

THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Development of fine-mesh  
methodologies for coupled  
calculations in Light Water Reactors

KLAS JARETEG



Department of Physics  
CHALMERS UNIVERSITY OF TECHNOLOGY  
Göteborg, Sweden 2017

Development of fine-mesh methodologies for coupled calculations in Light Water Reactors  
KLAS JARETEG  
ISBN 978-91-7597-626-6

©Klas Jareteg, 2017

Doktorsavhandling vid Chalmers tekniska högskola  
Ny serie nr 4307  
ISSN 0346-718X

Division of Subatomic and Plasma Physics  
Department of Physics  
Chalmers University of Technology  
S-412 96 Göteborg  
Sweden  
Telephone +46 (0)31 772 1000

Cover: Scalar neutron flux at mid-evelation for the thermal group in a steady-state coupled simulation (described in Section 4.2).

Typeset using L<sup>A</sup>T<sub>E</sub>X and TikZ

Chalmers Reproservice  
Göteborg, Sweden 2017

Development of fine-mesh methodologies for coupled calculations in Light Water Reactors

KLAS JARETEG

Division of Subatomic and Plasma Physics

Department of Physics

Chalmers University of Technology

## ABSTRACT

This thesis presents fine-mesh multiphysics methodologies and algorithms for numerical predictions of the behavior of Light Water Reactor (LWR) cores. The multiphysics aspects cover the distribution of neutrons, the fluid flow of the coolant and the conjugate heat transfer between the solid fuel pins and the fluid coolant. The proposed schemes are aimed at fine-mesh coupled effects, directly resolving the interdependencies of the different fields on the finest scales of the computations.

The solver is developed for both steady-state and transient LWR scenarios. For the steady-state simulations, the neutronics is solved both by the lower order, diffusion equation and the higher order, discrete ordinate transport method, and for transient cases by the former. The thermal-hydraulic solver is based on a computational fluid dynamics (CFD) approach. The implementation utilizes a finite volume method (FVM) computational framework, and to achieve feasible computational times, high performance computing (HPC) aspects such as parallelization by domain decomposition are considered.

The implemented tool is applied to cases of parts of a fuel assembly, analyzing systems of up to  $15 \times 15$  fuel pins and successfully resolving sub-pin resolution of all fields. Furthermore, the transient fine-mesh neutronic solver is verified based on a novel scheme utilizing the system response to a local perturbation.

In addition, the multiphase flow problem encountered in Boiling Water Reactors (BWRs) is studied. First, the transport of bubbles under subcooled boiling conditions is simulated based on a population balance approach. The novel formulation is shown to increase the computational efficiency and to capture a large range of bubbles sizes with few degrees of freedom. Second, the typical Eulerian-Eulerian approach for two-phase flow is studied from a stability and dynamics perspective. The latter investigations highlight the complexity of the two-fluid formulation and indicate the spontaneous emergence of meso-scale void structures under adiabatic conditions.

*Keywords:* Coupled neutronics/thermal-hydraulics, CFD, nuclear reactor multiphysics, multiphase flow



## Preface

The process of writing and defending a thesis is a long and interesting one, not least so in the politically and socially complicated field of nuclear power. The Swedish nuclear industry and research have both seen drastic reverses during my few years in the field and the future is, for better and worse reasons, indecisive. As a PhD student it has however been my privilege to focus primarily on the technological contributions. It is my sincere hope that the work presented in this thesis will help to improve the field of nuclear engineering.

It has been my intention to write a thesis which could be read not only by experts in nuclear reactor simulations but also by a more general audience. In particular the introduction is written to give an overview of some current simulation schemes, their shortcomings and some of the late developments in HPC multiphysics simulations for reactors. However, to keep this summary at a reasonable length I can only allow myself to scratch the surface.

The second chapter (Computational Methodology) is meant both as an introduction to methods relevant for the current work as well as to explain and discuss some of the choices that were done during the code development and setup of the multiphysics framework. My hope is that this chapter is clear and detailed enough as to allow someone else starting a similar project to understand and actually learn some of the practical details that I spent (a lot of) time on.

The thesis makes an attempt to tackle the multiphysics problem of the reactor core. As such, my research is at best multifaceted and at worst scattered. Nevertheless, it has been my intention to provide a holistic description of the methodologies developed and applied in this thesis. Consequently, a significant proportion of the details of the results of the applications are left for the interested reader to find in the appended papers.

I am grateful towards a large number of people (as further expressed in the Acknowledgements), and not least to my three supervisors for assisting and contributing to this work. All remaining errors are my own.

Klas Jareteg  
Gothenburg 2017/06/27



## Appended papers

This thesis is an introduction to and a summary of the work published in the following papers

PAPER I      **K. Jareteg**, P. Vinai, C. Demazière. "Fine-mesh deterministic modeling of PWR fuel assemblies: Proof-of-principle of coupled neutronic/thermalhydraulic calculations". *Annals of Nuclear Energy* 68 (2014), pp. 247–256

PAPER II      **K. Jareteg**, P. Vinai, S. Sasic, C. Demazière. "Influence of an  $S_N$  solver in a fine-mesh neutronics/thermal-hydraulics framework". *PHYSOR 2014, September 28 - October 3, Japan* (2014)

PAPER III      H. Ström, S. Sasic, **K. Jareteg**, C. Demazière. "Behaviour and Stability of the Two-Fluid Model for Fine-Scale Simulations of Bubbly Flow in Nuclear Reactors". *International Journal of Chemical Reactor Engineering* 13.4 (2015), pp. 449–459

PAPER IV      **K. Jareteg**, P. Vinai, S. Sasic, C. Demazière. "Coupled fine-mesh neutronics and thermal-hydraulics - modeling and implementation for PWR fuel assemblies". *Annals of Nuclear Energy* 84 (2015), pp. 244–257

PAPER V      **K. Jareteg**, R. Andersson, C. Demazière. "Development and test of a transient fine-mesh LWR multiphysics solver in a CFD framework". *M&C 2015, Nashville, Tennessee* (2015)

PAPER VI      **K. Jareteg**, S. Sasic, P. Vinai, C. Demazire. "A numerical framework for bubble transport in a subcooled fluid flow". *Journal of Computational Physics* 345 (2017), pp. 373–403

PAPER VII      **K. Jareteg**, H. Ström, S. Sasic, C. Demazière. "On the dynamics of instabilities in two-fluid models for bubbly flows". *Chemical Engineering Science* 170 (2017), pp. 184–194

PAPER VIII      C. Demazière, V. Dykin, **K. Jareteg**. "Development of a point-kinetic verification scheme for nuclear reactor applications". *Journal of Computational Physics* 339 (2017), pp. 396–411

## Contribution report

PAPER I      KJ formulated the model, implemented the code, performed all simulations and wrote the paper together with the coauthors.

PAPER II     KJ formulated the model, implemented the code, performed all simulations and wrote the first draft of the paper. The final version of the paper was prepared with support from the coauthors.

PAPER III    KJ contributed to the analysis, ideas for simulations and the preparation of the final version of the paper.

PAPER IV    KJ formulated the model, implemented the code, performed all simulations and wrote the first draft of the paper. The final version of the paper was prepared with support from the coauthors.

PAPER V     KJ formulated the model and implemented the code together with RA, performed all simulations and wrote the first draft of the paper. The final version of the paper was prepared with support from the coauthors.

PAPER VI    KJ formulated the model, implemented the code, performed all simulations and wrote the first draft of the paper. The final version of the paper was prepared with support from the coauthors.

PAPER VII   KJ formulated parts of the model, performed all simulations, contributed to the analysis and wrote the first draft of the paper. The final version of the paper was prepared with support from the coauthors.

PAPER VIII   KJ contributed with the implementation of the time-dependent version of the code and the corresponding verification simulations and data analysis, and was involved in the writing of the corresponding sections in the paper.

## Other related publications, not included in the thesis

**K. Jareteg**, P. Vinai, C. Demazière. "Investigation of the possibility to use a fine-mesh solver for resolving coupled neutronics and thermal-hydraulics". *M&C 2013, Sun Valley, Idaho* (2013)

H. Ström, S. Sasic, **K. Jareteg**, C. Demazière. "On the validity of the two-fluid model for simulations of bubbly flow in nuclear reactors". *Proceedings of the 13th International Conference on Multiphase Flow in Industrial Plants (MFIP-13)*. 2014

C. Demazière, **K. Jareteg**. "Developing a course in nuclear reactor modelling and going from campus-based to web-based teaching". *PHYSOR 2014, September 28 - October 3, Japan* (2014)

C. Demazière, **K. Jareteg**. "Students and teachers perspectives in going from campus-based to web-based teaching". *Conference on Teaching and Learning (Konferens om Undervisning och Lärande KUL2014)*. 2014

H. Ström, **K. Jareteg**, S. Sasic, C. Demazière. "Two-fluid model analyses of instabilities and non-uniformities in bubbly gas-liquid flows". *12th International Conference on Gas-Liquid and Gas-Liquid-Solid Reactor Engineering (GLS12)*. 2015

**K. Jareteg**, S. Sasic, P. Vinai, C. Demazière. "Development of a Coupled Two-Fluid/DQMOM Methodology for Heated Bubbly Flows". *CMFF2015, Budapest, Hungary* (2015)

**K. Jareteg**, H. Ström, S. Sasic, C. Demazière. "Numerical investigation of instabilities in the two-fluid model for CFD simulations of LWRs". *M&C 2015, Nashville, Tennessee* (2015)

A. Jareteg, **K. Jareteg**, S. Sasic. "Formulation of stresses in dry granular flows". *CMFF2015, Budapest, Hungary* (2015)

E. Pettersen, C. Demazière, **K. Jareteg**, E. Schönfeldt T, B. Lauritzen. "Development of a Monte-Carlo based method for calculating the effect of stationary fluctuations". *M&C 2015, Nashville, Tennessee* (2015)

K. Subramani Kannan, **K. Jareteg**, N. Lassen, J. Carstensen, M. Edberg Hansen, F. Dam, S. Sasic. "Design and performance optimization of gravity tables using a combined CFD-DEM framework". *Powder Technology* 318 (2017), pp. 423 –440

C. Demazière, V. Dykin, **K. Jareteg**. "Development and Test of a New Verification Scheme for Transient Core Simulators". *ANS Annual Meeting June 11-15, San Francisco, 2017*. 2017

G. Sardina, **K. Jareteg**, H. Ström, S. Sasic, C. Demazière. "The nature of instabilities in bubbly flows / a comparison between Eulerian-Eulerian and Eulerian-Lagrangian approaches". *MFIP2017, Desenzano del Garda, 13-15 September 2017* (2017)

# Abbreviations & Nomenclature

## Abbreviations

<b>BWR</b>	Boiling Water Reactor
<b>CFD</b>	Computational Fluid Dynamics
<b>CHF</b>	Critical Heat Flux
<b>CHT</b>	Conjugate Heat Transfer
<b>CRUD</b>	Chalk River Unidentified Deposits
<b>CV</b>	Control volume
<b>DNS</b>	Direct Numerical Simulation
<b>DQMOM</b>	Direct Quadrature Method of Moments
<b>FDM</b>	Finite Difference Method
<b>FEM</b>	Finite Element Method
<b>FVM</b>	Finite Volume Method
<b>HPC</b>	High performance computing
<b>LES</b>	Large Eddy Simulation
<b>LWR</b>	Light Water Reactor
<b>MUSIG</b>	Multiple-size-group method
<b>NK</b>	Neutron kinetics
<b>NPP</b>	Nuclear Power Plant
<b>PBE</b>	Population balance equation
<b>PDE</b>	Partial differential equation
<b>PISO</b>	Pressure Implicit with Split Operator
<b>PWR</b>	Pressurized Water Reactor
<b>RANS</b>	Reynolds Averaged Navier-Stokes equations
<b>SIMPLE</b>	Semi-Implicit Method for Pressure Linked Equations
<b>S<sub>N</sub></b>	Discrete ordinates method
<b>TH</b>	Thermal-hydraulics

# Nomenclature

## General

$\delta_{ij}$	Kronecker delta
$\rho$	Density
$\Theta$	Example quantity
$\Theta_p$	$\Theta$ for cell $p$
$\Theta_s$	Face interpolated value of $\Theta$
$\Omega$	Angular direction
$g$	Gravitational acceleration
$\mathbf{r}$	General space coordinate
$S_f$	Face area vector
$S_\Theta$	Source term for $\Theta$
$t$	General time coordinate
$V$	Volume of mesh cell

## Neutronics

$\beta$	Fraction of delayed neutrons
$\gamma$	Energy per fission
$\lambda$	Decay constant
$\mu$	Average scattering angle
$\nu$	Neutron fission yield
$\rho_A$	Atomic density
$\sigma_x$	Microscopic cross-section for reaction
$x$	
$\Sigma_a$	Absorption cross-section
$\Sigma_f$	Macroscopic fission cross-section
$\Sigma_s$	Macroscopic scattering cross-section
$\Sigma_{s0}$	Macroscopic isotropic scattering cross-section
$\Sigma_T$	Total macroscopic cross-section
$\phi$	Scalar neutron flux
$\varphi$	Expansion coefficient in real spherical harmonics base
$\chi$	Fission neutron spectrum
$\Psi$	Angular neutron flux
$C$	Precursor concentration
$D$	Diffusion coefficient
$F$	Fission source term
$G$	Number of energy groups
$\mathbf{J}$	Neutron current
$k_{\text{eff}}$	Multiplication factor
$n$	Neutron density
$P_N$	Power density
$P_l$	Legendre polynomials
$R_{lr}$	Real spherical harmonics
$S$	Scattering source
$w$	Quadrature weight

## TH - Single-phase

$\alpha$	Isothermal compressibility coefficient
$\beta$	Thermal expansion coefficient
$\epsilon$	Dissipation of turbulent kinetic energy
$\mu$	Dynamic viscosity
$\mu_t$	Turbulent kinetic viscosity
$\tau$	Stress tensor
$c_p$	Specific heat capacity at constant pressure
$h$	Instantaneous enthalpy
$H$	Time averaged enthalpy
$k$	Turbulent kinetic energy
$K$	Thermal conductivity
$p$	Instantaneous pressure
$P$	Time averaged pressure
$q''$	Surface heat flux
$q''''$	Volumetric heat source
$T$	Temperature
$u$	Instantaneous velocity
$\mathbf{U}$	Time averaged velocity

## TH - multiphase

$\alpha$	Void fraction
$\xi$	Abscissa (bubble size)
$\mu$	Dynamic viscosity
$\rho$	Density
$\bar{\tau}$	Stress tensor
$\bar{\tau}^t$	Turbulent stress tensor
$\Phi$	Time resolved uniformity index
$C$	Condensation rate
$d_{43}$	Mean diameter
$d_b$	Bubble size
$f$	Average number density
$g$	Vapour phase
$i$	Phase
$j$	Bubble size index
$l$	Liquid phase
$\mathbf{M}$	Momentum transfer due interfacial forces
$N$	Number of abscissas
$P$	Pressure
$S$	Source term for condensation, aggregation and breakage
$t$	Turbulent quantity
$\mathbf{U}$	Phase velocity
$w$	Weight

# Contents

<b>Abstract</b>	<b>I</b>
<b>Preface</b>	<b>III</b>
<b>Appended papers</b>	<b>V</b>
<b>Contribution report</b>	<b>VI</b>
<b>Other related publications, not included in the thesis</b>	<b>VII</b>
<b>Abbreviations &amp; Nomenclature</b>	<b>IX</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Multiphysics in the reactor core . . . . .	1
1.2 Neutronics and thermal-hydraulic simulations . . . . .	5
1.2.1 Lattice and core simulations of the neutronics . . . . .	5
1.2.2 System codes for thermal-hydraulics . . . . .	7
1.2.3 Segregated multiphysics schemes . . . . .	9
1.3 High-fidelity core simulations . . . . .	9
1.3.1 Motivations for novel approaches . . . . .	9
1.3.2 Overview of multiscale and multiphysics approaches . . . . .	10
1.4 Objectives of the research work . . . . .	11
1.4.1 Fine-mesh multiphysics simulation objectives . . . . .	12
1.4.2 Multiphase flow objectives . . . . .	13
1.4.3 Outline of the thesis . . . . .	15
<b>2 Computational methods</b>	<b>17</b>
2.1 High Performance Computing . . . . .	18
2.1.1 A brief historical perspective . . . . .	18
2.1.2 Code efficiency and optimization . . . . .	19
2.1.3 Parallelization . . . . .	20
2.2 The finite volume method . . . . .	23
2.2.1 Equation discretization . . . . .	23

2.2.2	Sparse matrix solvers . . . . .	25
2.2.3	OpenFOAM® . . . . .	26
2.3	Computational scheme . . . . .	28
2.3.1	Grid generation . . . . .	29
2.3.2	Mesh mapping . . . . .	35
2.3.3	Cross-section generation and utilization . . . . .	36
<b>3</b>	<b>Models for the coupled LWR single-phase problem</b>	<b>41</b>
3.1	Formulation of the neutronic problem . . . . .	41
3.1.1	Diffusion approximation . . . . .	44
3.1.2	Discrete ordinates method . . . . .	45
3.2	Single-phase fluid flow and heat transfer . . . . .	47
3.2.1	Turbulence . . . . .	48
3.2.2	Pressure and velocity algorithms . . . . .	49
3.2.3	Heat transfer problem . . . . .	50
3.3	Multiphysics formulation and algorithms . . . . .	51
<b>4</b>	<b>Steady-state coupled solver application and analysis</b>	<b>55</b>
4.1	Implementation and framework details . . . . .	55
4.1.1	SIMPLE algorithm and heat transfer . . . . .	56
4.1.2	Discrete ordinates solver . . . . .	57
4.1.3	Picard iteration scheme . . . . .	58
4.2	Application to a $15 \times 15$ assembly . . . . .	59
4.2.1	Convergence and performance . . . . .	60
<b>5</b>	<b>Transient coupled solver application and analysis</b>	<b>65</b>
5.1	Implementation and framework details . . . . .	65
5.2	Application to a $7 \times 7$ assembly . . . . .	66
5.3	Time-dependent neutronics verification method . . . . .	68
5.3.1	Overview of the methodology . . . . .	70
5.3.2	Application to a two-region slab system . . . . .	71
<b>6</b>	<b>On two-phase flow in LWRs</b>	<b>73</b>
6.1	Perspectives on the simulation challenges . . . . .	74
6.2	Models for two-phase liquid and gas flows . . . . .	76
6.2.1	DNS-like methods . . . . .	76
6.2.2	The two-fluid model . . . . .	77
6.3	Population balance for subcooled bubbly flow . . . . .	79
6.3.1	PBE formulation . . . . .	80
6.3.2	Application to channel flow with condensation . . . . .	81
6.4	On the dynamics of the two-fluid formulation . . . . .	84
6.4.1	Application to adiabatic cases . . . . .	85

<b>7 Conclusions and recommendations for future work</b>	<b>89</b>
7.1 Fine-mesh multiphysics simulations . . . . .	89
7.1.1 Summary of the methodology . . . . .	89
7.1.2 Results and conclusions . . . . .	90
7.2 Two-fluid simulations . . . . .	91
7.2.1 DQMOM coupled to a two-fluid solver . . . . .	91
7.2.2 Two-fluid instability results . . . . .	92
7.3 Future outlook . . . . .	93
<b>Acknowledgements</b>	<b>95</b>
<b>References</b>	<b>97</b>
<b>Papers I-VIII</b>	<b>109</b>
Papers I . . . . .	111
Papers II . . . . .	123
Papers III . . . . .	143
Papers IV . . . . .	157
Papers V . . . . .	173
Papers VI . . . . .	191
Papers VII . . . . .	225
Papers VIII . . . . .	239



---

# CHAPTER 1

---

## Introduction

This thesis begins with, and ultimately originates from, the physics in the core of the nuclear power plant (NPP). The core is at the heart of a reactor in every aspect, located in the center of the plant and containing the fuel with its enormous potential of heat generation from fissioning of heavy nuclei. The reactor core is a complex environment, governed by multiple fields of intertwined and coupled physics influencing the process on a wide range of length scales. This environment is the field of this thesis and the main goal shall be to increase the understanding of how such a system can be simulated with high resolution and accuracy.

This first chapter will introduce the main fields of physics that are of interest in the reactor core and in particular how these are mutually coupled (Section 1.1). To give a context of the contribution of this work, some standard (or even classical) schemes of nuclear reactor core simulations (Section 1.2) are described, followed by some more recent developments in reactor modeling (Section 1.3). Finally, the objectives of this thesis are presented in detail in Section 1.4.

### 1.1 Multiphysics in the reactor core

The core of the Light Water Reactor (LWR) contains the solid uranium fuel pins which contribute with the heat source in the reactor. A controlled chain reaction of fissions results in a continuous and enormous release of energy in the reactor core. The energy is conducted via the solid encapsulation, the cladding, of the fuel and then extracted from the pins and convected out of the core via the heated water. To understand the interplay between the neutron density and the fluid flow and heat transfer in the coolant, a more detailed description of the multiphysics is of interest.

The chain reaction of fissions is governed by the distribution of neutrons in

the core. If the population of neutrons is kept at a statistically steady concentration, the reactor is said to be critical. On the other hand, if there is an increase in neutrons with time, the reactor is in a supercritical state. Reversely, if fewer neutrons are born from fission than disappear due to fission, absorption or leakage, the reactor is in a subcritical state. To achieve an economic utilization of the fuel, the core should be designed to minimize the loss of neutrons due to leakage out of the physical domain and due to other nuclear reactions.

The cross-section for a certain reaction, i.e. the probability for a neutron to interact with a certain target, is determined by properties of the interacting material as well as the energy of the neutron. The microscopic cross-section ( $\sigma$ ) describes the probability for a certain reaction to take place, such as absorption, capture or fission to occur (see details below). Due to the dependence on the density of the target material, the neutron distribution is influenced by all processes which result in variations of the material concentrations. For this thesis, the coupling to the water density is of particular interest. In addition, the temperature of the fuel has a significant effect on the cross-sections due to the so-called Doppler broadening.

### Microscopic and macroscopic cross-sections

The microscopic cross-section gives the probability for a certain reaction to occur given the incident neutron energy such that the reaction rate  $R_x$  is given by

$$R_x(E) = \sigma_x(E)\phi(E)\rho_A, \quad (1.1)$$

where  $\phi(E)$  denotes the neutron flux and  $\rho_A$  the atomic density of the target material. Commonly the combination of the microscopic cross-section and the density is written as a macroscopic cross-section

$$\Sigma_x(E) = \sigma_x(E)\rho_A. \quad (1.2)$$

The figure below shows the microscopic fission cross-section of Uranium-235. For low energies the cross-section is inversely proportional to the neutron velocity, whereas a much more complicated dependence is seen for the intermediate region. The resonances in this energy range constitute a particularly challenging task in the process of cross-section condensation as an average of a longer energy interval in the resonance region strongly depends on the precise spectrum of the neutron flux.

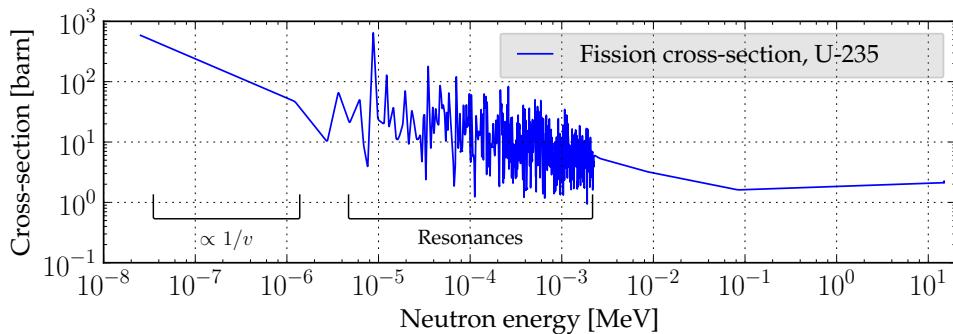
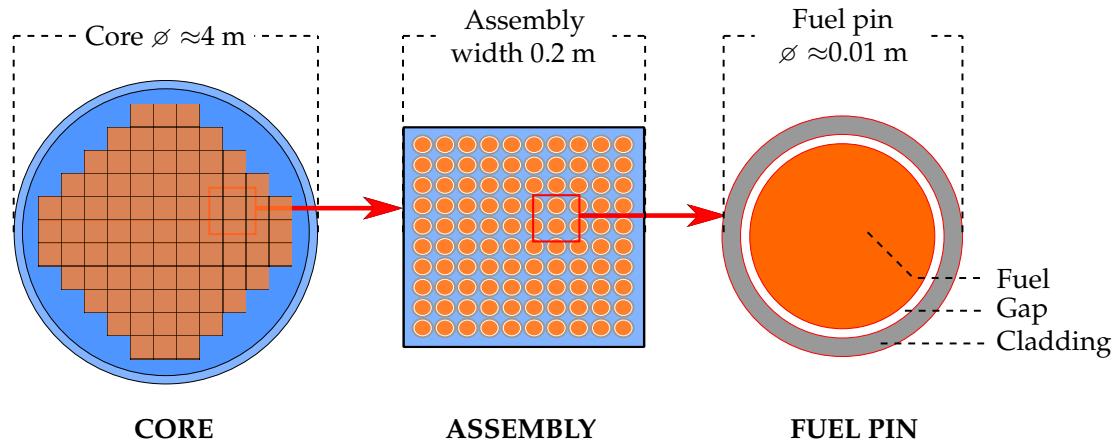


Figure 1.1: Fission cross-section for Uranium-235 [1].



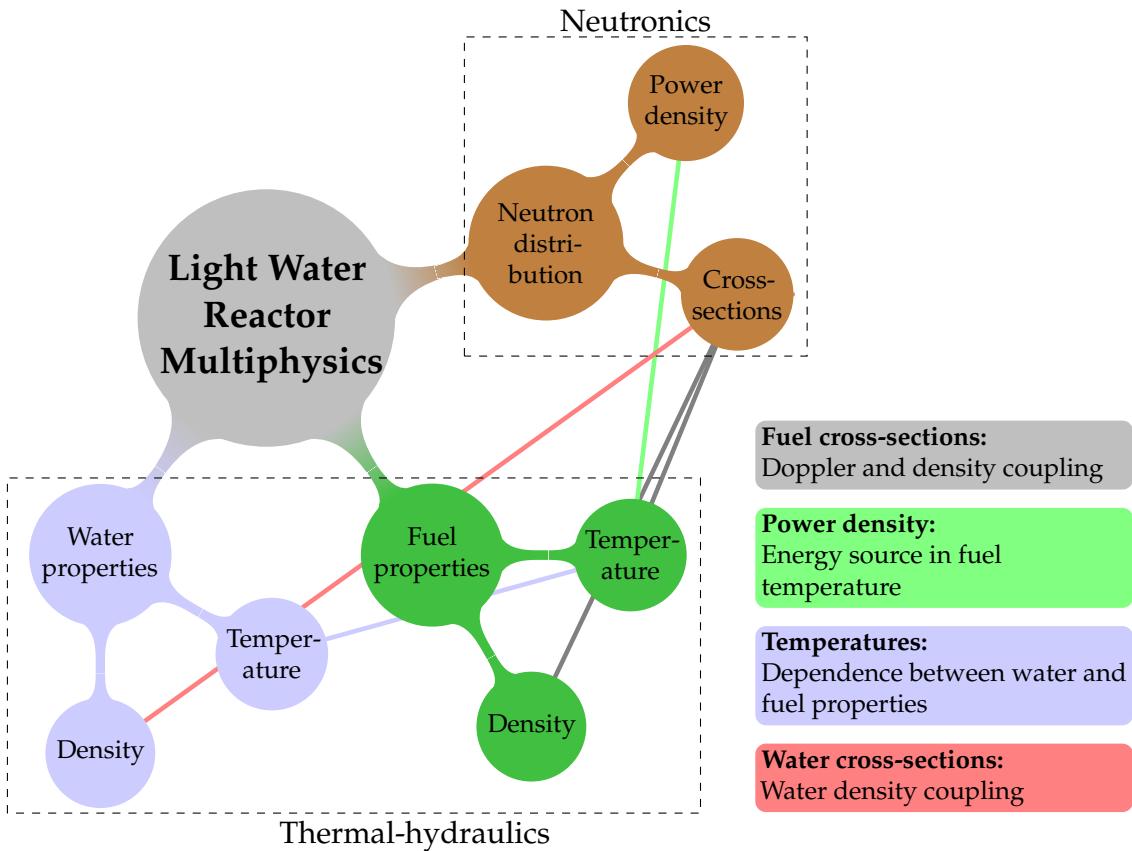
**Figure 1.2:** Schematic drawing of a horizontal plane of a LWR core.

The temperature profile in the fuel is determined by the energy release from the fissions and the conduction in the fuel pins, i.e. the transport of energy in the stack of fuel pellets in each rod in the assembly (see Figure 1.2). The conductivity of the solid fuel matrix is governed by the micro structure as well as the temperature of the material. Due to the neutron irradiation in the core, the fuel and cladding properties change with time in the reactor. The processes responsible for such defects constitute a complete field of material science, and complex phenomena such as cracking of the fuel might take place.

To allow thermal expansion and fuel swelling, LWR fuel rods have a small gap between the fuel and the cladding, initially filled by an inert gas. Due to the small distance between the solids, the major mechanism for heat transfer (under normal reactor operating conditions) is conduction also in the gap. The cladding is the first safety barrier in the reactor, designed to contain the fuel and the fission products. Although no fission events occur in the gap or cladding, these regions are still influential on the neutronics problem as the neutron distribution is affected via capture.

The heat conducted from the cladding is extracted from the core with forced convection. In the case of the LWR, the water acts as the coolant, and a high flow velocity through the core is maintained by pumps. In the case of a Pressurized Water Reactor (PWR) the water is nominally kept in liquid state, whereas a phase change from liquid to vapor is seen in a Boiling Water Reactor (BWR). The high flow velocity results in a turbulent flow with enhanced heat transfer properties. The turbulence in the coolant is further enhanced by the so-called spacers, essentially steel frames holding the fuel pins as well as interrupting the flow and inducing swirls.

In addition to acting as a coolant, the water in a LWR functions as a moderator for the neutrons. The moderation process cools, i.e. slows down, the high energy neutrons born from fission. The benefit of a higher concentration of low energy neutrons is best understood from the energy dependence of the fission



**Figure 1.3:** Diagram of the multiphysics couplings for the neutronics and thermal-hydraulics in the LWR core.

cross-section for U-235 as presented in Figure 1.1. Due to the inverse proportion of the microscopic fission cross-section, the chance for fission events is much larger for slow neutrons. The moderation primarily occurs due to elastic scattering between the neutrons and the hydrogen atoms in the water.

All described physical processes are connected and the reactor core problem is thus a true multiphysics problem in the sense that one field cannot be solved without knowledge of the others. To summarize the different couplings, Figure 1.3 shows the primary dependencies between the neutron behaviour, the so called neutronics, and the fluid flow and heat transfer in the water and the fuel, the so called thermal-hydraulics. The temperature and density of the fuel and the water both influence the neutronics, which in turn affects the fuel temperature directly through the energy release from fission, and water indirectly as the heat is transported from the fuel to the water.

In addition to the multiphysics aspects of the reactor core, the problem is further complicated by the many length scales to be resolved. The schematic representation in Figure 1.2 shows a horizontal plane in the core. The figure outlines a hierarchy of relevant scales, ranging from the full core size, via the fuel assem-

bles, to the separate fuel pins. In principle, the hierarchy could be continued with even smaller scales governing some of the phenomena discussed above (e.g. turbulence in the water and micro structures of the fuel matrix). In the extreme, we could even consider the atomic length scales and the sizes of the nucleons which ultimately govern the interaction between the neutrons and the nuclei.

