of the coolant evaporation. In some hypothetical scenarios, this could lead to a condition where the reactivity margins are sufficiently degraded and the Spent Fuel Pool recriticality scenario cannot be fully excluded. In Figure 2.5, a progression of this type of accidents is detailed (from [15]). It can be observed that, due to the loss of cooling from the coolant, the temperature rises and a natural convection loop is established inside the Spent Fuel Pool. Once the coolant reaches the saturation point, water boiling starts. After a few days (if evaporation is the only mechanism for the loss of coolant) the assemblies will be partially uncovered and natural convection will stop since the resulting cross flow will be blocked by the walls of the BWR Fuel Assemblies and the walls of the solid steel racks. The fuel temperature will continue to rise as a result of the reduction of the heat removal mechanisms. Moreover, the high temperature existing in the uncovered part of the Fuel Assemblies will lead to zirconium oxidation and hydrogen production. This will significantly degrade the fuel cladding, and, if no action is taken, it will cause the release of the fission products. As the water surface descends under the level of the racks, natural convection in the air-steam mixture will be established aiding the cooling. However, if not acted upon, severe fuel failure will be reached including fuel melting. From a neutronic point of view different phenomena play an important role in the accident. The large gaps of water between assemblies under normal conditions assures enough cooling and neutronic decoupling between racks to allow the system to be at a subcritical over-moderated state. At normal conditions the water density in the Spent Fuel Pool is much higher than in reactor conditions, giving better moderation but a much higher absorption. The Spent Fuel Pool is therefore overmoderated which explains why the Loss of Coolant Accident (LOCA) could lead to a potential significant reduction of the criticality margins. And, as said before, a scenario involving a criticality accident cannot be fully excluded.

An adequate numerical model to study this accident requires therefore coupled neutronics and thermohydraulic calculations to study all potential scenarios. The thermohydraulic model is necessary to correctly predict the water levels in the Spent Fuel Pool and in the racks and should include phenomena such as convective and conductive heat transfer, change from water to air natural convection, boiling and two-phase flow and hydrogen production and combustion from zirconium–steam reaction. The neutronics model should

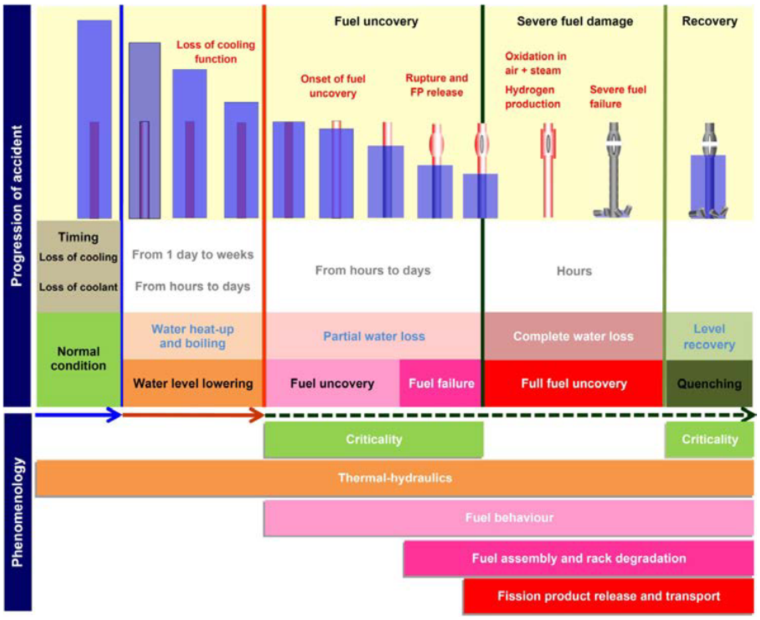


Figure 2.5 – Progression of a Spent Fuel Pool loss of cooling/coolant accident [15].

be able to accurately predict the Spent Fuel Pool reactivity and the phenomena associated to a hypothetical criticality accident. Due to its geometrical complexity, the Spent Fuel Pool containing the racks and the Fuel Assemblies is simplified as a porous medium as we will see in Section 3.3.7. The level of detail required for this highly heterogeneous type of systems makes it computationally unfeasible to model it otherwise. Results for an hypothetical accidental scenario are presented in section 5.3 to illustrate some of the capabilities of the tool to simulate different racks and Fuel Assemblies configurations.

Chapter 3

Model

As discussed in Chapter 2, a wide range of phenomena exists during a criticality accident (thermal expansion, compressibility effects such as shock waves, radiolysis, precursors advection, ...). In addition, nuclear systems concerned by criticality accidents or criticality experiments are very diverse and may employ homogeneous or heterogeneous nuclear fuel nuclear in different configurations, geometries and sizes (from centimeters to meters). Moreover, the time scales of these experiments can also be very different: transients ranging from some microseconds to many hours or even days. A comprehensive numerical tool to study these accidents has therefore to implement a multi-scale and multi-physics modeling approach.

The development of such a tool was initiated at CNRS for the study of Molten Salt Fast Reactor [18] and, as discussed in Chapter 1, the objective of this research work was to enhance this multi-physics tool by developing and numerically implementing the models required for the criticality accidents. These models include new transient neutronics models, a solid thermomechanical model and different thermohydraulical models. A numerical coupling between OpenFOAM and Serpent 2 codes was also needed for the neutronic modeling and thus implemented during this work (See Chapter 4). This chapter presents the theoretical basis used in developing these models and it is divided in three parts. The first part presents the development and numerical implementation of the transient neutronic models which include a Monte Carlo Quasi-Static Method and a deterministic Simplified $\mathrm{P_N}$ ($\mathrm{SP_1}$ and $\mathrm{SP_3}$). The second part presents the solid mechanics model used for studying the Godiva experiment, in particular the adaptation of a linear thermoelasticity model already existing in OpenFOAM. The last part of the chapter is devoted to the development of a porous medium model with two phase flow, phase change and two temperatures. This porous medium model was developed in this thesis work by enhancing an already existing model in OpenFOAM (Horgue et al. [19]) and it was used to study potential Spent Fuel Pool criticality accidents. This last part also discusses the turbulent models existing in OpenFOAM and used for studies of nuclear systems containing a liquid fissile such as a liquid fuel.

3.1 Neutronics

Adopting the notation of reference [20], the **neutron transport equation** 3.1 can be derived as a balance of the different mechanisms by which neutrons can be gained (neutron sources like fission or delayed neutrons, streaming, scattering from any energy or direction to the ones of interest ($E' \rightarrow E$, $\vec{\Omega}' \rightarrow \vec{\Omega}$)) or lost (leakage, absorption/scattering collision removal from energy and direction of interest) from an arbitrary volume within the considered system [21]. The neutron angular flux ψ depends on 7 variables: three spatial coordinates ($\vec{r}$), two directions ($\vec{\Omega}$), energy (E) and the time (t) and can be calculated from equation 3.1:

$$\begin{aligned}\frac{1}{v(E)}\frac{\partial \psi}{\partial t}(\vec{r},\vec{\Omega},E,t) \; = \; & \left[-\mathcal{L} \; - \; \mathcal{T} \; + \; \mathcal{S} \; + \; \frac{\chi_p(E)}{4\pi}(1-\beta)\mathcal{F} \right] \psi(\vec{r},\vec{\Omega},E,t) \\ & + \sum_{d=1}^{G_d} \frac{\chi_d(E)}{4\pi}\lambda_d C_d(\vec{r},t) \end{aligned} \tag{3.1}$$

where Σ_r is the macroscopic cross section for reaction r with r as the total (t), fission (f) or scattering (s) reaction, ν is the average number of neutrons emitted per fission, χ_p and χ_d are the fraction of the prompt p or delayed d fission neutrons spectrum emitted at a certain energy (E). $\mathcal{L}$, $\mathcal{T}$, $\mathcal{S}$ and $\mathcal{F}$ are the Transport, Removal, Scattering and Fission operators respectively defined as follows considering an isotropic fission source:

$$\mathcal{L}\psi(\vec{r},\vec{\Omega},E,t) \; = \; \vec{\Omega}\cdot\vec{\nabla}\psi(\vec{r},\vec{\Omega},E,t) \tag{3.2}$$

$$\mathcal{T}\psi(\vec{r},\vec{\Omega},E,t) \; = \; \Sigma_t(\vec{r},E,t)\psi(\vec{r},\vec{\Omega},E,t) \tag{3.3}$$

$$\mathcal{S}\psi(\vec{r},\vec{\Omega},E,t) \; = \; \iint \Sigma_s(\vec{r},\vec{\Omega'}\cdot\vec{\Omega},E^{'} \rightarrow E,t)\psi(\vec{r},\vec{\Omega'},E^{'},t)\,d\vec{\Omega'}\,dE^{'} \tag{3.4}$$

$$\mathcal{F}\psi(\vec{r},\vec{\Omega},E,t) \; = \; \iint \nu\Sigma_f(\vec{r},E^{'},t)\psi(\vec{r},\vec{\Omega'},E^{'},t)\,d\vec{\Omega'}\,dE^{'} \tag{3.5}$$

Among the neutrons sources, the delayed neutrons source produced as a consequence of the decay of certain fission products (known as delayed neutron precursors) was included in the previous analysis.

The delayed neutron precursors concentration (denoted as C_d) is obtained from a balance between their production proportional to the fission source and their radioactive decay. In the case of a solid fuel, the delayed neutrons source of equation 3.1 can be calculated using equation 3.6 below. Normally, the fission products generated by the nuclear fission are very diverse and usually not worthy to simulate in detail unless we are dealing with burn-up calculations. Indeed in most applications, such as neutronic transient calculations, we only want to know the rate at which delayed-neutrons are produced and in which energy range or group. For this reason, they are represented by a few-families (either 6 or 8 in general) having an effective decay constant λ_d and a relative delayed-neutrons fraction produced per fission β_d for family d [21]:

$$\frac{\partial C_d}{\partial t}(\vec{r},t) \;=\; \beta_d \iint \mathcal{F}\psi(\vec{r},\vec{\Omega},E,t)dE\; d\vec{\Omega} \;-\; \lambda_d C_d(\vec{r},t) \qquad d = 1\ to\ G_d \tag{3.6}$$

Note that for the case of a system containing a liquid fissile (e.g. a reactor using a liquid fuel or a fissile solution) two new terms will have to be added in equation 3.6: i) a convection term representing the precursors transport by the fluid and ii) a diffusion term that depends on the medium diffusion coefficient, which can be usually neglected. Therefore, for systems having a liquid fissile the precursors equation 3.7 has to be used [12]:

$$\begin{aligned}\frac{\partial C_d}{\partial t}(\vec{r},t) \;+\; \vec{u}\cdot\nabla C_d(\vec{r},t) \;&=\; \nabla\cdot[D_d \nabla C_d(\vec{r},t)]\\ &+\beta_d \iint \mathcal{F}\psi(\vec{r},\vec{\Omega},E,t)dE\; d\vec{\Omega} \;-\; \lambda_d C_d(\vec{r},t) \qquad d = 1\ to\ G_d\end{aligned} \tag{3.7}$$

There are several difficulties for solving the system of equations 3.1 to 3.7:

- There are seven independent variables as enumerated before,
- The nuclear reaction cross sections spatial dependence can become rapidly complicated in reactors with complex geometries, which is normally the case,
- The nuclear reaction cross sections dependence on energy has complex structure with resolved and unresolved resonances, threshold effects, etc.

An analytical solution for this integro-differential equation is only possible for very simple theoretical cases usually not representative of our applications. Moreover, the numerical resolution of these equations is in general not practical unless some approximations are introduced such as it is done in most of the deterministic methods. Among the various deterministic methods existing in neutronics we have selected for this work the approach known as Simplified $\mathrm{P_N}$ that is based on the Legendre Polynomial expansion of the angular dependence of the flux. This method was chosen for its simplicity and also because, in the formulation chosen here, the order 1 ($\mathrm{SP_1}$) is equivalent to the diffusion approximation. The Simplified $\mathrm{P_N}$ implemented in this work is presented in section 3.1.3. This method will serve, in this work, as a deterministic reference calculation and it will provide a fast and useful method in many applications involving large systems (in terms of neutrons mean free path).

The most precise method, with only a few approximations, to solve the neutron transport equation, is the statistical approach (also called stochastic method) based on Monte Carlo calculations which will be discussed in the next section 3.1.1. Nonetheless, the computational effort required by this method for transient calculations is still extremely demanding, in terms of CPUs and memory consumption, and for transients calculations unfeasible in most cases [22][23]. For this reason, we implemented the Monte Carlo method by using an approximation based on the Quasi-Static Method which is presented in section 3.1.2.

3.1.1 Monte Carlo

3.1.1.1 Introduction

The idea of the Monte Carlo method goes back in history but it became feasible in the 1940's with the increase of computing power. As stated by Lux and Koblinger [24], the general method to solve for particle transport was developed by Fermi, Ulam and Von Neumann, and the first comprehensive review was published by Kahn [25]. Most books, such as Bell and Glasstone [26], present this method as one of the tools to solve the transport equation. If a more in-depth mathematically analysis is desired by the reader Spanier and Gelbard [27] and, more recently, Lux and Koblinger [24] presents such a work for the bases of Monte Carlo applied to neutron transport.

It is not an easy task to define what Monte Carlo is. A very concise definition is given by Lux and Koblinger [24]: “In all applications of the Monte Carlo Method a stochastic model is constructed in which the expected value of a certain random variable is equivalent to the value of a physical quantity to be determined. This expected value is then estimated by the average of several independent samples representing the random variable introduced above. For the construction of the series of independent samples, random numbers following the distributions of the variable to be estimated are used". This is however a quite a complex definition but embodies all of the essential parts of a Monte Carlo simulation.

For the problem of neutron transport, the general approach normally consists in generating a set of neutrons (with given positions and velocities) and following them. Since the cross sections represent the probability of a given interaction per unit length, the length between collisions and type of interaction can be sampled from generated random numbers. In addition, energy and direction after collision can be sampled from the corresponding probability laws. This sampling of the path with the cross sections and the probabilities laws constitutes the **stochastic model**. Along the process, some statistical data is stored related to the quantities that we wish to calculate. This statistical data is therefore collected by formulating a **random variable**, which can be seen as a function that assigns values to each independent sample. The value it takes at each sample is called a **score**. This random variable is chosen in a way that the average of all scores will converge to the physical quantity to be determined. This convergence is true if the expected value of the random variable gives us back the physical quantity.

For the sake of clarity, an example by Spanier and Gelbard [27] is shown here. Let us consider an experiment with two possible outcomes: the event Ω forms a space consisting of two points ω and its complement ω' with the probability P defined as

$$\begin{aligned} P(\omega) &= p, \qquad 0 \le p \le 1 \\ P(\omega') &= q \;=\; 1-p \end{aligned} \tag{3.8}$$

Then, a random variable ξ can be defined as $\xi(\omega) = 1$ and $\xi(\omega') = 0$. The probability distribution of ξ is defined as

$$F(t) \;=\; P\{\xi < t\} \;=\; p\chi(t-1) + q\chi(t) \tag{3.9}$$

where

$$\chi(t) = \begin{cases} 0 & if\ t < 0 \\ \\ 1 & if\ t \geq 0 \end{cases} \tag{3.10}$$

When performing N independent trials of the experiment, the space to examine is Ω^N consisting of all N-tuples $(\omega_1, \ldots, \omega_N)$ of points of Ω. Its probability is defined as

$$P_N(\omega_1, \ldots, \omega_N) \ = \ \prod_{i=1}^{N} P(\omega_i) \tag{3.11}$$

N random variables ξ_i on Ω^N are defined as $\xi_i(\omega_1, \ldots, \omega_N) \ = \ \xi(\omega_i)$ with $1 \leq i \leq N$. Then, the sum

$$\xi^{(N)} \ = \ \sum_{i=1}^{N} \xi_i \tag{3.12}$$

is also a random variable representing the total count the event ω has happened. Its probability distribution is the binomial distribution

$$F_N \ = \ P\{\xi^{(N)} \leq t\} = \sum_{k \leq t} \begin{pmatrix} N \\ \\ k \end{pmatrix} p^k q^{N-k} \tag{3.13}$$

In addition the random variable

$$\xi^{(\bar{N})} \ = \ \xi^{(N)}/N \tag{3.14}$$

represents the frequency of occurrences of the event ω.

Then, calculating the expected value and standard deviation of the latter random variable we obtain

$$E\left[\xi^{(\bar{N})}\right] \ = \ p \qquad \sigma\left[\xi^{(\bar{N})}\right] \ = \ \sqrt{\frac{pq}{N}} \tag{3.15}$$

As we can see, the random variable representing the frequency of occurrence is the probability associated with the event ω. This way, information from the parent distribution can be obtained. This is a trivial case where we know from the beginning the probability. However, in real problems the stochastic model becomes more complex and the desired physical quantity is not explicitly available.

In general, when the stochastic search for the quantity of interest is faithful to the physical process it is known as an **analog** process. In particular, for particle transport, such as the neutron transport, this type of Monte Carlo simulation is known as an **analog random walk process**, where a random walk denotes the path of the neutron consisting of a succession of random steps. Nevertheless, it can be solved in an non-analog way if it is needed because of poor statistics or high variances for example. In addition, even though Monte Carlo has a direct application to random processes, it can also be used for solving problems that have no direct probabilistic interpretation.

The main advantage of the Monte Carlo method is that it can solve problems with few approximations in complex geometries. It eliminates the problem of additional calculations like the multi-group cross sections generation[1] for deterministic models for example. However, the computational effort required is normally high and, for many problems, deterministic methods are accurate enough[2]. Later on, an effort to circumvent this issue will be done by applying the Quasi-Static Method to Monte Carlo.

3.1.1.2 Tracking Methods

The core of the Monte Carlo neutron transports is the particle tracking routine, i.e., following the path that a neutron undertakes through the different regions of the geometry. The distance between two events (collision, leakage, absorption) is called **track**. The set of all tracks of a single neutron from its birth to its removal (either by absorption or leakage) is called a **neutron history**.

For this purpose, the sampling of the mean free path between two points is vital to the simulation. As explained in [24][29], this sampling procedure goes as follows:

First, a neutron travelling through an infinite homogeneous media is considered. The total cross section Σ_t describes the probability of interaction per unit length. The probability of interaction along a distance dx can be define as

$$dP = \Sigma_t \, dx \tag{3.16}$$

Now, $P_0(x)$ denotes the probability that a neutron has reached a position x without

[1] For deterministic calculations, information about the nuclear data should be adapted to the problem, i.e. homogenized and condensed from continuous data to the right amount of groups and regions to correctly describe the problem. In contrast, Monte Carlo can directly use continuous-energy nuclear data

[2] For example, for nuclear reactors in power plants, the mean free path of the neutron is several times smaller than the “neutronic” size of the same. Thus, the diffusion approximation, for example, yields accurate enough results given that the boundaries are not as sensitive as in smaller geometries. This boundaries represent a very small part of the geometry in terms of mean free path. In addition, some adapted boundary conditions simplifies the problem, for example, extrapolated length [28]

any interaction. The reduction of $P_0(x)$ as it moves a distance dx through x is equal to

$$dP_0 \;=\; -P_0(x)dP \;=\; -P_0(x)\Sigma_t dx \tag{3.17}$$

Then, the non-interaction probability is

$$P_0(x) \;=\; e^{-x\Sigma_t} \tag{3.18}$$

The probability of a neutron moving first a distance x without interactions and then interacting in the interval dx is

$$P_0(x)dP \;=\; P_0(x)\Sigma_t dx \;=\; \Sigma_t e^{-x\Sigma_t} dx \tag{3.19}$$

The *Probability Density Function (PDF)* and the *Cumulative Distribution Function (CDF)* of the free path are respectively

$$f(x) \;=\; \Sigma_t e^{-x\Sigma_t} \tag{3.20}$$

$$F(x) \;=\; 1 - e^{-x\Sigma_t} \tag{3.21}$$

Finally, the neutron's distance to the next collision can be sampled by means of the *inversion method* [24] :

$$x \;=\; -\frac{1}{\Sigma_t} log(1-\zeta) \;=\; -\frac{1}{\Sigma_t} log(\zeta) \tag{3.22}$$

where ζ is a generated random number, i.e., in this case, a number between 0 and 1 obtained from an equally probable distribution. In addition, the distributions of $(1-\zeta)$ and ζ are similarly distributed (both generates equally probable random numbers between 0 and 1) and, therefore, they can be interchanged.

The tracking can be a combination of the *ray tracing based surface-tracking* and the *rejection sampling based delta-tracking*. This methods are usually referred as **surface-tracking** and **delta-tracking**.

Surface-tracking

For calculating the neutron path length using equation 3.22 it is necessary for the material to be homogeneous. When the cross section is spatially dependent the integrals to obtain the PDF and CDF from equation 3.19 become more complex. Nevertheless, for most problems, the geometry consists of several regions with homogeneous materials, also called **cells**. When tracking the neutron, it can cross different regions which have different properties. For this reason, every time the neutron crosses a region interface the collision probability changes and the sampled path length is not statistically valid for the next region. To solve this problem, the neutron is stopped at the interface surface and the length is either adjusted or a new sample is made as can be seen in [24][29]. This strategy is a **ray tracing** algorithm that follows the neutron from one surface to another [30].

Delta-tracking

The delta-tracking method was first introduced by Woodcock in the 60's. This method is explained by Leppänen [31] and in more mathematical detail by Lux and Koblinger [24]. The main idea (and advantage) behind this method is that it samples the next collision point without handling surfaces interface crossing.

In this method, a **virtual collision** is created. It is an interaction characterized by the cross section $\Sigma_0(r, E)$ that preserves the incident energy and flight direction. An arbitrary number of virtual collisions can occur during the neutron lifetime without affecting the simulation converged results[3] by preserving the average of the scores. In each material a virtual collision cross section is added in order to have the same modified (called majorant) total cross section all over the geometry. This way, the total interaction probability is the same everywhere and thus the need to adjust the free path length at each surface disappears as well as the calculation of the distances to the surfaces. If the neutron undergoes a collision which is a virtual collision, the neutron history rests unchanged.

The main disadvantage of this method is that the crosses between surfaces are not registered and, therefore, the track length estimator (explained in the next section) is not available and the possibly less efficient collision estimator has to be used. Another problem arises when there are localised heavy absorbers like control rods or burnable poisons [31]. In this case, the virtual collision cross section necessary to homogenize the modified total cross section all over the system can be simply too big. As a consequence, the frequency of virtual collisions in regions with lower adsorptions increases and a lot of computational time is wasted in sampling several times the same collision site. In addition, the search for

[3] However, the statistics will be altered, in particular, affecting the variance and, therefore, the simulation time and performance will not be the same.

the majorant cross section adds to the computational effort required in this method.

Because of these limitations, and depending on the system, a combination of the surface-tracking and the delta-tracking is normally performed [31].

3.1.1.3 Estimators

In the deterministic approach the equations describing the neutrons transport using some approximation (diffusion, $\mathrm{SP_N}$, among others) are solved for the neutron flux either analytically (very rarely if a simple problem is given) or numerically via a discretization of the equations in a mesh. Different to this, Monte Carlo provides an estimate of a random variable whose expected value precisely converges (without any bias) to the integral of the flux (or more complex observables, such as currents or reaction rates, or deposited energy) in a phase-space volume; the flux (or the other associated observables) is the formal solution of the Boltzmann equation. In this way, it shares certain analogy to making experimental physical measurements.

A very important capability, specially for this work, is the evaluation of reaction rates by flux integrals of the type:

$$R = \iiint f(r,E)\phi(r,E)\,dt\,d^3r\,dE \tag{3.23}$$

where $f(r,E)$ represents any physical parameter, i.e., if $f(r,E) = \Sigma_a$ then R is the total absorption rate (or counts) in a certain volume, energy range and over a given period of time. As in an experiment, the longer it is being measured (more neutron histories) then the larger will be the obtained integral, i.e., the accumulated scores will continue to increase. To be able to have a physically meaningful quantity it is necessary to normalize the rate obtained from equation 3.23 in a specific method as it is explained in more detail in Leppänen [29].

In order to obtain the value of the integral an **estimator** is needed. In the words of Amit Majumdar [32]: “An estimator is a specific function of the random samples, of a random variable, that statistically represents a true unknown mean. If x is a random variable with an associated distribution and an unknown mean, then the function $X(x_1, x_2, x_3, \ldots, x_n)$ is an estimator of the unknown mean. The set $\{x_1, x_2, x_3, \ldots, x_n\}$ consists of n independent random samples selected from the probability density distribution of x. A good estimator should be unbiased, consistent, and efficient. An estimator is *unbiased* if its expected value equals the true mean, μ, i.e.,

$$E(X) \;=\; \mu$$

for all X. *Consistency* applies to a single sample where the sample size, n, becomes large. An estimator is *consistent* if it approaches, in a statistical sense, the true mean as n gets large. An efficient estimator is the one, among a group of unbiased estimators, that produces the minimum variance for a given sample size n. ".

In other words, the scores (recorded events) obtained during the simulation can be combined to form statistical estimates of the physical quantities. These counts of recorded events yielding estimates (and interpreted as Monte Carlo results) are colloquially known as **tallies** [33]. These tallies can be either **analog**, i.e., directly related to the simulation process such as the multiplication factor calculated from equation 3.24; or **implicit** (non-analog), this is the case of the integral reaction rates following equation 3.23.

$$k_n \;=\; \frac{number\ of\ source\ neutrons\ in\ cycle\ n+1}{number\ of\ source\ neutrons\ in\ cycle\ n} \tag{3.24}$$

The most common implicit estimators includes the **collision estimator** and the **track length estimator**. A detailed description of these two estimators can be found in Leppännen [34].

Collision Estimator

The analog collision estimator would consist on scoring each collision interaction of a certain type (e.g. counting absorbed neutrons for the absorption rate analog tally). Instead, if the probability of interaction α (e.g. a given nuclear reaction) is scored, the collision estimator can be described as

$$R \;=\; \sum_i \omega_i \, \frac{f(r_i, E_i)}{\Sigma(r_i, E_i)} \tag{3.25}$$

where ω_i, r_i and E_i are the weight[4], position and energy associated with the i-th simulated neutron and Σ is the cross section used to sample the path length. For example, if $f = \Sigma_a$, the probability $\alpha = \Sigma_a/\Sigma$ is the probability of absorption of the particle i and R is an

[4] At is essence, the weight is a tally multiplier. We will not go into detail on the theory of Monte Carlo (see [33]) but, often, a weight is introduced in order to reduce the variance. A departure from a direct physical simulation, thus, having a non-analog simulation.

estimate of the absorption rate after proper normalization. Similarly, other reaction rates can be obtained as it will be shown in Section 4.3.1.2.3.

This estimator is more efficient than the analog one since it is registered at each collision even if the interaction does not happen. It is always scored for physical collisions in both surface and delta-tracking mode, to avoid under-sampling and non-physical results.

Track Length Estimator

The track length estimator is defined as

$$R = \sum_i \omega_i f(r_i, E_i) l_i \tag{3.26}$$

where l_i is the length of the neutron track and r_i, E_i, f and ω_i are defined in the same way as in the collision estimator. Same as the collision estimator, it is used as a reaction rate tally by setting the parameter f according to the physical quantity we want to calculate.

The advantage of this estimator is that the efficiency is better since it records every time the neutron makes a track inside a region where it does not necessarily collide. It works even in void regions.

The disadvantage is that it cannot be used with delta-tracking because surface crossings are not recorded. Given that it can cross different regions with different material the response function f may change its value along the track. Delta tracking does not store this information either.

3.1.1.4 Transport simulation

The simulation of the neutron chain reaction can be done using different methods depending on the system to be modeled. Even though the operations involved in performing the neutron histories calculations in these methods are practically the same, there are big differences in how the neutron population is treated as a whole [35]. The methods can be divided into criticality and external source simulations that are discussed in the next subsections. As we will discuss later, both methods could be used with the Quasi-Static Method approach, however, one of the them will be easier to implement.

3.1.1.4.1 Criticality Source

There are two possible approaches among the criticality source methods: the α-eigenvalue method which will not be used in our applications and thus not treated here (see [36][29]) and the k-eigenvalue method [24]. The k-eigenvalue method (sometimes colloquially called k-code mode) is used for simulating self-sustaining chain reactions equivalent to equation 3.27. In this method, the simulation is divided into cycles, generations or **batches**. The source distribution in each cycle, from which the neutrons are sampled, is given by the fission source distribution from the previous one. The neutron histories are terminated when the neutrons are either captured, undergo fission or escape the system.

During criticality source calculations the neutron histories are started from an initial guess (normally Dirac or uniform distribution). In addition, the initial cycles or **inactive cycles** are discarded for scoring purposes to ensure that the results are not influenced by the initial guess. The population is normalized [24] at the beginning of each cycle forcing the system to a steady-state conditions. The multiplication factor is calculated as the ratio between the new source neutrons generated by the end of each cycle divided by the total statistical weight at the beginning.

Note that in the criticality calculations the converged source distribution corresponds to the fundamental flux mode of equation 3.27:

$$\left[\mathcal{L} \; + \; \mathcal{T} \; - \; \mathcal{S}\;\right] \psi(\vec{r},\vec{\Omega},E) \; = \; \frac{1}{k_{eff}}\frac{1}{4\pi}\left[\chi_p(E)(1-\beta)+\sum_{d=1}^{G_d}\chi_d(E)\beta_d\right] \; \mathcal{F}\;\psi(\vec{r},\vec{\Omega},E) \qquad (3.27)$$

with k_{eff} being the multiplication factor.

In addition, the algorithm jumps in a given cycle from one fission to another rather than continue to track the neutrons born after the fission. Opposite to the External Source mode or Transient Calculations explained in the next section, fission is considered a terminating event, i.e., the neutron fission source is saved along with tallies for different quantities, then the fission source is normalized and a new generation of neutrons is sampled from the saved source distribution. Also, all prompt or delayed neutrons are considered to be emitted instantly after fission following the corresponding spatial, angular and energy probability distributions.

3.1.1.4.2 External Source and Transient calculations

In the external source method, neutrons are started from an user-defined source distribution. Neutron histories are terminated by capture or leakage. In case of fission, the histories are split into new paths (as many as neutrons generated by the fission event).

In this mode, neutrons are not batched into generations (or cycles) and it is analogous to solving the time-dependent transport equation 3.1 in deterministic methods.

When the system is sub-critical, it reaches a steady-state (corresponding to a given source amplification).

This method is typically used for shielding and dose calculations as well as for the study of some advanced nuclear reactors such as Accelerator-Driven Systems (ADS). The physical consistency of this method allows to, given that the histories are not batched into generations, discretize the time into intervals and solve for the time-dependent problem. This is a variant of the external source calculation often called a Dynamic Monte Carlo simulation [7][35], which can provide very accurate information at the expense of a high computational cost (memory and processors).

Note that for the particular case of a super-critical nuclear system the neutron histories will grow exponentially to infinity. Then, the simulation is limited to very short periods of time. To solve for longer simulated time, special population control and normalization methods are necessary as it can be observed in the Monte Carlo transient approach (often called Dynamic Monte Carlo) already existing in Serpent 2 [35]. More detail about population control, time cut-off and binning, precursors treatment, normalization and tallying for transient Monte Carlo can be found in Sjenitzer and Hoogenboom [7][22].

3.1.1.4.3 Important Remarks for Systems Containing a Liquid Fissile

Monte Carlo codes for neutronics are usually developed assuming solid fuels since the vast majority of nuclear reactor used solid Fuel Assemblies. When using a Monte Carlo code for a system containing a liquid fissile material some modifications are needed to include the phenomenology associated to the delayed neutron precursors convection and diffusion. Indeed, as can be seen from equation 3.7, in this case the delayed neutrons will not be born at the same place where the precursor has been produced by the fission. Failing to consider this phenomenon could results in a significant bias on the results of the simulation (such as the effective fraction of delayed neutrons, the reactivity or the flux shape).

A simple way to take into account the precursors transport in a liquid fissile system is to use the External Source method [35] in the Monte Carlo simulation. In this case, the precursors concentration, equations 3.7, are solved by other means, for example using a deterministic solver such as OpenFOAM, and then the resulting delayed neutron precursors concentrations are used as the external neutron source for the Monte Carlo simulation. A fixed-point iteration between the precursor concentration solver and the Monte Carlo routine will be required in this case for obtaining adequate convergence. In addition only prompt neutrons have to be sampled from the fission source by using a neutron yield reduced by the physical total delayed neutron fraction (β).

Moreover, when using the Monte Carlo transient solver (Dynamic Monte Carlo) [37][38], two options can be imagined if the fluid velocity field ($\vec{u}$) is provided:

- If there is precursors tracking, the position of the precursors can be modified by $\vec{r} = \vec{r_0} + \vec{u}\Delta t$ to update its position with the advection of the liquid fissile.
- If there is no precursors tracking the position for the new born delayed neutrons can be obtained by first sampling the decay time from its Poisson distribution and then moving the position the same way as in the previous point.

However, this options may required modification to the source code.

Alternatively to the source method, the k-eigenvalue method can be used to obtain a steady-state solution by introducing some modifications in the original algorithm. In this case, delayed neutron precursors concentration distribution obtained from equation 3.7 is still solved by other means (e.g. OpenFOAM). Then, during the criticality calculations every time a delayed neutron is sampled by the tracking routines, the spatial coordinate of it emission is sampled from the precursors concentration distribution via a rejection method[5]. In other words, the k-eigenvalue method is only modified in the sense that the delayed neutron is emitted with an energy and direction sample by traditional means and the position will follow the externally solved precursor distribution. Iteration is also needed between the solvers for convergence. This will be further explained in Section 4.3.1.1.

3.1.1.4.4 Important Remarks for Transient Calculations

As discussed in the previous sections, a Monte Carlo code can be used to perform transient calculations using the source method and discretizing the time variable. This

[5] In some cases the inversion method is not practical [24][29][27] because the inverse is expensive to calculate or it cannot be solved at all (Maxwell-Boltzmann distribution). The rejection method offers an alternative to calculate a sample from a probability density function $g(x)$.

First, a function $g_1(x) = g(x)/g_{Max}$ is defined, where g_{max} is the maximum value taken by $g(x)$. Then, $0 \leq g_1(x) \leq 1$ in the interval $a \leq x \leq b$. After, two random numbers (ζ_1, ζ_2) are generated and interpreted as a point with plane coordinates $(a+\zeta_1(b-a), \zeta_2)$. This is a point inside the rectangle of base $(b-a)$ and height 1. If the point falls below the curve $g_1(x)$, it is accepted as a sample of $g(x)$; if not, it is rejected and a new sample is sorted. The samples obtained this way are proven to follow the distribution of $g(x)$[27]

approach called Dynamic Monte Carlo does not require new approximations but has a very high computational cost that limits its utilisation in most applications involving a multi-physics code. An alternative approach is proposed in this thesis. This approach is based on the Quasi-Static Method and can be used both with the source or the k-eigenvalue methods. The Quasi-Static Method is presented in the next section.

3.1.2 Quasi-Static Method

As already discussed, the full direct resolution of the neutron transport equation 3.1 is computationally expensive and, thus, not practical in most multi-physics applications. Many different methods (Deterministic or Stochastic) have been proposed to reduce the computational effort of solving the transport equation without losing significant precision. Among these methods and related to the resolution of the time dependency one can cite the Quasi-Static Method, which is a relatively well known method that was proposed firstly in the works published by A. Henry [39][40] and K. Ott [41][42]. One can note that several novel applications of this method for deterministic methods have been proposed until recent years (Dulla et al. [43]).

In this work, we propose to use the Quasi-Static Method along with some modeling approximations in order to perform transient calculations using a Monte Carlo code and a large range of nuclear systems. This is a relatively new application of the Quasi-Static Method that is also being investigated by other authors (Jo et al. [20]). In this section, we briefly present the derivation of the main equations of the Quasi-Static Method and we will focus the discussion on the approximations needed to adapt this method to a Monte Carlo code calculation.

3.1.2.1 Integral Notation

For the sake of simplicity we will use the following notation in the derivation of the Quasi-Static Method equations:

$$\iiint f(\vec{r},\vec{\Omega},E,t)W(\vec{r},\vec{\Omega},E,t)\,dV\,d\vec{\Omega}\,dE = \langle f(\vec{r},\vec{\Omega},E,t) \mid W(\vec{r},\vec{\Omega},E,t)\rangle \tag{3.28}$$

where $f(\vec{r},\vec{\Omega},E,t)$ and $W(\vec{r},\vec{\Omega},E,t)$ could be any arbitrary functions. In general, W will be a weight function as we will see in the next sections. We will use the definition given by equation 3.28 for the rest of this manuscript.

3.1.2.2 Method Description

The very first step on developing the Quasi-Static Method consists in proposing the factorization of the neutron angular flux in the two functions shown in equation 3.29. This is purely an arbitrary definition and embodies no physical assumption. As we can observe, the amplitude function $n(t)$ depends solely on time, whereas the flux shape function $\phi(\vec{r}, \vec{\Omega}, E, t)$ depends on the time and on the space, direction and energy coordinates in the phase space:

$$\psi(\vec{r}, \vec{\Omega}, E, t) = n(t)\phi(\vec{r}, \vec{\Omega}, E, t) \tag{3.29}$$

Replacing the angular flux in equations 3.1 and 3.6 by the equation 3.29 and rearranging the terms the new set of equations 3.30a and 3.30b for the flux shape function and the precursors concentration is obtained. As we can see, these equations are similar to the original set 3.1-3.6 except for the new term $\frac{1}{v(E)}\frac{1}{n(t)}\frac{dn(t)}{dt}$, that can be seen as an effective cross section and the normalization factor $n(t)$ for the delayed neutrons source.

$$\boxed{\begin{aligned}
\frac{1}{v(E)}\ \frac{\partial \phi}{\partial t}(\vec{r}, \vec{\Omega}, E, t) \ &= \Bigg[-\mathcal{L} \ - \ \mathcal{T} \ + \ \mathcal{S} \ - \frac{1}{v(E)}\frac{1}{n(t)}\frac{dn(t)}{dt} \\
&\quad + \frac{\chi_p(E)}{4\pi}(1-\beta)\mathcal{F} \Bigg] \ \phi(\vec{r}, \vec{\Omega}, E, t) \ + \ \frac{1}{n(t)}\sum_{d=1}^{G_d}\frac{\chi_d(E)}{4\pi}\lambda_d C_d(\vec{r}, t) && (3.30a) \\
\frac{\partial C_d}{\partial t}(\vec{r}, t) &= \beta_d \mathcal{F}\phi(\vec{r}, \vec{\Omega}, E, t)n(t) \ - \ \lambda_d C_d(\vec{r}, t) \qquad d = 1 \ to \ G_d && (3.30b)
\end{aligned}}$$

The Quasi-Static Method derives the differential equations 3.31 and 3.32 for the amplitude function $n(t)$ and the precursors concentration by multiplying equations 3.30a and 3.30b by an arbitrary weighting function $W(\vec{r}, \vec{\Omega}, E, t)$ (which depends on all the seven variables) and integrating them in all variables except for the time (according to equation 3.28), as explained in section 3.1.2.1, and dividing by $\langle v^{-1}(E)\phi(\vec{r}, \vec{\Omega}, E, t) \mid W(\vec{r}, \vec{\Omega}, E, t)\rangle$.

$$\frac{dn(t)}{dt} = \left[\frac{\left\langle \left[-\mathcal{L} \; - \; \mathcal{T} \; + \; \mathcal{S} \; + \frac{\chi_p(E)}{4\pi}(1-\beta)\mathcal{F} \right] \phi(\vec{r},\vec{\Omega},E,t) \;\middle|\; W(\vec{r},\vec{\Omega},E,t) \right\rangle}{\left\langle \frac{1}{v(E)}\phi(\vec{r},\vec{\Omega},E,t) \;\middle|\; W(\vec{r},\vec{\Omega},E,t) \right\rangle} - \frac{\left\langle \frac{1}{v(E)} \; \frac{\partial\phi}{\partial t}(\vec{r},\vec{\Omega},E,t) \;\middle|\; W(\vec{r},\vec{\Omega},E,t) \right\rangle}{\left\langle \frac{1}{v(E)}\phi(\vec{r},\vec{\Omega},E,t) \;\middle|\; W(\vec{r},\vec{\Omega},E,t) \right\rangle} \right] n(t) + \frac{\sum_{d=1}^{G_d} \left\langle \frac{\chi_d(E)}{4\pi}\lambda_d C_d(\vec{r},t) \;\middle|\; W(\vec{r},\vec{\Omega},E,t) \right\rangle}{\left\langle \frac{1}{v(E)}\phi(\vec{r},\vec{\Omega},E,t) \;\middle|\; W(\vec{r},\vec{\Omega},E,t) \right\rangle} \tag{3.31}$$

$$\frac{\left\langle \frac{\chi_d(E)}{4\pi} \; \frac{\partial C_d}{\partial t}(\vec{r},t) \;\middle|\; W(\vec{r},\vec{\Omega},E,t) \right\rangle}{\left\langle \frac{1}{v(E)}\phi(\vec{r},\vec{\Omega},E,t) \;\middle|\; W(\vec{r},\vec{\Omega},E,t) \right\rangle} = \frac{\left\langle \frac{\chi_d(E)}{4\pi}\beta_d \; \mathcal{F} \; \phi(\vec{r},\vec{\Omega},E,t) \;\middle|\; W(\vec{r},\vec{\Omega},E,t) \right\rangle}{\left\langle \frac{1}{v(E)}\phi(\vec{r},\vec{\Omega},E,t) \;\middle|\; W(\vec{r},\vec{\Omega},E,t) \right\rangle} - \lambda_d \frac{\left\langle \frac{\chi_d(E)}{4\pi} \; C_d(\vec{r},t) \;\middle|\; W(\vec{r},\vec{\Omega},E,t) \right\rangle}{\left\langle \frac{1}{v(E)}\phi(\vec{r},\vec{\Omega},E,t) \;\middle|\; W(\vec{r},\vec{\Omega},E,t) \right\rangle} \tag{3.32}$$

By rearranging and grouping the terms we obtain the equations 3.33a and 3.33b.

$$\frac{dn(t)}{dt} = \left[\frac{\rho - \beta^{eff}}{\Lambda} - \frac{\left\langle \frac{1}{v(E)} \frac{\partial\phi}{\partial t}(\vec{r},\vec{\Omega},E,t) \;\middle|\; W(\vec{r},\vec{\Omega},E,t) \right\rangle}{\left\langle \frac{1}{v(E)}\phi(\vec{r},\vec{\Omega},E,t) \;\middle|\; W(\vec{r},\vec{\Omega},E,t) \right\rangle} \right] n(t) \; + \; \sum_{d}^{G_d} \lambda_d \bar{c}_d \tag{3.33a}$$

$$\frac{\left\langle \frac{\chi_d(E)}{4\pi} \; \frac{\partial C_d}{\partial t}(\vec{r},t) \;\middle|\; W(\vec{r},\vec{\Omega},E,t) \right\rangle}{\left\langle \frac{1}{v(E)}\phi(\vec{r},\vec{\Omega},E,t) \;\middle|\; W(\vec{r},\vec{\Omega},E,t) \right\rangle} = \frac{\beta_d^{eff}}{\Lambda} \; n(t) \; - \; \lambda_d \bar{c}_d \tag{3.33b}$$

where ρ, β_d^{eff}, Λ, $\bar{c}_d$ are scalar time-dependent parameters defined in equations 3.34b to

3.34f as follows:

$$H(t) = \left\langle \left(\frac{\chi_p(E)}{4\pi}(1-\beta) + \sum_{d=1}^{G_d} \frac{\chi_d(E)}{4\pi}\beta_d \right) \mathcal{F}\ \phi(\vec{r},\vec{\Omega},E,t) \,\middle|\, W(\vec{r},\vec{\Omega},E,t) \right\rangle \tag{3.34a}$$

$$\rho(t) = \frac{1}{H(t)} \left\langle \Big(-\mathcal{L} - \mathcal{T} + \mathcal{S} \Big)\ \phi(\vec{r},\vec{\Omega},E,t) \,\middle|\, W(\vec{r},\vec{\Omega},E,t) \right\rangle + 1 \tag{3.34b}$$

$$\beta_d^{eff}(t) = \frac{1}{H(t)} \left\langle \frac{\chi_d(E)}{4\pi}\beta_d\ \mathcal{F}\ \phi(\vec{r},\vec{\Omega},E,t) \,\middle|\, W(\vec{r},\vec{\Omega},E,t) \right\rangle \tag{3.34c}$$

$$\beta^{eff}(t) = \sum_{d=1}^{G_d} \beta_d^{eff}(t) \tag{3.34d}$$

$$\Lambda(t) = \frac{1}{H(t)} \left\langle \frac{1}{v(E)}\phi(\vec{r},\vec{\Omega},E,t) \,\middle|\, W(\vec{r},\vec{\Omega},E,t) \right\rangle \tag{3.34e}$$

$$\bar{c}_d(t) = \frac{\left\langle \frac{\chi_d(E)}{4\pi}\ C_d(\vec{r},t) \,\middle|\, W(\vec{r},\vec{\Omega},E,t) \right\rangle}{\left\langle \frac{1}{v(E)}\phi(\vec{r},\vec{\Omega},E,t) \,\middle|\, W(\vec{r},\vec{\Omega},E,t) \right\rangle} \tag{3.34f}$$

By now, we arrived to equations 3.33a and 3.33b that are really close to the well-known point kinetic equations, which we want to obtain. Nonetheless, the temporal partial derivatives in the numerators forbid us from condensing these integrals as in the parameters defined between equations 3.34b to 3.34f. However, applying to the numerators of the uncondensed terms in equations 3.33a and 3.33b the *chain rule* and the *Leibniz integral rule* we are able to obtain equations 3.35a and 3.35b.

$$\begin{aligned} \frac{d}{dt}\left\langle \frac{1}{v(E)}\phi(\vec{r},\vec{\Omega},E,t) \,\middle|\, W(\vec{r},\vec{\Omega},E,t) \right\rangle = \\ \left\langle \frac{1}{v(E)}\frac{\partial \phi}{\partial t}(\vec{r},\vec{\Omega},E,t) \,\middle|\, W(\vec{r},\vec{\Omega},E,t) \right\rangle \\ + \left\langle \frac{1}{v(E)}\phi(\vec{r},\vec{\Omega},E,t) \,\middle|\, \frac{\partial W}{\partial t}(\vec{r},\vec{\Omega},E,t) \right\rangle \end{aligned} \tag{3.35a}$$

$$\begin{aligned} \frac{d}{dt}\left\langle \frac{\chi_d(E)}{4\pi}\ C_d(\vec{r},t) \,\middle|\, W(\vec{r},\vec{\Omega},E,t) \right\rangle = \\ \left\langle \frac{\chi_d(E)}{4\pi}\ \frac{\partial C_d}{\partial t}(\vec{r},t) \,\middle|\, W(\vec{r},\vec{\Omega},E,t) \right\rangle \\ + \left\langle \frac{\chi_d(E)}{4\pi}\ C_d(\vec{r},t) \,\middle|\, \frac{\partial W}{\partial t}(\vec{r},\vec{\Omega},E,t) \right\rangle \end{aligned} \tag{3.35b}$$

From equations 3.35a and 3.35b is easy to see that if W is independent of time, i.e., $W(\vec{r},\vec{\Omega},E,t) = W_0(\vec{r},\vec{\Omega},E)$ then $\frac{\partial W}{\partial t}(\vec{r},\vec{\Omega},E,t) = 0$ and the equivalent expressions shown by equations 3.36a and 3.36b can be used. Further explanation of this time-independent weight condition is discussed in section 3.1.2.5.

$$\frac{d}{dt}\left\langle \frac{1}{v(E)}\phi(\vec{r},\vec{\Omega},E,t) \,\middle|\, W_0(\vec{r},\vec{\Omega},E)\right\rangle = \left\langle \frac{1}{v(E)}\frac{\partial\phi}{\partial t}(\vec{r},\vec{\Omega},E,t) \,\middle|\, W_0(\vec{r},\vec{\Omega},E)\right\rangle \tag{3.36a}$$

$$\frac{d}{dt}\left\langle \frac{\chi_d(E)}{4\pi}\ C_d(\vec{r},t) \,\middle|\, W_0(\vec{r},\vec{\Omega},E)\right\rangle = \left\langle \frac{\chi_d(E)}{4\pi}\ \frac{\partial C_d}{\partial t}(\vec{r},t) \,\middle|\, W_0(\vec{r},\vec{\Omega},E)\right\rangle \tag{3.36b}$$

Replacing 3.36a and 3.36b in 3.33a and 3.33b we obtain equations 3.37a and 3.37b.

$$\frac{dn(t)}{dt} = \left[\frac{\rho-\beta^{eff}}{\Lambda} - \frac{\frac{d}{dt}\left\langle \frac{1}{v(E)}\phi(\vec{r},\vec{\Omega},E,t) \,\middle|\, W_0(\vec{r},\vec{\Omega},E)\right\rangle}{\left\langle \frac{1}{v(E)}\phi(\vec{r},\vec{\Omega},E,t) \,\middle|\, W_0(\vec{r},\vec{\Omega},E)\right\rangle}\right] n(t) + \sum_{d}^{G_d}\lambda_d\bar{c}_d(t) \tag{3.37a}$$

$$\frac{\frac{d}{dt}\left\langle \frac{\chi_d(E)}{4\pi}\ C_d(\vec{r},t) \,\middle|\, W_0(\vec{r},\vec{\Omega},E)\right\rangle}{\left\langle \frac{1}{v(E)}\phi(\vec{r},\vec{\Omega},E,t) \,\middle|\, W_0(\vec{r},\vec{\Omega},E)\right\rangle} = \frac{\beta_d^{eff}}{\Lambda}\ n(t) - \lambda_d\bar{c}_d(t) \tag{3.37b}$$

Now, arbitrarily forcing the denominator in equations 3.37a and 3.37b to be constant[6], as shown in equation (3.38), the time derivative in the numerator of the second term in the RHS of equation 3.37a vanishes. Moreover, we can move the now constant denominator in the term on the LHS of equation 3.37b inside the time derivative.

$$\left\langle \frac{1}{v(E)}\phi(\vec{r},\vec{\Omega},E,t) \,\middle|\, W_0(\vec{r},\vec{\Omega},E)\right\rangle = constant \tag{3.38}$$

Introducing these modifications and defining $\bar{c}_d$ as in equation 3.34f, we can obtain a set of equations for the neutron flux amplitude as shown in equations 3.39a and 3.39b:

$$\boxed{\begin{aligned} \frac{dn(t)}{dt} &= \frac{\rho-\beta^{eff}}{\Lambda}\ n(t) + \sum_{d}^{G_d}\lambda_d\bar{c}_d(t) && (3.39a)\\ \frac{d\bar{c}_d}{dt}(t) &= \frac{\beta_d^{eff}}{\Lambda}\ n(t) - \lambda_d\bar{c}_d(t) && (3.39b)\end{aligned}}$$

In Section 3.1.2.4, the justification of this choice will be discussed from a mathematical point of view and in Section 3.1.2.6.1 a physical interpretation will be given. Albeit, this does not intend to be a rigorous analysis of the normalization but only an approach to justify every step in the method and discuss later the numerical implementation of the method and the approximations that have been made.

[6] As it will be explained later, this step is not compulsory but it will simplify the implementation of the method.

Note that the equations 3.39a and 3.39b are similar now to the point kinetics equations but the coefficients are different since they have to be calculated accordingly to the weight function, which may be different from the adjoint flux normally used in the point kinetics approach. It is also important to stress that so far, at this point we have made no assumptions on the physics but rather we have used some mathematical tricks to derive the set of equations 3.30a, 3.30b, 3.39a and 3.39b following the weight constrain and the normalization constraint 3.38 and definitions in 3.34a to 3.34f. For now, it may seem as it is just a complex way of solving the original neutron transports equation 3.1 and the precursors equation 3.6 but this splitting will be advantageous to relax the computational demand, specially when is used combined with some of the simplifications shown in section 3.1.2.8.

3.1.2.3 Liquid Fuel

It is important to recall that some nuclear systems studied in criticality accidents contain a liquid fissile such as the reactors using a liquid fuel (e.g. SILENE or a MSR) or at some stage of the fuel fabrication (Tokai-mura accident). Therefore, the development of a Quasi-Static Method approach requires to consider phenomena such as the delayed neutron precursors motion. In particular, this new phenomenology will require using equation 3.7 rather than the classic precursors concentration equation 3.6 for solid fuels.

A general Quasi-Static Method approach for liquid fuel has been proposed by Dulla et al. [44], where a factorization similar to the one used in equation 3.29 for the angular flux is applied to each family of precursors. In this work, an alternative method is proposed and implemented. The implementation of this method is relatively simpler but it requires introducing some additional hypotheses concerning the limits of the fluid system that will be discussed at the end of this section. Firstly, the shape equation 3.30a rests the same and 3.30b is modified in the same manner as equation 3.7. Then, following the same process as the previous section for deriving the amplitude equations we arrived to two new integrals defined as follows:

$$\left\langle \frac{\chi_d(E)}{4\pi}\nabla\cdot\left[D_d\nabla C_d(\vec{r},E)\right] \;\middle|\; W_0(\vec{r},\vec{\Omega},E)\right\rangle \tag{3.40}$$

$$\left\langle \frac{\chi_d(E)}{4\pi}\vec{u}(\vec{r})\cdot\nabla C_d(\vec{r},E) \;\middle|\; W_0(\vec{r},\vec{\Omega},E)\right\rangle \tag{3.41}$$

At this point, we introduce the first simplification which consists in assuming a space independent weight function, i.e., $W_0(\vec{r},\vec{\Omega},E) = \hat{W}_0(\vec{\Omega},E)$. By using Gauss's Theorem the volume integral of equation 3.40 can be written as:

$$\begin{aligned} \int_V \nabla\cdot[D_d\nabla C_d(\vec{r},E)]\ \hat{W}_0(\vec{\Omega},E)\ dV &= \int_V \nabla\cdot\left[D_d\nabla C_d(\vec{r},E)\hat{W}_0(\vec{\Omega},E)\right]\ dV \\ &= \hat{W}_0(\vec{\Omega},E)\int_S [D_d\nabla C_d(\vec{r},E)]\cdot\vec{n}\ dS \qquad (3.42) \end{aligned}$$

In addition, if the liquid fissile can be approximated as an incompressible flow, the continuity equation is valid and $\nabla\cdot\vec{u} = 0$ as it will be derived in Section 3.3.3. Then, equation 3.41 becomes:

$$\begin{aligned} \int_V \vec{u}(\vec{r})\cdot\nabla C_d(\vec{r},E)\ \hat{W}_0(\vec{\Omega},E)\ dV &= \int_V \nabla\cdot\left[\vec{u}(\vec{r})C_d(\vec{r},E)\hat{W}_0(\vec{\Omega},E)\right]\ dV \\ &= \hat{W}_0(\vec{\Omega},E)\int_S [\vec{u}(\vec{r})C_d(\vec{r},E)]\cdot\vec{n}\ dS \qquad (3.43) \end{aligned}$$

Finally, if the liquid system is surrounded by walls, non-slip boundary conditions for the fluid (i.e. $\vec{u} = 0$) can be used and the precursors do not flow through the walls (i.e. $\nabla C_d = 0$), then both integrals (equations 3.40 and 3.41) are zero. By using these hypotheses, we recover the amplitude equations 3.39 once again. Note that the additional hypotheses that we have introduced in our model to simplify the equations are justified in most of the systems of interest. In particular, the geometry of most systems studied in criticality accidents (or nuclear reactor systems) can be defined in such a way that the limits of the system of interest can be approximated as walls. For example, in the case of a tank with a free surface between a liquid and a gas (e.g. water and air) there will not be non-slip conditions on the liquid-gas interface. Nonetheless, the Control Volume (domain of the system) can be expanded up to a size where there is a zero-velocity boundary, e.g, up to the walls of the tank including the gas inside, then the same equations stated before can be used taking into account the gas velocity.

3.1.2.4 Normalization Constant

As explained before, setting $\left\langle \frac{1}{v(E)}\phi(\vec{r},\vec{\Omega},E,t) \mid W(\vec{r},\vec{\Omega},E,t)\right\rangle = \textit{constant}$ serves as a mathematical trick to obtain the classical set of ODEs usually referred to as point kinetic equations [45] when the weight function is the adjoint neutron angular flux.

In addition, note that the factorization in equation 3.29 does not guarantee the uniqueness of the splitting. Since it is arbitrary, we could multiply $n(t)$ by a constant α and divide $\phi(\vec{r},\vec{\Omega},E,t)$ by the same constant and obtain the same angular flux $\psi(\vec{r},\vec{\Omega},E,t)$ as shown in equation 3.44: [46][43][47].

$$\begin{aligned}\psi(\vec{r},\vec{\Omega},E,t) &= n(t)\phi(\vec{r},\vec{\Omega},E,t)\\ &= \left[n(t)\boldsymbol{\alpha}\right]\left[\frac{\phi(\vec{r},\vec{\Omega},E,t)}{\boldsymbol{\alpha}}\right]\\ &= \hat{n}(t)\hat{\phi}(\vec{r},\vec{\Omega},E,t) \end{aligned} \tag{3.44}$$

In order to ensure this uniqueness the normalization constraint discussed earlier is used. Even though in this work this constraint is applied, it is possible to derive unconstrained kinetic equations introducing slight modifications as can be seen in equations 3.45 to 3.47 [48]. Note that the shape equations and the rest of the kinetic parameters rest the same:

$$\frac{dn(t)}{dt} = \left[\frac{\rho-\beta^{eff}}{\Lambda} - \lambda_s\right] n(t) + \sum_{d}^{G_d}\lambda_d\bar{c}_d(t) \tag{3.45}$$

$$\frac{d\bar{c}_d}{dt}(t) = \frac{\beta_d^{eff}}{\Lambda} n(t) - (\lambda_d + \lambda_s)\bar{c}_d(t) \tag{3.46}$$

$$\lambda_s = \frac{\frac{d}{dt}\left\langle \frac{1}{v(E)}\phi(\vec{r},\vec{\Omega},E,t) \,\middle|\, W(\vec{r},\vec{\Omega},E,t)\right\rangle}{\left\langle \frac{1}{v(E)}\phi(\vec{r},\vec{\Omega},E,t) \,\middle|\, W(\vec{r},\vec{\Omega},E,t)\right\rangle} \tag{3.47}$$

3.1.2.5 Weight Function

The choice of the weight function in the Quasi-Static Method is completely arbitrary. Indeed it could be time-dependent as developed by Becker [49] from a variational principle. Nonetheless, when the importance function (adjoint neutron angular flux) is used, the classic point kinetics coefficients are obtained. Moreover, it was proven by Monier [48][50] that the error in the ratio $\frac{\rho}{\Lambda}(t)$ is reduced by this choice as reproduced here from Monier's work.

Dividing equation 3.34b by equation 3.34e and rearranging terms we obtain equation

3.48

$$\frac{\rho}{\Lambda}(t) \;=\; \frac{\left\langle L(\vec{r},\vec{\Omega},E,t)\phi(\vec{r},\vec{\Omega},E,t) \;\middle|\; W_0(\vec{r},\vec{\Omega},E)\right\rangle}{\mathcal{E}(t)} \tag{3.48}$$

where,

$$L(\vec{r},\vec{\Omega},E,t) \;=\; \Bigg[-\mathcal{L}\;-\;\mathcal{T}\;+\;\mathcal{S}\;+\left(\frac{\chi_p(E)}{4\pi}(1-\beta)+\sum_{d=1}^{G_d}\frac{\chi_d(E)}{4\pi}\beta_d\right)\mathcal{F}\;\Bigg]\;(\vec{r},\vec{\Omega},E,t)$$

$$\mathcal{E}(t) \;=\; \left\langle \frac{1}{v(E)}\phi(\vec{r},\vec{\Omega},E,t) \;\middle|\; W_0(\vec{r},\vec{\Omega},E)\right\rangle$$

where $\mathcal{E}(t)$ is the weighted neutron density. Considering that we begin from a critical configuration, then at $t = t_0$ the operator $L(\vec{r},\vec{\Omega},E,t)$ is the steady-state operator that satisfies equation 3.49 equivalent to equations 3.30a and 3.30b in steady-state.

$$L(\vec{r},\vec{\Omega},E,t_0)\;\phi(\vec{r},\vec{\Omega},E,t_0) = 0 \tag{3.49}$$

And, therefore,

$$\frac{\rho}{\Lambda}(t_0) \;=\; \frac{\left\langle L(\vec{r},\vec{\Omega},E,t_0)\phi(\vec{r},\vec{\Omega},E,t_0) \;\middle|\; W_0(\vec{r},\vec{\Omega},E)\right\rangle}{\mathcal{E}(t_0)} \;=\; 0 \tag{3.50}$$

The temporal dependence of both $L(\vec{r},\vec{\Omega},E,t)$ and $\phi(\vec{r},\vec{\Omega},E,t)$ can be expressed as an initial value plus a perturbation term:

$$L(\vec{r},\vec{\Omega},E,t) \;=\; L(\vec{r},\vec{\Omega},E,t_0) + \delta L(\vec{r},\vec{\Omega},E,t) \tag{3.51}$$

$$\phi(\vec{r},\vec{\Omega},E,t) \;=\; \phi(\vec{r},\vec{\Omega},E,t_0) + \delta\phi(\vec{r},\vec{\Omega},E,t) \tag{3.52}$$

Now, substituting the definitions 3.51 and 3.52 in equation 3.48 we obtain

$$\begin{aligned}\frac{\rho}{\Lambda}(t) = \Bigg[&\left\langle L(\vec{r},\vec{\Omega},E,t_0)\phi(\vec{r},\vec{\Omega},E,t_0) \;\middle|\; W_0(\vec{r},\vec{\Omega},E)\right\rangle\\ &+\left\langle \delta L(\vec{r},\vec{\Omega},E,t_0)\phi(\vec{r},\vec{\Omega},E,t_0) \;\middle|\; W_0(\vec{r},\vec{\Omega},E)\right\rangle\\ &+\left\langle L(\vec{r},\vec{\Omega},E,t_0)\delta\phi(\vec{r},\vec{\Omega},E,t_0) \;\middle|\; W_0(\vec{r},\vec{\Omega},E)\right\rangle\\ &+\left\langle \delta L(\vec{r},\vec{\Omega},E,t_0)\delta\phi(\vec{r},\vec{\Omega},E,t_0) \;\middle|\; W_0(\vec{r},\vec{\Omega},E)\right\rangle\Bigg]\left[\frac{1}{\mathcal{E}(t_0)\;+\;\delta\mathcal{E}(t)}\right]\end{aligned} \tag{3.53}$$

If the changes in $L(\vec{r}, \vec{\Omega}, E, t)$ are small, the second order term $\delta L \delta \phi$ can be neglected. Since the initial flux shape function satisfies the initial flux shape equation 3.49 the first term in equation 3.53 is zero. When the perturbation term $\delta \mathcal{E}(t)$ is brought to the numerator using a truncated Taylor series, only second order terms results. Therefore, that term is dropped. Only two terms remain and the equation reduces to equation 3.54

$$\frac{\rho}{\Lambda}(t) = \frac{1}{\mathcal{E}(t_0)} \Big[\Big\langle \delta L(\vec{r}, \vec{\Omega}, E, t_0) \phi(\vec{r}, \vec{\Omega}, E, t_0) \mid W_0(\vec{r}, \vec{\Omega}, E) \Big\rangle + \Big\langle L(\vec{r}, \vec{\Omega}, E, t_0) \delta\phi(\vec{r}, \vec{\Omega}, E, t_0) \mid W_0(\vec{r}, \vec{\Omega}, E) \Big\rangle \Big] \tag{3.54}$$

It can be seen that any variation of $\phi(\vec{r}, \vec{\Omega}, E, t)$ produces a first-order variation in ρ/Λ. But, if the adjoint flux $\phi^+(\vec{r}, \vec{\Omega}, E, t)$ (defined in equation 3.55) is used this dependence can be eliminated.

$$L^+(\vec{r}, \vec{\Omega}, E, t_0)\ \phi^+(\vec{r}, \vec{\Omega}, E, t_0) = 0 \tag{3.55}$$

where $L^+(\vec{r}, \vec{\Omega}, E, t_0)$ is the adjoint operator defined such that it satisfies equation 3.56:

$$\Big\langle L^+(\vec{r}, \vec{\Omega}, E, t_0)\ v(\vec{r}, \vec{\Omega}, E) \mid u(\vec{r}, \vec{\Omega}, E) \Big\rangle = \Big\langle L(\vec{r}, \vec{\Omega}, E, t_0)\ u(\vec{r}, \vec{\Omega}, E) \mid v(\vec{r}, \vec{\Omega}, E) \Big\rangle \tag{3.56}$$

where $v(\vec{r}, \vec{\Omega}, E)$ and $u(\vec{r}, \vec{\Omega}, E)$ are arbitrary functions.

Applying equation 3.56 to the second term on the RHS of equation 3.54 we obtain 3.57.

$$\frac{\rho}{\Lambda}(t) = \frac{1}{\mathcal{E}(t_0)} \Big[\Big\langle \delta L(\vec{r}, \vec{\Omega}, E, t_0) \phi(\vec{r}, \vec{\Omega}, E, t_0) \mid W_0(\vec{r}, \vec{\Omega}, E) \Big\rangle + \Big\langle L^+(\vec{r}, \vec{\Omega}, E, t_0) W_0(\vec{r}, \vec{\Omega}, E) \mid \delta\phi(\vec{r}, \vec{\Omega}, E, t) \Big\rangle \Big] \tag{3.57}$$

If the adjoint flux is used as the weight function $W_0(\vec{r}, \vec{\Omega}, E) = \phi^+(\vec{r}, \vec{\Omega}, E, t_0)$ the second term in equation 3.57 is zero and we obtain equation 3.58.

$$\frac{\rho}{\Lambda}(t) = \frac{1}{\mathcal{E}(t_0)} \Big\langle \delta L(\vec{r}, \vec{\Omega}, E, t_0) \phi(\vec{r}, \vec{\Omega}, E, t_0) \mid W_0(\vec{r}, \vec{\Omega}, E) \Big\rangle \tag{3.58}$$

This way, the reactivity calculation does not depend on knowing $\delta\phi(\vec{r}, \vec{\Omega}, E, t)$.

3.1.2.6 Physical Interpretation

3.1.2.6.1 Normalization Condition

If we consider that $n(t)\ \phi(\vec{r},\vec{\Omega},E,t)/v(E)$ is the neutron density, then the value of $\left\langle \frac{1}{v(E)}\phi(\vec{r},\vec{\Omega},E,t) \mid W(\vec{r},\vec{\Omega},E,t)\right\rangle$ is asymptotically proportional to the fundamental mode power level. So, for a small perturbation, any asymptotic change in power level is expressed by a variation of the amplitude function $n(t)$, it having required that the factor $\left\langle \frac{1}{v(E)}\phi(\vec{r},\vec{\Omega},E,t) \mid W(\vec{r},\vec{\Omega},E,t)\right\rangle$ be time-independent (Henry [45]). Then, when the flux rises on an asymptotic period (all other modes vanishes), the shape ϕ is independent of time and the constant constraint from equation 3.38 on $\left\langle \frac{1}{v(E)}\phi(\vec{r},\vec{\Omega},E,t) \mid W(\vec{r},\vec{\Omega},E,t)\right\rangle$ is fulfilled automatically.

3.1.2.6.2 Kinetics Parameters

As previously discussed, the definition of the weight function will determine the physical meaning of the parameters 3.34b to 3.34f. If the adjoint flux is used as the weight function, these parameters will have the classical interpretation given in the reactor point kinetics model. Otherwise, these parameters do not have the standard meaning of the point kinetics model but this should be relativized since the total flux (Equation 3.29) can be obtained regardless the chosen weight function (by solving the set of shape and amplitude equations at the same time). If the adjoint-weighted parameters are desired, they can be straightforwardly calculated afterwards from the flux solution. Moreover, different weighted (subscript t) parameters can be related as in equation (10) as discussed by Henry in [45].

$$\rho = \rho_t + \Lambda\frac{dn/dt}{n} + \Lambda_t\frac{dn_t/dt}{n_t} + \frac{Q_t\Lambda_t H_t - Q\Lambda H}{Hn} \tag{3.59}$$

where H, Λ and ρ are the parameters define in equations 3.34a, 3.34e and 3.34b respectively, n is the amplitude function and Q an external source. The subscript t denotes that the parameters are weighted by a function different from the parameters without subscript.

This means that in the absence of a source Q and in slow transients ($dn/dt = 0$) the reactivity value will be approximately independent of the weight function.

Nevertheless, it is important to stress that any choice for the normalization is valid as long as all equation are defined consistently to this choice but it does not necessarily have a physical interpretation.

3.1.2.7 Summary

To summarize, the multi-variable system of partial integro-differential equations that will be used to solve the shape function $\phi(\vec{r},\vec{\Omega},E,t)$ are presented in equations 3.60 for solid precursors (equation 3.60b) and liquid precursors (3.60c).

Then, in 3.61, the amplitude equations (similar to the point kinetics) associated to the shape equations are shown, which will be used for solving the amplitude $n(t)$ of the factorization presented in 3.29. All of this given the parameters condensation stated in 3.62 following the assumptions made in the derivation of the method.

Shape Equations

$$\begin{aligned}\frac{1}{v(E)}\frac{\partial\phi}{\partial t}(\vec{r},\vec{\Omega},E,t) \;=\; &\Bigg[-\mathcal{L} - \mathcal{T} + \mathcal{S} - \frac{1}{v(E)}\frac{1}{n(t)}\frac{dn(t)}{dt}\\ &+ \frac{\chi_p(E)}{4\pi}(1-\beta)\mathcal{F}\Bigg]\,\phi(\vec{r},\vec{\Omega},E,t) \;+\; \frac{1}{n(t)}\sum_{d=1}^{G_d}\frac{\chi_d(E)}{4\pi}\lambda_d C_d(\vec{r},t)\end{aligned}\tag{3.60a}$$

$$\frac{\partial C_d}{\partial t}(\vec{r},t) = \beta_d \iint \mathcal{F}\phi(\vec{r},\vec{\Omega},E,t)dE\;d\vec{\Omega}\;n(t) \;-\; \lambda_d C_d(\vec{r},t) \qquad d = 1 \;to\; G_d \tag{3.60b}$$

$$\begin{aligned}\frac{\partial C_d}{\partial t}(\vec{r},t) \;+\; \vec{u}\cdot\nabla C_d(\vec{r},t) \;=\; &\nabla\cdot\left[D_d \nabla C_d(\vec{r},t)\right]\\ &+ \beta_d \iint \mathcal{F}\phi(\vec{r},\vec{\Omega},E,t)dE\;d\vec{\Omega}\;n(t) \;-\; \lambda_d C_d(\vec{r},t) \qquad d = 1 \;to\; G_d\end{aligned}\tag{3.60c}$$

Amplitude Equations

$$\frac{dn(t)}{dt} \;=\; \frac{\rho-\beta^{eff}}{\Lambda}\;n(t) \;+\; \sum_{d}^{G_d}\lambda_d\bar{c}_d(t) \tag{3.61a}$$

$$\frac{d\bar{c}_d}{dt}(t) \;=\; \frac{\beta_d^{eff}}{\Lambda}\;n(t) \;-\; \lambda_d\bar{c}_d(t) \tag{3.61b}$$

Parameters Condensation

$$H(t) = \left\langle \left(\frac{\chi_p(E)}{4\pi}(1-\beta) + \sum_{d=1}^{G_d} \frac{\chi_d(E)}{4\pi}\beta_d \right) \mathcal{F}\ \phi(\vec{r},\vec{\Omega},E,t) \ \middle|\ W_0(\vec{r},\vec{\Omega},E) \right\rangle \tag{3.62a}$$

$$\rho(t) \ = \ \frac{1}{H(t)} \left\langle \left(-\,\mathcal{L} \ - \ \mathcal{T} \ + \ \mathcal{S} \ \right) \phi(\vec{r},\vec{\Omega},E,t) \ \middle|\ W_0(\vec{r},\vec{\Omega},E) \right\rangle + 1 \tag{3.62b}$$

$$\beta_d^{eff}(t) = \frac{1}{H(t)} \left\langle \frac{\chi_d(E)}{4\pi}\beta_d\ \mathcal{F}\ \phi(\vec{r},\vec{\Omega},E,t) \ \middle|\ W_0(\vec{r},\vec{\Omega},E) \right\rangle \tag{3.62c}$$

$$\beta^{eff}(t) = \sum_{d=1}^{G_d} \beta_d^{eff}(t) \tag{3.62d}$$

$$\Lambda(t) = \frac{1}{H(t)} \left\langle \frac{1}{v(E)}\phi(\vec{r},\vec{\Omega},E,t) \ \middle|\ W_0(\vec{r},\vec{\Omega},E) \right\rangle \tag{3.62e}$$

$$\bar{c}_d(t) = \frac{\left\langle \frac{\chi_d(E)}{4\pi}\ C_d(\vec{r},t) \ \middle|\ W_0(\vec{r},\vec{\Omega},E) \right\rangle}{\left\langle \frac{1}{v(E)}\phi(\vec{r},\vec{\Omega},E,t) \ \middle|\ W_0(\vec{r},\vec{\Omega},E) \right\rangle} \tag{3.62f}$$

Assumptions

The main assumptions to inferred the systems of equations proposed are:

- The factorization of $\psi(\vec{r},\vec{\Omega},E,t)$ is possible.
- The weight function is independent of time $W(\vec{r},\vec{\Omega},E,t) \ = \ W_0(\vec{r},\vec{\Omega},E)$.
- The constraint $\left\langle \frac{1}{v(E)}\phi(\vec{r},\vec{\Omega},E,t) \ \middle|\ W(\vec{r},\vec{\Omega},E,t) \right\rangle \ = \ \textit{constant}$ is applied[7].
- Incompressible fluid with non-slip boundary condition and no loss of precursors through the boundaries (for liquid fissile).

3.1.2.8 Common simplifications used to solve the Quasi-Static Method equations

In this section, we will introduce some common approximations made in the Quasi-Static Method. These approximations will simplify its numerical implementation and more important they will allow us to take full advantage of the mathematical model shown before and thus relax the computational demand of the method numerical implementation.

3.1.2.8.1 Improved Quasi-Static Method

If we choose not to introduce any simplification and solve the full set of Quasi-Static

[7] Notice that, in the case of the system following the stable period this condition is automatically fulfilled since $\phi(\vec{r},\vec{\Omega},E,t)$ is already independent of time and the factorization $\psi(\vec{r},\vec{\Omega},E,t) \ = \ n(t)\phi(\vec{r},\vec{\Omega},E,t)$ becomes a separation of variables [45]

Method equations 3.60 and 3.61 with its constrains, the method is referred to as **Improved Quasi-Static Method**.

3.1.2.8.2 Original Quasi-Static Method

To simplify the numerical integration a common approximation is to assume that the rate of change of the flux shape function is significantly smaller than the rate of change of the amplitude, i.e. $\partial\phi/\partial t \; << (dn/n)/dt \rightarrow \partial\phi/\partial t = 0$. This approximation is known as **the Original Quasi-Static Method**. This way, the equation 3.60a can be solved as being steady-state for the flux shape function as can be seen from equation 3.63 [42]. Moreover, the term containing the derivative of the flux amplitude can be seen as a "correction" on the collision cross section. The precursors equation remain unchanged, i.e., no approximation is made:

$$\left[-\mathcal{L} \; - \; \mathcal{T} \; + \; \mathcal{S} \; - \frac{1}{v(E)} \frac{1}{n(t)} \frac{dn(t)}{dt} \; + \; \frac{\chi_p(E)}{4\pi}(1-\beta)\mathcal{F} \right] \phi(\vec{r}, \vec{\Omega}, E, t) \\ + \frac{1}{n(t)} \sum_{d=1}^{G_d} \frac{\chi_d(E)}{4\pi} \lambda_d C_d(\vec{r}, t) \; = \; 0 \qquad (3.63)$$

3.1.2.8.3 Adiabatic Method

In most of the adiabatic variants of the Quasi-Static Method an additional approximation is introduced. This approximation consists in neglecting both time derivatives ($\partial\phi/\partial t = 0$; $dn/dt(t) = 0$) only in the neutron angular flux shape equation 3.63:

$$\left[-\mathcal{L} \; - \; \mathcal{T} \; + \; \mathcal{S} \; + \; \frac{\chi_p(E)}{4\pi}(1-\beta)\mathcal{F} \right] \phi(\vec{r}, \vec{\Omega}, E, t) \\ + \frac{1}{n(t)} \sum_{d=1}^{G_d} \frac{\chi_d(E)}{4\pi} \lambda_d C_d(\vec{r}, t) \; = \; 0 \qquad (3.64)$$

Even thought this equation neglects an important part of the coupling between equations 3.60a and 3.61a, these equations are still related by the feedback in the removal and fission source operators and the precursors concentration [42]. Moreover, with the adiabatic approximation, the original transient equation 3.60a has now be transformed in a steady-state problem as can be seen in equation 3.64. Note that if a steady-state solution of the equation 3.64 exist (i.e.the system is not prompt super-critical), the flux shape can be calculated with a Monte Carlo code using for example the external source method. This approach may not be the most computationally efficient method but it could be used for sub-critical systems or for systems that are super-critical but well below the prompt criticality (as long as it exist a steady-state solution of equation 3.64).

The resolution of equation 3.64 can be further simplified by introducing an additional approximation (not inherent to the adiabatic approximation) that consists in considering that the precursors concentrations are in equilibrium with the fission source:

$$\frac{\chi_p(E)}{4\pi}(1-\beta)\mathcal{F}\ \phi(\vec{r},\vec{\Omega},E,t) \ + \ \frac{1}{n(t)}\sum_{d=1}^{G_d}\frac{\chi_d(E)}{4\pi}\lambda_d C_d(\vec{r},t) \ \rightarrow \\ \frac{1}{4\pi}\left[\chi_p(E)(1-\beta)+\chi_d(E)\beta\right]\ \mathcal{F}\ \phi(\vec{r},\vec{\Omega},E,t) \quad (3.65)$$

The benefit of this additional approximation (only used in for the calculation of the flux shape) is obvious, the problem can be transformed in the standard neutronics k_{eff} eigenvalue problem as follows:

$$\left[\mathcal{L}\ +\ \mathcal{T}\ -\ \mathcal{S}\ \right]\ \phi(\vec{r},\vec{\Omega},E,t)\ =\ \frac{1}{k_{eff}}\frac{1}{4\pi}\left[\chi_p(E)(1-\beta)+\chi_d(E)\beta\right]\ \mathcal{F}\ \phi(\vec{r},\vec{\Omega},E,t) \qquad (3.66)$$

Equation 3.66 has several advantages, first it ensures the existence of a steady non-trivial solution. Secondly the delayed neutrons source can be calculated directly from the fission source. Several standard methods are available to solve this problem. In particular, equation 3.66 can be simply calculated from a Monte Carlo code using the k-eigenvalue method. Nevertheless, by solving the flux shape using equation 3.66, we are considering that the precursors distribution is in equilibrium with the instantaneous fundamental mode of the neutron flux shape. We do not distinguish then the shape of the prompt neutron fission source from the shape of the delayed neutron source. This approximation may introduce a significant error in transients where the delayed neutron precursors have an important effect on the flux shape and their spatial distribution is far from equilibrium with the neutron flux. In such transients equation 3.64 should be preferred.

The numerical resolution of equation 3.64 can be simplified by transforming equation 3.64 in an eigenvalue problem. To do that, the delayed neutron source of equation 3.64 is estimated with the fission rate operator and corrected with the normalized precursors distribution shape. This replacement allows rewriting the equations as follows:

$$\left[-\mathcal{L} - \mathcal{T} + \mathcal{S} - \frac{1}{v(E)}\frac{1}{n(t)}\frac{dn(t)}{dt} + \frac{\chi_p(E)}{4\pi}(1-\beta)\mathcal{F}\right]\phi(\vec{r},\vec{\Omega},E,t) + \sum_{d=1}^{G_d}\frac{\chi_d(E)}{4\pi}\beta_d\int\mathcal{F}\phi(\vec{r},\vec{\Omega},E,t)dV\frac{C_d(\vec{r},t)}{C_{max}} = 0 \quad (3.67)$$

This equation can now be solved with most neutronics standard solvers. In particular a k-eigenvalue method in Monte Carlo code can be used with some small modifications. As we will discuss in Chapter 4, to obtain the solution of equation 3.67 the Monte Carlo k-eigenvalue solver will have to be modified. In particular, the neutron sampling algorithm will have to use the equation 3.61 to setup the position of delayed neutrons rather than using the neutron source obtained in the previous batch.

Since equation 3.67 does not assume equilibrium conditions between the neutron precursors distributions and the neutron flux, it can therefore be used in more general cases as for example in the case of a system with a liquid fissile such as a liquid fuel. As we have seen in section 3.1.2.3 and from equation 3.7, the spatial distribution of each precursor family can be heavily distorted in such systems and using equation 3.66 would introduce a significant error.

Note that while equation 3.67 uses the correct delayed neutron precursors distribution, it still has one minor approximation: the probability of the delayed neutron emission at a given position is calculated using the physical delayed neutron fraction and the neutron flux rather than the precursors concentration and the decay probability. In most situations, the error introduced by this last approximation in the calculation of the neutron shape function is small as we will see in the results of Chapter 5. This can be explained for various reasons: the velocity of motion of the precursors is relatively small (order of a few centimeters per second), neutron precursors will have a relatively low effect during transients resulting from larger reactivity insertions and, on the contrary, during slow transients (low reactivity insertion) the approximation will work better. Nevertheless, the error can be avoided by introducing a straightforward modification on the criterion used by the Monte Carlo code for sampling the neutrons. We chose not to do these modifications to minimize the changes on the original Serpent code and since the error was small for the studied cases.

Finally, it is important to remark that independently of which equation is used (either equation 3.66 or equation 3.67), the precursors time derivative in either case is not neglected (i.e. $dC_d/dt \neq 0$) in equations 3.6 and 3.67. The amplitude equations also rest the same (equations 3.1.2.7). Therefore, time evolution is followed mainly via the amplitude equations and the precursors equations.

3.1.2.9 Monte Carlo Quasi-Static Method

In this work, a Quasi-Static Method was implemented with a Monte Carlo method using the adiabatic form to simplify the developments. An important advantage of the adiabatic approximation is that the resulting equations 3.66 and 3.67 are equivalent to the transport problem solved by the criticality Monte Carlo calculation. Therefore a transient calculation of the neutron flux in the Adiabatic QSM can be performed by determining the flux shape function with a Monte Carlo code (using the standard criticality calculation mode) while the flux amplitude is determined with a simple ODE solver which numerically integrates the equations 3.1.2.7. This approach has be implemented in our work using Serpent 2 and OpenFOAM codes. The details on the strategy used to couple the Monte Carlo calculations for the flux shape function and with the flux amplitude calculations using OpenFOAM are given in Chapter 4.

Better approximations like the Original Quasi-Static Method can be implemented by adding an external source for the precursors (or correcting the amplitude of the source by modifying the probability of emission (β) of a delayed neutron) and modifying the cross sections by the $(dn/n)/dt$ term. However, it is shown in this work that the adiabatic approximation provides accurate enough results even during very fast transients or with liquid fissile systems as it will be illustrated by the results of Chapter 5. Equation 3.66 can be used for solid fuel systems assuming that the spatial distribution of the delayed neutrons is close to an equilibrium with the neutron flux or that the delayed neutron precursors have not an important effect on the flux shape (such as in some Godiva experiments as we will see in Chapter 5). For the other cases, equation 3.67 should be preferred, which is equivalent to Equation 3.66 at the limit of static liquid fuel ($\vec{u} = \vec{0}$).

3.1.3 Simplified $\mathrm{P_N}$

As discussed at the beginning of this chapter we have selected the Simplified $\mathrm{P_N}$ method as an alternative deterministic method to the Monte Carlo Quasi-Static Method approach described in the previous sections. The $\mathrm{SP_N}$ was chosen because of its relatively simple numerical implementation. In addition, the order one of this method ($\mathrm{SP_1}$) is equivalent, in this formulation, to the diffusion approximation. In this section, we will review the mathematical derivation with special focus on the modifications that were needed to obtain the final equations used in the algorithm implemented in our multi-physics modeling tool. The numerical implementation of the algorithm will be presented in Chapter 4.

3.1.3.1 Method Description

The Simplified $\mathrm{P_N}$ ($\mathrm{SP_N}$) was first proposed by Gelbard [51] based on heuristic arguments to justify the approximation. The $\mathrm{SP_N}$ method is indeed a 3D extension of the slab PN equations but due to the lack of a strong theoretical justification it was not considered until recently as an accurate method for the resolution of the transport equation. Since its first appearance, asymptotic and variational analysis have been done proving the validity of the $\mathrm{SP_N}$ method[52] which, in many cases, gives the same results as the PN equations and in others it represents an improvement over the diffusion approach. However as stated in the work of McClarren [52], this is true as long as the problem is either close to diffusive or locally 1D. The interested reader is encouraged to read McClarren's work as it gives a very good insight of the assumptions, the heuristic derivation as well as the asymptotical and variational derivations, the treatment on the interface and boundary conditions and the time-dependent $\mathrm{SP_N}$ equations. McClarren [52] also provides a brief discussion over the optimal order N of the $\mathrm{SP_N}$ approximation since in some cases the $\mathrm{SP_N}$ approximation is not asymptotic to the transport equation and, therefore, increasing in the order of the approximation does not necessarily means an increase in accuracy.

Most practical derivations of the $\mathrm{SP_N}$ method are for the steady-state reactor analysis, as seen presented in the work of Hamilton and Evans [28] in the framework of CASL DOE Energy Innovation Hub. The latter approach is used here to derive the transient equations. The formal derivation of Gelbard begins with the PN equations for 1-D slab geometry [51]. We will first show the multi-group one-dimensional formulation of the transport equation. As a difference from Hamilton and Evans [28] we will account for the time dependency by adding the time derivative of the neutron angular flux divided by the neutron speed. In addition, the delayed neutron source was incorporated as observed in equation 3.68.

$$\frac{1}{v^g}\frac{\partial\psi^g}{\partial t}(x,\mu) \;+\; \mu\frac{\partial\psi^g}{\partial x}(x,\mu) \;+\; \Sigma^g_{tot}(x)\psi^g(x,\mu) = \sum_{g^{'}=1}^{Ng}\int_{4\pi}\Sigma_s^{gg^{'}}(x,\Omega\cdot\Omega^{'})\;\psi^g(x,\Omega^{'}),d\Omega^{'}$$
$$+\frac{\chi_p^g}{4\pi}(1-\beta)\sum_{g^{'}=1}^{Ng}\int_{4\pi}\nu\Sigma_f^{g^{'}}(x)\;\psi^{g^{'}}(x,\Omega^{'})\,d\Omega^{'} + \frac{\chi_d^g}{4\pi}\sum_{d=1}^{G_d}\lambda_d C_d(x) \quad (3.68)$$

where all quantities are defined to take the multi-group formulation into account using the superscript g to denote the energy group. ψ^g, v^g and Σ_r^g are the angular flux, neutron speed and macroscopic cross section for reaction r, respectively. χ_p^g and χ_d^g are the fractions of prompt and delayed neutrons born in group g. $\Sigma_s^{gg^{'}}$ is the scattering macroscopic cross section from the energy group g' to g, i.e., is the coupling reaction between any two energy groups. Finally, μ is the cosine of the direction angle θ in the lab system, i.e. $\mu = cos\theta$.

The neutron angular flux and the scattering macroscopic cross section are written in these equations using the following Legendre polynomials expansions to obtain the PN equations:

$$\psi(\mu) = \sum_{n=0}^{N}\frac{2n+1}{4\pi}\phi_n P_n(\mu) \quad (3.69)$$

$$\Sigma_s(\mu_0) = \sum_{m=0}^{N}\frac{2m+1}{4\pi}\Sigma_{sm}P_m(\mu_0) \quad (3.70)$$

where $\mu_0 = \Omega\cdot\Omega^{'}$ and $P_l(\Omega\cdot\Omega^{'})$. The spatial dependence of the modes ϕ_n was omitted to simplify notation.

From these definitions is easy to see that the scalar flux will be the order zero of the expansion ($\phi(x) = 2\pi\int_{-1}^{1}\psi(x,\mu'),d\mu' = \phi_0(x)$). Replacing this definitions in equation 3.68 we obtain the PN equations for a slab geometry in Cartesian coordinates as shown in 3.71 (for simplicity the spatial and time dependency are not explicitly shown).

$$\frac{1}{v^g}\frac{\partial\phi_n^g}{\partial t} + \frac{\partial}{\partial x}\left[\frac{n}{2n+1}\phi^g_{n-1} + \frac{n+1}{2n+1}\phi^g_{n+1}\right] + \sum_{g'=1}^{Ng}\left(\Sigma^g_{tot}\delta_{gg'} - \Sigma^{gg'}_{sn}\right)\phi_n^g$$
$$= \left(\chi_p^g(1-\beta)\sum_{g'=1}^{Ng}\nu\Sigma_f^{g^{'}}\phi_n^{g'} + \chi_d^g\sum_{d=1}^{G_d}\lambda_d C_d\right)\delta_{n0} \quad (3.71)$$

For the SP$_{\mathrm{N}}$ equations an ad-hoc replacement was made by Gelbard [51] as pointed out by McClarren [52]: for n odd ϕ_n is replaced by a vector ($\phi_n \rightarrow \vec{\phi}_n = (\phi_n^x, \phi_n^y, \phi_n^z)^t$) and the spatial derivative by a spatial gradient ($d/dx \rightarrow \nabla$), and for n even the spatial derivative is replaced by the spatial divergence ($d/dx \to \nabla\cdot$). This way, we arrived to the multi-group time-dependent SP$_{\mathrm{N}}$ set of equations 3.72.

$$\frac{1}{v^g}\frac{\partial \phi_0^g}{\partial t} + \nabla\cdot\vec{\phi}_1^g + \sum_{g'=1}^{N_g}\left(\Sigma_{tot}^g\delta_{gg'} - \Sigma_{s0}^{gg'}\right)\phi_0^g = \chi_p^g(1-\beta)\sum_{g'=1}^{N_g}\nu\Sigma_f^{g'}\phi_0^{g'} + \chi_d^g\sum_{d=1}^{G_d}\lambda_d C_d \tag{3.72a}$$

$$\frac{1}{v^g}\frac{\partial \vec{\phi}_n^g}{\partial t} + \nabla\left[\frac{n}{2n+1}\phi_{n-1}^g + \frac{n+1}{2n+1}\phi_{n+1}^g\right] + \sum_{g'=1}^{N_g}\left(\Sigma_{tot}^g\delta_{gg'} - \Sigma_{sn}^{gg'}\right)\vec{\phi}_n^g = 0 \quad for\ odd\ n \tag{3.72b}$$

$$\frac{1}{v^g}\frac{\partial \phi_n^g}{\partial t} + \nabla\cdot\left[\frac{n}{2n+1}\vec{\phi}_{n-1}^g + \frac{n+1}{2n+1}\vec{\phi}_{n+1}^g\right] + \sum_{g'=1}^{N_g}\left(\Sigma_{tot}^g\delta_{gg'} - \Sigma_{sn}^{gg'}\right)\phi_n^g = 0 \quad for\ even\ n \tag{3.72c}$$

Similar to Hamilton and Evans [28], but adding into the SP$_{\mathrm{N}}$ equations the transient term and the delayed neutrons source, we can define the multi-group quantities from equations 3.72 in a matrix form:

$$\Phi_n = \left(\phi_0\ \phi_1\ \cdots\ \phi_{N_g}\right)^T \qquad \vec{\Phi}_n = \left(\vec{\phi}_0\ \vec{\phi}_1\ \cdots\ \vec{\phi}_{N_g}\right)^T$$

$$X_d = \left(\chi_d^0\ \chi_d^1\ \cdots\ \chi_d^{N_g}\right)^T \qquad S_d = \sum_{d=1}^{G_d}\lambda_d C_d$$

$$I_{1/v} = \begin{pmatrix} \frac{1}{v^0} & 0 & \cdots & 0 \\ 0 & \frac{1}{v^1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{v^{N_g}} \end{pmatrix} \qquad F = \begin{pmatrix} \chi_p^0\nu\Sigma_f^0 & \chi_p^0\nu\Sigma_f^1 & \cdots & \chi_p^0\nu\Sigma_f^{N_g} \\ \chi_p^1\nu\Sigma_f^0 & \chi_p^1\nu\Sigma_f^1 & \cdots & \chi_p^1\nu\Sigma_f^{N_g} \\ \vdots & \vdots & \ddots & \vdots \\ \chi_p^{N_g}\nu\Sigma_f^0 & \chi_p^{N_g}\nu\Sigma_f^1 & \cdots & \chi_p^{N_g}\nu\Sigma_f^{N_g} \end{pmatrix}$$

$$\Sigma_n = \begin{pmatrix} \left(\Sigma_{Tot}^0 - \Sigma_{sn}^{00}\right) & -\Sigma_{sn}^{01} & \cdots & -\Sigma_{sn}^{0N_g} \\ -\Sigma_{sn}^{10} & \left(\Sigma_{Tot}^1 - \Sigma_{sn}^{11}\right) & \cdots & -\Sigma_{sn}^{1N_g} \\ \vdots & \vdots & \ddots & \vdots \\ -\Sigma_{sn}^{N_g0} & -\Sigma_{sn}^{N_g1} & \cdots & \left(\Sigma_{Tot}^{N_g} - \Sigma_{sn}^{N_gN_g}\right) \end{pmatrix}$$

With these definitions, the SP$_{\mathrm{N}}$ equations 3.72 can be rearranged to form a more compact set of equations:

$$I_{1/v}\frac{\partial\vec{\Phi}_n}{\partial t}+\nabla\left[\frac{n}{2n+1}\Phi_{n-1}+\frac{n+1}{2n+1}\Phi_{n+1}\right]+\Sigma_n\vec{\Phi}_n=0 \quad for\ odd\ n \tag{3.73a}$$

$$I_{1/v}\frac{\partial\Phi_n}{\partial t}+\nabla\cdot\left[\frac{n}{2n+1}\vec{\Phi}_{n-1}+\frac{n+1}{2n+1}\vec{\Phi}_{n+1}\right]+\Sigma_n\Phi_n = \left[(1-\beta)F\Phi_0+X_dS_d\right]\delta_{n0} \qquad for\ even\ n \tag{3.73b}$$

Here we introduce two approximations that will be discussed later:

- The time derivative of the odd modes are set to zero, which means that the time variation of odd flux modes are much smaller than the spatial flux variations of even flux modes [53].
$$I_{1/v}\frac{\partial\Phi_n}{\partial t}=0 \quad for\ odd\ n \tag{3.74}$$
- The scattering is considered isotropic between energy groups, i.e. there is no group-to-group anisotropic scattering. This approximation is called the within-group approximation, which is a common assumption in multi-group diffusion theory as explained by Brantley and Larsen [54]. It can be expressed as:
$$\Sigma_{sn}^{gg'}=0 \quad g'\neq g \quad n>0 \tag{3.75}$$

Using the first assumption we can reduce in half the amount of equations to solve by replacing the odd modes in the even modes as shown in equation 3.76. It should be noted that this is particularly true for steady-state. The second assumption, eliminates all non-diagonal elements from Σ_n matrices of non-zero orders ($n>0$). In this work, it was applied to the non-zero odd orders. Then, the Σ_n matrices for odd n are easily inverted.

$$\vec{\Phi}_n=-\Sigma_n^{-1}\nabla\left[\frac{n}{2n+1}\Phi_{n-1}+\frac{n+1}{2n+1}\Phi_{n+1}\right] \quad for\ odd\ n \tag{3.76}$$

Then, replacing in 3.76 in 3.73b thus having the second-order form of the SP$_\text{N}$ equations:

$$I_{1/v}\frac{\partial\Phi_n}{\partial t} - \nabla\cdot\left[\frac{n}{2n+1}\Sigma_{n-1}^{-1}\ \nabla\left(\frac{n-1}{2n-1}\vec{\Phi}_{n-2}+\frac{n}{2n-1}\vec{\Phi}_n\right)+ \frac{n+1}{2n+1}\Sigma_{n+1}^{-1}\ \nabla\left(\frac{n+1}{2n+3}\vec{\Phi}_n+\frac{n+2}{2n+3}\vec{\Phi}_{n+2}\right)\right]+\Sigma_n\Phi_n = \left[(1-\beta)F\Phi_0+X_dS_d\right]\delta_{n0} \qquad for\ even\ n \tag{3.77}$$

For the current analysis we will be using up to the third mode of the method since this provides sufficient accuracy. The series of equations 3.77 are, therefore, truncated to N=3 (i.e. $\phi_i = 0$ for $i > 3$). Then, by working on the flux modes of order 0 and 2 (the odd modes are included in the second order equations) and by using a change of variables $U_1 = \Phi_0 + 2\Phi_2$, the SP$_3$ equations used in this analysis are obtained.

$$\boxed{\begin{aligned} I_{1/v}\frac{\partial U_1}{\partial t} &= \nabla\cdot\left(\frac{1}{3}\Sigma_1^{-1}\ \nabla U_1\right) - \Sigma_0\,(U_1 - 2\Phi_2) \\ &\quad + (1-\beta)\ F\ (U_1 - 2\Phi_2) + X_d S_d\ +\ 2\ I_{1/v}\frac{\partial \Phi_2}{\partial t} && (3.78a) \\ 3\ I_{1/v}\frac{\partial \Phi_2}{\partial t} &= \nabla\cdot\left(\frac{3}{7}\Sigma_3^{-1}\ \nabla\Phi_2\right)\ -\ \left(\frac{5}{3}\Sigma_2 + \frac{4}{3}\Sigma_0\right)\Phi_2 \\ &\quad -\frac{2}{3}(1-\beta)\ F\ (U_1 - 2\Phi_2)\ -\ \frac{2}{3}X_d S_d\ +\ \frac{2}{3}\Sigma_0 U_1\ +\ \frac{2}{3}\ I_{1/v}\frac{\partial U_1}{\partial t} && (3.78b) \end{aligned}}$$

This set of equations is similar, but not exactly the same, to those used by Fiorina [55]. It is easy to see that by setting $\Phi_2 = \vec{0}$ the SP$_1$ equation is obtained. Moreover, by defining the diffusion coefficient as $D = (1/3)\Sigma_1^{-1}$, the SP$_1$ equation becomes the standard diffusion equation. In this sense, the higher orders are a correction on the diffusion equations.

3.1.3.2 Boundary Conditions

The general boundary conditions for a given system can be expressed in terms of the angular flux coming into the system as follows:

$$\psi(\vec{r},\Omega) = \Psi^{-}(\Omega) \qquad for\ \Omega\cdot\hat{n} < 0,\ \vec{r}\in\partial\Gamma$$

where $\hat{n}$ is the outward normal of the boundary of the domain $\partial\Gamma$.

In our analysis, the boundary conditions are treated in an approximate way using the Marshak boundary conditions[8] from equation 3.79.

$$\begin{aligned} 2\pi\int_{\mu_{in}} P_i(\mu)\psi(\mu)\ d\mu &= \sum_{n=0}^{N}\frac{2n+1}{2}\phi_n(x)\int_{\mu_{in}} P_{2m-1}(\mu)P_n(\mu) \\ &= 2\pi\int_{\mu_{in}} P_{2m-1}(\mu)\Psi^{-1}(\mu) \qquad for\ x = 0, X\ and\ m = 1,2,\cdots(N+1)/2 \end{aligned} \tag{3.79}$$

[8] Bell and Glasstone [26] explain that the exact boundary conditions cannot be satisfied because these are imposed over half the angular range whereas the expansion coefficients apply over the whole range of $\mu\in[-1,1]$. There are two reasonable options for a free surface, either Marshak conditions, which have the virtue of including the diffusion theory zero incoming current case, and the Mark boundary conditions, which is equivalent to placing a pure absorber outside the domain.

For the case of vacuum, $\Psi^{-}(\mu) = 0$ making the RHS zero as well. In the one-dimensional problem the flux expansion is truncated at N (i.e., $\phi_{N+1} = 0$) but McClarren points out there are other possibles closures. This gives us $(N+1)/2$ equations at each side of the slab closing the system of $(N+1)$ equations 3.71.

Moreover, by using the ad-hoc replacement proposed in Gelbard's SP$_\text{N}$ formal deductions and considering that $\mu \rightarrow \hat{n} \cdot \vec{\Omega}$ we obtain the general form of the SP$_\text{N}$ method boundaries conditions:

$$\begin{aligned}\sum_{n\ even}^{N} \frac{2n+1}{4\pi}\phi_n(\vec{r}) \int_{\hat{n}\cdot\Omega>0} & P_{2m-1}(\hat{n}\cdot\Omega)P_n(\hat{n}\cdot\Omega)d^2\Omega \\ & + \sum_{n\ odd}^{N} \frac{2n+1}{4\pi}\ \vec{\phi}_n(\vec{r}) \int_{\hat{n}\cdot\Omega>0} P_{2m-1}(\hat{n}\cdot\Omega)P_n(\hat{n}\cdot\Omega)d^2\Omega \\ & \qquad = \int_{\hat{n}\cdot\Omega>0} P_{2m-1}(\hat{n}\cdot\Omega)\Psi^{-1}(\hat{n}\cdot\Omega) \\ & \qquad\qquad for\ \vec{r} \in \partial\Gamma\ and\ m = 1, 2, \cdots (N+1)/2 \end{aligned} \tag{3.80}$$

For the purpose of this work and without losing too much precision, we will once again truncate the solution to the third mode (N = 3) and apply the same assumptions as those used in equation 3.78. After some rearrangement, the Marshak vacuum boundary conditions for the SP$_3$ can be written as follows:

$$\boxed{\begin{aligned}\left(-\frac{1}{3}\Sigma_1^{-1}\nabla U_1\right)\cdot\hat{n} &= \frac{1}{2}U_1 \ - \ \frac{3}{8}\Phi_2 \qquad (3.81a)\\ \left(-\frac{3}{7}\Sigma_3^{-1}\nabla \Phi_2\right)\cdot\hat{n} &= -\frac{1}{8}U_1 \ + \ \frac{7}{8}\Phi_2 \qquad (3.81b)\end{aligned}}$$

Equations 3.78 and 3.81 together with the precursors concentration equations 3.6 (system containing a solid fissile) or 3.7 (system containing a liquid fissile) allow solving the transient neutronic problem. In this work, this system of equations was fully implemented and numerically solved in the code OpenFOAM. The details on the algorithms used for these equations are given in Chapter 4. In the next section, we will continue the discussion on the model development for the multi-physics tool by presenting the solid mechanics model.

3.2 Solid Mechanics

Different phenomena can be developed in solid components of nuclear systems undergoing a criticality accident. Some of these phenomena are relatively simple to solve such as the mechanical stresses resulting from thermal expansion. Other are more complicate to solve such as pressure or sound waves. In this section, we will describe the solid mechanic model used to calculate the displacements and the mechanical stresses resulting from temperature variations in a solid. This model is necessary to accurate estimate the solid fuel density variations and its associated effects on neutronics. This model will be key for solving some nuclear systems such as the one used in the Godiva experiments.

3.2.1 Basics notions

Mechanics is the study of the behaviour of bodies under action of forces and displacements. **Solid mechanics** is a branch of the **continuum mechanics** dealing with solid bodies [56]. In particular, the motion and deformation under action of forces, temperature changes, phase changes, and other external or internal agents.

3.2.1.1 Stress and Strain

Given a body in equilibrium under the action of external forces, the distribution of internal forces per unit area is called **stress** [57]. For these surface forces the notation is given by σ_i for **normal stress** and τ_{ij} for **shear stress**, where i is the normal coordinate axis to the surface where the stress is applied and j denotes to which axes the shear stress is parallel to. This can be observed in the infinitesimal cubic element from Figure 3.1.

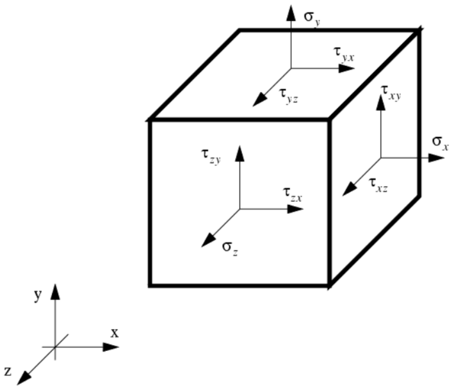


Figure 3.1 – Stress Element Diagram.

The stress tensor can be written as

$$\bar{\bar{\sigma}} = \begin{bmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_y & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_z \end{bmatrix} \tag{3.82}$$

If there are enough constraints to prevent the body from moving as a rigid body, any displacement of its particles is impossible unless there is a deformation of the same. This elongation per unit length is called **strain**. $\vec{D}$ (with components D_i) is the displacement field, that is, the elongation at any point in the material. If $\vec{D}^0$ is the displacement at a fixed point, then at any point P, it can be noted that for each component i the displacement is $D_i^P = D_i^0 + (\partial D_i/\partial i)di$ where each $\partial D_i/\partial i$ is the strain in the i direction. Given the the elongation on every direction the cube element from Figure 3.1 will experiment a tilt given by the **shear strain**. In conclusion, the relation between the strain and the displacement can be written as follows

$$\epsilon_i = \frac{\partial D_i}{\partial i} \qquad \gamma_{ij} = \frac{\partial D_i}{\partial j} + \frac{\partial D_j}{\partial i} \qquad for \quad i,j = x,y,z \quad \& \quad i \neq j \tag{3.83}$$

or in a more compact way

$$\bar{\bar{\epsilon}} = \frac{1}{2}\left[\nabla\vec{D} + \left(\nabla\vec{D}\right)^T\right] \tag{3.84}$$

where the strain tensor is

$$\bar{\bar{\epsilon}} = \begin{bmatrix} \epsilon_x & \gamma_{xy} & \gamma_{xz} \\ \gamma_{yx} & \epsilon_y & \gamma_{yz} \\ \gamma_{zx} & \gamma_{zy} & \epsilon_z \end{bmatrix} \tag{3.85}$$

3.2.1.2 Hooke's Law

Up to a certain limit the strain is proportional to the applied stress. This was determined experimentally by Robert Hooke and is now known as Hooke's law [57]. It can be expressed as

$$\epsilon_x = \frac{\sigma_x}{E} \tag{3.86}$$

where E is the **modulus of elasticity in tension** or **Young's modulus**. In addition, the lateral contractions of the cubic element are given by

$$\epsilon_y = -\nu\frac{\sigma_x}{E} \quad \epsilon_z = -\nu\frac{\sigma_x}{E} \tag{3.87}$$

where ν is **Poisson's ratio**. Within elastic limits Young's modulus and Poisson's ratio are the same in compression and in tension.

If the element of the body is submitted to a normal stress in all directions, the more general stress-strain relationship is given by

$$\epsilon_i = \frac{1}{E}\left[\sigma_i - \nu\left(\sigma_j + \sigma_k\right)\right] \quad for \quad i,j,k = x,y,z \quad i \neq j \neq k \tag{3.88}$$

A proportionality relationship can also be found between shear stress and shear strain given by

$$\gamma_{ij} = \frac{1}{G}\tau_{ij} \quad for \quad i,j = x,y,z \quad \& \quad i \neq j \tag{3.89}$$

where G is the **module of elasticity in shear** or the **modulus of rigidity** define as

$$G = \frac{E}{2(1+\nu)} \tag{3.90}$$

The **General Hooke's Law** can be written as

$$\bar{\bar{\sigma}} = 2\mu\bar{\bar{\epsilon}} + \lambda tr\left(\bar{\bar{\epsilon}}\right)\bar{\bar{I}} \tag{3.91}$$

or, alternatively,

$$\bar{\bar{\epsilon}} = \frac{1}{2\mu}\bar{\bar{\sigma}} - \frac{\lambda}{2\mu\left(3\lambda+2\mu\right)}tr\left(\bar{\bar{\sigma}}\right)\bar{\bar{I}} \tag{3.92}$$

where $\bar{\bar{I}}$ is the identity matrix and λ and μ are Lamé's coefficients defined as

$$\mu = \frac{E}{2\left(1+\nu\right)} \tag{3.93}$$

$$\lambda = \begin{cases} \frac{\nu E}{(1+\nu)(1-\nu)} & for\ plane\ stress \\ \frac{\nu E}{(1+\nu)(1-2\nu)} & for\ plane\ strain\ and\ 3D \end{cases} \tag{3.94}$$

3.2.1.3 Thermal Stress

When considering a thermal stress, the components of the strain tensor can be considered as the sum of the contribution from the stress field and the contribution from the temperature field. The latter is given by

$$\bar{\bar{\epsilon}}^{T} = \alpha \left(T - T_0\right) \bar{\bar{I}} \tag{3.95}$$

where T_0 is a reference temperature and α is the **thermal expansion coefficient**. The shear strain terms are not affected by the temperature because free thermal expansion does not produce angular distortion in an *isotropic* material [57]. The linear anisotropic case is treated in [58].

Summing equation 3.95 into the General Hooke's Law 3.92 we obtain the **Duhamel-Neumann** relation 3.96.

$$\bar{\bar{\epsilon}} = \frac{1}{2\mu}\bar{\bar{\sigma}} - \frac{\lambda}{2\mu\left(3\lambda + 2\mu\right)} tr\left(\bar{\bar{\sigma}}\right)\bar{\bar{I}} + \alpha\left(T - T_0\right)\bar{\bar{I}} \tag{3.96}$$

or, alternatively,

$$\bar{\bar{\sigma}} = 2\mu\bar{\bar{\epsilon}} + \lambda tr\left(\bar{\bar{\epsilon}}\right)\bar{\bar{I}} - \left(3\lambda + 2\mu\right)\alpha\left(T - T_0\right)\bar{\bar{I}} \tag{3.97}$$

3.2.2 Linear Elasticity

If a solid bar is subjected to traction and a stress-strain diagram is plotted the general behaviour from Figure 3.2 is observed. From O to A the stress and strain are proportional. After this point Hooke's law is no longer valid, the line becomes a curve and the strain rate quickly accelerates. Up to B (elastic limit) any deformation of the body disappears when the stress is no longer applied, i.e., the state of the system returns to O, this is, the body is **perfectly elastic**. From point B a sudden enlargement without appreciable increase of the stress is observed. This is called **yield** and there is a permanent deformation of the material, this is, if the stress is no longer applied only a part of the deformation is recovered given by the elastic deformation and a permanent strain remains. Then, the necessary stress keeps growing until point C where it reaches the peak known as **ultimate tensile strength**. After the strain increases with lower charge until **fracture** point D[59].

Linear elasticity is the most simple model for solid mechanics. It corresponds to

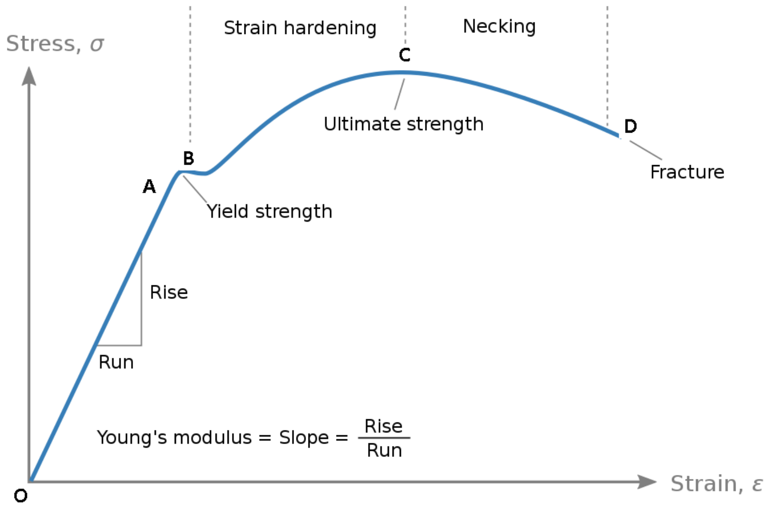


Figure 3.2 – Stress Strain Diagram.

the first part of the curve in Figure 3.2 where the bodies undergoing the action of the external forces are perfectly elastic. Only two physical parameters are needed in this model: the Young's Modulus (E) and the Poisson's ratio (ν), which can be easily obtained experimentally.

This method is valid for small deformations when it exists a linear stress-strain relationship. In addition, there is only one configuration and only one system of coordinates. Phenomena where the configuration changes cannot be accounted for in the **linear elasticity** model.

3.2.2.1 Equation of motion

For an isotropic homogeneous solid body undergoing small strains the equation for momentum balance of **Cauchy's first law of motion** [60][61][62] is

$$\frac{\partial^2(\rho\vec{D})}{\partial t^2} - \nabla \cdot \bar{\bar{\sigma}} = \rho\vec{f} \tag{3.98}$$

where $\vec{f}$ are the **body forces**.

If we combine the momentum equation 3.98 with the Duhamel-Neumann relation from equation 3.97 and the strain-displacement relation 3.84, the displacement equation for **linear thermoelasticity** is obtained:

$$\boxed{\frac{\partial^2\left(\rho\vec{D}\right)}{\partial t^2} = \nabla\left[\mu\nabla\vec{D} + \mu\left(\nabla\vec{D}\right) + \lambda\ I\ tr\left(\nabla\vec{D}\right)\right] - \nabla\left(\frac{E}{1-2\nu}\alpha T\right) + \rho\vec{f}} \tag{3.99}$$

Note that this equation requires knowledge on the temperature field. Equation 3.99 is therefore coupled to the solid phase energy conservation equation discussed later. Furthermore, the resolution of equation 3.99 requires setting-up adequate boundary conditions as discussed in the next section.

3.2.2.2 Boundary Conditions

Boundary Condition for solid mechanics can be of several types [60]. Among the most common ones it can be cited:

- Fixed displacement
- Planes of symmetry
- Fixed pressure
- Fixed traction
- Free surface

The first condition consists in specifying $\vec{D}$ at the boundary, i.e., a Dirichlet Condition. Second condition, is equivalent to mirroring the system, this is a Neumann condition. The last three conditions in the list must satisfy the following expression:

$$\vec{\vec{\sigma}} \cdot \vec{n} = \vec{t} - p\vec{n} \tag{3.100}$$

where $\vec{t}$ is the applied traction, p the external pressure and $\vec{n}$ is the normal vector pointing out of the boundary surface.

3.2.2.3 Energy conservation for a solid

We assume that energy transfer in the solid phase is caused only by thermal conduction. Therefore, the solid temperature field can be estimated by a simple energy balance:

$$\boxed{\rho c_p \frac{\partial}{\partial t}(T) = \nabla \cdot [k \nabla T] + \dot{q}_V} \tag{3.101}$$

where ρ, c_p and k are the solid's density, specific heat capacity at constant pressure and the thermal conductivity respectively. The term $\dot{q}_V$ is a power source.

Note that the source term in the energy balance equation is, in our case, related to the nuclear fission or the decay heat. This equation is therefore coupled to the neutronics equations discussed earlier in this chapter.

Finally, by setting adequate boundary conditions for our problem and given the temperature field (e.g. calculated from the energy equation), the linear thermoelasticity equation 3.99 can be finally solved. In our approach, this set of equations was numerically resolved by using existing libraries in OpenFOAM. An example of application is presented in Chapter 5 for a Godiva experiment. Note that while the modeling of the solid phase requires a high level of precision to adequately capture the key phenomena, most nuclear systems have also fluid phase that is often used as a coolant to extract heat. Therefore, in the next section, it will be discussed the modeling of the fluid equations or as it is often called the thermal-hydraulics aspects of the model.

3.3 Thermal-hydraulics

Most of the nuclear systems studied in criticality accidents (see Chapter 2) have one or several fluid components such as coolants, moderators or even liquid fuels. As for the solid phase, several complex phenomena will take place in the fluid phase. Some of these phenomena are related to the flow convective transport (e.g. heat convection, delayed neutron precursors transport, etc.) while others will be caused by the criticality accident itself (e.g. boiling, radiolysis, etc.). Many of these phenomena will have an important impact on the accident evolution because of the strong coupling with the neutronic and thermal-mechanical phenomena. In order to model them in the multi-physics model, fluid mechanics equation have to be implemented.

Fluid mechanics theory is the branch of continuum mechanics studying fluids, that is, liquids and gases. Opposite to solids, which resist by deforming under applied shear or tangential stress, liquids are put in motion by the applied stress [63][64].

3.3.1 Eulerian vs Lagrangian

Traditionally, there are two approaches to analyzing problems in mechanics:

- **Eulerian approach**
- **Lagrangian approach**

The first follows a domain as material flows through it, i.e., the Eulerian approach focuses on a specific location (Control Volume) as time passes. On the other hand, the Lagrangian approach follows a material parcel as it moves through space and time.

3.3.1.1 Material Derivative

The Lagrangian and Eulerian approaches are related by the **Material or Lagrangian derivative**. Applying the chain rule to the total rate of change of a field variable $f(t, x(t))$ with Eulerian specification in a specific flow parcel we obtain

$$\frac{Df}{Dt} = \frac{df}{dt} = \frac{\partial f}{\partial t}\frac{dt}{dt} + \frac{\partial f}{\partial x}\frac{dx}{dt} + \frac{\partial f}{\partial y}\frac{dy}{dt} + \frac{\partial f}{\partial z}\frac{dz}{dt} \tag{3.102}$$

$$= \frac{\partial f}{\partial t} + u_x\frac{\partial f}{\partial x} + u_y\frac{\partial f}{\partial y} + u_z\frac{\partial f}{\partial z} \tag{3.103}$$

$$= \frac{\partial f}{\partial t} + (\vec{u} \cdot \nabla) f \tag{3.104}$$

where $\vec{u}$ is the **velocity vector**, $\partial f/\partial t$ is the **Eulerian derivative** or **local rate of change** and $(\vec{u} \cdot \nabla) f$ is the **convective rate of change**.

The advantage of the Eulerian approach relies in that it focus its attention to a fixed Control Volume (CV) which is the domain of interest. Whereas the Lagrangian approach cannot be fixed to the domain of interest and the fluid parcels go to wherever the flow takes them, not necessarily inside the domain of interest. In addition, the Eulerian approach abandons the tedious task of tracking each individual particle. However, it introduces non-linearities through the convective rate of change term, which represents the product of the unknown velocity field ($\vec{u}$) and the gradient of an unknown variable field (∇f).

3.3.2 Reynolds Transport Theorem

The **principle of conservation** states that for an isolated system a particular property is conserved over a local region. This applies to moving Material Volumes (MVs) of fluids, not for fixed points or Control Volumes. To adapt the conservation principle to an Eulerian approach the **Reynolds Transport Theorem (RTT)** is needed [65][63].

If B is any property of the fluid (mass, momentum, energy) and $b = dB/dm$ the intensive value of B in any small element of the fluid then the Reynolds Transport Theorem can be expressed as

$$\left(\frac{dB}{dt}\right)_{MV} = \frac{d}{dt}\left(\int_{V(t)} b\rho dV\right) + \int_{S(t)} b\rho\vec{u}_r \cdot \vec{n} dS \tag{3.105}$$

where ρ is the **density** of the fluid, $\vec{n}$ is the vector **normal** to the Control Volume boundary, $V(t)$ is the deformable Control Volume, $S(t)$ is the deformable Control Surface (boundary of the CV), $\vec{u}_r$ is the relative velocity $\vec{u} - \vec{u}_s$ with $\vec{u}_s$ being the velocity of the deforming control surface.

If the volume is fixed, then $\vec{u}_s = \vec{0}$ and geometry is independent of time. In addition, by using the *Leibniz integral rule* and the *divergence theorem*, equation 3.105 becomes

$$\left(\frac{dB}{dt}\right)_{MV} = \int_V \left[\frac{D}{Dt}(\rho b) + \rho b \nabla \cdot \vec{u}\right] dV \tag{3.106}$$

We will use this equation for fixed volumes to obtain the mass, momentum and energy conservation equations of the fluid phase in the next sections.

3.3.3 Flow Conservation Equations

In this section the **Navier-Stokes equations** and the energy conservation equations are presented. The Navier-Stokes equations are a couple of highly nonlinear second order partial differential equations in four independent variables $(\vec{r}, t)$. They are used to describe a broad spectrum of phenomena involving **Newtonian** fluids which will be discussed in Section 3.3.3.2. Originally, the name Navier-Stokes was used to denominate the linear momentum conservation equation but nowadays it is also use including the conservation of mass.

3.3.3.1 Conservation of Mass

Assuming that the fluid does not experience phase change (e.g. solidification or boiling) and since the mass change due to nuclear reactions is negligible, then we can state that the mass m of the fluid system is conserved. In the Lagrangian description, if $B = m$, it can be stated as

$$\left(\frac{dm}{dt}\right)_{MV} = 0 \tag{3.107}$$

Now, using the Reynolds Transport Theorem in equation 3.106 given that $b = 1$ we obtain the equivalent expression in an Eulerian coordinate system.

$$\int_V \left[\frac{D\rho}{Dt} + \rho \nabla \cdot \vec{u}\right] = 0 \tag{3.108}$$

For this to be true for any Control Volume V, the integrand should be zero, thus obtaining the differential form of the mass conservation or **continuity equation**.

$$\boxed{\frac{D\rho}{Dt} + \rho\nabla \cdot \vec{u} = 0} \tag{3.109}$$

For fluid systems where the temperature gradients are relative small (no significant density variation) and having small Mach numbers (smaller than 0.3), this equation can be further simplified by considering the flow as incompressible, i.e.$D\rho/Dt = 0$ [63]. Using the incompressible flow approximation, the continuity equation can be reduced to:

$$\boxed{\nabla \cdot \vec{u} = \vec{0}} \tag{3.110}$$

The continuity equation for incompressible flow means that in this particular case the flow is divergence free or that the difference between outward and inward flow in the system is zero. The incompressible flow approximation can be used in many types of liquids and gases provided that the flow Mach number is small and the temperature variations do not cause significant thermal expansion across the system. If these hypotheses are not met then the flow should be considered as compressible and the more general continuity equation 3.108 has to be used. In our applications, the incompressible flow approximation provides sufficient accuracy.

3.3.3.2 Conservation of Linear Momentum

The conservation of the linear momentum is obtained by applying the **Newton's Second Law** with body ($\vec{f_b}$) or surface ($\vec{f_s}$) forces acting on the fluid Material Volume (MV) ($\vec{f} = \vec{f_b} + \vec{f_s}$). Once again, given the quantity $B = m\vec{u}$ the conservation of momentum in Lagrangian coordinates is:

$$\left(\frac{d(m\vec{u})}{dt}\right)_{MV} = \left(\int_V \vec{f} dV\right)_{MV} \tag{3.111}$$

Since the volume integral on the Right Hand Side (RHS) is performed over the volume occupied instantaneously by the moving fluid we can lose the MV subscript. In additon, $dB/dm = \vec{u}$ and applying the Reynolds Transport Theorem we obtain

$$\int_V \left[\frac{D}{Dt}(\rho\vec{u}) + (\rho\vec{u}\nabla \cdot \vec{u}) - \vec{f}\right] = 0 \tag{3.112}$$

Once again, the integrand should be zero to fulfill the previous relation. Then,

$$\frac{D}{Dt}\left(\rho\vec{u}\right)+\left(\rho\vec{u}\nabla\cdot\vec{u}\right)=\vec{f} \tag{3.113}$$

In addition, by regrouping and taking into account the continuity equation 3.109 the differential conservation equation for the flow momentum is obtained:

$$\rho\left[\frac{\partial\vec{u}}{\partial t}+\left(\vec{u}\cdot\nabla\right)\vec{u}\right]=\vec{f} \tag{3.114}$$

3.3.3.2.1 Surface Forces

Surface forces acting on the boundary surfaces of the Control Volume element are due to pressure and viscous stresses. The stress tensor can be described in the same way as in Figure 3.1 and defined in equation 3.82. The most important contribution to the normal stress is the pressure. Normally the stress tensor is split into two terms giving

$$\begin{aligned}\bar{\bar{\sigma}} &= -\begin{bmatrix} p & 0 & 0 \\ 0 & p & 0 \\ 0 & 0 & p \end{bmatrix}+\begin{bmatrix} \sigma_x+p & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_y+p & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_z+p \end{bmatrix} \\ &= -p\bar{\bar{I}}+\bar{\bar{\tau}} \end{aligned} \tag{3.115}$$

where p is the **thermodynamic pressure** and $\bar{\bar{\tau}}$ is the **deviatoric** or **viscous stress tensor**. The thermodynamic pressure p is equal to the hydrostatic pressure when the fluid is at rest.

Now, the integral of the surface forces on the volume applying divergences theorem is

$$\int_S \bar{\bar{\sigma}}\cdot\vec{n}dS=\int_V \nabla\cdot\bar{\bar{\sigma}}dV \tag{3.116}$$

Then,

$$\vec{f_s}=\nabla\cdot\bar{\bar{\sigma}}=-\nabla p+\nabla\cdot\bar{\bar{\tau}} \tag{3.117}$$

Newtonian Fluids

Most fluids in engineering applications can be considered as Newtonian fluids [63]. In our applications, all the fluids of interest can be considered this way. Newtonian fluids are defined as fluids where the stress tensor is a linear function of the strain rate. This property together with two general assumptions: (1) the fluid is a continuous isotropic medium and (2) the viscous stress tensor reduces to zero when the strain rate is zero, can be used to found the following expression for the viscous stress [66]:

$$\bar{\bar{\tau}} = \mu \left\{ \nabla \vec{u} + (\nabla \vec{u})^T \right\} + \lambda \left(\nabla \cdot \vec{u} \right) \bar{\bar{I}} \tag{3.118}$$

where μ is the **molecular viscosity coefficient**, λ is the **bulk viscosity coefficient** and the operator {} denotes the **dyadic product**.

Replacing the surface stress force in equation 3.114 we obtain the momentum equation for compressible and incompressible Newtonian fluids:

$$\boxed{\rho \frac{D\vec{u}}{Dt} = -\nabla p + \nabla \cdot \left\{ \mu \left[\nabla \vec{u} + (\nabla \vec{u})^T \right] \right\} + \nabla \left(\lambda \nabla \cdot \vec{u} \right) + \vec{f_b}} \tag{3.119}$$

where f_b are body forces. The previous equation can be further simplified by using the **Stoke's hypothesis**: ($\lambda = -(2/3)\mu$) (verified by most fluids), assuming that the flow is incompressible ($\nabla \cdot \vec{u} = \vec{0}$) and that the fluid viscosity μ is constant to:

$$\boxed{\rho \frac{D\vec{u}}{Dt} = -\nabla p + \mu \nabla^2 \vec{u} + f_b} \tag{3.120}$$

This is known as the Navier-Stokes equation for an incompressible flow that will serve as a basis for the models discussed in the rest of this chapter. It is worth to note that these equations (one for each of the three spatial directions) are highly non-linear due to the convective term in the Material derivative and thus only a few analytical solutions are known. Indeed the existence of a solution to these equations is not guaranteed and usually very complex numerical integration methods have to be used. As it is discussed later, depending on the competition between inertial (convective term) and viscous forces (viscous tensor) the flow can develop a very unstable behaviour that is called **turbulence**. Turbulent flows will require the introduction of additional approximations in order to decrease the complexity of the problem and the computational cost of the numerical resolution. Finally, it is important to remark that the above equations require both initial and boundary conditions to be completely solved. In our applications, the boundary conditions will often include: non-slip at the wall, free surface and inlet/outlet flow. More details on how these boundary conditions can be setup can be found in [64].

3.3.3.2.2 Body Forces

Body forces are expressed as force per unit volume. There are many examples among them but the most common ones are: **gravitational forces** given by the weight

$$\vec{f_b} = \rho \vec{g} \tag{3.121}$$

where $\vec{g}$ is the **gravitational field**; and systems in rotation, i.e, **Coriolis and Centrifugal forces**

$$\vec{f_b} = -2\rho \left[\vec{\omega} \times \vec{u}\right] - \rho \left[\vec{\omega} \times \left[\vec{\omega} \times \vec{r}\right]\right] \tag{3.122}$$

where $\vec{\omega}$ is the **angular velocity** of the rotating reference frame.

Boussinesq Approximation

In many applications the effects of temperature variations on the flow mass conservation (i.e. fluid density and velocity fields) can be neglected and thus the flow approximated as incompressible. These temperature effects are nevertheless more important in the momentum equation, more specifically in the gravity forces term. In many cases studied in criticality accidents a significant error would occur if they are not considered. For example, natural convection in a tank filled with a fissile liquid cannot be studied with a simple incompressible flow model without any corrective term for these effects.

The Boussinesq approximation [67] provides a corrective term in the momentum equations that allows using the incompressible flow equations but still taking into account these effects (and therefore studying phenomena such as natural convection). This is a better option than using the more complicate to numerically integrate compressible flow Navier-Stokes equations. Assuming that $\Delta T = T - T_0$ across the flow is small, with T_0 a temperature reference, the Boussinesq approximation estimates the density change caused by the temperature difference with a first degree Taylor expansion, as follows:

$$\rho(T) = \rho(T_0) + \frac{d\rho}{dT}(T_0)\left(T - T_0\right) \tag{3.123}$$

defining the coefficient of volume expansion β as

$$\beta = -\frac{1}{\rho}\left(\frac{\partial \rho}{\partial T}\right)_p \tag{3.124}$$

then in the Boussinesq approximation the fluid density is written as follows:

$$\rho = \rho_0 \left[1 - \beta \left(T - T_0\right)\right] \tag{3.125}$$

By using this equation to calculate the fluid density only for the gravity force term of the Navier-Stokes equations 3.120, the momentum conservation for an incompressible flow using the **Boussinesq approximation** can be written as follows:

$$\boxed{\rho \frac{D\vec{u}}{Dt} = -\nabla \left(p\right) + \mu \nabla^2 \vec{u} + \rho \vec{g} \left[1 - \beta \left(T - T_0\right)\right]} \tag{3.126}$$

In the Boussinesq approximation, the flow continuity equation remains unchanged (i.e. we still used equation 3.110). The energy equation will also be approximated. In particular, flow compressible effects are not taken into account.

3.3.3.3 Conservation of Energy

The first law of thermodynamics states that energy can be neither created nor destroyed during a process, it can only change from one form to another. Therefore, the total energy for an isolated system is constant. For a Material Volume this can be expressed as

$$\left(\frac{dE}{dt}\right)_{MV} = \dot{Q} - \dot{W} \tag{3.127}$$

where $\dot{Q}$ is the **rate of heat addition**, $\dot{W}$ is the **rate of work done by the flow** and E is the **total energy** defined as the sum of the internal and kinetic energy ($E = m(\hat{u} + (1/2)\vec{u} \cdot \vec{u})$).

To apply the Reynolds Transport Theorem we define $B = E$ and $b = dE/dm = \hat{u} + (1/2)\vec{u} \cdot \vec{u} = e$. Once again contributions of the work and heat can be split between the surface and body contributions.

We define $\dot{q}_V$ and $\dot{q}_S$ as the **rate of heat source or sink** per unit volume and per unit surface respectively, and the **rate of work** done by body and surface forces as

$$\dot{W}_b = -\int_V \left(\vec{f}_b \cdot \vec{u}\right) dV$$
$$\dot{W}_s = -\int_S \left(\vec{f}_s \cdot \vec{u}\right) dS$$

With these definitions the differential conservation of energy equation in terms of the specific total energy e is obtained.

$$\boxed{\frac{\partial}{\partial t}(\rho e) + \nabla \cdot [\rho \vec{u} e] = -\nabla \cdot \dot{q}_s - \nabla \cdot [p\vec{u}] + \nabla \cdot \left[\overline{\overline{\tau}} \cdot \vec{u}\right] + \vec{f_b} \cdot \vec{u} + \dot{q}_V} \tag{3.128}$$

where ρ is the fluid density, $\vec{u}$ its velocity, p is the thermodynamic pressure, $\overline{\overline{\tau}}$ is the viscous stress tensor, $\vec{f_b}$ are the body forces and $\dot{q}_V$ and $\dot{q}_S$ are the rate of heat source or sink per unit volume and per unit surface respectively.

For a Newtonian fluid, assuming the specific enthalpy $\hat{h}$ as a function of p and T given by the following equilibrium thermodynamic relations

$$d\hat{h} = c_p dT + \left[\hat{V} - T\left(\frac{\partial \hat{V}}{\partial T}\right)_p\right] dp \tag{3.129}$$

where $\hat{V}$ is the **specific volume** and c_p the **specific heat capacity** at constant pressure; and neglecting the thermal radiation in the fluid, only heat transfer by conduction can occur at the surface. By using the Fourier's law for describing conduction transfer through the surfaces of the system as

$$\dot{q}_S = -k\nabla T \tag{3.130}$$

then the energy equation in terms of the temperature can be expressed as

$$\boxed{\begin{aligned}\frac{\partial}{\partial t}(\rho c_p T) + \nabla \cdot [\rho c_p \vec{u} T] = \nabla \cdot [k\nabla T] + \rho T \frac{Dc_p}{Dt} \\ - \left(\frac{\partial(log\ \rho)}{\partial(log\ T)}\right)_p \frac{Dp}{Dt} + \lambda\Psi + \mu\Phi + \dot{q}_V\end{aligned}} \tag{3.131}$$

where k is the **thermal conductivity**, and Ψ and the **dissipation term** Φ are defined as

$$\Psi = \left(\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z}\right)^2 \tag{3.132}$$

$$\begin{aligned}\Phi = 2\left[\left(\frac{\partial u_x}{\partial x}\right)^2 + \left(\frac{\partial u_y}{\partial y}\right)^2 + \left(\frac{\partial u_z}{\partial z}\right)^2\right] \\ + \left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x}\right)^2 + \left(\frac{\partial u_x}{\partial z} + \frac{\partial u_z}{\partial x}\right)^2 + \left(\frac{\partial u_y}{\partial z} + \frac{\partial u_z}{\partial y}\right)^2\end{aligned} \tag{3.133}$$

In general, the dissipation term Φ is negligible except for supersonic flows (high Mach number). In addition, if the fluid is incompressible, $\Psi = 0$ and the constant density implies

$(\partial (log\ \rho) / \partial (log\ T)) = 0$. Then, the energy equation for an incompressible Newtonian fluid is

$$\boxed{\rho c_p \frac{\partial}{\partial t} (T) + \rho c_p \nabla \cdot [\vec{u}T] = \nabla \cdot [k \nabla T] + \dot{q}_V} \tag{3.134}$$

For a solid, the velocity is zero and the convective term disappears. Thus we obtain the equation presented in the solid mechanics section:

$$\boxed{\rho c_p \frac{\partial}{\partial t} (T) = \nabla \cdot [k \nabla T] + \dot{q}_V} \tag{3.135}$$

Note that the energy equation is coupled to the neutronics model through the source term (fission power and/or decay heat) and the Navier-Stokes equation through the velocity field.

3.3.4 Turbulence

As discussed earlier, the non-linearity introduced by the convective term of the Navier-Stokes equations cause the formation and growth of flow instabilities or fluctuations in the solution at sufficiently high velocities. The onset of these instabilities depends on the ratio between the inertial forces (convective term) which amplify the instabilities and the viscous forces (viscous stress tensor) which tend to damp them. The flow regime predicted by the Navier-Stokes equations accurately describes the real flow that, depending on the average velocity, can show the two distinctive flow regimes: laminar flow and turbulent flow. Laminar flows occurs at low velocity and are characterized by a stratified structure where the fluid particles follow smooth current lines. On the other hand, for high enough flow velocities a fully turbulent flow will be developed. Turbulence is a complex phenomenon characterized by chaotic, diffusive, highly mixing, time dependent and three-dimensional vorticity behaviour in a broad range of time and length scales. Between the laminar and turbulent flow regimes a transition (unstable) regime will exist.

As stated before, the transition from laminar to turbulent regime can be physically explained by the competition between two opposite flow forces: the inertia force $\rho (\vec{u} \cdot \nabla) \vec{u}]$ and the viscosity force $\mu \nabla^2 \vec{u}$. The dimensionless **Reynolds number** allows calculating an approximate ratio between these two forces and thus to predict the flow regime. This number is defined as follows:

$$Re = \frac{Inertial\ Forces}{Viscosity\ Forces} = \frac{\rho\left(\vec{u}\cdot\nabla\right)\vec{u}}{\mu\nabla^2\vec{u}} \sim \frac{\rho U^2/L}{\mu U/L^2} = \frac{UL}{\nu} \tag{3.136}$$

where U is a **characteristic velocity**, L a **characteristic distance** and $\nu = \mu/\rho$ is the **kinematic viscosity** or **momentum diffusivity**.

It has been experimentally observed that the onset of turbulence occurs when the flow exceeds a certain **Reynolds number**. Transition to turbulence flow begins as instabilities in the laminar flow, which are then amplified by the highly non-linear inertial terms. The theory of *energy cascade* developed by Kolmogorov [68] [69] says that turbulence is composed of vortex or **eddies** of different sizes. The larger eddies break up and transfer their energy to the smaller ones. This happens at every smaller size up to a certain size where the viscous dissipation is sufficiently effective. At the other end, the largest size is proportional to the size of the geometry involved. The transition from laminar to turbulent flow occurs for **Reynolds number** between approximately 1000 and 10000. This is an approximate range that depends on the geometry being considered.