Due to the extreme range of scales, it is immediately clear that it is not possible to directly resolve all parts of the problem from first principles. In order to solve the ultimate problem of the complete reactor core, we must rely on assumptions and closures from other scales. Inevitably, such closures introduce errors. As the reactor is also a multiphysics environment, the use and derivation of closures for the large scale problems must also consider that coupled physics phenomena occurring at the small scales need to be correctly represented at the larger scales, which is far from trivially granted. The latter is a key aspect to why the focus in this thesis is on fine-mesh simulations, i.e. simulations where the different fields of physics can be directly coupled without the approximations required to solve the full core problem.

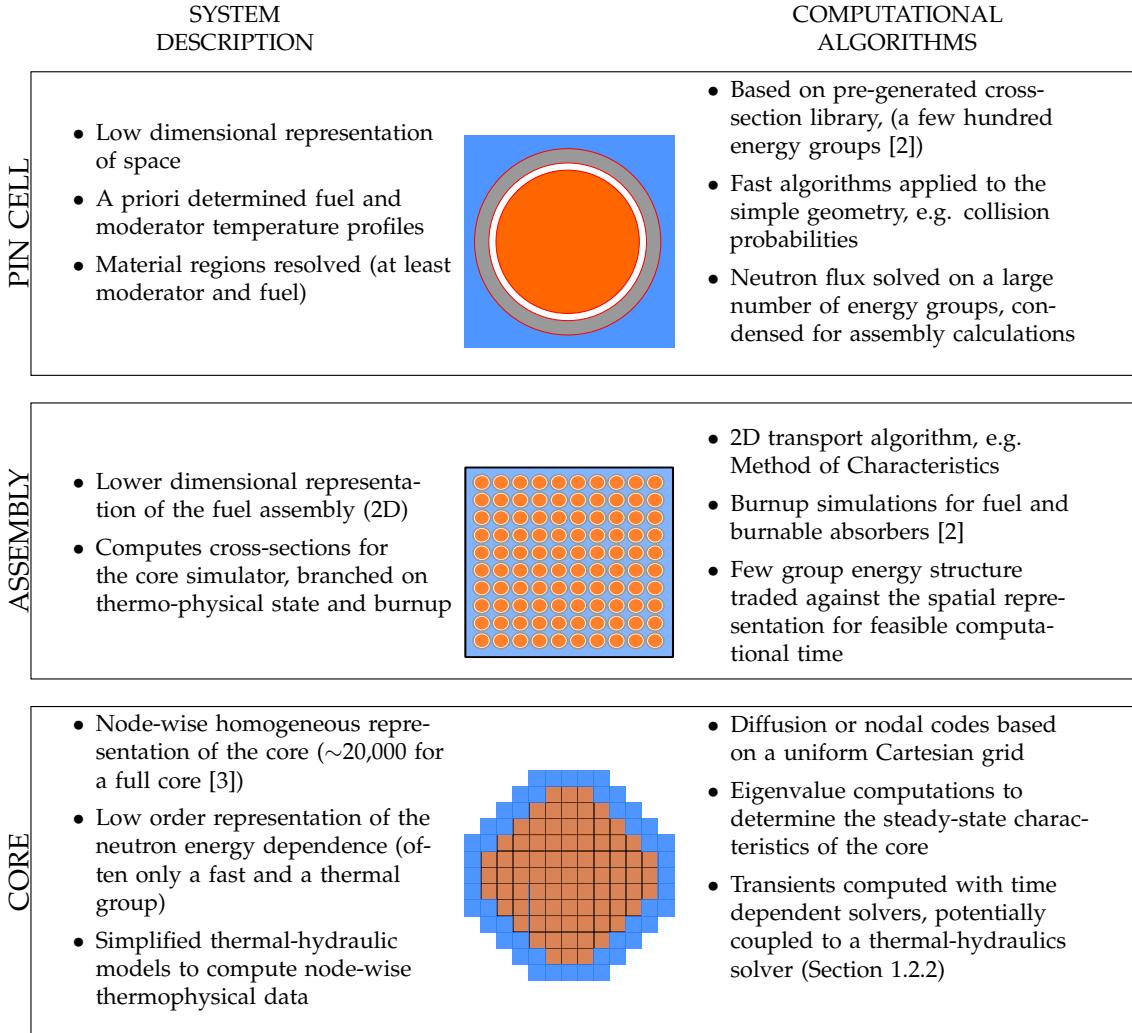
## 1.2 Neutronics and thermal-hydraulic simulations

Given the multiscale and multiphysics problem of the reactor core, we now turn the attention to different options and strategies to simulate and produce numerical predictions of the behavior of the core. To motivate the need of novel algorithms, a brief overview is first presented of some standard schemes applied in routine calculations for the core. The described procedures are well established and the current practices have prevailed for many decades in the same or at least similar shapes. Nevertheless, there are significant limitations and assumptions for multiphysics perspectives and for the physics of the finest scales resolved.

### 1.2.1 Lattice and core simulations of the neutronics

Although a rapid increase is seen in the use of Monte Carlo methods for the neutronic calculations, even on a full core scale, such methods are still too computationally expensive for routine industrial calculations (further discussed in Section 1.3). As a result, the industry still relies on deterministic computations for the neutronic problem, and the predominant schemes have for a long time relied on hierachal algorithms. Such multiscale approaches range from the simulations of a single pin cell (i.e. a fuel pin surrounded by the coolant), to fuel assembly calculations and finally to the full core scale. A brief overview of the hierarchy of scales is shown in Figure 1.4, where the three different levels are outlined together with some standard choices of algorithms.

In detail, the first two stages of the simulations (pin cell and assembly calculations) are typically computed in a lattice code. Conveniently, such a code



**Figure 1.4:** Overview of a multiscale deterministic neutronics scheme progressing from pin-wise 1D calculations, via 2D assembly calculations to full core 3D simulations

generates a library of cross-sections which are branched to cover desired state points for fuel burnout, thermo-physical state of the reactor, control rod positions etc. Due to the large number of different fuel assemblies and feasible states, the lattice calculators must rely on fast algorithms and sufficiently coarse approximations to give useful simulation times.

Relevant for this thesis, the approximations in the lattice code include assumptions of approximate fuel and moderator temperature profiles. Whereas the actual horizontal temperature profiles are multiphysics dependent (as depicted in Figure 1.3), a standard lattice code is run with explicit, and often discrete, temperature profiles.

The last stage, the core calculation, is for LWRs performed on a coarse Cartesian grid corresponding to the fuel assemblies in the core. A large number of

solver variations exist. However, the core simulators again require fast computations as the codes are routinely applied repetitively e.g. in the process of core design.

Furthermore, the core solver relies on lower dimensional thermal-hydraulic algorithms to predict the temperature and void distributions in the core. The computed states are used to interpolate the cross-section tables from the lattice solver. Roughly, each fuel assembly has one associated thermal-hydraulic channel in which the conservation equations of mass, momentum and energy are solved (see e.g. [4]). At the scale of the core solver, the multiphysics couplings are coarse and by no means resolve small fluctuations or pin-wise temperature profiles.

From this crude description of the neutronic scheme it is clear that the methodology is streamlined to give fast computations for the full core problem with the core solver and the cross-section tables computed in the lattice solver. The multiphysics aspects are limited and the direct coupling to the (simplified) thermal-hydraulic solvers is performed only at the coarsest level.

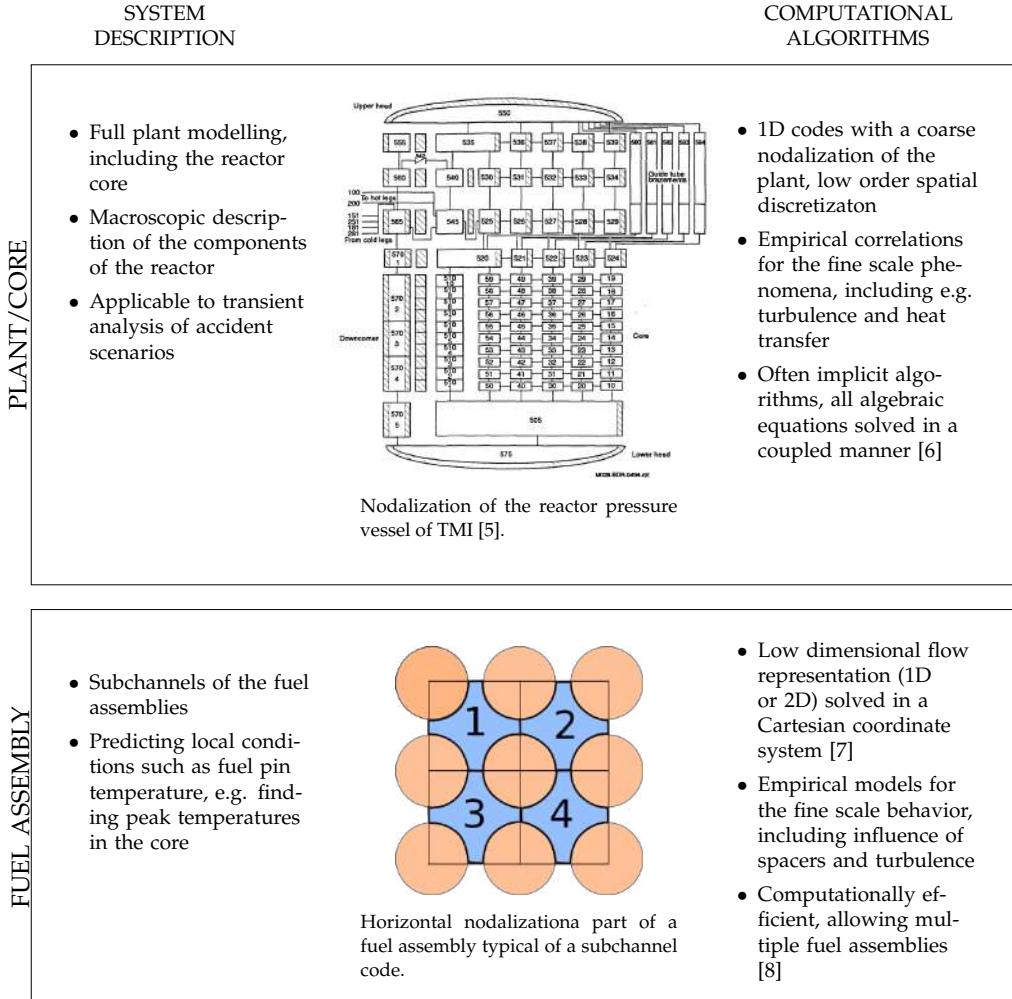
### 1.2.2 System codes for thermal-hydraulics

The thermal-hydraulic counterpart to the hierachal neutronics methodology consists of a full core solver, partitioned on so-called channels, and a sub-channel code simulating one or a few such channels with a higher resolution. Figure 1.5 presents a brief overview and characteristic algorithms for both methodologies. In contrast to the neutronics scheme, the thermal-hydraulic solvers are not necessarily combined in the same workflow.

The goal of the system code is to compute the complete plant response to a set of transient scenarios, including accidents such as a loss-of-coolant accident [9]. The representation of the flow is much simplified, relying on 1D transport equations, discretized with first order schemes in space and time [6]. As illustrated in Figure 1.5, the core, as well as the other components, is treated with a coarse nodalization. As a result of the crude representation, the effects from for example fluid fluctuations are not modeled but included in correlations. The benefit of the coarse approach is the short wall clock time for relatively long transient scenarios, even on desktop computers.

The algorithms for the neutronic response are typically based on much simplified models, such as the 0D point-kinetic model [10]. On the other hand, for scenarios where the neutronic response is crucial, a coupling to a 3D neutronic solver is advantageous (which is further discussed in Section 1.2.3).

In partial contrast to the macroscopic full plant modeling in the system code, a subchannel code is aimed at (more) local conditions in specific fuel assemblies. For safety reasons it is of interest to e.g. find the hottest channels in the reactor core in order to determine the peak cladding temperature (PCT) [11]. For such a



**Figure 1.5:** Overview of the characteristics of a coarse full plant system code and a higher resolution subchannel code.

purpose the subchannel approach gives more information than the coarse nodalization in the system code. From fluid dynamics and heat transfer perspectives, the subchannel code is still very far from the first principles, relying on correlations for pressure drops, heat transfer, multiphase flow and more [12]. Furthermore the flow is still not resolved in 3D manner but only 2D (see e.g. [7]).

In addition to the mentioned schemes, there is of course the option of a further resolved and much more fine grained simulation. In nuclear terms this would be denoted as CFD, although in principle also the previously described models are numerical predictions of fluid flows, thus deserving the epithet CFD. Although there are numerous examples of interesting core related applications of CFD in the open literature, the 30 years of experience of the macroscopic type of modeling should not be forgotten [13]. The current and future potentials of CFD are further discussed below (Section 1.3).

### 1.2.3 Segregated multiphysics schemes

For some accident scenarios the multiphysics aspects need to be better resolved than it is in the case of the system codes with simplified neutronic models [10]. For such scenarios, the classical choice is to perform an external code coupling between a system code and a neutronic core solver. Even though appreciated as multiphysics in a more integrated sense, the coupling algorithms are commonly based on the so-called operator splitting techniques [14] with low order time schemes. Nevertheless, such best estimate approaches are a key tool for current industrial simulation routines, in particular due to the enormous efforts of code verification and validation already spent on this type of simulations for industrial plants. Examples of attempts on higher order time schemes exist, e.g. by implicit formulations of a combined system matrix [15], such and similar attempts are still limited in the resolved scales of multiphysics. The mapping between the codes is often coarse in the sense that only macroscopic quantities or average properties are exchanged. Consequently, there is no increased fidelity in the physics simulated. The coupled calculations are limited by the very static geometries and resolutions of each of the separate codes and the simplistic coupling schemes and mappings. In contrast, the type of projects next described (Section 1.3) are to a majority focused on newly developed tools, without the legacy of the system codes or core solvers, or with more degrees of freedom when it comes to geometry and resolutions.

## 1.3 High-fidelity core simulations

Following the rapid development of the computational capacity, an increasing complexity in the fuel assembly designs and the continuous strive to perform more accurate and precise reactor core simulations, the last ten years have seen a large number of initiatives toward higher fidelity multiphysics simulations of the reactor core. From a larger perspective, the specific goals and aims are as many as the methodologies proposed and many of the initiatives are still in early development stages. Nevertheless, the trend is clear and true HPC applications are growing in interest in the nuclear community. In this section, some of the motivations of such novel schemes are discussed together with an overview of the HPC and high-fidelity efforts from the open literature.

### 1.3.1 Motivations for novel approaches

As evident from the discussion on the classical computational schemes, the separate neutronics and thermal-hydraulic reactor simulations are severely simplified in their representation of the multiphysics. As a motivation for new algorithms for nuclear reactor core simulations, a scattered list is given below. Some of the

points of this list are later referenced and rephrased in the actual objectives of the current thesis (Section 1.4):

- *Unresolved multiphysics* – The lattice codes apply simplified temperature and density profiles in the cross-section generation, implicitly introducing an error in terms of the actual thermophysical state in the fuel and moderator.
- *Small margins require higher resolution* – As plants are power uprated the margins to, for example, critical heat flux (CHF) decrease and arguably the local conditions are of increased interest [16].
- *Void heterogeneities* – Heterogeneities in the void fraction distribution in the subchannels of a BWR are potentially influential on the neutron moderation and thus a multiphysics understanding of such heterogeneities is of interest [17, 18].
- *Fuel behavior* – To simulate e.g. the local influence and deposition of CRUD on the fuel pins, novel fine-mesh and multiphysics schemes are required [19, 20]. Notably, simulation of CRUD deposition is one of the targeted problems in The Consortium for Advanced Simulation of LWRs (CASL) [21].
- *Spacer design* – As the spacers in the fuel assemblies increase in complexity, there is a greater need to understand the influence of the induced turbulence. In addition, CFD predictions of spacers are a cheap way to select the best candidates to a significantly smaller cost than using test rigs. [16, 22–24]

Whereas the list is by no means exhaustive, such and similar issues are interesting and constitute relevant drivers for the development of novel multiphysics and high-resolution strategies.

### 1.3.2 Overview of multiscale and multiphysics approaches

The drive towards stronger coupling and higher resolutions has taken a lot of different shapes, partly due to the fact that the complexities of the physics are mirrored in the complexities of the computations, partly because of unclear standards for reactor core multiphysics and partly because of the very different needs for the different issues mentioned in the previous section.

In many of the attempts on achieving the higher resolution multiphysics of the core, the schemes were based on externally coupled tools, e.g. applying a combination of a Monte Carlo solver for the neutronics and a commercial CFD solver for the thermal-hydraulics [25–27], or a deterministic neutronic solver [28–30], again, coupled to a CFD solver. The sophistication of such couplings varies from efficient data exchanges using external scripts to built-in coupling schemes. Large scale projects such as VERA [31] and MOOSE [19] likewise apply an existing code coupling approach but in massive HPC environments and with a large focus on

efficient parallelization and coupling. Although many of the efforts referenced are of interest from a physics point of view, many should be considered as proofs of principle, paving the way towards practical use of such computations.

In contrast to the external code couplings, there are notable examples of multiphysics solvers with a more integrated focus. An interesting example is again the MOOSE project, which can be employed as a general finite element framework with non-linear solvers and massive parallelization [32]. Many smaller scale projects with a tighter coupling were focused on commercial multiphysics solvers [33, 34], but there are also examples which are based on open source software [35], as is the case in the current thesis. Many of the multiscale and multiphysics initiatives in the open literature are focused on fuel performance in terms of local phenomena, where for example the deposition of CRUD is of importance [19, 20] or for fuel pin mechanistic behavior [36, 37].

Whereas the resolution of the CFD simulations has generally been low in the multiphysics coupled simulations discussed above (see e.g. [28, 30]), there are many pure CFD applications where a high-resolution approach is of particular interest. An example is seen in the simulation of grid-to-rod fretting for which time and space resolved turbulent fluctuations are of interest [22, 38]. Another class is the simulation of fuel pin spacers, where again an accurate prediction of the induced turbulence is of interest to predict the pressure drop in the core as well as the local heat transfer from the fuel pins to the coolant [16, 24, 39].

As mentioned above, multiphase flow is yet another topic for future high-fidelity simulations relevant for nuclear core predictions. Whereas the system codes rely on approximations and empirical relations, there are efforts made to perform 3D simulations on the scale of a fuel assembly (see e.g. Lo and Osman [40]). Although much less developed, there are examples of coupled multiphase CFD and neutronics [41], performed on coarse meshes and for steady-state purposes. However, the severe complexity of the multiple flow regimes makes the multiphase problem a theoretically more complicated CFD problem compared to the single phase counterpart, and much work remains in covering all flow regimes in a consistent manner [42]. An extended introduction connected to multiphase simulations for BWRs is postponed to Chapter 6.

## 1.4 Objectives of the research work

The objectives of the research are divided in two parts. First, the objectives for the fine-mesh multiphysics simulations are described together with the applied assumptions and resulting limitations. Second, the objectives for the investigations and method development for the multiphase flow research are outlined.

### 1.4.1 Fine-mesh multiphysics simulation objectives

The first part of the thesis is concerned with multiphysics simulations of the nuclear reactor core. The primary aim is to develop a computational tool which resolves the multiphysics dependencies already at the finest simulated scales. The physics covered is the same as in the classical coupled schemes (Section 1.2.3), i.e. the flow of the fluid coolant, the conjugate heat transfer (CHT) between the solid fuel pins and coolant and the neutron distribution in the core. However, the simulated scales are much finer, focused at a sub-fuel pin level and with a resolved water temperature and flow profile between each of the separated fuel pins. The described scales are throughout the thesis described as *fine-mesh*.

A part of the objective is to demonstrate the feasibility of such a computational code, including the computational cost and applicability in terms of HPC resources. However, the focus is also to resolve aspects of the multiphysics which are of potential importance for a safe operation of the reactor and, perhaps even more, for design of fuel assemblies.

In specific terms the objectives of the first part are to:

- develop a deterministic computational methodology with a fine-mesh approach to the neutronics for both steady-state and transient simulations, with a cross-section model relevant on a sub-pin scale,
- develop a CFD methodology, including heat transfer and the fluid flow, with a 3D representation of the flow between the fuel pins and treatment of turbulence,
- implement the methodologies in a single, multiphysics, computational tool deployable at computational clusters, and
- apply the solver to both steady-state and transient cases for parts of fuel assemblies.

As such, the objectives are related in a generic sense to many of the points mentioned in Section 1.3.1, and perhaps primarily to the need to resolve the multiphysics and to increase the resolution of the numerical predictions in the core. In relation to the previously referenced literature on coupled neutronics/CFD projects (Section 1.3.2), the current objectives are different in that a single computational tool should be developed, directly treating all covered aspects.

### Assumptions and limitations

Following the defined objectives, there are a number of implicit assumptions in the models, whereof the most important include:

- *No mechanistic modelling of fuel behavior* – As described in Section 1.3.2, many of current high-fidelity simulation schemes are primarily targeted toward

fuel behavior and mechanical properties. The fuel pins are here considered static, rigid bodies with only temperature dependent thermophysical properties.

- *No models for the surrounding environment* – Due to the computational cost (and the limited computational resources at hand) only parts of a fuel assembly are considered in the performed simulations. In principle a coarse model for the surrounding would be beneficial, serving a set of boundary conditions. However, for the presented cases all computations assume an infinite lattice of the simulated environment, which is realized through periodic or symmetry boundary conditions in horizontal direction. For the inlet and outlet conditions the boundary conditions are determined without any actual models of the nozzles or e.g. the turbulent spectra of the fluid flow entering the bottom of the assembly.

Whereas the previous assumptions are limitations of the implemented code, there are some further limitations imposed due to the computational cost of the simulations:

- *System size limited to parts of a fuel assembly* – As a result of the high resolution of the simulations, the computational grids contain a large number of degrees of freedom and thus an extensive computational burden. Due to the available computational resources, the system sizes must thus be limited. However, as part of the objective to implement the tools for HPC environments, proper parallelization schemes are still applied and with larger computational resources the code should be applicable to larger cases.
- *Number of neutronic energy groups limited* – The neutronic calculations are performed on a low number of energy groups (ranging from 2 to 16). Again the reason is the computational cost, and all algorithms are implemented for an arbitrary number of energy groups (an exception to this is the results related to Paper VIII, where the derivation is performed for 2 groups (Section 5.3). Similarly, the simulations performed with the discrete ordinates method are limited in the number of directions.

Practically there are additional limitations in terms of what could be considered within the above scope of the objectives. In particular, the geometrical details of the fuel assemblies are limited to the fuel pins, the gap, the cladding and the coolant, e.g. neglecting the influence of the spacers.

## 1.4.2 Multiphase flow objectives

The second part of the thesis concerns multiphase flow in the reactor core and in particular bubbly flows under subcooled and adiabatic conditions. As is further discussed in Section 6.1, the multiphase flow is challenging for multiple reasons

and the maturity of multiphase CFD is significantly lower than for single-phase. For the mentioned reasons, the research conducted for multiphase flow within this thesis has a more generic character than the fine-mesh multiphysics simulations. Nevertheless, the goal of the conducted studies is to increase the understanding of bubbly flows with the two-fluid model, which could be used for low void fraction simulations in the fine-mesh multiphysics solver.

Due to the extreme computational cost connected with the interface resolving methods, such methods are not of practical interest for simulations of systems of the size of a fuel assembly. Instead, the focus is on the two-fluid model, which gives only an average representation of the phases, with no explicit tracking of the interface between the gas and the liquid in the two-phase flow. The model is further introduced and discussed in Section 6.4, but for the sake of formulating the objectives for the multiphase research two of the issues with the two-fluid model is here outlined.

First, as a result of the lack of representation of the interface between the faces, information such as bubble sizes and shapes are unknown in the two-fluid formulation. A potential remedy for this is to introduce a population balance equation (PBE) to track one or more properties of the bubbles to a significantly lower cost as compared to explicitly computing the bubble interfaces. Such an approach is of interest not the least for diabatic simulations where the bubble distribution will change not only due to bubble breakage or aggregation but also due to condensation and evaporation. In practical terms, a two-fluid approach complemented by a PBE is a good candidate for simulation of the subcooled and bubbly flow regimes in a BWR, and as such the framework is worth investigating, also from the fine-mesh multiphysics point of view.

Secondly, from a more general perspective, the two-fluid formulation has previously been shown to be prone to numerical issues, not the least due to an apparent lack of hyperbolicity for some types of flow. Although a wide range of remedies have been proposed, the underlying potential stability issues are still of major interest for the application of the two-fluid model. In particular, the dynamic behavior of the model is key to the predicted mass and heat transfer within the reactor core, and emergence of heterogeneities in the flow can only be trusted if the underlying equations are understood to be sound. Again, investigations on the dynamics of the two-fluid formulation is of interest for the larger perspective of fine-mesh simulations as the dynamics and appearance of void heterogeneities are of potential importance in the coupling to the neutron distribution.

In specific terms, the objectives related to modeling of multiphase flow within this thesis are to:

- investigate the stability of the two-fluid model in terms of phase heterogeneities for bubbly gas-liquid flows, and
- develop novel methodologies for simulation of vapor bubbles in a subcooled liquid using a the two-fluid model complemented by a PBE.

As the case of multi-phase flow is particularly challenging from a CFD perspective and again limitations to the scope are necessary. The main simplifications applied in the developed models and performed simulations include:

- *No simulation of wall phenomena* – The bubbly flows arises due to evaporation at the wall. The bubble growth and departure are governed by micro structures at the surface as well as the flow at the wall. Such aspects are not covered in the thesis. Instead the transport of the bubbles is the primary target.
- *Empirical correlations for condensation* – In the simulation of subcooled bubbly flows, the condensation of the bubbles is, again, ruled by local conditions at the surface of the bubble. In this work no interface tracking simulations of the condensation are performed. Instead, empirical relations are used for the condensation rate.
- *No specific treatment of high void fractions* – The objectives are primarily related to low void fraction regions. This is not to say that high void fractions with other regimes than bubbly flow are not of interest. Rather, the bubbly flows are perhaps the most well researched and still there are significant opportunities for further research. From a larger perspective it is also natural to first focus on the low void fraction regimes.

### 1.4.3 Outline of the thesis

The thesis is structured in seven chapters, whereof this introductory chapter is the first. Chapter 2 gives an introduction to the computational methodologies later applied in the thesis, including detailed descriptions of mesh generation, cross-section generation and parallelization algorithms. In Chapter 3 the models for coupled LWR single phase problems are given for both neutronics and thermal-hydraulics. Next, Chapter 4 presents the implementation and an application to a steady-state multiphysics problem. In Chapter 5, the methodology is extended to transient conditions and again applied to a small lattice of fuel pins. Chapter 6 describes some of the complexities of the two-phase flow in BWRs, and presents a proposed algorithm for subcooled boiling and, additionally, simulations regarding the dynamics of two-phase CFD solvers. Finally, Chapter 7 gives a conclusion and recommendations for future work in the areas of multiphase flow for reactor core applications as well as fine-mesh multiphysics simulations.



---

## CHAPTER 2

---

# Computational methods

To perform multiphysics simulations of a nuclear reactor is, inevitably, strongly connected to development of computational methods, coding, and not least HPC. To approach both the complexity and the sheer size of the system, we need efficient algorithms and numerical methods run on modern hardware and implemented in the right languages. I therefore find it reasonable and enlightening to introduce the computational techniques, which are key to this thesis. As a matter of fact, a large effort has been invested in choosing and developing sufficiently performant algorithms; a task which has been equally challenging and joyful.

Due to the many fields of physics and thus many numerical solvers required, it would be difficult, not to say impossible, to within the same PhD project develop all the necessary computer code from scratch. Instead, I partly use some existing tools and codes, in some cases extended for the purpose of the project, which substantially increase progress and reduce the development time. At the same time, it is of large value to have full transparency of all codes and algorithms. With the current trend of open-source initiatives this is viable. An accessible code base gives the possibility to perform rapid development and the opportunity to modify and extend the software. This has been a cornerstone in the work for this thesis.

This chapter introduces the computational tools applied throughout the remaining chapters of the thesis. A brief introduction to HPC is given in Section 2.1, including a small historical perspective on HPC and the development of computers and efficient code in parallel algorithms. In Section 2.2, an outline of the key elements of the finite volume method (FVM) is presented together with the library which lies as the foundation for the developed multiphysics solver as well as the two-phase solvers later discussed. Finally, Section 2.3 introduces the specific framework developed within this thesis and its use of existing software.

## 2.1 High Performance Computing

High performance computing involves simulations or other computational tasks that are employed on multiple processors or multiple computers. In the area of computational physics and numerical simulations in general, HPC is necessary to solve large problems which would lead to prohibitive long simulation times on a laptop or desktop computer. Now, as we shall see, the notion of a large computational problem has changed and continues to change with the ever increasing capacity of the supercomputers. The computations performed in this thesis would have been almost impossible already 20-25 years ago, even considering, at that time the largest supercomputers in the world.

To further widen the concept of HPC, aspects such as computational efficiency and utilization of the hardware must also be considered. In particular, even though a large computational cluster is exploited, the single CPU utilization is still vital. Furthermore, the single CPU optimization is relevant irrespective of the cluster size. As discussed below (see Figure 2.1), the computational resources for this project have been limited to around maximum 80 and in average 20 CPUs on one of the Swedish computational resources.

### 2.1.1 A brief historical perspective

The history of HPC and supercomputers goes back to the 1960s, with computers such as LARC [43], featuring two CPUs. The early 1970s saw the emergence of RISC computers, where the CDC6600 was the first to use the idea of a reduced instruction set to simplify the CPU [44]. From the later part of the 1970s, HPC was dominated by vector computers which utilized a single operation on a vector of data (so called SIMD architecture) [45]. During the following two decades (ca. 1975-1995) the supercomputer CPUs were considerably more complex than personal computers. In the latter half of the 1990s this changed, with the emergence of clusters built up of thousands of simpler, commodity, CPUs. During the 2000s the trend with massively parallel supercomputers continued, with machines such as Blue Gene [46].

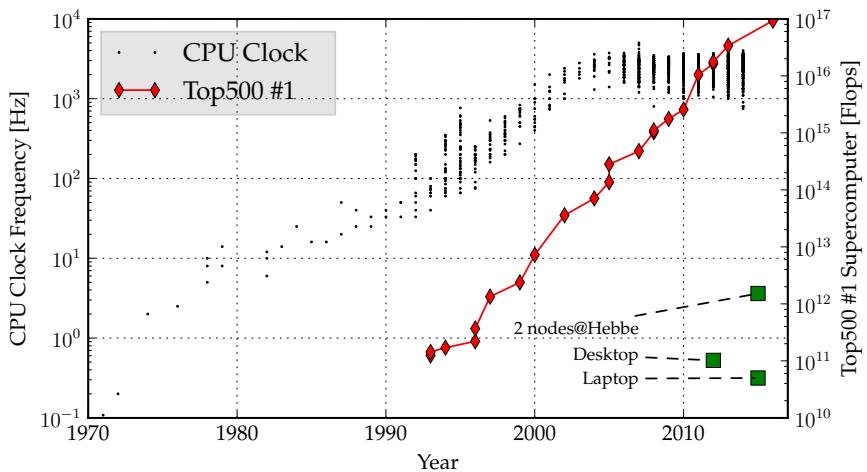
Although history has seen a large number of architectures of hardware the trend is clear; the computational capability is growing exponentially (see Figure 2.1). With the trend of ever increasing computational capacity it is tempting to rely on accelerated hardware, putting less effort in software development. However, in reality, the situation is rather the reversed one. Currently, the trend of the largest supercomputers is to use more heterogeneous architectures with accelerators or graphic cards connected to each standard computer. To use the full capacity of such resources major efforts on software development might be required to utilize the resources.