3.3.5 Reynolds Averaging

Turbulent flow regimes exist in most engineering applications and thus any multiphysics model has to be able to handle turbulence. However, handling the chaotic behaviour of the turbulence flow in time and space is practically impossible from a mathematical and numerical point of view. Several numerical methods exist to treat turbulence, we will start with the simplest one developed by Osborne Reynolds in 1895 [70] and which consists in rewriting the Navier-Stokes equations in terms of mean or time averaged turbulent variables. Indeed, Reynolds observed that in time the variables could be formulated as the sum of the mean and a fluctuating term as

$$\begin{aligned}
\vec{u} &= \overline{\vec{u}} + \vec{u}' \\
p &= \overline{p} + p' \\
T &= \overline{T} + T'
\end{aligned} \tag{3.137}$$

where the mean of a variable ϕ is indicated as $\bar{\phi}$ and the fluctuating term as ϕ'. Any two variables ϕ and ψ must follow some rules for deriving the Reynolds average equations.

$$\begin{aligned}
\overline{\phi'} &= 0 \\
\overline{\phi} &= \overline{\overline{\phi}} \\
\overline{\nabla\phi} &= \nabla\overline{\phi} \\
\overline{\phi + \psi} &= \overline{\phi} + \overline{\psi} \\
\overline{\overline{\phi} + \psi} &= \overline{\phi\psi} \\
\overline{\overline{\phi}\psi'} &= 0 \\
\overline{\phi\psi} &= \overline{\phi\psi} + \overline{\phi'\psi'}
\end{aligned} \tag{3.138}$$

If we replace the definitions in 3.137 into the incompressible Navier-Stokes equations 3.110, 3.120 and 3.134 and use the properties cited in 3.138 the average equations are obtained:

$$\boxed{\begin{aligned}
&\nabla \cdot \overline{\vec{u}} = 0 && (3.139a) \\
&\rho\frac{D\overline{\vec{u}}}{Dt} = -\nabla\overline{p} + \left[\nabla \cdot \left(\overline{\tau} - \rho\overline{\vec{u}'\vec{u}'}\right)\right] + \overline{f}_b && (3.139b) \\
&\rho c_p\frac{D\overline{T}}{Dt} = \nabla \cdot \left[k\nabla\overline{T} - \rho c_p\overline{\vec{u}'T'}\right] + \overline{Q^T} && (3.139c)
\end{aligned}}$$

We can identify two terms involving the fluctuations, one in the momentum equation 3.139b and another in the energy equation 3.139c. The first one is known as the **Reynolds stress tensor** $\overline{\overline{\tau}}^R$ which can be written as

$$\overline{\overline{\tau}}^R = -\rho\overline{\vec{u}'\vec{u}'} = -\rho\begin{bmatrix} \overline{u'_x u'_x} & \overline{u'_x u'_y} & \overline{u'_x u'_z} \\ \overline{u'_y u'_x} & \overline{u'_y u'_y} & \overline{u'_y u'_z} \\ \overline{u'_z u'_x} & \overline{u'_z u'_y} & \overline{u'_z u'_z} \end{bmatrix} \tag{3.140}$$

The second one is the **turbulent heat flux**

$$\dot{q}^R = -\rho c_p\begin{bmatrix} \overline{u'_x T'} \\ \overline{u'_y T'} \\ \overline{u'_z T'} \end{bmatrix} \tag{3.141}$$

This set of equations is not closed, i.e., there are more unknowns than equations. The strategy consists in eliminating the fluctuating terms ϕ' and solve for the average ones $\overline{\phi}$.

To fulfill this purpose a relation of the new unknowns in term of the averaged variables is needed. This is possible via the **Boussinesq Hypothesis** which is analogous to the Newtonian flows giving a linear relation given by

$$\overline{\overline{\tau}}^R = \mu_t \left\{ \nabla \overline{\vec{u}} + \left(\nabla \overline{\vec{u}} \right)^T \right\} - \frac{2}{3} \left[\rho k + \mu_t \left(\nabla \cdot \overline{\vec{u}} \right) \right] \overline{\overline{I}} \tag{3.142}$$

where $k = (1/2)\overline{\vec{u}' \cdot \vec{u}'}$ is the **turbulent kinetic energy** and μ_t the **turbulent eddy viscosity**, which is now flow dependent.

Usually the pressure is redefined as a turbulent pressure to include the term $-(2/3)\rho k \overline{\overline{I}}$. Then, $p \leftarrow p + (2/3)\rho k$. Now, the problem of computing the Reynolds stress tensor is reduced to calculating μ_t.

Similarly for the turbulent thermal fluxes an analogy with Fourier's law is given by

$$\dot{q}^R = -\rho c_p \overline{\vec{u}' T'} = k_t \nabla \overline{T} \tag{3.143}$$

where k_t is the **turbulent thermal diffusivity**.

To solve for μ_t and k_t, there are multiple turbulence models, for example, $k-\epsilon$ method, $k-\omega$ method, among others. As explained before, the purpose of this section is not to have an exhaustive treatment of turbulence but an idea of the complexity of thermal-hydraulics. For further information, the book of Moukalled [64] is recommended.

3.3.6 Numerical Resolution Strategies

For the Navier-Stokes equations the analytical solutions are, in general, not possible to obtained, except for very rare cases. For this reason, numerically resolution of the equations is required. The numerical science on the resolution of the flow equations is known as **Computational Fluid Dynamics**. Depending on the desired level of precision different approaches exist. In decreasing accuracy they are described as follows

3.3.6.1 Direct Numerical Simulation

The **Direct Numerical Simulation (DNS)** aims to simulate all eddies up to the minimum size possible. In DNS the original Navier-Stokes equations (without averaging) are used. Therefore, the time resolution needed to solve every vorticity size is tiny. This imposes a restriction on the mesh size given by the Courant number below 1, mathematically expressed as

$$Co = \frac{|\vec{u}|\,\Delta t}{\Delta x} < 1 \tag{3.144}$$

As a consequence, this approach is computationally expensive given the detailed mesh of the fluid and the small characteristic time of the system. At present, only relatively small systems with simple geometries can actually be solved. This approach is therefore not adapted at this day for integration in a multi-physics model for our applications.

3.3.6.2 Large Eddy Simulations

A less demanding method is the **Large Eddy Simulation (LES)** method. In LES the large scale of the vorticity is directly solved (as in DNS) and the smaller ones are modeled using sub-grid scale models. It works as a low-pass filter on the spatial domain filtering small eddies up to a set value. This method could be of interest in situations where the details of the flow vortexes is needed for the system studied. This is not the case for the transients or systems studied in this work where the neutron mean free path was usually much larger than the potential vortexes and thus a time-averaged flow description was sufficient.

3.3.6.3 Reynolds-averaged Navier-Stokes

The **Reynolds-averaged Navier–Stokes equations** method consists in solving for the time-averaged equations from Section 3.3.5 with the corresponding turbulence models ($k-\epsilon$, $k-\omega$, etc) and wall functions ([64]). In this model the conditions on the mesh size and time step are relaxed but the problem is solved for the averaged variables, therefore, the turbulence is not visible. Since a RANS approach provides the required precision for our analysis and it has a lower computational cost that the DNS and LES methods, it was adopted in the current work. As it will be discussed in Chapter 4, the existing RANS algorithms for incompressible flows of OpenFOAM were adapted for our studies. A short description of the RANS turbulent model used in the analysis of a tutorial test case for a Space Nuclear Reactor Design [71] is given in the Appendix A.

3.3.7 Porous Medium

A CFD code allows numerically resolving the RANS equations to obtain very high detail data on the local flow velocities and temperatures. Nevertheless, such a model for a Fuel Assembly from the Spent Fuel Pool from Chapter 2 would require a very fine mesh to take into account the various fuel assemblies' components: grids, nozzles, fuel rods, etc. Given the size of the Spent Fuel Pool, this will quickly lead to a prohibitive mesh size and would preclude any practical multi-physics coupling. On the other hand, the level of details provided by such approach is not necessary for the criticality accident unless one is interested on calculating local parameters. For these reasons, a porous medium modeling approach was taken for system comporting a solid fuel such as Fuel Assemblies. As shown in Figure 3.3, this approach consists in replacing the heterogeneous original system by a homogeneous one having equivalent macroscopic properties (the porous equivalent medium). In our analysis, a multi-phase multi-component porous medium was developed and implemented.

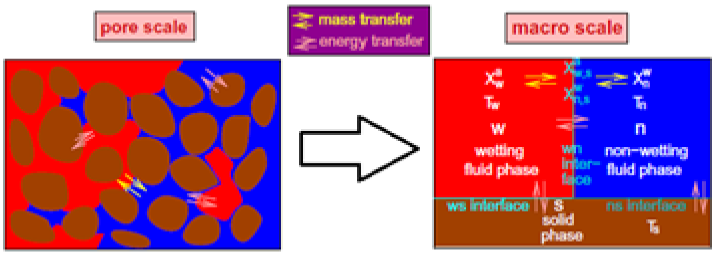


FIGURE 3.3 – Pore scale and macro scale averaging procedure and main phenomena (from [72]).

While the mesh discretization required for numerically solving the porous medium equations is relatively modest, all the complexities of the model are now shifted to the determination of the macroscopic properties. To obtain these properties, an averaging procedure [73][74][75] over a **Representative Elementary Volume (REV)** is usually done in such a way that new macroscale properties can be obtained. These properties depend on parameters such as: (a) the **porosity**, that is the ratio between the void space in the porous medium and the total volume of the element ($\phi = V_{void}/V_{total}$), (b) the **saturation**, which is the ratio between the volume of the **phase** α and the **void space in between pores** ($S_\alpha = V_\alpha/V_{void}$) and (c) the **permeability** K, which measures the flow conductance of the matrix. The main idea behind this strategy is to obtain a model that can take into account as much as possible of the microscale phenomena existing in the multi-phase multi-component flow while being calculated in a REV big enough to diminish the computational cost, i.e. moving from microscopic pore scale to a macroscopic mixed model. The equations of the equivalent porous medium representing the behavior of the SFP were developed based on the works of [75], [76], [77], [72] and [78]. This porous medium model considers non-equilibrium conditions between three-phases (wetting or liquid (denote as w), non-wetting or gas (denote as n) and solid (denote as s)) and has two components (Air and water denoted as H_2O). The solid phase is considered as being non-deformable and can only exchange energy with the surroundings. The model can be enhanced to include more phenomena but for the present study, the level of details was considered as sufficient for the studied cases. The next subsection provides an overview of the various conservation equations.

3.3.7.1 Conservation of Mass

The conservation of mass in a porous medium is required only for the fluid phase containing two phases: **wetting** w and **non-wetting** n. When considering a two-phase two-component system, the conservation of mass reduces to four equations (3.145) [76]. Each equation states the mass conservation of species k in the α phase, where ρ_α is the density of the α phase, $x_\alpha^k, x_{(\alpha,equil)}^k$ are the mass fraction and the equilibrium mass fraction of the species k in the α phase respectively and L is a characteristic length, normally the pore size:

$$\frac{\partial}{\partial t}\left(\phi S_w \rho_w x_w^{H_2O}\right)+\nabla\cdot\left(\rho_w x_w^{H_2O} V_w + j_w^{H_2O}\right) = q_w^{H_2O} - \rho_n Sh(Re,Sc) a_{wn} \frac{D_{n,pm}^{H_2O}}{L}\left(x_{n,equil}^{H_2O} - x_n^{H_2O}\right) \tag{3.145a}$$

$$\frac{\partial}{\partial t}\left(\phi S_w \rho_w x_w^{Air}\right)+\nabla\cdot\left(\rho_w x_w^{Air} V_w + j_w^{Air}\right) = q_w^{Air} + \rho_w Sh(Re,Sc) a_{wn} \frac{D_{w,pm}^{Air}}{L}\left(x_{w,equil}^{Air} - x_w^{Air}\right) \tag{3.145b}$$

$$\frac{\partial}{\partial t}\left(\phi S_n \rho_n x_n^{H_2O}\right)+\nabla\cdot\left(\rho_n x_n^{H_2O} V_n + j_n^{H_2O}\right) = q_n^{H_2O} + \rho_n Sh(Re,Sc) a_{wn} \frac{D_{n,pm}^{H_2O}}{L}\left(x_{n,equil}^{H_2O} - x_n^{H_2O}\right) \tag{3.145c}$$

$$\frac{\partial}{\partial t}\left(\phi S_n \rho_n x_n^{Air}\right)+\nabla\cdot\left(\rho_n x_n^{Air} V_n + j_n^{Air}\right) = q_n^{Air} - \rho_w Sh(Re,Sc) a_{wn} \frac{D_{w,pm}^{Air}}{L}\left(x_{w,equil}^{Air} - x_w^{Air}\right) \tag{3.145d}$$

The first term of the equations represents the mass rate change inside the REV. The second term on the LHS is a stream term representing the advection due to the velocity field. A macroscale diffusive term given by **Fick's Law** is modelled ($j_\alpha^k = -D_{\alpha,pm}^k \nabla(\rho_\alpha^k x_\alpha^k)$) using a macroscale diffusion coefficient calculated as $D_{\alpha,pm}^k = \tau\phi S_\alpha D^k$, where the tortuosity τ can be obtained from $\tau = (\phi S_\alpha)^{\frac{2}{3}}\phi^2$ as discussed in [79]. Finally, an external mass source term q_α^k is added in the equations (to model for example the change of phase). The last term on the RHS represents the migration effect due to a local chemical non-equilibrium. Normally, the mass fraction of the components is usually determined via equilibrium relations such as Henry's and Raoult's law [80]. This point will be further discussed in the Section 3.3.7.5 as well as the determination of the average interfacial surface a_{wn}. Dimensionless quantities such as the Sherwood number Sh (function of Re and Sc), the Reynolds number Re and the Schmidt number Sc are all defined in Table 3.1.

Table 3.1 – Dimensionless quantities and their definitions.

Dimensionless Quantity	**Value**	**Interpretation**
Reynolds (Re)	$\frac{\rho V L}{\mu}$	$\frac{InertiaForces}{ViscosityForces}$
Prandtl (Pr)	$\frac{cp\mu}{\lambda}$	$\frac{MomentumDiffusion}{ThermalDiffusion}$
Schmidt (Sc)	$\frac{\mu}{\rho D}$	$\frac{MomentumDiffusion}{MassDiffusion}$
Nusselt (Nu)	$\frac{hL}{k}$	$\frac{ConvectiveHeatTransfer}{ConductiveHeatTransfer}$
Sherwood (Sh)	$\frac{\beta L}{D}$	$\frac{ConvectiveMassTransfer}{DiffusiveMassTransfer}$

3.3.7.2 Conservation of energy

As seen in the works of [76] and [78], the porous medium phases can have different temperatures, i.e. thermal non-equilibrium conditions have been considered in this model. A standard formulation of the energy conservation for each of the three phases is provided by equations 3.146:

$$\begin{aligned}\frac{\partial}{\partial t}\left(\phi\rho_w S_w u_w\right) + \nabla\cdot\left(\rho_w V_w h_w + \sum_k j_w^k h_w^k\right) - \nabla\cdot\left(\phi S_w \lambda_w \nabla T_w\right) = \sum_k h_w^k q_w^k + q_w^{energy} \\ + \sum_k \dot{\eta}^k_{wn\to w} h_n^k + Nu\left(Re, Pr\right) a_{wn}\frac{\overline{\lambda}_{wn}}{L}\left(T_n - T_w\right) + Nu\left(Re, Pr\right) a_{ws}\frac{\overline{\lambda}_{ws}}{L}\left(T_s - T_w\right)\end{aligned} \tag{3.146a}$$

$$\begin{aligned}\frac{\partial}{\partial t}\left(\phi\rho_n S_n u_n\right) + \nabla\cdot\left(\rho_n V_n h_n + \sum_k j_n^k h_n^k\right) - \nabla\cdot\left(\phi S_n \lambda_n \nabla T_n\right) = \sum_k h_n^k q_n^k + q_n^{energy} \\ + \sum_k \dot{\eta}^k_{wn\to n} h_w^k - Nu\left(Re, Pr\right) a_{wn}\frac{\overline{\lambda}_{wn}}{L}\left(T_n - T_w\right) + Nu\left(Re, Pr\right) a_{ns}\frac{\overline{\lambda}_{ns}}{L}\left(T_s - T_n\right)\end{aligned} \tag{3.146b}$$

$$\frac{\partial}{\partial t}\left((1-\phi)\rho_s c T_s\right) - \nabla\cdot\left((1-\phi)\lambda_s \nabla T_s\right) = q_s^{energy} - Nu\left(Re, Pr\right) a_{ws} \frac{\overline{\lambda}_{ws}}{L}\left(T_s - T_w\right)$$
$$-Nu\left(Re, Pr\right) a_{ns} \frac{\overline{\lambda}_{ns}}{L}\left(T_s - T_n\right) \qquad (3.146c)$$

where u_α, h_α, λ_α, T_α are, respectively, the **internal energy**, **enthalpy**, **conductivity** and **temperature** of the α **phase**, h_α^k is the **enthalpy of the species** k **in the** α **phase** and c is the **heat capacity** of the solid phase. The first term on the LHS of the equation represents the rate of change of the energy in the REV. The second one represents the advection of enthalpy carried by the same phase and the third is the diffusion of each component of the phase. The last term on the LHS is the heat conduction inside the phase. The first term on the RHS is the enthalpy that enters in the system due to the external mass source and the second term is an external energy source for the phase q_α^{energy}. From equations 3.145 we can calculate the mass flux for each component from one phase to the other as $\dot{\eta}_{wn\to\alpha}^k = \rho_\alpha Sh(Re, Sc) a_{wn}(D_{\alpha,pm}^k/L)(x_{\alpha,equil}^k - x_\alpha^k)$. In this model the interface wn is considered two-dimensional, therefore one phase mass loss must end up in the other, that is why we can find the latter quantity in two equations at a time with opposite sign. This latter definition is used in the third term on the RHS to take into account the enthalpy exchange due to change of phase of a component. The last two terms represent the thermal exchange between the phases and as expected these terms appear in two of the equations with opposite signs. The discussion on the determination of the **average interfacial surface** a_{ns}, a_{ws} is left to Section 3.3.7.5. Dimensionless quantities such as the Nusselt (Nu), Reynolds (Re) and Prandtl (Pr) numbers are described in Table 3.1.

3.3.7.3 Darcy's Equations

To calculate the velocity fields a relative simple version of the momentum conservation equations in the porous medium was used which is the extended **Darcy's Law** was used:

$$V_w = -\frac{k_{r,w}K}{\mu_w}\left(\nabla p_w - \rho_w \vec{g}\right) \qquad (3.147a)$$

$$V_n = -\frac{k_{r,n}K}{\mu_n}\left(\nabla p_n - \rho_n \vec{g}\right) \qquad (3.147b)$$

where $k_{r,\alpha}$ is the **relative permeability**, which is the correction in the permeability due to the presence of another phase, μ_α is the **dynamic viscosity of the** α **phase**, $\vec{g}$ is the gravity and p_α is the **pressure of the** α **phase**. Even though this is an empirical relationship, this law can be demonstrated by applying different averaging procedures in the REV to the Navier-Stokes equations [81][82] and when the viscous forces are the dominating forces in the porous medium. When the inertial forces and turbulence become more significant a corrective term has to be added to these equations known as Forchheimer term. Nevertheless for the cases studied in this work, this addition was not necessary.

3.3.7.4 Closure equation

So far, we have 9 equations and 13 unknowns (2 saturations, 4 species concentrations, 2 velocities, 3 temperatures and 2 pressures fields). The closure equations 3.148-3.151 arise naturally to close the system.

Equation 3.151 relates the pressure in each phase by a quantity known as **capillary pressure**: the differential pressure between two immiscible fluid phases occupying the same pores caused by interfacial tension between the two phases that must be overcome to initiate flow. Determination of this parameter is discussed in the next section.

$$S_w + S_n = 1 \tag{3.148}$$

$$x_w^{H_2O} + x_w^{Air} = 1 \tag{3.149}$$

$$x_n^{H_2O} + x_n^{Air} = 1 \tag{3.150}$$

$$p_n - p_w = p_c \tag{3.151}$$

3.3.7.5 Other relationships and considerations

By now, we have a close system of 13 equations and 13 unknowns, but it is still very important to define how the thermodynamics parameters (density, viscosity, etc.) and volume averaged interfacial areas (a_{wn}, a_{ws}, a_{ns}) are determined and how to model the capillary pressure and relative permeability. For the latter, the most common models are the Brooks and Corey Model [83] and the Van Genuchten Model [84]. Both of these thermodynamics models associate the relative permeabilities and the capillary pressure with the saturation of the phase. In this work, the Brooks and Corey Model was used following equations 3.152-3.154:

$$k_{r,n} = (1 - S_w)^m \tag{3.152}$$

$$k_{r,w} = S_w^m \tag{3.153}$$

$$p_c = p_{c,0} S_w^{\frac{-1}{\lambda_p}} \tag{3.154}$$

where m is an empirical coefficient (in nuclear $m = 3$ has given good results [85]), $p_{c,0}$ is the **entry capillary pressure** and λ_p is the **pore size distribution index**.

The interfacial areas are not straightforward to model and no general accepted function has been identified. However, many ways of estimating them exist, such as developing a pore-network model and estimating them as a function of saturation [76] or by lattice Boltzmann calculations [78] for example. Another important point on the discussion is how to compute the thermodynamic parameters. A precise approach would compute the chemical potential of each component in each phase and then derive the non-equilibrium properties [80][86] in the REV. The equilibrium concentrations and properties could be obtained using the same approach. An approximation would be needed to estimate the mixture properties as a function of the components equilibrium values in each phase given the pressure and the temperature fields. For the sake of simplicity and without losing any generality, in this work the equilibrium molar concentration $X^k_{\alpha,equil}$ are calculated using the Henry's law [80] for the air (wetting) and considering that the water pressure in the gas phase is equal to the saturation vapor pressure, thus obtaining respectively the equations 3.155 and 3.156 [77]:

$$x^{Air}_{w,equil} = H^{Air}(T_w) p^{Air}_n \tag{3.155}$$

$$x^{H_2O}_{n,equil} = \frac{p^{H_2O}_{sat}}{p_n} \tag{3.156}$$

The intensive properties are thus considered as being function of the concentrations, pressures and temperatures. In addition, given the relatively low power densities existing in the fuel assemblies of the SFP, thermal equilibrium between the phases is assumed, i.e. $T_s = T_w = T_n = T$. This approximation will be valid only if recriticality is not reached in the SFP.

Finally the whole set of equations presented in Section 3.3.7 were implemented and solved in OpenFOAM to predict the velocity and temperature fields of a multiple phases porous medium having two temperatures. This model is used to study a Spent Fuel Pool as discussed in Chapter 5.

3.4 Conclusions

This chapter presented the theoretical basis used in developing these models required to enhance the multi-physics tool for the study of criticality accidents. The first part covered the development and numerical implementation of the transient neutronic models which include the Monte Carlo Quasi-Static Method and the deterministic Simplified $\mathrm{P_N}$ ($\mathrm{SP_1}$ and $\mathrm{SP_3}$). The second part presented the solid mechanics model used for Godiva experiment, in particular the adaptation of a linear elasticity model already existing in OpenFOAM. The last part was devoted to the development of a porous medium model for two phase flow, with phase change and two temperatures. This model, implemented in this thesis work, enhancing an already existing model in OpenFOAM (Horgue et al. [19]), was used to study a potential Spent Fuel Pool recriticality accident. This last part also discusses the RANS models used for a system containing a liquid fissile such as a liquid fuel.

Chapter 4

Codes and Implementation

This chapter provides a description of the key algorithms used in the multi-physics tool for the study of criticality accidents. The chapter begins with an overview of the multi-physics tool, its main components and the codes used for developing it. Then, the key algorithms of the developed tool to perform neutronics, solid mechanics and thermal-hydraulics calculations are reviewed.

4.1 The Multi-Physics Code

The multi-physics tool to study criticality accidents has been developed from a coupling between OpenFOAM and Serpent 2 codes. As shown in the Figure 4.1, the tool contains three main modules: neutronics, thermal-hydraulics and thermal-mechanics. The neutronics module calculates the reactor neutron flux, the reaction rates and the power distribution. Two main neutronics models are available: the stochastic Monte Carlo Quasi-Static Method and the deterministic Simplified $\mathrm{P_N}$ Method ($\mathrm{SP_1}/\mathrm{SP_3}$). The volumetric power distribution obtained from this module is used by the thermal-hydraulics module to calculate the temperature, density and velocity fields of the fluid and solid phases existing in the nuclear system. The thermal-hydraulics module also calculates the delayed neutron precursors concentration if the nuclear system contain a fissile solid or liquid such as a liquid fuel. Finally, the thermal-mechanics module can be used for the determination of the displacements, strain and stress fields in the solid phase. As shown in the figure, the three main modules composing the multi-physics tool allow taking into account some of the key coupled phenomena that may exist in criticality accidents: Doppler effect, density effects, precursors convection, melting and solidification, among the ones cited in Chapter 2.

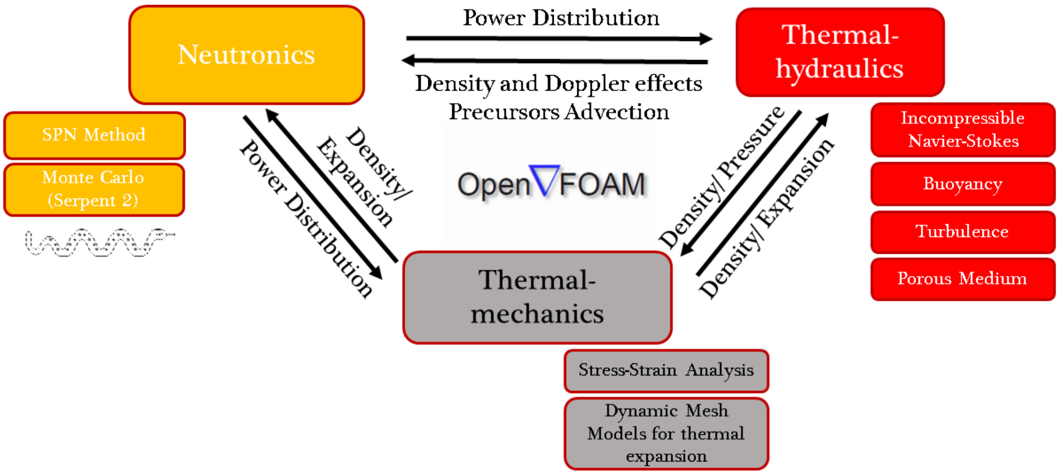


Figure 4.1 – Multi-Physics Coupling Main Modules.

Most of the methods and algorithms used by the tool were developed with the aim of obtaining a good compromise between flexibility and precision while minimizing the source code modifications required. Serpent 2 code is used by the tool for the neutron flux shape calculations needed by the Monte Carlo Quasi-Static Method and for the neutron cross section libraries generation for the $\mathrm{SP_N}$ Method while all the other models (thermal-hydraulics and thermal-mechanics models) are numerically solved using OpenFOAM libraries. In addition, the deterministic approach $\mathrm{SP_N}$ was fully implemented in OpenFOAM.

4.2 Codes

4.2.1 Serpent

Serpent 2 [5][6] is a Monte Carlo particle transport code under development at the VTT Technical Research Centre of Finland. It has been programmed in **C language**. The general working scheme is very similar to other Monte Carlo codes like MCNP [30]. Serpent 2 includes all the features explained in Section 3.1.1 and more.

Originally, it was developed for three-dimensional continuous-energy Monte Carlo reactor physics calculations including, for example, group cross sections generation, burn-up, point kinetics parameters, among others. Nowadays, it has been expanded beyond traditional application into multi-physics simulations, photon transport, sensitivity calculations, among others. It also supports parallel calculations with OMP and MPI. More details on **Serpent** can be found in [5][6].

4.2.2 OpenFOAM

Open-source Field Operation and Manipulation (OpenFOAM) was created in 1989 by Henry Weller [2][3]. It is an open-source C++ Library for Computational Fluid Dynamics distributed by **The OpenFOAM Foundation** [87]. It is based on the Finite Volume Method (FVM) [64] for CFD. Even though its main purpose is the fluid mechanics, it is capable of solving continuum mechanics equations in general. Many numerical solvers and schemes, and pre-/post-processing utilities are available. It is capable of solving incompressible, compressible and multi-phase flows with conjugate heat transfer and chemical reactions, among others.

The main advantage of OpenFOAM is the friendly syntax for the Partial Differential Equation, which, in conjunction with the accessibility to the source makes coupling with other models (such as neutronics) easier. In addition, the use of unstructured meshes is advantageous for modeling complex geometries. For these reasons, it was chosen for this work.

Among the drawbacks of this code, one can mention that there are multiple enterprises/-communities developing different versions of OpenFOAM, making it confusing when modifying existing solvers or trying to find the right one to modify afterwards. For the present work we use OpenFOAM Foundation [87] Version 6[1]. Also, late versions of OpenFOAM have steeply increased in complexity with a Programmer's guide not detailed enough, thus making development of new applications or functionalities slower.

4.3 General Algorithm

This section presents the key algorithms of the multi-physics tool. The section starts with the general algorithm used for the coupling of the neutronics, thermal-mechanics and thermal-hydraulics modules. Then, algorithms used by the different models implemented in these three modules are presented.

As explained in Chapter 1, the main motivation for the work was to develop a multi-physics tool capable of modeling complex geometries and a wide range of time scales

[1] At the moment of writing this work OpenFOAM Version 8 is the latest release

encountered in the full spectrum of criticality accidents. For this reason, ad-hoc coding or modifications for particular problems, that is, hard coding, were limited to a minimum. This strategy means that most algorithms were developed trying to be as general and flexible as possible while minimizing the modifications required in the original source codes Serpent 2 and OpenFOAM. This strategy has the advantage of minimizing the tool maintenance while it has the drawback of penalizing the computational cost of the algorithms.

Figure 4.2 shows the general algorithm of the multi-physics tool that was used for modeling the different criticality systems presented in Chapter 2. As we can see, given the nature of OpenFOAM, the solver is separated into several blocks covering different aspects of the physics presented in Chapter 3.

OpenFOAM uses an **segregated** strategy. If we have a system of equations

$$u = f_u(u, v) \tag{4.1a}$$

$$v = f_v(u, v) \tag{4.1b}$$

the corresponding algorithm would involve solving for the system with a **fixed point iteration** scheme. Accordingly, we would solve first for u with v constant from equation 4.1a and then for v with an updated constant u. Then, subsequently solving and updating u and v at each iteration until convergence. In addition, each equation can be splitted in explicit (using variables from the previous iteration/time step) and implicit (using variables in current iteration/time step) terms, for example

$$u = f_u(u, v) = f_u^{implicit}(u, v) + f_u^{explicit}(u, v) \tag{4.2}$$

and then solved for u with u and v fixed in the explicit term and only v fixed in the implicit term. In general that is done when non-linear terms are present or when the discretized matrix has a high condition number.

The previous strategy will become clearer in the following sections where the procedure within each block is explained. Firstly, we will start with the neutronics algorithms.

4.3.1 Neutronics Block

The numerical implementation of the neutronics models presented in Section 3.1 is discussed in this section. Firstly, Section 4.3.1.1 describes the algorithms used for the coupling

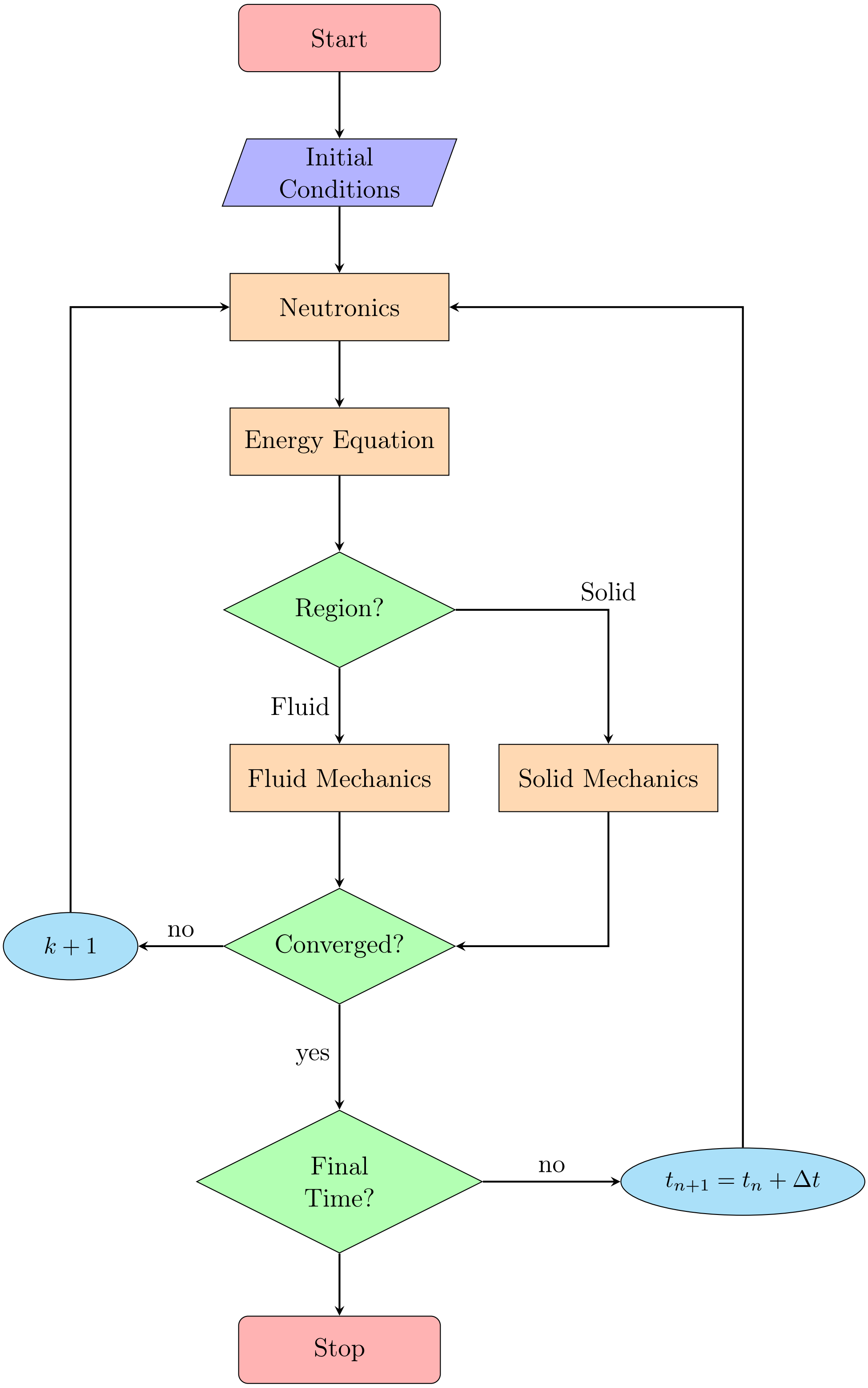


Figure 4.2 – General Algorithm Structure.

between the codes OpenFOAM and Serpent 2. This coupling is key for the multi-physics tool since it is used for performing steady-state simulations and also during the flux shape calculations needed by the Monte Carlo Quasi-Static Method introduced in the previous chapter. Various specific aspects of the coupling between OpenFOAM and Serpent 2 are covered in Section 4.3.1.1: the meshes used by the two codes, the transfer of the variable fields such as the temperature and the density between the codes and the treatment given to the delayed neutron precursors concentration. The implementation of the algorithm for the Monte Carlo Quasi-Static Method is described in Section 4.3.1.2. Finally, the algorithms related to the Simplified $\mathrm{P_N}$ are presented in Section 4.3.1.3. Algorithms used by the Dynamic Monte Carlo (transient Monte Carlo) calculations presented in the next chapter will not be discussed in this manuscript since they were not developed in the framework of this PhD but a description can be found in references [35][7].

The neutronics module of the multi-physics tool can be simplified as a black box shown in the Figure 4.3 with the temperature field T, density field ρ and the velocity field $\vec{u}$ as inputs and the volumetric power field $P = \epsilon\Sigma_f\phi$ as an output, where ϵ is the energy released at fission and deposited in the system. The neutronics model chosen by the *user* (Monte Carlo Quasi-Static Method or Simplified $\mathrm{P_N}$) will be run a finite number of times according to the set value of k_{neut}^{MAX} to obtain the desired convergence.

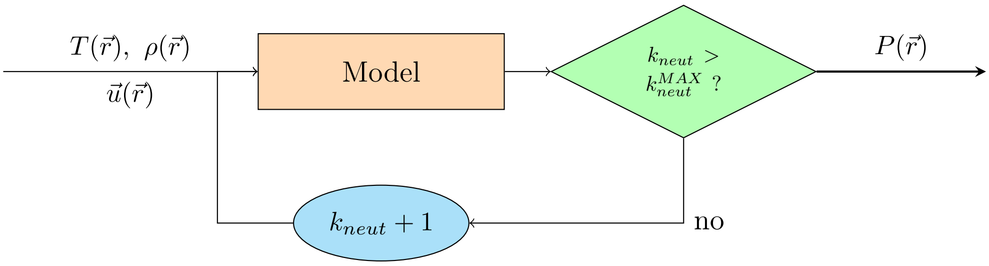

$T(\vec{r})$, $\rho(\vec{r})$
$\vec{u}(\vec{r})$
Model
$k_{neut} >$ k_{neut}^{MAX} ?
$P(\vec{r})$
$k_{neut} + 1$
no

FIGURE 4.3 – Neutronics Block.

4.3.1.1 OpenFOAM - Serpent 2 main coupling

As can be seen in Figure 4.4, Serpent 2 code was integrated as an internal function of OpenFOAM. Since C++ language used by OpenFOAM is compatible with C employed by Serpent 2, it was possible to compile Serpent as an internal function of OpenFOAM. This strategy is not mandatory but allows decreasing the computational cost of the iterations between the two codes during the multi-physics calculation. Indeed, an external coupling is possible at the expense of a slower coupling scheme: details on Serpent 2 multi-physics interface can be found on its website [88]. Moreover, for the case of a nuclear system containing a liquid fissile, the information exchanges required for tracking the delayed neutron precursors positions would be more complicate when handled externally, the main reason being that it would require the use of the Monte Carlo external source method having to deal with the control population techniques and time binning as explained in Section 3.1.1.4.2 and, not the criticality mode used in this work (k-code). An additional advantage of the internal coupling strategy is that both codes have access to all memory allocated runtime variables. Important information not normally stored by Serpent can then be treated and stored in OpenFOAM mesh-based variables for future use. Also, the variables calculated by OpenFOAM can be transmitted directly into the Monte Carlo tracking routine.

As discussed in the previous chapter, the criticality k-eigenvalue method of Serpent 2 was adopted in our work during the Monte Carlo simulations, more specifically, for the flux shape calculations. The criticality k-eigenvalue algorithm of Serpent has then been integrated in OpenFOAM as shown in Figure 4.4. This Serpent 2 algorithm consists in a typical tracking routine sorting the free path length, the direction and the energy of the neutrons the way it has been explained in Sections 3.1.1.2 and Section 3.1.1.4. The information exchange between Serpent and the OpenFOAM environment are also shown in the figure. We will focus next on different specific aspects of the coupling, starting with the input data.

4.3.1.1.1 Input

Even though, Serpent source code was compiled within OpenFOAM environment in a single solver, it was decided that the input management will remain segregated between the two codes as shown in Figure 4.4. Geometry, materials and neutronics boundary conditions must then be defined in the Monte Carlo input. That is straightforward using the current capabilities developed by Aufiero et al. [89][90] for unstructured mesh-based geometries for multi-physics calculations. As the system properties change during a transient calculation, by taking advantage of the already implemented features, the non-uniform temperature and density distributions calculated by OpenFOAM are available to Serpent from the shared memory as it will be explained in this section.

It is important to note that each time the Serpent tracking routine is called, the whole

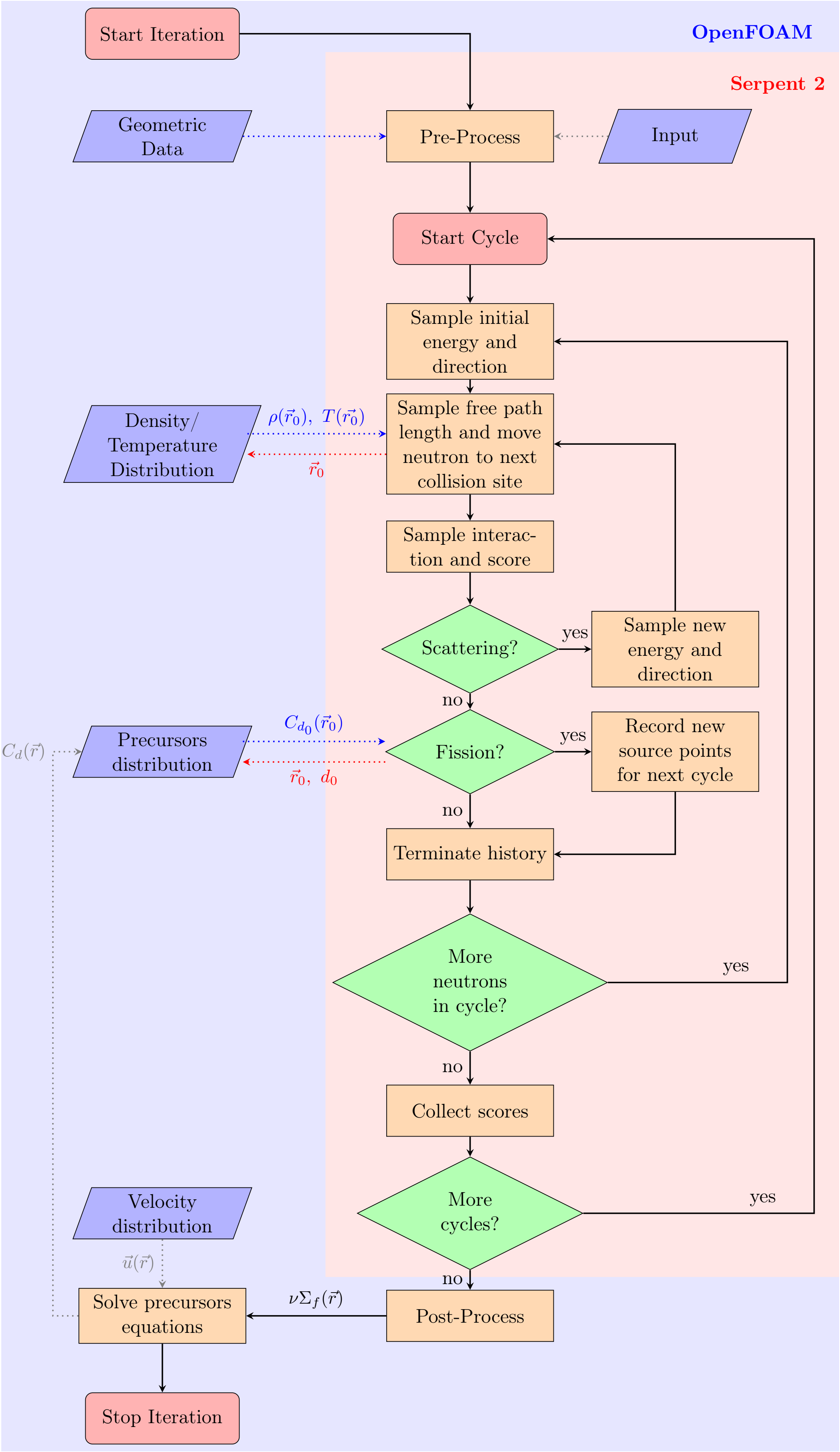


Figure 4.4 – k-eigenvalue criticality source mode (adapted from Leppänen [29]) coupled to precursors solver.

input data file has to be reprocessed and internally updated in the OpenFOAM values/-fields which is a non-negligible time consuming operation. This implies that when using the internal coupling all the neutronics input data will be processed every time Serpent is used to perform a neutron flux shape calculation and, thus, impacting in the tool performance if too many flux shape calculations are needed. In the case of an external coupling scheme, Serpent can go to "sleep" after the shape calculation has been completed and while OpenFOAM solvers are being executed. This way, when Serpent is called for a new shape calculation all necessary variables are still in the running memory and only temperature and densities fields have to be loaded. This will avoid the input data processing time delay on the contrary to the internal coupling scheme. If necessary, in the internal coupling scheme, the time penalisation arising from the input processing could be avoided by using asynchronous routines existing in Serpent. A final remark on this topic is that this time penalty will not be avoided in transients involving quick changes on the system geometry (e.g. Godiva experiment), since both internal or external couplings require loading and processing the mesh and all the associated properties at each calculation of the neutron shape function.

4.3.1.1.2 Mesh variables

As discussed earlier, Serpent 2 and OpenFOAM codes may use in the calculation different meshes. Therefore, variables calculated in these codes may belong to different meshes and, thus, require a conversion processing through mapping routines. For example, one of the key variables of the coupling between the neutronics and the thermal-hydraulics and thermal-mechanics modules is the fission volumetric power $P(\vec{r})$ (or power density). The volumetric fission power distribution is determined by the neutronics module. To calculate the volumetric power distribution a mesh-based tally is used in Serpent. The tally in each cell of the Serpent mesh is calculated according to equation 3.25 using the collision estimator technique described in Section 3.1.1.3. This tally value is then divided by the volume of the cell to obtain the volumetric value. Accordingly, the volumetric power of cell j is calculated setting-up the response function f of the tally (from equation3.25) to $f(r, E) = \epsilon\nu\Sigma_f(\vec{r}, E)$ thus obtaining:

$$P(\vec{r_j}) = \frac{1}{V_j}\sum_i \omega_i \frac{\epsilon\nu\Sigma_f(\vec{r_i}, E_i))}{\Sigma(\vec{r_i}, E_i)} \qquad\qquad \forall\ \vec{r_i} \in V_j \tag{4.3}$$

It should be noticed that, since the Monte Carlo simulation can be thought as an experiment, the value of the cell tallies will increase with the number of the simulated neutron histories. For this reason, the volumetric power, as any other stored variable, has to be normalized. For steady-state calculations, the volumetric tally is usually normalized with the total power. However, by default neutron transport simulations in Serpent 2 are normalized to unit total loss rate (i.e. neutrons absorptions and leaks).

Once the volumetric power has been determined in Serpent, it will have to be saved in the corresponding cell of the unstructured mesh of OpenFOAM. Besides the volumetric power, other parameters of interest will be required for the Monte Carlo Quasi-Static Method. These variables will be discussed later in Section 4.3.1.2.

4.3.1.1.3 Non-uniform density and temperature distributions

The ability to correctly model non-uniform density and temperature distributions in materials is a key feature for the multi-physics coupling since, during a criticality accident, most real systems can develop a large density and temperature gradients. As we pointed out in Section 3.1.1.2, the Monte Carlo method for particle transport is derived considering an homogeneous medium or, at least, that the domain of interest can be divided in smaller homogeneous regions, then the cross sections are piece-wise uniform functions. Nevertheless, when the cross sections are considered as continuous functions (e.g. due to temperature or density gradients) this theory will no longer be valid since the function to which the inversion method is applied (equation 3.16) has no general solution and the Probability Density Function and Cumulative Distribution Function become unknown distributions [91].

A simple correction to take into account continuous density changes inside a same-material region can be nevertheless straightforward implemented. For the case of the cross sections' temperature correction on-the-fly, the modifications are less trivial because the Doppler effect have to be addressed explicitly taking into account the thermal motion of the target nuclei. Other strategies like interpolating tabulating data exist but it requires narrow spacing between the temperature points to reduce interpolation errors to an acceptable value [92]. Fortunately, Serpent 2 already provides some algorithms that allow dealing with these particular issues and only some minor internal modifications are required in the source code to take advantage of the already implemented routines. This section provides a brief overview to the Serpent 2 algorithms that allow dealing with the effects of density and temperature gradients. A more complete description for non-uniform density can be found in [91] and concerning non-uniform temperature in [92][93][94].

Concerning the density gradient effects, Serpent 2 modeling strategy relies on the rejection sampling[2] on the particle path lengths and on the use of the collision estimator. For a non-uniform density region, the neutron path length sampling is the same as the one of an homogeneous material using the inversion method but the maximum possible cross section at the collision point (Σ_{max}) is used instead. Then, a modification factor or probability is calculated as follows:

$$g = \frac{\Sigma_t(\vec{r})}{\Sigma_{max}} = \frac{\rho(\vec{r})}{\rho_0} = \rho_k \tag{4.4}$$

where Σ_{max} is the maximum value of the cross section in a single material region and the sample is then accepted with a probability g. The scores of all collision estimators are modified by the factor g. This technique is very similar to delta-tracking where a majorant cross section (Σ_{maj}) is used and defined as the maximum value of the cross section in the whole system, not only the single material region as the case of Σ_{max}. It should be noticed that any macroscopic cross section is proportional to the density and the system is normally defined in such a way that Σ_{max} is calculated with the density at the beginning of the transient. Then, the ratio g defined as the division between the updated cross section and its original value is the ratio between the densities (referred as ρ_k). In addition, Σ_{max} must be defined in order to have $g < 1$ at all times for this method to work [91].

Doppler effect on the neutron cross sections due to the target nucleus thermal agitation is calculated by Serpent 2 using the Target Motion Sampling (TMS) treatment. This technique allows taking into account the Doppler effect on-the-fly during the tracking routine and also dealing with a non-uniform temperature distribution in the region. In its implementation, the code samples the target nucleus velocity at each collision and then solves explicitly the neutron-nucleus collision equations using a target-at-rest reference

[2] In some cases the inversion method is not practical because the inverse is expensive to calculate or it cannot be solved at all (Maxwell-Boltzmann distribution). The rejection method offers an alternative to calculate a sample from a probability density function $g(x)$.

First, a function $g_1(x) = g(x)/g_{Max}$ is defined, where g_{max} is the maximum value taken by $g(x)$. Then, $0 \leq g_1(x) \leq 1$ in the interval $a \leq x \leq b$. After, two random numbers (ζ_1, ζ_2) are generated and interpreted as a point with plane coordinates $(a + \zeta_1(b - a), \zeta_2)$. That is a point inside the rectangle of base $(b - a)$ and height 1. If the point falls below the curve $g_1(x)$, it is accepted as a sample of $g(x)$; if not, it is rejected and a new sample is sorted. The samples obtained this way are proven to follow the distribution of $g(x)$[27]

frame and the neutron cross sections of the material. The code handles the fact that the material total cross section's temperature is a distributed quantity using rejection techniques [92][93][94].

Finally, it is important to note that Serpent 2 can not handle thermal scattering laws for bounded atoms at thermal energies and neither probability tables for unresolved resonances on-the-fly. For these cases, the neutron-cross sections have to be pre-processed and, afterwards, interpolated on-the-fly. This situation was not found in the cases that we have studied in this work, however it may be important in other cases.

4.3.1.1.4 Delayed Neutrons Source

As discussed in the previous chapter, to implement the Quasi-Static Method using the eigenvalue method some minor modifications in the Monte Carlo code (tracking routine) are needed to correctly take into account the effect of the precursors in the flux shape equations. In the case of the adiabatic variant of the Quasi-Static Method these modifications will concern mostly the systems where the delayed neutron precursors can be transported during the accident, such as a system containing a liquid fissile.

In systems containing a solid fissile the fission products stays where the fission took place and the delayed neutrons will be emitted at the same place. For such a system and assuming that the adiabatic approximation holds, one can use equation 3.66 for the resolution of the flux shape equation. The approximation assuming that at each fission point there is a probability β of a delayed neutron emission should not introduce a significant error. Accordingly, delayed neutrons will be sampled with a probability β by the Monte Carlo tracking routine and assigned the corresponding energy and direction distribution. Note that assuming equilibrium conditions between the prompt population and the precursors concentration for the only purpose of the flux shape calculation (i.e. we are not assuming equilibrium conditions in equation 3.60b nor 3.61b) would be a good approximation in most realistic cases. During slow transients, equilibrium can be assumed while for fast transients one can show that the neutron flux shape will be driven by the prompt neutrons and, thus, the equilibrium condition will have a small impact. Also in some systems where non-equilibrium conditions exist and reactivity is close to prompt criticality this approximation gives good results as shown for the case of Godiva. For other cases and as explained in Section 3.1.2.8 the value of β can be modified to take into account the non-equilibrium conditions.

For systems containing a liquid fissile in motion and even if the adiabatic conditions are valid, assuming that the precursors concentration are in equilibrium with the prompt neutron population for solving equation 3.64 will introduce an error in most cases small. As discussed in section 3.1.2.8, we solved this by using equation 3.67 which requires the Monte Carlo tracking routine to sample the delayed neutron position with the precursors concentration. The precursors concentrations are obtained by solving equations 3.60c with the OpenFOAM toolkit for the Finite Volume Method. The precursors distribution is therefore shared with the tracking routine as shown in Figure 4.4. Each time a delayed neutron is sampled, its position will later be assigned, via the rejection method, from OpenFOAM's precursor distribution for the corresponding family. Energy and direction is later handled normally. This strategy is similar to the work of Aufiero et al. [95] developed at the LPSC. As discussed for the solid fuel case, the probability of emission is still taken equal to β thus introducing in most cases a very small error on the flux shape (as it can be seen for the Molten Salt Cavity Benchmark results presented in the next chapter) but it could be easily corrected by modifying the probability of emission in the Monte Carlo tracking routine after sampling the fission reaction.

4.3.1.2 Monte Carlo Quasi-Static Method

In the previous section we established the strategy to solve the coupling between OpenFOAM and Serpent 2 neutronics using the eigenvalue method (thus a steady calculation). However criticality accidents involve transient analysis and, for this reason, in Section 3.1.2 we have proposed a Quasi-Static Method to be able to perform transient neutronics with a Monte Carlo code. We have also seen that there are three main variants of the Quasi-Static Method depending on the approximations that are made: (i) Improved Quasi-Static Method (no approximations made), (ii) Original Quasi-Static Method ($\partial\phi/\partial t \; << (dn/n)/dt \rightarrow \partial\phi/\partial t = 0$) and (iii) Adiabatic Method ($\partial\phi/\partial t = 0$; $dn/dt(t) = 0$).

In this section, we will see that concerning the numerical implementation of any of the variants of the Quasi-Static Method, there are two principal strategies:

- Improved Quasi-Static Method (IQM) (not to be confused with the Improved Quasi-Static Method variant)
- Predictor-Corrector Quasi-Static Method (PCQM)

We will firstly describe these two algorithms to show their advantages and inconveniences and then we will present the hybrid approach that was taken in this work to perform the numerical implementation. As we will see neither of them are used in their totality and rather features from both are taken for our approach. Indeed, the adiabatic variant discussed in Section 3.1.2.8 was implemented in the multi-physics tool in a similar way (but not exactly) to the PCQM and thus it will be referred here as OpenFOAM PCQM (OF-PCQM).

4.3.1.2.1 Improved Quasi-Static Method (IQM)

In this numerical approach, both systems of equations (neutron flux shape and amplitude) are converged at the same time as proposed by Dulla et al. [43] and later used also in the work of Patricot [47]:

1. An initial angular neutron flux $\psi(\vec{r}, \vec{\Omega}, E, t_0)$ and a weight function $W_0(\vec{r}, \vec{\Omega}, E)$ are provided. With this initialisation the normalization constant is established in order that $\psi(\vec{r}, \vec{\Omega}, E, t_0)$ also satisfies it (equation 4.5). There is still one degree of freedom so $n(t_0)$ can be arbitrarily set. But once it is chosen this will set the value of the *constant*.

$$\begin{aligned}\left\langle \frac{1}{v(E)} \frac{\psi(\vec{r}, \vec{\Omega}, E, t_0)}{n(t_0)} \;\middle|\; W_0(\vec{r}, \vec{\Omega}, E) \right\rangle &= \left\langle \frac{1}{v(E)} \phi(\vec{r}, \vec{\Omega}, E, t_0) \;\middle|\; W_0(\vec{r}, \vec{\Omega}, E) \right\rangle \\ &= constant \end{aligned} \tag{4.5}$$

   During the neutron flux resolution two different time steps are used. The smaller time step (δt) is used to integrate the point kinetics (or amplitude) equations 3.61 for solving the neutron amplitude while a larger time step (Δt) is used for the neutron flux shape equations 3.60. In order to decrease the computational cost, δt will be set so that $\delta t \; < \Delta t$ and thus an effort to maximize Δt must be made. This way the resolution of the multi-variable equations for the neutron flux shape (which are also the most expensive computationally) is done less frequently as possible.

2. Using the equations 3.62 and the shape function $\phi(\vec{r}, \vec{\Omega}, E, t_0)$ normalized with equation 4.5, the point kinetics parameters are calculated.

3. Using the kinetics parameters from step 2 a first estimation of $n(\Delta t)$ is made integrating the amplitude equations (with the time step δt) during the interval Δt.

4. Using $n(\Delta t)$ and its derivative $d\left[n(\Delta t)\right]/dt$ the shape equations are solved to determine the neutron shape function $\tilde{\phi}(\vec{r},\vec{\Omega},E,\Delta t)$. The error induced in the normalisation condition is calculated as follows:

$$error_{norm} \;=\; \left| \frac{\left\langle \frac{1}{v(E)}\tilde{\phi}(\vec{r},\vec{\Omega},E,\Delta t) \;\middle|\; W_0(\vec{r},\vec{\Omega},E) \right\rangle \;-\; constant}{constant} \right| \tag{4.6}$$

5. Since $\tilde{\phi}(\vec{r},\vec{\Omega},E,\Delta t)$ must satisfy the normalization condition set by equation 4.5 the shape function has to be corrected by the scalar factor Z as:

$$\phi(\vec{r},\vec{\Omega},E,\Delta t) \;=\; Z\;\tilde{\phi}(\vec{r},\vec{\Omega},E,\Delta t) \tag{4.7}$$

where Z is calculated from:

$$Z \;=\; \frac{constant}{\left\langle \frac{1}{v(E)}\tilde{\phi}(\vec{r},\vec{\Omega},E,\Delta t) \;\middle|\; W_0(\vec{r},\vec{\Omega},E) \right\rangle} \tag{4.8}$$

6. With the normalized flux shape the point kinetic parameters are recalculated (step 2) and the flux amplitude equation integrated (step 3) again to obtain a new estimation of $n(\Delta t)$.

7. Steps 4-6 are also repeated until convergence of the normalization error $(error_{norm})$.

An schematic description of this process can be seen in Figure 4.5

4.3.1.2.2 Predictor-Corrector Quasi-Static Method (PCQM)

In the PCQM the shape equations are not used but rather the original transport equations 3.1 and 3.6 are used to calculate the flux shape. The algorithm can be schematized as follows [43]:

1. As it was done in section 4.3.1.2.1, an angular neutron flux $\psi(\vec{r},\vec{\Omega},E,t_0)$ and a weight function $W_0(\vec{r},\vec{\Omega},E)$ are set. Then the normalization constant and $n(t_0)$ are set such as $\psi(\vec{r},\vec{\Omega},E,t_0)$ verifies the following equation (note that there is one degree of freedom so either $n(t_0)$ or the *constant* can be arbitrarily set):

$$\left\langle \frac{1}{v(E)}\frac{\psi(\vec{r},\vec{\Omega},E,t_0)}{n(t_0)} \;\middle|\; W_0(\vec{r},\vec{\Omega},E) \right\rangle \;=\; constant \tag{4.9}$$

2. Again there are two time scales to be considered: Δt for the transport equation

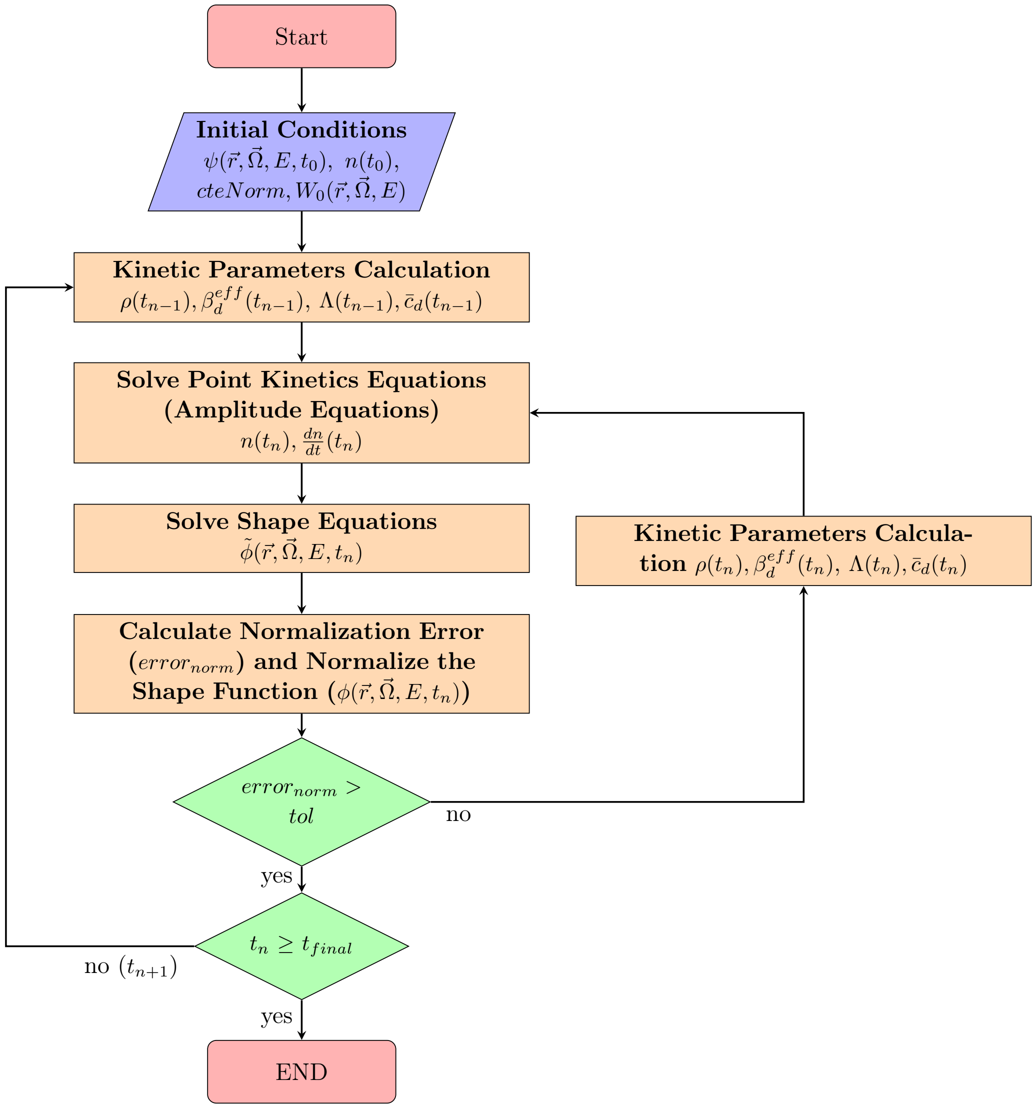

Start
Initial Conditions
$\psi(\vec{r},\vec{\Omega},E,t_0),\ n(t_0),$
$cteNorm, W_0(\vec{r},\vec{\Omega},E)$
Kinetic Parameters Calculation
$\rho(t_{n-1}), \beta_d^{eff}(t_{n-1}),\ \Lambda(t_{n-1}), \bar{c}_d(t_{n-1})$
Solve Point Kinetics Equations (Amplitude Equations)
$n(t_n), \frac{dn}{dt}(t_n)$
Solve Shape Equations
$\tilde{\phi}(\vec{r},\vec{\Omega},E,t_n)$
Kinetic Parameters Calculation $\rho(t_n), \beta_d^{eff}(t_n),\ \Lambda(t_n), \bar{c}_d(t_n)$
Calculate Normalization Error ($error_{norm}$) and Normalize the Shape Function ($\phi(\vec{r},\vec{\Omega},E,t_n)$)
$error_{norm} > tol$
no
yes
$t_n \geq t_{final}$
no (t_{n+1})
yes
END

Figure 4.5 – Improved Quasi-Static Method scheme.

and δt for the amplitude equations. By solving the transport equations 3.1 and 3.6 in Δt a prediction for $\psi(\vec{r},\vec{\Omega},E,\Delta t))$ is obtained (any available resolution method for the transport equation can be used for this). This predicted value is called $\tilde{\psi}_{predicted}(\vec{r},\vec{\Omega},E,\Delta t))$. It should be noticed that *solving the transport equations* is not a trivial task as explained in Section 3.1 and a numerical method has to be used for this purpose (Diffusion, $\mathrm{SP_N}$, Monte Carlo, among others).

3. This first estimation must verify the normalization condition defined in equation 4.9. Thus a corrective scalar factor $\tilde{Z}$ is defined such as:

$$\psi_{predicted}(\vec{r},\vec{\Omega},E,\Delta t) \;=\; \tilde{Z}\ \tilde{\psi}_{predicted}(\vec{r},\vec{\Omega},E,\Delta t) \qquad (4.10)$$

where $\tilde{Z}$ is calculated as

$$\tilde{Z} = \frac{constant}{\left\langle \frac{1}{v(E)}\tilde{\psi}_{predicted}(\vec{r},\vec{\Omega},E,\Delta t) \;\middle|\; W_0(\vec{r},\vec{\Omega},E) \right\rangle} \tag{4.11}$$

Note that if we want to keep the constraint set by equation 4.9, the predicted flux should be divided by the amplitude function evaluated at the shape time ($n(\Delta t)$). However, $n(\Delta t)$ would divide the predicted flux ($\tilde{\psi}_{predicted}$) in equation 4.10 and also divide the predicted flux ($\tilde{\psi}_{predicted}$) inside the integral in the denominator of the factor $\tilde{Z}$ in equation 4.11. Given that the integration is not carried in time, the amplitude equation value ($n(\Delta t)$) exits the integral and vanishes when carrying out equation 4.10 if $\tilde{Z}$ is replaced by its definition (the factor $1/n(\Delta t)$ appears both in the numerator and the denominator, thus, it is omitted).

4. Using $\psi_{predicted}$ the kinetic parameters are evaluated and the amplitude equations are solved during the interval Δt.
5. Having calculated $n(\Delta t)$ the flux can be corrected as shown in equation 4.12 and find corrected values of the precursors concentration from equation 3.6.

$$\psi_{corrected}(\vec{r},\vec{\Omega},E,\Delta t) = n(\Delta t)\psi_{predicted}(\vec{r},\vec{\Omega},E,\Delta t) \tag{4.12}$$

Steps 2-5 are sequentially repeated for all time steps Δt until the final time is reached. An schematic description of this process is presented in Figure 4.6.

The two algorithms explained (IQM and PCQM) have some limitations either in the implementation or the accuracy. As we will see in the next section a compromise solution have been opted to keep both to some extent.

4.3.1.2.3 OpenFOAM Predictor-Corrector Quasi-Static Method (OF-PCQM)

As we have seen in the IQM or the PCQM numerical implementations two time steps are normally required by the Quasi-Static Method. A smaller one δt for the amplitude equations and a larger one Δt for the shape equations. However, OpenFOAM code allows the *user* to define only one time step[3] to integrate the partial differential equations and thus all the solvers will be forced to use the same time step. Moreover, information on the

[3] There has been some attempts to use multiple time steps as seen in [96]

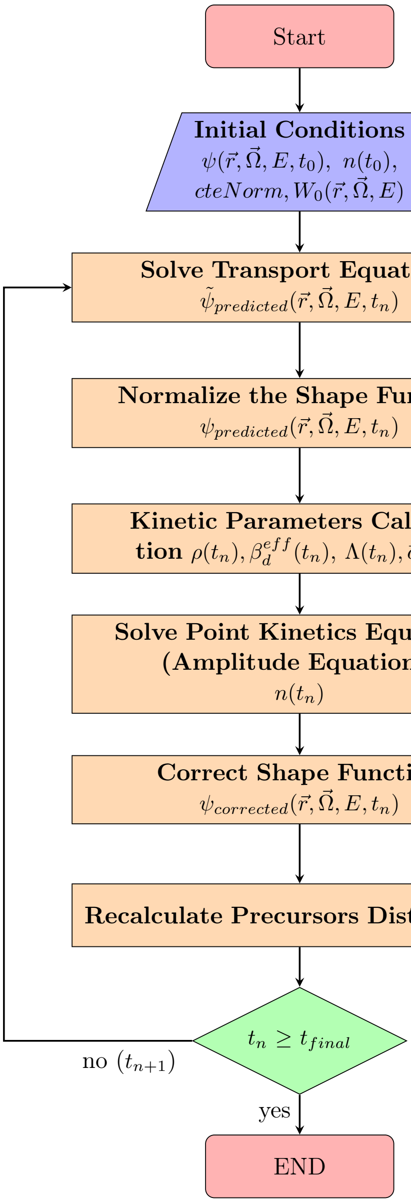

Start
Initial Conditions
$\psi(\vec{r},\vec{\Omega},E,t_0),\ n(t_0),$
$cteNorm, W_0(\vec{r},\vec{\Omega},E)$
Solve Transport Equations
$\tilde{\psi}_{predicted}(\vec{r},\vec{\Omega},E,t_n)$
Normalize the Shape Function
$\psi_{predicted}(\vec{r},\vec{\Omega},E,t_n)$
Kinetic Parameters Calculation $\rho(t_n), \beta_d^{eff}(t_n),\ \Lambda(t_n), \bar{c}_d(t_n)$
Solve Point Kinetics Equations (Amplitude Equations)
$n(t_n)$
Correct Shape Function
$\psi_{corrected}(\vec{r},\vec{\Omega},E,t_n)$
Recalculate Precursors Distribution
$t_n \geq t_{final}$
no (t_{n+1})
yes
END

Figure 4.6 – Predictor-Corrector Quasi-Static Method Scheme.

runtime variables is only stored for the current time step. These limitations will have two consequences:

1. The time step used by all the OpenFOAM solvers will have to be the smaller one, i.e. the corresponding to the amplitude time step. By using the amplitude time step, the parameters of the point kinetics equations can be updated as the material properties change (e.g. density and temperature) because of the coupling with the other solvers (e.g. energy balance equations). This choice will also provide the non-neutronic solvers with updated power amplitude estimations for each time step.

2. The IQM numerical implementation strategy becomes challenging to implement in the multi-physics tool. To iterate as shown in Figure 4.5 we should be able to store

all the OpenFOAM runtime variables at the beginning of the current flux shape iteration and to restart the shape time step at the end of it (i.e. many δt steps before the beginning of the shape interval). That is not at all straightforward nor obvious to implement in OpenFOAM.

For these reasons, an alternative strategy close to the PCQM was preferred. In this alternative strategy for the numerical implementation, called OF-PCQM, only one internal iterations in the IQM shape step (Δt) is used. Therefore, this implies that in Figure 4.5 the internal loop is not done. This approximation will provide accurate results if the normalization error ($error_{norm}$) is low enough. In any case, accuracy can be increased, as in any explicit scheme, by reducing the shape time step. Therefore, calculation time is penalized for the sake of simplicity by using smaller flux shape steps.

As discussed in previous sections, in this work we use the adiabatic simplification (Section 3.1.2.8) thus the neutron shape equations are similar to those of the criticality calculation from Section 4.3.1.1. The resulting algorithm for the OF-PCQM using the adiabatic variant is shown in Figure 4.7.

It should be noticed that when using the IQM or the OF-PCQM any of the simplifications from Section 3.1.2.8 can be used. In the case of the PCQM the only simplification possibility is to neglect the angular flux time derivative ($\partial\psi/\partial t \simeq 0$) which is equivalent to neglecting both the amplitude and the shape ($dn/dt \simeq 0$ and $\partial\phi/\partial t \simeq 0$). The treatment of the delayed neutron source is equivalent for all methods as explained in Sections 3.1.2.8.3 and 4.3.1.1.4.

Tallies and amplitude parameters

The OF-PCQM algorithm, previously described, implemented in the multi-physics tool requires the determination several neutronics parameters during the Monte Carlo calculation. As discussed, these parameters are estimated from tallies (see Section 3.1.1.3). These tallies include cell-wise reaction rates and neutron current calculated in the same manner as described for the volumetric power in section 4.3.1.1 with the collision estimator as:

$$R(\vec{r_j}) = \frac{1}{V_j}\sum_i \omega_i \frac{f(\vec{r}_i, E_i))}{\Sigma(\vec{r}_i, E_i)} \qquad \forall\ \vec{r}_i \in V_j \tag{4.13}$$

where ω_i, $\vec{r}_i$ and E_i are the weight, position and energy of the i-th neutron simulated, V_j is the j-th cell volume, f is a response function (defined in Table 4.1), and Σ is the cross

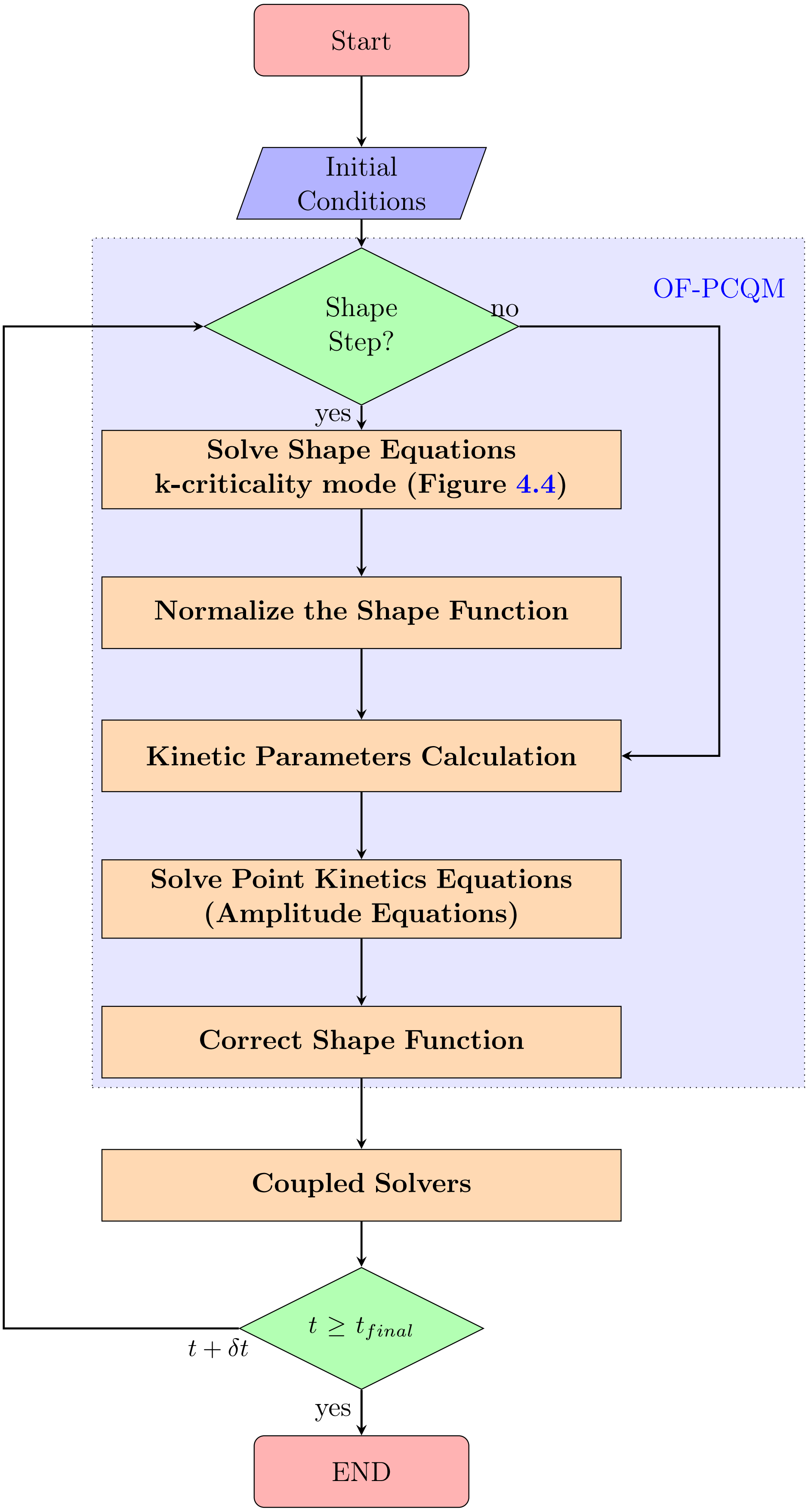

Start
Initial Conditions
OF-PCQM
Shape Step?
no
yes
Solve Shape Equations k-criticality mode (Figure 4.4)
Normalize the Shape Function
Kinetic Parameters Calculation
Solve Point Kinetics Equations (Amplitude Equations)
Correct Shape Function
Coupled Solvers
$t \geq t_{final}$
$t + \delta t$
yes
END

Figure 4.7 – OpenFOAM Predictor Corrector Quasi-Static Method (OF-PCQM) algorithm.

section that was used for sampling the path length.

When the above tallies involve neutron cross sections, the response function f is multiplied by the density correction g and the tallies take into account the Doppler effect by using the Target Motion Sampling (TMS) treatment. These two corrections allow the tool to take into account the density and temperature gradients in the materials. The full list of the different response functions needed in our calculations are summarized in Table 4.1.

Table 4.1 – Response functions for the unweighted collision estimator. $\vec{\omega}$ are the direction cosines of $\vec{v}$.

	Response Function $f(\vec{r}, E)$	**Parameter**
Reaction Rate r	$g\Sigma_r$	$\Sigma_r\phi(\vec{r})$
Production Rate	$g\nu\Sigma_f$	$\nu\Sigma_f\phi(\vec{r})$
Volumetric Power	$g\epsilon\Sigma_f$	$\epsilon\Sigma_f\phi(\vec{r})$
Neutron Current	$\vec{\Omega}$	$\vec{J} = \int \vec{\omega}\psi(\vec{r}, \vec{\Omega})d\Omega$
Neutron Density	$1/v$	$N\vec{r} = \frac{1}{v}\phi(\vec{r})$

The response functions make it possible the estimation of the volumetric power and the different point kinetics parameters needed for the amplitude equations. These point kinetics parameters are determined according to the integral equations 3.62. Note that the tallies calculated with equation 4.13 represent integrals defined in an unweighted manner, i.e., in these tallies the weight function $W_0(\vec{r}, \vec{\Omega}, E)$ is equal to a unity constant value. That is because in our work we have indeed chosen to set $W_0(\vec{r}, \vec{\Omega}, E) = 1$ to simplify the calculation of the integral kinetic parameters of equations 3.62. In addition, this choice has the additional advantage of simplifying the modeling of a liquid fissile system with closed boundaries as we had discussed in see Section 3.1.2.3. A more traditional option would be, as explained in Section 3.1.2.5, to use the adjoint flux as a weight function. Nevertheless, in a Quasi-Static Method approach the weight function is set at the beginning of the transient and kept constant during the entire transient calculation. If the initial adjoint flux is used it may have little relation with the system as it evolves. On the other hand, if an instantaneous adjoint flux is used the derivation of the method used here is no longer valid and a time dependent derivation approach such as the one of Becker [49] should be implemented, if possible. Note that Serpent 2 capabilities can calculate adjoint-weighted parameters in the criticality mode. However, this adjoint flux corresponds to the fundamental mode at steady-state given the material properties and geometry of the system at that particular moment. Then, the more complex variational approach developed by Becker [49] (Section 3.1.2.5) for the Quasi-Static Method should be instead use. Its compatibility with a Monte Carlo solver should be investigated but it is certainly more complex than the approach taken in this work.

4.3.1.3 Simplified $\mathrm{P_N}$

As discussed in the previous chapter, an alternative neutronics model based on the Simplified $\mathrm{P_N}$ method has been adopted. The Simplified $\mathrm{P_N}$ model described in Section 3.1.3 was fully implemented in OpenFOAM. Since the sequential approach is necessary in OpenFOAM, the multi-group scheme has to be divided into implicit and explicit terms as shown in equation 4.2.

Using this idea, equations 3.78 for each energy group can be rewritten as follows:

$$\overbrace{\frac{1}{v^g}\frac{\partial U_1^g}{\partial t} = \nabla\cdot\left(\frac{1}{3\Sigma_1^g}\nabla U_1^g\right) - \Sigma_0^g U_1^g}^{Implicit} \\ \underbrace{+2\Sigma_0^0\Phi_2^g + 2\,\frac{1}{v^g}\frac{\partial \Phi_2^g}{\partial t} + \frac{F^g}{k} + S_d^g + S_0^g}_{Explicit} \tag{4.14a}$$

$$\overbrace{3\,\frac{1}{v^g}\frac{\partial \Phi_2^g}{\partial t} = \nabla\cdot\left(\frac{3}{7\Sigma_3^g}\nabla \Phi_2^g\right) - \left(\frac{5}{3}\Sigma_2^g + \frac{4}{3}\Sigma_0^g\right)\Phi_2^g}^{Implicit} \\ \underbrace{+\frac{2}{3}\,\frac{1}{v^g}\frac{\partial U_1^g}{\partial t} - \frac{2}{3}\frac{F^g}{k} - \frac{2}{3}S_d^g - \frac{2}{3}\left(S_0^g - \Sigma_0^g U_1^g\right) + \frac{5}{3}S_2^g}_{Explicit} \tag{4.14b}$$

where

$$F^g = (1-\beta)\chi_p^g \sum_{g'} \nu\Sigma_f^{g'}\left(U_1^{g'} - 2\Phi_2^{g'}\right) \qquad \textbf{Prompt Fission Source}$$

$$S_d^g = \chi_d^g \sum_l \lambda_l C_l \qquad \textbf{Delayed Neutron Source}$$

$$S_0^g = \sum_{g'\neq g} \Sigma_{s0}^{gg'}\left(U_1^{g'} - 2\Phi_2^{g'}\right) \qquad \textbf{Order 0 Scattering Source}$$

$$S_2^g = \sum_{g'\neq g} \Sigma_{s2}^{gg'}\Phi_2^{g'} \qquad \textbf{Order 2 Scattering Source}$$

$$\Sigma_n^g = \Sigma_t^g - \Sigma_{sn}^{gg} \qquad \textbf{Order n Removal Cross Section}$$

where all quantities were defined to take the multi-group formulation into account using the superscript g to denote the energy group. v^g and Σ_r^g are the neutron speed and the macroscopic cross section for reaction r, respectively; χ_p^g and χ_d^g are the fractions of prompt and delayed neutrons born in group g; $\Sigma_{sn}^{gg'}$ is the scattering macroscopic cross section from the energy group g' to g of order n, i.e., is the coupling reaction between any two energy groups; and Φ_0^g and Φ_2^g are the modes of the flux expanded in Legendre Polynomials of orders 0 and 2 respectively for group g as shown in Section 3.1.3 with $U_1 = \Phi_0 + 2\Phi_2$.

In Figure 4.8 the algorithm used to solve the $\mathrm{SP_N}$ equation is sketched. Depending on the order of the expansion the algorithm can solve either the $\mathrm{SP_1}$/Diffusion equations or the $\mathrm{SP_3}$ equations. Precursors advection in a system containing a liquid fissile is also taken into account if needed.

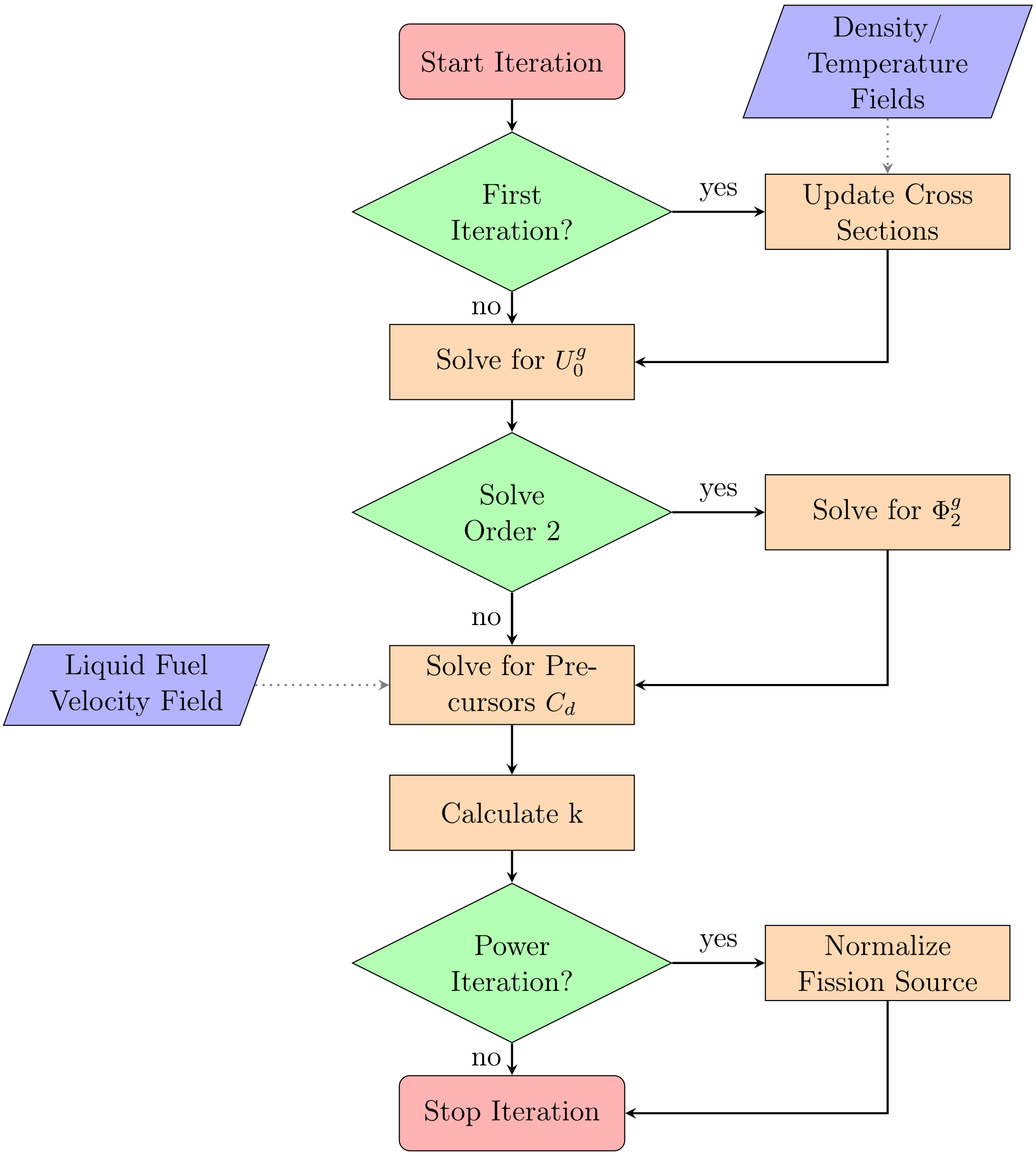


FIGURE 4.8 – Simplified PN implementation.

For steady-state calculations a classic **power iteration** technique is used [21] to converge the flux solution. The **power iteration** consists in normalizing the fission source by the multiplication factor k after each iteration until convergence of the fission source and the k value. For example, for the iteration $j+1$ the multiplication factor k is calculated as follows:

$$k_{j+1} = \frac{\sum_g \int \left(F^g_{j+1} + S^g_{d,j+1}\right) dV}{\frac{1}{k_j}\sum_g \int \left(F^g_j + S^g_{d,j}\right) dV} \tag{4.15}$$

This procedure is similar to Monte Carlo algorithm in criticality source mode. Note that the multiplication factor has been placed in the explicit term of equations 4.14.

It is easy to see that for a solid fuel it is equivalent to calculate (both numerator and denominator) as the integral of the ν-fission rate, i.e.,

$$\int \left(F^g_{j+1} + S^g_{d,j+1}\right) = \int \sum_{g'} \nu\Sigma_f^{g'} \left(U_1^{g'} - 2\Phi_2^{g'}\right) dV \tag{4.16}$$

For a liquid fuel, it is not so trivial and the previous equivalence holds only in case of no loss of precursors through the system boundaries and incompressible fuel with non-slip condition, which is exactly the case treated in this work. Indeed, taking equation 3.7 with $\partial C_d/\partial t = 0$ and summing over all precursors family and then applying the divergence theorem as in Section 3.1.2.3 for the advection and diffusion terms we obtain that the delayed neutron source is:

$$\begin{aligned}
\sum_d \int \lambda_d C_d(\vec{r},t)dV &= \sum_d \int \Big[\nabla\cdot\left[D_d\nabla C_d(\vec{r},t)\right] \;+\; \beta_d\mathcal{F}\psi(\vec{r},\vec{\Omega},E,t) \;-\; \vec{u}\cdot\nabla C_d(\vec{r},t)\Big]\, dV \\
&= \sum_d \left[\int \nabla\cdot\left[D_d\nabla C_d(\vec{r},t) + \vec{u}C_d(\vec{r},t)\right] dV \;+\; \int \beta_d\mathcal{F}\psi(\vec{r},\vec{\Omega},E,t)dV\right] \\
&= \sum_d \left[\cancelto{=0}{\int \left[D_d\nabla C_d(\vec{r},t) + \vec{u}C_d(\vec{r},t)\right]\cdot\vec{n}\; dS} \;+\; \int \beta_d\mathcal{F}\psi(\vec{r},\vec{\Omega},E,t)dV\right] \\
&= \beta \int \mathcal{F}\psi(\vec{r},\vec{\Omega},E,t)dV
\end{aligned} \tag{4.17}$$

If we add this result to the prompt fission, equation 4.16 holds but it is limited to solid fuels and the incompressible case with non-slip and no loss of precursors through the boundaries condition. For this reason, equation 4.15 is used in this work to keep things as general as possible.

4.3.1.3.1 Cross Sections

Simplified $\mathrm{P_N}$ model requires the *user* to provide the nuclear system neutron reactions macroscopic cross sections condensed to N_G energy groups and homogenized in a volume representative of the cells of the spatial mesh. Once the transient calculation has started, these macroscopic cross sections have to be updated according to the system modifications. The **Update Cross Section** block from Figure 4.8 takes into account how density and temperature changes in the system affects the macroscopic cross sections. Density effects are again straightforward to be taken into account in the macroscopic cross sections since they are proportional to the atomic density which is also proportional to the material density. For this reason, a cell-wise correction, like the one for Monte Carlo, is used (equation 4.18). Thermal effects, on the other hand, are more challenging. However, for small changes in the temperature, a logarithmic interpolation between cross sections library data for two pre-calculated temperatures bounding the system was implemented. In total, the macroscopic cross sections are updated cell-wise in the following way:

$$\Sigma(\vec{r}) = \frac{\rho(\vec{r})}{\rho_0}\left(\Sigma_{ref} - \delta\Sigma_{\alpha} log\left(\frac{T(\vec{r})}{T_{ref}}\right)\right) \tag{4.18}$$

where

$$\delta\Sigma_{\alpha} = \frac{\Sigma_{max} - \Sigma_{ref}}{log\left(T_{max}/T_{ref}\right)}$$

with Σ_{max} and Σ_{ref} the cross section corresponding to the maximum and reference (or lower limit) temperature considered T_{max} and T_{ref}, respectively.

It is important to remark that the determination of the condensed and homogenized macroscopic cross sections of the system can be very challenging in the case of highly heterogeneous systems or in systems with important neutron leaks. In such systems spatial and energy self-shielding effects could be very important (e.g. in PWR fuel rods or in the control rods). Neutron leaks will also complicate the homogenization process. For such heterogeneous systems very fine energy and space meshes are usually required together with specific algorithms to treat neutron cross sections resonances and leakage. Most of the cases studied in this PhD were homogeneous (Godiva and the Molten salt Cavity) with the exception of the Spent Fuel Pool and thus these effects were less of an issue for the Monte Carlo code. The latter can deal with these through detail modeling of the geometry, continuous in energy nuclear data and the discussed routines for density and temperature changes (Section 4.3.1.1.3) along with its computational cost.

4.3.1.3.2 Boundary Conditions

The Marshak boundary conditions from equations 3.81 were implemented. Moukalled [64] gives a good insight on how to implement new boundary conditions in OpenFOAM. A generic boundary condition for a variable ϕ can be written as:

$$\nabla\phi = a(b\phi + c) \tag{4.19}$$

where a, b and c are parameters that have to be set according to the system conditions. The gradient is discretized as follows:

$$\nabla\phi \cdot \vec{n} = \left(\phi_{face} - \phi_{center}\right)\Delta \tag{4.20}$$

where Δ is the inverse center-face distance parallel to the normal vector $\vec{n}$ at the boundary. Then, the value used in OpenFOAM for calculating the ϕ variable at the boundary face center is:

$$\phi_{face} = \frac{ac\Delta}{1 - \Delta ab} + \phi_{center}\frac{1}{1 - \Delta ab} \tag{4.21}$$

This was applied to each flux order and energy group fluxes BCs. Ideally, the whole system (including the boundary conditions) would be evaluated in a block coupling scheme (everything solved at the same time). However, the sequential nature of OpenFOAM requires an iterative process. Nonetheless, some efforts to do block coupling in non-official OpenFOAM versions (e.g. extend version) exist at this time.

Boundary conditions 3.81a is assigned to solve for the order 0 equations (U_1^g) and boundary condition 3.81b to solve for the order 2 (Φ_2^g). Then, iterating over all orders and energies groups the system of equations is converged.

4.3.1.3.3 Acceleration

Straight numerical resolution of the SP$_3$ method in an iterative process has the disadvantage of slow convergence. For this reason, an Aitken acceleration routine as explained by Fiorina et al. [97] was implemented. In addition to the Aitken acceleration routine, an Anderson acceleration [98] technique was also tested[4]. Both involve a prediction of the next iteration solution based on several previous ones. Both techniques were able to significantly improve the performance of the model as will be shown in Section 5.1.6.

4.3.2 Solid Mechanics

The main thermomechanics model required for the cases identified in Chapter 2 was the linear thermoelasticity model presented in equation 3.99 (Section 3.2.2.1). A numerical solver implementing this model already exists in the libraries of OpenFOAM under the name *solidDisplacementFoam*. The details on this solver equations and the boundary conditions can be found in the work of Jasak and Weller [60] and will not be reviewed here.

The strategy used by *solidDisplacementFoam* is very similar to the algorithms discussed in the previous sections: equation 3.99 is divided into implicit and explicit terms as shown in equation 4.22. Then, this new equation is used to find the solution by an iterative process until a convergence is attained by minimizing the residual error or until a maximum number of iterations is achieved.

$$\overbrace{\frac{\partial^2\left(\rho\vec{D}\right)}{\partial t^2} - \nabla\cdot\left[(2\mu+\lambda)\,\nabla\vec{D}\right]}^{implicit} = \underbrace{\nabla\left[\mu\left(\nabla\vec{D}\right)^T + \lambda\ I\ tr\left(\nabla\vec{D}\right) - (\mu+\lambda)\,\nabla\vec{D}\right] - \nabla\left(\frac{E}{1-2\nu}\alpha T\right) + \rho\vec{f}}_{explicit} \tag{4.22}$$

An interesting discussion can be found in the work of of Jasak and Weller [60] on the performance obtained for solving solid mechanics problems using FVM methods (principal method employed by OpenFOAM) in comparison to the Finite Element Method (FEM) formulation more commonly used for these type of problems, as well as other numerical considerations. From the considerations of Jasak and Weller's work it can be concluded that FEM methods have not necessarily a better performance than the FVMs. For example, the direct solvers used in the FEM are computationally more expensive than the iterative FVM solvers. In addition, as non-linearities and explicit coupled terms appear, especially when coupling with other physics, the FVM seems to be an interesting alternative to the FEM. Finally, using a Finite Volume Method to discretize the solid mechanics equations

[4] It was implemented using the C++ liner algebra library Armadillo [99]

has the advantage of keeping the same environment for the resolution of all the models in the multi-physics tool reducing the complexity of the implementation.

4.3.2.1 Geometry

The linear thermoelasticity model used here is valid when the problem does not involve volumes with moving boundaries. All equations in this model were derived from the Reynolds Transport Theorem from Section 3.3.2 using a fixed Control Volume (i.e. non-moving). Indeed, for very small deformations, which is the main hypothesis for the linear elasticity equations, it can be shown that an approximately fixed, even when deformed, Control Volume can be used to discretize the equations as explained by Demirdžić et Muzaferija [62].

To keep the errors introduced to a minimum, when studying a transient in systems where the solid phase undergoes a thermal expansion as a result of a power excursion such as in the Godiva experiments, two meshes were used in the multi-physics tool. A fixed-volume mesh is used to solve for the solid displacements, temperatures and velocities fields. For the neutronics, on the other hand, the equations are solved in a deformable mesh. Therefore, the neutronics mesh expands or contracts according to the displacement field calculated by the thermal-mechanics module. Input/Output fields used by the neutronics calculations are thus interpolated from one mesh to the other taking into account the volume expansion or contraction at each cell in order to conserve quantities. For example, density is cell-wise modified with the cell volumes to conserve total mass in the deformable mesh.

4.3.3 Thermal-Hydraulics

The third physics field solved in the multi-physics tool is the thermal-hydraulics domain for the fluid phase existing in the nuclear system. Thermal-hydraulics is a vast field that can cover very different fluid systems. In our studies, three different types were encountered: laminar flow, turbulent flow and flow in porous medium. OpenFOAM contains many numerical solvers that were developed for very different types of flow. In the case of the laminar and turbulent flows, the already existing OpenFOAM solvers were therefore implemented in the tool. For the case of the flow in a porous medium, the existing solvers were not sufficiently adapted for the study of the Spent Fuel Pool and thus a new solver had to be developed and implemented in OpenFOAM.

This section provides an overview of the PIMPLE algorithm used by OpenFOAM to solve the incompressible Navier-Stokes equations since very little information can be found in the literature and also because this solver has to be coupled with the other algorithms of the multi-physics tool. Then, the porous medium algorithm is discussed and at the end a short discussion is provided on the multi-region capabilities of the tool. That is a key feature of the tool that allows solving conjugated heat transfer (e.g. heat transfer and exchange a flow and a solid phase). While this feature was not used in the cases presented in this manuscript, it was implemented in the tool and will be necessary for future studies.

4.3.3.1 PIMPLE Algorithm

The incompressible Navier-Stokes Equations from Section 3.3.3 are not easy to solve mainly due to their non-linearities given by the advection term. In addition, there is no explicit equation to calculate the pressure. To overcome these problems, historically, there are two algorithms to solve the Navier-Stokes equations: the Semi-Implicit Method for Pressure Linked Equations (SIMPLE) algorithm for steady-state calculations and the Pressure-Implicit with Splitting of Operators (PISO) algorithm for transient calculations.

In both algorithms (SIMPLE and PISO), the velocity field is obtained from the momentum equation 3.120 with a first guess of the pressure p. In addition, the velocity field should satisfy the constrain imposed by the continuity equation 3.110. Normally, that is not the case, and therefore a new guess of the pressure is necessary. This back and forth between pressure guess and the continuity equation constrain can continue aimlessly. As a consequence, a way to predict the next pressure guess is needed. This critical step is accomplished by the particular formulation of the SIMPLE/PISO algorithms.

First, the momentum equations can be rewritten in a matrix shape as:

$$\rho \frac{D\vec{u}}{Dt} = -\nabla p + \mu \nabla^2 \vec{u} + f_b \qquad \Rightarrow \qquad \mathcal{M}U = -\nabla p \tag{4.23}$$

with ∇p the matrix form of the pressure gradient, $\mathcal{M}$ a discretized operator and U being the velocity matrix.

Further, it can be splitted between a matrix holding only the diagonal elements of $\mathcal{M}$, to which we will refer as $\mathcal{A}$, and a residual matrix $\mathcal{H}$:

$$\mathcal{H} = \mathcal{M}U - \mathcal{A}U \tag{4.24}$$

To find an equation to estimate the pressure we take advantage of the continuity equation, which is independent of the others and, after some manipulation an expression for U can be found by replacing equation 4.24 in equation 4.23.

$$\mathcal{M}U = \mathcal{A}U - \mathcal{H} = -\nabla p \qquad \Rightarrow \qquad U = \mathcal{A}^{-1}\mathcal{H} - \mathcal{A}^{-1}\nabla p \tag{4.25}$$

Replacing this definition for U in the continuity equation we obtain our pressure corrector:

$$\nabla\left(\mathcal{A}^{-1}\nabla p\right) = \nabla\left(\mathcal{A}^{-1}\mathcal{H}\right) \tag{4.26}$$

Now, we have a momentum predictor and a pressure corrector. So the pressure-velocity coupling algorithmic can be establishes as follows:

1. Solve the momentum predictor from equation 4.23 using a pressure initial guess or the value from the previous iteration or time step.
2. Calculate the residual $\mathcal{H}$ from equation 4.24 with the new velocity field
3. Solve the pressure corrector from equation 4.26
4. Solve for other equations: turbulent quantities such as $k-\epsilon$ for example, or the energy balance, or species transport.
5. Check residuals, i.e., to what degree the fields $\vec{u}$ and p satisfies the Navier-Stokes equations
6. Check convergence

Up to this point, both SIMPLE and PISO algorithms are identical. The difference relies on how the iterations are performed. For the SIMPLE algorithm, if it is not yet converged, the whole iteration is restarted from the momentum predictor. These iterations are referred to as **outer iterations**. For the PISO algorithm, the momentum predictor is solved only once, i.e., only one outer iteration. Then, the pressure equation is run many times by updating the velocity field $\vec{u}$ and, therefore, the residual $\mathcal{H}$ in what are called **inner iterations**. Both algorithms can be observed in Figure 4.9.

The SIMPLE algorithm was developed by Spalding and Patankar [101][102] at the

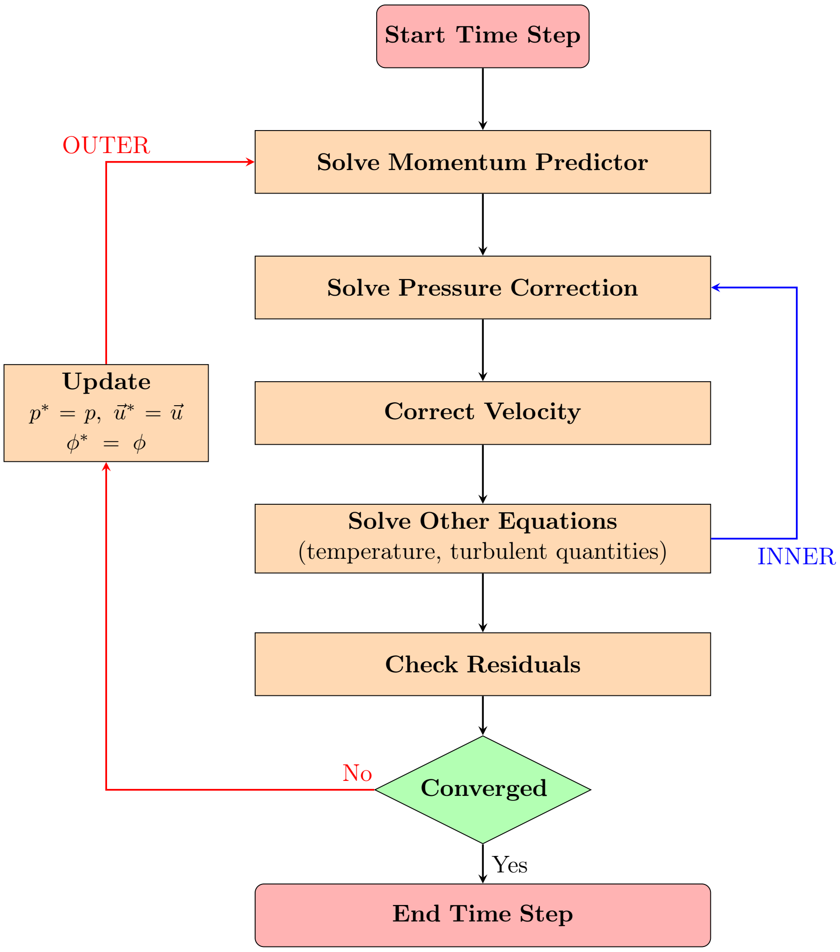


Figure 4.9 – SIMPLE, PISO and PIMPLE Algorithms [100].

Imperial College for steady-state calculations, i.e., there is no time derivative. It can take several thousand outer iterations to converge. This algorithm can be also used for transient calculations by using the same procedure at each time step. However, the computational cost of solving the momentum predictor involving many thousand iterations at each time step is too high. This is mainly why the PISO algorithm was proposed by Issa in 1986 [103]. If the Courant number is inferior to one ($Co < 1$) a couple of inner iterations are normally enough and the algorithm is stable for transient calculations. However, for detailed meshes (for example, to observed vortex formation) the time step Δt has to be reduced in order to satisfy the Courant restriction.

When solving transient equations, the additional time derivative term adds a $1/\Delta t$ to the diagonal. Because of this, with a lower time step the matrix becomes more diagonal dominant and, therefore, more stable. In the case of the SIMPLE algorithm, this diagonal dominant effect is achieved via under-relaxation techniques (for steady-state there is no time derivative). There is extensive bibliography on these methods such as Moukalled et al. [64] and Ferziger and Perić [65].

The OpenFOAM developers chose to implement an hybrid algorithm called **PIMPLE** [104][105]. PIMPLE results from the combination of the SIMPLE and PISO algorithm. With PIMPLE, both outer and inner iterations are performed for each time step. By doing this, Courant numbers can surpass the unity ($Co > 1$). That is the algorithm used in this work to solve the Navier-Stokes equations in laminar flow (Figure 4.9).

In addition, the PIMPLE routine was implemented by the OpenFOAM developers in a way to take into account turbulence if needed. The incompressible RANS or LES additional equations for obtaining the coefficients of the Reynolds stress tensor (explained in Section 3.3.5) can, therefore, be solved. Moreover, the presence of non-slip boundary conditions presents a challenge in the accurate modeling of the near wall behaviour. Near a wall, the high turbulent stress away from it reduces to values comparable to the viscous stress. Then, if we want to accurately solve the near wall layer then a fine mesh is needed. Low Reynolds turbulence models are capable of simulating this effect with a large number of cells . With high Reynolds this mesh requirement is avoided by implementing **wall functions** which reduces the computational cost. These functions are based on universal flow profiles in the boundary layer along a wall [64]. However, they are not always valid and, for this reason, many wall functions models exist. OpenFOAM provides a vast offer of turbulent models ($k-\epsilon$, $k-\omega$, $k-\omega$ SST, ...) and wall functions. In Appendix A an example for the Space Nuclear Reactor Design using the RANS turbulent model is shown.

4.3.3.2 Porous Medium

For relative complex geometries such as the core of a Pressurized Water Reactor or a Spent Fuel Pool, the resolution of the Navier-Stokes equations in the coolant would require a prohibitive large mesh to capture all the Fuel Assemblies details necessary for the adequate resolution of the equations (grids, fuel rods, nozzles, inlet and outlet structures, etc.). As we have seen in Section 3.3.7 an alternative approach is to homogenize the geometry and replace the heterogeneous system by an homogeneous porous medium. The numerical resolution of the equations of the latter can be performed using a coarse mesh but will require the determination of the more or less complex porous medium properties.

The model presented in Section 3.3.7, based on the mass conservation equations 3.145, the energy conservation 3.146, the Darcy's equations 3.147 and closure equations 3.148-

3.151 developed by Nuske et al. [76], was implemented in OpenFOAM. The approach used the numerical approach presented by Horgue et al. [19]. As for the other thermal-hydraulics models, the porous medium model was coupled to the neutronic models of the multi-physics tool. However as discussed later, the coupling was only in one direction since the studies for the Spent Fuel Pool using this porous medium model did not allow to identify a credible scenario were the SFP reaches criticality and thus fission power is generated. In this section, the general approach of this implementation is presented.

Even though, the model implemented by Horgue only considers incompressible monospecies (one specie in each phase) and two-phase flow in porous medium, some of its features were used in our porous medium model. Note that Nuske's mass equations can be reduced to this model in case of chemical equilibrium with no mass transfer between the phases and negligible diffusion. The features from Horgue's model that were used here are:

- The IMplicit Pressure Explicit Saturation (IMPES) method
- Center to face interpolation
- Boundary Conditions

Firstly, the pressure-velocity equations were solved using the IMPES method [19] since the sequential nature of OpenFOAM, as seen before, forbids us from developing fully implicit solvers for coupled systems. This method is similar to the SIMPLE algorithm where the mass conservation equation is used to solve for the pressure. The difference lies in the fact that the phases' velocity fields are calculated straightforwardly from the pressure fields through the generalized Darcy's model (equations 3.147).

If we use Horgue's model, the mass conservation equation for each α phase can be written as:

$$\phi \frac{\partial}{\partial t}(S_\alpha) + \nabla \cdot V_\alpha = q_\alpha \tag{4.27}$$

while Darcy's model rest the same.

If we replace the velocity field by Darcy's model and sum both equations and include the closure equations 3.148 and 3.151 we obtain the new system:

$$\nabla \cdot \left(-\frac{k_{r,n}K}{\mu_n}(\nabla p_n - \rho_n \vec{g})\right) + \nabla \cdot \left(-\frac{k_{r,w}K}{\mu_w}(\nabla p_n - \rho_w \vec{g} - \nabla p_c)\right) = q_w + q_n \tag{4.28}$$

$$\phi \frac{\partial}{\partial t}(S_w) + \nabla \cdot V_w = q_w \tag{4.29}$$

This new system is then solved implicitly for the pressure, and the saturation is calculated with the explicit value of the last velocity field. The process is iterated until convergence updating the parameters at each step.

A similar system is obtained with the expanded model from this work. The temperature equations are solved treating every term except for the temperatures fields as explicit. The general algorithm can be observed in Figure 4.10.

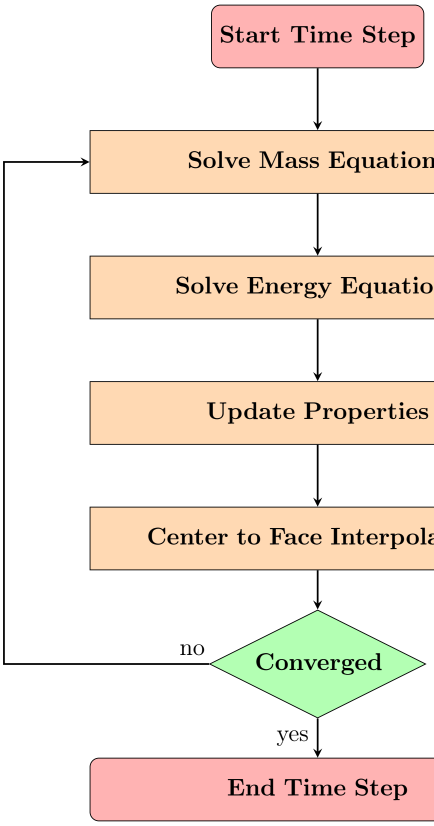


FIGURE 4.10 – Porous Medium Solver.

An important challenge in a porous medium approach is how to treat the interface with a flow region where porosity is equal to one. For example, in the Spent Fuel Pool this would be a region without a Fuel Assembly and filled with water. To treat this type of **sharp interfaces** or also **saturation fronts** OpenFOAM's specific approach was done in the same way as Horgue for his model: all parameters that are not implicitly being solved are interpolated from the center of the mesh cells to the faces of the mesh cells. By doing this, smoother interfaces are formed and the solver becomes more stable. Fortunately, many numerical schemes proposed by OpenFOAM are available at run time (upwind scheme, harmonic average, TVD, etc.).

Lastly, pressure and velocity must be in accord to Darcy's model at every point in the domain, including the boundary. As shown before, the pressure field rather than the velocity field is solved and this implies applying correct boundary conditions for the pressure field. In Horgue's work it can be seen that the corresponding boundary condition for a fixed velocity can be expressed as:

$$\nabla p_n \cdot \vec{n} = \vec{n} \cdot \left[\left(\frac{k_{r,n}K}{\mu_n} + \frac{k_{r,w}K}{\mu_w} \right)^{-1} \left(\mathbf{V_{fixed}} - \left(\frac{k_{r,n}K}{\mu_n}\rho_n + \frac{k_{r,w}K}{\mu_w}\rho_w \right) \vec{g} + \frac{k_{r,w}K}{\mu_w} \nabla p_c \right) \right] \quad (4.30)$$

where $\mathbf{V_{fixed}} = \mathbf{V_{n,fixed}} + \mathbf{V_{w,fixed}}$ is the imposed velocity at the boundary.

For this work, the porous medium solver was used for an heterogeneous system formed by the Spent Fuel Pool from Section 2. In the adopted coupling scheme, the materials density and temperature fields are calculated by the porous medium model in OpenFOAM and then transferred to the Serpent neutronics model to update the materials cross sections. For now, only water density and temperature are updated internally in Serpent. The fields are passed internally and when water material is sampled by Serpent, then the density and temperature are changed accordingly. As we will see later all the analysis has been done without fission power, i.e. only decay heat since a credible criticality scenario was not found. For this reason, Serpent's criticality mode (k-code) is enough. Adding the full coupling (i.e. in the two senses) between the porous medium model and the neutronics module to be able to model an hypothetical criticality accident would be straightforward

4.3.3.3 Multi-Region

One interesting capability of OpenFOAM is the possibility of defining multiple regions. For example, if we want to simulate thermal exchange between a solid and a liquid (Conjugated Heat Transfer) it is possible to define two separated mesh regions and solve the physics models independently at each region. During runtime, coupled solvers are linked by the interface and boundary conditions.

For Monte Carlo, as we said, geometry definition has an independent input. However, the same meshes can be used. Then, internally, all fields are transfer from OpenFOAM to Serpent. In addition, the search algorithm was adapted to find the information necessary (cell position of the desired variable: density, temperature, precursor) from OpenFOAM taking into account the multiple regions.

While this feature was not used in the cases presented in this manuscript, it was implemented in the tool and tested with positive results. This feature will be necessary to study systems containing for instance a coolant in contact to a fuel, a liquid fissile in contact with a solid wall or a coupled neutron reflector region.

Chapter 5

Results

This chapter presents the main results obtained from the multi-physics tool for the three cases of study described in Chapter 2. The main objectives of the analysis of these cases were to investigate the validity of the proposed models, to verify their numerical implementation and to evaluate their performance. When possible, the tool predictions were compared against either experimental data or other codes predictions. As discussed in Chapter 3, the multi-physics tool has been developed from the coupling of two existing codes: the CFD open-source code OpenFOAM and the neutronics Monte Carlo code Serpent 2. A description of the equations and the numerical algorithms of the models of the tool have been presented in Chapters 3 and 4, respectively.

As can be seen in Table 5.1, the three cases described in Chapter 2 were selected because they allow to cover several of the key phenomena identified during criticality accidents. These cases will be therefore very useful to understand the limitations of the models and the codes. The table identifies the key phenomena for each case and the associated models. In addition, as shown in the table, depending on the characteristics of the system being investigated the model may change. For example, the flow model used for the Molten Salt Cavity Benchmark case is the Incompressible Navier-Stokes with Boussinesq approximation model while for the Spent Fuel Pool case we used a porous medium model.

The chapter is organized as follows: firstly, in Section 5.1 the results for the super-prompt burst of the Godiva Experiment are presented. This case was selected since it provides a very fast neutronics transient associated to a thermal expansion phenomenon in a solid homogeneous medium. Secondly, in Section 5.2, the advection of a liquid fissile (and of the delayed neutron precursors) is explored for the case of the CNRS Molten Salt Cavity

Benchmark. This study case provides an example of a slow transient in a power reactor based on a molten salt fuel (i.e. a liquid fissile homogeneous medium). Finally, Section 5.3 presents a heterogeneous Spent Fuel Pool system. This last study case allows investigating the performance of the porous medium model well suited for complex geometries such as the Fuel Assemblies.

TABLE 5.1 – Phenomena and solvers used in the analyzed cases.

Case	Phenomena	Solver
Godiva Experiment	○ Solid Homogeneous Medium ○ Super-prompt Critical Transient ○ Thermal Expansion ○ Geometry Deformation	○ Monte Carlo Quasi-Static Method (Equations 3.67, 3.60c, 3.39a and 3.39b) ○ Simplified P_N (Equations 3.7 and 3.78) ○ Linear Thermoelasticity (Equations 3.99 and 3.101)
Molten Salt Cavity Benchmark	○ Liquid Homogeneous Medium ○ Precursors Convection ○ Laminar Flow ○ Buoyancy ○ Forced Convection	○ Monte Carlo Quasi-Static Method (Equations 3.67, 3.60c, 3.39a, 3.39b) ○ Simplified P_N (Equations 3.7and 3.78) ○ Precursors in liquid medium (Equation 3.7) ○ Incompressible Navier-Stokes with Boussinesq Approximation (Equations 3.110 and 3.126) ○ Energy Equation (Equation 3.134)
Spent Fuel Pools	○ Solid-Liquid Heterogeneous Medium ○ Recriticality	○ Multi-phase multi-component flow in porous medium (Equations 3.145, 3.146 and 3.147) ○ Monte Carlo (Equations 3.1 and 3.6)

5.1 Solid Homogeneous Medium: Godiva Experiment

In this section, the multi-physics tool is used to predict the LANL measured data for the Godiva Experiment super-prompt 29.5 μs burst. This is, as we will see in this section, a very fast transient characterized by a tight coupling between neutronics and thermal expansion phenomena. This study case is particularly interesting to investigate the performance of the Monte Carlo Quasi-Static Method.

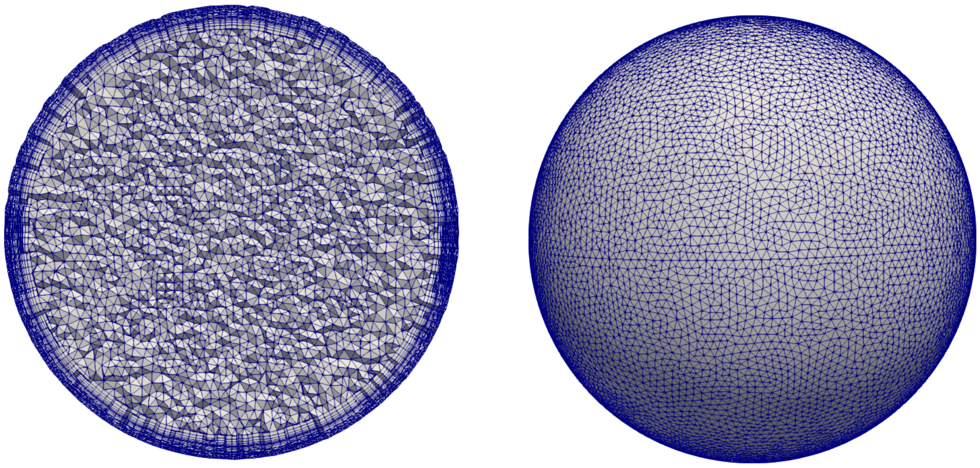
The geometrical data, material properties and numerical parameters used in the simulations of Godiva are given in Section 5.1.1. Since the power burst is induced by a reactivity insertion of just a few cents over prompt criticality, then the statistical uncertainties of the Monte Carlo results have to be carefully investigated as illustrated in Section 5.1.2. In addition, we show in Section 5.1.3 that to take full advantage of the Quasi-Static Method, the point kinetic parameters used in the neutron flux amplitude equations of this method have to be updated inside the flux shape calculations time step (during amplitude time steps). The Monte Carlo Quasi-Static predictions are compared against those obtained from a classic point kinetics model in Section 5.1.5 and also from the results of the deterministic SP_1/SP_3 models in Section 5.1.6. A final comparison between all the models implemented in this work and a pure Monte Carlo transient calculation with Serpent's Dynamic mode by Aufiero et al. [37] is discussed in Section 5.1.7 ending with some remark in Section 5.1.8.

5.1.1 Model

As shown in Section 2.1, the Godiva setup was modeled as an Oralloy sphere, although the real geometry is not a perfect sphere but an ellipsoid. Present capabilities of the codes allow us, if necessary, to solve the physics models (neutronics, thermal-mechanics and thermal-hydraulics) in different meshes and then, map fields from one mesh to the other for coupling purposes (i.e. variables transfer between models). For the Godiva experiment two unstructured meshes were defined. One was used to solve the linear thermoelasticity model described in Section 3.2. The other mesh was used for the neutronic model. The latter mesh was allowed to expand following the solution from the thermal-mechanics model. Even though they were defined independently, both meshes were identical tetrahedral unstructured meshes (Figures 5.1(A) and 5.1(B) shows a cross-cut and a surface view of the mesh) with approximately 7×10^5 cells each.

Material and geometrical specifications for the simplified spherically-shaped Godiva were taken from NEA's benchmark [106]. Uranium isotopes' atomic fraction are shown in Table 5.2. The continuous energy **Nuclear data** library ENDF/B-VII was used to calculate the neutronics properties needed by the different models available in the multiphysics tool. For the SP_3 a three (3) energy groups cross sections scheme was adopted with energy cuts at 0.49787 MeV and 2.2313 MeV[1]. The neutron macroscopic cross sections were condensed and homogenized with Serpent 2 capabilities for energy group constants gener-

[1] Godiva has a fast spectrum and the subdivision into a thermal plus epithermal group and two fast ones proved to be enough for this application (a posteriori) as it is shown in section 5.1.6.



(A) Cross Section

(B) External Surface

FIGURE 5.1 – Tetrahedral mesh for the Godiva experiment model (ca. 7×10^5 elements).

ation, including the Legendre polynomial expansion of the scattering cross sections. The **Mechanical properties** for Oralloy are specified in Table 5.3.

TABLE 5.2 – Uranium istopes atomic fraction [106].

Isotope	Atomic Fraction
U-235	0.9377
U-238	0.0521
U-234	0.0102

TABLE 5.3 – Mechanical properties of uranium [37].

Property	Value	Units
Density (ρ)	18.74	$g\ cm^{-3}$
Poisson's ratio (ν)	0.23	-
Young's modulus (E)	208	GPa
Thermal conductivity (k)	27.5	$W\ m^{-1}\ K^{-1}$
Thermal expansion coefficient (α)	$1.39x10^{-5}$	K^{-1}
Specific heat capacity (c)	117.72	$J\ kg^{-1}K^{-1}$

The system **initial conditions** used in the simulations were set to an uniform 300 K temperature field with zero initial displacement in the solid and no mechanical stresses. In a similar manner as other numerical studies of the Godiva experiments [37][47], power and flux were set to the fundamental mode with a total power of 1 MW and zero delayed neutrons precursors. The main burst characteristics, the power peak and the total energy (analyzed in this section), are not sensitive to them. Setting these initial assumptions are not obvious. The burst peak time depends, on the other hand, on these initial conditions (in particular the power, i.e., the initial neutron count and distribution) which have been adjusted in all these numerical studies to match the experimental data. Indeed, for other experiments the absolute value of the burst peak time could be affected by some significant uncertainties, since once the Oralloy rod is being inserted into the Godiva assembly the initiation of the power burst is actually a random phenomenon and thus the peak time may change for the same experiment setup. Modeling the burst initiation is out of the scope of this analysis [107] and thus we adopt the same initial conditions as other aforementioned numerical studies.

In the NEA work [106], the criticality benchmark for the Godiva Experiment reports a radius of 8.74 cm for achieving criticality in the experiment. Using the data provided in LANL report [8] and assuming that Godiva was a true sphere of radius 8.74 cm then a mass of 52.25 kg would allow reaching criticality. However, as explained in Section 2.1 the Serpent calculations predict a slightly different critical radius (and mass) and thus the system radius had to be slightly adjusted to obtain the desired reactivity. In our study, the spherical system radius was set, by trial and error, to 8.81 cm to achieve a reactivity corresponding to an approximate reactor period of 29.5 μs (which is the value reported in LANL report [10]). Nevertheless, since this radius does not provide the exact reactor period, the fission source was normalized (i.e. multiplied by a scalar factor) in order to obtain the corresponding initial reactor period T_r. Mathematically, the normalization has to satisfy at the initial time t_0:

$$\frac{\rho\left(t_0\right)-\beta\left(t_0\right)}{\Lambda\left(t_0\right)}=\frac{1}{T_r} \tag{5.1}$$

with the parameters defined as

$$H(t_0) = \frac{\alpha_{norm}}{k_0}\left\langle \left(\frac{\chi_p(E)}{4\pi}(1-\beta) + \sum_{d=1}^{G_d}\frac{\chi_d(E)}{4\pi}\beta_d \right) \mathcal{F}\ \phi(\vec{r},\vec{\Omega},E,t_0) \;\middle|\; W_0(\vec{r},\vec{\Omega},E) \right\rangle \tag{5.2}$$

$$\rho(t_0) \;=\; \frac{1}{H(t_0)}\left\langle \Big(-\mathcal{L} \;-\; \mathcal{T} \;+\; \mathcal{S} \Big)\ \phi(\vec{r},\vec{\Omega},E,t_0) \;\middle|\; W_0(\vec{r},\vec{\Omega},E) \right\rangle + 1 \equiv 1 - \frac{1}{\alpha_{norm}} \tag{5.3}$$

$$\Lambda(t_0) = \frac{1}{H(t_0)}\left\langle \frac{1}{v(E)}\phi(\vec{r},\vec{\Omega},E,t_0) \;\middle|\; W_0(\vec{r},\vec{\Omega},E) \right\rangle \equiv \Lambda_0\frac{k_0}{\alpha_{norm}} \tag{5.4}$$

where k_0 is the initial critical multiplication factor (without normalization) and α_{norm} the normalization factor we want to calculate. It can be seen that the effective delayed neutron fraction from equation 3.62c remains the same since the normalization factor appear both in the numerator and the denominator.

Replacing equations 5.3 and 5.4 into 5.1 the normalization factor can be expressed as:

$$\alpha_{norm} = \frac{1}{1-\beta}\left(\frac{\Lambda_0 k_0}{T_r}+1\right) \equiv \frac{1}{1-\beta}\left(\frac{l}{T_r}+1\right) \tag{5.5}$$

where l is the **neutron lifetime** (mean lifetime for one neutron to disappear from the reactor by an absorption or a leak).

As discussed in Chapters 3 and 4, the weight factor used for the normalisation of the neutron shape function in the Monte Carlo Quasi-Static Method was set to unity ($W_0(\vec{r},\vec{\Omega},E) = 1$) since this choice simplifies the determination of the parameters of the amplitude equation. On the contrary, the classical Point Kinetics (PK) formulation (i.e. assuming that the flux shape does not change over the transient), has to be calculated using the adjoint flux as the weighting function.

Concerning the boundary conditions, a vacuum **Boundary Condition** was imposed for the flux and zero gradient for the precursors Boundary Condition (i.e., no loss of precursors through the boundary). At the sphere surface, a fixed temperature Boundary Condition of 300 K was also set with a free surface traction free condition (no stress) for the displacements. At a glance, adiabatic conditions for temperature in the Godiva model seem to be more appropriate, however, in the LANL report [8] it is stated that the "local" temperature of the air near the surface was controlled with a stream of air controlled by heating system.

As discussed in Chapter 4, the scheme for the **coupling strategy** consisted in an explicit coupling: at each time step, the neutronics model equations (SP$_1$/Diffusion, SP$_3$ and the OF-PCQM) are solved to provide the fission power field to the thermal-mechanical model. Then, the energy and displacement equations are resolved with the estimated power. Before the next time step, the neutronics mesh is expanded following the calculated displacement field from the thermal-mechanics module. Then, the Oralloy density field is updated to verify the mass conservation in the deformable mesh. In the OF-PCQM the precursors contributions were not taken into account since they can be neglected during the transient, contrary to the SP$_3$, which automatically includes them in the equations.

As it was explained in Chapter 4, the Quasi-Static Method requires two time steps: a larger one for the neutron flux shape equations 3.66 and a smaller one for the amplitude equations 3.39a and 3.39b. In our calculations for Godiva, the amplitude time step was set to 0.5 μs and as discussed in previous chapters it is the same time step used in the other solvers, such as the solid mechanics solver, as shown in Figure 5.2. For the neutron flux shape equations, different uniform and non-uniform time steps were used and it will be discussed in this section. On the contrary for the SP$_3$ model one unique time step was used and set to 0.5 μs as for the thermal-mechanical model.

For modeling the Godiva experiments it is necessary to accurately take into account the **feedback effects** of the system. Therefore, after density and temperature changes, the neutron macroscopic cross sections have to be updated according to the new density and temperature fields. On-the-fly Doppler broadening routines and S(α,β) interpolation are available in Serpent. However, the hard neutron spectrum (no moderator) and the lack of U-238 resonances due to the highly enriched Oralloy fuel reduce significantly this effect in Godiva. Therefore, cross sections were updated only by the density (ρ_k) changes in the domain and Doppler effects can be neglected.

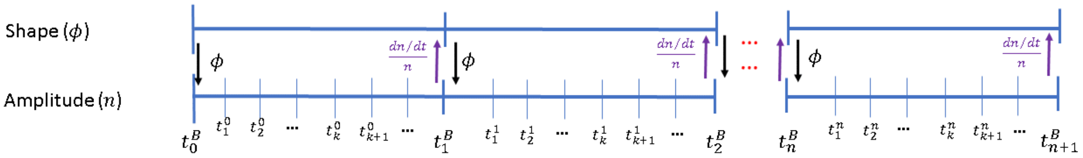

Shape (ϕ)
Amplitude (n)
ϕ
$\frac{dn/dt}{n}$
t_0^B t_1^0 t_2^0 ··· t_k^0 t_{k+1}^0 ··· t_1^B t_1^1 t_2^1 ··· t_k^1 t_{k+1}^1 ··· t_2^B
t_n^B t_1^n t_2^n ··· t_k^n t_{k+1}^n ··· t_{n+1}^B

Figure 5.2 – Time Step Scheme.

From the works of Patricot [47] and Prince [108] about the Quasi-Static Method, it can be seen that, during the flux amplitude iterations, the point kinetic parameters used in these equations have to be updated following the changes of the system properties (density and temperature). This task is straightforward in a deterministic solver, where the cross sections can be immediately updated at each energy group and then point kinetic parameters (equations 3.62) of the amplitude equations can be recalculated considering an interpolated flux. Along these lines, Prince and Ragusa [108] have developed a QSM approach for deterministic methods with temperature feedback using an iterative Improved Quasi-Static Method (IQM). In addition, their iterative IQM formulation has an important advantage which is that the flux shape can be interpolated during the flux amplitude iterations to increase the accuracy and then reduce the number of flux shape calculations.

Unfortunately, the same strategy can not be straightforward implemented in a Quasi-

Static Method with a Monte Carlo code. For example, the update of the point kinetic parameters of the amplitude equations would require saving and processing a large amount of information. This is impractical from two points of view: first, the memory requirement increases proportionally to the amount of neutrons sampled, and secondly, the time to process this information is equivalent to re-running a Monte Carlo flux shape time step. Indeed, recalculating the integrals 3.34b to 3.34f in Monte Carlo would demand access to the continuum cross sections ENDF/B-VII (managed by Serpent). It would also require the storage of all particles with the energy and angular dependencies included. If all neutron particles are stored the time to sum particle by particle with the updated continuum cross section would take as long as the Monte Carlo running time, let alone the memory consumption that would be required.

Moreover, beside these problems the IQM approach would imply jumping back and forth between non-contiguous time steps (recall that many amplitude steps will exist between two shape calculations) which OpenFOAM is not meant for. Implementing the close loop required by the Improved Quasi-Static Method method for the amplitude iterations between two flux shape calculations will then be a complex task for OpenFOAM, which has been developed to run sequentially using an unique time step. Therefore, jumping forth and back between shape time steps would be expensive from a memory point of view and complex to implement (all discretizations should be set to the initial state at the beginning of the shape time step including thermal expansion, which should be reverted).

Based on these observations, it was concluded that while the IQM can definitely increase the accuracy and speed of the Quasi-Static Method for deterministic codes, its implementation in OpenFOAM for the Monte Carlo Quasi-Static Method using Serpent is not practical. As discussed in Chapter 4 an alternative method has been explored here.

It is interesting to note that very recently Kooreman and Griesheimer [109] have proposed an alternative method to update these coefficients that could be implemented in a Monte Carlo based Quasi-Static Method. In their work, a spatially dependant temperature feedback coefficient was calculated during the shape calculation. Then, the system reactivity was updated during amplitude iterations to take into account the temperature feedback. This approach is very interesting but has some drawbacks when used for complex systems as those found in criticality accidents. Indeed, implementing this approach would require many different coefficients to be determined to be able to take into account temperature, density, thermal expansion and, eventually, precursors transport (liquid fuel in the Section 5.2) effects at eventually different positions of the system. This was judged not optimal.

In this work, an alternative more general approach was therefore taken. This approach is based on storing at the end of the flux shape calculation a reduced number of key neutronics variables that allow later to extrapolate during the amplitude iterations all the necessary point kinetic parameters. The stored variables are the reaction rates and the neutron fluxes. Then, while the amplitude equations were being solved, density feedback (and any other feedback) can be implemented by modifying the reaction rates and neutron current fields saved during the shape calculation according to the change of the density $\rho_k(t) = \rho(t)/\rho_0$. For example, the multiplication factor is calculated as:

$$k(t) = \frac{\int \nu\Sigma_f(\vec{r})\ \phi(\vec{r})\ \rho_k^m(\vec{r},t)\,dV}{\int \Sigma_a(\vec{r})\ \phi(\vec{r})\ \rho_k^m(\vec{r},t)\,dV + \int \nabla\cdot\left(\vec{J}(\vec{r})\ \rho_k^n(\vec{r},t)\right)\,dV} \tag{5.6}$$

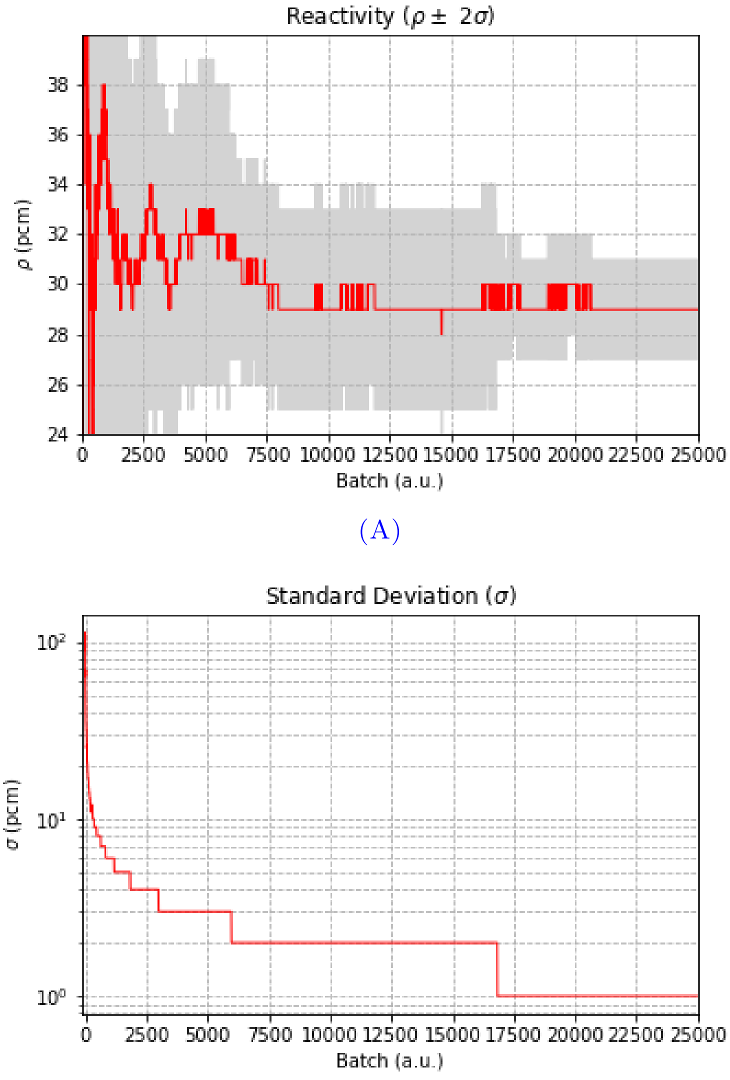
where m and n are set to 1.5 and 0 respectively. Similarly, the other integrals for the amplitude parameters are updated (equations 3.62). In the cases studied in this work the Doppler effect was not important. But, if needed, the reaction rates tallies can be calculated at two different (but close) temperatures in the same shape calculation and interpolation similar to the one applied for the SP$_\text{N}$ method in equation 4.18 could be applied, for example.

As we will see, this more general method provides sufficient accuracy although, contrary to the Quasi-Static Method for deterministic codes, it does not allow to extrapolate the flux shape during the amplitude iteration. Other alternative methods to perform the update of the amplitude equations by extrapolating the information obtained in the previous shape calculation could be considered (such as hybrid approaches) but were not implemented and they will be left for future improvements.

5.1.2 Statistical Tolerance

For our analysis we have chosen the Godiva experiment corresponding to the 29.5 μs burst. According to the LANL Godiva report [8] the super-prompt reactivity of this particular burst experiment was $\sim$ 3.3 cents (¢). Considering a delayed neutron fraction of $\sim$ 640 pcm, this reactivity corresponds to $\sim$ 21 pcm over prompt critical. As a consequence, accurate prediction of the transient requires very small errors on the estimation of the system reactivity: an error of 1 pcm already introduces ∼ 5% of fluctuation.

We can see from Figure 5.3 that, setting 200000 neutron histories per batch, we obtained in our Serpent simulations standard deviations (i.e. 65%) of 4 pcm, 2 pcm and 1 pcm for 2500 batches, 7500 batches and 17500 batches respectively. This means that 3.5×10^9 particles are required in only one shape time step if we want to obtain an standard deviation (σ) of 1 pcm.