From a nuclear engineering perspective, it is interesting to note that, whereas

the nuclear industry for a long time has taken a lead in the development of HPC resources, Turinsky [47] points out that during the mid 1980s the industry turned from the use of large computers to standard desktop machines. Although the industry to a large extent relies on codes running on desktop computers, many of the above described efforts (Section 1.3) are again heavily focused on taking the lead in use of large scale computations.

### Hardware development and resources

The trend for the growth of the supercomputers is shown in Figure 1, with a linear increase in the floating point operations per second (FLOPS) in the log diagram. In relation to the single CPU performance, the increase in clock frequency of the CPUs is also important. However, as indicated by the trend in the figure, the growth in clock frequency has stalled and we cannot expect the next generation of CPUs to necessarily keep accelerating our applications. To quote Herb Sutter, the convener of the ISO C++ committee, "The free lunch is over" [48].



**Figure 2.1:** Development of the CPU clock frequency over time [49] (data extracted from [50]) with a comparison of the #1 computing cluster according to Top500 [51]. The green squares indicate desktop and laptop computers of the thesis author, with Flops estimated by Intel Math Kernel Library Benchmarks [52].

### 2.1.2 Code efficiency and optimization

To write a fast computational code we need to consider both hardware aspects, such as e.g. cache sizes and memory bandwidth, as well as software aspects such as algorithms and code languages. Although a PhD project in multiphysics of nuclear reactors does not specifically target such areas, it is arguably of high value with a general knowledge of such code aspects for the development of a HPC framework.

In order to illustrate some of the many challenges in code efficiency and optimization, a few of the key aspects for the multiphysics simulations performed are briefly mentioned:

- *Memory bandwidth* – To solve the discretized neutronic and thermal-hydraulic equations, a sparse matrix solver is applied (Section 2.2.2). The standard iterative sparse matrix solvers typically rely on matrix-vector multiplication, which in turn are limited by the bandwidth of the memory [53]. Whilst the modern CPU architecture relies on multiple levels of caches to speed up the data fetching, our large sparse systems will not be close to fit the cache.
- *Memory locality* – Not only sparse matrix calculations are restricted by slow memory access in relation to the CPU throughput. For data processed together it is always beneficial to keep the data close in memory. An important example is the ordering of the computational mesh cells which could help to minimize the number of fetches from memory (and the larger cache levels).
- *Disk access* – The disk (spinning disk or SSD) is the slowest data storage on the computer and we generally want to avoid excessive write or read operations. As an example, we can avoid disk access by condensing the results as much as possible already in the simulation stage, minimizing the storage operations.
- *Vectorized operations* – A modern CPU support vectorized operations where multiple variables are processed within the same instruction (see e.g. [54]). On a commodity CPU this means that the throughput of arithmetic operations for double precision (64 bit) numbers increases with a factor of two to eight (see e.g. [55]).

For any of the above aspects we are helped both by the CPU routines (e.g. for handling data fetching) and the compiler (e.g. for translating code in to vectorized machine instructions). Nevertheless, it must be in the interest of the developer to understand the basics of the hardware and software interaction. Any small knowledge helps to increase the understanding of the computer and what we can expect in terms of CPU efficiency and performance.

Directly or indirectly, the programming language is an additional key aspect for good utilization of the hardware. HPC applications have historically been implemented in compiled languages, where no additional effort is spent on run-time code interpretation. Whereas FORTRAN is a classical choice for HPC for nuclear applications in particular, a large proportion of modern software is developed using other languages, not seldom C++. The latter has the advantage of a large user base (outside the nuclear community and outside HPC) with good compiler and library support.

### 2.1.3 Parallelization

A fast single CPU code is a good starting point for a fast parallelized code. Modern architectures support parallelization on many different levels, which might

require significantly different efforts from the developer. The three most commonly discussed parallelization regimes include:

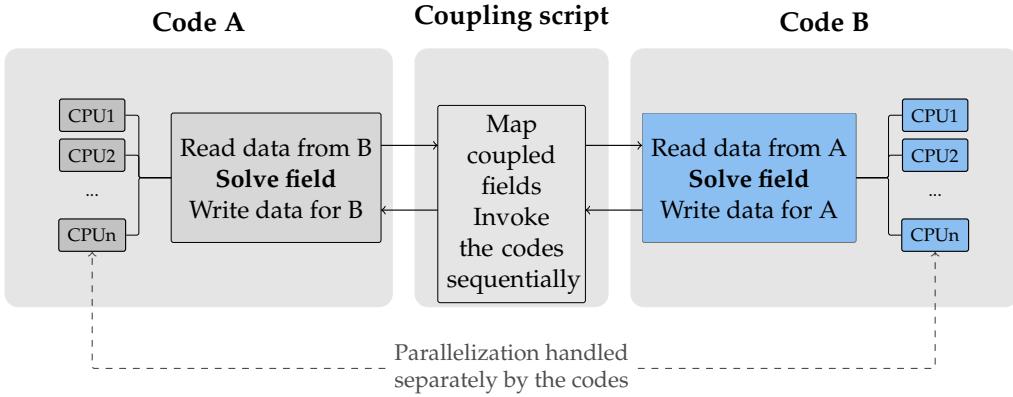
- *Shared memory* – The easiest way to parallelize the code is often to use all available threads on the same computer, which allows the memory to be shared between the different processes. Shared memory parallelization is limited to the number of CPUs on the motherboard (i.e. 2-6 on a commodity desktop computer). A major advantage is that the interprocessor communication is avoided, i.e. the parallelization incurs no substantial overhead.
- *Message passing* – To combine multiple separate computers (aka nodes) on e.g. a computer cluster, message passing is utilized. In contrast to the previous scheme, no shared memory exists and the application must handle exchange of all data common between the different computers. While this is de facto standard seen in e.g. commercial CFD solvers (as well as the open source alternatives) it has drawbacks in terms of interprocessor communication overhead and complexity of data exchange scheduling.
- *Accelerators/Graphic cards* – A recent trend on some of the largest supercomputers has been to use separate hardware for computational acceleration. For example, graphic card computations based on, e.g., CUDA [56] is a current trend in which massive computational performance can be achieved on a single gaming graphics card. The major drawback here is the induced complexity of another hardware architecture.

In addition, for completely independent simulations, we can of course parallelize by running multiple separate processes and combine the results. The latter is e.g. used in some Monte Carlo codes, and was exploited in the cross-section generation (Section 2.3.3), where the system can be duplicated since the neutron histories are assumed independent. A further detailed introduction to different types and levels and some key aspects for nuclear engineering parallelization is given by Calvin and Nowak [57].

In the same manner as the above discussion on efficiency and optimization, the code developer will benefit from general knowledge about the parallelization. However, in many cases an underlying framework for parallelization is desirable, e.g. a parallelized matrix solver library. In the current project the parallelization is achieved using message passing as defined in the MPI standard [58]. The system is decomposed in multiple domains, each solved in a separate process. The message passing handles the communication by, simply put, exchanging the boundary values of the domains.

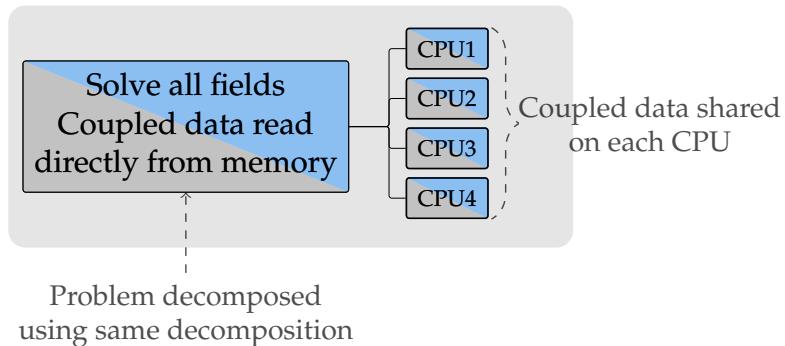
In more specific terms of reactor core simulations, many of the CFD/neutronics multiphysics examples from the literature are formulated as a multiple code scheme. Such an approach is exemplified in Figure 2.2. Advantageously, each of the codes can use the existing parallelization capabilities, which is efficient from a development point of view. However, as apparent from the figure, the communication is handled in a gather-scatter manner, where the internal parallelization, e.g. by

domain decomposition, must gather data before the exchange to the other code occurs. After retrieving the data from the other code a scatter must again take place. This clearly limits the efficiency of the coupling and induces further complications [31].



**Figure 2.2:** Example of data transfer in the multiple codes approach [Paper IV].

In contrast, in a single code approach, as applied in the thesis, all parts of the multiphysics problem can be decomposed in the same spatial domains as exemplified in Figure 2.3. In such a scheme, the data transfer between e.g. the neutronics and the thermal-hydraulics can be performed directly on each separate CPU or thread. As the same, monolithic, application runs both the CFD and the neutronics there are no costs associated with waiting for one of the fields to be finished or similar. Instead all fields are directly solved in the same code. This is particularly important for transient applications where a large number of data exchanges are required.



**Figure 2.3:** Example of single code coupling scheme avoiding external transfer [Paper IV].

## 2.2 The finite volume method

In this thesis all partial differential equations (PDEs) are solved using the finite volume method (FVM). FVM is the classical choice for CFD applications, including many commercial examples (e.g. Star-CCM+ [59] and ANSYS Fluent [60]), as well as long term open source projects (e.g. Gerris [61], OpenFOAM® [62] and MFIX [63]). Although the finite difference method (FDM) was the historical choice, in particular for the very early CFD research [64], it is much less popular today. FDM suffers from restrictions on the grids and the conservation properties are less favorable than in FVM [65]. Furthermore, the finite element method (FEM) is applied in some CFD codes (e.g. COMSOL [66]), but still only in a minority of the commercially available tools. FEM has undoubtedly many advantages, not the least for higher order discretization, but still to some extent considered less mature than FVM for fluid systems [67].

In terms of 3D implementations of neutronics (i.e. core solvers), the classical choices include the FDM or nodal methods [68]. However, in terms of differential operators in the neutron diffusion and transport equation, the FVM is again a viable option. To fulfill the objective of a single multiphysics code, the neutronic equations are here solved with the FVM, as done for the CFD. In particular, FVM is well suited for unstructured meshes required in the fine-mesh approach in this thesis.

### 2.2.1 Equation discretization

The finite volume method relies on a discretization of the computational domain into control volumes (CVs). The process of discretizing the domain is considered separately below (Section 2.3.1) and at this point only the discretization of the conservation equations on an existing grid of CVs is considered. Even though a full description of the discretization mathematics is out of scope for the current thesis, a brief and very basic overview is given. A more complete and general picture can be found in numerous CFD books and publications (for good examples see e.g. [65, 69, 70]).

For simplicity we consider a standard example from CFD, namely the transport equation of the generic (scalar) quantity  $\theta$ :

$$\frac{\partial \theta \rho}{\partial t} + \nabla \cdot (\rho \mathbf{U} \theta) - \nabla \cdot (D_\theta \nabla \theta) = S_\theta(\theta), \quad (2.1)$$

which is transported in a fluid by a convective velocity  $\mathbf{U}$ , with a fluid density  $\rho$ , a diffusivity  $D_\theta$  and some source term  $S_\theta$ . For the sake of brevity, notation on the space and time dependence has been removed from all quantities in eq. (2.1). Upon integration over the CV and after applying the Gauss theorem we get

$$\int_V \frac{\partial \theta \rho}{\partial t} dV + \sum_{\forall f} \mathbf{S}_f \cdot (\rho \mathbf{U} \theta)_f - \sum_{\forall f} \mathbf{S}_f \cdot (D_{\theta,f} \nabla \theta_f) = \int_V S_\theta dV, \quad (2.2)$$

where the divergence operator convective term and the diffusion term are transformed to discrete sums over the faces of the control volume by application of the Gauss theorem.

Concerning the space dependence, FVM assumes a linear variation of a field both with respect to time and space and, for the collocated approach, cell centered values of all fields [70]. Accordingly, the volume integrals in eq. (2.2) are converted to multiplications of the integrand and the volume of the current cell. For the right hand side source term a linearization in the field can be performed, increasing the implicitness of the method, such that

$$\int_V S_\theta(\theta) dV = V_p \theta_p S_p + V_p S_u, \quad (2.3)$$

where  $p$  indicates the cell currently under consideration,  $S_p$  and  $S_u$  correspond to the coefficients for the implicit and explicit parts of the linearization, respectively. This is of particular interest for the implementation of the neutronics, where linearizations of both the fission and scattering source terms are needed.

For the convective term, face values of the velocity, density and the generic quantity are required. The interpolation to faces is of major importance for the accuracy as well as the robustness of the method. Whereas low order methods, such as the upwind differencing guarantee boundedness, higher order schemes such as central differences come at the expense of issues with stability [69].

Similar to the convection, the diffusion term requires a face interpolation of the density, the diffusivity and the gradient of the field ( $\nabla\theta$ ). For orthogonal meshes the gradient can be directly estimated as the difference between the cell center values in the cells on each side of the face. For non-orthogonal meshes on the other hand, additional explicit terms are needed (for a detailed overview see [70]).

To handle the time dependence, eq. (2.2) is integrated in time. The time derivative is then approximated as

$$\int_t^{t+\Delta t} \int_V \frac{\partial \theta \rho}{\partial t} dV dt = \rho_p \frac{\theta_p^n - \theta_p^o}{\Delta t} V_p, \quad (2.4)$$

where the indices  $n$  and  $o$  refer to the new and the old time step, respectively. What concerns the rest of the terms of eq. (2.2), for all variables a time step ( $n$  or  $o$ ) must be chosen. If the old time step is inserted an explicit scheme is achieved, whereas using the new time step (currently solved for) results in an implicit scheme. The previous is first order accurate and limited by the Courant number whereas the latter is in theory first order accurate but unconditionally stable. In practice however, the first order accuracy of both schemes are undesirable and therefore the unconditional stability is of secondary importance. Other, higher order time schemes are therefore more popular, and in the current thesis the default choice is the Crank-Nicholson scheme (second order in time).

For later discussion on mesh generation (see Section 2.3.1) the following conditions imposed by the applied FVM should be noted:

- *Only first neighbors* – In the discretization of the convective term and the diffusion term relations to neighboring cells are computed. In the current work only first neighbors are considered, resulting in a so called compact computational molecule. The latter is beneficial for unstructured meshes, where second neighbors are not trivially defined.
- *No hanging nodes* – The faces of the CVs are assumed to be one-to-one in the sense that no single face is connected to more than one face of another cell.
- *No curved faces* – All faces of the discretized mesh are assumed to be flat. A curved boundary (such as the outer radius of the fuel pin) is thus represented by a set of flat faces. In contrast to FEM, where the order of accuracy can be increased by an increase of degree of base polynomials and with a non-linear representation of the boundary (see e.g. [71]), FVM instead relies on a refinement of the mesh.

Although seemingly restrictive, the above assumptions are important to achieve a performant method with fast discretization and a, generally, well structured sparse matrix system with few off-diagonal elements.

## 2.2.2 Sparse matrix solvers

The discretization in time and space results in a set of algebraic equations. It should be noted that all non-linearities are linearized (as discussed for the source term), and thus result in a sparse linear matrix system.

In terms of computations, the discretization (and matrix assembly) routines might take significant computational time. However, the major effort of the CFD solver is spent on solving the linear system itself. Whereas direct methods are useful for very small matrices, such methods are out of question for the large and sparse matrices found in CFD, both due to excessive memory usage and too high a cost in terms of floating point operations. Instead, iterative methods are used such as e.g. the Gauss-Seidel method for which a huge number of acceleration techniques have been developed over the years [72]. Again more efficient methods are found in so-called projection methods, which include Krylov subspace solvers such as Conjugate Gradient (CG) for symmetric matrices and Bi-Conjugate Gradient Stabilized (BiCGStab) which are routinely used in many CFD solvers. For certain classes of problems, relaxation techniques, such as the Algebraic Multigrid method (AMG), are a good option.

Although the mathematical details of the sparse matrix solvers are out of scope for the thesis (for good introductory texts see e.g. [73, 74]), a general knowledge is of importance both for selecting the correct class of methods based on the matrix properties (hyperbolic, parabolic or elliptic) as well as for tuning the

methods to achieve good performance. The latter is often a (time consuming) process of trial and error. Nevertheless, due to the long simulation times for highly-resolved simulations presented in this thesis, well chosen parameters can easily save days of simulation time.

In the presented example (eq. (2.1)), the solution variable  $\theta$  was assumed to be a scalar quantity. For vectorial (or tensorial) solution variables a set of such conservation equations need to be solved for each computational cell. Additionally, with interdependent conservation equations (such as pressure and velocity, or different neutron energy groups) the couplings must be resolved. In principle all equations could be discretized together, and after applying the required linearizations, a coupled system of equations is obtained. In CFD the early day computers were limited by computer memory and as a result, the matrices were kept at a minimum size, resulting in the development of segregated algorithms for the pressure and velocity dependence. The coupled aspects are further discussed in Section 3.3.

### 2.2.3 OpenFOAM®

All solvers developed in this thesis are implemented as extensions of the finite volume framework OpenFOAM®[75]<sup>1</sup>, which is a C++ open-source tool for development of CFD solvers. The code consists of a library of general routines focused on formulation, discretization and solution of tensorial equations with FVM. Furthermore, a set of existing solvers are provided, including classical approaches such as segregated pressure and velocity solvers for single phase flow, multi-fluid solvers, conjugate heat transfer and more. For an overview, see the documentation [77, 78] or third party literature on the code [79, 80].

Considering the objectives of this thesis, some of the key aspects and benefits of the library are:

- *Unstructured meshes* – OpenFOAM® handles unstructured meshes and implements a computational molecule consisting of the nearest neighbors. The unstructured meshes captures the geometry of the fuel pins and potentially more complicated core geometries (e.g. the spacers). The library provides corresponding FVM discretization and differential operators for the fluid flow and heat transfer applications.
- *High level equation format* – The code provides a high-level equation interface in which the system equations can be directly written in a user-friendly manner which effectively hides the underlying equation discretization and treatment of boundary conditions, etc.

---

<sup>1</sup>OpenFOAM® has since its first release in 2004 been forked multiple times, and in this thesis the multiphysics code and the subcooled population balance transport code are implemented in the community version foam-extend [76] whereas the two-fluid stability computations are performed in OpenFOAM foundation version [62].

- *Existing CFD algorithms* – Importantly, the library provides the standard segregated fluid solver approaches, including SIMPLE [81] and PISO [82], with a Rhie-Chow for collocated grid [83]. The previous are the foundations for the fluid solver for the coolant of the core.
- *Turbulence models* – Libraries of both Reynolds Averaged models (RANS) and Large Eddy Simulations (LES) are included, which enable fast integration of turbulence in new solvers.
- *Parallelization routines* – The library supports MPI parallelization and all necessary data operators (such as discretization routines, algebra operators) are written with the parallelization in mind. The code uses a so-called zero halo layer domain-decomposition, i.e. partitioning of space in non-overlapping domains.

Except for the above mentioned points, the freely available source code (released under GNU GPL 3) is another important aspect as it can be extended not only on the top-level but also for key, lower level, elements of the library.

For this thesis, many parts of the library are exploited without any modifications. However, for the multiphysics and two-fluid parts of the project, significant code development was needed. Some of the newly developed parts<sup>2</sup> are (in order of size of code):

- *Neutronics solvers* – A library for neutronic simulations (diffusion and transport) was required. This includes handling of the cross-sections and treatment of both diffusion and transport equations ( $S_N$ ).
- *LWR multiphysics solver* – The multiphysics application was developed to combine neutronic solvers and fluid solvers.
- *Population balance solvers* – For Paper VI, PBE solvers for DQMOM and MUSIG were developed, including the relevant boundary conditions for vapor generation.
- *Mesh intersection utility* – A library for intersecting overlapping meshes was developed. The tool handles the data transfer between the neutronic mesh and the separate meshes for the solid fuel pins and the coolant (see Section 2.3.1).

---

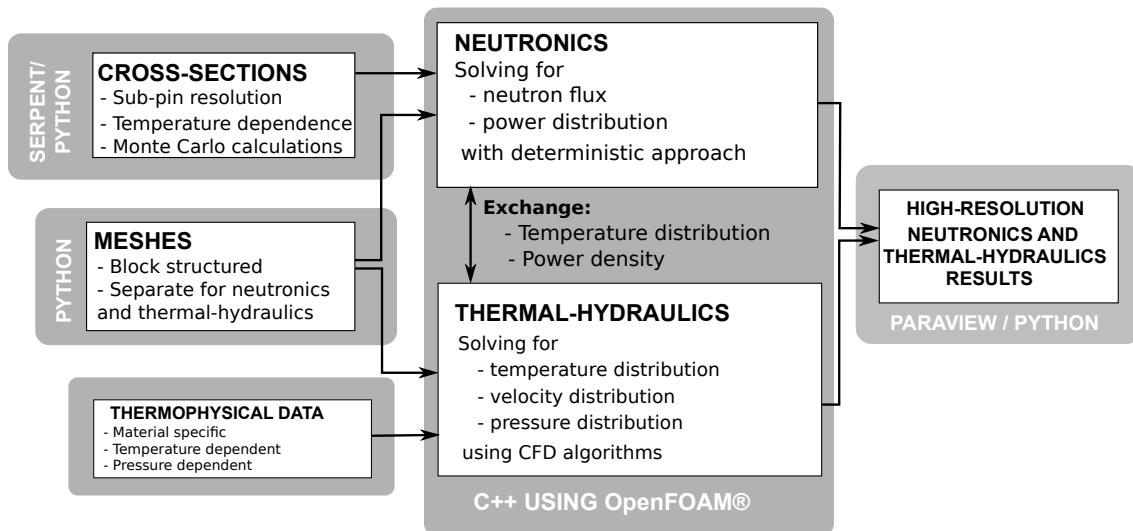
<sup>2</sup>The mentioned additions to the code are not part of OpenFOAM®, and the work here presented is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OpenFOAM and OpenCFD trade marks. For any ambiguity on the rights (and wrongs) of the trademark please refer to <http://www.openfoam.com/legal/trademark-guidelines.php>.

## 2.3 Computational scheme

An overall view of the computational workflow of the multiphysics solver is presented in Figure 2.4. The leftmost column corresponds to the pre-processing steps, namely mesh generation (Section 2.3.1), cross-section generation (Section 2.3.3) and formatting and acquiring of thermo-physical data for the materials. As for the material properties, data are extracted from openly available sources and interpolated in the multiphysics solver.

The center column corresponds to the coupled fine-mesh solver for the neutronics and the thermal-hydraulics. As indicated, the two modules exchange all coupled data, and the procedure for the data mapping is briefly covered in Section 2.3.2. The results of the computations are stored on the OpenFOAM internal format and post-processed by a combination of ParaView [84] and Python utilities.

Although invisible in all included publications and in the current description, the setup of a coupled case with all initial fields, meshes, thermophysical data, cross-section sets and decomposition schemes is typically time-consuming and repetitive. To simplify the process, Python utilities were developed for each of the applications presented in the appended papers, and the value of such utilities for work like this should not be underestimated.



**Figure 2.4:** Overview of the computational framework for the multiphysics simulations.

### 2.3.1 Grid generation

The generation of computational grids, is a key to the multiphysics simulations, not least due to the fact that the problem benefits from the use of multiple meshes, with one or more for each field.

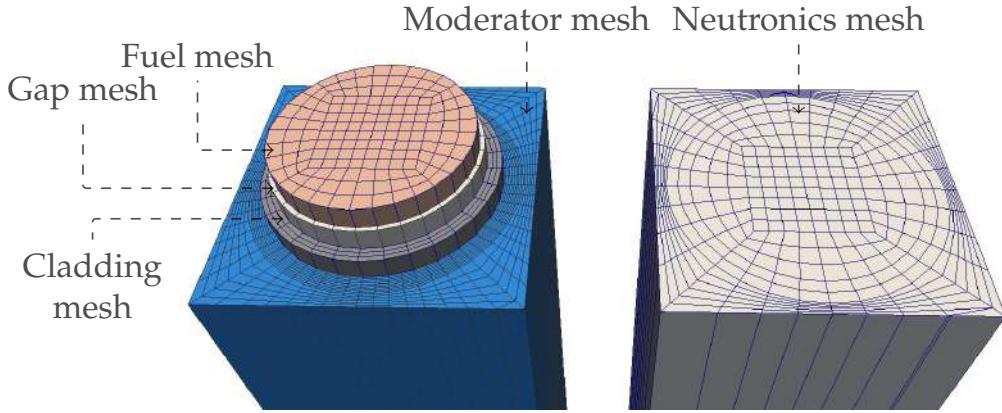
#### Mesh influence on the results

As regards the influence of the grid quality on the results, some of the key sources which contribute to mesh-induced errors are [70, 85]:

- *Insufficient mesh resolution* – If the mesh is too coarse the physics might not be correctly represented. In particular, large gradients will be poorly represented and the solution might not be only imprecise but also inaccurate.
- *Non-orthogonality error* – In the discretization of the diffusion operator, the gradient at the surface between two neighboring cell centers must be computed. In the case of non-parallel normal of the surface and the vector connecting the cell centers a correction of gradient at the surface is required. Such a correction is often computed explicitly [65], but might lead to instabilities.
- *Skewness error* – The term skewness is applied when the vector binding neighboring cells does not cross the common surface in the center. In particular, the order of interpolation from the nodes to the face centers decreases from second order to first order.
- *Non-uniformity* – The mesh uniformity affects the discretization, namely via the order of accuracy of the computation of face gradient.

In addition, it should be mentioned that in the case of a preferred direction of the flow (as the case with the dominating axial flow velocity component in the reactor) it is preferable to have the flow crossing the cells parallel to the normal of the crossed surface. In practical terms we thus desire prismatic or hexahedral elements, axially aligned with the flow.

As the gradients in the field govern what regions require a finer discretization, a non-uniform grid is generally beneficial. The resolution requirements would either stem from some a priori knowledge of the solution or on posteriori estimated errors used to automatically update the mesh accordingly (so called adaptive refinement). Whereas we can directly apply the former from a general knowledge of the reactor core, considering phenomena such as a thermal neutron group peak in the top and bottom reflectors, the latter is, although tractable, practically cumbersome to achieve. In particular, for a refinement by splitting cells to smaller cells (h-refinement) we want to work on a fully unstructured grid [86], which has other drawbacks for the multiphysics framework as discussed in the next section.



**Figure 2.5:** Example domain discretization for a single fuel pin with surrounding subchannel. Exploded for the thermal-hydraulics (left) and the neutronics (right) [Paper IV].

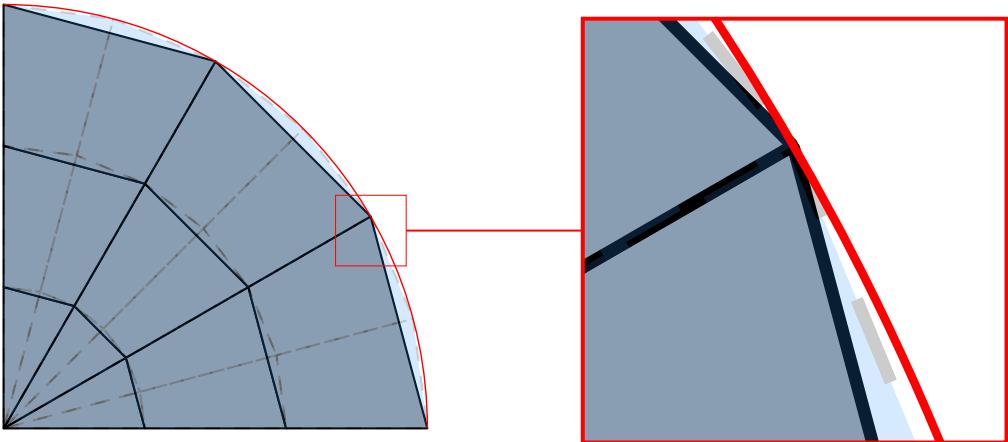
### Multiple meshes

To solve both neutronics and thermal-hydraulics in an efficient manner, multiple overlapping meshes are required. The neutronic grid must cover the entire domain, whereas the fuel, gap and cladding meshes discretize the separate regions only. In addition, the fluid problem is solved on a mesh covering the domain outside the solids. An example of the required regions and meshes is given Figure 2.5, where the separate domains are highlighted.

Depending on the algorithms that couple the different fields of physics, different conditions are imposed on the mesh characteristics. A good overview of some mesh generation issues specific to the reactor geometry is given by Hansen and Owen [87], where among others the following key aspects are notable:

- *Mass preservation* – A mesh with flat surfaces induces an error in the representation of the underlying body. Figure 2.6 gives an example where a fuel pellet is discretized with two alternative meshes; the full lines apply a discretization of 3 elements in the azimuthal direction whereas the dashed discretization uses 6 elements. In the magnification of the point at  $\phi = \pi/6$  it is seen that the actual cylinder (red) is not exactly captured by any of the two meshes. Although seemingly small, the impact on the criticality value of the reactor might be important.
- *Multiple mesh consistency* – Not many available meshing tools are built for multiple mesh support, especially not considering the need for overlapping meshes with preserved material regions. Preferably, a mesh generator should handle situations as the one presented in Figure 2.6 in a consistent manner for all the meshes.

In addition, two more aspects were found important during the method development for this thesis:



**Figure 2.6:** Quarter fuel pin discretization with a 9 elements (solid lines) or 18 elements (dashed lines).

- *Domain decomposition* – As all the calculations are performed on HPC resources with MPI, the domain must be decomposable into different regions. To achieve an efficient decomposition of the multiphysics problem, one approach is to decompose all fields of physics using the same spatial partitioning. This ensures that the data transfer between the modules (e.g. neutronics power level to the fuel heat transfer mesh) is done on the same CPU. To achieve this, a strictly controlled meshing process is required, where all the different meshes have cell faces coinciding at the surfaces used for spatial partitioning.
- *Mesh repeatability* – Since the geometry of the reactor consists of square or hexagonal lattices, at least in the case of the commercial designs, a consistent meshing throughout the fuel assembly is needed. That is, for the same pin cell geometry, the same mesh topology should be computed. This is achievable with block structured meshes where one can assure a perfect match at the interface of the pin cells. For an unstructured mesh this is not directly feasible (or at least difficult to achieve given the previous point of material region preservation).

The described key aspects are not very closely related to the archetype case of the single monolithic mesh covering the full domain. Although the mentioned issues are not particularly challenging from a theoretical point of view, the combination of requirements still makes for a practical problem requiring special procedures (in the sense of not directly applicable in the commercial CFD codes). As an example from the literature, Tautges and Jain [88] describe a hierachal procedure for building meshes for hexagonal as well as square lattices with a sufficient resolution for CFD, taking advantage of the repeated structures. Such procedures are a good attempt, at least up until the point where the level of detail is further increased. For the case of explicit modeling of spacers, a fully unstructured mesh is generally unavoidable (see e.g. [40, 89]).

## Development of mesh capabilities

If we summarize the above discussion, the criteria include that the meshes should be:

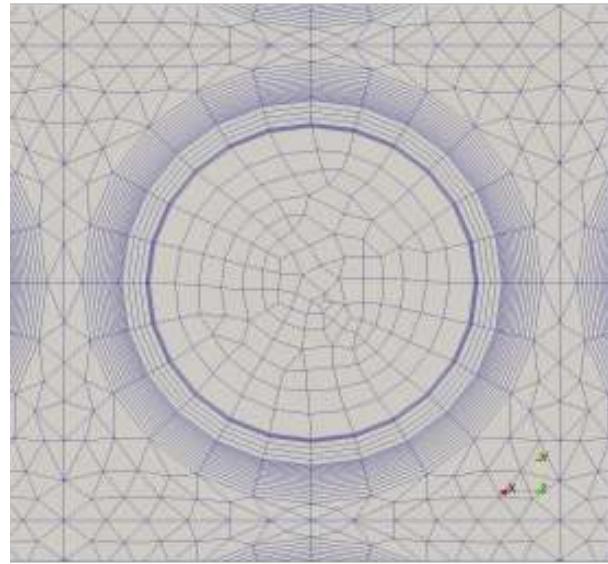
- preferably structured with minimized non-orthogonality and surface normals aligned with the flow direction,
- conformal with the domain decomposition such that no inter-node data transfer is required for the multiphysics couplings,
- easy to control in terms of the resolution, in order to minimize the degrees of freedom,
- repeatable for all pin cells, and
- body-fitted to capture advanced geometries such as spacers.

To fulfill as many criteria as possible, two different solutions were tested. In the first part of the project (essentially [Paper I]) the meshes were generated with the SALOME platform [90]. An example of the meshes produced is given in Figure 2.7. The SALOME platform includes both pre-processing and post-processing together with different capabilities for code coupling and was applied as the framework tool for the NURESIM project [91]. For the current example, only the meshing tools were tested.

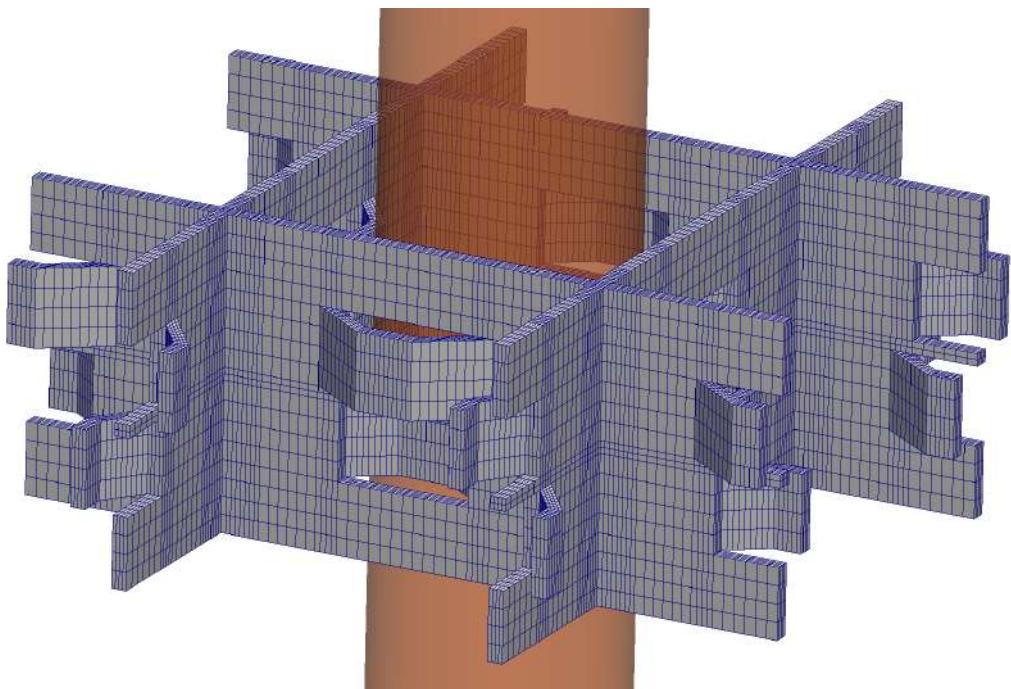
The meshes from [Paper I] are of hybrid character, with structured regions for the boundary layers (i.e. fluid regions close to the walls) and unstructured regions in the bulk of the fluid and the fuel pin. Although the software could be controlled via a Python interface, the possibilities to full control the mesh was found limited (at least in 2012).

To increase control, especially to get full consistency between all mesh regions and in the whole fuel lattice, an alternative meshing code was developed within the project. The code consists of an object-oriented Python framework in which the grids are built from macro objects in a block-structured manner. An example of a computed block-structured mesh of a spacer from the PSBT benchmark [92] is illustrated in Figure 2.8. The macro objects are discretized as blocks with an internal Cartesian discretization in  $(n_x, n_y, n_z)$  cells. The FVM formulation applied does not allow hanging nodes, and thus all intersecting blocks must have a consistent face discretization.

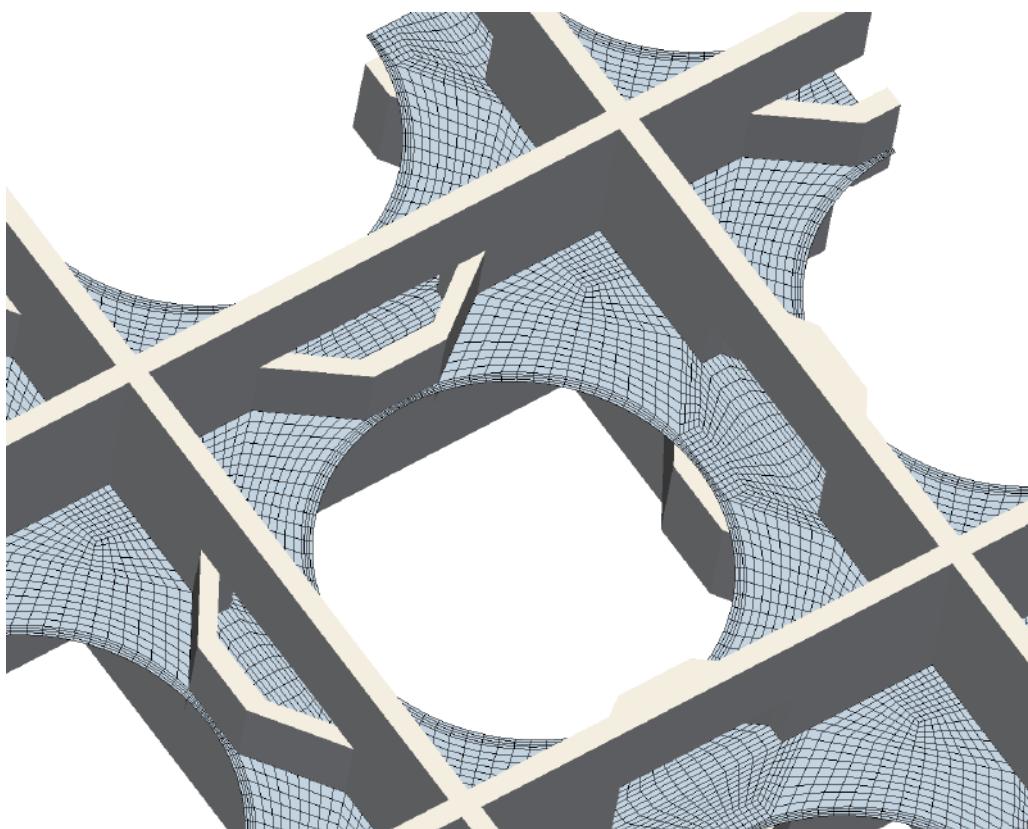
The block-structured method has the benefit of exact control of the discretization of each block and, accordingly, a shared preservation of all geometric regions between different meshes. However, there are drawbacks, and in particular the block-structured mesh will be governed by the regions of the finest mesh resolution which will propagate through the mesh. An example issue is seen in Figure 2.9, where the corners of the moderator mesh are influenced by the structure of the spacer, which gives artificial transitions between different regions in the mesh.



**Figure 2.7:** Example of mesh generated via SALOME (adaptation from [Paper I]).



**Figure 2.8:** Block-structured mesh of a spacer for a pin cell. The solid fuel pin is indicated by the low opacity orange region.



**Figure 2.9:** Example of moderator mesh for a pin cell including a spacer as shown in Figure 2.8.

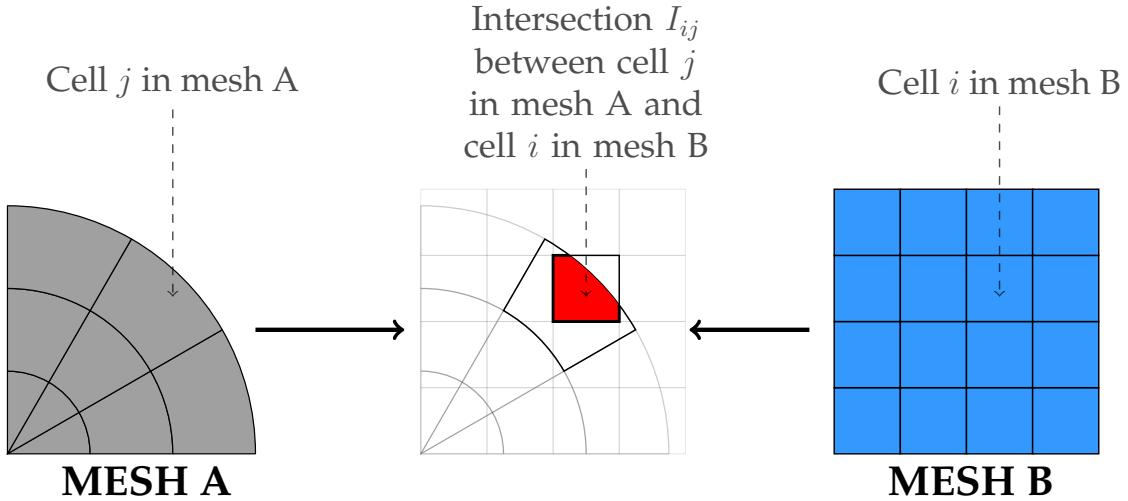
A configuration file ensures rapid control of the discretization and the system geometry (fuel pin radius, number of spacers, fuel pin lattice size, etc.). On the drawback side, it should be noted that the geometrical template structures are internally specified, e.g. the spacer geometry is hard coded from blueprints except for some general parameters such as height and material thickness. This is the price of the block-structured approach; the developer needs to more or less manually build the block structures internally. However, with the developed object oriented library, modifying the geometry or building new geometries are sufficiently fast. Also, this should be considered a programmatic approach to pure hexahedral meshing as can be done through interfaces in some mesh generators (such as e.g. ANSYS® ICEM [93]).

The developed mesh tool additionally creates the required initial and boundary conditions for all fields. Furthermore, the output of the mesh generator contains blocks with corresponding discretization, points and groups of block faces for the boundary conditions. The application generates the exemplified mesh (Figures 2.8 and 2.9) for 3 spacers in a 1 m long system in approximately 10 seconds. The short generation time is a benefit of the block-structured approach where not every single cell needs to be computed but only the blocks building the structures.

It should be noted that the generality of the tool is limited and, as discussed above, a more complicated spacer geometry would require an unstructured mesh. An extension of the tool would be to use unstructured meshes for certain regions of the multiphysics meshes, and specifically ensuring that the outer surfaces of such regions are consistent with neighboring block regions.

### 2.3.2 Mesh mapping

As described above, different meshes are applied for the different fields. To achieve data transfer between meshes of different resolution and structure a mapping algorithm is required. The problem can be solved by a point-to-point interpolation, advantageously accelerated by clever data sorting and decomposing the problem (see e.g. [94]). Alternatively, a direct overlap between the meshes could be computed such that volumetric intersections are used to map the data transfer. A schematic example of the latter approach is displayed for a 2D problem in Figure 2.10. An advantage of intersection approach is that the interpolation is directly conservative, and such a property was the rationale for choosing this approach for the current work. A more detailed discussion on the algorithm and its limitations is given in Paper IV.



**Figure 2.10:** Example of mapping of two overlapping meshes. The intersection  $I_{ij}$  is used to compute interpolation weights for two-way interpolation between cells  $i$  and  $j$ . [Paper IV]

### 2.3.3 Cross-section generation and utilization

Except for meshes and the material thermophysical data, the multiphysics tool requires macroscopic cross-sections for the neutron solvers. The actual application of the cross-sections in the neutronic equations is postponed to Section 3.1, but as the preparation of the cross-sections requires a methodology in itself, a description is given of the criteria for and how to efficiently compute such cross-sections.

#### Prerequisites and potential methodologies

Based on the objectives of sub-pin resolution in the fuel and the fine-mesh multiphysics coupling, the prerequisites are that the cross-sections should be:

- valid at a sub-fuel pin scale,
- discretized in  $G$  energy groups (where  $G \ll 100$ ),
- providing higher order ( $P_N$ ) scattering matrices,
- parametrized on the thermophysical state (temperature, density, etc), and
- useful in combination with the multiphysics setup as described in the previous section.

In particular, the last and the first point turns out to be challenging as the required scales are not the same as in the standard neutronics workflow (as described in Section 1.2.1). Three alternative approaches to generate the cross-sections were identified:

1. Generate macroscopic cross-sections directly from the nuclear data libraries with correct handling of resonances, with a large number of energy groups. This could either be done in a lower dimension spatial calculation or on parts of the actual simulated 3D domain. A detailed overview of the stages associated with lattice computations and cross-section generation is given by Knott and Yamamoto [3] and it is clear that this process would require a major effort in terms of methodology development, and it would later risk to significantly increase the simulation time.
2. Directly apply cross-sections generated by a lattice code. This option was investigated in the early stages of the project (primarily for the commercial codes CASMO-4E [95]), but it was found that the available output was provided on a scale much coarser than required. As described in Section 1.2.1, the lattice codes are aimed at generating fuel assembly cross-sections in few energy groups for the core solvers. As a result, the geometry and scales of the condensation and homogenization processes do not match the criteria for the fine-mesh simulations. It should be noted that there are examples of open source lattice codes (e.g. DRAGON [96]) which could potentially be modified or extended to produce the desired set of cross-sections. However, also for such alternatives there are severe limitations when it comes to non-standard geometries (e.g. a spacer) and again a major effort would be required to extend the code and develop such a model.
3. Compute macroscopic cross-sections with a Monte Carlo approach. This option has the benefit of a very flexible geometry, where most Monte Carlo codes support combinations of a range of primitive mathematical geometries. The obstacle for this alternative is that macroscopic cross-section generation is not historically a standard functionality of such solvers. The long term state-of-the-art code MCNP [97] has been user modified for such generation [98], and the proposed methodology was tested. However, the implementation achieved was found inefficient and inconsistent wherefore the early attempts were abandoned. It should however be noted that the same group performed the reversed operation, i.e. performing calculations in MCNP based on macroscopic cross-section [99], which was successfully applied in noise calculations related to the current thesis [100]. As an alternative to MCNP, the code Serpent [101] was tested and found more suitable since macroscopic cross-section generation is a built-in utility and can be applied for specific regions of the simulated domain (such as a small part of a fuel pin).

Decidedly, all three alternatives have benefits and in particular the first option is tractable from a theoretical point of view. Under transient conditions, where the reactor is potentially far from critical, a general assumption of criticality might give significant impact on the cross-section [102]. With the first option, an on-the-fly approach could potentially compute the cross-sections valid for the actual, and not necessarily critical, state of the reactor. However, as noted above, such an

alternative is associated with a large development effort and a significant computational cost.

From a framework point of view, the third option was judged as the choice easiest to implement and deploy in the multiphysics solver since Serpent includes advanced geometries and cross-section generation and extraction. Nevertheless, the resolution required in this project is still not a default case and the code is not explicitly prepared for generation of many sets of cross-sections on small parts of a fuel assembly or fuel pin. For this reason, and to pre-process the cross-sections for use with the deterministic solver, a wrapper code was developed (described below).

Given the choice of Serpent it is tempting to entirely discard the idea of the deterministic neutron transport solver. Why should we use a discretized, condensed and homogenized approach to the neutron transport when we can readily perform simulations in continuous energy with no discretization errors associated using Monte Carlo? In particular, the availability of detailed geometrical descriptions and accurate solution to the transport problem are appealing. Although the arguments for a Monte Carlo approach are indeed strong for steady simulations, there are still some aspects that support the choice of a deterministic approach:

- *Transient simulations* – Whereas steady-state coupled reactor problems have been performed with Monte Carlo methods for a number of years, transient algorithms are less developed. Examples of simulations of short transients with Serpent (without delayed neutron handling) exist in the literature [103, 104]. Also methodologies for longer transients, i.e. where delayed neutron handling is essential have been proposed [105] but still seem to be under development and not yet sufficiently mature to be a viable option.
- *Simulation time* – Monte Carlo simulations are associated with a significant cost, in particular when compared to heavily optimized core simulators. In the current case of fine meshes the difference could be expected to be smaller but nevertheless significant.
- *Resolution of coupling* – In order to perform a direct fine-mesh coupling between the CFD and the Monte Carlo solvers, a fine geometrical resolution of the tallies would be required. Consequently, the Monte Carlo simulations must be run for a long time to reach low statistical errors for each small volume coupled to the fluid solver. To get a high precision for very small regions in space would induce a much higher cost than getting precise global parameters.
- *Inconsistent parallelization schemes* – Due to the independence between the simulated neutron histories the Monte Carlo software is most conveniently parallelized by duplicating the system, run on separate processors/threads and then recombining the data in a statistically consistent manner. In contrast, the large scale CFD solvers parallelize by domain decomposition. A tightly coupled parallelization schemes, such as presented in Figure 2.3,

would thus not be readily achievable. The monolithic code approach is particularly important for transient solutions with a lot of data exchanges.

Whereas the latter three points could be overcome by (extended) simulation times, the first point is decisive for the current project. In the long run, transient simulations are of particular interest and to get a relevant coupling to the CFD solver and to the scenario length of the transients, requires handling of the delayed neutrons.

#### Cross-section generation using Serpent

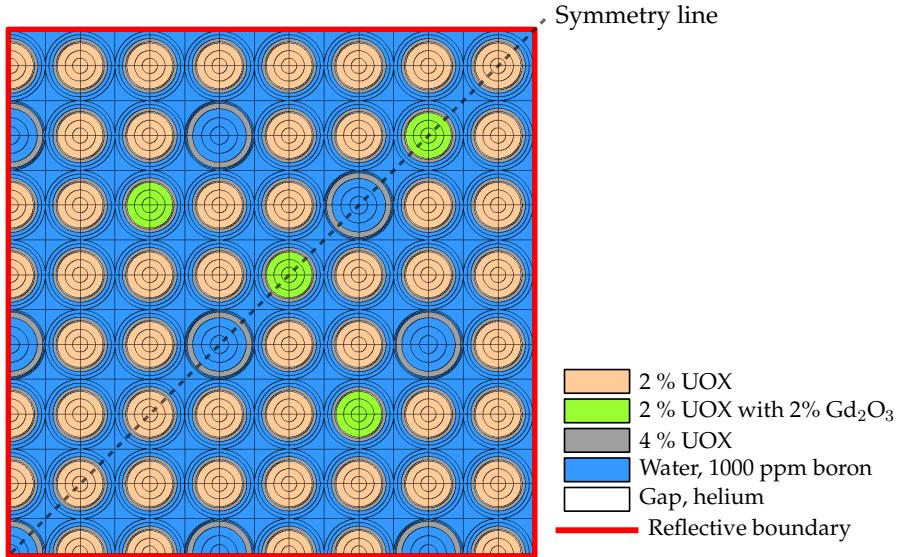
To generate cross-sections for the fine-mesh resolution in Serpent a wrapper application was developed. Similar to the mesh application described above, the tool is developed to automatically create a fuel assembly geometry, but here for the purpose of detailed cross-section generation. The application has the major benefit that the, quite error prone, input format of the Monte Carlo code is hidden. Instead a configuration file with much fewer option and only the necessary geometry details is sufficient to setup a case. The utility requires the following information:

- the lattice pitch together with the number of fuel pins, and the respective type of pin at each position,
- the desired resolution of the cross-section set in terms of radial and azimuthal regions in the fuel, gap, cladding and moderator as exemplified in Figure 2.11,
- the desired number of energy groups,
- the material properties, including densities and isotopic compositions, and
- the radial temperature profile in the fuel as well as in the moderator.

For the moderator, the application computes the relevant density properties and applies this to the material of the particular region in space associated with the specified temperature. It should be noted that Serpent handles the Doppler broadening associated with a certain material temperature, but the application still need to determine and supply Serpent with the correct cross-section table.

In principle three dimensional sets of cross-sections could be computed with both a horizontal and axial resolution. However, as discussed above, finer resolutions require longer simulation times to get the desired statistical confidence intervals. Furthermore, as the cross-section generation is not at the core of the project, a reasonable assumption is to use axially independent cross-sections, i.e. relying on a horizontal slice with reflective boundary conditions in the axial direction.

Given the above specified information, an input file is created for Serpent and then run with the same tool. After the simulation is finished, the same application



**Figure 2.11:** Fuel pin discretization in horizontal plane, using 4 azimuthal and 8 radial regions per pin cell, in total 1775 regions [Paper IV].

computes temperature dependent cross-section files for the multiphysics and geometrical descriptions of each of the regions (radial and azimuthal information) in an OpenFOAM® specific format. The latter is necessary to compute sets of cells from the computational meshes for which each group of cross-sections should be applied.

The geometry in Figure 2.11 shows the actual cross-section regions, in total 1775 regions for the multiphysics computation in [Paper IV]. The Serpent input file for the presented case is more than 8000 lines long, which emphasizes the need for the automatic procedure. The input for the wrapper is 100 lines long.

Implicitly, the described process includes a number of assumptions whereof the most notable are that:

- the cross-section generation is performed under the assumption of periodic boundary conditions, i.e. a system with an infinite number of identical fuel assemblies, and
- the cross-sections are computed for static temperature profiles, a priori determined but potentially updated by an iterative process between the Serpent wrapper and the multiphysics simulations.

Similar to the meshing utility, the tool is implemented as an object-oriented library in Python. The object-orientation has many practical features when it comes to the geometry modeling. The simulation time of the script itself is negligible in comparison to the Monte Carlo simulation.

---

# CHAPTER 3

---

## Models for the coupled LWR single-phase problem

After the introduction of the multiphysics problem with current standard and high-fidelity approaches in Chapter 1 and the overview of the computational methods in Chapter 2, this chapter introduces the neutronic and thermal-hydraulic models. Except for describing the transport equations for each of the modules a brief discussion is given on the options available to solve the non-linear multi-physics problem.

On a side note, the step going from the mathematical formulation in this chapter to the implementation in the FVM framework described in the previous chapter is seemingly large. To provide enough detail to directly repeat the numerical experiments is unsatisfactorily difficult (and page consuming). Such a deep rift between reported equations and actual implementation is commonly seen in the open literature, and is unfortunate. Although the next chapters (4 and 5) introduce some additional details, the current chapter should be better seen as an overview description of the neutronics and thermal-hydraulics.

### 3.1 Formulation of the neutronic problem

The transport of neutrons in the reactor core is governed by the linear Boltzmann equation, here written in an integro-differential form, such that [106]

$$\begin{aligned} \frac{\partial}{\partial t} n(\mathbf{r}, \Omega, E, t) + \Omega \cdot \nabla \Psi(\mathbf{r}, \Omega, E, t) + \Sigma_T(\mathbf{r}, E, t) \Psi(\mathbf{r}, \Omega, E, t) = \\ \int_{(4\pi)} \int_0^\infty \Sigma_s(\mathbf{r}, \Omega' \rightarrow \Omega, E' \rightarrow E, t) \Psi(\mathbf{r}, \Omega', E', t) d\Omega' dE' + \end{aligned}$$

$$\frac{\chi(E)}{4\pi} \int_0^\infty \nu \Sigma_f(\mathbf{r}, E', t) \Phi(\mathbf{r}, E', t) dE', \quad (3.1)$$

where the neutron density  $(n(\mathbf{r}, \Omega, E, t))$  is related to the angular neutron flux as

$$\Psi(\mathbf{r}, \Omega, E, t) = v(E) n(\mathbf{r}, \Omega, E, t), \quad (3.2)$$

which in turn is related to the scalar flux  $\Phi(\mathbf{r}, E, t)$  as

$$\Phi(\mathbf{r}, E, t) = \int_{(4\pi)} \Psi(\mathbf{r}, \Omega', E, t) d\Omega'. \quad (3.3)$$

The balance equation eq. (3.1) is not analytically solvable except for simplified problems and in general we need to rely on numerical methods to resolve the neutron distribution in the core.

In order to solve eq. (3.1) in the FVM framework as described above, we discretize the solution space consisting of time ( $t$ ), space ( $\mathbf{r}$ ), angle ( $\Omega$ ) and energy ( $E$ ). The space discretization was already briefly discussed in Section (2.2.1), and instead, the balance equation is first discretized in terms of neutron energy such a set of  $G$  energy intervals is defined as

$$[E_{\min}, E_{\max}] = \prod_{g=1}^G [E_g, E_{g-1}], \quad (3.4)$$

with the purpose of writing eq. (3.1) as a set of  $G$  discrete equations, coupled in the discrete energy space. Such a set of relations are computed by integrating the balance equation over each energy interval  $g$  with  $1 < g \leq G$ . As an example the scalar neutron flux  $\Phi(\mathbf{r}, E, t)$  is discretized as

$$\Phi_g(\mathbf{r}, t) = \int_{E_g}^{E_{g-1}} \Phi(\mathbf{r}, E, t) dE. \quad (3.5)$$

For the cross-sections and the other energy dependent parameters of eq. (3.1), flux averaged quantities need to be computed, where e.g. the energy discretized total cross-section is given by

$$\Sigma_{T,g}(\mathbf{r}) = \frac{\int_{E_g}^{E_{g-1}} \Sigma_T(\mathbf{r}, E) \Phi(\mathbf{r}, E) dE}{\int_{E_g}^{E_{g-1}} \Phi(\mathbf{r}, E) dE}. \quad (3.6)$$

The weighting procedure is a key aspect of the cross-section generation and due to the complex energy dependence (see the example of the fission cross-section in Figure 1.1), such weighting must be carefully and consistently performed. In the current work this is implicitly achieved by the internal flux weighting in Serpent. After discretizing all terms of eq. (3.1), the balance equation for energy group  $g$  is given by

$$\begin{aligned} \frac{1}{v_g} \frac{\partial}{\partial t} \Psi_g(\Omega) + \Omega \cdot \nabla \Psi_g(\Omega) + \Sigma_{T,g} \Psi_g(\Omega) = \\ \int_{(4\pi)} \sum_{g'=1}^G \Sigma_{s,g' \rightarrow g}(\Omega' \rightarrow \Omega) \Psi_{g'}(\Omega') d\Omega + \frac{\chi_g}{4\pi} \sum_{g'=1}^G \nu_{g'} \Sigma_{f,g'} \Phi_{g'}, \end{aligned} \quad (3.7)$$

where the space and time dependencies are left out for the sake of brevity and the neutron density  $n(\mathbf{r}, \Omega, E, t)$  is written in terms of the angular flux according to eq. (3.2).

For the time dependence, the thesis includes both steady-state and time-dependent neutronics solvers. For the steady-state solver the time dependence in eq. (3.8) is discarded and a normalization factor  $1/k_{\text{eff}}$  is multiplying the fission source of the balance equation such that

$$\begin{aligned} \Omega \cdot \nabla \Psi_g(\Omega) + \Sigma_{T,g} \Psi_g(\Omega) = \\ \int_{(4\pi)} \sum_{g'=1}^G \Sigma_{s,g' \rightarrow g}(\Omega' \rightarrow \Omega) \Psi_{g'}(\Omega') d\Omega + \frac{\chi_g}{4\pi k_{\text{eff}}} \sum_{g'=1}^G \nu_{g'} \Sigma_{f,g'} \Phi_{g'}, \quad g = 1, \dots, G. \end{aligned} \quad (3.8)$$

The equation now takes the form of an eigenvalue problem where the eigenvectors are the (angular) neutron fluxes and the smallest eigenvalue corresponds to the inverse of the criticality factor ( $k_{\text{eff}}$ ).

For the transient approach the time derivative is retained and in addition the fission source of neutrons is split in two parts, the contributions from the prompt neutrons and the delayed neutrons. The prompt neutrons are released immediately after the fission event. In contrast, the delayed neutrons are released after decay of the fission products with a varying time delay. In order to account for the accumulation of such precursors of delayed neutrons and the contribution to the balance equation, additional conservation equations for the precursors are added. Due to the large number of different fission products decaying, the precursors are routinely grouped in  $I$  groups, with one conservation equation per group. The balance equation for the transient problem is then given by

$$\begin{aligned} \frac{1}{v_g} \frac{\partial \Phi_g(\mathbf{r}, t)}{\partial t} \nabla \cdot \mathbf{J}_g(\mathbf{r}, t) = -\Sigma_{T,g}(\mathbf{r}, t) \Phi_g(\mathbf{r}, t) + \sum_{g'=1}^G \Sigma_{s0,g' \rightarrow g}(\mathbf{r}, t) \Phi_{g'}(\mathbf{r}, t) \\ + (1 - \beta) \chi_g^p \sum_{g'=1}^G \nu_{g'}(\mathbf{r}, t) \Sigma_{f,g'}(\mathbf{r}, t) \Phi_{g'}(\mathbf{r}, t) + \chi_g^d \sum_{i=1}^I \lambda_i C_i(\mathbf{r}, t), \quad g = 1, \dots, G, \end{aligned} \quad (3.9)$$

with the precursor concentrations ( $C_i$ ) calculated as

$$\frac{dC_i(\mathbf{r}, t)}{dt} = \beta_i \sum_{g'=1}^G \nu_{g'}(\mathbf{r}, t) \Sigma_{f,g'}(\mathbf{r}, t) \Phi_{g'}(\mathbf{r}, t) - \lambda_i C_i(\mathbf{r}, t), \quad i = 1, \dots, I. \quad (3.10)$$

As seen from eqs. (3.9) and (3.10) the equations are interdependent and must be solved together, either by explicit iteration or in a more implicit manner.

For the angular discretization, two different alternatives will be considered in this thesis, namely the diffusion approximation, in which the angular dependences are removed, and the discrete ordinates method.

### 3.1.1 Diffusion approximation

The diffusion approximation for the angular dependent neutron transport equation is achieved in three steps. First, the neutron transport equation is integrated over all angular space such that eq. (3.8) gives

$$\nabla \cdot \mathbf{J}_g + \Sigma_{T,g} \Phi_g = \int_{(4\pi)} \int_{(4\pi)} \sum_{g=1}^G \Sigma_{s,g}(\Omega' \rightarrow \Omega) \Psi_g(\Omega') d\Omega d\Omega' + \frac{\chi_g}{k_{\text{eff}}} \sum_{g'=1}^G \nu_{g'} \Sigma_{f,g'} \Phi_{g'}. \quad (3.11)$$

Secondly, the scattering kernel ( $\Sigma_{s,g}(\Omega' \rightarrow \Omega)$ ) is assumed to be isotropic, i.e.

$$\Sigma_s(\mathbf{r}, \Omega' \rightarrow \Omega, E' \rightarrow E) = \frac{\Sigma_{s0}(\mathbf{r}, E' \rightarrow E)}{4\pi}. \quad (3.12)$$

Finally, we apply Fick's law to approximate the current as [107]

$$\mathbf{J}(\mathbf{r}, E) \approx -D(\mathbf{r}, E) \nabla \Phi(\mathbf{r}, E). \quad (3.13)$$

Inserting eqs. (3.12) and (3.13) in eq. (3.11) then gives the final form of the steady-state diffusion equation as

$$-\nabla (D_g \nabla \Phi_g) + \Sigma_{T,g} \Phi_g = \sum_{g'=1}^G \Sigma_{s0,g' \rightarrow g} \Phi_{g'} + \frac{\chi_g^p}{k_{\text{eff}}} \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'}. \quad (3.14)$$

Advantageously, the diffusion equation efficiently reduces the solution space for the neutron distribution as only one equation is solved per energy group. However, the approximation of isotropic scattering and the use of Fick's law is known to reduce the accuracy, in particular for heterogeneous regions such as close to the interface between the fuel pins and the moderator. A more rigorous derivation of the diffusion equation using  $P_1$  theory is given in Bell and Glasstone [107].