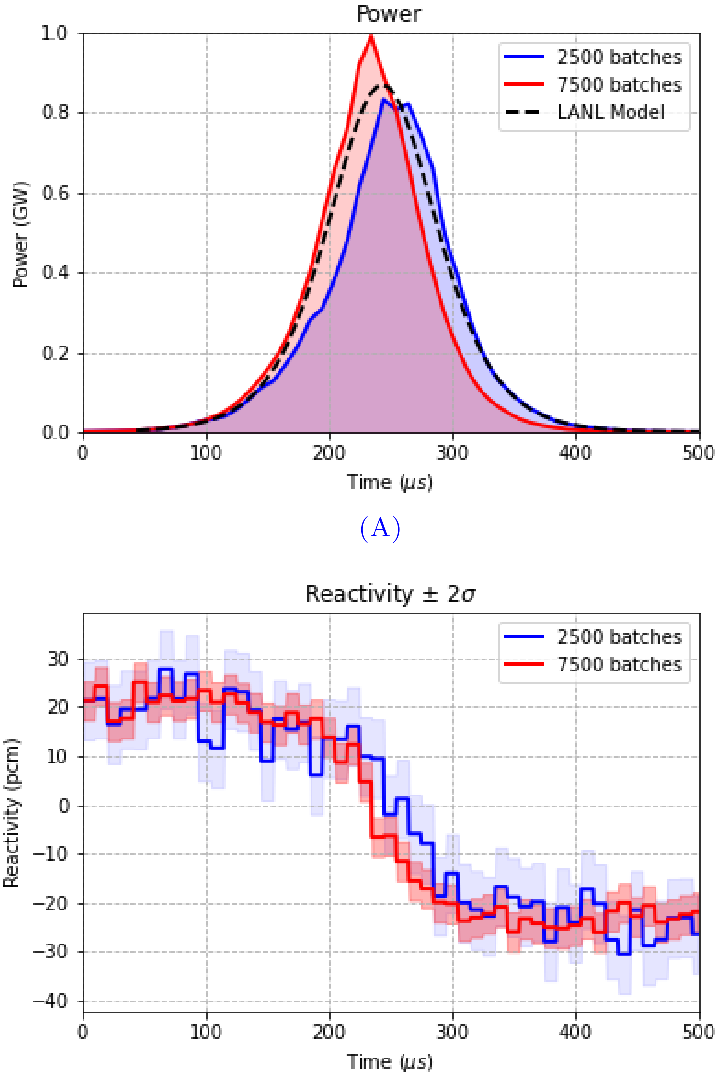
(A)

(B)

FIGURE 5.3 – Monte Carlo convergence error vs batches for : (A) Initial reactivity and (B) Standard Deviation.

In Figures 5.4(A) and 5.4(B) power and reactivity versus time are shown compared to the LANL model. At the beginning of the transient, an accurate estimation of the reactivity mean value is needed in order to have a good normalization of the fission source and to avoid systematic errors or biases afterwards. For this reason, the batches were set to 25000 for the initial shape calculation.

Then, to test to what degree the uncertainties arising from the statistical error of the Monte Carlo method affects the simulation, a comparison between 2500 and 7500 batches was done. The calculations were performed without updating the reactivity (and the other parameters) inside the flux shape time step at each amplitude time step (i.e. the resolution of the amplitude equations). As we can see in Figure 5.4(B) a small variation in the reactivity values can have indeed a significant impact in the predicted burst power as shown in Figure 5.4(A).



(A)

(B)

Figure 5.4 – (A) Power and (B) Reactivity as a function of time for 2500 (blue line) and 7500 (red line) batches with no reactivity extrapolation.

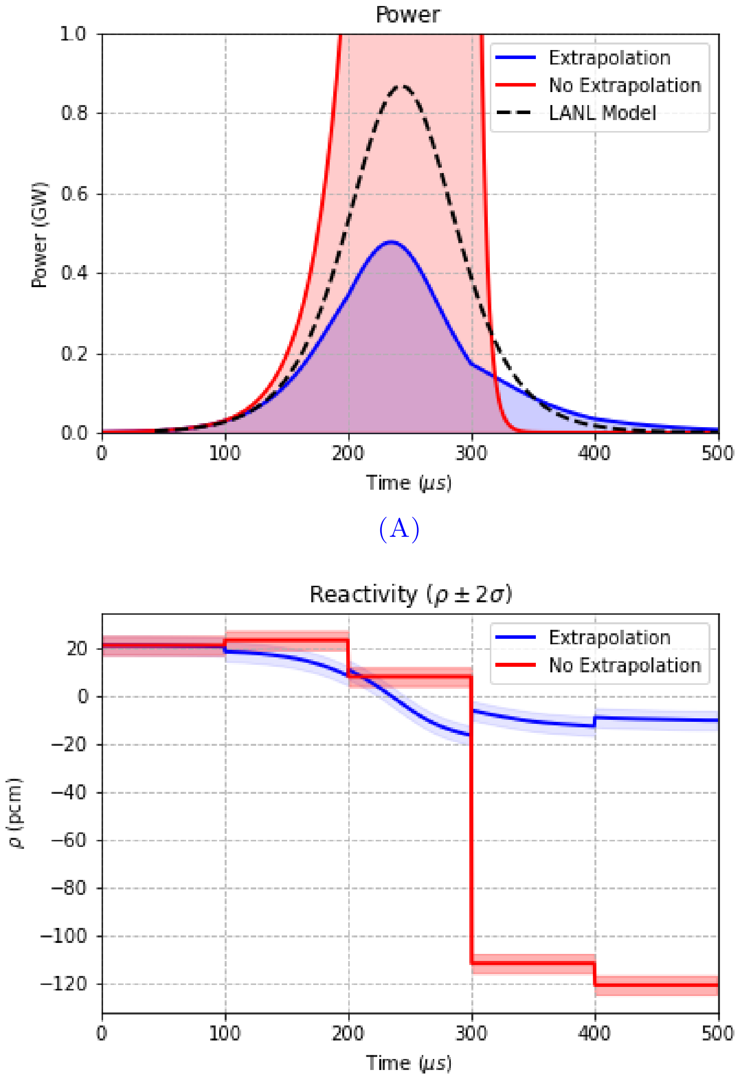
As expected the best agreement on the burst parameters (peak and time) is obtained for the 7500 batches simulation. Nevertheless, the 2500 batches simulation still provides a good approximation for the transient. While the reactivity for the simulation with 2500 batches has a larger dispersion as shown in Figure 5.4(B), one can conclude that if the mean value is close to the experiment the reactivity fluctuations will be smoothed by performing sufficient Monte Carlo flux shape calculations. In other words, one could expect that a higher reactivity uncertainty can be compensated by performing more often the flux shape calculations, i.e. an optimal value for the number of batches should be chosen together with the size of the shape time step.

5.1.3 Update of the Point Kinetic parameters of the Flux Amplitude Equations

So far, results are good and everything suggests that improving the results can be achieved by increasing the accuracy of the Monte Carlo statistics, i.e., more neutron histories, or increasing the frequency of the flux shape calculations. Still, during the transient the neutron flux shape does not change quick enough to justify this unnecessary consumption of resources that Monte Carlo represents. At least, not during the *plateau* at the beginning and the end of the transient.

In an effort to reduce the amount of flux shape time steps, the point kinetic parameters of the amplitude equations were updated during the amplitude iterations using the proposed extrapolation method illustrated by equation 5.6 for the reactivity. This operation demands more memory usage since reaction rates, neutron density and neutron current fields have to be saved at the end of the flux shape calculation and then used to modify the parameters according to the density change. Once again, the absence of resonances (93% U-235) and the hard spectrum makes the Doppler effect negligible for this transient.

In Figures 5.5(A) and 5.5(B) the predicted power and reactivity are presented either with or without the proposed extrapolation method for a very large (100 μs) fixed time step in the shape calculations. As can be seen in the figures, if the point kinetics parameters are not extrapolated (i.e. not updated) a large overestimation of the power peak is obtained. A much better, not yet accurate, power evolution is obtained with the proposed extrapolation methodology although the shape time steps is very large (100 μs) thus making only six flux shape calculations.



(A)

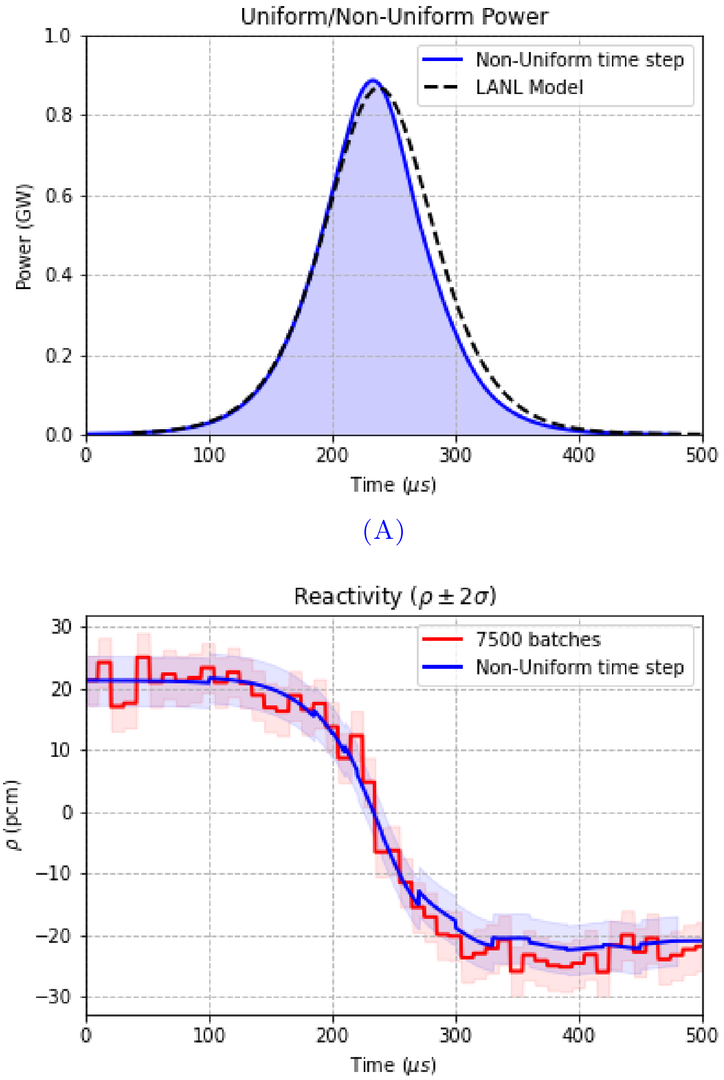
(B)

FIGURE 5.5 – (A) Power and (B) reactivity distribution with and without extrapolation, and uniform time stepping.

5.1.4 OpenFoam predictor-corrector Quasi-statics method (OF-PCQM)

Combining the results of the two previous sections the simulation was further improved by using an adaptive time step for the shape calculations together with the extrapolation of the point kinetics parameters during the amplitude iterations. Results were highly improved as it can be seen in Figure 5.6(A) for the non-uniform prediction. Note that the time steps have been refined around the burst peak. Indeed, the flux shape was recalculated at 0, 100, 185, 200, 210, 220, 240, 270 and 300 μs. After 300 μs some flux shape calculations were made but they can be neglected and the same result will be obtained.

The change of the flux shape in the peak of the burst is significant and, therefore, there is an impact on the reactivity, which can not be corrected solely by the density. This is observed in Figure 5.6(B), which is now similar to recalculating as frequent as the case where the parameters did not were recalculated. From now on we will refer to the Monte Carlo Quasi-Static Method developed here as Monte Carlo QS or OF-PCQM (OpenFOAM Predictor-Corrector Quasi-statics method).

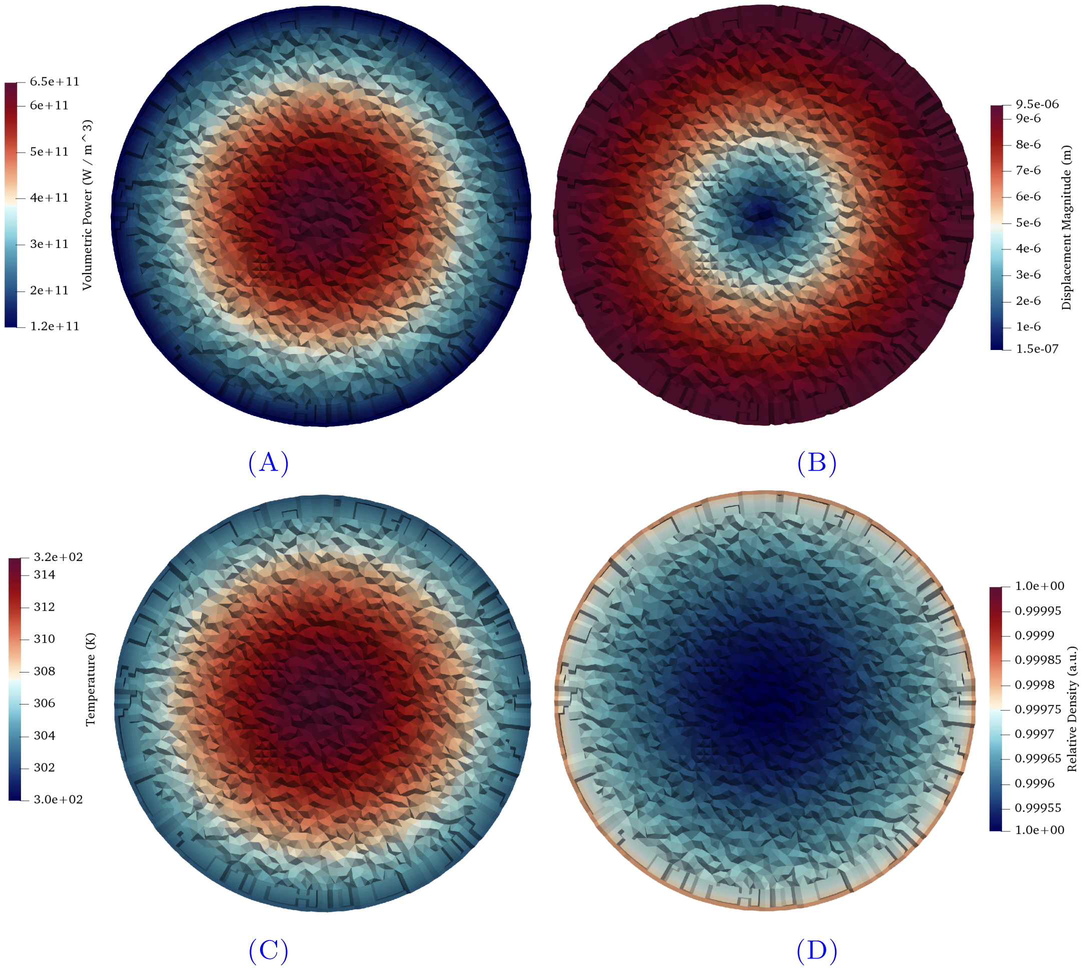


(A)

(B)

FIGURE 5.6 – (A) Power and (B) reactivity distribution with uniform and non-uniform time stepping.

Finally, Figure 5.7 shows the 2D power, displacement, temperature and relative density (ρ_k) fields at the crosscut at the center of the sphere near the power peak time. As can be seen, the system has spherical symmetry and thus it could be modeled in 1D. This was not done since one of the objectives of the present work was to demonstrate the flexibility of the developed tool.



(A)

(B)

(C)

(D)

Figure 5.7 – (A) Power, (B) Displacement, (C) Temperature and (D) Relative Density fields for the OF-PCQM with non-uniform time stepping at $t = 230\mu s$.

5.1.5 Point Kinetics

In this section, we will compare the results obtained from the OF-PCQM with those of the classic Point Kinetics (PK) model. As a reminder the PK model assumes that the neutron flux of the system can be approximated by the fundamental mode and that the fundamental model does not change during the transient, i.e. only flux amplitude is allowed to change. The goal of this comparison is to show the importance of the flux shape change during the transient.

The OF-PCQM model developed here can be used to predict the results of a PK model by making three modifications: (i) only one flux shape is allowed at the beginning of the transient, (ii) the point kinetics parameters of the amplitude equations are not updated (i.e.extrapolated) during the transient and (iii) the point kinetics parameters are calculated using the adjoint of the fundamental model flux as weight function.

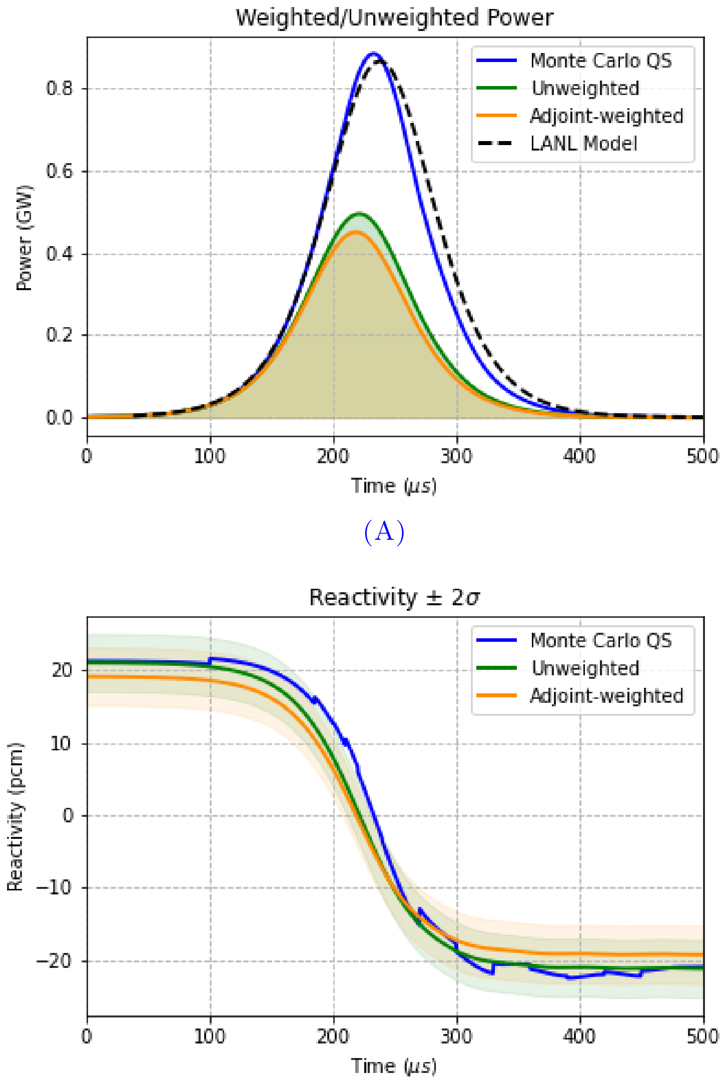
Table 5.4 shows the Serpent-OpenFOAM unweighted (i.e. $W_0\left(\vec{r},\vec{\Omega},E\right) = 1$) and adjoint-weighted (via Iteration Fission Probability) point kinetic parameters calculated in this work and compared them to the TRIPOLI 4 parameters using the Iteration Fission Probability method by Truchet et al. [110] for the initial Godiva sphere. As can be seen, a good agreement exist between Serpent 2 and TRIPOLI. It is also important to remark that the difference between the unweighted and adjoint-weighted is significant.

Table 5.4 – Unweighted and Adjoint-weighted Serpent's kinetic parameters comparison with Truchet et al. [110] calculations with TRIPOLI4 using the Iterated Fission Probability method.

	Unweighted (Serpent 2)	**Unweighted (Truchet et al.)**	**Adjoint-Weighted (Serpent 2)**	**Adjoint-Weighted (Truchet et al.)**
Generation Time Λ $(10^{-9}s)$	6.27	6.25	5.71	5.69
Delayed Fraction β (pcm)	640	641	650	649

In Figure 5.8(A) the power variation predicted by the PK models (using the unweighted and adjoint-weighted point kinetics parameters from Table 5.4) during the transient is compared to the OF-PCQM showing a clear difference between recalculating the shape or not. It is worth that the calculations for the PK models were done with the OF-PCQM using the three approximations explained at the beginning of this section. It is also very interesting to note that the differences between the reactivities predicted by these models are more subtle as shown in Figure 5.8(B). Moreover the differences observed in the initial reactivities of these three models is due to the initial normalization 5.5 required to obtain the system period of 29.5 μs. A smaller generation time Λ implies then a smaller reactivity.

Note that both, the unweighted and adjoint-weighted power transients, are very similar. The unweighted one is slightly higher but still far from the experimental peak value. The main difference relies in that the unity weight adds all contributions to the parameters uniformly whereas the adjoint case uses the importance function, thus, giving more weight to the sphere center. On the other hand, as can be seen in Figure 5.7(D), in the borders the Oralloy density (main feedback mechanism) is still close to its initial value but it is lower at the center of the Godiva sphere.



(A)

(B)

FIGURE 5.8 – (A) Power and (B) reactivity distribution with unweighted and adjoint-weighted parameters in comparison to the OF-PCQM non-uniform time stepping.

5.1.6 Simplified P_3

In this section, we present the results for the Godiva transient obtained from the Simplified P_3 model[2] developed in Chapter 3. We do not present the results of the SP_1 model (i.e. diffusion model) since Godiva is a too small system to be accurate modeled by it. The Simplified P_3 calculations were performed using a 3 energy groups scheme since it gives a good agreement in comparison to the LANL model although it tends to slightly overestimate the power. Results of the comparison are shown in Figure 5.9.

[2] The SP_1 steady-state multiplication factor was not better than the SP_3 model. For this reason, transient studies were not performed for the SP_1 case. Nonetheless, it should be evaluated if a similar transient response to SP_3 is achieved in spite of the poor prediction in the steady-state k_{eff}.

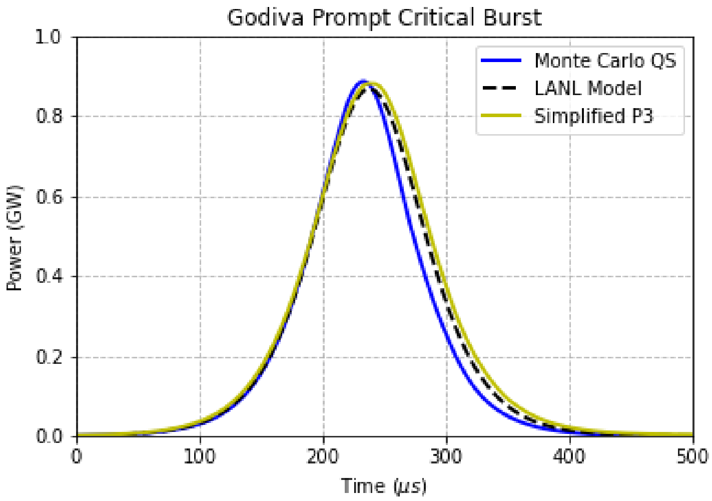


FIGURE 5.9 – Simplified P_3 Power transient for the Godiva Experiment model.

From the results of Figure 5.9, one can conclude that the Simplified P_3 provides an excellent accuracy while having relatively lower computational cost in comparison to the OF-PCQM. However, the prediction of the initial state (reactivity) from this method is far from the accuracy given by Monte Carlo and from the experimental expected radius to achieve criticality as it can be seen in Table 5.5. In this table, the multiplication factor calculated with Serpent 2 with JEFF-3.1.1 and ENDF/B-VII for the Godiva benchmark radius of 8.74 *cm* and the one used in this work (8.81 *cm*) are shown. A bias of around 2500 pcm can be observed for the SP_3 model with respect to Serpent. For comparison purposes, the table also reports the multiplication factor obtained by Patricot [47] using SP_3 implemented in CAST3M code. This independent SP_3 model uses a 9-groups scheme condensed with JEFF-3.1.1. Once the effects of the difference on the radius and libraries has been taken into account, a close agreement between the two Simplified P_3 models is found. This led to think that the SP_3 method has some inherent bias when modeling this type of systems. As already discussed, another challenge of the SP_3 method similar to many other deterministic methods is also the determination of the neutron macroscopic cross section for very heterogeneous systems.

It is interesting at this point to note that, while using SP_3, it was observed that the number of convergence iterations in a single time step penalized the total simulation time (as pointed out by Fiorina et al. [97]). In Figure 5.10 we can see that if no acceleration methods are used the solver needs over 120 iterations per time step. Nonetheless, if the acceleration methods presented in Section 4.3.1.3.3 are used, the amount of iterations can

Table 5.5 – Reactivities calculated with Serpent 2 and the 3-group SP_3 at initial time (temperature and density constant) based on ENDF/B-VII compared to calculations performed by Patricot with the code CAST3M with a 9-groups scheme based on JEFF-3.1.1.

Method	SP_3 (Results Patricot)	Monte Carlo (Serpent 2)	Monte Carlo (Serpent 2)	Monte Carlo (Serpent 2)	SP_3 (OpenFOAM)
Nuclear Data Library	JEFF-3.1.1		ENDF/B-VII	ENDF/B-VII	
Sphere Radius [cm]	8.74		8.74	8.81	
Reactivity (ρ) with delayed neutrons [pcm]	3515	-364	-15	694	3225
Reactivity (ρ) without delayed neutrons [pcm]	-	-1029	-669	38	2600

be rapidly decreased. For this transient, the Anderson Acceleration method [98] with $m = 10$ was used, where m is the amount of previous iterations solutions used to estimate the next one. Both, Aitken and Anderson Accelerations methods embody no physical assumptions. They are extrapolations based on the previous tendency of the system. An effort to use an acceleration that takes into account the physics was done by Fiorina et al. [111] using a reduced order acceleration.

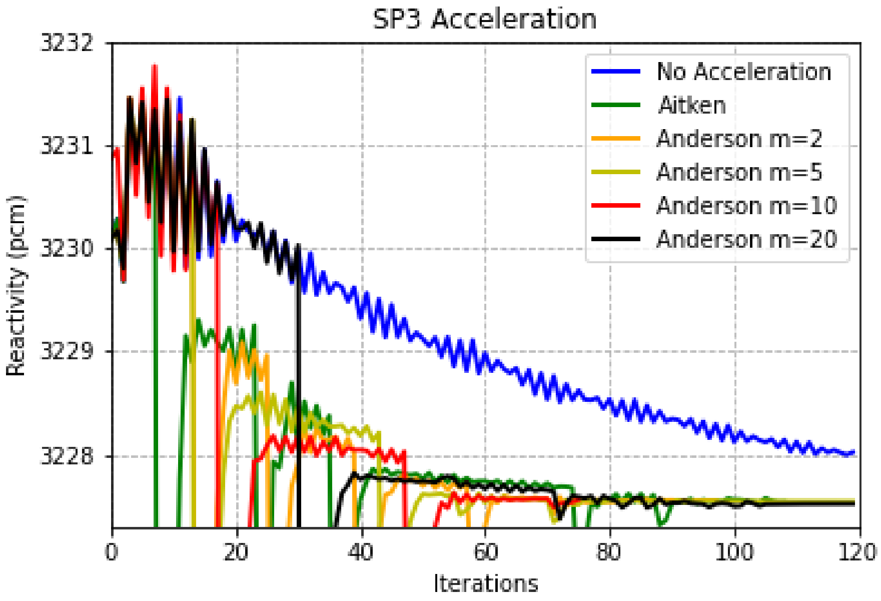


Figure 5.10 – Acceleration methods tested in the SP_3 solver: Aitken method and Anderson method with $m = 2,\ 5,\ 10$ and 20.

5.1.7 Models Comparison

To complete the analysis, in this section we compare the three previous models: (i) the Monte Carlo Quasi-Static Method with extrapolation with a non-uniform shape time stepping, (ii) the Point Kinetics model and (iii) the Simplified P_3 against the LANL experimental data and the transient Monte Carlo with Serpent Dynamic mode from Aufiero et al. [37]. Results are presented in Figure 5.11.

It can be seen from the figure that the Point Kinetics model is the least accurate but it is a fair approximation given its simplicity. Using a feedback coefficient previously calculated, the PK model could be improved as it is in fact done in the LANL semi-empirical model. However, this would imply previous knowledge of the system. Then, the Simplified P_3 and the Monte Carlo Quasi-Static Method are in good agreement with LANL and Aufiero et al. [37].

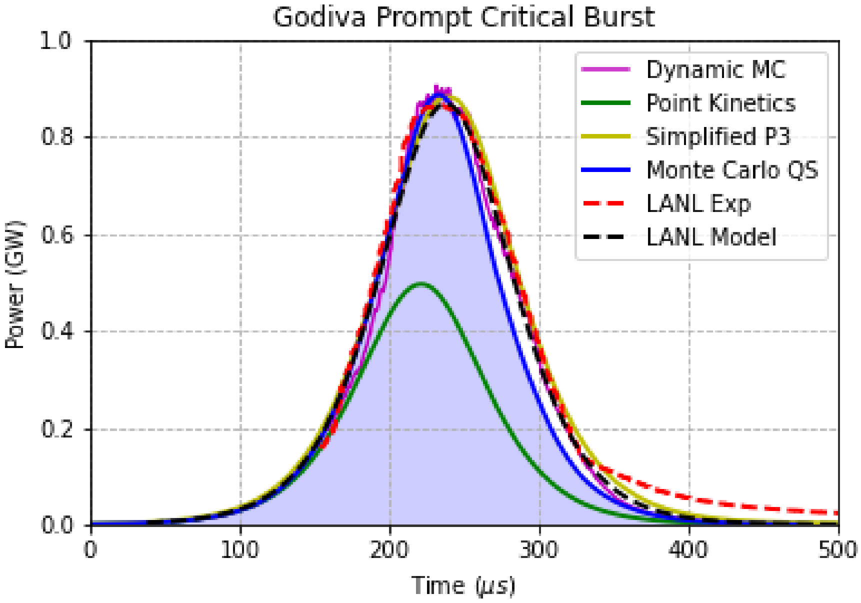


FIGURE 5.11 – Godiva Prompt Critical Burst comparison between: LANL Model, LANL Experimental values, Dynamic MC by Aufiero et al., Point Kinetics, Monte Carlo Quasi-Static Method and Simplified P_3.

One of the objectives of this study case was to evaluate the performance of these different methods. In Table 5.6 a comparison between the key burst parameters is reported regarding: the peak power and the energy released in the system. We can see that all of them are consistent in comparison to the reference LANL model and to the experimental data.

Table 5.7 shows the computational effort of the SP_3 and the Monte Carlo Quasi-Static Method. First, the 2500 and 7500 batches are less demanding in terms of time and memory but precision and smoothness is not as good as for the extrapolated case. However, the latter needs more histories. This is because an smoother distribution in the mesh is needed to perform mathematical operations on it (such as calculating a flux gradient to estimate

Table 5.6 – Power Peak and Total Energy deposited in the system for the LANL Model, Monte Carlo Quasi-Static Method (OF-PCQM), the Simplified P_3 and the Dynamic Monte Carlo (Aufiero et al. [37]).

	Power Peak ($x10^{-1}GW$)	**Energy** ($x10^{1}kJ$)
LANL Model	8.7	10.2
OF-PCQM	8.9	9.5
SP_3	8.8	10.9
Dynamic MC	9.0	10.0

the leaks). Moreover, less statistical error on the reactivity avoids biases on the solution. The Simplified P_3 time evolution is similar to the OF-PCQM extrapolation case. The main issue with this method is the lack of predictability for the reactivity at the beginning of the transient. In the steady-state the value was more than 3500 pcm off (see [47]) in comparison to Monte Carlo calculations for the same radius. For this reason, the fission source had to be normalized at the beginning of the simulation to achieve the desired reactor period/ initial reactivity. Lastly, in the work by Aufiero et al. [37] the reported calculation time is 11 hours in an 8 cores Intel(R) Xeon(R) CPU E5-2670 @ 2.60GHz for a total of 2000 time steps (10^{-3} s). This result is less than what we obtain with Monte Carlo Quasi-Static Method. It is understandable if we compared the amount of particles needed. For Aufiero et al. it is 1.25×10^7 particles while in this work is almost 10^4 times more. Nevertheless, OF-PCQM demands much less memory and it can be easily implemented with any Monte Carlo criticality code with an external coupling scheme.

Table 5.7 – Computational Cost for cases calculated in an Intel(R) Xeon(R) Gold 5118 CPU @ 2.30GHz using 24 CPUs.

Case	**Shape Steps**	**Batches per Shape Step**	**Histories per batch**	**Total Histories Simulated** ($x10^{10}$)	**Computational Time (hours)**	**Serpent 2 Memory (MB)**
2500	50	1st 25000, then 2500	200000	2.95	54	2705
7500	50	1st 25000, then 7500	200000	7.85	144	2706
Non-Uniform Extrapol	9	50000	200000	9.00	165	2718
SP_3	Every Step	-	-	-	15	-

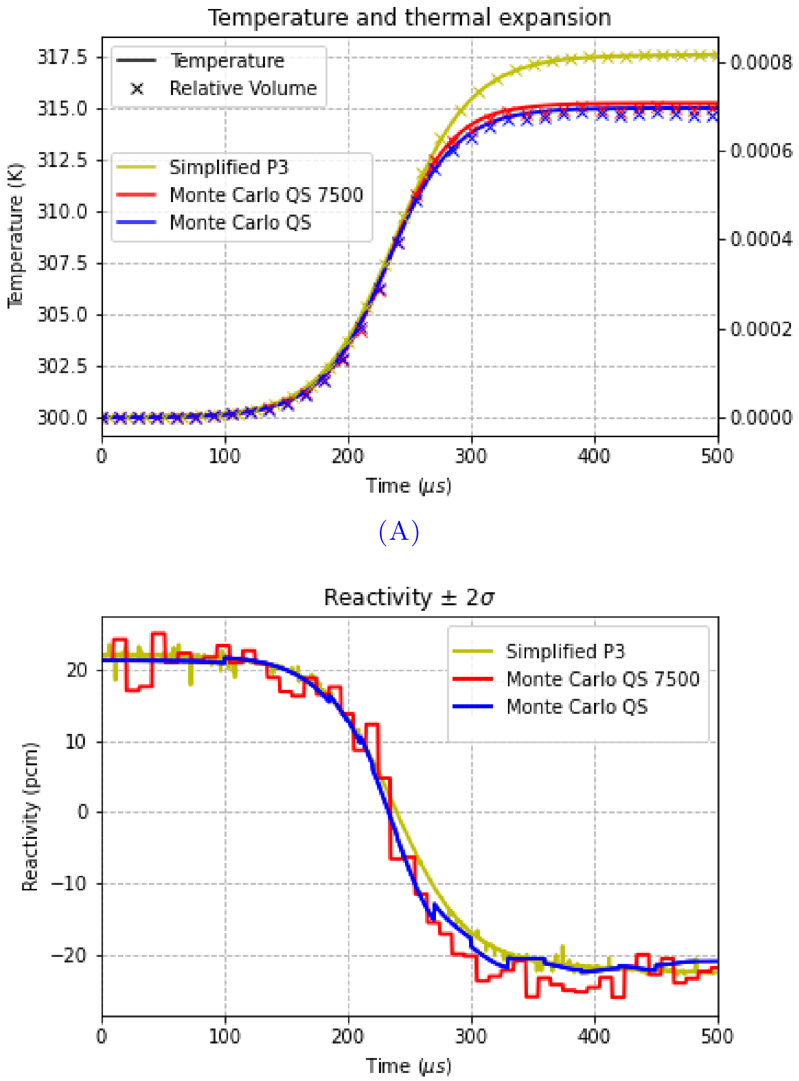
At this point, it is worth mentioning that we have not focused the effort on optimizing the computational cost of the Monte Carlo Quasi-Static Method and several important improvements could be made. The value reported in the table is then a conservative estimate. If a strategy similar to Aufiero et al. [37] is adopted a reduction on the computational cost could be obtained. In Aufiero's work, the Monte Carlo code does not read the unstructured mesh to form the geometry, it rather defines a sphere bigger than the initial radius and assign a zero density to the excess volume. As it expands, the density value is updated internally from the mesh-based variables on-the-fly in the tracking routine. In this work, to keep the tool as general as possible, the multi-physics solver geometry is updated via the unstructured mesh and the Monte Carlo geometry is divided in cells assigned with individual materials even if it is the same. The time penalisation of this should be studied.

Figure 5.12 displays the mean temperature, reactivity and total thermal expansion values during the transient. As it can be seen, the SP_3 reactivity value is slightly higher than Monte Carlo during the peak and, therefore, temperature differs about 2.5 K due to the wider power burst. The sphere expansion follows the same tendency. Initial and final reactivities are similar.

Finally, Figure 5.13 shows the flux radial shape calculated from the two methods developed in this work: the Monte Carlo Quasi-Static Method and the SP_3. It can be observed that the SP_3 overpredicts the flux value near the borders of the sphere. This illustrates why the Marshak boundaries conditions should be considered as an approximation. Even thought the curves are slightly similar, the offset produces an overestimation of the multiplication factor and without a proper normalization of the fission source the 29.5 μs transient would not be obtained.

5.1.8 Final Remarks on the Godiva Study Case

A good agreement between the Monte Carlo Quasi-Static Method, the Simplified P_3 methods, LANL experimental data, LANL model and Dynamic Monte Carlo was observed. It was found that the Monte Carlo Quasi-Static Method is accurate enough for a fast transient even using its adiabatic approximation. Ott concludes in [42] that the adiabatic approximation predicts well enough the reactivity value but there is up to a 30% error on the flux shape. From the results presented in this section, we can conclude that it is correct to state that when solving the eigenvalue problem of equation 3.66 the real problem is not being solved but a similar one is and thus an error may be introduced. Moreover, this analysis is true for our system but the overall error is very small. Indeed, Ott used in his analysis a system with a 90$/s reactivity insertion. In the Godiva Experiment the prompt reactivity is just a few cents, so, even if this introduces an error, it has a negligible impact



(A)

(B)

Figure 5.12 – (A) Temperature and Thermal Expansion, and (B) Reactivity for the Monte Carlo Quasi-Static Method and Simplified P_3.

on the flux shape. Moreover, we have seen from the comparison with SP_3 in Figure 5.13 that the shapes obtained with both methods are very similar.

The main advantage of the OF-PCQM lies in its simplicity. If no extrapolation is used (Figure 5.4(A)), any Monte Carlo code can be coupled externally with a thermal-hydraulics/thermal-mechanics code given that the power distribution and amplitude parameters integrals from equation 3.62 can be tallied and information on the density and temperature can be passed to the Monte Carlo code. This is the case of Serpent 2, thus resulting in a less invasive implementation. If mesh-wise distributions can be saved, then, shape time steps can be enlarged (less shape recalculations) by implementing the parameters extrapolation. In this work, it was done internally because it is faster: read/write of

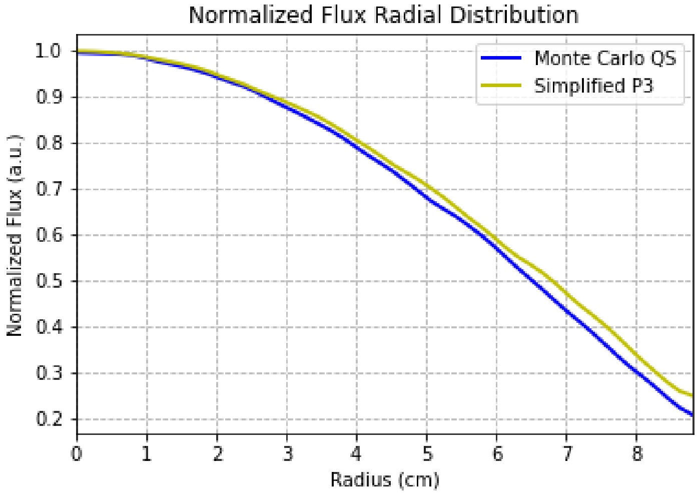


Figure 5.13 – Normalized flux comparison at the peak between Monte Carlo Quasi-Static Method and Simplified P_3.

external files is avoided and search algorithms are quicker.

Another advantage of the proposed method is that a Monte Carlo provides flexibility in geometry capabilities modeling. In particular, Serpent is adapted to read the unstructured mesh and fields used by OpenFOAM. On the other hand, the Quasi-Static Method permits different time scales to be modeled without wasting computational effort in recalculating the full transport equation at every step. Then, the ratio between both time steps (shape and amplitude) can be adjusted. An adaptive time stepping can be implemented as in the code QX1 developed by Argonne National Laboratory [112]. Time control routines based on the QX1 code were implemented in OpenFOAM and discussed in Appendix B but due to the lack of time it was not tested in any of the cases presented in this work. Its adaptive capabilities are available but still to be tested. Finally, when comparing the OF-PCQM against the Dynamic Monte Carlo, one has to take into account that the memory consumption of the OF-PCQM is much less than Monte Carlo in dynamic mode. This is explained by the fact that criticality calculations during the shape steps are independent between each other and very few parameters are saved and the rest can be freed from memory.

Concerning the SP_3 method, it failed to accurately predict initial reactivity. However, once the fission source was normalized to the desired reactivity insertion, this method is capable of following the transient and has a lower computational cost. Note that the SP_N method reduces the amount of equations from $(N+1)^2$ in PN to $(2N+1)/2$ in SP_N. For this reason, it is considered a fast deterministic method. In this work, an Anderson acceleration [98] was used to improve the convergence rate of the fixed-point iterations. As we seen, the computational cost of SP_3 is significantly lower than the OF-PCQM. However, the first has been optimized by using an acceleration technique. As earlier discussed, there is a lot of room for decreasing the computational cost of the OF-PCQM.

As a final remark, the initial state was considered as the fundamental mode at a power of 1 MW. It does not affect the results because the neutron generation time is very short ($10^{-9}s$) and when the feedback begins to affect the transient many generations have already passed (time step of $0.5\mu s$). This is also why the adiabatic method works fine: there is enough time for the approximate fundamental mode of the flux to be established with the system temperature and density at that time. In other words, the fundamental mode is established faster than the systems changes.

5.2 Liquid Homogeneous Medium: Molten Salt Cavity Benchmark

In this section we will present the results obtained for the second study case: the Molten Salt Cavity Benchmark. This benchmark was designed [12] to study the performance of the different couplings of a multi-physics model. It is based on a simple 2D system containing a molten fuel salt with strong coupling between neutronics and thermal-hydraulics phenomena. For our purposes, this case will let us study the performance of the multi-physics models with respect to other phenomena such as the precursors convection, laminar flow, buoyancy effects and forced convection to mention the most important ones. Then, the Molten Salt Cavity Benchmark is divided in various phases. A summary on the conditions for each step in each phase is presented in Table 5.8.

In the first part of this section, the steady-state results of **Phase 0** and **Phase 1** of the benchmark corresponding to the single physics (steps 0.1, 0.2 and 0.3) and the coupled problem (steps 1.1, 1.3, 1.5 and 1.7) for a thermal power of 1 GW and a top wall velocity of 0 ms^{-1} and 0.5 ms^{-1} are presented. The second part of this section is devoted to the transient analysis due to an oscillatory volumetric heat exchange coefficient (**Phase 2**). Considering the system as a black box, the gain and phase change from the heat exchange to the power was analyzed.

Table 5.8 – CNRS Molten Salt Cavity Benchmark summary of the steps involved [12][113][14].

Phase	Step	Velocity	Temperature	Power/Fission Rate	Objective
0	**0.1**	Top-wall forced convection $U_x = 0.5\ m\ s^{-1}$	N/A	N/A	Hydraulics Stand-Alone Calculations
	0.2	Null Velocity Field $\vec{v} = \vec{0}$	Uniform 900 K	Calculated Normalized to 1GW	Neutronics Stand-Alone Calculations
	0.3	Fixed from Step 0.1	Calculated $h_{sink} = 10^6\ WK^{-1}m^{-3}$	Fixed from Step 0.2	Temperature Stand-Alone Calculations
1	**1.1**	Fixed from Step 0.1	Uniform 900 K	Calculated Normalized to 1GW	Neutronics-Circulating Fuel Coupling (Precursors Transport)
	1.2	Fixed from Step 0.1	Calculated $h_{sink} = 10^6\ WK^{-1}m^{-3}$	Calculated Normalized to 1GW	Thermal-hydraulics- Neutronics Coupling (Precursors transport + density reactivity feedback)
	1.3	Calculated without Forced Convection $U_x = 0\ m\ s^{-1}$	Calculated $h_{sink} = 10^6\ WK^{-1}m^{-3}$	Calculated Normalized to 1 GW	Full-coupling: Precursors Transport Density Feedback Buoyancy
	1.4	Calculated with Forced Convection $U_x = 0.5\ m\ s^{-1}$	Calculated $h_{sink} = 10^6\ WK^{-1}m^{-3}$	Calculated Normalized to 1 GW	Full-coupling
2	**2.1**	Calculated $U_x = 0.5\ m\ s^{-1}$	Calculated Oscillating Heat Sink $h(t) = h_{sink}(1 + sin(2\pi t/T))$ $h_{sink} = 10^6\ WK^{-1}m^{-3}$ $T = 1.25, 2.5, 5, 10, 20, 40, 80s$	Calculated	Full-coupling Step 1.4 is the Initial Condition

Both the steady and the transient calculations were performed using the SP_1/SP_3 and Serpent-OpenFOAM coupled models such as the OF-PCQM presented in the previous section. The results obtained for the cavity benchmark with SP_1/SP_3 were included in the work published by Tiberga et al. [14], obtaining good agreement among the partners of the project. In the cited work the results from members of European project SAMOFAR were compared. In addition, the SP_3 model was also developed in Python using FiPy [114], which is a Finite Volume Method PDE solver library in Python developed by the National Institute of Standards and Technology (NIST) in the USA.

The OpenFOAM models used for the results of this section have a 200×200 structured mesh (Figure 5.14) refined progressively towards the walls while the FiPy model uses a 115x115 structured mesh regular inside the cavity and refined near the walls.

Material and geometrical data was obtained from [12][113] in agreement with Tiberga et al. [14]. The salt composition is presented in Table 5.9 and its properties in Table 5.10. JEFF-3.1.1 nuclear data library was used for the Monte Carlo calculations. For the multi-group SP_3 calculations a 6 energy-groups macroscopic cross sections were taken from the cited bibliography that were condensed from JEFF-3.1 with energy cuts at 7.485×10^{-4}, 5.531×10^{-3}, 2.479×10^{-2}, 4.979×10^{-1} and 2.231 MeV.

Concerning the boundary conditions, vacuum **Boundary Conditions** were used for the SP_1/SP_3 and Monte Carlo models. The system walls were considered as adiabatic, i.e.

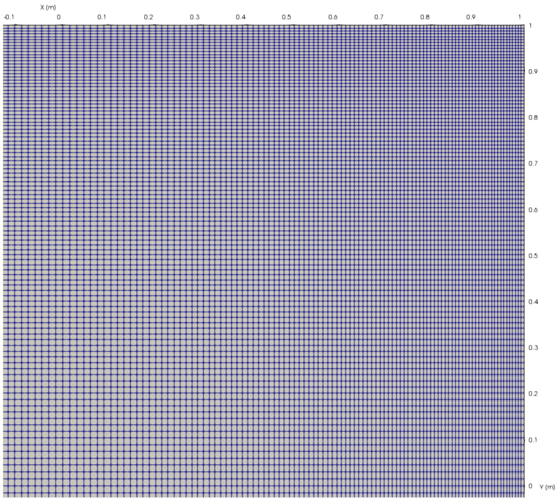


FIGURE 5.14 – Quarter of the 200x200 Structured Mesh for the lid-driven cavity.

$\nabla T \cdot \vec{n} = \vec{0}$ with no-slip velocity conditions ($\vec{v} = 0$) at the walls. The upper wall velocity was set to either 0 m s^{-1} or 0.5 m s^{-1} depending on the benchmark step (see Table 5.8). Note that in this system the delayed neutron precursors can not leave the geometry, i.e., $\nabla C_d \cdot \vec{n} = \vec{0}$ and thus we verify the hypothesis of the neutronics models.

TABLE 5.9 – LiF-BeF2-UF4 composition.

Isotope	**Li-6**	**Li-7**	**Be-9**	**F-19**	**U-235**
Atomic Fraction [%]	2.11488	26.0836	14.0992	56.3969	1.30545

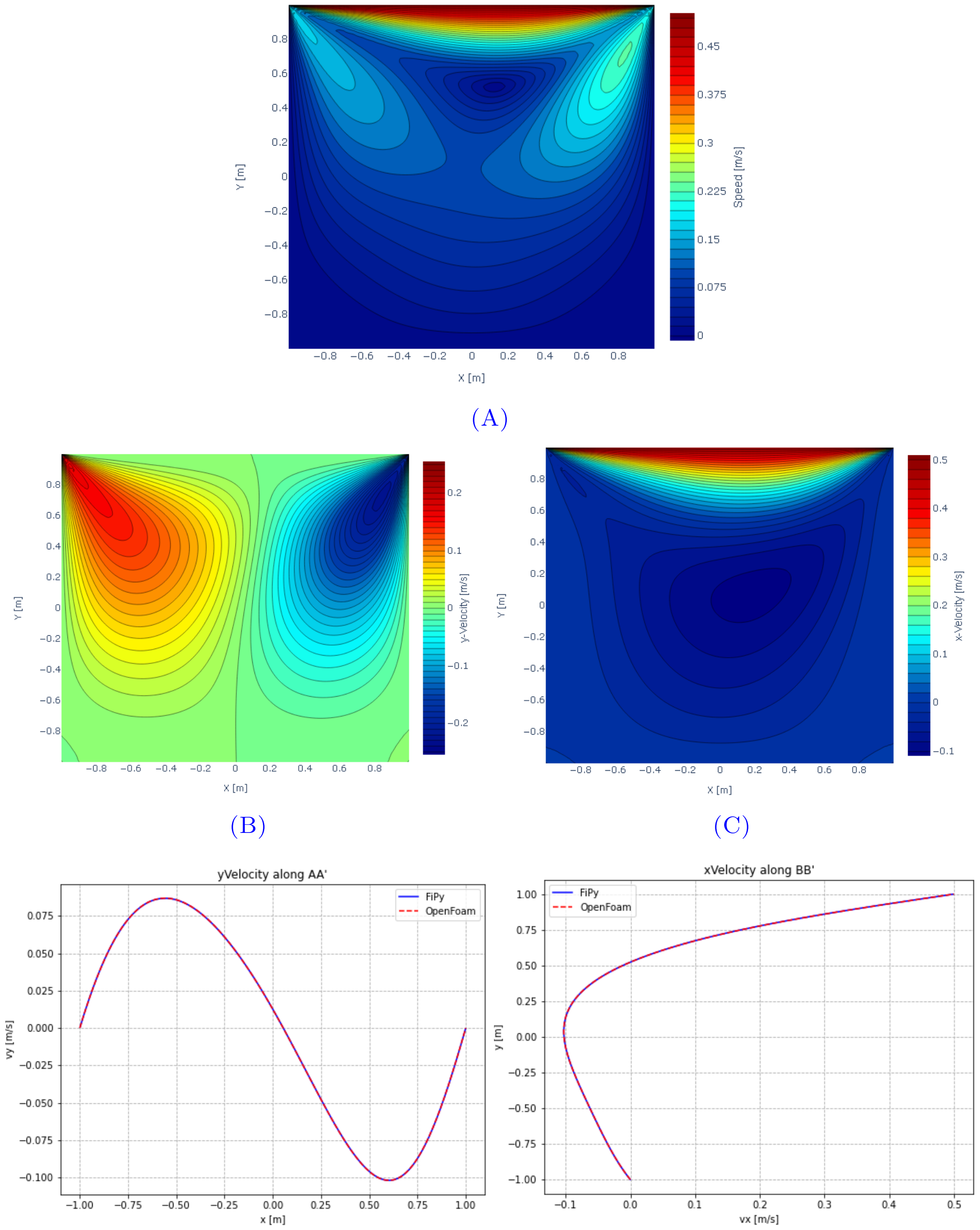
TABLE 5.10 – Salt properties.

Density (ρ)	2.0 $g\ cm^{-3}$
Kinematic Viscosity (ν)	0.025 $m^2 s^{-1}$
Volumetric Heat Capacity (C)	6.15x10^6 $J\ K^{-1} m^{-3}$
Prandtl Number (Pr)	3.075x10^5
Schmidt Number (Sc)	2.0x10^8
Thermal Expansion Coefficient (α)	2.0x10^{-4} K^{-1}

5.2.1 Steady-State

Single Physics

As explained in Section 2.2, **Step 0.1** consisted in the fluid mechanics stand-alone calculations. Figures 5.15(A), 5.15(B) and 5.15(C) show the speed, x-Velocity and y-Velocity inside the 2D cavity with a forced convection set by a 0.5 ms^{-1} imposed at the top wall. In Figures 5.15(D) and 5.15(E) a good agreement between FiPy and OpenFOAM is observed.



(A)

(B)

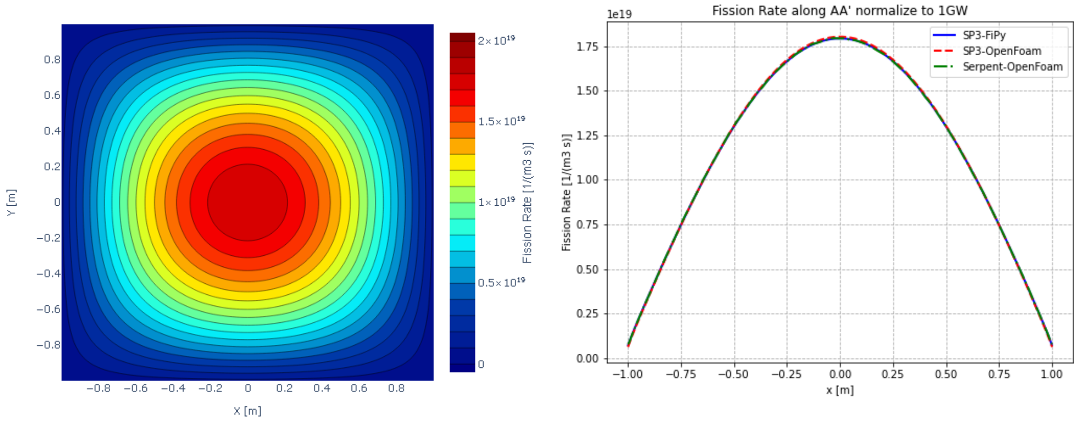
(C)

(D)

(E)

Figure 5.15 – Step 0.1: Velocity (A) Magnitude, (B) Y-component, (C) X-component, (D) Y-component along AA' and (E) X-component along BB' calculated in FiPy and OpenFOAM.

In **Step 0.2** the neutronics stand-alone calculations for the SP_1, SP_3 and Monte Carlo were compared. The fission rate distributions in Figure 5.16(B) show excellent consistency among all models. The SP_1 model is not plotted since it overlaps with the SP_3-OpenFOAM model. However, the reactivity obtained is reported in Table 5.11 along with the other models also demonstrating an excellent agreement. One can note that as the system is relatively large, SP_1 provides very good results. Difference between the SP_3 model implemented in FiPy compared to the SP_3-OpenFOAM could be due to the mesh size and implementation of the Marshak Boundary Conditions (equations 3.81).



(A) (B)

Figure 5.16 – Step 0.2: Fission Rate (A) inside the cavity and (B) along AA' for SP_3 (in FiPy and OpenFOAM) and Serpent-OpenFOAM internal coupling models.

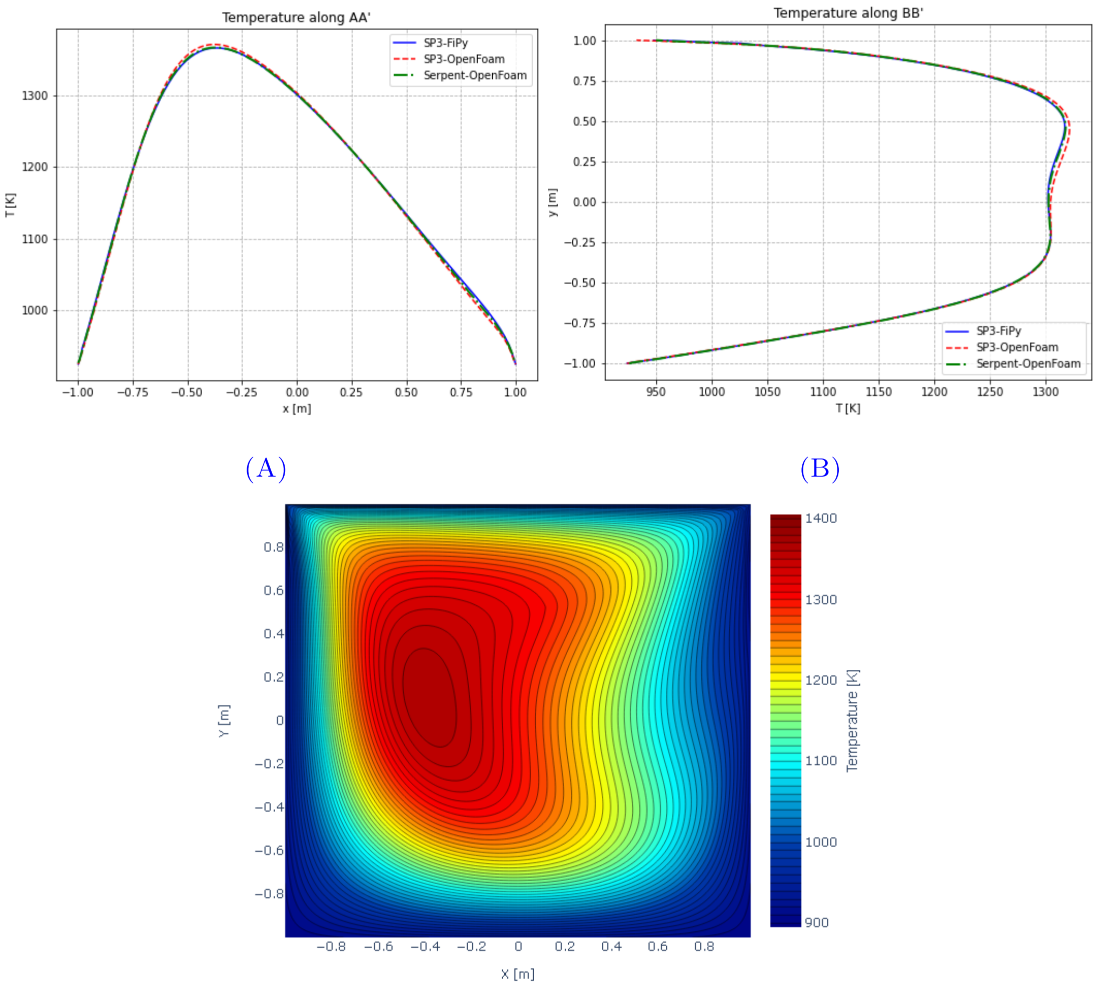
Table 5.11 – Step 0.2: Reactivity values comparison for the neutronics models (SP_3 in FiPy, SP_1 and SP_3 in OpenFOAM and Serpent-OpenFOAM internal coupling).

Model	**Reactivity ρ [pcm]**
Serpent-OpenFOAM	402
SP_1-OpenFOAM	411
SP_3-OpenFOAM	354
SP_3-FiPy	405

Step 0.3 is the last of the single physics calculations. It can be observed from Figures 5.17(A) and 5.17(B) the results from the different models practically overlaps each other. There is less than 5 K temperature difference between the models. When the advection of enthalpy due to the circulating fuel is not present, the temperature distribution should resemble the power distribution proportional to the fission rate distribution from Figure 5.16. However, the advective phenomena distorts the temperature field.

Coupled Physics

The first coupled scenario **step 1.1** assess the changes in the criticality calculations considering the fuel motion set by the velocity field from **step 0.1**. The circulating fuel transports the precursors away from where the fission took place and distorts the distribution. Then, the delayed neutrons are born in a position different from where the precursor



(A)

(B)

(C)

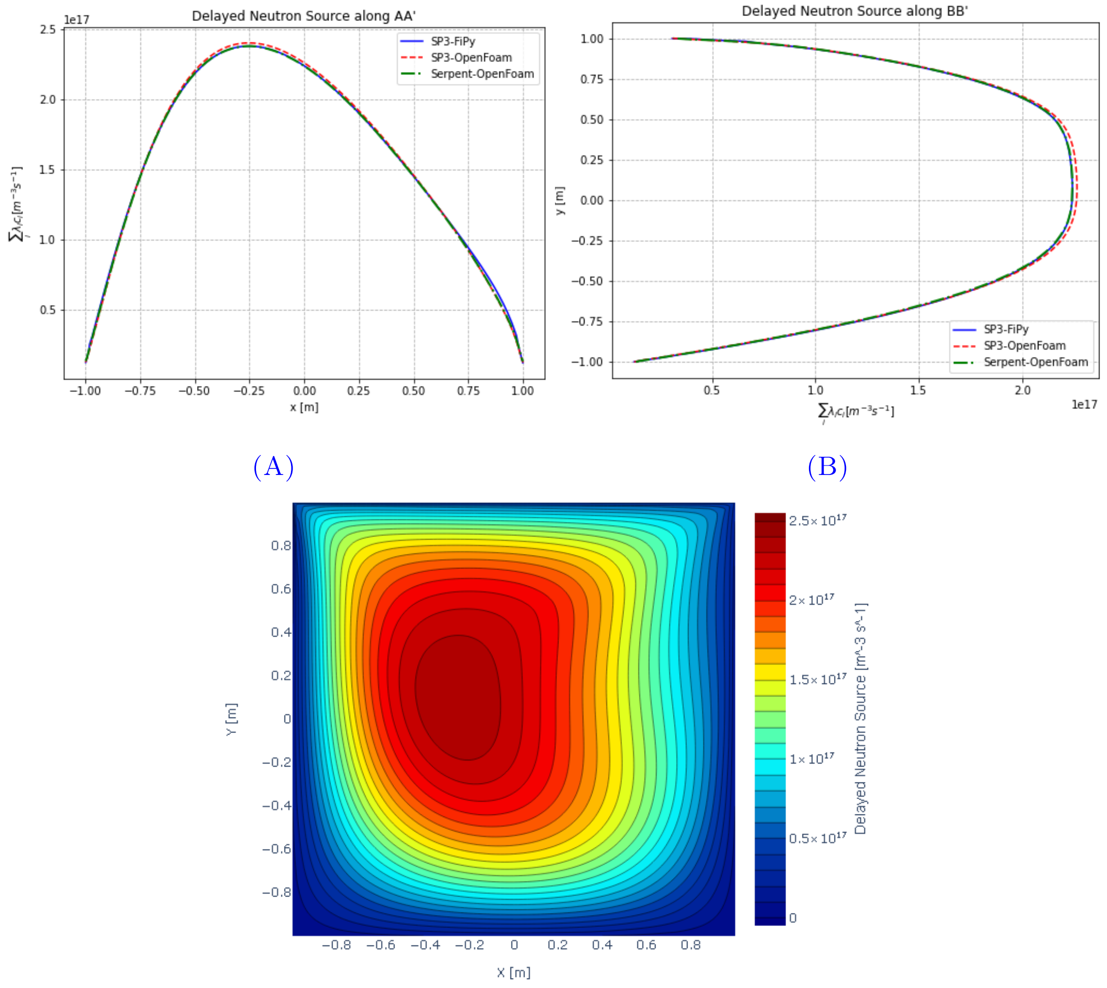
FIGURE 5.17 – Step 0.3: Temperature (A) along AA', (B) along BB' and (C) inside the cavity for SP_3 (in FiPy and OpenFOAM) and Serpent-OpenFOAM internal coupling models.

was born, thus, affecting the reactivity of the system. This results in a lower reactivity as observed in step 0.2 once again in Table 5.12. It seems that, on average, neutrons are taken away from the center to borders, this is, from zones with bigger importance to zones with lower importance.

TABLE 5.12 – Step 1.1: Reactivity values comparison for the neutronics models (SP_3 in FiPy, SP_1 and SP_3 in OpenFOAM and Serpent-OpenFOAM internal coupling) against neutronics stand-alone results from Step 0.2.

	Reactivity ρ [pcm]		
Model	**Step 0.2**	**Step 1.1**	**Difference**
Serpent-OpenFOAM	402	338	-65
Diffusion/SP_1-OpenFOAM	411	349	-63
SP_3-OpenFOAM	354	291	-63
SP_3-FiPy	405	345	-60

Once again, the resemblance between models for the delayed neutron source in Figures 5.18(A) and 5.18(B) is good. Nonetheless, the SP$_3$-OpenFOAM reactivity is once again 60 pcm smaller than the SP$_3$-FiPy. However, the cause of this difference seems to be systematic as the change of reactivity from step 0.2 to step 1.1 is similar for all models. This issue is not observed in the FiPy model, pointing to a possible implementation difference of the Marshak Boundary Conditions in OpenFOAM.



(A)

(B)

(C)

FIGURE 5.18 – Step 1.1: Delayed Neutron Source (A) along AA', (B) along BB' and (C) inside the cavity for SP$_3$ (in FiPy and OpenFOAM) and Serpent-OpenFOAM internal coupling models.

In the next step the neutronics - thermal-hydraulics coupling was tested (**Step 1.2**). In addition to the circulating fuel, density feedback was considered due to the power generation in the cavity. Heat is evacuated through a volumetric sink with a convection coefficient (h_{sink}) as explained before. Temperature and fission rate difference over the line

AA' (crosscut showed in Figure 2.3(A)) respect to Step 0.2 ($[\Sigma_f\phi]_{step1.2} - [\Sigma_f\phi]_{step0.2}$) can be observed in Figure 5.19. The fission rate in this step is shifted to the right in comparison to step 0.2, where the fuel is at lower temperature and, therefore, a higher density. There is a very good agreement between all models.

The fission rate difference for the Serpent-OpenFOAM coupling has a noise around the mean value due to the stochastic nature of this method and the small mesh size ($\sim 1cm$). The tallies volumes (i.e. each cell) are small and the amount of scores arriving to each cell would need to be increased in order to reduce the noise in the distribution or the cell size increased with the accuracy cost it implies. However, Figures 5.19(D) and 5.19(F) are the difference calculated between two stochastic distributions and since the sampling is random this function is not necessarily smooth, nevertheless, the fission distribution alone (and therefore the power) is smooth enough as we can see in Figure 5.16.

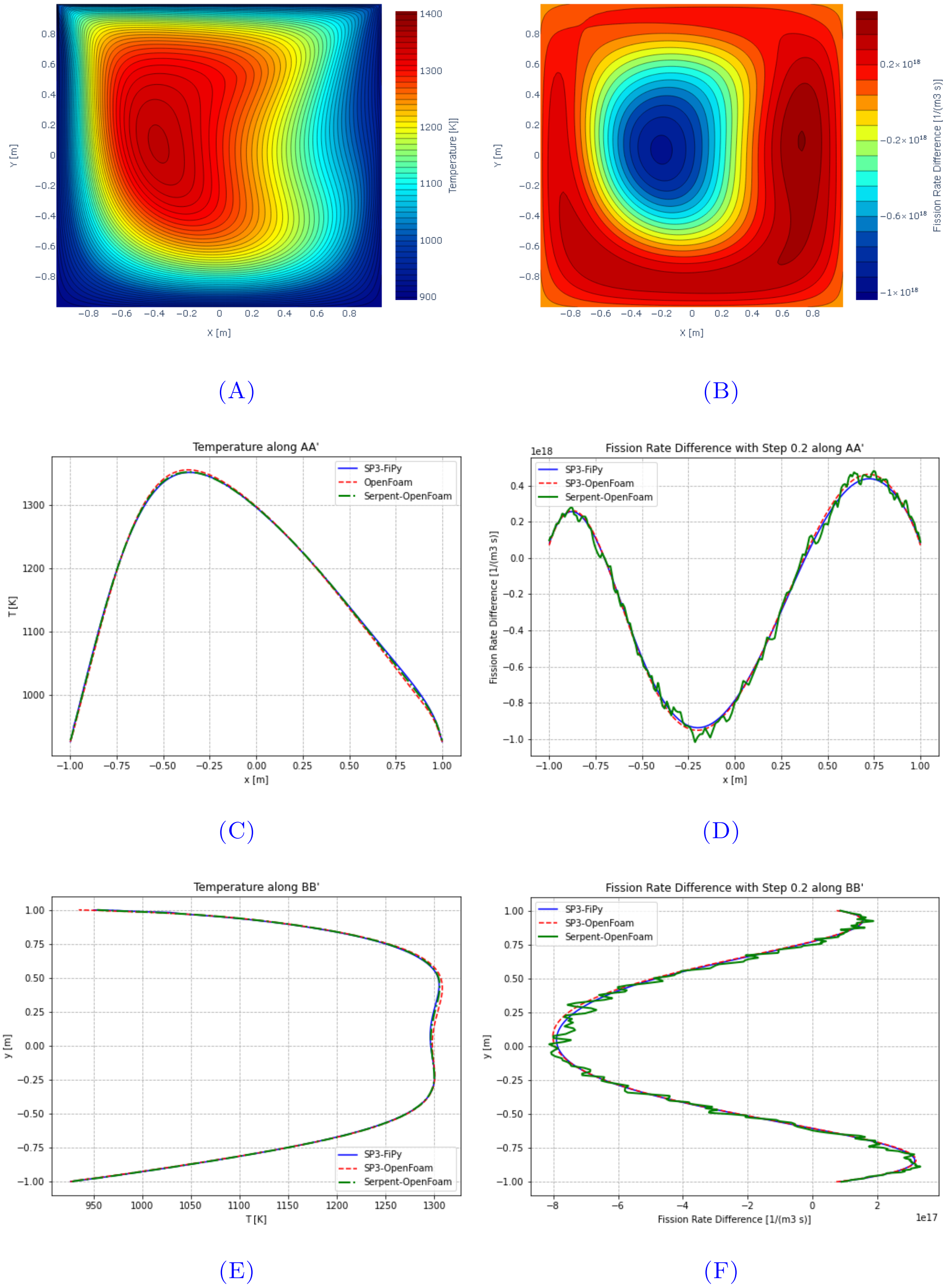
The reactivity for the Step 1.2 is shown in Table 5.13. The SP_3-OpenFOAM bias of ~60 pcm is still consistent to previous steps. On the other hand, the difference between steps 1.2 and 1.1 are consistent among all methods.