### 3.1.2 Discrete ordinates method

As an alternative to the diffusion approximation of the angular flux, there are many methods to actually resolve the angular flux, or at least some degree of it. Just like space or time, the angular dimension requires discretization. In the spherical harmonics ( $P_N$ ) method the angular flux is expanded in terms of a spherical harmonics base of order  $N$  (with couplings to higher order terms neglected), which results in a set of coupled equations for the flux expansions (see e.g. [68, 106]). In the discrete ordinates method ( $S_N$ ) the angular space is instead covered with a set of discrete directions. The neutron transport equation is solved for each such direction. The latter method ( $S_N$ ) is the method applied in this thesis.

In more specific terms, for the steady state discrete energy problem, eq. (3.8) is written for one specific ordinate (i.e. streaming direction)  $\Omega_m$  [108]

$$\Omega_m \cdot \nabla \Psi_{m,g} + \Sigma_{T,g} \Psi_{m,g} = S_{m,g} + \frac{1}{k} F_{m,g}, \quad (3.15)$$

where the fission source ( $F_{m,g}$ ) is defined as

$$F_{m,g} \equiv \chi_g \sum_{m'}^M w_{m'} \sum_{g'=1}^G \nu_{g'} \Sigma_{f,g'} \Psi_{m',g'}, \quad (3.16)$$

and the scattering term is expanded in terms of Legendre polynomials ( $P_l$ ) and written for a discrete number of ordinates  $M$  such that

$$S_{m,g} \equiv \sum_{l=0}^L (2l+1) \sum_{m'}^M P_l(\Omega_m \cdot \Omega_{m'}) w_{m'} \sum_{g'=1}^G \Sigma_{s,l,g' \rightarrow g} \Psi_{m',g'}. \quad (3.17)$$

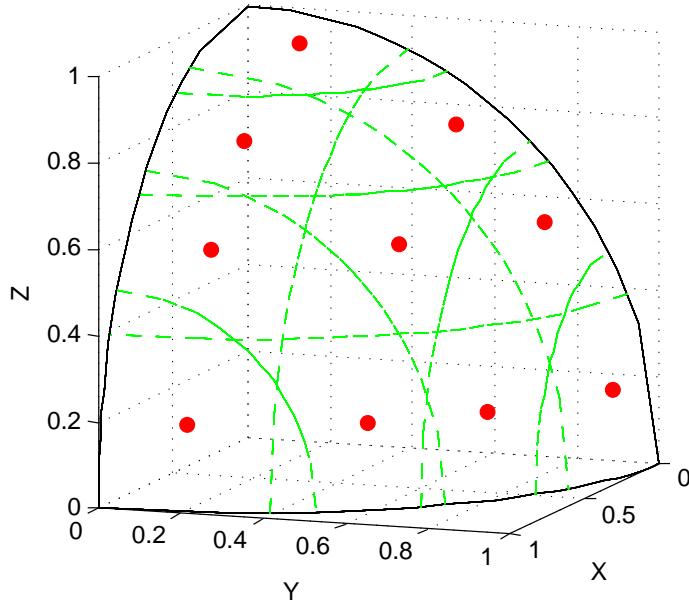
The scalar flux ( $\Phi_g$ ) is computed as a weighted sum of the flux for each direction, such that

$$\Phi_g = 4\pi \sum_m^M w_m \Psi_{m,g}, \quad (3.18)$$

where the weights ( $w_m$ ) are associated with the chosen quadrature set as discussed next.

The set of discrete directions ( $\Omega_m$ ) and weights ( $w_m$ ) is decisive for the accuracy of the solution of the  $M$  coupled transport equations. The optimal set is unfortunately problem specific (see e.g. [110]). However, in general a larger number of discrete directions will recover the angular flux with a better accuracy. In this thesis specifically, the level symmetric quadrature set is applied (for further details see e.g. Hébert [68]). An example of the set of directions for such a set of order 9 is shown for the first octant in Figure 3.1. As seen in the figure, the set of directions is symmetric as regards all Cartesian axes.

Computationally, the most costly part of eq. (3.15) is the evaluation of the scattering term, where each evaluation of  $S_{m,g}$  for a specific  $m$  and  $g$  requires



**Figure 3.1:** Example discretization of an octant of the angular space using level symmetric quadrature of order 8 with  $\mu_1 = 0.20$  according to [109].

$(L \times M \times G)$  summations. In order to reduce the cost of this computation the scattering source is projected on a spherical harmonics basis, specifically the real spherical harmonic functions as defined by Hébert [68], such that eq. (3.17) can be written

$$S_{m,g} = \sum_{g'=1}^G \sum_{l=0}^L (2l+1) \sum_{r=-l}^l R_{lr} \phi_{g,l,r} S_{s,l,g' \rightarrow g}. \quad (3.19)$$

The quadrature set applied in this thesis represents the spherical harmonics orthogonality up to a degree  $L = N/2$  [68] and therefore a smaller number of expansion coefficients  $\phi_{g,l,r}$  than the actual number of discrete ordinates are required.

The  $S_N$  method is well known to converge slowly for many types of problems [111]. Commonly, different accelerating schemes are applied to decrease the number of iterations required to resolve the angular interdependence between the different directions in the scattering source (explicitly in eq. (3.17) or implicitly via the projection on the spherical harmonics in eq. (3.19)). One classical method is the diffusion synthetic acceleration (DSA) [112]. Successful implementations of such accelerated schemes are known to be dependent on a discretization consistent with the original  $S_N$  implementation [113]. Moreover, some classical schemes like the DSA have been shown to work worse in cases of multidimensional cases with strong material heterogeneities (which very well describe the systems of interest in this thesis). Another possible approach, is to solve the problem using a Krylov method, which has been reported to decrease the dependence on the discretization [113]. What regards the neutronics solver developed for this thesis, a Krylov method is by hypothesis a better candidate for the unstructured (or at least non-Cartesian) approach. However, in the current scope, the coupling

between the ordinates was resolved by source iteration.

## 3.2 Single-phase fluid flow and heat transfer

The single-phase flow problem in a LWR is governed by the conservation equations for mass, momentum (the so called Navier-Stokes equations) and enthalpy written as [114]

$$\frac{\partial \rho(\mathbf{r}, t)}{\partial t} + \nabla \cdot (\rho \mathbf{u})(\mathbf{r}, t) = 0, \quad (3.20)$$

$$\frac{\partial(\rho \mathbf{u})}{\partial t}(r, t) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u})(r, t) = \nabla \cdot \tau(r, t) - \nabla p(r, t) + \rho(r, t)g, \quad (3.21)$$

and

$$\frac{\partial(\rho h)}{\partial t}(\mathbf{r}, t) + \nabla \cdot (\rho \mathbf{u} h)(\mathbf{r}, t) = -\nabla \cdot q''(\mathbf{r}, t) + q'''(\mathbf{r}, t) + \bar{\tau}(\mathbf{r}, t) : \nabla \otimes \mathbf{u}(\mathbf{r}, t) + \nabla \cdot (\mathbf{u} p)(\mathbf{r}, t). \quad (3.22)$$

The interpretation of the heat source terms ( $q''$  and  $q'''$ ) is discussed in detail in Section 3.2.3. The complexity of the equations is manifold. First, eqs. (3.20)–(3.22) are all interdependent through the fluid velocity ( $\mathbf{u}$ ) and (except for the continuity equation) through the pressure ( $p$ ). Secondly, the equations are non-linear in the solution variables  $\mathbf{u}$ ,  $h$  and  $p$  (or alternatively  $\rho$ ). Thirdly, all material data in eqs. (3.20)–(3.22) vary with the thermophysical state of the system, which in turn will change due to the enthalpy rise in the heated channels in the core. For all three mentioned reasons, an iterative (or non-linear) algorithm is required to resolve the dependencies, and this is further discussed in Section 3.2.2 below.

The single-phase problem with low Mach numbers can often be solved as an incompressible flow, for which the criteria on the density is that

$$\frac{1}{\rho} \frac{D\rho}{Dt} = 0. \quad (3.23)$$

For the single-phase reactor core problem, the characteristic velocity is indeed significantly lower than the speed of sound in the coolant ( $Ma < 0.01$ ). However, due to the heating of the water the density change will violate the condition in eq. (3.23). Considering the thermodynamic equation of state we can write [114]

$$\frac{1}{\rho} \frac{D\rho}{Dt} = \alpha \frac{Dp}{Dt} + \beta \frac{DT}{Dt}, \quad (3.24)$$

where  $\alpha$  is the isothermal compressibility and  $\beta$  is the bulk thermal expansion coefficient. Although  $\alpha$  is small in the case of liquid water, the flow should still be considered as low speed compressible due to the changes caused by the temperature change.

For the incompressible flow, the coupled eqs. (3.20)–(3.21) are typically solved in terms of pressure and velocity, whereas a compressible solver is often posed

in terms of velocity and density (see e.g. [65, 115]). In the current case, a so-called weakly compressible approach is needed, with special attention given to the dependence on temperature for all transport coefficients.

In addition to the fluid conservation equations, an energy equation for the fuel pins must be formulated. Such a balance equation is coupled to the energy equation of the water at the interface of the fuel pin and accordingly results in a conjugate heat transfer problem between the fuel pins and the fluid. The treatment of the heat transfer is discussed in detail in Section 3.2.3.

### 3.2.1 Turbulence

The conservation equations (3.20)–(3.22) describe local and instantantaneous flow and due to the large range of time and length scales involved a direct solution to industrial scale problems is still not feasible. Indeed, solutions resolving all the characteristic scales of the flow, i.e. to the smallest dissipation scales of the turbulence, can be computed with Direct Numerical Simulation (DNS) (see e.g. [116]). However, due to the enormous computational requirements only limited domains with relatively low Reynolds numbers can be simulated even with the current state-of-art HPC resources.

Instead of the direct approach, we need to approach the equations with a filtered or averaged approach. From a computational perspective, two methodologies, or rather classes of methodologies, are commonly found in CFD, namely Reynolds-Averaged Navier Stokes (RANS) and Large Eddy Simulations (LES), which are both popular and widely applied in nuclear engineering [117]. In the current thesis the prior has been used as a first approach. Later experimentation with LES was performed and due to the generally increased resolution, the cost of such simulations are significantly larger than the RANS counterpart. Nevertheless, LES is a future candidate for the coupled transient calculations within the multiphysics tool.

#### RANS

In RANS, a filtering of the conservation equations is performed via a decomposition of each field in its mean value and a temporally fluctuating component. The details of the procedure is well covered in the literature (see e.g. [65, 116, 118, 119]), and here we will be satisfied with the final form of the mass, momentum and enthalpy equations after the insertion of the decomposed velocity, pressure, and enthalpies and time averaging the equations given by

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{U}) = 0, \quad (3.25)$$

$$\frac{\partial(\rho \mathbf{U})}{\partial t} + \nabla \cdot (\rho \mathbf{U} \otimes \mathbf{U}) = \nabla \cdot \tau - \nabla \cdot \overline{\rho \mathbf{u}' \otimes \mathbf{u}'} - \nabla P + \rho g, \quad (3.26)$$

and

$$\frac{\partial(\rho H)}{\partial t} + \nabla \cdot (\rho \mathbf{U} H) = -\nabla \cdot \overline{q''} + \overline{q'''} + \nabla \cdot (\mathbf{U} P) + \overline{\nabla \cdot (\mathbf{u}' p')} + \tau \nabla \otimes \mathbf{U} - \overline{\nabla \cdot (\rho \mathbf{u}' h')}, \quad (3.27)$$

where space and time dependence  $(\mathbf{r}, t)$  is omitted for brevity, capital letters are used for mean quantities  $(\mathbf{U}, H, P)$  and primes indicate the fluctuating components  $(\mathbf{u}', h', p')$ .

To close eqs. (3.25)–(3.27), the terms with fluctuating components must be modeled. As regards the momentum equations, the Reynolds stress tensor  $\mathbf{u}' \otimes \mathbf{u}'$ , could be solved for by using six additional (all permutations of  $u'_i u'_j$ , assuming symmetry) equations which then model the Reynolds Stresses. Alternatively, and computationally cheaper, the Boussinesq assumption could be applied such that in a tensor notation

$$\overline{\rho u'_i u'_j} = \mu_t \left( U_{i,j} + U_{j,i} - \frac{2}{3} U_{k,k} \delta_{i,j} \right) - \frac{2}{3} \rho k \delta_{i,j}, \quad (3.28)$$

where the model is posed in terms of a turbulent kinetic viscosity  $\mu_t$  and the turbulent kinetic energy  $k$ . Instead of solving for the Reynolds stresses, an equation for  $k$  is solved, and in turn additional equations could be added to model some source term in the  $k$ -equation. The open literature contains a plethora of different models to close the equation for turbulent kinetic energy with different supposed strengths and weaknesses. In this thesis the standard  $k - \epsilon$  model is used, i.e. a two-equation model to compute the turbulent kinetic energy and the turbulent dissipation combined into a model for the turbulent kinetic viscosity.

In addition to the closure for the Reynolds stresses, models are required for terms with primed quantities in the enthalpy equation. As regards the last term of eq. (3.27), this can be modeled as an additional contribution to the thermal conductivity in the spirit of the Boussinesq assumption (covered in Section 3.2.3) whereas the term  $\overline{\nabla \cdot (\mathbf{u}' p')}$  is assumed negligible.

It should be noted that the equations above are in principle the Unsteady RANS equations (URANS) as the time derivatives are kept. For the steady-state version of the model (RANS) the time-derivatives are taken out, and this is the model applied in Papers I, II and IV. The interpretation of URANS is not without controversy (see e.g. [117]) and the fluctuations in the results computed with such a model are likely doubtful at the best.

### 3.2.2 Pressure and velocity algorithms

As noted above, eqs. (3.20)-(3.22) are coupled via pressure and velocity and an iterative or non-linear approach is required to resolve the dependencies. The most classical approaches in CFD is to solve the problem linearized and one component and equation at a time, i.e. using a so-called segregated approach. The SIMPLE algorithm [81] has been the de facto standard for steady-state pressure-based

incompressible solvers since the 1970s. An important element of the method is the formulation of a pressure equation given the continuity and Navier-Stokes equations. Interestingly, the mass (or continuity) equation lacks an explicit dependence on pressure. In a mathematical formalism, eqs. (3.20) and (3.21) written in a matrix format give a saddle-point problem, essentially lacking the diagonal term in the system matrix (see e.g. [120]). The solution proposed in SIMPLE could formally be seen as a two stage procedure where the velocity is approximated from the previous pressure solution, followed by the solution of the pressure field computed via the Schur complement of the system matrix.

In a similar manner as the SIMPLE algorithm, the unsteady PISO algorithm [82] relies on an iterative approach to resolve the pressure and velocity dependencies (for further details on the formulation and the implementation see e.g. [80]). The SIMPLE algorithm was applied in all steady-state simulations in this thesis (Papers I, II and IV) and PISO for the transient simulations (Papers V and VIII).

### 3.2.3 Heat transfer problem

The LWR core requires the conjugate heat transfer problem to be solved as the heat generated in the fuel pins is extracted via the fluid coolant. The enthalpy balance in the fluid was already described by eq. (3.22). However in order to directly solve both the fluid and the solid regions in a monolithic, implicit system it is convenient to use the same solution variable for energy throughout the domain. As the solid region is described by the temperature conduction equation

$$\rho(\mathbf{r}, T, t)c_p(\mathbf{r}, T, t)\frac{\partial T(\mathbf{r}, t)}{\partial t} = \nabla \cdot (K(T)\nabla T(\mathbf{r}, t)) + q''(\mathbf{r}, t), \quad (3.29)$$

the fluid enthalpy equation is re-written in terms of temperature [109]. The resulting fluid equation is written as

$$\frac{\partial(\rho c_p T)}{\partial t} + \rho c_p \vec{U} \cdot \nabla T = \beta \vec{U} \cdot \nabla P + \nabla \cdot (K \nabla T) + q''', \quad (3.30)$$

where  $\beta$  is the thermal expansion coefficient of the fluid and where Fourier's law of conduction has been applied to model the surface heat flux in terms of a diffusion term. The thermal conductivity parameter ( $K$ ) additionally includes a contribution from the turbulent diffusivity.

Although the volumetric source term  $q'''$  is kept in eq. (3.30), no actual sources in the liquid are introduced. In principle the term could have introduced some source terms due to capture of gamma rays [3], released in the fission events. However, no transport of the gamma is covered in this thesis. In the temperature equation for the fuel, the volumetric source term is computed from the recoverable energy of the fissions and is thus the explicit coupling from the neutronics to the thermal-hydraulics.

In contrast to the pressure-velocity coupling described in Section 3.2.2, the temperature equations require no special procedures in the discretization or equation reformulation. Instead a potential issue arises from the many solid regions which are separately connected to the fluid region. To solve the entire heat transfer problem all fuel pins and the fluid region need to be simultaneously converged. For iterative methods, such as the Neumann-Dirichlet method [121], or in a more general form non-overlapping Schwarz decomposition methods [122], the many material regions risk to give a very slow iterative problem. As an alternative, all regions could be formulated in a combined system of equations and consequently solved in a concurrent and fully implicit manner. The latter approach is applied in this thesis and in practice realized by coupling the temperature at the boundaries implicitly via a harmonic interpolation that preserves the conservation over the boundary faces between the solid and fluid regions.

In theory, the heat transfer by thermal radiation is required, at least for the gap and gas inside it which is not opaque to the radiation. As a first approximation, a frequency independent radiative heat transfer equation could be applied (see e.g. [123]). The black-body source term for the radiation would then be computed according to the Stefan-Boltzmann law such that,

$$E_b(\mathbf{r}, T) = \sigma_{SB} n^2 T^4(\mathbf{r}), \quad (3.31)$$

where  $\sigma_{SB}$  and  $n$  correspond to the Stefan-Boltzmann constant and the refractive index of the medium, respectively. Due to the large exponent on the temperature, the thermal heat radiation is either not significant (low temperatures) or completely dominating the heat transfer (high temperatures). A thermal radiation model for the gap was included in Paper I. The effect of radiation was studied in more detail in a previous work [124], where it was found that, for nominal PWR conditions, the heat transfer due to radiation in the gap was insignificant. For Paper IV the heat transfer in the gap was therefore modeled similarly to the solid regions, i.e. as dominated by conduction.

### 3.3 Multiphysics formulation and algorithms

After the formulation of the separate neutronic and thermal-hydraulic models, the attention is turned to the multiphysics and the algorithms to solve the combined problem of neutron transport, fluid continuity and momentum conservation and the conjugate heat transfer between the fluid and the solid fuel pins.

As touched upon in the introduction (Section 1.3.2), the first choice altogether is whether to solve the coupling in a single code or by application of multiple tools. Some of the respective benefits and drawbacks of the approaches were mentioned in Section 2.1.3, and in particular the potential excessive computational cost of parallelizing the segregated approach was discussed. As outlined in Section 2.2, this thesis is based on a single code approach.

To understand the complete multiphysics algorithm, it should be emphasized that there are many layers to the problem that need to be addressed, namely:

- *Multiphysics* – The thermal-hydraulic equations rely explicitly on the power density computed from the neutronics via the source term in the fuel temperature eq. (3.29), whereas the neutronics implicitly relies on the thermophysical state via the density and temperature dependence of the cross-sections.
- *Non-linearities* – As discussed in Section 3.2 the conservation equations for the fluid are all coupled and non-linear in the solution variables. Also the steady-state neutronic problem is non-linear due to the criticality factor.
- *Implicit dependencies* – Both the neutronics and the thermal-hydraulics rely on material parameters that are dependent on the thermophysical state and such dependencies are not expressed in algebraic relations, rather computed in a black box manner, i.e. relying on e.g. a database of values.

From the above list it is immediately clear that the system cannot be directly described with a linear equation system (i.e.  $Ax = b$ ) rather it is described in a non-linear manner such that in a generic notation we need to solve

$$F(x) = 0, \quad (3.32)$$

where  $x$  would be a solution vector of all unknowns ( $\Phi_g/\Psi_{m,g}$ ,  $\mathbf{U}$ ,  $T$ ,  $P$  in the steady-state case and with turbulence excluded).

In principle a problem like eq. (3.32) could be solved using a non-linear solution method like Newton's method, by directly computing the inverse of the Jacobian of the system of equations such that for an iterate  $m$  of the solution variable  $x$  we have

$$J(x_m)(x_{m+1} - x_m) = F(x_m), \quad (3.33)$$

where the solution of the system would give the next iterate of the solution variable  $x_{m+1}$ . To avoid the inverse of the Jacobian, eq. (3.33) can be solved by a standard linear solver. However, to even formulate the Jacobian might be both expensive and difficult. Due to the use of "black-boxes" in terms of the thermophysical parameters and cross-sections, a direct analytical form of the Jacobian is in principle impossible, and for all practical methods based on eq. (3.33) a numerical estimate of  $J$  is instead computed.

In the other end of the scale, a fully explicit approach is to solve the system with Picard iterations, where each separate module of physics is solved for constant values of all other equations. Considering the steady-state problem eq. (3.32) could be formulated in terms of the above fluid conservation equations and neutron transport equation as

$$F \begin{pmatrix} \phi \\ P \\ \mathbf{U} \\ T \end{pmatrix} = \begin{pmatrix} F_\phi(\phi, p, T) \\ F_P(\mathbf{U}, T) \\ F_{\mathbf{U}}(P, T) \\ F_T(\mathbf{U}, P, \Phi) \end{pmatrix} = \begin{pmatrix} M(T, p)\phi - \frac{1}{k_{\text{eff}}}F(T, p)\phi \\ H_P(\mathbf{U}, T)P = S_P \\ H_{\mathbf{U}}(P, T)\mathbf{U} = S_{\mathbf{U}} \\ H_T(\mathbf{U}, T)T = S_T(\mathbf{U}, P, \Phi) \end{pmatrix} = 0, \quad (3.34)$$

where the operators  $M$  and  $F$  are determined from the neutron diffusion equation (eq. 3.14) or the  $S_N$  equation (eq. 3.15), and  $H$  and  $S$  are determined from the mass conservation (eq. 3.25, formulated as a pressure eq. via SIMLPE or PISO), the momentum conservation (eq. 3.26) and the temperature equations for the solid (eq. 3.29) and the fluid (eq. 3.30), respectively. It should be noted that the operators  $M$  and  $F$  are also dependent on the criticality factor ( $k_{\text{eff}}$ ), which could in principle be added to the state vector above. Although the exact definition of each of the operators is not important for the reasoning, it should be noted that each such operator in itself corresponds to a non-linear problem, i.e. even when considering all other variables fixed.

A Picard iteration would, with the formulation as in eq. (3.34), be computed by solving each of the four equations with the other parameters frozen. In terms of coding effort such an approach is simple and the respective algorithms for solving the neutronics and thermal-hydraulics could be kept without modification. For example, the iteration between the pressure and velocity as performed in the SIMPLE algorithm could be applied unchanged. The direct use of Picard iterations has been shown to converge slowly for some multiphysics problems (as discussed in [125]). However, the method avoids formulating the actual Jacobian and thus a higher number of iterations could be accepted due to the lower cost per iteration.

Recently, the use of non-linear formulations has drawn a lot of attention in the multiphysics communities, including formulations such as the Jacobian Free Newton-Krylov method (JFNK) (see e.g. [126] and applied in [127]) and Anderson Mixing (originally in [128] and e.g. applied to reactor multiphysics by [129]). The former is known to require sufficiently good preconditioning [127], for example realized by a Picard style inner iteration. The latter could be seen as a direct acceleration of the fixed point iterations and constitute an interesting choice as the modifications required are small.

It should be noted that in principle two layers of non-linear solvers could be applied to solve eq. (3.34). Besides solving the outer, multiphysics iteration in a non-linear fashion, a non-linear approach could be applied for the separate equations, which has been done for neutronics [130, 131]. Also for CFD the implicit approaches have gained interest (see e.g. [132]) although the segregated fixed point algorithms like SIMPLE and PISO still prevail.

In the current thesis, the multiphysics couplings and the respective physics modules are all solved in a Picard iteration manner. The primary reason for this was the simplicity of the method and that the performance of the multiphysics iterations was judged acceptable in the early tests of the solver as discussed further in the results. The use of acceleration by e.g. Anderson mixing is interesting and should be a candidate for further studies on the coupled CFD/neutronics problem. Notably, many of the referenced examples of application of e.g. JFNK are targeted at macroscopic models, on the scale of the system codes, whereas the current thesis is focused on coupling on much smaller scales. The extent to which

this difference would manifest in better or worse acceleration with the mentioned methods is not clear and thus it is open for future investigations.

---

## CHAPTER 4

---

# Steady-state coupled solver application and analysis

In this chapter the outlined computational methodology and the defined physical models are applied together to simulate fine-mesh multiphysics for PWR like conditions. The chapter includes the results from Papers I, II and IV. Paper I reports the first version of the code applied to a system of  $5 \times 5$  fuel pins with a checkerboard pattern of low and high enrichment fuel pins. Whereas Paper I is entirely based on a diffusion solver for the neutronics, Paper II presents an implementation of  $S_N$  and accompanying results that compare the flux profiles from diffusion and discrete ordinates. Paper IV is again focused on the overall behavior of the framework and the code presented corresponds to a re-write as compared to Paper I.

Many of the presented results are to some extent a proof of principle and an attempt to present the feasibility of highly-resolved multiphysics simulations of LWR single-phase systems. As examples of this, the papers include figures for the computational effort and the relative cost of each of the modules in combination with the number of iterations required to solve the steady-state problem. Nevertheless there are results which are of physical interest, such as the results of the fully resolved simulations compared to some averaged and non-resolved simulations.

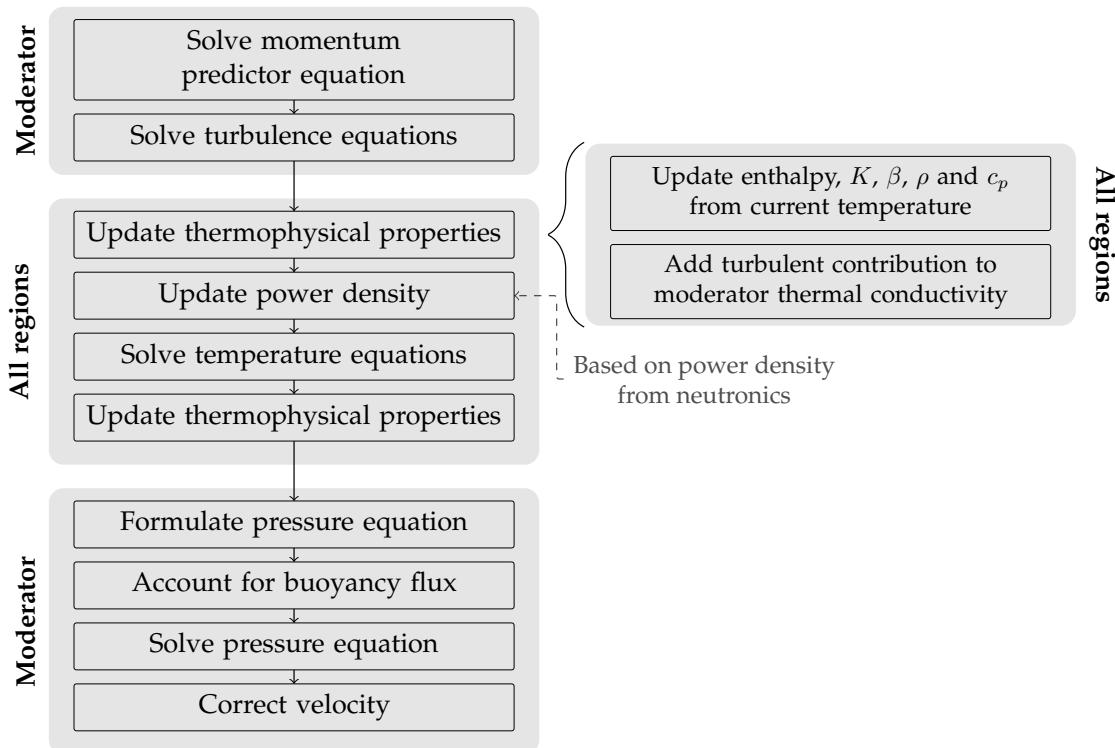
### 4.1 Implementation and framework details

Although many of the general aspects of the implementation of the multiphysics FVM code were described already in Chapter 2, some additional details specific to the steady-state solver are presented here. The framework was re-written between Paper I and Paper II. One of the major differences between the different

implementations was that whereas the first version used the same mesh for all fields of physics, multiple different meshes were supported in the re-written version. The single mesh approach was early judged as a drawback and the version including the mesh-to-mesh transfer capability (see Section 2.3.2), resulted in a more flexible code. In this summary, the description covers only the methodology corresponding to the second implementation.

#### 4.1.1 SIMPLE algorithm and heat transfer

The steady-state single phase solver relies on the briefly described SIMPLE algorithm (see Section 3.2.2). The steps of the algorithm are schematically outlined in Figure 4.1. As seen from the algorithm, the moderator momentum predictor equations and the turbulence are solved prior to the CHT problem, where the latter treated with the described monolithic approach for the temperature equation.



**Figure 4.1:** Thermal-hydraulics solver methodology.

As regards the stability of the scheme, the SIMPLE approach typically requires some under-relaxation and for the iterative algorithm presented in Figure 4.1, the pressure and velocity under-relaxations were typically 0.3 and 0.7, respectively. For the temperature and the turbulence equations less under-relaxation was generally needed.

### 4.1.2 Discrete ordinates solver

The algorithm of the discrete ordinates and eigenvalue solver, applied in Paper II and Paper IV, is outlined in Figure 4.2. The approach follows a standard scheme of an iterative approach to resolve the group and ordinate interdependencies.