Table 5.13 – Step 1.2: Reactivity values comparison for the neutronics models (SP_3 in FiPy, SP_1 and SP_3 in OpenFOAM and Serpent-OpenFOAM internal coupling) against neutronics in presence of a circulating fuel from Step 1.1.

	Reactivity ρ [pcm]		
Model	**Step 1.1**	**Step 1.2**	**Difference**
Serpent-OpenFOAM	338	-803	-1141
Diffusion/SP_1-OpenFOAM	349	-803	-1152
SP_3-OpenFOAM	291	-862	-1153
SP_3-FiPy	345	-795	-1140

Step 1.3 tests the full coupling without forced convection, i.e. the velocity at the top wall is zero ($U_x = 0$). Buoyancy effects observed in this step of the benchmark were calculated via the Boussinesq approximation. In Figures 5.20 and 5.21, the velocity, temperature and delayed neutron precursors concentration show a very good agreement between all models. Table 5.14 shows similar reactivity variations between them. At this point, we can conclude that the coupling between neutronics and thermal-hydraulics phenomena works well.

Finally, results for the **step 1.4** are presented in Figure 5.22: the temperature, velocity, fission rate and precursors fields are shown. There, the combined effect of precursors transport, buoyancy, forced convection, and density feedback effects perturb the system.



(A)

(B)

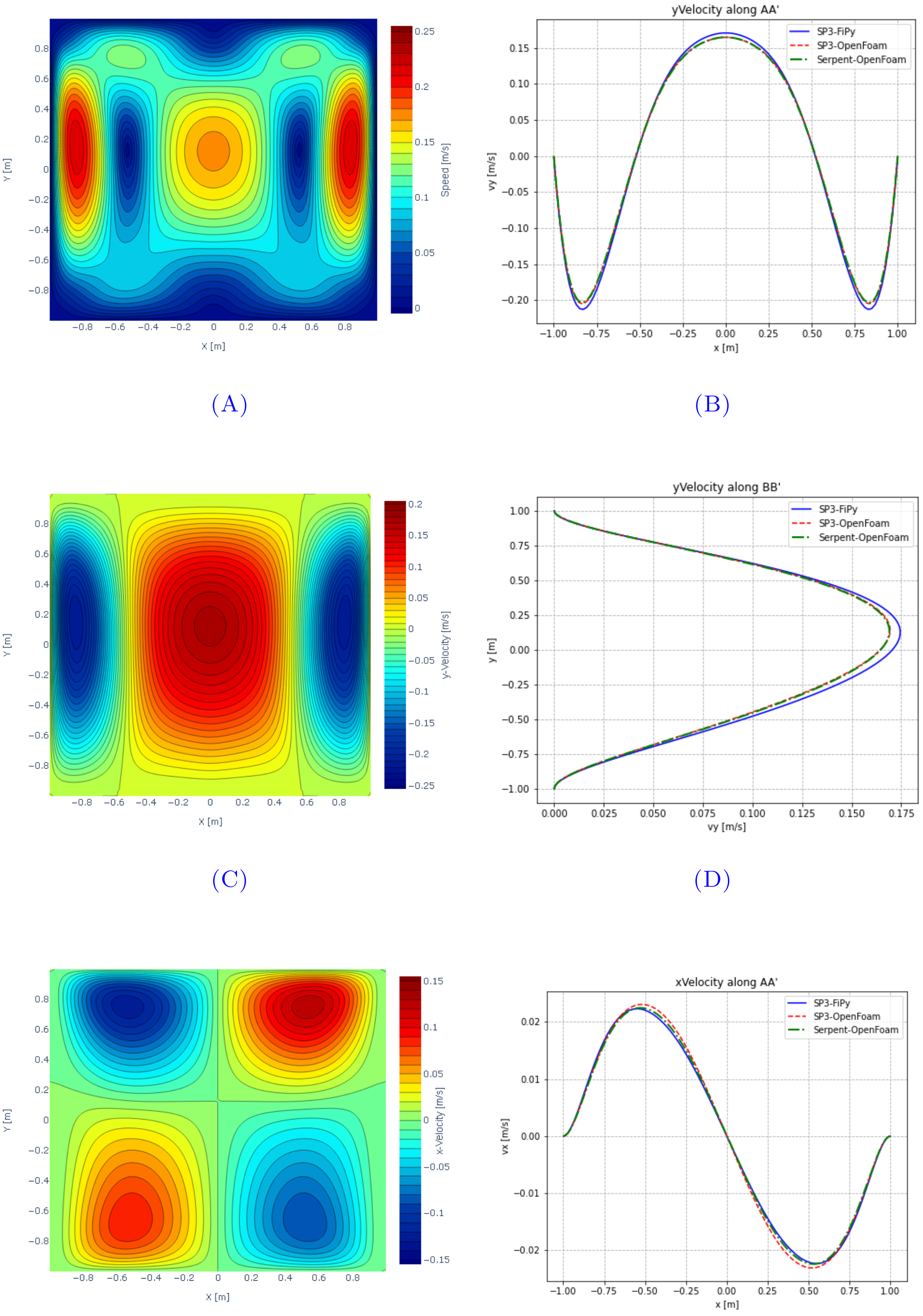
(C)

(D)

(E)

(F)

FIGURE 5.19 – Step 1.2: Temperature (A) inside the cavity, (C) along AA' and (E) along BB'. Fission rate difference respect to Step 0.2 ($[\Sigma_f \phi]_{step1.2} - [\Sigma_f \phi]_{step0.2}$) (B) inside the cavity, (D) along AA' and (F) along BB' for SP$_3$ (in FiPy and OpenFOAM) and Serpent-OpenFOAM internal coupling models.

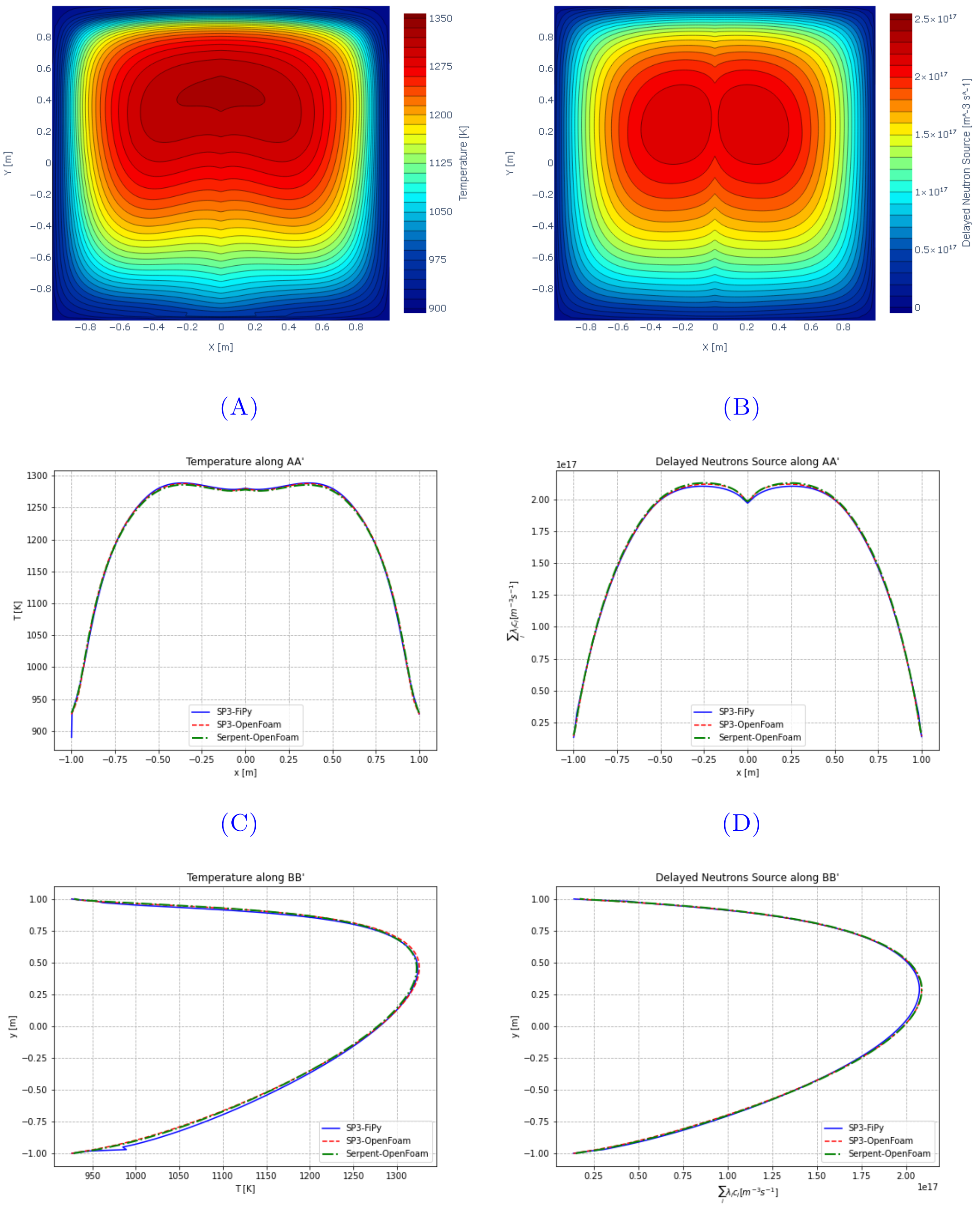


(A) (B)

(C) (D)

(E) (F)

Figure 5.20 – Step 1.3: Velocity (A) Magnitude, (C) Y-component, (E) X-component, (D) Y-component along AA', (D) Y-component along BB' and protect(D) X-component along AA' calculated in FiPy and OpenFOAM for SP$_3$ (in FiPy and OpenFOAM) and Serpent-OpenFOAM internal coupling models.



(A)

(B)

(C)

(D)

(E)

(F)

Figure 5.21 – Step 1.3: Temperature (A) inside the cavity, (C) along AA' and (E) along BB'. Delayed neutron source (B) inside the cavity, (D) along AA' and (F) along BB' for SP_3 (in FiPy and OpenFOAM) and Serpent-OpenFOAM internal coupling models.

TABLE 5.14 – Step 1.3: Reactivity values comparison for the neutronics models (SP_3 in FiPy, SP_1 and SP_3 in OpenFOAM and Serpent-OpenFOAM internal coupling) against neutronics calculations from Step 0.2.

	Reactivity ρ [pcm]		
Model	**Step 0.2**	**Step 1.3**	**Difference**
Serpent-OpenFOAM	402	-800	-1202
Diffusion/SP_1-OpenFOAM	411	-809	-1220
SP_3-OpenFOAM	354	-867	-1221
SP_3-FiPy	405	-803	-1208

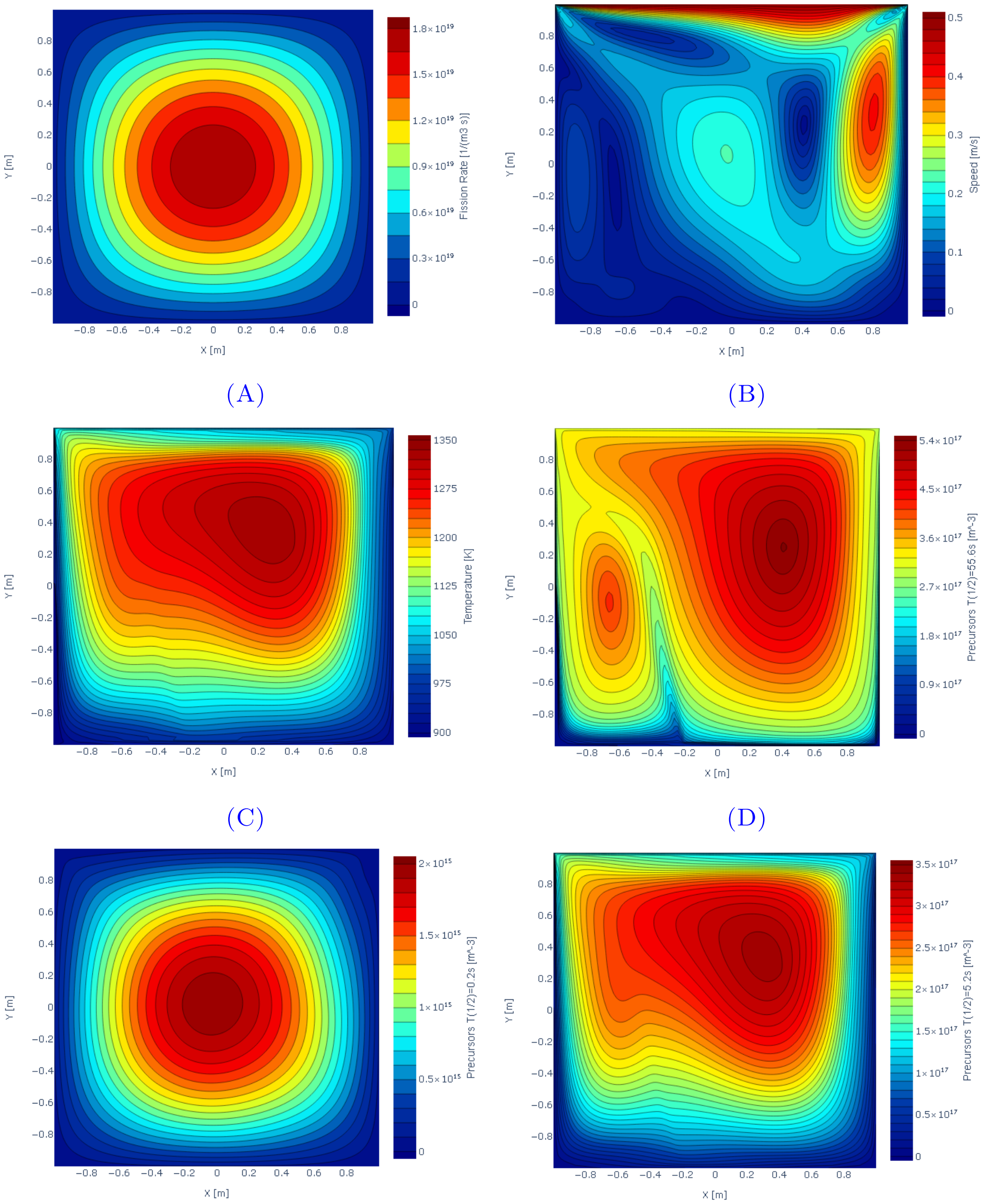
The curves along the AA' and BB' are not shown here but all methods coincides as in previous steps. They are shown together with the results from the other three models in Appendix C.

From the velocity fields the effects of the forced convection given by the 0.5 m s^{-1} speed set at the top and the buoyancy effects combined effects can be observed. Now, the advection impacts directly on the temperature distribution. Even more important is its effect on the precursors field. Normally, in solid fuels, where there is no fuel circulation, the precursors concentration is proportional to the fission source. In this case, we observed from the bottom figures the advection effect on the precursors concentration. Depending on the family half-life the effects become more or less important, the higher the half-life the more time before the delayed neutron is emitted and, therefore, the precursors are advected further from its source.

Another important result is the coupling impact on the reactivity. From Table 5.15 we can see ~ -1200 *pcm* worth change in the reactivity between the neutronics stand-alone calculations (so no fuel convection and constant temperature) and the full coupling situation. Also, the comparison between neutronics models shows very similar results for the three methods. SP_3-OpenFOAM has the biggest difference as observed before (~60 pcm).

TABLE 5.15 – Step 1.4: Reactivity values comparison for the neutronics models (SP_3 in FiPy, SP_1 and SP_3 in OpenFOAM and Serpent-OpenFOAM internal coupling) against neutronics calculations from Step 0.2.

	Reactivity ρ [pcm]		
Model	**Step 0.2**	**Step 1.4**	**Difference**
Serpent-OpenFOAM	402	-798	-1200
Diffusion/SP_1-OpenFOAM	411	-794	-1205
SP_3-OpenFOAM	354	-855	-1205
SP_3-FiPy	405	-787	-1192



(A)

(B)

(C)

(D)

(E)

(F)

FIGURE 5.22 – Step 1.4: (A) Fission Rate, (B) Velocity Magnitude, (C) Temperature and (D) - (F) Precursors fields at Steady-State for SP$_3$ (in FiPy and OpenFOAM) and Serpent-OpenFOAM internal coupling models.

5.2.2 Transient

As discussed in Chapter 2 the goal of the transient calculations of the Molten Salt Cavity Benchmark was to determine the transfer function of the tool. Then, by comparing the transfer functions of different tools one can evaluate if the time responses of the tools are equivalent for small perturbations and thus if the coupling are equivalent.

The transient calculations for Phase 2 (Figure 5.23) were performed by setting the heat exchange coefficient to $h_{sink}(t) = h_{sink,0}(1 + Asin(2\pi t/T_p))$ where A is the amplitude of the perturbation and T_p its period. In our analysis, A was set to 0.1 (i.e. 10% change) in order to minimize the effects on the non-linear phenomena and be able to use this linear analysis tool. The value of T_p was set to 1.25s, 2.5s, 5s, 10s, 20s, 40s and 80s. The gain and the phase shift were calculated as $Gain = 10(P_{max}/P_{ref} - 1)$ and $\theta = 360°\Delta t/T_p$, where P_{max} is the maximum power, P_{ref} is 1 GW and $\Delta t = t^P_{max} - t^h_{max}$ is the difference between the peak time of the heat exchange coefficient and the peak time of the power.

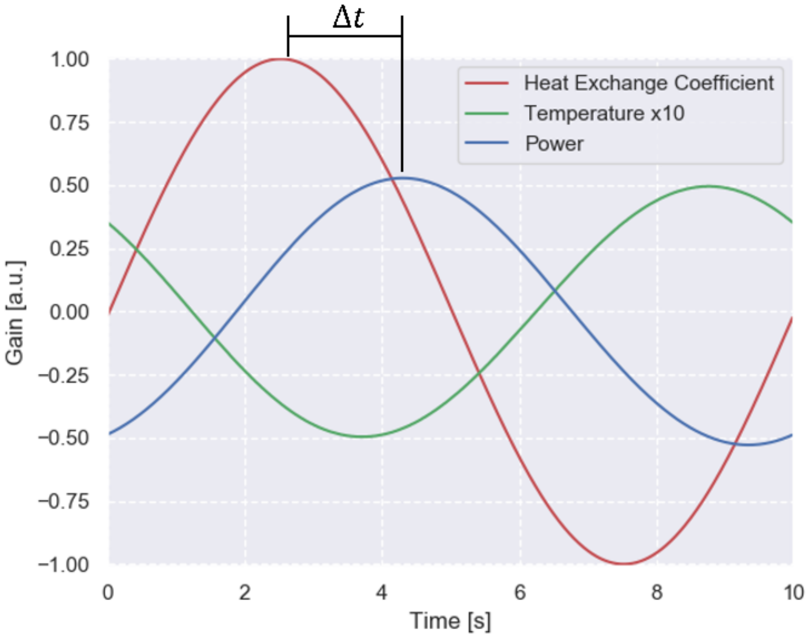


FIGURE 5.23 – Scheme of the integral power, heat exchange and mean temperature gain transient evolution.

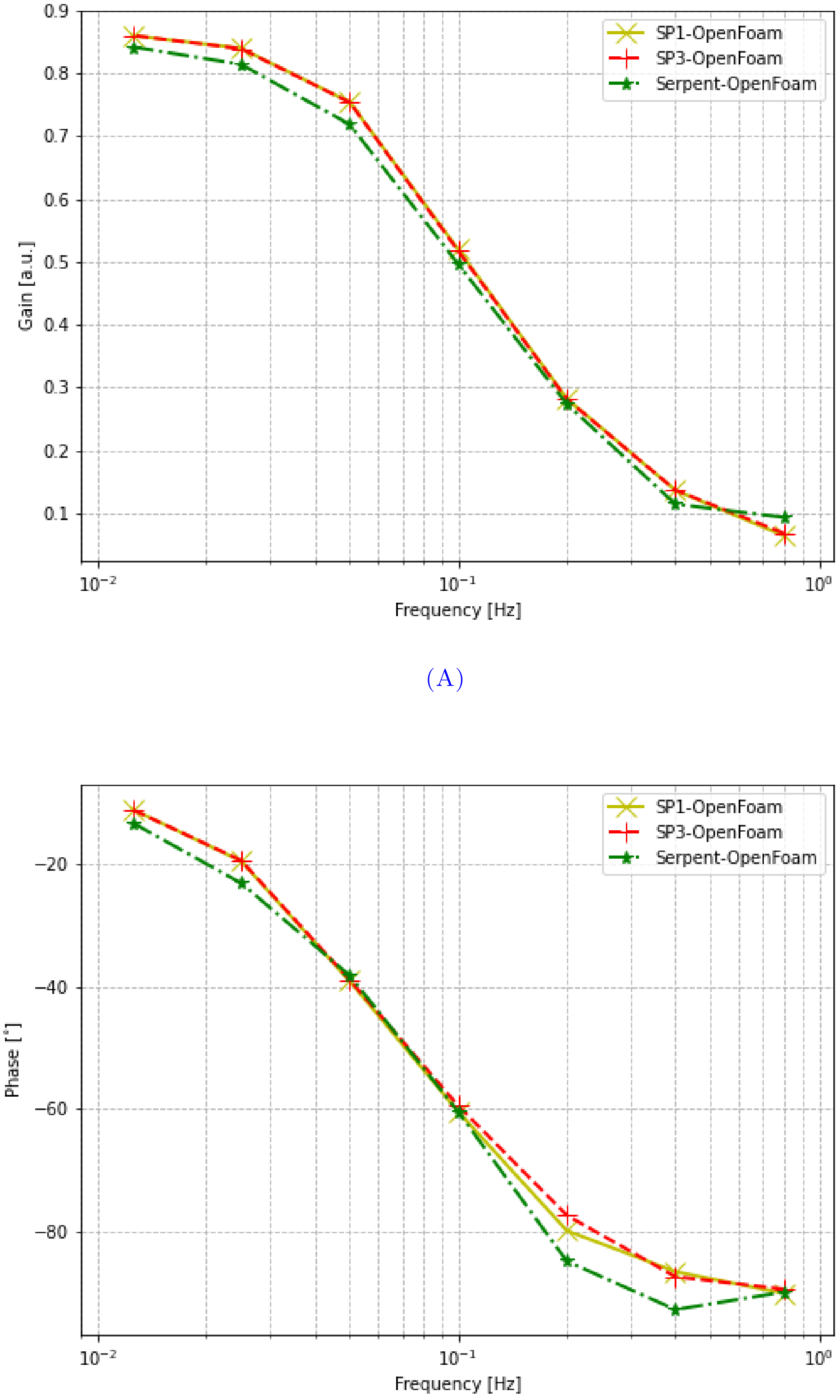
In Figure 5.24, both gain and phase change are presented. As we can see, for long periods the transient is slow, i.e., the system has time to follow the perturbation, and, as expected, gain is close to unity and there is little phase change. This means that a 10% change in the heat exchange coefficient will translate into a 10% change in the power. On the contrary, for smaller periods (higher frequencies) the system has little time to adjust.

Therefore, a perturbation at one time will impact later in the transient. For example, precursors with long half-life will emit delayed neutrons at a time where the perturbation has once again changed, with no time to reach the equilibrium. In addition, as expected, as the frequency of the perturbation increases the system becomes less sensitive, thus, any periodic perturbation (alternating signs, i.e., ±10 %) will compensate and the system will remain close to the 1 GW mean value.

As can be seen in Figure 5.24 the results of the three neutronics models agree very well. Even though the Monte Carlo adiabatic Quasi-Static Method predictions are systematically slightly below the other two models results. Further studies in the numerical convergence of all the methods should be done to verify is this deviation can be attributed to the Monte Carlo adiabatic Quasi-Static Method model and what the origin of this error is. For example, one source of error in the Quasi-Static Method is that the adiabatic approach implemented by equation 3.67 considers that the delayed neutron source is constantly in equilibrium with the fission source when calculating the flux shape function. This assumption can be approximately true when the period is large and the system has time to adjust but it introduces more error as long as the period becomes shorter. As we discussed in Chapter 3 this approximation can be easily removed. Also, since the adiabatic method supposes that both the time derivative of the flux shape and the flux amplitude can be neglected thus introducing a small error. The flux shape does not change much though but the fluctuations in the precursors' distribution shape has an impact in the kinetic parameters' calculations[3]. It is important to highlight that the agreement between the three methods is a numerical confirmation that the terms from Section 3.1.2.3 are indeed either zero or at least negligible.

As for performance, diffusion and SP$_3$ methods were calculated with OpenFOAM domain decomposition parallel strategy with 20 processors. For the Serpent-based cases in OpenFOAM 1 processor were used for temperature and velocity resolution and data management and 24 processors for Monte Carlo with 10000 batches with 20000 histories each. 20 uniformly distributed flux shape steps were performed per cycle (each whole period time) for the Quasi-Static Method. In Table 5.16, the calculation time per cycle is shown.

[3] The point kinetics' parameters update inside the shape time step (as with the density in the Godiva model) is needed. The density feedback update inside the shape time step (as for Godiva) was considered but it was not enough to give a smooth reactivity like in the Godiva Experiment. In Appendix C, the transient evolution of the integral power, reactivity and mean temperature are presented showing this behaviour.



(A)

(B)

Figure 5.24 – (A) Gain and (B) Phase change vs frequency for the diffusion/SP_1 and SP_3 models in OpenFOAM and the Monte Carlo Quasi-Static Method with Serpent-OpenFOAM (OF-PCQM) model.

An Intel(R) Xeon(R) Gold 5118 CPU @ 2.30GHz was used. We can see that the additional cost of Monte Carlo is 3 times the one from diffusion and SP_3. The latter is slightly larger than diffusion. In all cases a Courant number of 0.8 was fixed giving a time step of approximately 0.012 s. These results show that the practical application of the Monte Carlo method through a Quasi-Static Method approach is feasible and provide very good results. It is important to recall that the analysis of some of the nuclear systems of interest for criticality studies could be very challenging with SP_1 or SP_3 methods and in those case a Monte Carlo method will offer significant advantages. Using the Dynamic Monte Carlo method discussed in the previous section for this study case would be very challenging (if feasible) from a computational point of view (memory and CPU time).

Table 5.16 – Computational cost of the transient simulation per cycle of the perturbation (time to complete one perturbation period time T_p) for the neutronics models SP_1 and SP_3 in OpenFOAM and the Monte Carlo Quasi-Static Method with Serpent-OpenFOAM (OF-PCQM).

Model	Average Execution Time (s) per Cycle
Diffusion/SP_1-OpenFOAM	6
SP_3-OpenFOAM	8
Serpent-OpenFOAM (OF-PCQM)	21

5.2.3 Final Remarks on the Molten Salt Cavity Study Case

Comparison between the numeric performances of the Monte Carlo Quasi-Static Method, Simplified P_3 and diffusion methods for a liquid nuclear fuel (molten salt) system has been presented. The numerical results obtained for a study case show that the three models provide similar results for transient calculations in a nuclear system similar to a Molten Salt Fast Reactor. Also, the computational cost of the Monte Carlo implemented through a Quasi-Static Method appears to be very reasonable. Moreover, as discussed in the previous study case, the performance of the Monte Carlo Quasi-Static Method can be improved in the future by optimizing the number of flux shape calculations and the amplitude parameters update due to the precursors shape change. Finally, it was found that as expected the use of a constant weight function for the Quasi-Static Method normalization did not introduce errors while allowing a significantly decrease the complexity of the delayed neutrons precursors equations as one could expected in a liquid fuel. This is a very important result and confirm the conclusions of the Chapter 3.

5.3 Heterogeneous Medium: Spent Fuel Pools

In this section, we analyze the last study case: the Spent Fuel Pool. This case lets us study the performance of the multi-physics models regarding to phenomena encountered in an heterogeneous multi-phase porous medium. This type of model is very useful to reduce the geometrical complexity of large systems. The section is divided in four parts. In the first part, some remarks on the model and the coupling strategy are made. In the second part (Section 5.3.2) the results of a sensitivity analysis performed with the Monte Carlo neutronics model are discussed. The main objective of this study was to gain a better insight on the mechanisms acting during these accidents. In particular, the effects of the following key parameters on the pool reactivity were studied: water level, water density and boron concentration, cladding absorption and fuel burn-up. The third part (Section 5.3.3) is devoted to presenting the results from a first application case of the porous medium model described in Section 3.3.7. In this study case, a drainage of the fuel assembly rack was assumed, as a result of a sudden drop of the pool water level from the full level to 2.5 m measured from the base of the active Fuel Assembly length. For the purpose of evaluating the performance of the numerical model it is also supposed that this occurs in a short period of time. Moreover, due to the Fuel Assemblies decay heat the coolant temperature rises and a difference between the rack water level H and the pool water level h appears as a result of the coolant densities variations (water density becomes smaller in the rack with respect to the pool and if there is not a close circuit between the rack and the pool, the level of the rack H will be higher than that of the pool h). The system water level in the rack, its temperature and the multiplication factor were analysed until the onset of boiling. Finally, general remarks about this case are made. The results of this work were published in the 11^{th} International Conference on Nuclear Criticality Safety (ICNC) 2019 [115].

5.3.1 Model

For the criticality analysis of the Spent Fuel Pool a detailed model of the racks in the Spent Fuel Pool 4 was developed in the Monte Carlo code Serpent. In the next subsections a description of the assembly and the rack is given along with the coupling strategy chosen.

5.3.1.1 Assembly

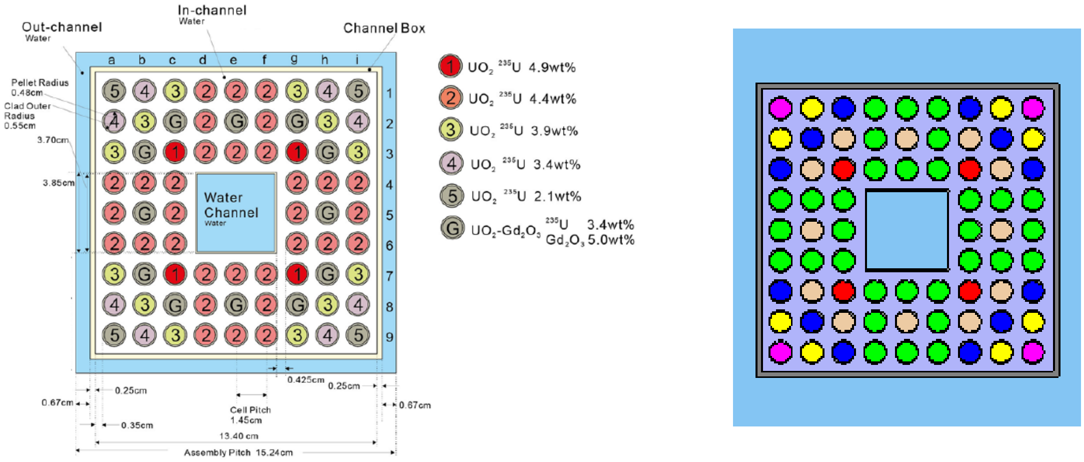
The fuel inventory of the pool at the moment of the accident was taken from [15]. Among the various fuel types the STEP3-B (see Table 5.17) was chosen in this work since they have the highest decay heat (favouring a fast evaporation and thus a larger difference of water level between the racks and the pool). This type of Fuel Assemblies had different levels of burn-up. All the geometrical details as well as isotopic composition for different burn-up degrees are given in the work of [116], which discusses the results of an OECD-NEA Benchmark for a BWR Spent Fuel Pool. Figure 5.25 displays the model used in the OECD-NEA Benchmark and the one developed in Serpent for this study. Note that the OECD-NEA benchmark model is a 2D model (i.e. using a unit Fuel Assembly model) while in our case the thermal-hydraulics calculations requires the use of a 3D model. This 3D model was built considering an active length of 3.7084 m [16]. As in the benchmark, the top handle, bottom nozzle, upper and lower tie-plate and spacers were not modelled but could be added in the model. However, they are not expected to significantly change the results.

TABLE 5.17 – Fuel composition and decay heat in SFP 4 FAs [15].

Fuel Type	Assembly Configuration	Number of Fuel Assemblies	Discharge Data	Cooling Duration as of March 11, 2011 (yrs)	Average Assembly Decay Heat as of March 11, 2011 (W)
STEP3-B	9x9	1 87 100 548	October 2, 2006 March 28, 2008 September 29, 2009 November 30, 2010	4.4 3 1.5 0.3	472.5 676.9 1267 3416
STEP3-B Fresh Fuel	9x9	204	-	-	0

5.3.1.2 Rack

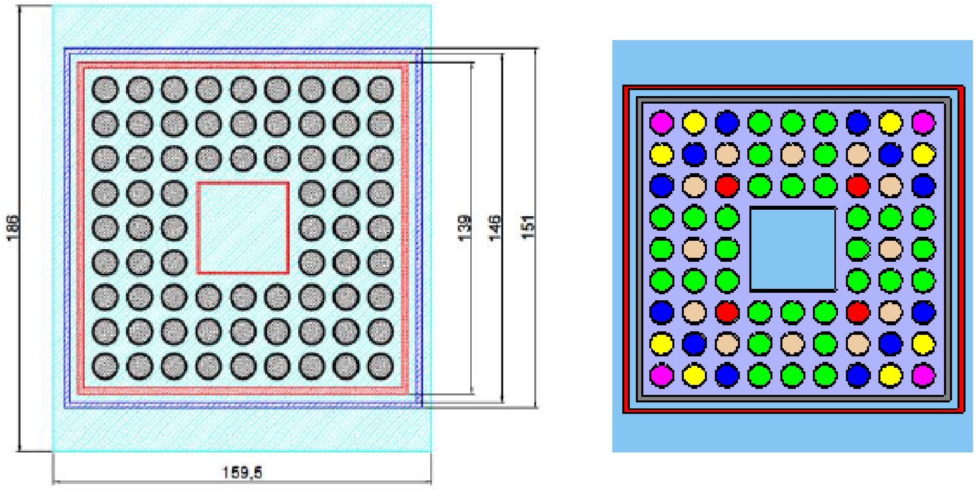
The rack walls were modelled by considering that each Fuel Assembly was placed inside a square stainless steel (SS304) envelope (15.1 cm outer side and 0.25 cm thickness ([117]) as displayed in Figure 5.26(A). The upper end of the walls was set equal to the upper active length of the fuel rods while the lower end was set equal to the elevation corresponding to 8 cm below the active length of the fuel rods. Figure 5.26(B) shows the Serpent model for one Fuel Assembly with the rack steel walls in red. More details could be added but for the purpose of the present study it was judged that the current level of details was sufficient for this preliminary work. The FAs are placed in a 3×10 arrangement inside the rack with



(A) (B)

FIGURE 5.25 – STEP3-B BWR Assembly Model: (A) OECD-NEA Benchmark 2D model. (B) Serpent 3D model used in this work.

dimensions 160.85 cm by 54.5 cm as detailed in [118]. FAs have pitches of 15.95 cm and 18.6 cm [117] along the two horizontal directions. For the rack and using symmetry, a 3x5 system surrounded by a 5 cm water gap[4] was modelled with reflective conditions on the outer sides and void conditions at top and bottom limits. The full Serpent model for the rack (walls in red) and the Fuel Assemblies is presented in Figure 5.27.



(A) (B)

FIGURE 5.26 – STEP3-B BWR and steel walls of the rack: (A) Fuel assembly inside a rack cell. (B) Serpent 3D model used in this work.

[4] Estimated from Unit 4-SFP dimension and racks inside [16]

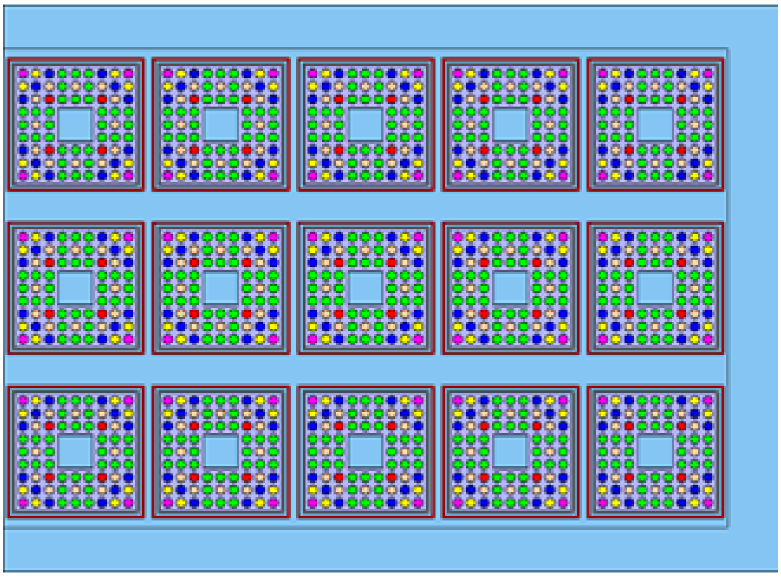


FIGURE 5.27 – Full 3x5 Rack Model with the Fuel Assemblies.

5.3.1.3 Coupling Strategy

As explained in Section 4.3.3.2, the porous medium equations from Section 3.3.7 were implemented in OpenFOAM to be able to model wit less computational cost the Spent Fuel Pool thermal-hydraulics phenomena in comparison to a Navier-Stokes approach. To summarize, the thermal-hydraulics model and the neutronics model in Serpent from the previous Section were coupled as shown in Figure 5.28. In the adopted coupling scheme, the materials density and temperature fields are calculated by the porous medium model implemented in OpenFOAM and then sent to the Serpent neutronics model to update the materials cross sections. For now, only the water density and temperature are updated internally in Serpent. The dashed red line in Figure 5.28 indicates that the power field due to fission rate will only be possible if criticality is reached. If not, only the decay heat will contribute to the porous medium power source term and the coupling between the thermal-hydraulics model and the neutronics model will be weak.

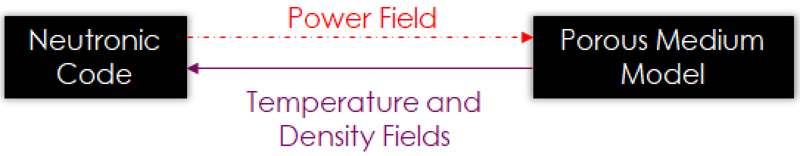


FIGURE 5.28 – Neutronic – Thermal Hydraulics coupling scheme.

It is reminded that even though the whole system of thermal-hydaulics equations for a multi-phase multi-component porous medium was implemented, some simplifications were introduced in Section 3.3.7.5, such as Henry's and Raoult's laws (equations 3.155 and 3.156) and thermal equilibrium between the phases, i.e. $T_s = T_w = T_n = T$ because of the low power density (decay heat). In case of recriticality, this should be modified and a two or three temperatures model should be implemented.

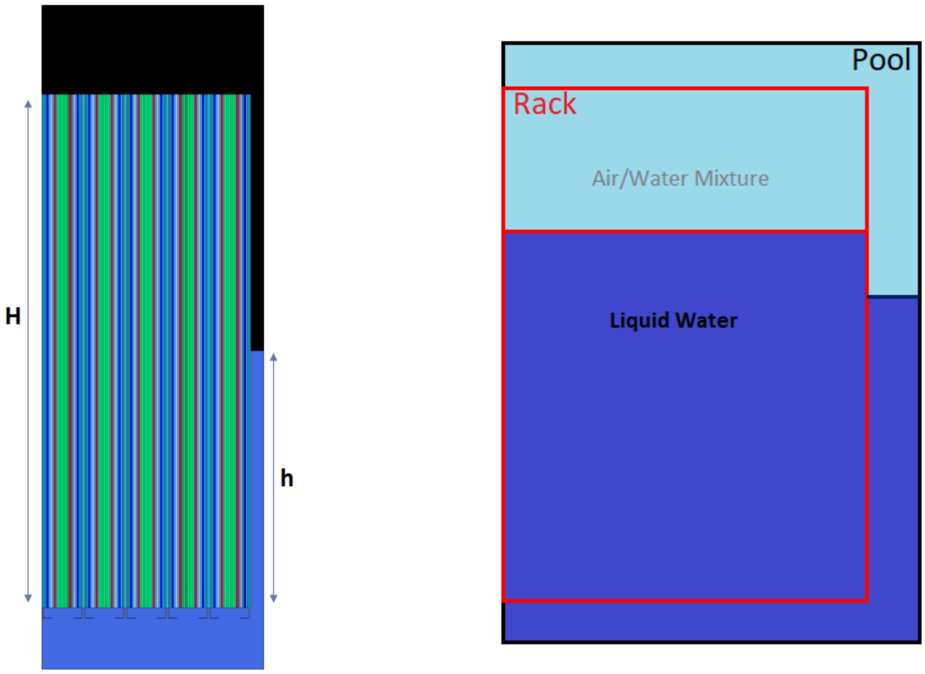
5.3.2 Sensitivity Analysis

As expressed before, boron concentration, burn-up, water level and density in the rack and pool and stainless-steel effect were analyzed. Table 5.18 summarizes the parameters that were studied and their values. Different combinations of these values were considered. As displayed in Figure 5.29, H denotes the height of the water in the rack while h denotes the water level of the pool outside the racks. For the initial sensitivity cases these two levels were considered to be the same ($H = h$) and both function of time. As explained in Section 5.3.1 the isotopic composition of the Fuel Assemblies were taken from [116]. As can be seen in Figure 5.29, to simplify the sensitivity calculations, the fluid phase was approximated as having two different regions: (a) the lower part of the rack (below $H = h$) filled with pure liquid pure water ($\rho = 1000\ kg\ m^{-3}$) and (b) the upper part of the rack (above $H = h$) composed by a mixture of water and air (with density varying from 0 to 1000 kg/m^3). The parameters used in this study are reported in Table 5.18.

Table 5.18 – Parameters studied and values undertaken by them.

Parameter	Values
Water Level H=h [m]	0, 0.9271, 1.8542, 2.7813, 3.7084
Water/Air Mixture [kg/m3]	0, 100, 200, 250, 300, 400, 500, 750, 1000
Burn-up [MWd/kgU]	0, 12, 50
Boron Concentration	0, 1000, 2000

Firstly, the predicted multiplication factor (k_{eff}) value of the Spent Fuel Pool as a function of the water level (the water level in rack and in the pool are the same, i.e. $H = h$) is presented in Figure 5.30. To identify the effect of each of the factors discussed earlier, the results correspond to the case were the water-air mixture density was set to zero (i.e. vacuum), the pool was filled only with fresh fuels (zero burn-up) and there was no boron in the coolant (note that one of the criterion of the US Nuclear Regulatory Commission for this type of analysis is k_{eff} being less than 1.0 for Spent Fuel Pool filled



(A)

(B)

FIGURE 5.29 – (A) Axial cross section of the rack-pool system. (B) Two fluid regions are considered: a lower region filled with water (dark blue) and the upper region filled with a water-air mixture right (light blue).

with unborated water). The zero level on the horizontal axis of the figure corresponds to the beginning of the assemblies active length while the upper limit corresponds to the top of the active fuel length (3.7084 m). As one could expect the figure shows that as the water level decrease so does k_{eff}.

A different trend is observed if one assumes that the liquid water level decreases and the space left on the top of the pool and the rack is filled with a water-air mixture. Figure 5.31 extends the analysis of Figure 5.30 by considering the densities reported in Table 5.18 for the water-air mixture added above the liquid water level. To increase the Spent Fuel Pool reactivity, assemblies with an averaged burn-up of 12 MWd/kg were considered for this study. The "Void" case is similar to the one studied in Figure 5.30 but with an irradiated Fuel Assembly. As can be seen on the left of Figure 5.31, as the top water-air mixture density increases, the variation of the k_{eff} with the water level changes. At first, the reactivity decreases with the water level reduction but for mixture densities beyond 100 kg/m^3 the trends are different. Eventually, the maximum increase of the pool reactivity is reached when a water-air mixture of about 250 kg/m^3 is used to fill the top of the pool and the rack. For mixture densities higher or equal to 250 kg/m^3 we observe the same

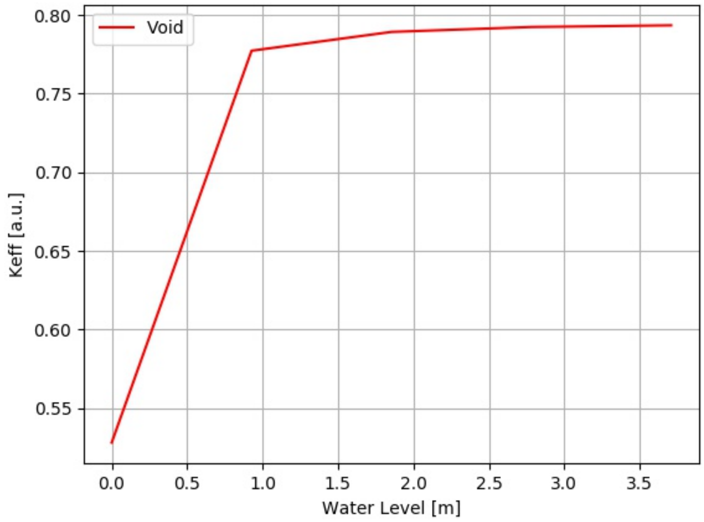
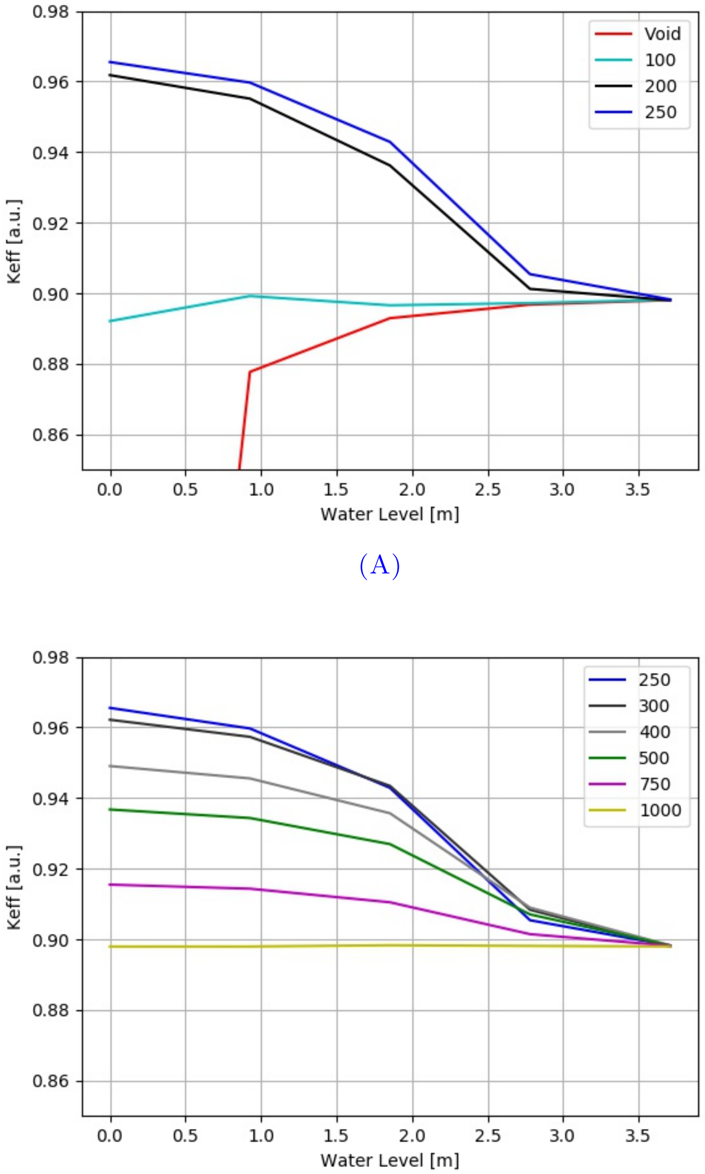


FIGURE 5.30 – k_{eff} as a function of the water level in the rack-pool system. The mixture zone is void. The SFP is filled with fresh Fuel Assemblies. Boron concentration was set to zero.

dependency but the maximum peak decreases. Finally, the yellow curve gives a reference of the k_{eff} when the pool is full of water (1000 kg/m^3 implies liquid water above). From these results we can conclude that in the case of a LOCA, the presence of a mixture of water and air resulting from the water boiling could result in an augmentation of the k_{eff} as the level of the Spent Fuel Pool decreases (although the density values of the mixture are not necessary realistic). This is opposed to the case of a simple drainage (no mixture) where the reactivity decreases as the water level decreases.

This overall behaviour confirms that as discussed in Chapter 2 the Spent Fuel Pool is over-moderated. This is confirmed by the results displayed on Figure 5.32 which shows the reactivity k_{eff} of the Spent Fuel Pool in function of the water-air mixture density at different liquid water levels ($H = h$). Once again, the composition corresponds to a burn-up of 12 MWd/kg and there is no boron. Each of the curves from Figure 5.31 are, in fact, the cuts of Figure 5.32 for a given density. Two observations can be made from Figure 5.32. First, we can see that the maximum reactivity corresponds, as observed before, to 250 kg/m^3 which is the point where the system passes from under-moderated to over-moderated regions. This means that, in the case of a LOCA/LOFA accident, if the vapor-air mixed density is on the left of 250 kg/m^3 limit, any increase of the coolant temperature and thus a decrease in density will diminish the k_{eff}. On the other hand, if the mixture is cooled-down due to, for example, the spray of water to cool down the



(A)

(B)

FIGURE 5.31 – Multiplication factor k_{eff} as a function of the water level in the rack-pool system. The mixture zone density varies from $0 - 1000 kg/m^3$. The FAs have a burn-up 12 MWd/kg. Boron concentration was set to zero.

rack the effect will be the opposite. On the right of the 250 kg/m^3 limit, the contrary is observed. Secondly, at 100 kg/m^3 there is an inversion of the curves. This could be linked to the fact that at this point the mean free path of the neutron is so big that moderation is by no means more important than absorption and leakage: decreasing the water level will quickly decrease the pool reactivity.

Figures 5.33, 5.34 and 5.35 present the effects of the burn-up level, the stainless-steel used in the rack structure and the boron concentration in the water. We can see that the

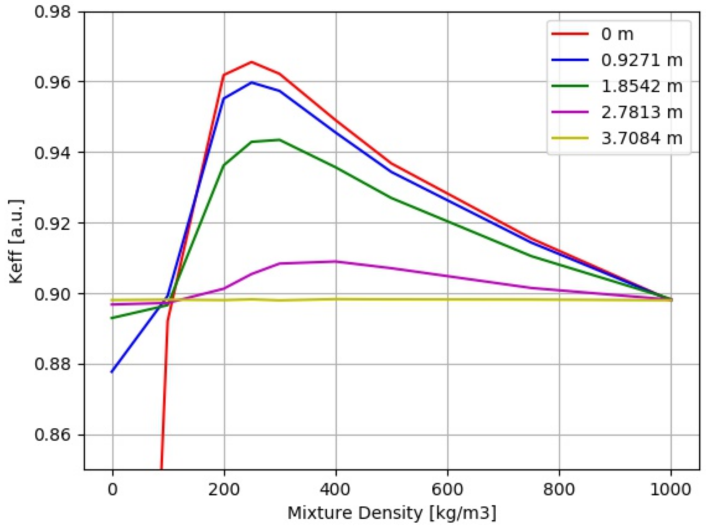
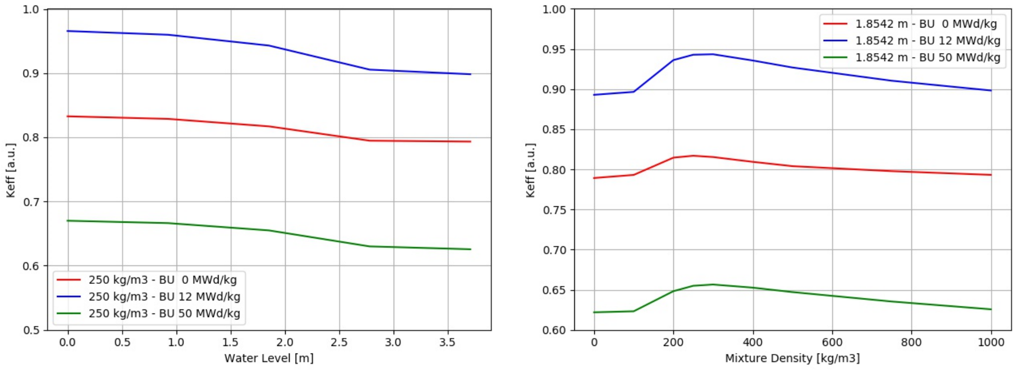


FIGURE 5.32 – Multiplication Factor k_{eff} as a function of the mixture density in the rack-pool system. The water level varies from 0-3.7084 m. The FAs have a fuel burn-up of 12 MWd/kgU^{-1}. Boron concentration was set to zero.

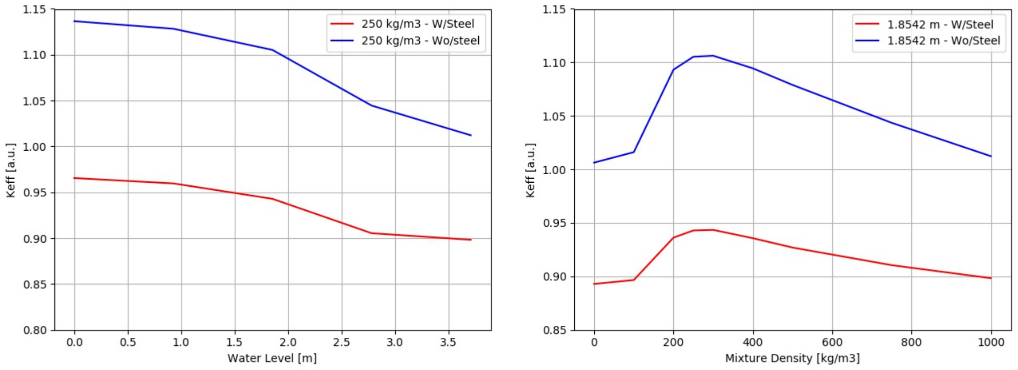
highest reactivity is obtained for the irradiated Fuel Assemblies having the lowest burn-up (12 MWd/kgU^{-1}). The difference of approximately 10000 pcm with respect to the fresh fuel is possible due to conversion of ^{238}U to ^{239}Pu. Nevertheless, it should be noted that the fresh Fuel Assembly has a heterogeneous composition with different enrichment and gadolinium concentrations per pin as discussed in Section 5.3.1, while uniform fuel pins composition are used to model the other burn-ups. The effect of the stainless-steel cladding in the rack is shown in Figure 5.34. As can be seen the presence of steel is very important to keep the pool system subcritical. Its weight is of roughly 11000 pcm. Note that as expected the presence of boron shifts the k_{eff} vs mixture density curves down and to the left. The increase of absorption explains the reduction of up to nearly 30000 pcm for 2000 ppm of boron. In addition, the moderation to absorption ratio variation changes the peak and therefore the point at which the system moves from under to over-moderated regions.

The effect of the difference between the water levels inside the fuel rack (H) and the pool (h) is shown in Figure 5.36. This difference between the water levels would occur as a result of the density differences between the water in the pool and inside the racks (where heating from the assemblies is important and thus causes a decrease of the coolant density). Figure 5.36 shows the predicted value of k_{eff} as a function of the water level in the racks and for different levels of the pool water. It was considered that the Spent Fuel Pool was filled with fresh Fuel Assemblies (zero burn-up), zero boron concentration in the



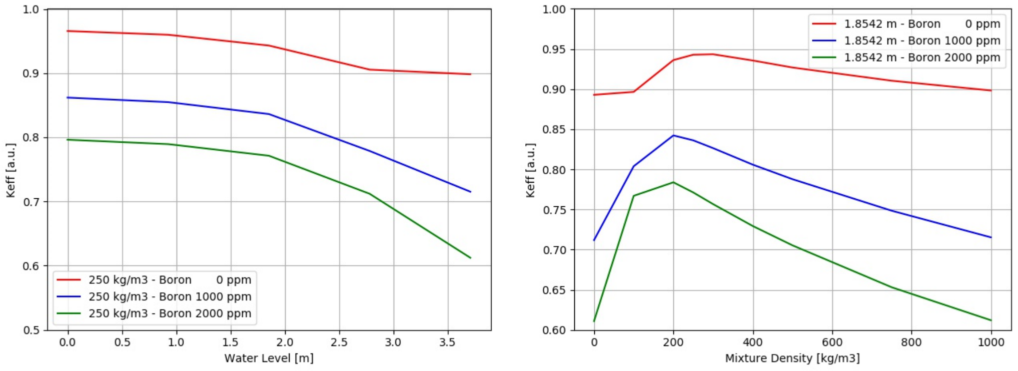
(A) (B)

FIGURE 5.33 – Multiplication factor k_{eff} as a function of the water level (A) with a fixed water-air mixture density value of 250 kg/m^3 and (B) water level set to 1.8542 m. The FAs have a burn-up of 0, 12 and 50 MWd/kg respectively. Boron concentration was set to zero.



(A) (B)

FIGURE 5.34 – Multiplication factor k_{eff} as a function of the water level (A) with a fixed water-air mixture density of 250 kg/m^3 and (B) liquid water level set to 1.8542 m with and without the SS304 rack. Boron concentration was set to zero.



(A) (B)

FIGURE 5.35 – Multiplication factor k_{eff} as a function of the water level (A) with a fixed water-air mixture density of 250 kg/m^3 and (B) liquid water level was set to 1.8542 m. The FAs have a burn-up 12 MWd/kg. The following boron concentrations were used: 0, 1000 and 2000 ppm.

coolant and the water-air mixture (above the water level) was set to zero (void).

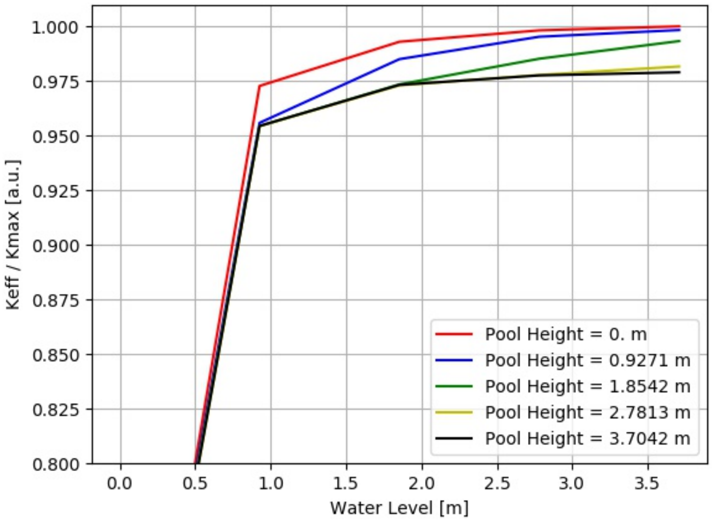


Figure 5.36 – Multiplication Factor k_{eff} as a function of the rack water level (H) for different pool water levels (h). The SFP is filled with fresh Fuel Assemblies, the boron concentration is zero and the water-air mixture is zero.

Although the values used in the analysis in Figure 5.36 are not necessarily realistic, they allow to understand the full range of variation of k_{eff} resulting from this effect. As can be seen, as the pool level decreases, the reactivity increases. This is explained by the fact that, as the pool level decrease, the degree of neutronic coupling between the racks makes the k_{eff} increase (i.e. neutrons leaked from one rack system can now reached another since water absorption no longer exist between them).

To summarize the important remarks that can be made from the results of the sensitivity analysis:

- Different water levels between the pool and the racks can lead to an increase of reactivity.
- The Spent Fuel Pool is over-moderated and thus replacing the water in the racks by a mixture of liquid water and air (or steam) can lead to an increase of the reactivity.
- Steel in the rack has a significant contribution to keep the Spent Fuel Pool subcritical.
- No credible scenario was identified where the pool could return to criticality (and thus fission power generated in the Fuel Assembly) although important reductions of the criticality margins may occur during the accident.

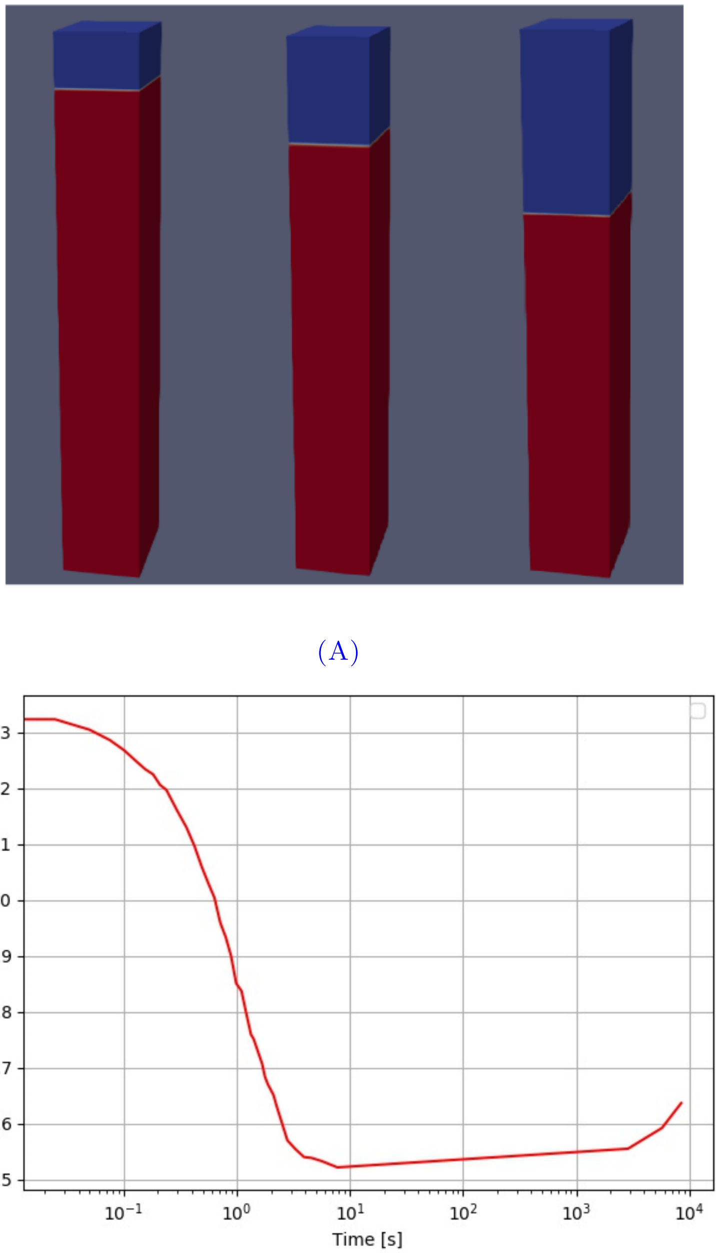
As discussed in Section 2.3, the variation of the water level between the racks and the pool requires the use of a relatively complex thermal-hydraulics model, which in our case was done by using a porous medium approach. A strong coupling between the two models (neutronics and thermal-hydraulics) was also adopted to allow for the possibility of studying scenarios where a criticality accident occurs in the Spent Fuel Pool. However, in the next study we consider only the effects of the decay heat since we could not identify a credible scenario with a criticality accident.

5.3.3 Coupled Model for Transient Analysis

This section presents an example of a coupled calculation using the multi-phase multi-components porous medium model described in Chapter 3. For this first application case, the following assumptions were made: there are two immiscible components air and water, each one has only one phase, gas (n) and liquid (w) respectively. Thermal equilibrium conditions between all phases and components is considered (this is correct if no return to criticality). The energy equations were reduced accordingly to only one temperature (thermal equilibrium). Water properties were taken from [119] while the air is considered an ideal gas with $h_w = 1005(T - 273.15) J/kg$ and $u_\alpha = h_\alpha - p_\alpha/\rho_\alpha$. The porosity is set to $\phi = 0.57$ and the permeability $K = 2.11e-61/m^2$ according to the BWR fuel geometry. For the solid phase (fuel rods), a density of 9087.2 kg/m^3, a heat capacity of 300 $J/(kgK)$ and a conductivity of 8.5 $W/(mK)$ were used.

An arrangement of 3x10 Fuel Assemblies in a rack was modelled using the dimensions reported in Section 5.3.1. The rack was considered as closed and therefore there are no exchanges between the pool and the Fuel Assemblies (except from the bottom part of the rack). As for the Boundary Conditions, the system is considered isolated on the lateral sides (Neumann condition), the pressure was set equal to the atmospheric pressure at the top of the rack and to 25 kPa (∼2.5 m hydraulic head) greater at the bottom of the rack. The saturations of the components are set to $S_W = 1$ at the bottom and $S_w = 0$ at the top (contact with liquid water and pure air respectively) and T=300 K at both ends.

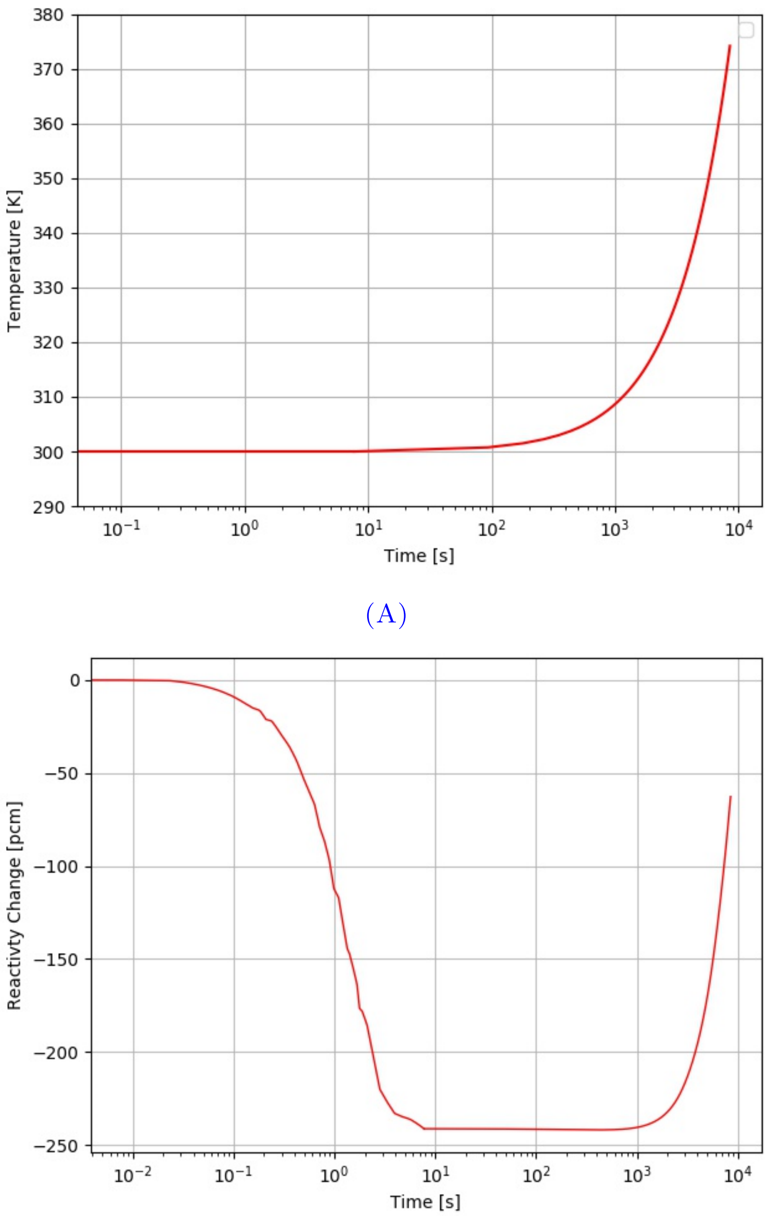
For this particular simulation (and to test the numerical resolution) it is considered that the pool level decreases instantaneously to 2.5 m at the beginning of the transient while the water in the rack has an initial level of 3.32 m of water at 300 K as we can see from Figure 5.37. The system is then let to change overtime until the level reaches the equilibrium with the pool level (equivalent to 25 kPa, about 2.5 m) after 7.47 s. Then due to the decay heat from the Fuel Assemblies (considered 3416 W [15]) the temperature slowly rises (Figure 5.38) and the water expands (density diminishes).