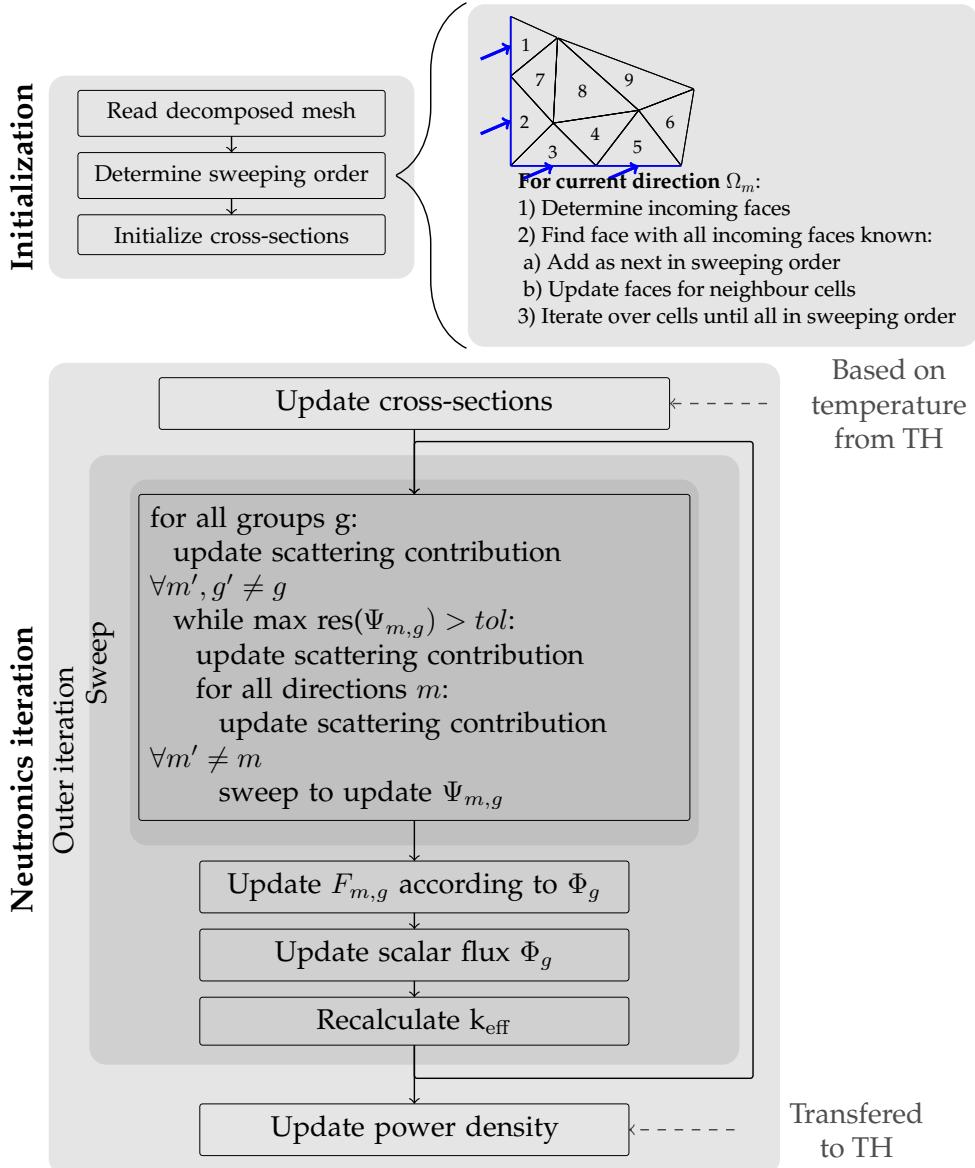


Figure 4.2: Applied algorithm for the discrete ordinates method. [Paper IV]

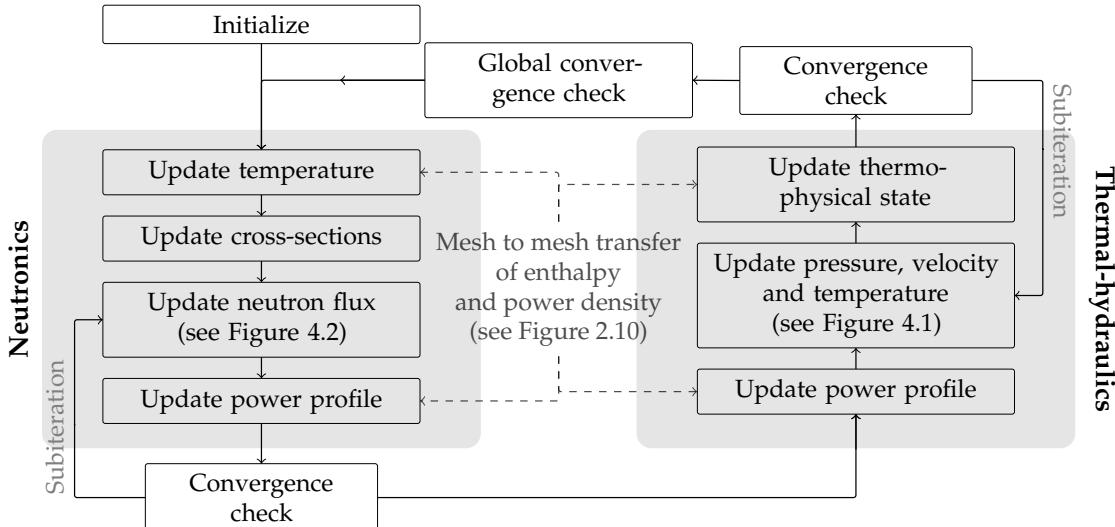
The solver is applied to the generally unstructured neutronics mesh which requires some additional attention. In the reactor context, discrete ordinates have often been implemented only for Cartesian grids, although late high-fidelity exceptions discretized with FEM are notable (see e.g. [133]). Unstructured implementations of  $S_N$  with FVM have for a long time been applied for radiative heat transfer (see e.g. [134]) with a step scheme discretization of the  $S_N$  equations. The

latter is in principle identical to an upwind differencing scheme as applied for the convective terms in the fluid momentum equation. Although the step scheme is accurate to first order only, it is stable and easy to implement and has therefore been the choice for this thesis.

As indicated in Figure 4.2, the  $S_N$  equations are solved using a pre-calculated sweep order. Since the only spatial dependence coupling with neighbor cells enters in the streaming term, each given direction ( $\Omega_m$ ) can be swept in an order such that all cells only need to be updated once. The algorithm applied is based on the work by Plimpton, Hendrickson, Burns, McLendon, and Rauchwerger [135]. It should however be noted that due to framework technical details, the parallelization is not modified accordingly (see Paper IV for further details). The eigenvalue problem is solved by the power iteration method [136].

### 4.1.3 Picard iteration scheme

The implementation of the fixed point iterative scheme discussed in Section 3.3 is presented in terms of a flow chart in Figure 4.3. The scheme alternates between the neutronics and the thermal-hydraulics, applying sub-iterations in each of the modules to increase the overall convergence rate and limit the number of multiphysics iterations.



**Figure 4.3:** Iterative scheme applied for the coupling of the thermal-hydraulics and the neutronics [Paper IV]

The most sensitive part of the simulation is the start of the iterations, and, as seen from Figure 4.3, the algorithm is initiated with an update of the neutronics and the power profile. In general, the thermal-hydraulic equations were slower to converge and found most probable to cause failure of the solver and for this reason the first solution of the temperature equation was delayed to the third multiphysics iteration. However, after a few of these modified iterations, no under-

**Table 4.1:** Geometry specification for the simulated  $15 \times 15$  assembly, with control rod guide tube values in brackets.

Fuel pin radius	0.41 cm
Cladding inner radius	0.43 cm (0.48 cm)
Cladding outer radius	0.49 cm (0.58 cm)
Pitch	1.25 cm
Fuel height	100 cm
Bottom reflector	20 cm
Top reflector	20 cm

**Table 4.2:** Mesh specification for the simulated assembly.

Region	Number of cells
Moderator	6,088,000
Fuel (per pin)	8,000
Cladding (per pin)	4,800
Gap (per pin)	1,600
Neutronics	798,000

relaxation was performed between the multiphysics iterations. An interpretation of this is that the problem is not sufficiently coupled to give stability issues.

## 4.2 Application to a $15 \times 15$ assembly

To exemplify the results produced with the described algorithms, the system presented in Figure 2.11 is simulated. The geometry for the fuel pins is provided in Table 4.1. A summary of the number of computational grids is provided in Table 4.2. As seen from the table, the moderator mesh is significantly finer than the one for the neutronics. Further details in terms of boundary conditions and the domain decomposition are provided in Paper IV, but in overall the thermal conditions are taken to be typical for the conditions in a PWR.

The neutronic calculations were performed for 8 energy groups and with  $S_8$  discretized according to the level symmetric quadrature set. The resulting scalar flux profile of the system is illustrated for the fastest energy group ( $g = 0$ ) and the thermal (lowest energy) group ( $g = 7$ ) in Figure 4.4. In addition to the surface plot, a line plot of the scalar flux at the symmetry line is shown.

Both from the line and the surface plot, artifacts of the so called ray effect can be seen. Such an effect occurs due to the inability of the provided ordinates set to accurately reproduce the angular neutron flux [137, 138] and the solution to this problem is generally to increase the number of directions simulated. The effect was clearly seen in Paper III where a 2D validation of the neutronics was performed against Serpent. Quadratures ranging from  $S_2$  to  $S_{16}$  were compared and the effect was shown to diminish with the increase in the number of ordinates.

DeHart [139] emphasizes that the effect in principle decreases with increasing numerical diffusion (typically due to a coarser grid) and increases with material heterogeneities. In the fine-mesh approach with resolved material regions, both factors mentioned are clearly problematic. In addition, the results from Paper II illustrated the shortcoming of the diffusion solver for the fine-mesh applications as it induced extensive smearing of the neutron flux as compared to the discrete ordinates solver.

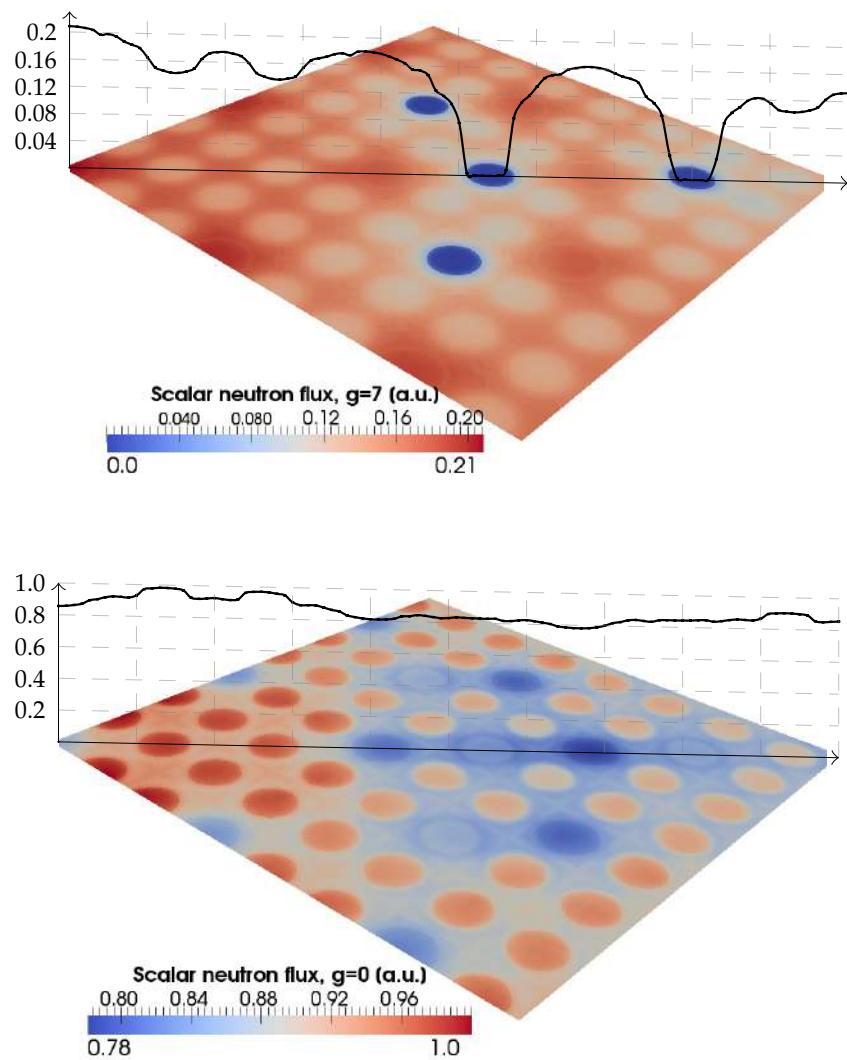
As regards the thermal-hydraulic results, a visualization of the moderator temperature in the assembly is presented in Figure 4.5. As indicated by the plotted horizontal planes, the heterogeneities of the fuel temperature are resolved with the applied mesh resolution. Whereas the total temperature rise for the simulation is a bit more than 10 K for the hot channels, the temperature difference inside one of the sub-channels is less and, accordingly, the density difference for the liquid is small. In turn, such small heterogeneities mean that the influence on the cross-sections is relatively small.

In Paper I, a study was performed on comparing  $k_{\text{eff}}$  computed with a heterogeneous temperature (as in Figure 4.5 but for a  $5 \times 5$  system) against a horizontally averaged profile. It was found that the moderator averaging had no effect on the criticality value. However, the horizontal averaging of the fuel temperature was found to give a significant effect. It should be emphasized that significant simplifications are done during the simulations, including the lack of spacers and the use of the  $k - \epsilon$  model which has clear limitations (as discussed by e.g. [116]). In this context, a time averaging from an unsteady simulation utilizing LES would be of interest to get a second, and potentially more accurate, model for the magnitude of the heterogeneities.

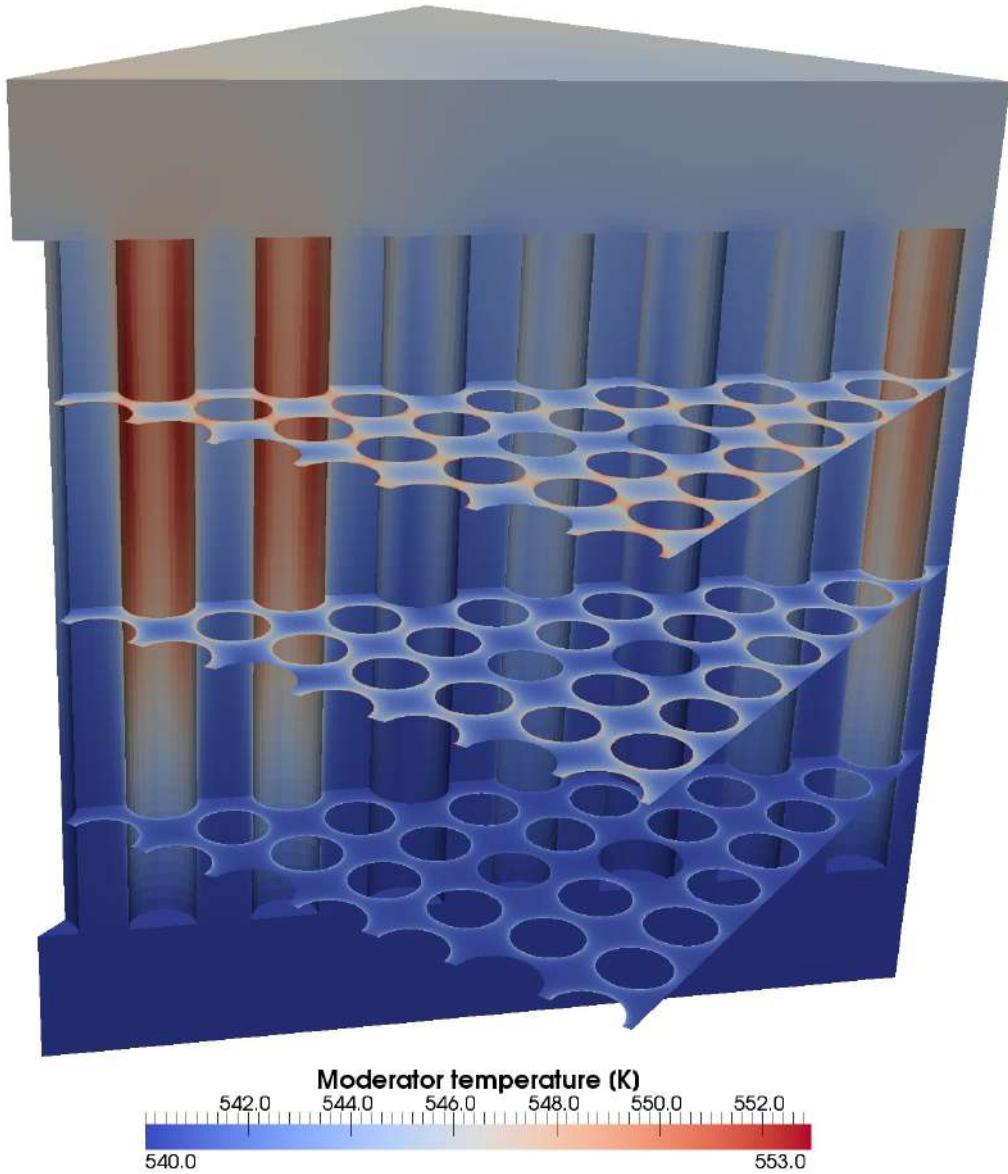
### 4.2.1 Convergence and performance

To evaluate the performance of the schemes for the separate models (Figures 4.1 and 4.2) and the multiphysics scheme (Figure 4.3) the convergence profiles for the multiphysics and the separate module iterations are provided for the first eight outer iterations in Figure 4.6. As described earlier, the first iteration is reserved for neutronics only, followed by a thermal-hydraulic iteration without the temperature equation, and from the third iteration all fields are solved. The iterations following after the 8 first ones require only a single  $S_N$  sweep and the total change in  $k_{\text{eff}}$  from outer iteration 9 to 50 is only 10 pcm (not shown), which suggests that the multiphysics dependencies are more or less fully resolved after the last iteration of Figure 4.6.

As seen from the figure, a limit of 100 sub-iterations is applied for the thermal-hydraulics. Such a limit was found to be increasing the acceleration rate, and in terms of the reasoning on the iterative schemes in Section 3.3, it is reasonable to not directly strive for full convergence for each submodule as the sub-iterations precondition the Picard iterations. In terms of the same section on the coupling, it



**Figure 4.4:** Scalar flux at mid-elevation for the fast group ( $g = 0$ , bottom) and the thermal group ( $g = 7$ , top) for a quarter of a  $15 \times 15$  system as outlined in Figure 2.11 [Paper IV]



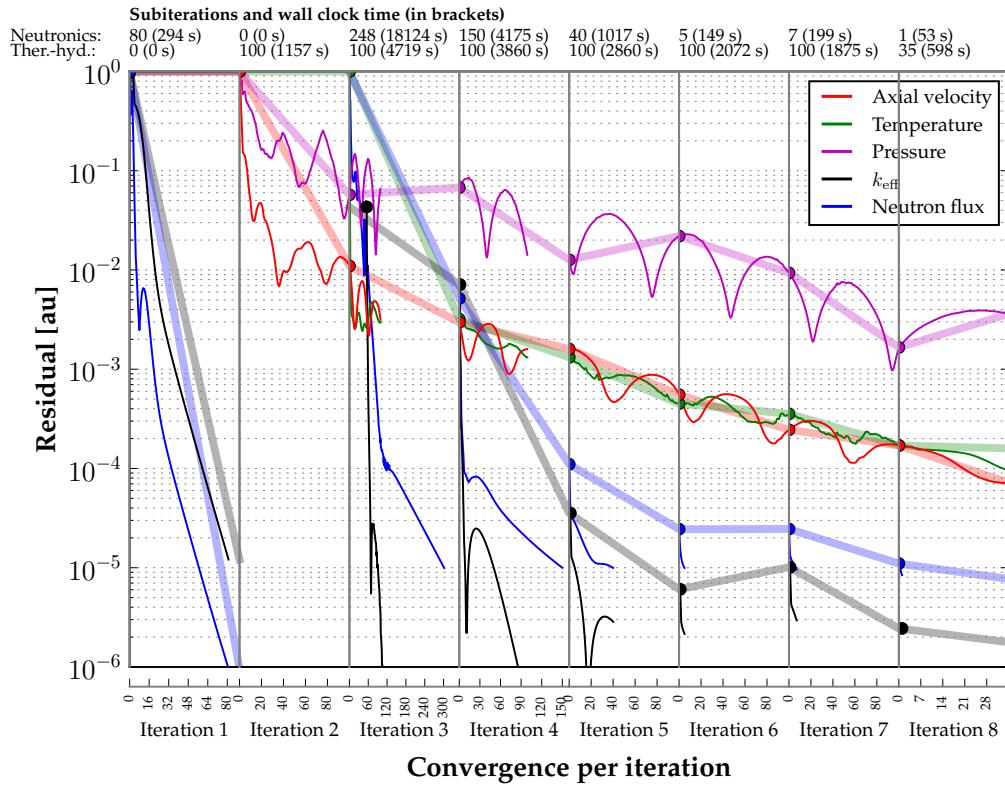
**Figure 4.5:** Moderator temperature at three horizontal planes, with the axial dependence at a diagonal cut in the background [Paper IV].

is interesting to consider the potential benefits of a non-linear solver. As regards the Anderson acceleration, it would be of interest to see the effect on the number of multiphysics iterations. However, as the number of iterations are already relatively few, it is perhaps not to be expected that the acceleration would significantly change the convergence. Instead, as also discussed in Section 3.3, better solvers for the separate fields of physics would likely be more beneficial.

Needless to say, the relative computational time between the neutronics and thermal-hydraulics is strongly dependent on degrees of freedom in the equations, i.e. for neutronics on the number of groups, directions and the grid resolution and for the thermal-hydraulics only on the latter. Nevertheless, it is interesting to com-

pare the simulation effort and we found that for the presented discretization (8 groups,  $S_8$ ) the neutronics and thermal-hydraulics had similar computation time. It should be emphasized that all fields are initiated as flat (i.e. space independent) which is possibly slow but at least giving an honest view of the convergence properties of the system.

As regards the earlier descriptions and discussions on the parallelization implementation (Section 2.1.3), the presented case was run on 64 processors and a steady-state converged solution was computed in a total of 14 wall-clock hours. A full benchmark of the parallelization would require a varied number of cores (strong scaling) or by increasing the problem size (weak scaling).



**Figure 4.6:** Convergence results for the coupled system, with multiphysics iteration convergence as opaque broader lines and the corresponding sub-iterations as thinner lines [Paper IV].



---

# CHAPTER 5

---

## Transient coupled solver application and analysis

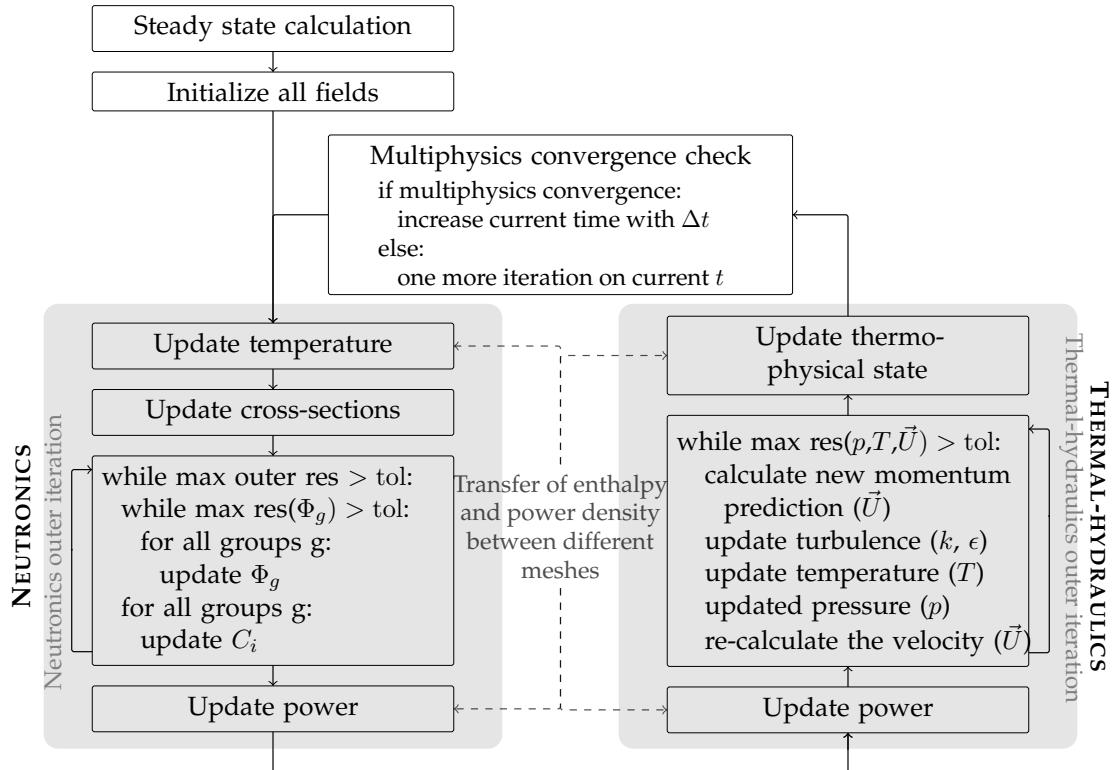
After the steady-state simulations in the previous chapter it follows naturally to continue to transient cases for the same type of systems. For understandable reasons the unsteady models are computationally even more demanding than the steady-state equivalents; instead of converging the multiphysics problem once, all couplings must be resolved in every time step. Nevertheless, the transient cases are potentially of greater interest than the steady simulations as local temporal responses are recovered, and thus an additional contributor to local heterogeneities can be investigated.

This chapter includes implementation details and some of the results from Paper V as well as a description and the results of Paper VIII. In the former, the overall framework is tested for unsteady simulations of a short transient with changing the moderator inlet temperature. The latter proposes a verification method for transient neutronics codes, here applied to a simplified 2D system and shown to give a good agreement with the analytical expressions.

### 5.1 Implementation and framework details

The transient algorithm is implemented with an iterative scheme as outlined in Figure 5.1. In contrast to the steady solver, the transient scheme is purely solving the diffusion equation for the neutronics. Furthermore, the precursor equations are solved by additional iterations. Other schemes are proposed in the literature (see e.g. [140]), where the precursors are implicitly treated directly in the neutron flux equations, which is an area for future improvement. The solution algorithm for the transient solver is again very similar to the steady solver, solved with a segregated approach (PISO).

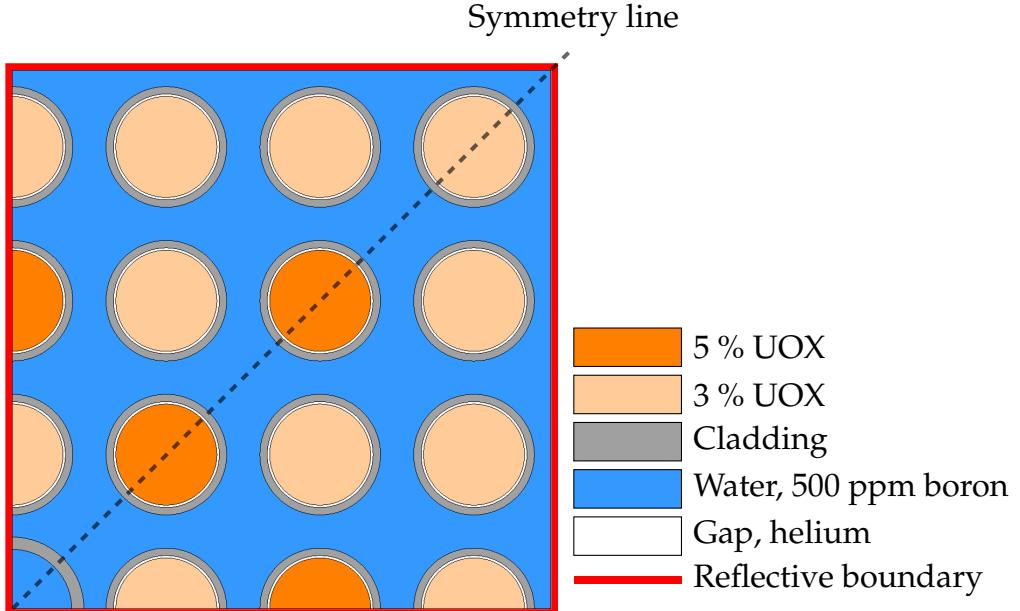
As indicated in Figure 5.1, the transient algorithm is initiated with a solution of the steady multiphysics problem, which was found important for two reasons. First, a good starting guess was found necessary to avoid stability issues with the multiphysics coupling. Second, the system needs to be close to criticality, i.e.  $k_{\text{eff}} = 1$ , at the start of the transient. If the system is far from criticality it either needs to be modified in terms of geometry or material composition, or a renormalization of the fission source needs to be applied. The former remedy was generally found problematic as it requires a re-computation of the cross-sections for a new fuel composition or moderator boron concentration (see Paper V). It should be noted that a desirable (but not implemented) feature would be to include a criticality search in terms of a dynamic boron concentration. For the case of fission source renormalization, the criticality value from the steady simulation is applied as a static (i.e. not changed during the simulation) renormalization. Although such a measure is not physically correct it provides a simple method to test and feature the transient solver.



**Figure 5.1:** Iterative scheme applied to solve neutronics and thermal-hydraulics coupling [Paper V].

## 5.2 Application to a $7 \times 7$ assembly

In Paper V the transient solver was applied to a quarter of a  $7 \times 7$  assembly, again under PWR-like conditions. The horizontal geometry and the material composi-

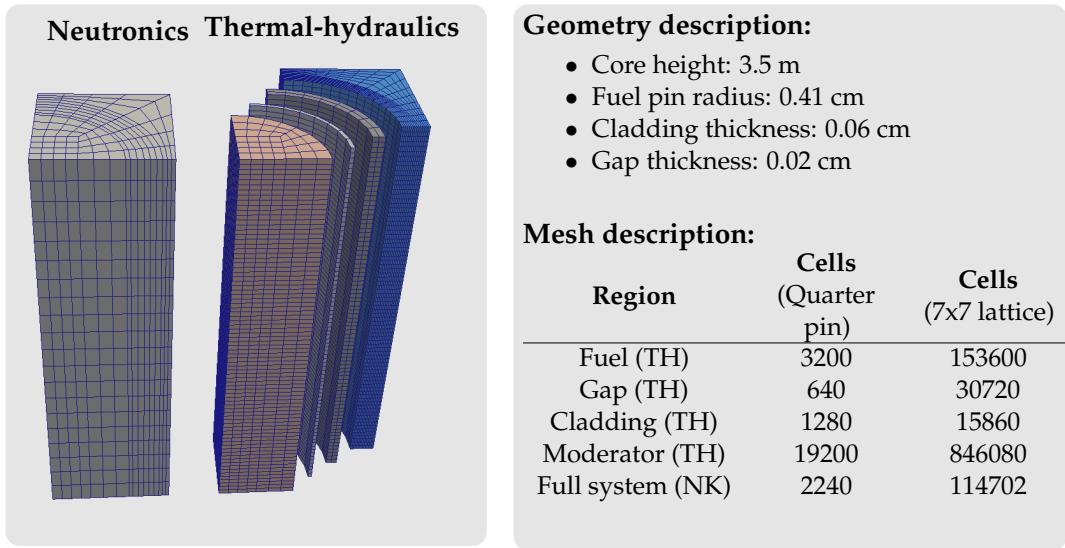


**Figure 5.2:** Horizontal geometry for a quarter of a  $7 \times 7$  system with material regions and the reflective boundary conditions indicated. [Paper V]

tion of the system are outlined in Figure 5.2. Two different enrichments of fuel pins were included and the water contained a boron concentration tested to give a system close to criticality. The overall geometry, the axial mesh resolution and the number of computational cells are provided in Figure 5.3.

The case is run for 10 seconds and between 2 and 3 simulated seconds the moderator inlet temperature is linearly decreased from an initial value of 550 K to 540 K. The uniform temperature decrease at the inlet is propagated through the system and, as an illustration of the response of the system, Figure 5.4 shows the instantaneous moderator temperature (Figure 5.4a) and the fuel power density (Figure 5.4c) after 3 seconds. Similar to the previously presented steady-state results, the subchannel heterogeneities are resolved in the moderator.

In addition, Figure 5.4b shows the temporal development of the relative difference between the maximum and minimum horizontal temperatures in the moderator at mid-elevation of the system. As seen from the figure, the response to the inlet perturbation is not homogeneous, and an increase in the heterogeneity is seen following the temperature decrease. Similarly, Figure 5.4d shows the mid-elevation relative difference in the maximum and minimum horizontal power densities. For the power, the transient leads to an initial decrease in the heterogeneity followed by an increase. Such results would argue that the fine-mesh approach captures the physics not seen when assuming a flat moderator temperature (as e.g. done in a subchannel code). Notably, the magnitude of the difference from the beginning of the transient to the end is relatively small and is arguably of little significance for the presented case.

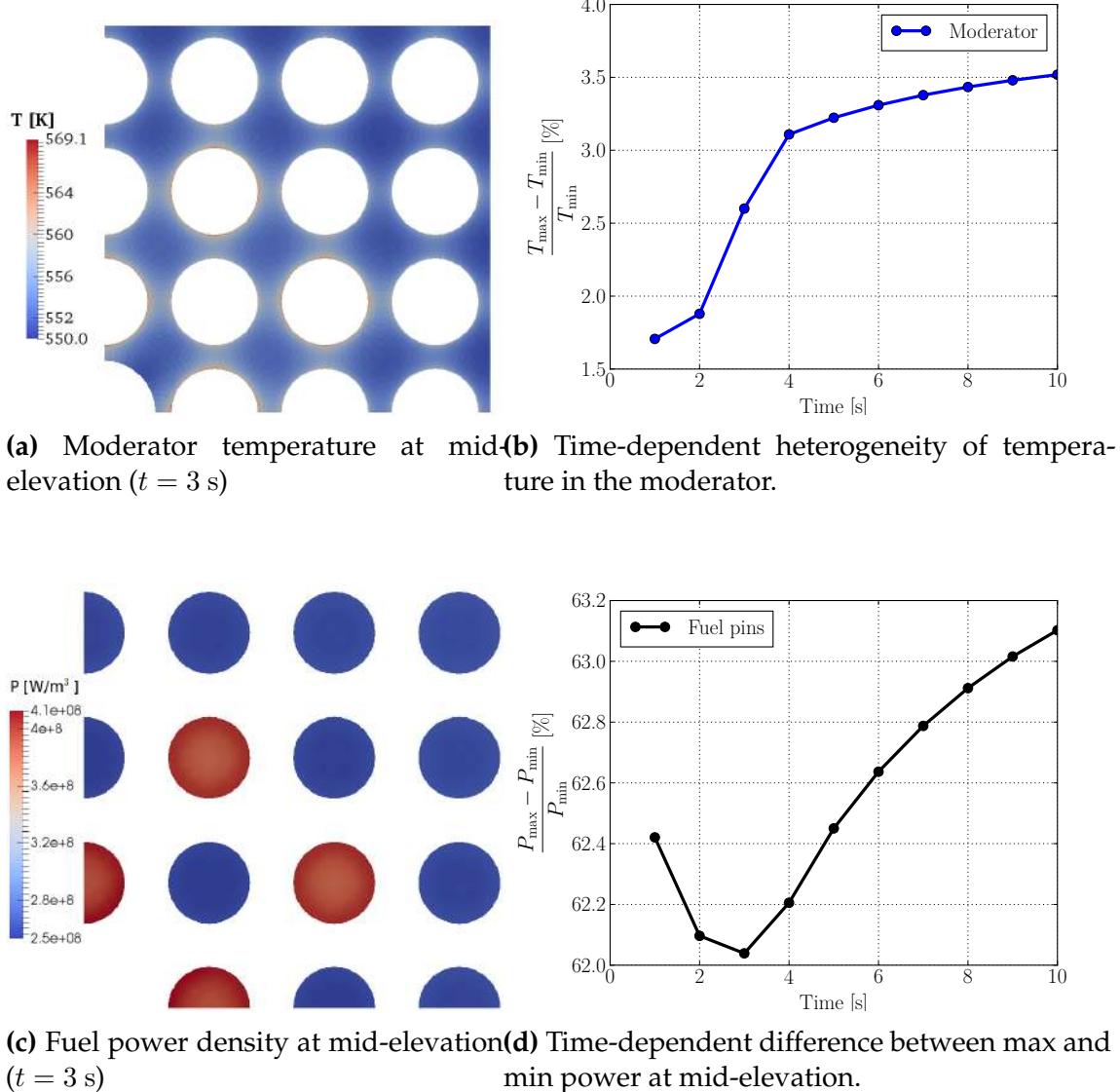


**Figure 5.3:** Description of the geometry and the computational grid for the  $7 \times 7$  system used for transient simulations. Index TH indicates meshes for the thermal-hydraulics, whereas NK indicates the mesh for the neutronics [Paper V].

Considering the computational effort, the case was run on 16 CPUs for a total wall-clock time of 59 hours. In contrast to the case presented in Section 4.2, the time is now primarily spent with the thermal-hydraulics ( 89%). However, the simulation was here performed with only 4 energy groups and using the diffusion approximation. It is interesting to note that for the majority of the time steps, only a limited number of both the multiphysics iterations and the inner thermal-hydraulics iterations are required. Such an observation suggests that a non-linear formulation would be of limited value for the presented simulation. Moreover, as the applied time step is limited by the Courant number in the CFD simulations an implicit algorithm (e.g. solving all neutronics and thermal-hydraulics equations together in a non-linear fashion) would not necessarily extend the time step.

## 5.3 Time-dependent neutronics verification method

To verify a correct code implementation, in terms of the modeled conservation equations as well as the multiphysics methodology, is typically a difficult task. Except for a direct validation against measurements, codes are not seldom compared and, speciously, verified against each other. The latter is particularly questionable if both codes are based on the same method, i.e. likely to produce the same inaccuracies or similar problems with e.g. discretization. For the fine-mesh multiphysics simulations presented in this thesis, a direct comparison to numerical experiments is difficult. To the knowledge of the author, no equivalent mea-



**Figure 5.4:** Axial slices of moderator temperature and power density with corresponding time development of maximum and minimum horizontal values at mid-elevation [Paper V]

sured data exist. For steady neutronics solvers, verifications against Monte Carlo codes are often performed, and are in some sense an accepted practice in the nuclear field. Such a comparison was done in Paper III for the  $S_N$  solver.

In Paper VIII a method for transient neutronics verification is proposed. The foundation of the method is the application of a stationary perturbation and extraction of the point-kinetic component of the system response. A component of the computed response is verified against an analytical point-kinetic prediction of the same system. The derivation is given in its full detail in Paper VIII, and here only the main characteristics are covered together with an example case.

### 5.3.1 Overview of the methodology

The proposed verification scheme is applicable in both frequency and time space, where the latter approach is exemplified here. The methodology is based on a stationary perturbation applied in one of the fuel regions and in Paper VIII implemented as a sinusoidal variation of the thermal group total cross-section, such that

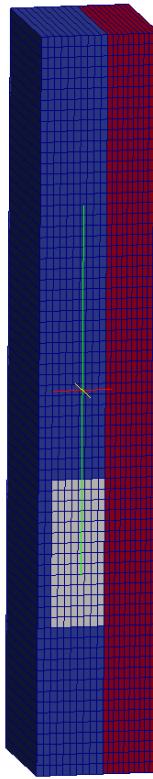
$$\Sigma_{T,2}(\mathbf{r}, t) = \Sigma_{T,2,0}(\mathbf{r}) + A \sin(\omega t), \quad (5.1)$$

where  $\Sigma_{T,2,0}(\mathbf{r})$  is the unperturbed total cross-section of the thermal group and  $A$  and  $\omega$  correspond to the amplitude and frequency of the perturbation, respectively. It should be noted that the derivation presented in Paper VIII was done for a two-group formulation, which is thus followed in the example. In addition to the two-group diffusion equation, the solution to the adjoint problem is required and was consequently implemented and added to the neutronics module.

In order to compute the response of the system, the variation of the power ( $\delta P(t)$ ) and the reactivity ( $\delta \rho(t)$ ) of the system need to be extracted during the simulation. The quantities are computed as volume integrals (i.e. discrete summations over the computational grid) involving the adjoint flux, the cross-sections and the neutron group velocity (for a detailed explanation of the integral expressions refer to Paper VIII). The computed integral values are then fit by a post-processing utility to estimate the phase change as compared to the original perturbation (eq. 5.1) and the amplitude of the responses for  $\delta P(t)$  and  $\delta \rho(t)$ . The fitting function is defined as

$$y(t) = a \sin(\omega t + p) + kt + c, \quad (5.2)$$

where the constant ( $c$ ) and linear variation with time ( $kt$ ) allows for an offset and slow variation of the base line of the fitted functions, which is of importance if the system is not exactly critical at the start of the perturbation.



**Figure 5.5:** Slab system geometry for the example of the point-kinetic verification method. The system is compressed in axial direction, actual length of the system was 50 cm. The blue region indicates the fuel (0.45 cm in width) and the red region indicates the moderator region (0.25 cm in width). The grey region corresponds to the region of the fuel where the time-dependent perturbation was applied. Symmetry boundary conditions are imposed in the horizontal direction. [Paper VIII]

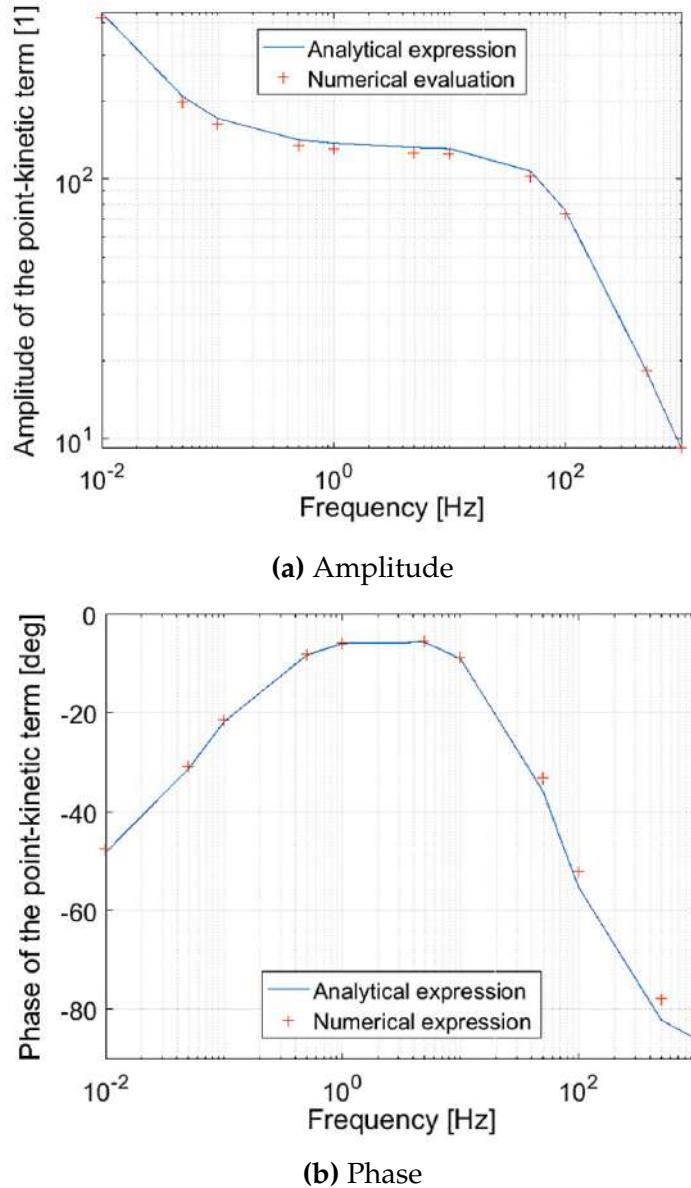
### 5.3.2 Application to a two-region slab system

As an example of the method, a simplistic 2D slab case is presented. The geometry and the computational grid are shown in Figure 5.5. The simulations are initiated from a steady-state multiphysics solution, but performed with a frozen state of the thermal-hydraulics. In addition, a renormalization of the fission source is performed (as discussed in Section 5.1).

To verify the solver, the neutronics solver is run for a range of different frequencies, each providing one point for verification. As a result of the many repeated runs, the verification process is computationally heavy. On the other hand, it was found sufficient to run each simulation for a few periods of the perturbation in eq. (5.1), at least as long as the power is not significantly drifting after the renormalization.

The results of the simulations are provided in Figure 5.6, where the amplitude and frequency of the point-kinetic components of the system response are com-

pared to the analytical solution of the point-kinetic equation. As seen from the figure, the results of the FVM code match both the amplitude and phase well, with a maximum deviation in the amplitude of less than 5%. Such a result is important not only for the presented solver but also as an example of a method with great potential to other 3D transient neutronics codes, thus filling an earlier vacuum in terms of lacking verification methods.



**Figure 5.6:** Frequency dependence of the amplitude (top) and the phase (bottom) of the point-kinetic zero-power transfer function for the slab presented in Figure 5.5 [Paper VIII]

---

## CHAPTER 6

---

# On two-phase flow in LWRs

Up to this point in the thesis the coolant of the reactor was considered to be single-phase liquid water. Such a flow is representative for nominal conditions in a PWR, but not at all for describing the flow in a BWR. In the latter, the water enters the core in a subcooled liquid phase but boils and over the height of the channel it traverses a number of flow regimes. Whereas single-phase flow is relatively well understood and well reproduced with simulations, two-phase flow is still, after more than half a century of studies, a very challenging topic.

In this chapter, the two-phase flow problem of liquid and vapor water in the BWR is briefly introduced (Section 6.1), with primary focus on different strategies for simulation and modeling (Section 6.2). Due to the difficulties of formulating a single universal model covering all regimes and scales in a computationally feasible manner, the problem inevitably needs to be narrowed for the scope of this thesis. In the current work two aspects are of primary focus, namely simulation of subcooled boiling flows (Section 6.3) and dynamic characteristics of the two-fluid model (Section 6.4).

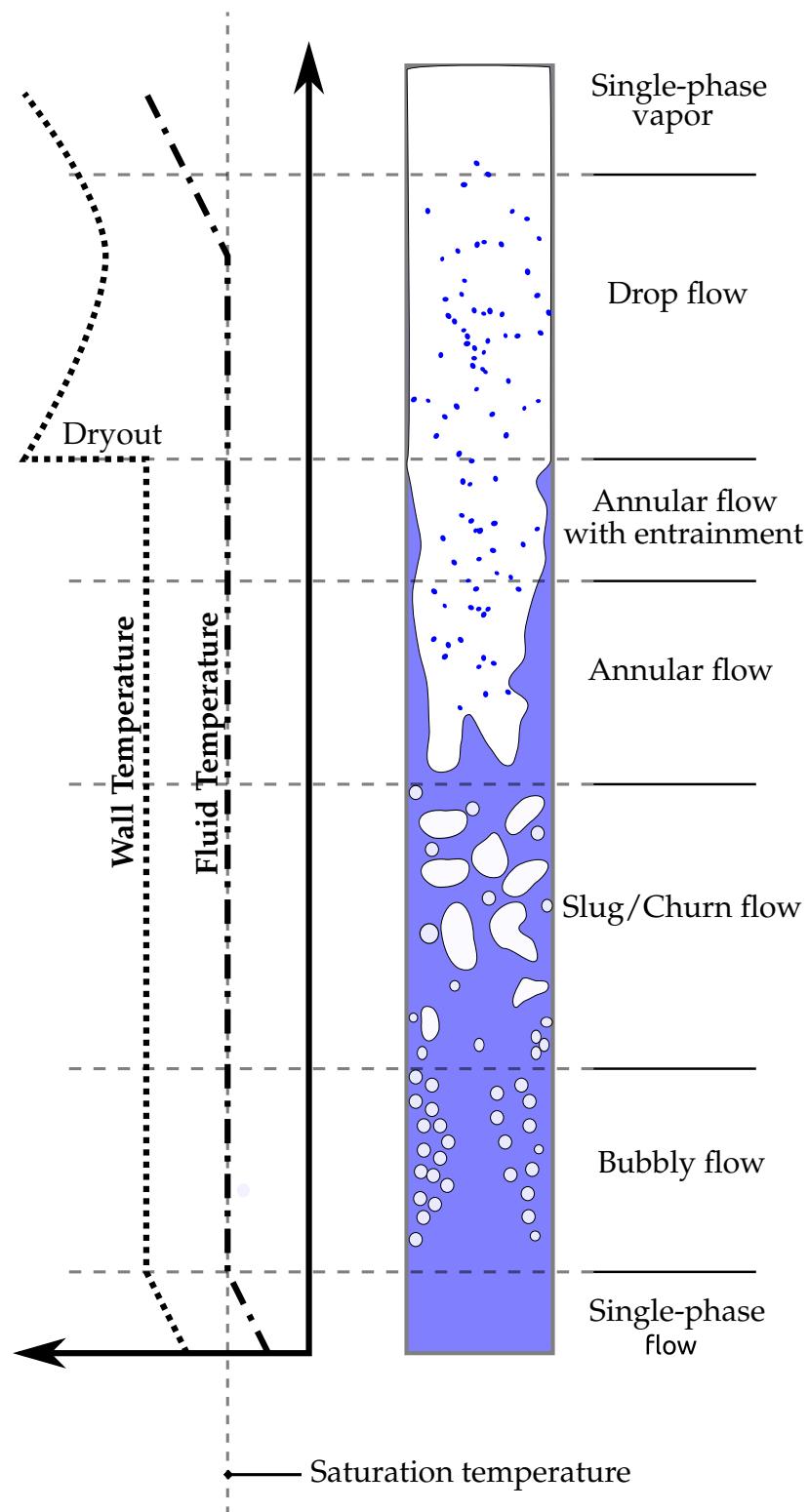
In relation to the previous two chapters on the application of the multiphysics tool, the current chapter is in a sense more generic. The simulations are performed on simplistic geometries and in parts reduced in terms of model complexity. To a large extent this is done to better illustrate the proposed ideas, and limit the distraction of the huge number of force models, flow regime correlations, etc. Furthermore, the purpose of the seemingly reduced models is to clarify some of the underlying complexities in the models often foreseen in routine application of multiphase CFD.

## 6.1 Perspectives on the simulation challenges

The methods to simulate multiphase flows are in many sense as multifaceted as the configurations of the flow systems themselves. To model a particulate flow of gravel requires completely different strategies as compared to a slug flow in a channel of the BWR. Needless to say, the difficulties are accordingly distinct for each problem and undoubtedly multiphase flow systems constitute some of the most outstanding challenges in the field of fluid mechanics.

The BWR core is an important example of the intricacies of two-phase flow in industrial devices. To extend on the complexities and the motivations for CFD simulations of BWR subchannels, Figure 6.1 is used as an illustration. The figure shows a heated channel with flow regimes ranging from bubbly flow to the extreme of single-phase vapor. Some of the reasons for complications include the phase changes due to boiling and condensation, the different characteristics of the flow regimes and the issues with overlapping scales of the phase heterogeneities and the geometry in the narrow channels. In some more detail:

- *Phase change* – The energy released from fission in the fuel heats the water entering the core to the saturation temperature and induces boiling at the cladding surface. The liquid phase reaches saturation conditions close to the wall after only a short axial distance (see the sketch in Figure 6.1), whereafter the water boils at the wall. In addition, during the subcooled phase the bubbles are transported from the wall to the bulk of the flow and they condense, which is one of the phenomena targeted in Section 6.3.
- *Flow regimes* – As the vapor phase continues to increase, the flow regime in the channel changes. The initial bubbly phase of vapor is transformed into larger regions of void, depicted in Figure 6.1. If the flow is further heated, the regime might even become the reverse of the initial bubbly flow, where instead the liquid phase is dispersed as droplets in the vapor bulk. It should be noted that the latter conditions, starting from the point of dryout, must by all means be avoided as there is a risk to melt the fuel due to the severe decrease in the heat transfer from the fuel.
- *Separation of scales* – In contrast to the single-phase flow regime, separation of scales is not trivially fulfilled in the case of multiphase flow. In short, the lack of separation of scales is a result of that the size of the void regions approaches the size of the computational grid. As an illustrative example consider the slug regime in Figure 6.1. It is readily imagined that in order to resolve the velocity profile of the depicted channel, the computational cells would need to be on the size of (or rather even smaller than) the void structures. Such an issue has been a long standing challenge of multiphase flow [142], and due to the interfering scales it becomes difficult to formulate a space averaged model to resolve fine scales of the simulations (such as the flow inside the BWR subchannel) while still fulfilling the separation of scales [143].



**Figure 6.1:** Overview of forced convection boiling phases in a channel (in part redrawn from [141]).

As indicated in the motivation for the fine-mesh simulations (Section 1.3.1), numerical predictions of CHF are both interesting and important challenges in CFD simulations of BWRs and a major driver for the development of new models [144]. To accurately model CHF, multiple scales are relevant, ranging from the micro scales of the growth of a bubble on the surface of the cladding to the transport of bubbles (or larger chunks of void) in the subchannels, and to the scales of the full fuel assembly. For the first, the growth of the bubble and the departure from the wall, very detailed descriptions of the flow are generally required (see e.g. [145]) and this type of simulations fall outside the scope of the current work. For the latter two scales (bubble departure and the entire subchannel), the focus is primarily on the flow of two phases and the potential phase change inside the fluid which are the topics of the next section, introducing some modeling approaches of the flow problem.

## 6.2 Models for two-phase liquid and gas flows

In contrast to the subchannel approach (Section 1.2.2), which to a large extent is driven by empirical, macroscopic relations and tuned models, the fine-mesh or CFD approach focuses on a physics-based modeling of the flow. In coarse terms, the methods could be divided into two groups: interface tracking methods and averaging methods. The former is sometimes denoted DNS-like methods and the latter is often deployed as the two-fluid model.

### 6.2.1 DNS-like methods

The concept of DNS in multiphase flow is not as straightforward as in the single-phase case. In the latter, the Navier–Stokes equations are resolved on every scale down to the smallest fluctuations in the flow [116], with small risk of adventuring the separation of scales. For multiphase flow, DNS is occasionally referred to as simulations in which the interfaces of the multicomponent flow are resolved. However, the exact nomenclature has for sure been debated (see for example the note by Yadigaroglu [146]), and it should be noted that the increase in computational resources has enabled bubbly flow DNS simulations in the single-phase sense, i.e. with full resolution of each of the phases [147].

Even though the flow fields can be fully resolved with the Navier–Stokes equations without modifications or models, there are still fundamental challenges with phenomena such as bubble coalescence and breakage related to the effects of surface tension. For the latter, a certain amount of modeling is still required (see e.g. [147]). Furthermore, the systems that can be resolved in a DNS manner are still small, extremely computationally demanding and primarily focused on the bubbly flow regime. Consequently, a simulation of a full BWR channel with anywhere close to realistic void fractions is still out of reach. Nevertheless, DNS

like simulations are of interest to formulate models for macroscopic correlations, or to perform multiscale hierachal simulations (such as reported for gas-solid flows [148]).

Despite that the BWR subchannel problem does not easily lend itself to full DNS calculations, interface tracking methods (without all length scales resolved) are still of interest and relevance. A range of different methods have been proposed, whereof the most notable include the Volume of Fluid method (VOF) [149, 150], the Level Set method (LS) [151, 152] and front tracking [153]. Interesting examples include single bubble condensation [154, 155], which is potentially valuable to formulate correlations for a coarser (two-fluid) model. The DNS simulations alike, the computational effort required also by coarse applications of interface tracking methods precludes their use for full subchannel simulations. For larger scale CFD simulations an averaging method is instead required.

### 6.2.2 The two-fluid model

The two-fluid model describes the two phases in an Eulerian-Eulerian frame of reference [143, 156] and instead of tracking the interface, the phases are treated as interpenetrating continua. The presence of the phases is described in terms of a volume fraction, for liquid-vapor systems typically denoted void fraction. Due to the fact that the interface is not directly tracked, the computational burden of the method is much smaller than in the case of DNS-like simulations. On the other hand, the averaging comes with a price. In comparison to the DNS-like methods described in the previous section, a larger degree of modeling is required, in particular for the interphase mass and momentum exchanges.

The governing equations of the two-fluid are here briefly outlined for the purpose of the applications in Sections 6.3 and 6.4, but without a detailed derivation (for a detailed discussion on the procedure see e.g [6]). The model relies on one or more averages on the Navier-Stokes equations for each of the phases. Such an averaging can be performed as volume averages (see e.g. [6, 157]), ensemble average (see e.g. [158, 159]) and/or in terms of time-averages. The result is a set of mass, momentum and energy conservation equations for each of the phases. In the current work the mass conservation is given as [157]

$$\frac{\partial \alpha_i \rho_i}{\partial t} + \nabla \cdot (\alpha_i \rho_i \mathbf{U}_i) = 0, \quad (6.1)$$

where  $\rho_i$  is the density,  $\alpha_i$  the phase fraction and  $\mathbf{U}_i$  is the velocity, and the momentum equations as

$$\frac{\partial \alpha_i \rho_i \mathbf{U}_i}{\partial t} + \nabla \cdot (\alpha_i \rho_i \mathbf{U}_i \mathbf{U}_i) = -\nabla \cdot (\alpha_i (\bar{\tau}_i + \bar{\tau}_i^t)) - \alpha_i \nabla (P) + \alpha_i \rho_i \mathbf{g} + \mathbf{M}_i, \quad (6.2)$$

again written for each of the phases and where  $\bar{\tau}_i$  and  $\bar{\tau}_i^t$  are the viscous and turbulent stress tensors, respectively,  $P$  is the pressure and  $\mathbf{M}_i$  represents the interfacial momentum transfer. Furthermore, conservation equations for energy are

required for diabatic simulations, and in the case of boiling or condensation additional terms appear in all the equations. The latter is the case in Paper VI, whereas only eq. (6.1) and (6.2) are solved in the stability and dynamics investigations in Paper III and VII. The set of equations (6.1) and (6.2) is commonly solved in a similar manner as the PISO algorithm for single-phase flow. The implementation utilized in the thesis closely follows the derivations by Rusche [160] and Weller [159].

As already mentioned the two-fluid model suffers from a number of shortcomings. The following (non-exclusive) list of issues and limitations is notable:

- *Lack of stability* – It has previously been reported that the two-fluid model in a pure form (i.e. without additional artificial or physical model-based viscosity) suffers from lack of hyperbolicity, which might lead to instable behavior of the simulation [161, 162]. Such instabilities are the topic of Paper III and Paper VII discussed further in Section 6.4.
- *Lack of size distributions* – As a result of the averaging, the specific information of the phase interface is lost. Consequently, for the example of a bubbly flow the bubble size distribution is unknown and as a result all size-dependent correlations (e.g. momentum exchange terms and the condensation rate) cannot be accurately applied. A potential remedy for this is to regain size distribution information from a PBE, which is the topic of Paper VI.
- *Lack of separation of scales* – As the two-fluid model equations are commonly derived under the assumption of small variations of all fields (at least for volume averaging) only low void fractions or small bubbles could theoretically be treated. This is commonly abused as the computer codes often run also for higher void fractions. Arguably, carrying out ensemble average is advantageous since separation of scales is not an immediate issue for that procedure. However, in practical cases the interphase exchange terms are typically implemented in a volumetric sense [163], thus again a prey for the mentioned issues. Interestingly there are some (theoretical) examples of derivations not requiring separation of scales/small local gradients [164, 165], which are of future interest.

From a historical perspective the two-fluid model, in a lower dimensional formulation, has been the standard choice for the system codes to describe the multi-phase flow. Also in so-called mechanistic CFD modeling, i.e. 3D simulations with (more or less) well-founded physics-based models (see e.g. [166]), the two-fluid framework has been and still prevails as the dominant methodology. By large the two-fluid model should perhaps be seen as the workhorse of the two phase simulations, and although in many senses imperfect, it enabled interesting 3D calculations already 20 years ago [167].

In terms of the first part of this thesis, the fine-mesh multiphysics framework, the two-fluid solver is a good candidate for coupling to the neutronics. In par-

ticular, the possibility to simulate a complete subchannel is relevant. A DNS-like simulation of only a few bubbles is arguably too small to be of relevance for the coupling.

## 6.3 Population balance for subcooled bubbly flow

As described in the previous section, the averaging of the Navier–Stokes equations for the two–fluid model results in that the microscopic details of the flow are lost. For a bubbly flow, as depicted in the region close to the inlet of the channel in Figure 6.1, the loss of interface information between the vapor and the liquid phase results in an unknown distribution of bubbles. In practical terms, this implies that the two-fluid simulations are performed for a single size and shape of bubbles, or potentially estimated from an average interfacial area concentration [168]. A remedy for the information lost in the averaging is to apply a PBE to retrieve additional knowledge of the state of the dispersed phase. For gas–liquid flows it is common to track the bubble size or volume [169–171], but in principle other parameters such as bubble shape or velocity could also be described with the PBE.

In particular, for simulations of the onset of boiling and the transport of bubbles in a subcooled liquid the size distribution has been argued to be of importance [172]. Such a distribution is applicable both in the description of the condensation in the liquid and the aggregation and breakage of the bubbles. Thus, a subcooled bubbly flow is an interesting candidate for PBE simulations. In the nuclear community, the primary choice for subcooled boiling simulations with PBE has been the Multiple-size-group method (MUSIG), which represents the size distribution with a fixed set of, *a priori* determined, sizes. MUSIG has previously been successfully applied to subchannel simulations [172–174].

An alternative approach to solve the PBE is given by the Direct Quadrature Method of Moments (DQMOM), which relies on dynamic sizes allowed to vary dynamically throughout a simulated domain [175]. An advantage with DQMOM over MUSIG is that the use of non-fixed sizes allows to describe the distribution with the same accuracy for fewer degrees of freedom [170, 176]. DQMOM has previously been applied for adiabatic cases in the field of nuclear engineering [171] and in other fields for evaporation simulations [177]. In Paper VI, a formulation for the DQMOM of condensation of bubbles is proposed, implemented and compared to MUSIG. To give relevance to the example provided in Section 6.3.2, a brief overview of the methodology is given in the following section.

### 6.3.1 PBE formulation

The PBE for a bubble size distribution is written as [178]

$$\frac{\partial f(\xi, \mathbf{r}, t)}{\partial t} + \frac{\partial}{\partial \xi} \left( \frac{\partial \xi(\mathbf{r}, t)}{\partial t} f(\xi, \mathbf{r}, t) \right) + \nabla \cdot (\mathbf{U}(\mathbf{r}, t) f(\xi, \mathbf{r}, t)) = S(\xi, \mathbf{r}, t), \quad (6.3)$$

where  $f$  is average number density,  $\xi$  is the length scale (diameter) of the bubbles,  $\nabla$  refers to the convection in space and  $S$  is a source term which appears due to condensation and aggregation of bubbles. The velocity in eq. (6.3) is computed from the momentum equation for the vapor phase and is in the current methodology independent of the bubble size. To simulate condensation, the second term on the left hand side is written in terms of the condensation as

$$\frac{\partial \xi(\mathbf{r}, t)}{\partial t} = C(\xi, \mathbf{r}, t). \quad (6.4)$$

It is particularly important that the condensation model is allowed to have a size-dependence as empirical models typically introduce the size in the correlations [179]. As briefly mentioned, a two-fluid solver not complemented by the PBE is limited to a static size of bubbles.