(A)

(B)

FIGURE 5.37 – Water level inside the rack as a function of time.

The reactivity changes observed in Figure 5.38 are due to various mechanisms: (a) the initial decrease of the rack level (to equate to the pool level), (b) the variation of water density as the coolant heats-up and (c) the increase of the water level in the rack as a result of the thermal expansion. Note that the thermal expansion leads to an increase in the reactivity but it does not reach its original value. As already discussed, the increase in reactivity is due to, in part, the fact that as the water expands it enters a zone with a



(A)

(B)

FIGURE 5.38 – (A) Temperature increase and (B) Reactivity change as a function of time.

higher view factor with the other racks (H increases but h remains the same). Globally speaking the results are consistent with by hand calculations (for instance the equilibrium water level) and the overall coupling shows a good numerical performance although the initial conditions of the problem are quite stiff.

5.3.4 Final Remarks on the Spent Fuel Pool Study Case

An adequate numerical model for Spent Fuel Pools accidents requires coupled neutronics and thermohydraulics calculations to study all potential scenarios, including the event of a criticality accident. The thermal-hydraulics model is necessary to correctly predict the water levels in the pool and in the racks and should include phenomena such as convective and conductive heat transfer, change from water to air natural convection, boiling and two-phase flow and hydrogen production and combustion from zirconium–steam reaction. The neutronics model should be able to accurate predict the Spent Fuel Pool reactivity and the phenomena associated to a hypothetical criticality accident. In this work, a tool using multi-physics coupling was presented to asses this type of accidents. In the tool, a porous medium model of the Spent Fuel Pool has been developed using OpenFOAM code to solve the thermal-hydraulics equations and couple to a neutronics model based on the Monte Carlo code Serpent. The model was then used to perform a sensitivity analysis on various Spent Fuel Pools parameters using the Spent Fuel Pool of Fukushima Daiichi NPP Unit 4 as a reference.

It was observed from this study that criticality is never achieved even though some parts of the assembly lacks modelling (i.e. spacers, grids). The sensitivity analysis lets us nevertheless identify the mechanisms that could eventually lead to an increase of the Spent Fuel Pool reactivity. Moreover, it was found that under certain conditions the margin to criticality could be significantly reduced: certain levels of burn-up, vapor or void fraction as the fuel pool drains, decrease of soluble boron, among others.

Finally, the multi-physics approach has been used to perform a first calculation of an accident without reaching criticality. Further testing using the full extent of the model is needed for studying the system in the approach to criticality and in the case of a re-criticality. Additionally, this should be tested coupling the porous medium model to a Navier-Stokes multi-phase model outside the rack system and representing, this way, the whole pool system. A general overview on how this could be possible to implement is shown in Appendix D.

Chapter 6

Conclusions

This research has been developed in the framework of the current industry and academia efforts to develop more accurate and flexible numerical tools for nuclear energy application. The work performed during this PhD has allowed developing a new multi-physics and multi-scale numerical tool for the study of criticality accidents. A criticality accident can be defined as an uncontrolled nuclear chain reaction event that occurs in a nuclear system. These accidents can occur in very diverse nuclear systems such as in a fuel factory, during transportation of nuclear materials, in research or power reactors, etc. Therefore developing a suitable numerical tool that can deal with the large spectrum of geometries, materials, coupled phenomena and time scales encountered in a criticality accident is not straightforward. The proposed tool is based on the open-source code OpenFOAM (open-source C++ Library for CFD), and Serpent 2 (Monte Carlo particle transport code). The tool contains various models that have been developed and implemented during this PhD that allow studying different neutronics, thermohydraulics and thermomechanics phenomena and their coupling. The tool uses detailed mechanistic models to reduce the number of approximations and thus obtain Best Estimate predictions.

In order to investigate the accuracy of the numerical models a set of experiments/ benchmarks were identified in the early stages of the PhD as discussed in Chapter 2. While these experiments/ benchmarks are not exhaustive and more comparisons are needed for a full validation of the tool, they have proven to be very useful for improving the precision of the models. They also provide many (but not all) of the phenomena that one could expect to find in criticality accidents. The selection of these experiments/benchmarks was based on a Phenomena Identification and Ranking Table (PIRT) analysis carried-out at the beginning of the PhD. It allowed identifying the following key phenomena in these accidents. Most of these key phenomena (but not all) have been studied in this work and the mathematical models have been presented in Chapter 3. They include:

- Transient neutron transport including super-prompt criticality conditions
- Precursors transport in solid and liquid medium
- Incompressible Navier-Stokes
- Laminar and turbulent flow conditions
- Buoyancy with Boussinesq approximation
- Linear thermoelasticity
- Thermal Expansion with geometry deformation
- Multi-phase multi-component flow in porous medium

Other phenomena will have to be added in the future in order to be able to capture additional scenarios or systems involved in criticality accidents. One can mention: boiling, radiolysis, pressure and acoustic waves, etc. Thank to the strategy used for the development of the tool, these models can be added in a more or less straightforward way.

Once the theoretical models were developed, the next task of the work consisted in their numerical implementation. The details on the algorithms used for the numerical implementation of these models were discussed in Chapter 4. Although the numerical implementation was performed using OpenFOAM and Serpent codes, the proposed coupling algorithms are quite independent of these codes and could be used with other codes. The choice of using the state-of-the-art codes (OpenFOAM and Serpent 2) was made because these codes have large communities of users and developers that continue to maintain and innovate in these codes. In addition, OpenFOAM is open-source. In this regards, it is interesting to note the recent initiative of the International Atomic Energy Agency (IAEA) to create the Open-source Nuclear Code for Reactor Analysis (ONCORE) [120] working group whose objective is to develop the future open-source safety codes. One of the codes used by this working group is OpenFOAM.

In order to perform a first validation and verification of the models and algorithms developed in this PhD, a set of comparisons among the tool predictions and experimental data or independent codes results were done. The results from the comparisons were presented in Chapter 5. Among the various models developed in this work, the Monte Carlo Quasi-Static Method is probably the most original among them, and opens very interesting perspectives for the analysis of criticality accidents. The alternative deterministic method Simplified P_N for the resolution of the neutron transient transport equation was also developed and implemented in the multi-physics tool. The analysis of the results presented in Chapter 5 includes the study of the performance of the Monte Carlo Quasi-Static Method along with a deterministic Simplified P_N method including accuracy and computational cost. Different time and size scales cases ranging from centimeters (Godiva) to meters (hypothetical Spent Fuel Pool accident) with various coupling phenomena (thermal expansion, convection and buoyancy) were explored. The multi-physics tool was also used to perform a benchmark against independent codes on a liquid fuel system (CNRS Molten Salt Cavity Benchmark) in the framework of an European project (SAMOFAR) dealing with the safety studies of the Molten Salt Reactors. Finally, Chapter 5 presented a first approach with a coupled porous medium model to study hypothetical criticality accidents in Spent Fuel Pool. While the comparisons presented in Chapter 5 are not exhaustive and more studies should be performed, the very good agreement found in these exercises shows that the accuracy and the numerical performance of the multi-physics multi-scale tool are adequate. These results do not preclude the possibility of further improvements of some of the models as we will discuss later in this chapter.

As mentioned earlier, one of the novel models developed in this work is the Monte Carlo Quasi-Static Method. As discussed in Chapter 4, this model has been implemented using an internal coupling between Serpent 2 and OpenFOAM for the Adiabatic formulation. We have shown in Chapter 5 that the Adiabatic formulation is able to follow transients with different time scales and amplitudes within a reasonable computational time as illustrated by the analysis of the simulation of the small sphere-shaped Godiva Experiment prompt burst fast transient and the Molten Salt Cavity Benchmark. Some key remarks can be made for the Monte Carlo Quasi-Static Method presented in Chapter 3 and from the results of Chapter 5:

- As expected, the proposed use of a normalized constant weight function for this method does not seem to impact the accuracy of the predictions for the flux, reaction rates and power distributions. This choice significantly simplifies the Monte Carlo Quasi-Static Method equations at the expense of losing the usual meaning of the point kinetic parameters used to solve the flux amplitude equations. This does not appear to be a significant drawback and, if needed, importance or any other weight function can be calculated a posteriori alongside with the point kinetic parameters.

- The analysis of Godiva results (Chapter 5) showed that it is necessary to recalculate the point kinetic equations parameters used to determine the flux amplitude function ($n(t)$) during the flux shape function ($\phi(\vec{r}, \Omega, E, t)$) time step. This allows to significantly reduce the number of neutron shape function calculations (shape time steps) with the Monte Carlo code. In the present work, the following point kinetic parameters were updated during the flux amplitude iterations: the reactivity (ρ), the delayed neutron concentration ($\bar{c}_d$) and the generation time (Λ). The use of a constant weight function allows to use the physical delayed neutron fraction (β_d). These point kinetics parameters were updated during the amplitude iterations based on the neutron fluxes and materials density variations. Other effects affecting the macroscopic cross sections such as the temperature feedback should be implemented in future work along with the precursors shape feedback inside the flux shape time step (i.e. at each amplitude time step), especially in the reactivity. These improvements will allow to further reduce the amount of flux shape calculations. For example, for the Molten Salt Cavity Benchmark, the density update was not enough for obtaining a smooth reactivity and power curves (see Appendix C) in comparison to those obtained for Godiva. Nonetheless, cavity's results were sufficiently accurate.
- It is interesting to note that some authors have suggested using feedback coefficients pre-calculated before the transient as a method to update the reactivity during the flux amplitude iterations [108]. We have found in our studies that the estimation of the reactivity based on the neutron flux and macroscopic cross section variations proposed here is more accurate and effective. In addition, we do not need to know the system feedback a priori.
- As discussed earlier, the multi-physics coupling was implemented internally. Beside improving the speed of data exchange (no read and write of external files), the internal coupling allows both Serpent and OpenFOAM to access all the internal variables fields during runtime. The operations to update the fields can then be treated internally with OpenFOAM routines (direct memory access). This is particularly helpful for some fields such as the flux, the reaction rates, the neutron current and the neutron density that can then be stored and used by OpenFOAM to update (as discussed in the previous point) the point kinetics parameters during the amplitude function

iterations.

- The statistical tolerance of the analyzed system needs to be assessed beforehand. For a fast transient like Godiva, involving an initial reactivity of a few pcm (or cents) over the prompt criticality, a very little standard deviation of the reactivity factor (∼2-4 pcm) will have a significant impact on the transient. For that case, there are two viable options:
  - Increasing significantly the number of neutron histories, while making the flux shape time steps longer and correctly updating the point kinetic parameters of the flux amplitude equations inside the shape step: this option penalizes the calculation time as reported in Table 5.7 but it results in a smoother time-wise solution and more accurate fields spatial distributions (better statistics gathered at each cell of the mesh).
  - Reducing the simulated neutron number while increasing the number of shape time steps: doing so, the randomness of the sampling will compensate the peaks and valleys obtained in the reactivity curve after each shape calculation due to Monte Carlo stochastic nature. The disadvantage of this option is that depending on the cell size of the mesh, the volumetric fields (flux, power, etc) might not be smooth enough and instabilities on the coupling or lower accuracy in the coupled phenomena could occur. This said, this approach takes into account the changes on the flux shape more often -but with higher statistical error.
- It is important to remark that the Adiabatic formulation of the Quasi-Static Method proposed in this work can easily be implemented with any Monte Carlo criticality code to calculate transients and should be seen as complementary with the precise but resource-consuming Dynamic Monte Carlo approaches [37]. Moreover, while we have chosen to use an internal coupling for Serpent and OpenFOAM for the reasons discussed earlier, an external coupling is also possible. Indeed, as long as the power distribution and amplitude parameters' integrals from equations 3.62 can be scored (tallied) and the density and temperature updated in Monte Carlo at the beginning of each flux shape calculation, the implementation can be done externally. This is the case of Monte Carlo codes such as Serpent 2 which, in addition, can update the geometry at each criticality calculation.
- From the results obtained in Section 5.1, one might think that a Dynamic Monte

Carlo is a better option due to the reduced number of particles simulated. However, the memory requirements are significantly higher than the criticality source mode used in this work. This additional cost means that only very short transients with simple geometries and few isotopes and materials can be simulated with standard computers. On the contrary, the Quasi-Static Adiabatic Method has the standard limitations of the k-eigenvalue method. Moreover, the additional time required by the Quasi-Static Method can be decreased with further parallelization, allowing a single routine to solve slow and fast transients.

- In comparison to a deterministic method, the Monte Carlo Quasi-Static Method has the advantage of having very few approximations concerning the transport problem. Additionally, multi-group deterministic methods require to homogenize and condense the neutron cross sections that will have to be updated if the spectrum, flux shape or composition of the materials change. The Monte Carlo codes have the advantage of using continuous energy cross sections libraries and the numerical simulations are continuous in spatial and direction coordinates as well.
- It should be noted that, when numerically solving the Adiabatic formulation of the Quasi-Static Method proposed here, a k-eigenvalue problem is being solved and thus not the actual system. This approximation implies that the nuclear system is at each instant close to the fundamental mode although the fundamental mode shape itself could significantly change over the transient. To consider high order modes the flux shape derivative should not be neglected in the flux shape equation. In most applications related to criticality accidents, this approximation does not introduce a significant error. Other less standard applications such as modeling of Accelerator-Driven Systems (ADS) reactors would require the use of a better formulation than the Adiabatic.
- Regarding the limits of the accuracy of the Adiabatic formulation of the Quasi-Static Method, the development of the method presented in Section 3.1.2 indicates that very large rate of variation of the neutron flux during the transient could lead to inaccurate modeling results. This may result from neglecting the shape and amplitude time derivatives in the flux shape equation 3.60a. Moreover, although the flux shape time step could be reduced, an error associated to the methodology will still be present. However, as studied by Ott et al. in [42], in the case of an event with very high flux

shape deformations caused by a 90 \$ / s reactivity insertion the predictions obtained from the Adiabatic approach are less than 30 % error in some parts of the neutron flux and less than 15 % for the original quasi-static approach (which does not make any assumption of the amplitude change rate, i.e. $dn/dt \neq 0$). Note that this high reactivity insertion scenario is a rather unrealistic configuration.

- As discussed in Chapter 3 an approximation was introduced in the modeling of the delayed neutron precursors in the adiabatic approach: as shown in equation 3.67 the delayed neutrons position are sampled from the precursors concentration. However, for simplicity this equation assumes that the number of emitted delayed neutrons is $\int \beta\nu\Sigma_f\phi dV$ and not $\sum_d \int \lambda_d C_d dV/n(t)$. This approximation does not cause a significant error neither in Godiva experiments nor for the Cavity Benchmark as showed in Chapter 5. However, in other scenarios this approximation should be eliminated by modifying the value of β in the Monte Carlo algorithm that samples the neutrons. This is a straightforward task but requires further modifying the source code (one of the objectives of this work was to be as minimally invasive as possible in the modification of the codes). In the present state, the error caused by this approximation should be less that the one of the classic Adiabatic approach but higher than the original quasi-static approach in transient calculations.
- An improvement in the current methodology could be obtained by adding the dn/dt term as an effective cross section in the flux shape calculations and by correcting the amplitude of the delayed neutrons emission thanks to their sampling on effective β values rather than standard β values. Moreover, the flux shape time derivative ($\partial\phi/\partial t$) could also be taken into account. Nevertheless, as discussed below the accuracy obtained thanks to this improvement is probably not worth the increased complexity of the algorithms. Two solutions could be envisioned to include the effect of the flux shape time derivative ($\partial\phi/\partial t$) in the flux shape equation:
  - Discretization of the term $\partial\phi$ ∂t with an Euler scheme and rearrangement of it in such a way that it could be included as an external source (see Ott et al. [42]). This solution would require the use of the dynamic source mode with the neutron population control as explained in 3.1.1.4. This solution will however introduce a significant complexity.
  - The second option is to use transient Monte Carlo as done by Aufiero et al. for

> Godiva [37] for the flux shape calculation. In addition to the memory issues, the problem of this solution relies on the neutrons scoring process: it is likely that if the time steps were too large, neutrons will either all die out before the end of the step (subcritical system) or diverge (supercritical system). For this reason, transient calculations should be used together with a small time binning (or Monte Carlo "time step") in order to renormalized the neutron source size to a statistically adequate one.

As an alternative method to calculate the neutron flux during the criticality accident, a SP_1 and SP_3 methods were implemented to solve the neutron transient transport equation. It was shown in Chapter 5 that these methods are able to simulate both fast and slow transients such as the Godiva Experiment and the Cavity Benchmark. However, they required the calculation of homogenized and condensed neutron cross sections which can be challenging if the system is very heterogeneous or the geometry change as results of the accident. Moreover in the case of Godiva, the SP_1/ SP_3 methods failed to provide good initial reactivity estimate ($\sim$ 2500 pcm). More studies should be performed in order to determine whether the source of this error comes from: i) the need of using a higher order (>3) Legendre Polynomial to accurate modeling of a high flux anisotropy (due to highly absorbing medium for example), ii) the Marshak boundary conditions approximation or (iii) other SP_N model simplifications such as the within-group approximation explained in Section 3.1.3. Indeed, this last approximation was implemented since it simplifies the equations but it has been mainly used for Light Water Reactor calculations. It is worth to note that the computational cost of the SP_N method specially by using acceleration techniques is nevertheless significantly lower than the Monte Carlo. Moreover, as has been shown in Chapter 5 once the fission source was normalized, Godiva's power burst was correctly reproduced.

On a different physics, the linear thermoelasticity model used to predict the thermal expansion in Godiva seems to be accurate enough. As discussed in Chapter 5 similar results to those of Aufiero et al. [37] were obtained with our tool. OpenFOAM capabilities to perform geometry changes and solve models and map fields between different meshes were key to the development of the thermal-mechanics module. In addition, OpenFOAM can handle dynamic meshing during runtime and the unofficial OpenFOAM extended version includes models for more complex solid mechanics phenomena like plasticity developed by

Cardiff et al. [121].

Concerning the thermohydraulics models, two types were investigated: (i) CFD models solving the laminar Navier-Stoke equation and (ii) porous medium models. RANS models for turbulent flow were also used but are not reported here. The laminar flow model was used for the Molten Salt Cavity Benchmark. This benchmark defined by the CNRS to test multi-physics models for Molten Salt Reactors enabled the investigating of various neutronics and thermohydraulics coupled phenomena (e.g. the precursors transport or the fuel salt density feedbacks). The benchmark was carried-out in the framework of the European H2020 project SAMOFAR (2014-2019) that included various leading European universities and laboratories. For this particular benchmark, we used the incompressible laminar Navier-Stokes model with the Boussinesq approximation already implemented in OpenFOAM. The comparison of the results obtained by our multi-physics tool with those produced by the partners of the European project SAMOFAR (see for example Tiberga et al. [14]) and also in comparison to a FiPy model gives excellent agreement. This exercise illustrates some of the advantages of using OpenFOAM for the multi-physics tool: the tool can be easily modified to change from a laminar to a turbulent flow solver. In addition, the availability of separate solvers with many models already implemented for compressible and incompressible flow makes future improvements of this multi-physics solver easier.

The last case studied in this work concerned a Loss of Coolant Accident in the Spent Fuel Pool of a BWR which allows testing another thermohydraulics model implemented in the tool. The Spent Fuel Pool has the particularity of being a strongly heterogeneous medium (BWR fuel assemblies immersed in a coolant) having solid, liquid and gas phases with potential different averaged temperatures and where water coolant could undergo phase change (pool coolant boiling). This case requires the implementation of a thermohydraulics model coupled to a transient neutronics one that allow calculating the system reactivity and eventual fission power if criticality conditions are reached. The Fuel Assemblies geometry complexity (top and bottom nozzles, grids, fuel rods, etc.) and large number of Fuel Assemblies make it impractical to solve the laminar or turbulent Navier-Stokes equations using a CFD mesh. A rather more practical method was instead the implementation of a porous medium approach. In this approach, the differential conservation equations (continuity, Navier-stokes and Energy equations) of the heterogeneous system (fuel, coolant and pool structures) are homogenized. The equations for the resulting homogeneous medium can then be integrated using a very coarse mesh. This approach shifts the complexity related to the heterogeneous geometry and the microscopic phenomena existing at the microscopic level to the determination of the equivalent homogeneous porous medium properties. In our work, we implemented a multi-phase multi-component porous medium model in OpenFOAM coupled to the other multi-physics modules. This porous medium model was then used to investigate a hypothetical Spent Fuel Pool recriticality accident. While, a broad sensitivity analysis on different potential scenarios for this accident did not allow to identify any credible scenario leading to a recriticality accident in the Spent Fuel Pool, this study demonstrated some of the capabilities and potentialities of the porous model used.

Note that OpenFOAM already has a porous medium model but only for single-phase flow [122]. This original OpenFOAM porous medium model uses the Navier-Stokes equations with a Darcy-Forchheimer source term. The added Darcy-Forchheimer term allows taking into account the pressure drop in the porous medium region. This term contains two parts: the classis Darcy's Law (viscous forces in the porous medium) and the Forchheimer term that allows to take into account inertial forces in the pore (e.g. turbulence). Nevertheless this model does not allow to study all the phenomena existing in the Spent Fuel Pool which involves multi-phase flow (e.g. water as vapor or liquid) and different chemical species (e.g. water, air, boron). In addition, local non-equilibrium conditions between the solid and fluid phases will occur when significant heat generation (e.g. fission energy if recriticality) exists requiring different temperatures for porous medium phase (solid, liquid and gaz) and non-equilibrium mass transfer (e.g. Henry and Raoult's law do not apply). Therefore, the more comprehensive porous medium model described in Section 4.3.3.2 was implemented. This model is based on the work of Horgue et al. [19]. This model was not fully used in this work because of the lack of some required input data and also some convergence issues. These issues together with potential solutions are discussed in Appendix D together with future work related to the possible multi-region coupling between OpenFOAM's Navier-Stokes Volume of Fluid (VOF) multi-phase model and the multi-phase multi-component porous medium with the interface conditions between regions.

Finally we can summarize the main prospects for improvement of the multi-physics tool: (i) implementation of the precursors transport and temperature feedback inside the flux shape time step; (ii) addition of the Original Quasi-Static Method terms to improve the Monte Carlo QSM; (iii) relaxation of some of the approximations of the Simplified $\mathrm{P_N}$ Method to improve accuracy on the steady-state; (iv) inclusion of more advanced Solid Mechanics methods;(v) further testing of the new capabilities in the porous medium model with sufficient data; (vi) inclusion of the VOF method in a multi-region coupling for the Spent Fuel Pool case; and (vii) optimization of the solver routines, specially, of the cell search algorithm which penalizes greatly the Monte Carlo routines.

Future works should also enhance and complete this work with the addition of other phenomena such as: radiolysis, pressure waves and free liquid surface, in order to cover a larger spectrum of nuclear systems and hazards.

Appendix A

RANS Methods Calculations

In this Appendix, the discussion from Sections 3.3.5 and 4.3.3.1 the multi-physics tool was extended to include the case of turbulence modeling. Indeed, the multi-physics tool can be used to study other reactor concepts. For example the turbulence flow model was employed for the particular case of the evaluation of a Space Nuclear Reactor Design tutorial. This tutorial was developed by the Reactors Physics Group from the LPSC. Only the $k-\epsilon$ and $k-\omega$ SST models are discussed in this appendix since the objective here is merely to mention the turbulence modeling capabilities of the multi-physics tool has already been implemented using the OpenFOAM environment. This is not an extensive analysis neither on the turbulence models nor on the Space Nuclear Reactor Design tutorial.

In Section A.1 the $k-\epsilon$ and $k-\omega$ SST models are briefly explained and in Section A.2 the application case is set as a general example.

A.1 RANS methods

As stated in Section 3.3.5, when the Boussinesq hypothesis is used, the RANS problem consisting in the determining the Reynolds Stress Tensor $\overline{\overline{\tau}}^R$ is reduced to the calculation of the turbulent eddy viscosity μ_t and the turbulent kinetic energy k. In addition, to solve the flow energy balance equation, the turbulent thermal diffusivity must also be calculated. The Boussinesq hypothesis used in this approach assumes that μ_t is an isotropic scalar quantity, which is not strictly true [123]. One of the common methods to calculate the turbulent eddy viscosity μ_t and the turbulent kinetic energy k is to use a Reynolds Stress Transport Model. The discussion of the basis of this type of approach is out of the scope of this manuscript but can be found in [123].

Among the various RANS models using the Boussinesq hypothesis the Spalart-Allmaras, $k-\epsilon$ and $k-\omega$ models can be named. In this work, $k-\epsilon$ and the $k-\omega$ *SST* were used based on previous sensitivity studies and will be then described in the rest of this appendix.

The $k-\epsilon$ model of Jones and Launder [124] is the most popular and considered as the standard $k-\epsilon$ model. The turbulent viscosity and thermal diffusivity are defined as:

$$\mu_t = \rho C_\mu \frac{k^2}{\epsilon} \tag{A.1}$$

$$k_t = \frac{c_p \mu_t}{Pr_t} \tag{A.2}$$

where ρ is the density, c_p is the specific heat capacity at constant pressure, Pr_t is the turbulent Prandtl, k is the turbulent kinetic energy defined in Section 3.3.5 and ϵ is the rate of dissipation of turbulence kinetic energy per unit mass due to viscous stresses defined as:

$$\epsilon = \frac{1}{2}\frac{\mu}{\rho}\overline{\left\{\nabla\vec{u}' + (\nabla\vec{u}')^T\right\} : \left\{\nabla\vec{u}' + (\nabla\vec{u}')^T\right\}} \tag{A.3}$$

where μ is the molecular viscosity coefficient and : is the double dot product.

To solve for k and ϵ and then be able to close the RANS equations, these quantities are calculated from the following transport equations:

$$\frac{\partial}{\partial t}(\rho k) + \nabla\cdot\left(\rho\overline{\vec{u}}k\right) = \nabla\cdot\left(\mu_{eff,k}\nabla k\right) + P_k - \rho\epsilon \tag{A.4}$$

$$\frac{\partial}{\partial t}(\rho \epsilon) + \nabla\cdot\left(\rho\overline{\vec{u}}\epsilon\right) = \nabla\cdot\left(\mu_{eff,\epsilon}\nabla \epsilon\right) + C_{\epsilon 1}\frac{\epsilon}{k}P_k - C_{\epsilon 2}\rho\frac{\epsilon^2}{k} \tag{A.5}$$

where

$$\mu_{eff,k} = \mu + \frac{\mu_t}{\sigma_k} \tag{A.6}$$

$$\mu_{eff,\epsilon} = \mu + \frac{\mu_t}{\sigma_\epsilon} \tag{A.7}$$

$$P_k = \tau^R : \nabla\overline{\vec{u}} \tag{A.8}$$

with P_k being the production of turbulent energy. The dimensionless coefficients $C_{\epsilon 1}$, $C_{\epsilon 2}$, C_μ, σ_k, σ_ϵ and Pr_t are defined in Table A.1.

The $k-\epsilon$ model assumes a fully turbulent free shear flows, then, it is valid only for high enough Reynolds numbers such as the molecular viscosity is negligible. It cannot

TABLE A.1 – Standard values for the $k - epsilon$ model.

Coefficient	$C_{\epsilon 1}$	$C_{\epsilon 2}$	C_{μ}	σ_k	σ_ϵ	Pr_t
Value	1.44	1.92	0.09	1.0	1.3	0.9

be integrated up to the wall where the destruction-of-dissipation term is singular, and thus a fine grid along with wall functions (discussed in Section 4.3.3.1) is needed to yield reasonable solutions for this model.

To overcome the limitation of the ϵ model near the wall, first Kolmogorov [125] and, then, Wilcox [126, 127] transported the turbulent kinetic energy k and the specific turbulent dissipation ω defined as:

$$\omega = \frac{\epsilon}{\beta^* k} \tag{A.9}$$

to yield the well-known $k - \omega$ model. Even though, it performs better near wall with segregated flow, it shows some sensitivity for the free stream ω specified values. Then, the solution highly depends on this arbitrary specification.

Menter [128] combined the $k - \epsilon$ and $k - \omega$ formulation to construct a new model that behaves like the $k - \omega$ near the wall and like the $k - \epsilon$ in the free-stream. It is known as $k - \omega$ SST model which consists in a system of two equations that have to be solved for k and ω as follows:

$$\frac{\partial}{\partial t}(\rho k) + \nabla \cdot \left(\rho \bar{\vec{u}} k\right) = \nabla \cdot (\mu_{eff,k} \nabla k) + \tilde{P}_k - \beta^* \rho k \omega \tag{A.10}$$

$$\frac{\partial}{\partial t}(\rho \omega) + \nabla \cdot \left(\rho \bar{\vec{u}} \omega\right) = \nabla \cdot (\mu_{eff,\omega} \nabla \omega) + \tilde{C}_\alpha \frac{\omega}{k} P_k - \tilde{C}_\beta \rho \omega^2 + 2(1 - F_1)\sigma_{\omega 2} \frac{\rho}{\omega} \nabla k \cdot \nabla \omega \tag{A.11}$$

where F_1 is a blending function affecting also all the coefficients in the model as:

$$\tilde{\Phi} = F_1 \Phi_1 + (1 - F_1)\Phi_2 \tag{A.12}$$

with F_1 defined as:

$$F_1 = tanh\left(\gamma_1^4\right) \tag{A.13}$$

$$\gamma_1 = Min\left(Max\left(\frac{\sqrt{k}}{\beta^* \omega \left(d_\perp\right)}, \frac{500\nu}{\left(d_\perp\right)^2 \omega}\right), \frac{4\rho\sigma_{\omega 2} k}{CD_{k\omega}\left(d_\perp\right)^2}\right) \tag{A.14}$$

$$CD_{k\omega} = Max\left(2\rho\sigma_{\omega 2}\frac{1}{\omega}\nabla k \cdot \nabla \omega, 10^{-10}\right) \tag{A.15}$$

and the turbulent viscosity is bounded :

$$\mu_t = \frac{\rho a_1 k}{Max\left(a_1\omega, \sqrt{2}\sqrt{S_t \cdot S_t}F_2\right)} \tag{A.16}$$

$$F_2 = tanh(\gamma_2^2) \tag{A.17}$$

$$\gamma_2 = Max\left(2\frac{\sqrt{k}}{\beta^*\omega\left(d_\perp\right), \frac{500\nu}{\left(d_\perp\right)^2\omega}}\right) \tag{A.18}$$

$$S_t = \frac{1}{2}\left(\nabla\overline{\overline{u}} + \left(\nabla\overline{\overline{u}}\right)^T\right) \tag{A.19}$$

Additionally, the production of turbulence kinetic energy P_k is modified in the k-equation A.10 by:

$$\tilde{P}_k = Min\left(P_k, c_1\epsilon\right) \tag{A.20}$$

Finally, the turbulent thermal conductivity and the effective turbulent viscosities are calculated as:

$$k_t = \frac{\mu_t}{Pr_t} \tag{A.21}$$

$$\mu_{eff,k} = \mu + \frac{\mu_t}{\tilde{\sigma}_k} \tag{A.22}$$

$$\mu_{eff,\omega} = \mu + \frac{\mu_t}{\tilde{\sigma}_\omega} \tag{A.23}$$

The coefficients normally used in the $k-\omega$ SST model are shown in Table A.2. A detailed description on the actual implementation of a very similar set of equations but based in Menter [128] can be found in [129]. In the same way, slightly changes on the coefficients used in the $k-\epsilon$ model and the walls conditions for both models can be found also in [129]. More details on turbulent modeling can be found in Moukalled et al. [64] and for the case of turbulent molten salt flows a significant amount of work has been performed in the thesis work of Mauricio Tano Retamales [18] at LPSC.

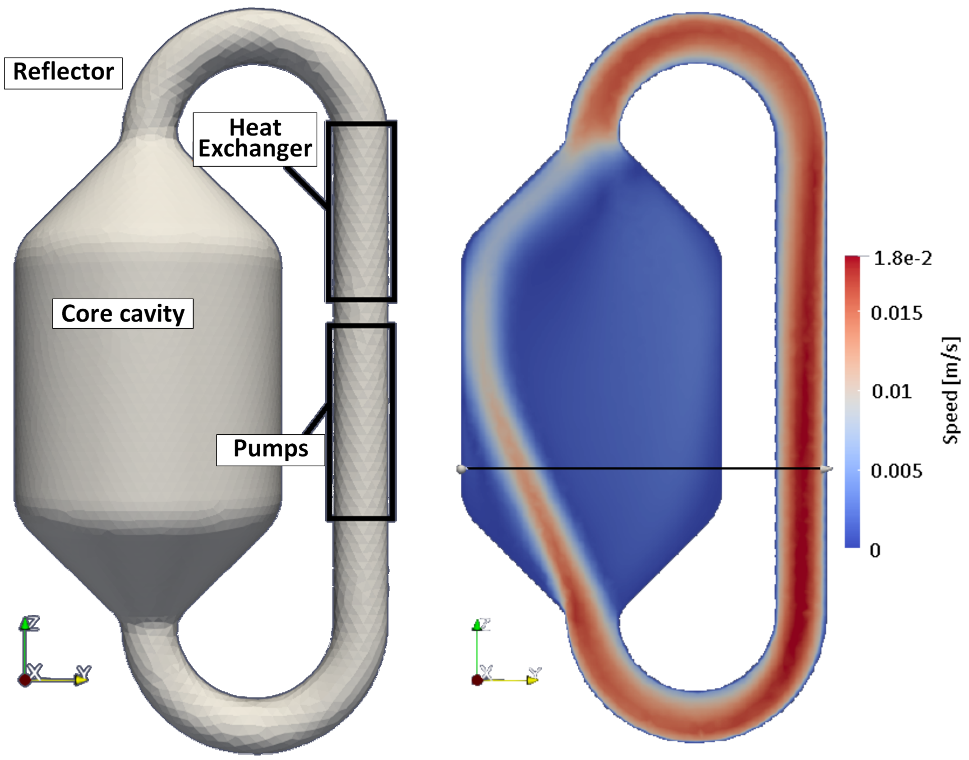
TABLE A.2 – Standard values for the $k-\omega$ SST model.

Coefficient	$C_{\alpha 1}$	$C_{\beta 1}$	σ_{k1}	$\sigma_{\omega 1}$	β^*	c_1
Value	0.5532	0.075	2	2	0.09	10
Coefficient	$C_{\alpha 2}$	$C_{\beta 2}$	σ_{k2}	$\sigma_{\omega 2}$	Pr_t	a_1
Value	0.4403	0.0828	1	1.186	0.9	0.31

A.2 Space Nuclear Reactor Design Tutorial

The same strategy that has been presented for the Molten Salt Cavity Benchmark explained in Section 2.2 and whose results have been shown in Section 5.2, has been used to develop a tutorial for a multi-physics design tool for q Nuclear Electric Propulsion (NEP) engine based on a Molten Salt Reactor concept. This work takes advantage of the experienced gained by the Reactor Physics Group of the LPSC on numerical and experimental modeling of molten salts.The multi-physics tool used in this PhD work was also used for the development of a tutorial for space reactors.

The geometrical model of this tutorial is not meant to be realistic but, as said before, to covered as much of the relevant phenomena as possible: laminar/turbulent flow, Conjugated Heat Transfer, neutronics, phase change (solidification/fusion), and radiative heat transfer, for example. In Figure A.1(A) the 3D simplified geometry is shown consisting in the core cavity, the Heat Exchanger (HX), the pumps and the reflector. The core cavity has a diameter of 0.5 m and height of ca. 1 m and the pipe diameter is 0.1 m. The reflector is made of Beryllium and the working fluid is FLiNaK molten salt with U-235.



(A)

(B)

FIGURE A.1 – (A) Simplified geometry of the Space Nuclear Reactor Design tutorial and (B) Speed field calculated with the turbulence $k - \omega$ SST model plot on the x-z mid plane.

The main steps in the tutorial [71] are shown in Table A.3 and a comparison for the Step 1 fixing the momentum source (pumps) using the $k-\epsilon$ and $k-\omega$ SST turbulence models is presented in Figure A.2. The speed field of the $k-\omega$ SST is observed in Figure A.1(B).

Table A.3 – Steps for the Space Nuclear Reactor Design tutorial.

Step	Physics	Flow Type	Time
1	**Hydraulics Stand-Alone**	Laminar Turbulent	Steady-State Transient
2	**Thermal-Hydraulics (Imposed Power Field)**	Laminar Turbulent	Steady-State Transient
3	**Conjugated Heat Transfer**	Laminar Turbulent	Steady-State Transient
4	**Neutronics Coupling (Full Coupling)**	Laminar Turbulent	Steady-State Transient
5	**Phase Change**	Laminar Turbulent	Transient

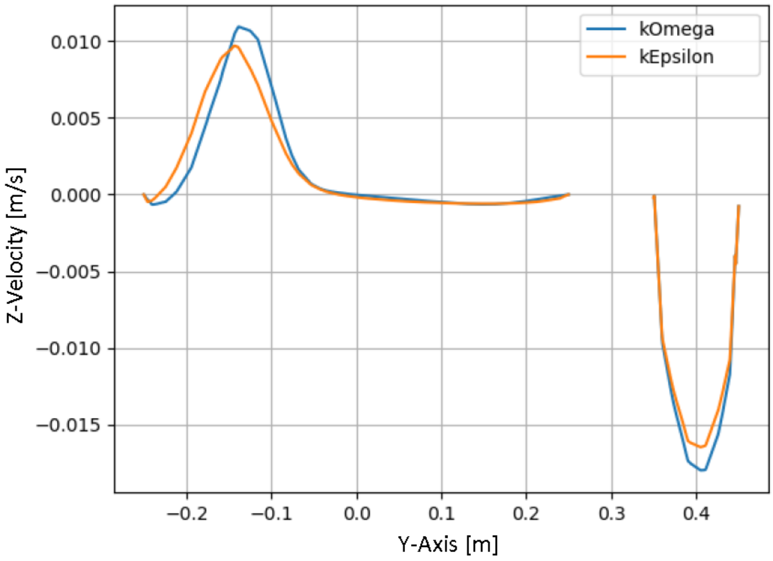


Figure A.2 – Z-component of the velocity field for the horizontal line marked in Figure A.1(B) with the $k-\epsilon$ and $k-\omega$ SST turbulence models (Source: [130]).

This tutorial was presented at the ANS Annual Meeting 2020 [71] and the results presented here were performed by Richard [130] using the multi-physics tool developed in this work and in the thesis work of Mauricio Tano Retamales [18].

Appendix B

Time Control routines

In Section 5.1.8 it was pointed out that an adaptive time step strategy could optimize the simulation by determining the necessary flux shape and amplitude time steps as well as their respective sizes. Motivated by this, a **time control C++ Class** was implemented in OpenFOAM based on the ideas of the QX1 code [112] from the Argonne National Laboratory and adding some new features.

As discussed in this manuscript, due to the sequential nature of OpenFOAM the time step size value is determined in a predictive way. This is, the amplitude time step size is determined for the following time step, and a flag is set to determine (also before the end of the step) if the shape calculation has to be performed. Different criteria has been set for the user to adapt the time step. The overall strategy is described in this Appendix.

Remembering the algorithms described in Section 4.3.1.2 and the time step scheme from Figure B.1 the criteria used for choosing each time step can be enumerated.

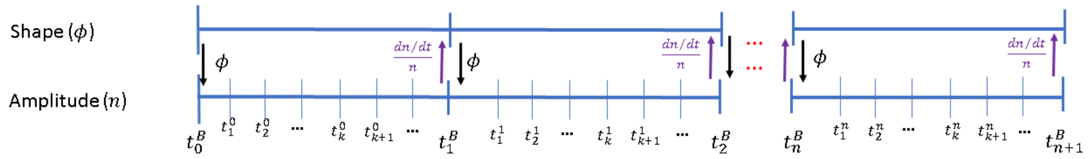


FIGURE B.1 – Time Step Scheme.

For determining the amplitude time step two (2) options are available:

1. **Logarithmic fitting:** the last three calculated amplitude values are saved and a

logarithmic fitting of the constants $(a,\ b,\ c)$ of the following equation are obtained:

$$log\ n(t) = a + bt + ct^2 \tag{B.1}$$

Then, the allowed error is set via the user-set constant ϵ_n as:

$$10^{-\frac{1}{\epsilon_n}} \leq \frac{n\left(t_{i+1}^n\right)}{n\left(t_i^n\right)} \leq 10^{\frac{1}{\epsilon_n}} \tag{B.2}$$

Thus, the following estimation of the next amplitude time step is obtained:

$$\delta t_{i+1}^n = \frac{log\ 10}{\epsilon_n} |b + 2ct_i^n|^{-1} \tag{B.3}$$

2. **Linear fitting:** Using the same reasoning as before but with a linear extrapolation the fitting can be expressed as:

$$n(t) = \frac{dn(t_0)}{dt}\left[t - t_0\right] + n(t_0) \tag{B.4}$$

Then, the amplitude value is allowed a relative change of ϵ_l also set by the user:

$$\delta n = \left|\frac{\Delta n}{n}\right| \leq \epsilon_l \tag{B.5}$$

The amplitude time step is predicted by:

$$\delta t_{i+1}^n = \epsilon_l \left|\frac{dn/dt}{n}\right|^{-1} \tag{B.6}$$

The user can choose either one of this options in the adaptive mode or set a fixed value for the amplitude time step.

For determining the flux shape time step five (5) options are available:

1. **Shape Variation:** A strategy similar to the linear fitting of the amplitude equation is done. The shape field of the last two shape time steps are saved. Then a cell-wise linear fitting is performed and the j-th cell coefficient d_j and e_j are calculated for each cell to form the following equation:

$$\phi_j(t) = d_j t + e_j \qquad \forall j \in V \tag{B.7}$$

where ϕ_j is the j-th cell value of the shape function and V the total volume of the domain. The error criteria is set through the user-defined constant ϵ_ϕ as:

$$max_j \left| \frac{\phi_j(t^B_{i+1}) - \phi_j(t^B_i)}{\phi_j(t^B_{i+1})} \right| \leq \epsilon_\phi \tag{B.8}$$

This way, the shape time step is determined with the values of d_j and e_j for the cell j that maximizes equation B.8:

$$\Delta t^B_{i+1} = \frac{\epsilon_\phi}{1-\epsilon_\phi}\left(t^B_i + \frac{e_j}{d_j}\right) \tag{B.9}$$

2. **Delayed Neutrons Variation:** The precursors variation between shape time steps is treated the same way as for the shape. Then, the shape field $\phi(\vec{r}, t)$ is replaced by the delayed neutron source field $(\sum_d \lambda_d C_d(\vec{r}, t))$ and the user-defined constant for the error is now ϵ_d.
3. **Reactivity Variation:** The local reactivity field is calculated as:

$$\rho_j = \frac{(\nu\Sigma_f\phi)_j}{(\Sigma_a\phi + \nabla\cdot\vec{J})_j} \qquad \forall j \in V \tag{B.10}$$

where $(\nu\Sigma_f\phi)_j$, $(\Sigma_a\phi)_j$ and $(\nabla\cdot\vec{J})_j$ are the j-th cell value of the fission rate, absorption rate and divergence of the neutron current respectively, and V is the total volume of the domain. Then, the same procedure as for the shape variation is done replacing the shape field $\phi(\vec{r}, t)$ by the local reactivity field $\rho(\vec{r}, t)$ and the user-defined constant for the error is now ϵ_ρ.

4. **Amplitude Test:** This test uses the same logarithmic fitting as for the amplitude time step criteria number 1 but in equation B.2 it replaces the amplitude time value t^n_i by the last registered shape function time value t^B_i, and the next amplitude time value t^n_{i+1} by the next shape time value t^B_{i+1} to be estimated. This way the new shape time step size is estimated similarly to equation B.3 but replacing ϵ_n by the user defined constant ϵ_B as:

$$\Delta t^B_{i+1} = \frac{log\ 10}{\epsilon_B}\left|b + 2ct^B_i\right|^{-1} \tag{B.11}$$

5. **Inverse Period:** the inverse period value ($\alpha_r = 1/T_r$) is calculated as the time

derivative of expression B.1:

$$\begin{aligned} \alpha(t) &= \frac{d\ log\ n(t)}{dt} \\ &= \frac{1}{n(t)} \frac{dn(t)}{dt} \\ &= b + 2ct \end{aligned} \tag{B.12}$$

Then the error criteria to be satisfied requires setting the user-defined value ϵ_α shown in the following equation:

$$\left|\alpha(t^B_{i+1}) - \alpha(t^B_i)\right| \leq \epsilon_\alpha \tag{B.13}$$

The shape time step is, therefore, calculated as:

$$\Delta t^B_{i+1} = \left|\frac{\epsilon_\alpha}{2c}\right| \tag{B.14}$$

Each of these criteria can be turn on and off separately. If more than one is selected, the time control class chooses the minimum time step among them. All of these estimations of the shape time step are evaluated at each amplitude time step since parameters a, b and c in equation B.1 are always being updated with the last three amplitude values.

In addition, limiters on the values of the time steps can be set, either relative or absolute. Relative limiters forbids any of the steps to be higher than the double of its last value or smaller than half its last value. The absolute limiter is set directly by the user, who indicates the maximum and minimum values tolerated for the amplitude and shape time steps.

The values of the amplitude and shape time steps can be set to fixed values (this is the option by default). The shape time step size can be set non-uniform by a list of values given at the input (like in the case of Godiva from Section 5.1).

In the listing B.1 an example of the user-interface for the time control is shown. One additional feature that should be considered is the Courant number time control ($\delta_t = Co\Delta x/\left|\vec{u}\right|$) already implemented in OpenFOAM for fluid mechanics where δx is the mesh size. This value is calculated in all the domain for each mesh and the minimum is chosen. This way, the minimum between the amplitude time step and the Courant time step should be prioritized.

Listing B.1 – Time Control User Input example

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  5.x                                   |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      timeParameters;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

/************** FIXED STEP *************/
fixedStep               no;
// Shape Step Value (s)
formStep                100.e-6;
//Amplitude Step Value (s)
amplitudeStep           0.5e-6;

/************** TABLE STEP *************/
formTable               yes;
// Table Values
timeTable               (
                        100.e-6
                        185.e-6
                        195.e-6
                        205.e-6
                        210.e-6
                        220.e-6
                        230.e-6
                        245.e-6
                        270.e-6
                        300.e-6
                        );
// amplitude value for table
amplitudeTable          0.5e-6;

/*********** AMPLITUDE CONTROL *********/
// logarithmic OR lineal
```

```
ampStepControl          logarithmic;
// Logaritmic amplitude time step estimation parameter
epsilonLogAmp           10.;
// Lineal amplitude time step estimation parameter (maximum relative error)
relErrAmp               1.e-1;

/************ SHAPE CONTROL **************/
// Shape Function Change Test
shapeFunct              yes;
relErrForm              0.02;
// Prec Function Change Test
precFunct               yes;
relErrPrec              1.;
// React Function Change Test
reactFunct              yes;
relErrReact             0.001;
// Amplitude Function Test
ampTest                 yes;
relErrAmpForm           1.;
// Inverse Period Test
invPeriodTest           yes;
inversePeriodParameter  1.e+6;
// Tolerance in the case amplitude arrives too close to the shape step
// value and the resulting amplitude step is nearly 0.
matchTolerance          1.e-12;

/************* LIMITERS *****************/
// Limits Amplitude and Shape Steps to either half of double from its previous value
relativeLimiter         yes;
// Limits Amplitude and Shape Steps to max or min values set
maxMinLimiter           no;
deltaAmpMax             1000;
deltaAmpMin             1;
deltaFormMax            1;
deltaFormMin            0.1;

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
```

Appendix C

Cavity Additional Results

In this Appendix the full results discussed in Section 5.2.1 for the Step 1.4 of the Molten Salt Cavity Benchmark are presented followed by a comparison of the transient evolution between the SP$_3$ and the OF-PCQM for a perturbation period of 40 s are discused (Section C.2).

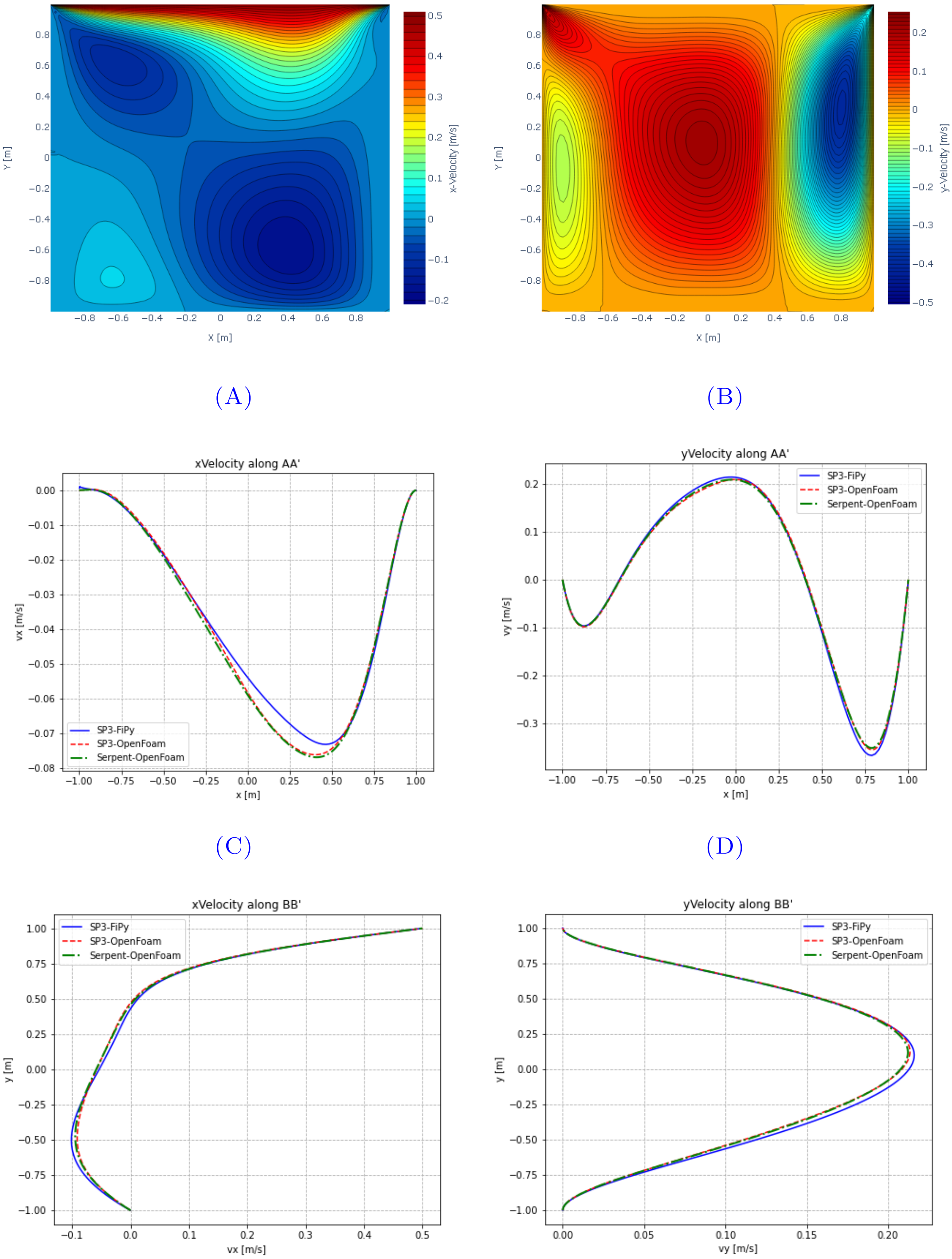
C.1 Steady-State: Step 1.4 Results

In Section 5.2.1 the steady-state results for the MSCB were presented and the 2D contour map results for the Step 1.4 were shown. The mentioned comparison along the AA' (x-axis) and BB' (y-axis) are presented in this Appendix in Figure 2.3(A). Moreover, Figure C.2 shows the x- and y-components contour maps and the their values along AA' and BB', respectively. The AA' and BB' plots for the fission rate, temperature and delayed neutrons source are in Figure C.2. All curves still shows a correct agreement between the methods.

C.2 Transient Results

As stated in Section 5.2.2 and Chapter 6 an important source of error is the correct update of the kinetic parameters (equations 3.62) inside the flux shape time step, also studied in Prince et al. [108] and Patricot et al. [47].

During the cavity transient, the flux shape does not change much but the distortion of the precursors' distribution shape have an impact on the kinetic parameters during the amplitude calculations. These changes on the precursors distribution are taken into account



(A)

(B)

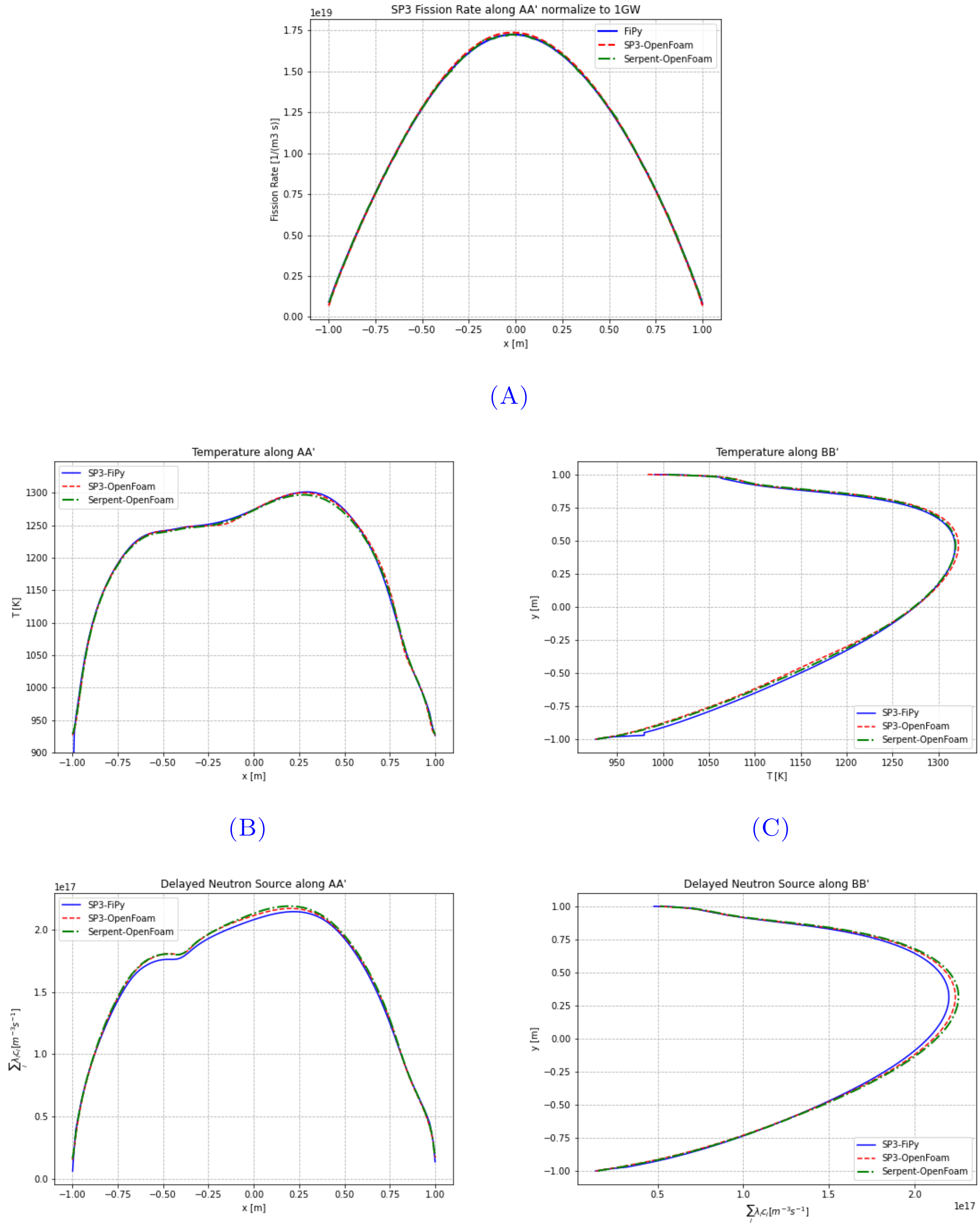
(C)

(D)

(E)

(F)

Figure C.1 – Step 1.4: Velocity (A) X-Component, (C) Y-component, (E) X-component along AA’, (D) Y-component along AA’, (D) X-component along BB’ and protect(D) Y-component along AA’ calculated in FiPy and OpenFOAM for SP3 (in FiPy and OpenFOAM) and Serpent-OpenFOAM internal coupling models.



(A)

(B)

(C)

(D)

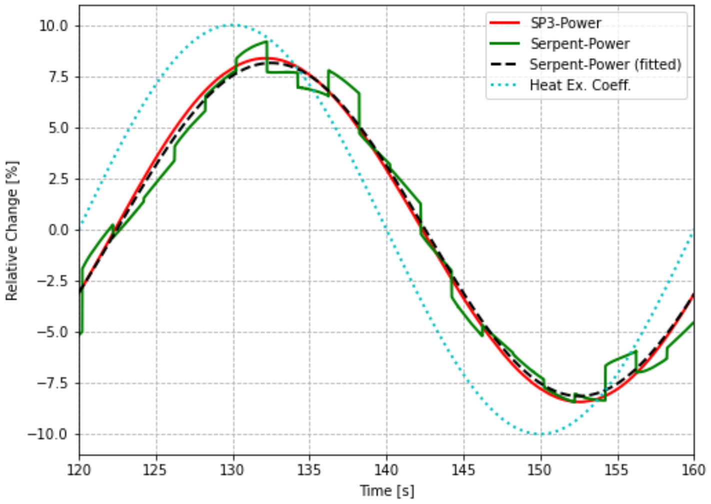
(E)

Figure C.2 – Step 1.4: (A) Fission Rate, (B) Temperature along AA’, (C) Temperature along BB’, (D) Delayed Neutron Source along AA’, (E) Delayed Neutron Source along BB’ calculated in FiPy and OpenFOAM for SP3 (in FiPy and OpenFOAM) and Serpent-OpenFOAM internal coupling models.

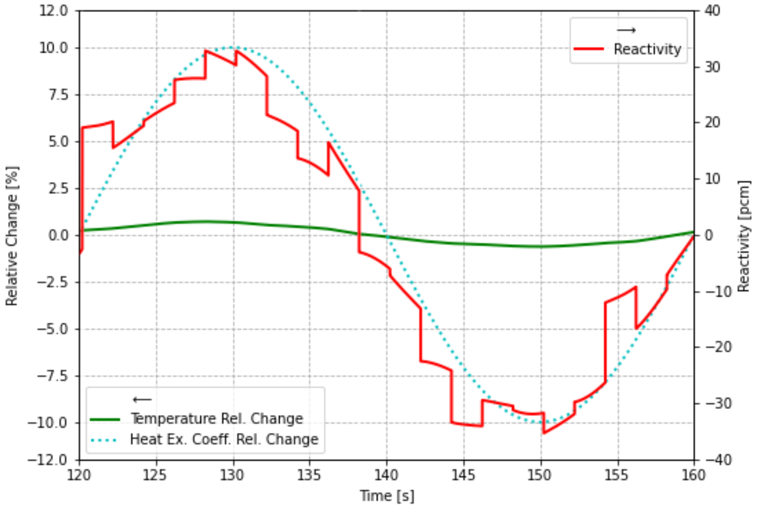
for the flux shape calculations but not for the PK parameters update during the amplitude calculations. Thus, a non-smooth evolution is predicted for the reactivity as can bee seen on Figure C.3. Density correction is also needed, and it was indeed taken into account by saving all fields cell-wise as explained in Section 4.3.1.2.3 and correcting reaction rates by the relative density ρ_k defined in equation 4.4 and recalculating the kinetic parameters. Nonetheless, this has not been enough to correct this step-wise reactivity. Then, it should be considered adding such precursor shape update inside the amplitude time step.

Eventually, Doppler correction on the reactivity would also be needed but as explained in section 2.2 the benchmark was designed to avoid doing so. Temperature effect can be activated in the solver to account for it during the shape calculations. However, as explained in Section 5.1.1, accounting for it during the amplitude time step has some complexity.

Finally, it should be noted that the stochastic nature of Monte Carlo simulations explain the fluctuations of the reactivity and could be improved by increasing the number of neutron histories simulated.



(A)



(B)

Figure C.3 – Power relative change during the transient evolution comparison for the neutronics models (SP3 in OpenFOAM and Serpent-OpenFOAM Monte Carlo Quasi-Static Method with internal coupling) and heat exchange coefficient relative change for a period of $T = 40\ s$.

Appendix D

Navier-Stokes and Porous Medium coupling

It was mentioned in Section 3.3.7 that the porous medium model implemented in this work is a multi-phase multi-component one. However, in Section 5.3 thermal and species equilibrium conditions were set and it was not exploited to is maximum performance for a couple of motives that will be enumerated in Section D.1.

In addition, a more comprehensive model to study the Spent Fuel Pool from Chapter 5 should include a multi-phase Navier-Stokes model for the free stream (i.e. the water in the pool outside the racks) as mentioned in the perspectives of Chapter 6. For this purpose, already existing algorithms in OpenFOAM can be added to the multi-physics tool. This models are briefly introduced in Section D.2.1 and the interface conditions necessary to coupled the phenomena between the porous medium model (inside the racks) and the free stream (outside the racks) are shown in Section D.2.2.

D.1 Porous Medium Model

The most crucial aspect to use a porous medium model to its full extent is the **input data**. The complexity in geometrical modeling, mesh refinement and computational cost from solving Navier Stokes in a FA (heterogeneous system) can be reduced when a REV is used to homogenise the equations. However this approach only switch the problem complexity from the geometry description (mesh) to the determination of the porous medium equations. As a consequence, the link between microscopic phenomena is often done through parameters that are empirically (or pseudo-empirically) correlated to macroscopic variables. For instance, the dimensionless quantities defined in Table 3.1 for mass and heat transfer are normally based on pseudo-empirical correlations as functions of the main variables (velocity, temperature, pressure). These correlations are not unique since they depend on the amount of phases, the phase material, geometry, type of flow, variables values range, among others. For this reason, depending on the type of problem a different correlation is needed (if it even exists). As discussed in Section 3.3.7.1 the relationship between the macroscopic and microscopic diffusion coefficient has to take into account the presence of different phases and the porous structure of the system, the model presented in that section is for soil studies and it should be studied its application to other systems. The non-equilibrium of mass requires the knowledge of the equilibrium concentration for each species in each phase, which can be obtained from equilibrium relations such as Henry's and Raoult's law but a more thourough option would be to account for the equilibrium of chemical potential taking into account different phase pressures (due to capillarity) and temperatures (due to heat generation). This latter equilibrium discussion also applies to all thermodynamics properties that are needed no be closed with corresponding state relationships. For the relative permeability and capilarity calculations some models are needed, being the most known of them the Brooks & Corey model and the Van Genuchten model as explained in Section 3.3.7.5, however this were derived for soils and not necessarily applicable to Spent Fuel Pools. Finally, the information on the interfacial areas between phases is needed to be estimated as function of the saturation or by proposing more complex models based on lattice Boltzmann for example.

Once all of these considerations are taking into account, the model consists in a thirteen (13) equations system with thirteen (13) unknowns (2 saturations, 4 species concentrations, 2 velocities, 3 temperatures and 2 pressures fields) for a three phase (solid, wetting, non-wetting) and two species per phase (H_2O and Air). Thus, the numerical stability in such a system (which can also be compressible) is not trivial. Specially, the treatment of sharp interfaces requires very robust discretization schemes for the jumps present in all properties of the system. Just as Horgue [19] did, all parameters are interpolated from the centers to the face in order to smooth the sharp edges. Additionally, the IMPES method needs to be tested and evaluate its performance in the mass and thermal non-equilibrium case.

In the case of a recriticality accident in the SFP, boiling and evaporation will be important to asses the neutronical coupling between racks, as it was studied statically with the Air-Vapor mixture in Section 5.3.2. Then, the non-equilibrium model will address this aspect of the transient phenomena . In addition, when a significant heat generation, such as fission power, is present in the system, temperatures are no longer in equilibrium and multiple energy balances are required setting a temperature for each phase.

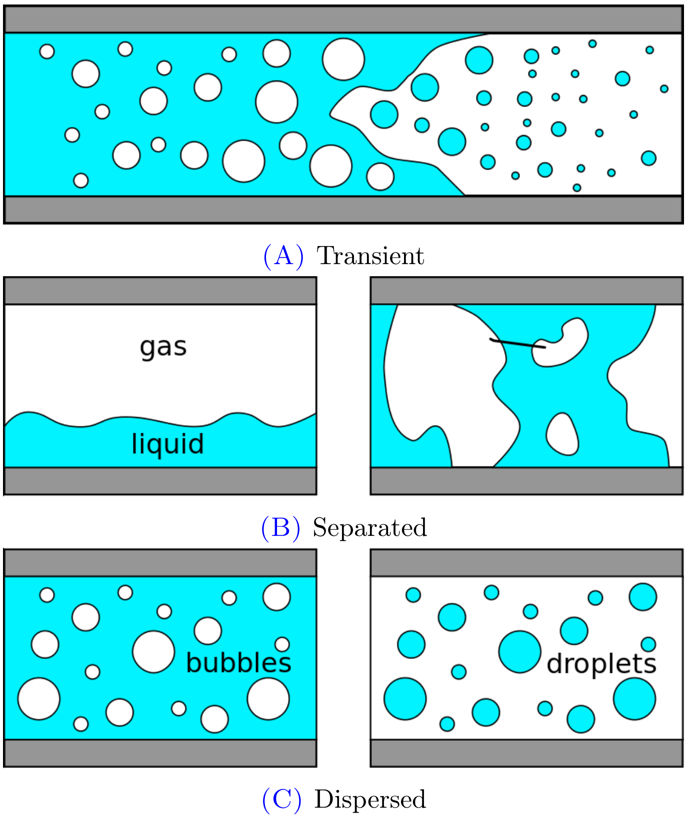
D.2 Spent Fuel Pool comprehensive transient

A more comprehensive model to treat the phenomena in a SFP accident will require the interaction with the pool system. This is, everything outside the rack, to which we will refer to as the free stream. Outside the rack, the complexity in the geometry does not exist anymore and we can once again based our analysis in the Navier-Stokes equations. For this purpose we would need to consider mainly two aspects: the type of flow we want to model and the coupling conditions with the porous medium model at the interface. The first aspect will be addressed in Section D.2.1 and the latter in Section D.2.2.

D.2.1 Multi-phase models

During a LOCA/LOFA different situations can be encountered with one or more multi-phase flow patterns. For simplicity, we are going to analyse the case of a two-phase flow, but this behaviour can be extended to more phases. The flow pattern depends on the properties of the phases, the fraction of the phases (volume % of the phase in the domain of interest) present and the flow condition. If one of the phases is present in much larger amount than the other (i.e. the fraction is a lot larger) it can be considered as a *background* phase with the other phase *dispersed* in it, for example if the gas phase is the dominant one there will be droplets dispersed in it, and opposite, if the liquid phase is dominant, bubbles will be dispersed in it. This is called a **dispersed flow** as it can be observed in Figure D.1. If the fraction difference is not large between the phases, then significant interfacial areas will be formed between them producing, depending on the shear stress rate, transient or separated two-phase flow. More on two-phase flow can be found in Levy [131].

As stated in Adorni et al. [15] there can be considered four different scenarios which may affect the FAs during a SFP LOCA/LOFA: completely covered, partly uncovered



(A) Transient

(B) Separated

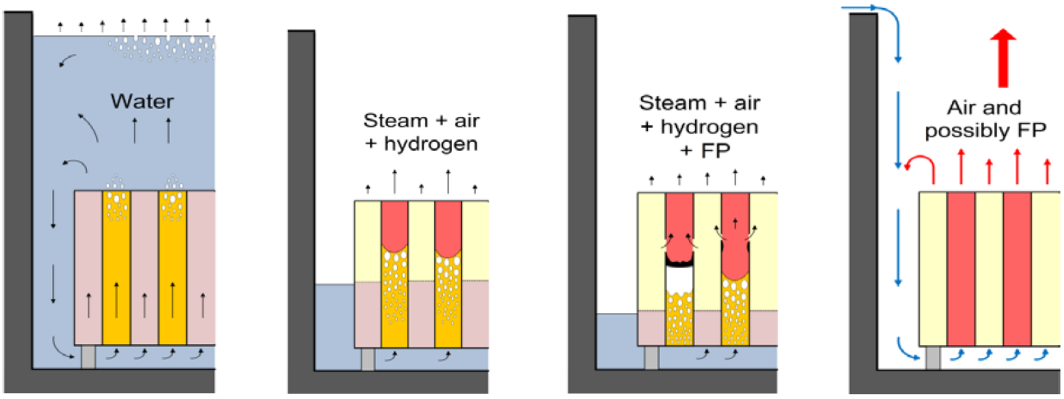
(C) Dispersed

Figure D.1 – Two Phase Flow (Source: Picture by Rudolf Hellmuth, distributed under a CC BY-SA 4.0 license).

but intact, partly covered and damaged, and completely uncovered as shown in Figures D.2(A) to D.2(D). For modeling the event of fuel damaged in Figure D.2(C) we would need more complex fuel performance models which are not the purpose of this Appendix but nonetheless important for serious accidents. The attention will be focused in the hydraulics mainly. In the other situations, the existence of a moving free surface (uncover of assemblies) marks the need for a segregated flow modeling and the presence of boiling and evaporation shows the appearance of a dispersed phase in it. Both in the presence of gravity and, therefore, natural convection. The production of hydrogen due to the clad - vapor interaction could eventually be evaluated, being hydrogen concentration in air an important factor in safety.

The computational modeling of multi-phase flows is not unique and a wide variety of models exists. A major classification can be done between an **Eulerian-Eulerian** approach or an **Eulerian-Lagrangian** approach [132].

In the Eulerian-Eulerian approach the phases are considered as interpenetrating continua and the concept of volume fraction is introduced which adds up to one (1) at each



(A) (B) (C) (D)

FIGURE D.2 – Thermal-hydraulics conditions in different scenarios of an SFP LOCA/LOFA (Source: [15]).

point of the space. Conservation equations are derived for each phases closed by empirical information (drag laws for example). This approach can account for both segregated and dispersed flow. One particular case in this Eulerian multi-phase approach is the Volume of Fluid (VOF) model. In the VOF model it is never allowed to have both phases present in a cell at the same time. Then, the momentum equations are collapsed to only one, a new term due to surface tension at the interface between phases is added and fluid properties are redefined as functions of the volume fractions. The sharp gradients at the interface are accounted for by adequate discretization of the advection term for capturing the interface such as **compressive interface captruing (CICSAM)** or **piece-wise interface construction (PLIC)**, or by the rewritten the Eulerian two-phase form of the phase fraction equation as done by Rusche [133].

In the Eulerian-Lagrangian approach one of the phases is considered as a continuous and the second one as dispersed. In this case, each individual particle of the dispersed phase is track in what is called a **Lagrangian Particle Tracking** method. Some approaches will model parcels of particles because of computational cost. The continuous phase is modeled with conservation balance equations and additional terms are added to account for the interactions with the particles of the dispersed phase. In general, this approach is opted when the mesh refinement of the Eulerian-Eulerian approach mesh can no longer follow the individual bubbles or droplets present.

In OpenFOAM a wide range of multi-phase solvers following both approaches exists. However, choosing the correct one is not trivial. Perhaps, the most comprehensive solver is the *reactingMultiphaseEulerFoam* [134] following an Eulerian-Eulerian approach known as Euler multi-phase model. It is capable of solving any number of phases with a common pressure and separate properties. In addition, it can represent multiple species and in-phase reaction (useful for hydrogen production modeling or radiolysis) as well as momentum, heat and mass transfer. Moreover, since its implementation is based on the work of Rusche [133] previously mentioned, it can account for sharp interfaces and as a consequence model both segregated flow (for example free surfaces) and dispersed flow (for example bubbles). However, this type of approach is computationally expensive since it has to model all conservation balance equations for each phase. Then, the VOF model is preferred up to a certain droplet/bubble size under which the surface tension and the particle interaction between them is no longer correctly considered. If the dispersed fraction becomes too small it should be evaluated if an Eulerian-Lagrangian approach is more convenient. In [135] both Eulerian-Eulerian and Eulerian-Lagrangian approaches are used in the same domain, so a combination of methods is eventually possible. Further explanation on the Euler multi-phase model can be found in Rusche [133], Capelli [136] and Orungati [137].

Other aspects to consider is the distribution of the droplet/bubble (if mono-dispersed or not) and the modeling of drag, lift, virtual mass and mass transfer phenomena. Many modeling features for this exists in OpenFOAM, however, it is not guarantee to fit the Spent Fuel Pool accident modeling. Nonetheless, OpenFOAM is specially useful for tailoring existing solvers to the user (with some programming knowledge) needs.

D.2.2 Navier-Stokes - Porous Medium Interface Conditions

Normally, the Navier-Stokes and the porous medium models for one phase are modeled as one equation with a Darcy-Forchheimer [138] term that is turn on or off depending at which place of the domain is being solved. This way, under a single mesh a porous zone is defined and solve with the strategy previously mentioned. However, as more phenomena are involved in the model (such as multiple phases, natural convection, species transport, chemical reactions, boiling/evaporation, CHT, among others) these modification on the Navier-Stokes equations are far from trivial. In addition, the mesh refinement ant the space and time resolution schemes can be very different at each domain (porous or free fluid). For these reason, it would be better to have the models as independent as possible. Fortunately, OpenFOAM allows the definition of the multiple subdomains or **regions**, i.e., the system to be modeled can be split in independent mesh domains with independent models. Then, they will be linked at their interfaces. To this purpose, interfacial conditions at the boundaries of the regions have to be defined, these are known as **baffles** in **openfoam** and some examples of how to implement them are available at the OpenFOAM package.

Now, the problem resides in defining which mathematical expressions links both models. Depending on the multi-phase model used in the pool system, the coupling conditions differ mathematically. Nonetheless, in general mechanical, compositional and thermal conditions needs to be imposed. An example of such conditions, based on a REV averaging local thermodynamic equilibrium and continuity of fluxes conditions is shown in the work of Grüninger [139] for a two-phase two-component porous medium coupled to a free flow Navier-Stokes model only in the non-wetting (gas) phase. This is not the case presented in this work but it highlights the main key points to be considered in the coupling conditions.

First, the mechanical coupling condition are the mass conservation across the interface or continuity of the total mass fluxes

$$(\rho_n V_n)^{NS} \cdot \vec{n} = (\rho_n V_n + \rho_W V_W)^{PM} \cdot \vec{n} \tag{D.1}$$

and the continuity of normal stress

$$\vec{n} \cdot \left(\rho_n V_n V_n^T - \mu_n \nabla V_n\right)^{NS} \cdot \vec{n} + p_n^{NS} = p_n^{PM} \tag{D.2}$$

As said before, this formulation considers that the free flow is only in the gas phase and water vaporises when it reaches it from the porous medium. A more convenient formulation for our case would be continuity of mass fluxes and normal stresses for each phase, i.e.

$$(\rho_\alpha V_\alpha)^{NS} \cdot \vec{\alpha} = (\rho_\alpha V_\alpha)^{PM} \cdot \vec{n} \tag{D.3}$$

$$\vec{n} \cdot \left(\overline{\overline{\tau}}_\alpha\right)^{NS} \cdot \vec{n} + p_\alpha^{NS} = p_\alpha^{PM} \tag{D.4}$$

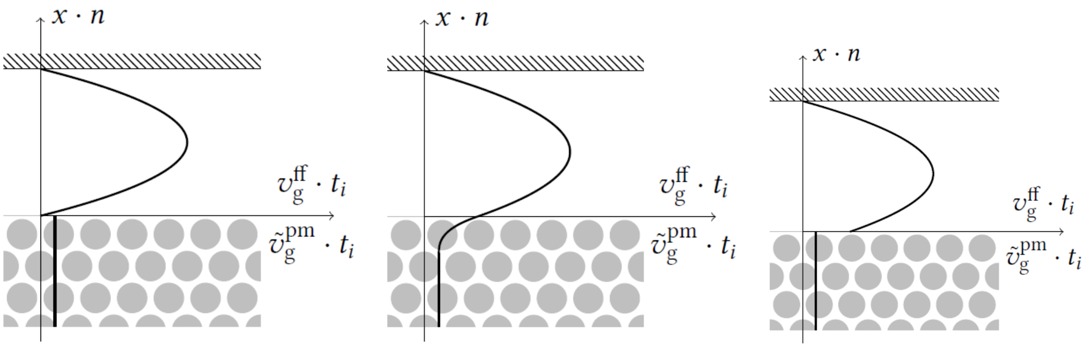
where τ_α is the (viscous and Reynolds) stress tensor for the α phase. It should be noted that the difference in the models will led to a pressure jump at the interface.

The tangential momenta should be evaluated as well. It was usually used no-slip conditions (Figure D.3(A)) at the interface, which simplifies the coupling. However, Beavers and Joseph [140] concluded that using the following expression is more accurate due to the creeping of the free flow into the porous medium:

$$\left(\gamma_{BJ}\left(V_{\alpha}^{NS} - \hat{V}_{\alpha}^{PM}\right) - \sqrt{k}\nabla V_{\alpha}^{NS}\vec{n}\right) \cdot \vec{t_i} = 0 \tag{D.5}$$

where $\hat{V}^{PM}$ is the seepage velocity, γ_{BJ} is an experimental slip parameter. In addition, Saffman [141] simplifies this condition arguing that the seepage velocity in comparison to the free flow velocity is small and can be neglected reducing it to:

$$\left(\gamma_{BJ} V_{\alpha}^{NS} - \sqrt{k}\nabla V_{\alpha}^{NS}\vec{n}\right) \cdot \vec{t_i} = 0 \tag{D.6}$$



(A) (B) (C)

FIGURE D.3 – Tangential momentum condition: (A) No-Slip, (B) Beavers-Joseph and (C) Beavers-Joseph-Saffman. Source: [139]

In Figure D.3 this approaches are compared. Other approaches to do this exists and are briefly discussed in [139]. It is remarked that the analysis in the cited work is done for soil and its extrapolation to fuel assemblies should be evaluated.

The compositional coupling consists in the continuity of mass or molar fractions and the continuity of component fluxes. For the mass local equilibrium assumption in Grüninger [139] they can be written as:

$$x_{\alpha}^{k,NS} = x_{\alpha}^{k,PM} \tag{D.7}$$

$$\left(\rho_{\alpha} x_{\alpha}^{k} V_{\alpha} + j_{\alpha}^{k}\right)^{NS} \cdot \vec{n} = \left(\rho_{\alpha} x_{\alpha}^{k} V_{\alpha} + j_{\alpha}^{k}\right)^{PM} \cdot \vec{n} \tag{D.8}$$

recalling that j_{α}^{k} is the diffusive flux of the α phase for the k component. In our case, the continuity of mass fraction and component fluxes is also needed but taking into account the non-equilibrium terms.

Similarly, the thermal coupling in [139] (with local thermal equilibrium assumption) consists in the continuity of temperature and heat fluxes:

$$T^{NS} = T_{PM} \tag{D.9}$$

$$\left(\rho_n h_n V_n + h_n^{H_2O} j_n^{H_2O} + h_n^{Air} j_n^{Air} - \lambda_n \nabla T\right)^{NS} \cdot \vec{n} = \left(\rho_n h_n V_n + \rho_w h_w V_w - \lambda_{PM} \nabla T\right)^{PM} \cdot \vec{n} \tag{D.10}$$

which for our case of non-equilibrium should be assured the continuity of temperatures (T_α) and heat fluxes for each phase.

This interfacial coupling is by no meas exhaustive, and it should be adapted to the chosen model for the multi-phase free flow (Section D.2.1) and the thermal and mass non-equilibrium in the porous medium model presented in Section 4.3.3.2. Other works treating this coupling are in [142][143][144]. In addition, DuMu^x [145][146] is a vast porous medium open-source code including free-flow coupling, although, not extensive, that can be used to verified models.

Nomenclature

Section 2.1

$\dot{F}$	Fission Rate
ρ_0	Initial Super-Prompt Reactivity
t_m	Burst Fission Rate Peak Time
T_r	Reactor Period

Section 2.2

ρ	Density
Σ	Macroscopic Cross Section
h_{sink}	Volumetric Heat Removal Coefficient
$q^{'''}$	Volumetric Heat Sink
T	Temperature
T_p	Perturbation Period
T_{ref}	Reference Temperature
U_x	Density

Section 3.1

$\bar{c}_d$	W-weighted Precursor Family d Concentration
β	Total Delayed Neutron Fraction
β_d	Delayed Neutron Fraction for Precursor Family d
β_d^{eff}	W-weighted Delayed Neutron Fraction for Precursor Family d
χ_d	Delayed Neutron Emission Spectrum Fraction for Precursor Family d
χ_d^g	Delayed Neutron Emission Spectrum Fraction for Precursor Family d for the Energy Group g
χ_p	Prompt Neutron Emission Spectrum Fraction
χ_p^g	Prompt Neutron Emission Spectrum Fraction for the Energy Group g
$\delta_{gg'}$	Kronecker Delta

Λ	W-weighted Generation Time
λ_d	Decay Constant for Precursor Family d
λ_s	Monier's W-weighted Constant Correction
$\mathcal{F}$	Fission Operator
$\mathcal{L}$	Transport Operator
$\mathcal{S}$	Scattering Operator
$\mathcal{T}$	Removal Operator
μ	Cosine of the Direction Angle θ in the Lab System
ν	Average Number of Neutrons Emitted per Fission
ν	Fission Macroscopic Cross Section
ϕ	Neutron Angular Flux Shape Function
ϕ^+	Adjoint Neutron Angular Flux Shape Function
Φ_n	Multi-group Neutron Angular Flux Matrix of order n in the Legendre Polynomial Expansion
ϕ_n	Neutron Flux Coefficient of order n in the Legendre Polynomial Expansion of the Angular Flux
ψ	Neutron Angular Flux
Ψ^-	Inward Angular Flux
ψ^g	Neutron Angular Flux for the Energy Group g
ρ	W-weighted Reactivity
Σ	Macroscopic Cross Section
Σ_a	Absorption Macroscopic Cross Section
Σ_f	Fission Macroscopic Cross Section
Σ_f^g	Total Macroscopic Cross Section for the Energy Group g
Σ_m	Scattering Macroscopic Cross Section Coefficient of order m in the Legendre Polynomial Expansion of the Scattering Macroscopic Cross Section
Σ_n	Multi-group Removal Cross Section Matrix of order n in the Legendre Polynomial Expansion
Σ_s	Scattering Macroscopic Cross Section
$\Sigma_s^{gg'}$	Scattering Macroscopic Cross Section from Energy Group g' to g
Σ_{tot}^g	Total Macroscopic Cross Section for the Energy Group g
Σ_t	Total Macroscopic Cross Section

$\vec{\Omega}$	Direction Cosines
$\vec{n}$	Normal Vector
$\vec{r}$	Spatial Coordinates
$\vec{u}$	Fluid Velocity
$C_{d,max}$	Maximum Precursors Concentration Value for Precursor Family d
C_d	Precursor's Family d
D	Diffusion Coefficient of order 1 in the Legendre Polynomial Expansion
D_d	Molecular Diffusion Coefficient for Precursor Family d
E	Energy
F	Multi-group Neutron Fission Source Matrix
f	Physical Parameter
G_d	Number of Precursor Families
H	W-weighted Fission Source
$I_{1/v}$	Multi-group Neutron Inverse Speed Matrix
k_{eff}	Effective Multiplication Factor
k_n	Multiplication Factor for cycle $n+1$
l_i	Length of the Track of the i-th Simulated Neutron
n	Neutron Angular Flux Amplitude Function
N_g	Number of Energy Groups
P_n	Legendre Polynomial of order n
R	Flux Integral for a Physical Parameter f
S_d	Multi-group Delayed Neutron Source Matrix
t	Time
t_0	Initial Time
v	Neutron Speed
W	Weight Function
W_0	Time-independent Weight Function
w_i	Weight of the i-th Simulated Neutron
X_d	Multi-group Delayed Neutron Emission Spectrum Fraction for Precursor Family d Matrix

Section 3.2

α	Thermal Expansion Coefficient

ϵ_i	Normal Strain in i-direction
γ_{ij}	Shearing Strain Normal to the i-direction and Parallel to the j-direction
$\mu,\ \lambda$	Lamé's Coefficients
ν	Poisson's Ratio
$\bar{\bar{\epsilon}}$	Strain Tensor
$\bar{\bar{\sigma}}$	Stress Tensor
ρ	Density
σ_i	Normal Stress in i-direction
τ_{ij}	Tangencial Stress Normal to the i-direction and Parallel to the j-direction
$\vec{D}$	Displacement Vector
$\vec{f}$	Body Forces
$\vec{n}$	Normal Vector
$\vec{r}$	Spatial Coordinates
$\vec{t}$	Applied Traction
E	Young's Modulus
G	Module of Elasticity in Shear or Modulus of Rigidity
I	Identity Matrix
p	External Pressure
t	Time
T_0	Initial/Reference Temperature

Section 3.3

β	Coefficient of Volume Expansion
$\dot{\eta}^{k}_{wn\rightarrow\alpha}$	Mass Flux between Phases
$\dot{Q}$	Rate of Heat Addition
$\dot{q}^{R}$	Turbulent Heat Flux
$\dot{q}_S$	Rate of Heat Source or Sink per Unit Surface
$\dot{q}_V$	Rate of Heat Source or Sink per Unit Volume
$\dot{W}$	Rate of Work Extraction
$\dot{W}_b$	Rate of Work done by Body Forces
$\dot{W}_s$	Rate of Work done by Surface Forces
$\frac{D}{Dt}$	Material or Lagrangian Derivative
$\hat{h}$	Specific Enthalpy

$\hat{V}$	Specific Volume
λ	Bulk Viscosity Coefficient
λ_α	Heat Conductivity of the α Phase
μ	Molecular Viscosity Coefficient
μ_α	Dynamic Viscosity of the α Phase
μ_t	Turbulent Eddy Viscosity
ν	Kinematic Viscosity or Momentum Diffusivity
$\overline{\overline{\sigma}}$	Stress Tensor
$\overline{\overline{\tau}}$	Deviatoric or Viscous Stress Tensor
$\overline{\overline{\tau}}^R$	Reynolds Stress Tensor
$\overline{\overline{I}}$	Identity Matrix
ϕ	Porosity
ρ	Density
ρ_0	Initial/Reference Density
ρ_α	Density of the α Phase
ρ_α^k	Density of the Species k in the α Phase
τ	Tortuosity
$\vec{g}$	Gravity Vector
$\vec{n}$	Normal Vector
$\vec{r}$	Spatial Coordinates
$\vec{u}$	Fluid Velocity
$\vec{u}_r$	Relative Velocity
$\vec{u}_s$	Velocity of the Deforming Control Surface
$\vec{w}$	Angular Velocity
a_{ij}	Average Interfacial Surface Between i and j Phases
B	Property of the Fluid
b	Intensive Value of B
c	Specific Heat Capacity of the Solid Phase
c_p	Specific Heat Capacity at Constant Pressure
D^k	Diffusion of the k Species
$D^k_{\alpha,pm}$	Macroscale Diffusion Coefficient of the Species k in the α Phase
E	Energy

e	Specific Energy
f	Body or Surface Force
f_b	Body Force
f_s	Surface Force
H^k	Henry's Law Coefficient of the k Species
h_α	Specific Enthalpy of the α Phase
h_α^k	Specific Enthalpy of the Species k in the α Phase
j_α^k	Macroscale Diffusion of the Species k in the α Phase
K	Permeability
k	Thermal Conductivity
$k_{r,\alpha}$	Relative Permeability of the α Phase
k_t	Turbulent Thermal Diffusivity
L	Characteristic Length
m	Mass
n	Non-Wetting Phase
p	Pressure
p_α	Pressure of the α Phase
p_c	Capillary Pressure
q_α^{energy}	Power Source of the α Phase
q_α^k	Mass Source of the Species k in the α Phase
S_α	Saturation of Phase α
T	Temperature
T_0	Initial/Reference Temperature
T_α	Temperature of the α Phase
U	Characteristic Velocity
u_α	Specific Internal Energy of the α Phase
V	Volume
$V(t)$	Deformable Control Volume
V_α	Velocity of the α Phase
w	Wetting Phase
$x_{\alpha,equil}^k$	Equilibrium Mass Fraction of the Species k in the α Phase
x_α^k	Mass Fraction of the Species k in the α Phase

Section 4.3.1

$\bar{c}_d$	W-weighted Precursor Family d Concentration
β	Total Delayed Neutron Fraction
β_d	Delayed Neutron Fraction for Precursor Family d
β_d^{eff}	W-weighted Delayed Neutron Fraction for Precursor Family d
χ_d^g	Delayed Neutron Emission Spectrum Fraction for Precursor Family d for the Energy Group g
χ_p^g	Prompt Neutron Emission Spectrum Fraction for the Energy Group g
Δt	Flux Shape Time Step
δt	Amplitude/Point Kinetics Time Step
Δ	Mesh Cell Inverse Center-Face Distance Parallel to $\vec{n}$ at the Boundary
ϵ	Fission Energy
Λ	W-weighted Generation Time
λ_l	Decay Constant for Precursor Family l
$\mathcal{F}$	Fission Operator
ν	Average Number of Neutrons Emitted per Fission
ω_i	Weight of the i-th Neutron Simulated
ϕ	Neutron Angular Flux Shape Function
ϕ_n^g	Neutron Flux Coefficient of order n in the Legendre Polynomial Expansion of the Angular Flux for Energy Group g
ψ	Neutron Angular Flux
ρ	W-weighted Reactivity
ρ	Density
ρ_0	Initial/Reference Density
Σ	Macroscopic Cross Section
Σ_n^g	Remotion Macroscopic Cross Section Coefficient of order n in the Legendre Polynomial Expansion of the Scattering Macroscopic Cross Section for Energy Group g
Σ_{sn}^g	Scattering Macroscopic Cross Section Coefficient of order n in the Legendre Polynomial Expansion of the Scattering Macroscopic Cross Section for Energy Group g
Σ_f	Fission Macroscopic Cross Section

Σ_f^g	Total Macroscopic Cross Section for the Energy Group g
Σ_{ref}	Initial/Reference Macroscopic Cross Section
Σ_r	r-Reaction Macroscopic Cross Section
$\Sigma_s^{gg'}$	Scattering Macroscopic Cross Section from Energy Group g' to g
Σ_t	Total Macroscopic Cross Section
Σ_t^g	Total Macroscopic Cross Section for the Energy Group g
$\vec{\Omega}$	Direction Cosines
$\vec{J}$	Neutron Current
$\vec{n}$	Normal Vector
$\vec{r}, \vec{r}_0$	Spatial Coordinates
$\vec{r}_i$	Spatial Coordinates for the i-th Neutron Simulated
$\vec{r}_j$	Spatial Coordinates for the j-th Mesh Cell
$\vec{u}$	Fluid Velocity
C_d, C_{d_0}, C_l	Precursors Distribution for Family d, d_0, l respectively
D_d	Molecular Diffusion Coefficient for Precursor Family d
E	Energy
E_i	Energy of the i-th Neutron Simulated
f	Physical Parameter
F^g	Prompt Fission Source for the Energy Group g
F^g	Prompt Fission Source
k_j	Multiplication Factor of the j-th Iteration
k_{neut}	External Iterations Counter for Neutronics
N	Neutron Density
n	Neutron Angular Flux Amplitude Function
P	Volumetric Power
R	Flux Integral for a Physical Parameter f
S_d^g	Delayed Neutron Source for Precursor Family df or the Energy Group g
S_n^g	n-th Order Scattering Source for the Energy Group g
T	Temperature
t_0	Initial Time
T_{ref}	Initial/Reference Temperature
U_n^g	Neutron Flux Coefficient of order n in the Legendre Polynomial Expansion of the

	Angular Flux with Change of Coordinates for Energy Group g
v	Neutron Speed
v^g	Neutron Speed for Energy Group g
V_j	Volume of j-th Mesh Cell
W_0	Time-independent Weight Function

Section 4.3.2

α	Thermal Expansion Coefficient
$\mu,\ \lambda$	Lamé's Coefficients
ν	Poisson's Ratio
ρ	Density
$\vec{D}$	Displacement Vector
$\vec{f}$	Body Forces
$\vec{r}$	Spatial Coordinates
E	Young's Modulus
I	Identity Matrix
T	Temperature
t	Time

Section 4.3.3

$\frac{D}{Dt}$	Material or Lagrangian Derivative
μ	Molecular Viscosity Coefficient
μ_α	Dynamic Viscosity of the α Phase
ϕ	Porosity
ρ	Density
ρ_α	Density of the α Phase
$\vec{g}$	Gravity Vector
$\vec{n}$	Normal Vector
$\vec{u}$	Fluid Velocity
f_b	Body Force
K	Permeability
$k_{r,\alpha}$	Relative Permeability of the α Phase
n	Non-Wetting Phase
p	Pressure

p_α	Pressure of the α Phase
p_c	Capillary Pressure
q_α	Mass Source of the α Phase
S_α	Saturation of Phase α
t	Time
$V_{\alpha,fixed}$	Imposed Velocity at the Boundary for the α Phase
V_α	Velocity of the α Phase
V_{fixed}	Imposed Velocity at the Boundary
w	Wetting Phase

Section 5.1

α	Thermal Expansion Coefficient
α_{norm}	Normalization Factor
β	W-weighted Delayed Neutron Fraction
β_d	Delayed Neutron Fraction for Precursor Family d
χ_d	Delayed Neutron Emission Spectrum Fraction for Precursor Family d
χ_p	Prompt Neutron Emission Spectrum Fraction
Λ	W-weighted Generation Time
$\mathcal{F}$	Fission Operator
$\mathcal{L}$	Transport Operator
$\mathcal{S}$	Scattering Operator
$\mathcal{T}$	Removal Operator
ν	Poisson's Ratio
ϕ	Neutron Angular Flux Shape Function
ρ	W-weighted Reactivity
ρ	Density
σ	Standard Deviation
$\vec{\Omega}$	Direction Cosines
$\vec{J}$	Neutron Current
$\vec{r}$	Spatial Coordinates
c	Specific Heat Capacity
E	Energy
E	Young's Modulus

G_d	Number of Precursor Families
H	W-weighted Fission Source
k	Multiplication Factor
k	Thermal Conductivity
k_0	Initial Multiplication Factor
n	Neutron Angular Flux Amplitude Function
T	Temperature
t	Time
t_0	Initial Time
T_r	Reactor Period
W_0	Time-independent Weight Function

Section 5.2

α	Thermal Expansion Coefficient
ν	Kinematic Viscosity
ϕ	Neutron Angular Flux Shape Function
ρ	Density
ρ	Reactivity
Σ_f	Fission Macroscopic Cross Section
Θ	Phase Shift
$\vec{n}$	Normal Vector
$\vec{v}$	Fluid Velocity Field
A	Perturbation Amplitude
C	Volumetric Heat Capacity
C_d	Precursors Distribution for Family d
h_{sink}	Volumetric Heat Removal Coefficient
P_{ref}	Initial/Reference Power
T	Temperature
T_p	Perturbation Period
U_x	Top Lid Tangential Velocity

Section 5.3

ϕ	Porosity
ρ	Density

ρ_α	Density of the α Phase
H	Rack Water Level
h	Pool Water Level
h_α	Specific Enthalpy of the α Phase
K	Permeability
k_{eff}	Multiplication Factor
n	Non-Wetting Phase
p_α	Pressure of the α Phase
S_α	Saturation of Phase α
T	Temperature
T_α	Temperature of the α Phase
u_α	Specific Internal Energy of the α Phase
w	Wetting Phase

Chapter 6

β	Total Delayed Neutron Fraction
λ_d	Decay Constant for Precursor Family d
ν	Average Number of Neutrons Emitted per Fission
ϕ	Neutron Angular Flux Shape Function
Σ_f	Fission Macroscopic Cross Section
C_d	Precursors Distribution for Family d
n	Neutron Angular Flux Amplitude Function
t	Time

Bibliography

[1] W R Stratton and D R Smith. *A review of criticality accidents.* Tech. rep. DOE/NCT-04 ON: DE90015896; TRN: 90-029466. Lawrence Livermore National Laboratory, 1989.

[2] H.G. Weller, G. Tabor, H. Jasak, and C. Fureby. "A Tensorial Approach to Computational Continuum Mechanics Using Object Orientated Techniques". In: *Computers in Physics* 12 (Nov. 1998), pp. 620–631. DOI: 10.1063/1.168744.

[3] H. Jasak, A. Jemcov, and Ž Tukovíc. "OpenFOAM: A C++ Library for Complex Physics Simulations". In: *International Workshop on Coupled Methods in Numerical Dynamics, IUC, Dubrovnik, Croatia.* 2007.

[4] W. M. Stacey. *Nuclear Reactor Physics.* Wiley, 2007. ISBN: 9783527406791. URL: https://books.google.fr/books?id=y1UgcgVSXSkC.

[5] J. Leppänen, M. Pusa, T. Viitanen, V. Valtavirta, and T. Kaltiaisenaho. "The Serpent Monte Carlo code: Status, development and applications in 2013". In: *Annals of Nuclear Energy* 82 (2015). 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, pp. 142–150. ISSN: 0306-4549. DOI: https://doi.org/10.1016/j.anucene.2014.08.024. URL: http://www.sciencedirect.com/science/article/pii/S0306454914004095.

[6] VTT Technical Research Centre. *Serpent - a Continuous-energy Monte Carlo Reactor Physics Burnup Calculation Code.* URL: http://montecarlo.vtt.fi/index.htm (visited on 09/02/2020).

[7] B. L. Sjenitzer and J. E. Hoogenboom. "Dynamic Monte Carlo Method for Nuclear Reactor Kinetics Calculations". In: *Nuclear Science and Engineering* 175.1 (2013),

pp. 94–107. DOI: 10.13182/NSE12-44. URL: https://doi.org/10.13182/NSE12-44.

[8] R. E. Peterson and G. A. Newby. *Lady Godiva: an Unreflected Uranium-235 Critical Assembly*. Tech. rep. LA-1614. Los Alamos Scientific Laboratory, 1953.

[9] F. Barbry, P. Fouillaud, P. Grivot, and L. Reverdy. “Review of the CRAC and SILENE criticality accident studies”. In: *Nuclear science and engineering* 161.2 (2009), pp. 160–187.

[10] T. F. Wimett, L. B. Engle, G. A. Graves, G. R. Keepin, and J. D. Orndoff. *Time behavior of Godiva through prompt critical*. Tech. rep. LA-2029. Los Alamos Scientific Laboratory, 1956.

[11] G. E. Hansen. *Burst characteristics associated with the slow assembly of fissionable materials*. Tech. rep. LA-1441. Los Alamos Scientific Laboratory, 1952.

[12] M. Aufiero and P. Rubiolo. “Testing and Verification of Multiphysics Tools for Fast-Spectrum MSRs: The CNRS Benchmark”. In: *Transactions of the American Nuclear Society* 118.1 (2018), pp. 837–840.

[13] J. Blanco, P. Rubiolo, and E. Dumonteil. *Neutronic modeling strategies for a liquid fuel transient calculation*. Poster presented at Physics of Reactors Conference (PHYSOR 2020), Cambridge, UK.

[14] M. Tiberga, R. Gonzalez Gonzaga de Oliveira, E. Cervi, J. Blanco, S. Lorenzi, M. Aufiero, D. Lathouwers, and P. Rubiolo. “Results from a multi-physics numerical benchmark for codes dedicated to molten salt fast reactors”. In: *Annals of Nuclear Energy* 142 (2020), p. 107428. ISSN: 0306-4549. DOI: https://doi.org/10.1016/j.anucene.2020.107428. URL: http://www.sciencedirect.com/science/article/pii/S0306454920301262.

[15] M. Adorni, H. Esmaili, W. Grant, T. Hollands, Z. Hozer, B. Jaeckel, M. Munoz, T. Nakajima, F. Rocchi, M. Strucic, N. Tregoures, P. Vokac, K. I. Ahn, L. Bourgue, R. Dickson, P. A. Douxchamps, L. E. Herranz, L. O. Jernkvist, A. Amri, and M. P. Kissane. *Status Report on Spent Fuel Pools under Loss-of-Cooling and Loss-of-Coolant Accident Conditions - Final Report*. Tech. rep. NEA-CSNI-R–2015-2. Nuclear Energy Agency of the OECD (NEA), 2015.

[16] R. O. Gauntt, D. A. Kalinich, J. N. Cardoni, J. Phillips, A. S. Goldmann, S. Y. Pickering, M. Francis, K. Robb, L. J. Ott, D. Wang, C. Smith, S. Germain, D. Schwieder, and C. Phelan. *Fukushima Daiichi accident study : status as of April 2012*. Tech. rep. SAND2012-6173. Sandia National Laboratories, 2012.

[17] D. Wang, I. C. Gauld, G. L. Yoder, L. J. Ott, G. F. Flanagan, M. W. Francis, E. L. Popov, J. J. Carbajo, P. K. Jain, J. C. Wagner, and J. C. Gehin. "Study of Fukushima Daiichi Nuclear Power Station Unit 4 Spent-Fuel Pool". In: *Nuclear Technology* 180.2 (2012), pp. 205–215. DOI: 10.13182/NT12-A14634. URL: https://doi.org/10.13182/NT12-A14634.

[18] M. Tano Retamales. "Developement of multi-physical multiscale models for molten salts at high temperature and their experimental validation". PhD Thesis. Université Grenoble Alpes, Nov. 2018. URL: https://tel.archives-ouvertes.fr/tel-02007262.

[19] P. Horgue, C. Soulaine, J. Franc, R. Guibert, and G. Debenest. "An open-source toolbox for multiphase flow in porous media". In: *Computer Physics Communications* 187 (2015), pp. 217–226. ISSN: 0010-4655. DOI: https://doi.org/10.1016/j.cpc.2014.10.005. URL: http://www.sciencedirect.com/science/article/pii/S0010465514003403.

[20] Y. G. Jo, B. Cho, and N. Z. Cho. "Nuclear Reactor Transient Analysis by Continuous-Energy Monte Carlo Calculation Based on Predictor-Corrector Quasi-Static Method". In: *Nuclear Science and Engineering* 183.2 (2016), pp. 229–246. DOI: 10.13182/NSE15-100. URL: https://doi.org/10.13182/NSE15-100.

[21] J. J. Duderstadt and L. J. Hamilton. *Nuclear reactor analysis*. Wiley, 1976.

[22] M. Faucher, D. Mancusi, and A. Zoia. "New kinetic simulation capabilities for Tripoli-4®: Methods and applications". In: *Annals of Nuclear Energy* 120 (2018), pp. 74–88. ISSN: 0306-4549. DOI: https://doi.org/10.1016/j.anucene.2018.05.030. URL: https://www.sciencedirect.com/science/article/pii/S0306454918302676.

[23] D. Ferraro, M. Faucher, D. Mancusi, A. Zoia, V. Valtavirta, J. Leppänen, and V. H. Sanchez-Espinoza. "Serpent and Tripoli4® transient calculations comparisons for diverse reactivity insertion scenarios in a 3D PWR minicore benchmark". In: *Proceedings of M&C19* (2019).

[24] I. Lux and Koblinger L. *Monte Carlo Particle Transport Methods*. Boca Raton, FL: CRC Press, 1991.

[25] H. Kahn. *Applications of Monte Carlo*. Santa Monica, CA: RAND Corporation, 1956.

[26] G. I. Bell and S. Glasstone. *Nuclear Reactor Theory*. Van Nostrand Reinhold Company, 1970.

[27] J. Spanier and E. M. Gelbard. *Monte Carlo principles and neutron transport problems*. Dover ed. Dover Publications, 2008. ISBN: 978-0-486-46293-6. URL: http://catdir.loc.gov/catdir/toc/fy0804/2007037512.html.

[28] S. P. Hamilton and T. M. Evans. "Efficient solution of the simplified PN equations". In: *Journal of Computational Physics* 284 (2015), pp. 155–170. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2014.12.014. URL: http://www.sciencedirect.com/science/article/pii/S0021999114008225.

[29] J. Leppänen. "Development of a New Monte Carlo Reactor Physics Code". PhD thesis. VTT Technical Research Centre of Finland, 2007.

[30] C. J. Werner et al. *MCNP6.2 Release Notes*. Tech. rep. LA-UR-18-20808. Los Alamos National Laboratory, 2018.

[31] J. Leppänen. "Performance of Woodcock delta-tracking in lattice physics applications using the Serpent Monte Carlo reactor physics burnup calculation code". In: *Annals of Nuclear Energy* 37.5 (2010), pp. 715–722. ISSN: 0306-4549. DOI: https://doi.org/10.1016/j.anucene.2010.01.011. URL: http://www.sciencedirect.com/science/article/pii/S0306454910000320.

[32] A. Majumdar. "Development of a multiple perturbation Monte Carlo method for eigenvalue problems and implementation on parallel processors". PhD thesis. University of Michigan, 1996.

[33] X-5 Monte Carlo Team. *MCNP — A General Monte Carlo N-Particle Transport Code, Version 5. Volume I: Overview and Theory*. Tech. rep. LA-UR-03-1987. Los Alamos National Laboratory, 2005.

[34] J. Leppänen. "On the use of delta-tracking and the collision flux estimator in the Serpent 2 Monte Carlo particle transport code". In: *Annals of Nuclear Energy* 105 (2017), pp. 161–167. ISSN: 0306-4549. DOI: https://doi.org/10.1016/j.anucene.2017.03.006. URL: http://www.sciencedirect.com/science/article/pii/S0306454916311367.

[35] J. Leppänen. "Development of a dynamic simulation mode in serpent 2 Monte Carlo code". In: *Proceedings*. International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering, M&C 2013. 2013, pp. 117–127. ISBN: 978-0-89448-700-2.

[36] A. Zoia, E. Brun, and F. Malvagi. "Alpha eigenvalue calculations with Tripoli-4®". In: *Annals of Nuclear Energy* 63 (2014), pp. 276–284. ISSN: 0306-4549. DOI: https://doi.org/10.1016/j.anucene.2013.07.018. URL: https://www.sciencedirect.com/science/article/pii/S0306454913003733.

[37] M. Aufiero, C. Fiorina, A. Laureau, P. Rubiolo, and V. Valtavirta. "Serpent–OpenFOAM coupling in transient mode: simulation of a Godiva prompt critical burst". In: *Proceedings of M&C+ SNA+ MC* (2015), pp. 19–23.

[38] M. Aufiero. "Development of advanced simulation tools for circulating fuel nuclear reactors". PhD Thesis. Politecnico di Milano, Mar. 2014. URL: http://hdl.handle.net/10589/89364.

[39] A. F. Henry. "The Application of Reactor Kinetics to the Analysis of Experiments". In: *Nuclear Science and Engineering* 3.1 (1958), pp. 52–70. DOI: 10.13182/NSE58-1. URL: https://doi.org/10.13182/NSE58-1.

[40] A. F. Henry and N. J. Curlee. "Verification of a Method for Treating Neutron Space-Time Problems". In: *Nuclear Science and Engineering* 4.6 (1958), pp. 727–744. DOI: 10.13182/NSE4-727. URL: https://doi.org/10.13182/NSE4-727.

[41] K. Ott. "Quasistatic Treatment of Spatial Phenomena in Reactor Dynamics". In: *Nuclear Science and Engineering* 26.4 (1966), pp. 563–565. DOI: 10.13182/NSE66-A18428. URL: https://doi.org/10.13182/NSE66-A18428.

[42] K. O. Ott and D. A. Meneley. "Accuracy of the Quasistatic Treatment of Spatial Reactor Kinetics". In: *Nuclear Science and Engineering* 36.3 (1969), pp. 402–411. DOI: 10.13182/NSE36-402. URL: https://doi.org/10.13182/NSE36-402.

[43] S. Dulla, E. H. Mund, and P. Ravetto. "The quasi-static method revisited". In: *Progress in Nuclear Energy* 50.8 (2008), pp. 908–920. ISSN: 0149-1970. DOI: https://doi.org/10.1016/j.pnucene.2008.04.009. URL: http://www.sciencedirect.com/science/article/pii/S014919700800084X.

[44] S. Dulla, P. Ravetto, and M. M. Rostagno. "Neutron kinetics of fluid–fuel systems by the quasi-static method". In: *Annals of Nuclear Energy* 31.15 (2004), pp. 1709–1733. ISSN: 0306-4549. DOI: https://doi.org/10.1016/j.anucene.2004.05.004. URL: http://www.sciencedirect.com/science/article/pii/S0306454904000982.

[45] A. Radkowsky and J. R. Stehn. *Naval reactors physics handbook*. U.S. Atomic Energy Commission. Division of Reactor Development, 1958.

[46] J. Devooght and E. Mund. "Generalized Quasi-Static Method for Nuclear Reactor Space-Time Kinetics". In: *Nuclear Science and Engineering* 76.1 (1980), pp. 10–17. DOI: 10.13182/NSE80-A19288. URL: https://doi.org/10.13182/NSE80-A19288.

[47] C. Patricot. "Couplages multi-physiques : évaluation des impacts méthodologiques lors de simulations de couplages neutronique/thermique/mécanique". PhD thesis. Université Paris-Saclay, 2016.

[48] A. Monier. "Application of the collocation technique to the spatial discretization of the generalized quasistatic method for nuclear reactors". PhD thesis. Université de Montréal, 1993.

[49] M. Becker. "A Generalized Formulation of Point Nuclear Reactor Kinetics Equations". In: *Nuclear Science and Engineering* 31.3 (1968), pp. 458–464. DOI: 10.13182/NSE68-A17588. URL: https://doi.org/10.13182/NSE68-A17588.

[50] "Mathematical Methods in Nuclear Reactor Dynamics". In: *Mathematical Methods in Nuclear Reactor Dynamics*. Ed. by A. Akcasu, G. S. Lellouche, and L. M. Shotkin. Academic Press, 1971, p. iv. ISBN: 978-0-12-047150-8. DOI: https://doi.org/10.1016/B978-0-12-047150-8.50003-8. URL: https://www.sciencedirect.com/science/article/pii/B9780120471508500038.

[51] E. Gelbard. *Application of the spherical harmonics to reactor problems*. Tech. rep. WAPD-BT-20. Westinghouse Atomic Power Department, 1960.

[52] R. G. McClarren. "Theoretical Aspects of the Simplified $\mathrm{P_N}$ Equations". In: *Transport Theory and Statistical Physics* 39.2-4 (2010), pp. 73–109. DOI: 10.1080/00411450.2010.535088. URL: https://doi.org/10.1080/00411450.2010.535088.

[53] D. Lee, T. Kozlowski, and T. J. Downar. "Multi-group SP3 approximation for simulation of a three-dimensional PWR rod ejection accident". In: *Annals of Nuclear Energy* 77 (2015), pp. 94–100. ISSN: 0306-4549. DOI: https://doi.org/10.1016/j.anucene.2014.10.019. URL: http://www.sciencedirect.com/science/article/pii/S0306454914005581.

[54] P. S. Brantley and E. W. Larsen. "The Simplified $\mathrm{P_3}$ Approximation". In: *Nuclear Science and Engineering* 134.1 (2000), pp. 1–21. DOI: 10.13182/NSE134-01. URL: https://doi.org/10.13182/NSE134-01.

[55] C. Fiorina, M. Hursin, and A. Pautz. "Extension of the GeN-Foam neutronic solver to SP3 analysis and application to the CROCUS experimental reactor". In: *Annals of Nuclear Energy* 101 (2017), pp. 419–428. ISSN: 0306-4549. DOI: https://doi.org/10.1016/j.anucene.2016.11.042. URL: http://www.sciencedirect.com/science/article/pii/S0306454916305126.

[56] J. Lubliner and P. Papadopoulos. *Introduction to Solid Mechanics*. Springer International Publishing Switzerland, 2017.

[57] S. Timoshenko and J. N. Goodier. *Theory of Elasticy*. McGraw-Hill Book Company, 1951.

[58] T. M. Atanackovic and A. Guran. *Theory of Elasticy for Scientists and Engineers*. Birkhäuser, 2000.

[59] S. Timoshenko. *Strength of Materials: Elementary theory and problems*. Van Nostrand, 1930.

[60] H. Jasak and H. G. Weller. "Application of the finite volume method and unstructured meshes to linear elasticity". In: *International Journal for Numerical Methods in Engineering* 48.2 (2000), pp. 267–287. DOI: 10.1002/(SICI)1097-0207(20000520)48:2<267::AID-NME884>3.0.CO;2-Q. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291097-0207%2820000520%2948%3A2%3C267%3A%3AAID-NME884%3E3.0.CO%3B2-Q.

[61] I. Demirdžić and D. D. Martinović. “Finite volume method for thermo-elasto-plastic stress analysis”. In: *Computer Methods in Applied Mechanics and Engineering* 109.3 (1993), pp. 331–349. ISSN: 0045-7825. DOI: https://doi.org/10.1016/0045-7825(93)90085-C. URL: http://www.sciencedirect.com/science/article/pii/004578259390085C.

[62] I. Demirdžić and D. D. Martinović. “Numerical method for coupled fluid flow, heat transfer and stress analysis using unstructured moving meshes with cells of arbitrary topology”. In: *Computer Methods in Applied Mechanics and Engineering* 125.1 (1995), pp. 235–255. ISSN: 0045-7825. DOI: https://doi.org/10.1016/0045-7825(95)00800-G. URL: http://www.sciencedirect.com/science/article/pii/004578259500800G.

[63] F. M. White. *Fluid Mechanics.* Mcgraw-Hill Kogakusha, 1979.

[64] F. Moukalled, L. Mangani, and M. Darwish. *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM®and Matlab.* Springer International Publishing, 2016.

[65] J. H. Ferziger and M. Peric. *Computational Methods for Fluid Dynamics.* Springer-Verlag Berlin Heidelberg, 2002.

[66] S. V. Patankar. *Numerical heat transfer and fluid flow.* Series on Computational Methods in Mechanics and Thermal Science. Hemisphere Publishing Corporation (CRC Press, Taylor & Francis Group), 1980. ISBN: 978-0891165224. URL: http://www.crcpress.com/product/isbn/9780891165224.

[67] T. L. Bergman, F. P. Incropera, D. P. DeWitt, and A. S. Lavine. *Fundamentals of Heat and Mass Transfer.* Wiley, 2011. ISBN: 9780470501979. URL: https://books.google.fr/books?id=vvyIoXEywMoC.

[68] A. N. Kolmogorov. “Local structure of turbulence in an incompressible viscous fluid at very large Reynolds numbers”. In: *Doklady Akademiia Nauk SSSR* 30.4 (1941), pp. 299–301. URL: https://cds.cern.ch/record/739746.

[69] A. N. Kolmogorov. “Dissipation of Energy in Locally Isotropic Turbulence”. In: *Akademiia Nauk SSSR Doklady* 32 (Apr. 1941), p. 16.

[70] O. Reynolds. “On the Dynamical Theory of Incompressible Viscous Fluids and the Determination of the Criterion”. In: *Philosophical Transactions of the Royal Society of London Series A* 186 (Jan. 1895), pp. 123–164. DOI: 10.1098/rsta.1895.0004.

[71] P. Rubiolo, M. Tano, J. Blanco, V. Ghetta, J. Giraud, and V. Richard. “A Numerical Tool for Space Nuclear Reactor Design based on Molten Salt Reactors (MSRs)”. In: *Transactions of the American Nuclear Society (Annual Meeting 2020)* 122 (2020), pp. 679–681. URL: https://ans.org/pubs/transactions/a_48332.

[72] J. Niessner and S. M. Hassanizadeh. “Non-equilibrium interphase heat and mass transfer during two-phase flow in porous media—Theoretical considerations and modeling”. In: *Advances in Water Resources* 32.12 (2009), pp. 1756–1766. ISSN: 0309-1708. DOI: https://doi.org/10.1016/j.advwatres.2009.09.007. URL: http://www.sciencedirect.com/science/article/pii/S0309170809001511.

[73] W. G. Gray and S.M. Hassanizadeh. “Averaging theorems and averaged equations for transport of interface properties in multiphase systems”. In: *International Journal of Multiphase Flow* 15.1 (1989), pp. 81–95. ISSN: 0301-9322. DOI: https://doi.org/10.1016/0301-9322(89)90087-6. URL: http://www.sciencedirect.com/science/article/pii/0301932289900876.

[74] M. Hassanizadeh and W. G. Gray. “General conservation equations for multi-phase systems: 2. Mass, momenta, energy, and entropy equations”. In: *Advances in Water Resources* 2 (1979), pp. 191–203. ISSN: 0309-1708. DOI: https://doi.org/10.1016/0309-1708(79)90035-6. URL: http://www.sciencedirect.com/science/article/pii/0309170879900356.

[75] M. Hassanizadeh and W. G. Gray. “General conservation equations for multi-phase systems: 3. Constitutive theory for porous media flow”. In: *Advances in Water Resources* 3.1 (1980), pp. 25–40. ISSN: 0309-1708. DOI: https://doi.org/10.1016/0309-1708(80)90016-0. URL: http://www.sciencedirect.com/science/article/pii/0309170880900160.

[76] P. Nuske, V. Joekar-Niasar, and R. Helmig. “Non-equilibrium in multiphase multicomponent flow in porous media: An evaporation example”. In: *International Journal of Heat and Mass Transfer* 74 (2014), pp. 128–142. ISSN: 0017-9310. DOI:

https://doi.org/10.1016/j.ijheatmasstransfer.2014.03.011. URL: http://www.sciencedirect.com/science/article/pii/S0017931014002178.

[77] J. Niessner and S. M. Hassanizadeh. "Modeling Kinetic Interphase Mass Transfer for Two-Phase Flow in Porous Media Including Fluid–Fluid Interfacial Area". In: *Transport in Porous Media* 80 (2009). DOI: 10.1007/s11242-009-9358-5. URL: https://doi.org/10.1007/s11242-009-9358-5.

[78] B. Ahrenholz, J. Niessner, R. Helmig, and M. Krafczyk. "Pore-scale determination of parameters for macroscale modeling of evaporation processes in porous media". In: *Water Resources Research* 47.7 (2011). DOI: 10.1029/2010WR009519. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2010WR009519.

[79] R. J. Millington and J. P. Quirk. "Permeability of porous solids". In: *Transactions of the Faraday Society* 57 (1961), pp. 1200–1207.

[80] P. Atkins and J. De Paula. *Elements of physical chemistry*. Oxford University Press, USA, 2013.

[81] S. Whitaker. "Flow in porous media I: A theoretical derivation of Darcy's law". In: *Transport in porous media* 1.1 (1986), pp. 3–25.

[82] M. Kaviany. *Principles of heat transfer in porous media*. Springer Science & Business Media, 2012.

[83] R. H. Brooks and A. T. Corey. "Hydraulic properties of porous media". In: *Hydrology papers (Colorado State University); no. 3* (1964).

[84] M. T. van Genuchten. "A Closed-form Equation for Predicting the Hydraulic Conductivity of Unsaturated Soils". In: *Soil Science Society of America Journal* 44.5 (1980), pp. 892–898. DOI: 10.2136/sssaj1980.03615995004400050002x. URL: https://acsess.onlinelibrary.wiley.com/doi/abs/10.2136/sssaj1980.03615995004400050002x.

[85] W. Chao-Yang and C. Beckermann. "A two-phase mixture model of liquid-gas flow and heat transfer in capillary porous media—I. Formulation". In: *International Journal of Heat and Mass Transfer* 36.11 (1993), pp. 2747–2758. ISSN: 0017-9310. DOI: https://doi.org/10.1016/0017-9310(93)90094-M. URL: http://www.sciencedirect.com/science/article/pii/001793109390094M.

[86] R. C. Reid, J. M. Prausnitz, and B. E. Poling. *The properties of gases and liquids.* McGraw Hill Book Co., New York, NY, 1987.

[87] The OpenFOAM Foundation Ltd. *OpenFOAM - The OpenFOAM Foundation.* 2011. URL: https://openfoam.org/ (visited on 09/02/2020).

[88] VTT Technical Research Centre. *Coupled multi-physics calculations.* URL: http://serpent.vtt.fi/mediawiki/index.php/Coupled_multi-physics_calculations (visited on 09/04/2020).

[89] J. Leppänen and M. Aufiero. "Development of an unstructured mesh based geometry model in the Serpent 2 Monte Carlo code". English. In: *Proceedings of the international conference on physics of reactors.* Vol. JAEA-Conf 2014-003. 2015, p. 10. DOI: 10.11484/jaea-conf-2014-003.

[90] J. Leppänen, V. Valtavirta, T. Viitanen, and M. Aufiero. "Unstructured mesh based multi-physics interface for CFD code coupling in the Serpent 2 Monte Carlo code". English. In: *Proceedings of the international conference on physics of reactors.* Vol. JAEA-Conf 2014-003. 2015. DOI: 10.11484/jaea-conf-2014-003.

[91] J. Leppänen. "Modeling of Nonuniform Density Distributions in the Serpent 2 Monte Carlo Code". In: *Nuclear Science and Engineering* 174.3 (2013), pp. 318–325. DOI: 10.13182/NSE12-54. URL: https://doi.org/10.13182/NSE12-54.

[92] T. Viitanen and J. Leppänen. "Explicit Treatment of Thermal Motion in Continuous-Energy Monte Carlo Tracking Routines". In: *Nuclear Science and Engineering* 171.2 (2012), pp. 165–173. DOI: 10.13182/NSE11-36. URL: https://doi.org/10.13182/NSE11-36.

[93] T. Viitanen and J. Leppänen. "Temperature Majorant Cross Sections in Monte Carlo Neutron Tracking". In: *Nuclear Science and Engineering* 180.2 (2015), pp. 209–223. DOI: 10.13182/NSE14-46. URL: https://doi.org/10.13182/NSE14-46.

[94] T. Viitanen and J. Leppänen. "Target Motion Sampling Temperature Treatment Technique with Elevated Basis Cross-Section Temperatures". In: *Nuclear Science and Engineering* 177.1 (2014), pp. 77–89. DOI: 10.13182/NSE13-37. URL: https://doi.org/10.13182/NSE13-37.

[95] M. Aufiero, P. Rubiolo, and M. Fratoni. “Monte Carlo/CFD coupling for accurate modeling of the delayed neutron precursors and compressibility effects in Molten Salt Reactors ”. In: *ANS Annual Meeting*. Vol. 116. San Francisco, United States, June 2017, pp. 1183–1186. URL: http://hal.in2p3.fr/in2p3-01573636.

[96] Y. Li. “Implementation of multiple time steps for the multi-physics solver based on chtMultiRegionFoam”. In: *Proceedings of CFD with OpenSource Software*. Nilsson. H., 2016. URL: http://www.tfd.chalmers.se/~hani/kurser/OS_CFD_20.

[97] C. Fiorina, N. Kerkar, K. Mikityuk, P. Rubiolo, and A. Pautz. “Development and verification of the neutron diffusion solver for the GeN-Foam multi-physics platform”. In: *Annals of Nuclear Energy* 96 (2016), pp. 212–222. ISSN: 0306-4549. DOI: https://doi.org/10.1016/j.anucene.2016.05.023. URL: http://www.sciencedirect.com/science/article/pii/S0306454916303309.

[98] H. Walker and P. Ni. “Anderson Acceleration for Fixed-Point Iterations”. In: *SIAM J. Numerical Analysis* 49 (Aug. 2011), pp. 1715–1735. DOI: 10.2307/23074353.

[99] C. Sanderson and R. Curtin. “Armadillo: a template-based C++ library for linear algebra”. In: *Journal of Open Source Software* 1.2 (2016), p. 26. DOI: 10.21105/joss.00026. URL: https://doi.org/10.21105/joss.00026.

[100] J. Tu, G.H. Yeoh, and C. Liu. *Computational Fluid Dynamics: A Practical Approach*. International Series of Monographs in the Science of the Solid State. Elsevier Science, 2012. ISBN: 9780080982779. URL: https://books.google.fr/books?id=qVqA8P5jd4kC.

[101] S. V. Patankar and D.B. Spalding. “A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows”. In: *International Journal of Heat and Mass Transfer* 15.10 (1972), pp. 1787–1806. ISSN: 0017-9310. DOI: https://doi.org/10.1016/0017-9310(72)90054-3. URL: https://www.sciencedirect.com/science/article/pii/0017931072900543.

[102] S. Patankar. *Numerical heat transfer and fluid flow*. Taylor & Francis, 2018.

[103] R.I. Issa. “Solution of the implicitly discretised fluid flow equations by operator-splitting”. In: *Journal of Computational Physics* 62.1 (1986), pp. 40–65. ISSN: 0021-9991. DOI: https://doi.org/10.1016/0021-9991(86)90099-9. URL: https://www.sciencedirect.com/science/article/pii/0021999186900999.

[104] T. Holzmann. *Mathematics, Numerics, Derivations and OpenFOAM®*. Nov. 2019.

[105] T. Holzmann. *OpenFOAM guide/The PIMPLE algorithm in OpenFOAM*. 2018. URL: https://openfoamwiki.net/index.php/OpenFOAM_guide/The_PIMPLE_algorithm_in_OpenFOAM (visited on 09/11/2020).

[106] Nuclear Energy Agency - NEA. *International Handbook of Evaluated Criticality Safety Benchmark Experiments - ICSBEP*. Tech. rep. Organization for Economic Co-operation and Development - OECD, 2013.

[107] J. Wang, K. Fan, H. Gao, and Q. Xie. "Coupled Dynamics of burst in a fast burst reactor with reactivity addition". In: *Proceedings of Physics of Reactors (PHYSOR) 2020* (Cambridge, UK). 2020.

[108] Z. M. Prince and J. C. Ragusa. "Multiphysics reactor-core simulations using the improved quasi-static method". In: *Annals of Nuclear Energy* 125 (2019), pp. 186–200. ISSN: 0306-4549. DOI: https://doi.org/10.1016/j.anucene.2018.10.056. URL: http://www.sciencedirect.com/science/article/pii/S030645491830584X.

[109] G. Kooreman and D. P. Griesheimer. "Predictor-Corrector Quasi-Static Method for Tightly-Coupled Reactor Multiphysics Transient Calculations". In: *PHYSOR 2020* (2020).

[110] G. Truchet, P. Leconte, A. Santamarina, E. Brun, F/ Damian, and A. Zoia. "Computing adjoint-weighted kinetics parameters in Tripoli-4®by the Iterated Fission Probability method". In: *Annals of Nuclear Energy* 85 (2015), pp. 17–26. ISSN: 0306-4549. DOI: https://doi.org/10.1016/j.anucene.2015.04.025. URL: http://www.sciencedirect.com/science/article/pii/S030645491500225X.

[111] C. Fiorina, S. Radman, A. Scolaro, and A. Pautz. "A reduced order accelerator for time-dependent segregated neutronic solvers". In: *Annals of Nuclear Energy* 121 (2018), pp. 177–185. ISSN: 0306-4549. DOI: https://doi.org/10.1016/j.anucene.2018.07.032. URL: http://www.sciencedirect.com/science/article/pii/S0306454918303815.

[112] D. A. Meneley, K. O. Ott, and E. S. Wiener. *Fast-reactor kinetics: the QX1 code*. Tech. rep. ANL–7769. Argonne National Laboratories, 1971.

[113] D. Lathouwers, M. Tiberga, A. Cammi, E. Cervi, S. Lorenzi, R. de Oliveira, M. Aufiero, A. Laureau, and P. Rubiolo. *Verification and Validation studies.* Tech. rep. D4.3. SAMOFAR, 2018.

[114] J. E. Guyer, D. Wheeler, and J. A. Warren. "FiPy: Partial Differential Equations with Python". In: *Computing in Science & Engineering* 11.3 (2009), pp. 6–15. DOI: 10.1109/MCSE.2009.52. URL: http://www.ctcms.nist.gov/fipy.

[115] J. Blanco, P. Rubiolo, and E. Dumonteil. "Multiphysics coupling analysis for spent fuel pool loss of coolant accident". In: *ICNC 2019.* Sept. 2019.

[116] K. Suyama, Y. Uchida, T. Kashima, T. Ito, and T. Miyaji. *Burn-up Credit Criticality Safety Benchmark Phase III-C Nuclide Composition and Neutron Multiplication Factor of a Boiling Water Reactor Spent Fuel Assembly for Burn-up Credit and Criticality Control of Damaged Nuclear Fuel.* Tech. rep. NEA-NSC-R–2015-6. Nuclear Energy Agency of the OECD (NEA), 2016.

[117] A. Guglielmelli, F. Rocchi, and G. Bandini. "SCALE 6.1.3 and Serpent 2.1.24 criticality safety analysis of a Fukushima Daiichi-like Spent Fuel Pool". In: *NENE 25th International Conference Nuclear Energy for New Europe, Portoroz, Slovenia* (2016).

[118] O. Coindreau. *Task final report (AIR-SFP).* Tech. rep. D6.8.5. AIR-SFP NUGENIA+ European Research Project, NUGENIA, 2016.

[119] W. Wagner and A. Pruß. "The IAPWS Formulation 1995 for the Thermodynamic Properties of Ordinary Water Substance for General and Scientific Use". In: *Journal of Physical and Chemical Reference Data* 31.2 (2002), pp. 387–535. DOI: 10.1063/1.1461829. URL: https://doi.org/10.1063/1.1461829.

[120] International Atomic Energy Agency (IAEA). *Open-source Nuclear Code for Reactor Analysis (ONCORE).* 2020. URL: https://www.iaea.org/topics/nuclear-power-reactors/open-source-nuclear-code-for-reactor-analysis-oncore (visited on 10/21/2020).

[121] P. Cardiff, Ž. Tuković, P. De Jaeger, M. Clancy, and A. Ivanković. "A Lagrangian cell-centred finite volume method for metal forming simulation". In: *International Journal for Numerical Methods in Engineering* 109.13 (2017), pp. 1777–1803. DOI:

10.1002/nme.5345. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.5345.

[122] The OpenFOAM Foundation. *OpenFOAM v8 User Guide*. URL: https://cfd.direct/openfoam/user-guide (visited on 10/15/2020).

[123] ANSYS Inc. *ANSYS FLUENT 12.0/12.1 Documentation*. 2009. URL: https://www.afs.enea.it/project/neptunius/docs/fluent/index.htm (visited on 10/21/2020).

[124] W. P. Jones and B. E. Launder. "The prediction of laminarization with a two-equation model of turbulence". In: *International Journal of Heat and Mass Transfer* 15.2 (1972), pp. 301–314. ISSN: 0017-9310. DOI: https://doi.org/10.1016/0017-9310(72)90076-2. URL: http://www.sciencedirect.com/science/article/pii/0017931072900762.

[125] A. N. Kolmogorov. "Equations of turbulent motion in an incompressible fluid". In: *Dokl. Akad. Nauk SSSR* 30.4 (1941), pp. 299–303. URL: https://cds.cern.ch/record/739754.

[126] D. Wilcox. "Reassessment of the scale-determining equation for advanced turbulence models". In: *AIAA Journal* 26 (1988), pp. 1299–1310.

[127] D. C. Wilcox. *Turbulence modelling for CFD*. DCW Industries, La Cañada, 1993.

[128] F. R. Menter, M. Kuntz, and R. Langtry. "Ten Years of Industrial Experience with the SST Turbulence Model". In: ed. by K. Hanjalic, Y.Nagano, and M.Tummers. Begell House, Inc., 2003.

[129] OpenFOAM. *OpenFOAM: User Guide v2006: Linear eddy viscosity models*. URL: https://www.openfoam.com/documentation/guides/latest/doc/guide-turbulence-ras-linear-eddy-viscosity-models.html (visited on 10/21/2020).

[130] V. Richard. *Rapport de stage: Mise au point d'un modèle multi-physique pour la conception d'un réacteur de propulsion nucléaire dans l'espace*. Tech. rep. Laboratoire de Physique Subatomique et de Cosmologie (LPSC/IN2P3/CNRS), 2019.

[131] S. Levy. *Two-phase flow in complex systems*. John Wiley & Sons, 1999.

[132] V. V. Ranade. *Computational flow modeling for chemical reactor engineering*. Elsevier, 2001.

[133] H. Rusche. “Computational fluid dynamics of dispersed two-phase flows at high phase fractions”. PhD Thesis. Imperial College London (University of London), Jan. 2003. URL: http://hdl.handle.net/10044/1/8110.

[134] OpenFOAM. *A.1 Standard solvers.* URL: https://www.openfoam.com/documentation/user-guide/standard-solvers.php (visited on 11/20/2020).

[135] A. Mackenzie. “A hybrid slurry CFD model: Euler-Euler to Euler-Lagrange”. In: *Proceedings of CFD with OpenSource Software.* Nilsson. H., 2016.

[136] D. Cappelli. “A detailed description of reactngTwoPhaseEulerFoam focussing on the links between mass and heat transfer at the phase interface”. In: *Proceedings of CFD with OpenSource Software.* Nilsson. H., 2018. URL: http://dx.doi.org/10.17196/OS_CFD#YEAR_2018.

[137] S. K. Orungati. “Implementation of cavitation models into the multiphaseEulerFoam solver”. In: *Proceedings of CFD with OpenSource Software.* Nilsson. H., 2017. URL: http://dx.doi.org/10.17196/OS_CFD#YEAR_2017.

[138] C. Fiorina, I. Clifford, M. Aufiero, and K. Mikityuk. “GeN-Foam: a novel OpenFOAM®based multi-physics solver for 2D/3D transient analysis of nuclear reactors”. In: *Nuclear Engineering and Design* 294 (2015), pp. 24–37. ISSN: 0029-5493. DOI: https://doi.org/10.1016/j.nucengdes.2015.05.035. URL: http://www.sciencedirect.com/science/article/pii/S0029549315003829.

[139] C. Grüninger. “Numerical coupling of Navier-Stokes and Darcy flow for soil-water evaporation”. PhD Thesis. Stuttgart : Eigenverlag des Instituts für Wasser- und Umweltsystemmodellierung der Universität Stuttgart, May 2017. URL: http://elib.uni-stuttgart.de/handle/11682/9674.

[140] G. S. Beavers and D. D. Joseph. “Boundary conditions at a naturally permeable wall”. In: *Journal of fluid mechanics* 30.1 (1967), pp. 197–207.

[141] P. G. Saffman. “On the boundary condition at the surface of a porous medium”. In: *Studies in applied mathematics* 50.2 (1971), pp. 93–101.

[142] K. Mosthaf, K. Baber, B. Flemisch, R. Helmig, A. Leijnse, I. Rybak, and B. Wohlmuth. “A coupling concept for two-phase compositional porous-medium and single-phase compositional free flow”. In: *Water Resources Research* 47.10 (2011). DOI: https:

//doi.org/10.1029/2011WR010685. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2011WR010685.

[143] I. Rybak. "Mathematical modeling of coupled free flow and porous medium systems". Stuttgart Research Centre for Simulation Technology und der Fakultät Mathematik und Physik der Universität Stuttgart, Dec. 2016. URL: http://elib.uni-stuttgart.de/handle/11682/8995.

[144] M. Amara, D. Capatina, and L. Lizaik. "Coupling of Darcy-Forchheimer and compressible Navier-Stokes equations with heat transfer". In: *SIAM Journal on Scientific Computing* 31.2 (2009), pp. 1470–1499. DOI: 10.1137/070709517. URL: https://hal.inria.fr/inria-00341870.

[145] T. Koch, D. Gläser, K. Weishaupt, S. Ackermann, M. Beck, B. Becker, S. Burbulla, H. Class, E. Coltman, S. Emmert, T. Fetzer, C. Grüninger, K. Heck, J. Hommel, T. Kurz, M. Lipp, F. Mohammadi, S. Scherrer, M. Schneider, G. Seitz, L. Stadler, M. Utz, F. Weinhardt, and B. Flemisch. "DuMu^x 3 – an open-source simulator for solving flow and transport problems in porous media with a focus on model coupling". In: *Computers & Mathematics with Applications* (2020). ISSN: 0898-1221. DOI: 10.1016/j.camwa.2020.02.012.

[146] B. Flemisch, M. Darcis, K. Erbertseder, B. Faigle, A. Lauser, K. Mosthaf, S. Müthing, P. Nuske, A. Tatomir, M. Wolff, and R. Helmig. "DuMux: DUNE for multi-{phase, component, scale, physics, . . . } flow and transport in porous media". In: *Advances in Water Resources* 34.9 (2011). New Computational Methods and Software Tools, pp. 1102–1112. ISSN: 0309-1708. DOI: 10.1016/j.advwatres.2011.03.007.