### DQMOM

In DQMOM, the average number density is discretized in terms of  $N$  abscissas ( $\xi_i$ ) and weights ( $w_i$ ) such that

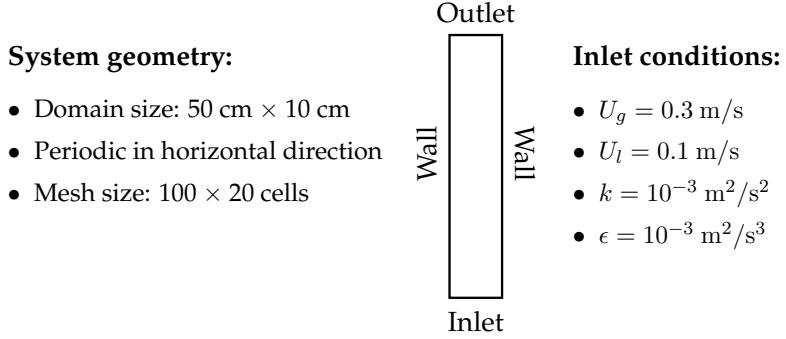
$$f(\xi; \mathbf{x}, t) \approx \sum_{i=1}^N w_i(\mathbf{x}, t) \delta(\xi - \xi_i(\mathbf{x}, t)). \quad (6.5)$$

To close the set of equations for the weights and abscissas, a moment transform is applied to eq. (6.3). The transform results in coupled transport equations for the abscissas and the weights based on the  $2N$  first moments of the PBE. An advantage of DQMOM over some similar, moment based, methods is that the equations are relatively easy to implement, only requiring a discretized time derivative and convective term. The source terms of the transport equations are computed from a cell-wise linear system of size  $2N \times 2N$ , and to resolve the couplings between the equations an iterative scheme is applied. For further computational details the interested reader is referred to Paper VI.

### MUSIG

As a reference for the proposed DQMOM formulation for subcooled boiling, a MUSIG model was implemented. The method is entirely based on models from the literature, where the PBE is often written in terms of bubble mass. Accordingly, eq. (6.3) is formulated as a set of vapor fraction equations

$$\frac{\partial \alpha_{g,j} \rho_g}{\partial t} + \nabla \cdot (\alpha_{g,j} \rho_g \mathbf{U}_g) = S_j, \quad (6.6)$$



**Figure 6.2:** Geometry and boundary conditions for the rectangular, horizontally confined system applied in Case 3 of Paper VI.

where  $j$  is the number of the class,  $S$  is again a source term and  $U_g$  is common to all bubble sizes. For MUSIG the condensation is implemented as a source term coupling the equation for the current  $j$  to the sizes below and above ( $j - 1$  and  $j + 1$ ).

### 6.3.2 Application to channel flow with condensation

To exemplify the DQMOM formulation, the case presented in Figure 6.2 is studied. The simulation domain is a 2D channel with no-slip conditions for the liquid at the horizontal walls. A small superficial velocity is applied for the bubbles and the inlet liquid temperature is subcooled by 1 K as compared to the saturation temperature. Furthermore, all thermophysical properties are computed by interpolation from tables handling both the pressure and temperature dependence. Finally, the inlet bubble size distribution is computed according to a normal size distribution with an average size of 7 mm bubbles.

The PBE methods (DQMOM and MUSIG) are both coupled to the two-fluid model with all terms handling the phase change due to condensation included. As the convection of the bubbles is directly simulated by DQMOM/MUSIG, the continuity equation is not explicitly solved in the two-fluid solver. The model includes drag, virtual mass, lift and turbulent dissipation interphase forces, where the  $k - \epsilon$  model is used to solve the turbulence in the liquid phase. For the mentioned forces, the size distribution is used to compute the total momentum force as a sum of the contributions from each abscissa or class for DQMOM and MUSIG, respectively. In addition, the condensation model has a size dependence as previously discussed.

Example results of the simulation are presented in Figure 6.3. The top plot (Figure 6.3a) shows the development of the void fraction over the simulated channel. As seen, the initial void fraction quickly decreases as the vapor bubbles condense and shrink. The latter effect is detailed in the bottom plot (Figure 6.3b)

where the average size of the bubbles in the system is computed as

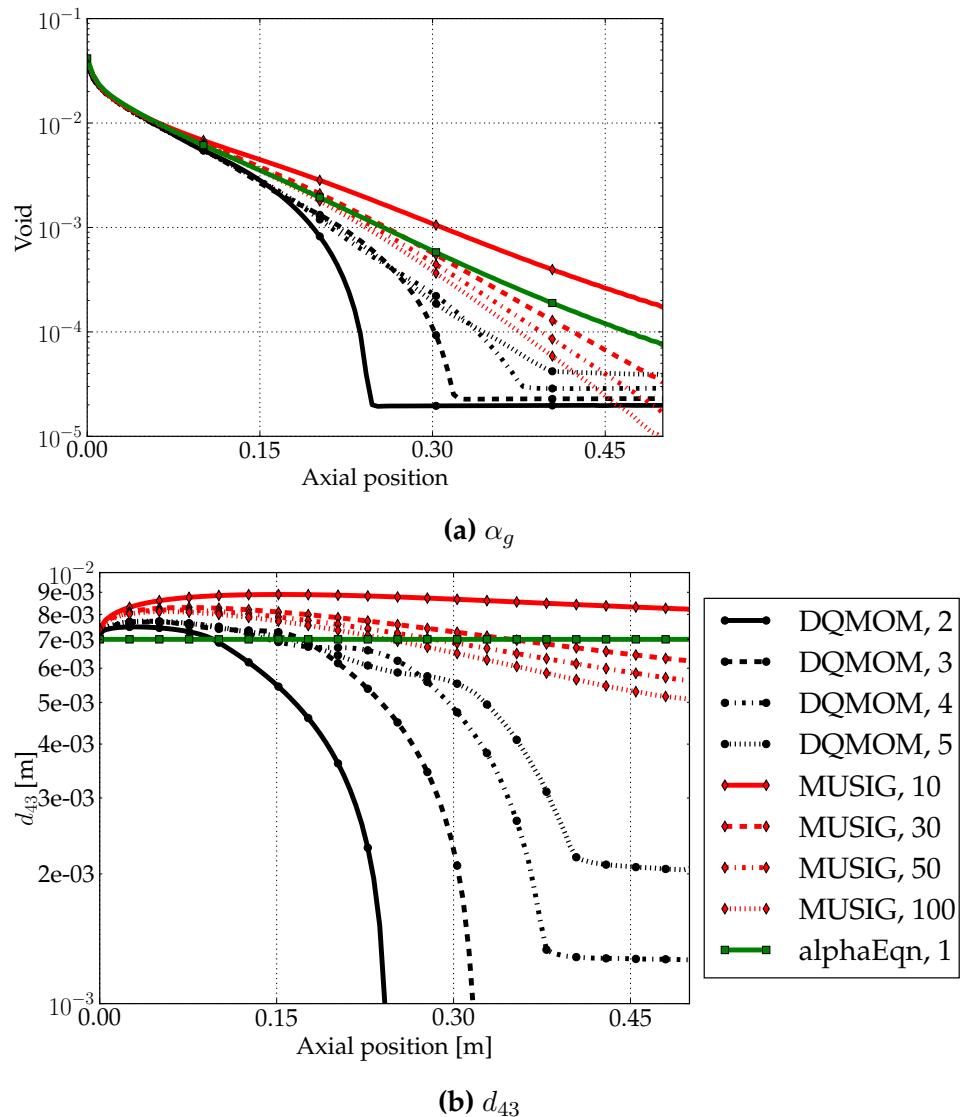
$$d_{43} = \frac{\sum_{i=1}^N \xi_i^4 w_i}{\sum_{i=1}^N \xi_i^3 w_i}. \quad (6.7)$$

As seen from the figure, the shrinkage of the bubbles accelerates over the channel, particularly visible for DQMOM, and the effect is explained by an inverse proportionality of the condensation rate with the bubble size. Due to the fixed sizes in the MUSIG method, a much smaller range of bubble sizes is covered and this was in Paper VI shown to be a severe limitation, further emphasized by other examples in the mentioned paper. It should be noted that both for the void fraction results and for the average size, DQMOM reach minima at around half the distance of the channel. For intricate reasons of the formulation, a threshold size of the abscissas is needed to avoid numerical issues. However, as seen from the figures, such a remedy occurs only at void fractions which are too low to be of physical significance.

In addition to the example above, different studies of bubbles inserted at the walls were conducted and a formulation useful for wall boiling models with DQMOM was proposed. Wall boiling is a particularly challenging topic as the insertion of bubbles takes place locally at the wall and in the discretized domain only in the first layer of cells.

Furthermore, under even more simplistic conditions, DQMOM was shown to converge for much fewer abscissas than the number of classes needed in MUSIG. For a case without the coupling to the two-fluid solver and with an initial mono-size distribution, MUSIG is shown to require more than 100 classes to capture the size change predicted with only a few abscissas. The difference in number of required sizes also results in a significant difference in the computational time which was, advantageously, shorter for DQMOM. However, for the examples coupled to the two-fluid solver, the differences in the computational effort are smaller. In practical terms MUSIG seems to have advantages in the simpler implementation and generally more stable characteristics, whereas the primary advantage of the DQMOM is the feature of dynamic abscissas which can cover very different ranges of bubble sizes in different parts of the domain.

As clear from the system description of the example case the intention in Paper VIII is neither to directly mimic the geometry nor do reconstruct the exact conditions in a subchannel in a BWR assembly. Instead, the purpose is to propose and evaluate a potential candidate for PBE simulations, which is also closely related to the second objective outlined in Section 1.4.2. In addition, the evaluation forced additional studies of the coupling between the two-fluid model and the PBE methods which are of future value for simulations dedicated to the actual BWR problem.



**Figure 6.3:** Vapor fraction (top) and average bubble size (bottom) along the axial centerline compared between MUSIG and DQMOM and for a void fraction equation (labeled alphaEqn) with a single static class. [Paper VI]

## 6.4 On the dynamics of the two-fluid formulation

As suggested in the introduction to the two-fluid model, the formulation is known to be prone to instabilities. In particular from the nuclear perspective, the 1D version of the conservation equations with no viscosity has been reported to have issues with such instabilities [161]. From a mathematical point of view, it is well known that the degree of hyperbolicity in the equations affects the stability and, for the same 1D formulation, mathematical regularization of the problem has been proposed as a remedy to achieve hyperbolicity [162]. Another approach often seen in the literature is numerical regularization, basically achieved by inducing numerical diffusion, either through the use of a coarse spatial discretization or in many papers, implicitly, by a diffusive turbulence model.

The potential instabilities are not only interesting for the 1D formulation, but also for the fine-mesh 3D approach in the current thesis. In particular, the influence of such instabilities on the dynamics of the two-fluid model is of interest. Although the phases are only predicted in an average sense, the dynamics of the fields are influential for mass and heat transfer applications. Additionally, it is of importance to understand any potential discrepancy between instabilities induced only by numerical issues and the heterogeneities actually predicted with experiments (such as in [180]). Numerical experiments in terms of 3D simulations with the two-fluid model have previously been studied from a stability perspective for gas-solid flows. It has been shown that such flows exhibit the so-called meso-scale instabilities, i.e. regions with fluctuations smaller than the physical domain but larger than the characteristic size of the particles [181].

From the literature on two-fluid simulations for gas-liquid problems it is clear that there are still a large number of open questions, and the formulations of e.g. momentum exchange mechanisms are basically as many as the authors. In addition, the treatment of turbulence for such systems is an area with a lot of proposed methods reflected in additional terms for the momentum conservation equations. A comprehensive discussion on such terms and on the lack of consensus is provided in Paper VIII.

In the work done for the thesis, the studies of the dynamics of the adiabatic two-fluid model are deliberately based on a much simplified model. The approaches reported in Papers III and VII are to prune the studied equations of any additional (unnecessary) terms and apply the model to simplistic cases. In Paper III, the two-fluid model according to eqs. (6.1) and (6.2) is simulated including only the drag term for the momentum exchange between the phases. For Paper VII the drag-only simulations are extended and compared to simulations also including the virtual mass force. In specific, the performed simulations are an attempt to understand the possibilities of resolving the dynamics of the gas-liquid flow with the two-fluid model. Additionally, it has been the goal to formulate, with physical or numerical arguments, criteria to discern numerical issues from instabilities of physical origin.

As regards the presented results, it should be noted that the two included papers are based on different CFD solvers (ANSYS Fluent and OpenFOAM®). Furthermore, variations of the implementation of the continuity equations was studied elsewhere [182]. In Paper III a 2D system was studied and the results from the paper are omitted here. Instead, the brief discussion in the next section is based on the results from the 3D system in Paper VIII.

### 6.4.1 Application to adiabatic cases

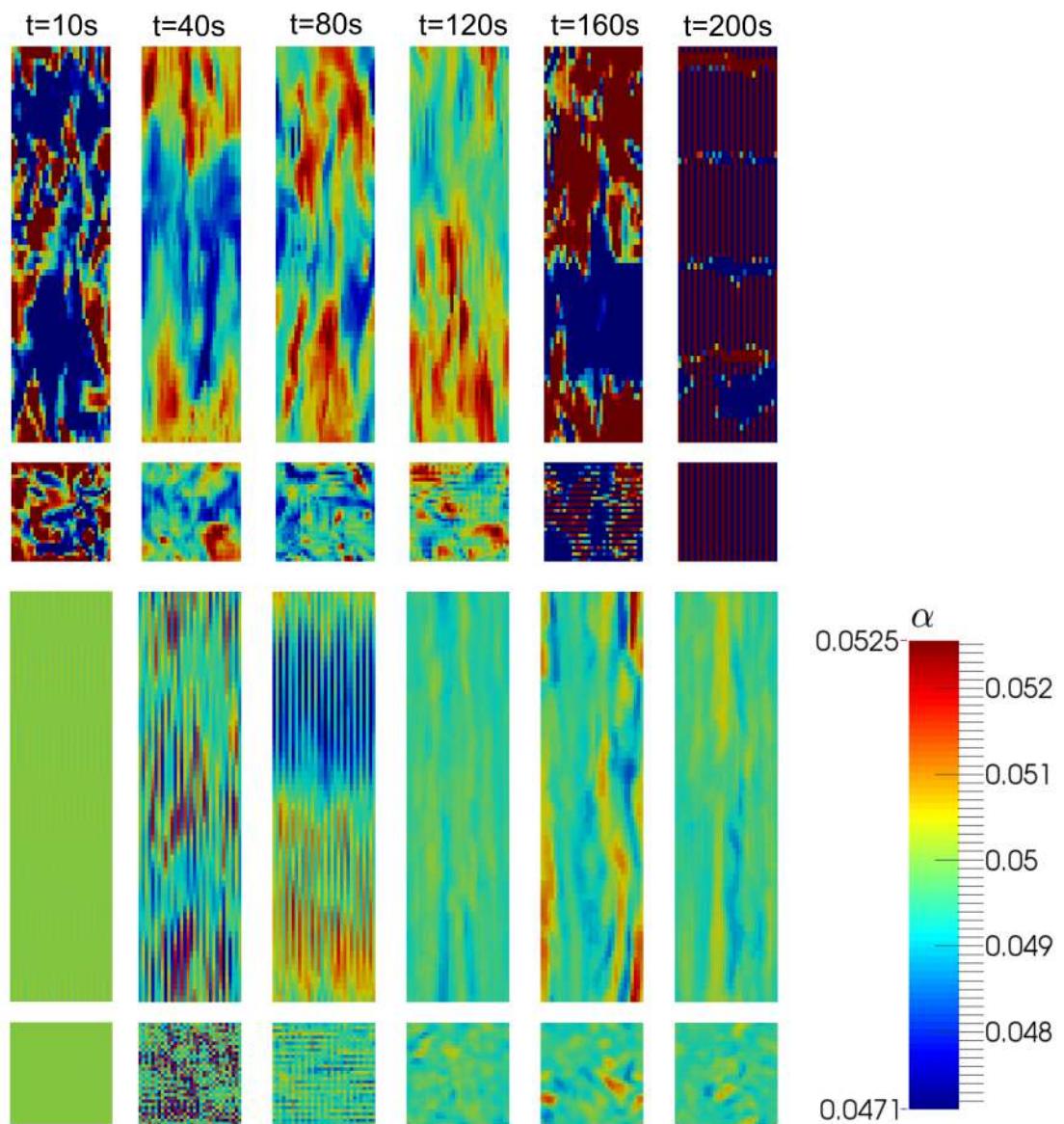
The cases studied are simple in the sense that only adiabatic conditions are considered and the simulated domain is periodic in all directions. The effect of gravity is included and to outweigh the combined mass of the gas and liquid phases a jump condition is applied for the pressure. The initial fields and the thermophysical properties are presented in Table 6.1. All fields are initiated with spatially uniform values, i.e. no initial perturbations are applied to induce the instabilities.

**Table 6.1:** Thermophysical parameters and initial conditions as applied in Paper VII.

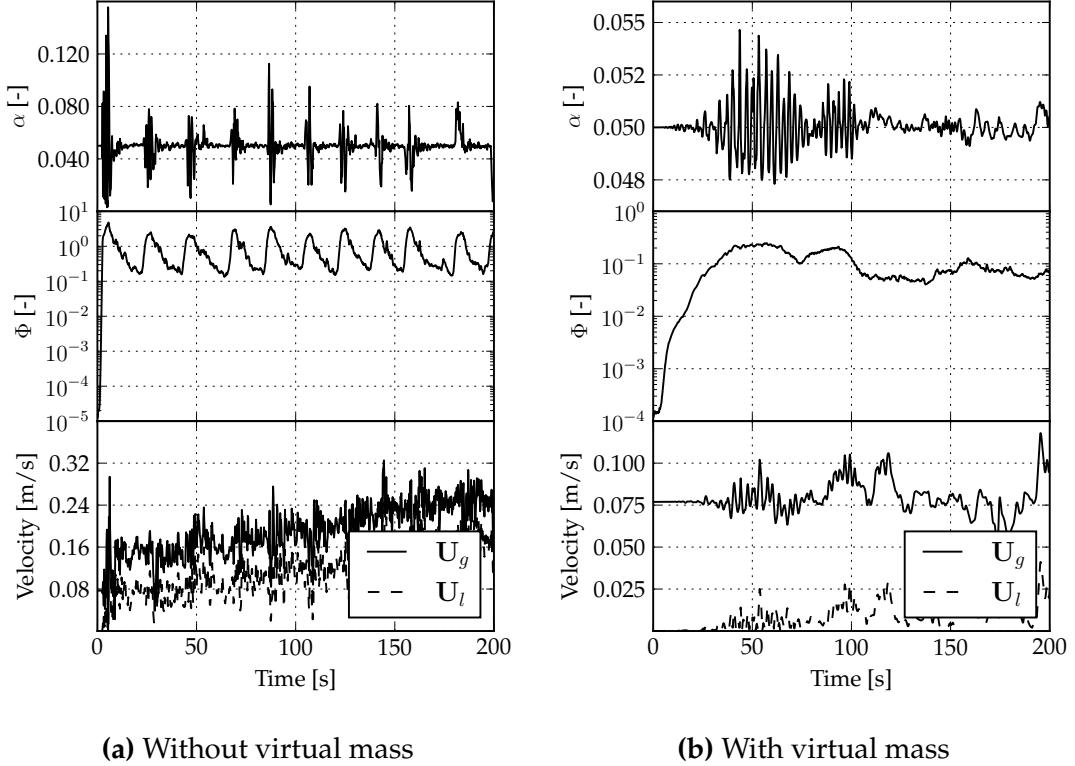
Liquid density, constant $\rho_l$	1000 kg/m <sup>3</sup>
Gas density, constant $\rho_g$	1 kg/m <sup>3</sup>
Liquid viscosity, constant $\mu_l$	$10^{-3}$ Pa s
Gas viscosity, constant $\mu_g$	$10^{-3}$ Pa s
Bubble size, constant $d_b$	0.68 mm
Void fraction, uniform initial condition $\alpha_g$	0.05
Liquid velocity, uniform initial condition	(0, 0, 0) m/s
Gas velocity, uniform initial condition	(0, 0, 0) m/s
Gauge pressure, uniform initial condition	0 Pa

The simulation is run for 200 s and snapshots of the void fraction distribution are presented for six chosen time steps in Figure 6.4. The figure presents a comparison between simulations performed with only the drag force (upper rows) or complemented with the virtual mass force (bottom rows). Considering first the results with only the drag force it is seen that the initially flat distribution of the void fraction has developed to a non-uniform state. For the results from  $t = 40$  s to  $t = 120$  s, a pattern with meso-scale structures is evident. However at  $t = 200$  s, a more disruptive void behavior is seen, with a checkerboard formation in the horizontal direction. In contrast, the results including the virtual mass force show initial strange patterns, perceived as being of a numerical character, whereas a more physically sound void fraction distribution is seen at the later time steps.

The results from Figure 6.4 are further examined in Figure 6.5, where the temporal development of the void fraction and the velocity field are presented to-



**Figure 6.4:** The gas fraction field displayed for 6 time steps (as indicated at the top of the figure) with the case of no virtual mass (top two rows) and virtual mass included (bottom two rows), displayed in the horizontal plane (rectangular figures) and the vertical plane (square figures). [Paper VII]



**Figure 6.5:** Temporal development of the void fraction field (top), the uniformity index (middle) and the magnitude of the velocity field for both phases (bottom). Both cases exhibit an initial transient in the void fraction and the uniformity index, although significantly faster for the case without the virtual mass force. The simulations are performed with the initial condition  $\alpha = 0.05$ . [Paper VII]

gether with the time-resolved uniformity index

$$\Phi(t) = \frac{\alpha_{g,\max} - \alpha_{g,\min}}{\alpha_{g,\text{ave}}}, \quad (6.8)$$

where  $\alpha_{g,\max}$  and  $\alpha_{g,\min}$  are the instantaneous maximum and minimum void fractions in the domain, respectively. The index is used as a measure of the global heterogeneity of the system. The average void fraction ( $\alpha_{g,\text{ave}}$ ) is directly given by the initial conditions for each of the simulations, and, for the presented results, a flat initial volumetric fraction of 5% gas is applied. The figure again shows a significant difference in the dynamics of the vapor fraction. For the cases with the drag only a repetitive instable behavior is visual. Whereas an initial rapid increase of  $\Phi$  is seen for the cases with the virtual mass, the later stage of the simulation follows a smoother behavior.

The results of the simulations are interesting from multiple perspectives. First, an initially homogeneous distribution of all fields quickly change and a variety of heterogeneous states is seen. The divergence from the uniform fields is interesting as there are no actual mechanisms in the equations forcing the change, and,

perhaps even more interesting as there are no mechanisms driving the system back to the smooth state. Second, the inclusion of the virtual mass force significantly changes the characteristics of the void fraction distribution. Although the actual magnitude of the force is small in comparison to the drag, the alteration of the behavior is distinct and seems to give a physically more plausible behavior. Third, it is interesting that the clean formulation, i.e. with no excessive diffusivity due to coarse turbulence modeling or additional momentum exchanges, exhibits clear instabilities. Such dynamic behavior would potentially be hidden with the inclusion of the mentioned type of terms, and arguably the latter would potentially hide the instabilities.

A key question raised in Paper VII is that of the trustworthiness of dynamic results from the two-fluid model. In detail, although the simulations might appear as sound and physical after the initial instabilities of numerical character (as was the case for the simulations including the virtual mass force), how are these results to be perceived? As argued in the paper, the quantitative values of the dynamic simulations are not immediately to be trusted which again emphasizes the complexities of the two-fluid model and the many questions marks yet to be resolved.

---

# CHAPTER 7

---

# Conclusions and recommendations for future work

Finally, the thesis is to be summarized and concluded. As the work was already split into two parts from the objectives point of view, the fine-mesh multiphysics and the two-phase flow studies, also the summary is presented in two parts. Last, I provide an outlook for the future of simulations for the coupled neutronic and thermal-hydraulic problem and how efforts like the one currently presented could be of importance for the nuclear industry in the future.

## 7.1 Fine-mesh multiphysics simulations

As the development of the multiphysics solver has been an integral part of the thesis, a summary of the methodology is first given. Second, a summary of the results is provided and accompanied by a conclusion on the achieved fidelity of the simulations.

### 7.1.1 Summary of the methodology

The presented methodology is aimed at fine-mesh simulations of the multiphysics problem of neutronics and thermal-hydraulics. The computational tool described in the thesis is based on the finite volume method and implemented in the open source framework OpenFOAM®. The thermal-hydraulic equations are solved by a CFD approach with segregated pressure and velocity solvers complemented by a RANS model for turbulence. The neutronic problem is handled by the multi-group diffusion equation (steady and transient simulations) and the discrete ordinates method (steady cases only), and solved with iterative, fixed-point, algorithms. The multiphysics couplings are handled in a Picard iteration style, with

sequential updates of each of the two modules. The multiphysics solver is parallelized based on the MPI implementation in OpenFOAM®, and with specific handling of the decomposition such that all overlapping cells of the meshes are kept at the same computational node.

The multiphysics solver is complemented by a utility for cross-section generation for sub-pin few group simulations. The tool is based on Serpent and handles a 2D fuel assembly geometry, specified in a configuration file. Similarly, a mesh tool is developed and to produce multi region body-fitted meshes for the thermal hydraulic problem, with resolved fuel, gap, cladding and moderator regions and a single, monolithic, mesh for the neutronics. The meshes are computed based on a block structure that gives a high level of user influence on the discretization.

### 7.1.2 Results and conclusions

The multiphysics tool was applied to both transient and steady simulations, where the results from the latter showed that:

- The fine-mesh simulations are able to provide the fuel and moderator temperature gradients on a sub-pin and resolved subchannel level with an equally high resolution neutronics solution on a quarter of a  $15 \times 15$  fuel assembly with PWR like thermophysical conditions.
- The convergence characteristic indicates that, generally, a few multiphysics iterations are required to resolve the couplings, whereas large number of sub-iterations are required for the neutronics and the thermal-hydraulics, respectively.
- The neutronics diffusion solver is, as expected, inferior to the  $S_N$  solver as shown on a two-dimensional case, validated against a Monte Carlo solution. The discrete ordinates method exhibits a significant ray effect which diminishes with the increasing order of the method.

As regards the transient simulations the results showed that:

- The fine-mesh simulation of a quarter of a  $7 \times 7$  fuel assembly captures the temporal development of the heterogeneities following a ramping transient at the inlet of the fuel assembly.
- The solver produces transient responses to local perturbations which could be verified with a novel approach based on the point-kinetic component of the system response.

In general, it can be concluded that all objectives were fulfilled for this part of the thesis. Nevertheless, there are also improvements to be developed (or implemented) for nearly every single aspect covered in the thesis and the solver methodology. To mention a few, it would be of interest and value to:

- Develop a transient version of the  $S_N$  solver. The steady-state validation results suggest that the diffusion approximation induces a significant error, not the least due to the highly resolved system and the resulting heterogeneous material regions. Clearly a transport method is required also for the transient cases. In addition, the  $S_N$  solver needs to be significantly accelerated, advisably based on a Krylov approach.
- Investigate the effect of a more detailed geometry, such as spacers. Such an effort would be even more interesting in combination with a LES approach to the turbulence to capture heterogeneities in the heat transfer and fluid thermo-physical state due to resolved fluctuations.
- Develop a multiphase version of the thermal-hydraulics method. As suggested in the introduction, the heterogeneities in the moderator are significantly larger for voided cases and thus a stronger fine-mesh coupling could be anticipated.
- Assess non-linear techniques to handle the couplings in the overall multi-physics problem as well as the separate modules. In particular, it would be interesting to implement the Anderson mixing methodology for the multi-physics problem and a more implicit approach to the pressure and velocity coupling.
- Add a multiscale methodology to compute boundary conditions for the fine-mesh simulations. Whereas the presented work was all based on periodic or symmetry boundary conditions in horizontal direction it would be of interest to investigate the effect of a more realistic environment of the simulated systems. To limit the computational effort a multiscale strategy would be a good candidate, simulating a hierarchy of scales in the same solver.

In addition, the implementation and the framework itself could be extended with other physics modules and with more generic simulation capabilities such as fluid-structure interaction or thermo-mechanical modules. Finally, it would be interesting to test the current approach for larger simulation domains, i.e. based on significantly larger computer resources.

## 7.2 Two-fluid simulations

The results and conclusions for the two-fluid simulations are separately reported for the DQMOM methodology and the dynamics investigations.

### 7.2.1 DQMOM coupled to a two-fluid solver

The presented methodology for subcooled boiling flows is based on a coupled PBE and two-fluid solver approach. The PBE is solved using DQMOM with a

newly proposed formulation for the condensation of bubbles. The method is compared to a MUSIG method, which in contrast to DQMOM applies a set of fixed bubble sizes.

The comparisons of the mentioned methodologies propose that:

- DQMOM needs significantly fewer discrete bubble sizes to reproduce the distribution as compared to the same accuracy as MUSIG. This further implies an edge in terms of shorter computational time required for DQMOM.
- DQMOM requires specific treatment for the extreme of very small bubbles, typical for subcooled flows where the bubbles grown at the wall fully condense in the bulk of the flow. Remedies in terms of regularization of the weights and the abscissas are shown for this purpose and successfully stabilize the solver.

In addition, the coupling to the two-fluid solver was carefully analyzed and, in particular, wall conditions for the insertion of bubbles were studied for both DQMOM and MUSIG.

As discussed in the analysis, the simulations presented in the thesis were not directly targeted to the BWR geometry or conditions. Instead, the study should be seen as an attempt to extend the field of CFD simulations for bubbly flows, and with particular focus on algorithms for the solution of PBEs. As a next step it would be of interest to investigate the benefits and drawbacks of the DQMOM method on a system more closely mimicking the subchannels in the reactor and of significantly larger size.

### 7.2.2 Two-fluid instability results

To investigate the dynamics of the two-fluid model, simulations based on adiabatic conditions were performed on fully periodic systems. The momentum exchange was based on the drag force and the virtual mass force only and no turbulence model was involved. The simulations showed that:

- The initially uniform void fraction distribution changed to a heterogeneous spatial distribution, exhibiting meso-scale structures.
- The inclusion of the virtual mass force stabilized the solver and resulted in a more physical character of the dynamics.

The analysis emphasized that the numerical character of the initial stage of the instabilities raises questions on the trustworthiness of such simulations and further illustrates the complexities of the two-fluid model. For future investigations it would be particularly interesting to evaluate the above proposed conclusions based on an entirely different methodology. In detail, to prove, or at least more firmly confirm, the existence of meso-scale structures, a comparison to a Lagrangian simulations framework should be a reasonable approach.

## 7.3 Future outlook

The core of a nuclear reactor is really an astonishing challenge from so many perspectives, and although high-fidelity multiphysics has been a hot topic in the last few years, much remains to be done.

For neutronics, the recent rapid development of the Monte Carlo solvers makes such types of methodologies important candidates for future tools. We have seen that there are many problematic aspects (not the least concerning the cross-section generation) which are completely eluded with the Monte Carlo approach and, although there are still gaps as regards transient simulations, this type of solver is likely to be an important component of future high-fidelity multiphysics tools.

As regards thermal-hydraulics, the areas of applications of CFD for reactor core simulations are likely to increase in number as well as in importance. The continuing growth of computational resources will successively allow for ever finer scales to be resolved and the application of high-resolution turbulence methods will for sure play an even more important role in future design of fuel assemblies and reactor cores. For multiphase CFD much theoretical work still remains, and, although, larger clusters can allow for more industrial use of interface tracking methodologies, full fine-mesh assembly simulations are for a long time still going to rely on averaged approaches, which are thus a continued important area for research.

Another future important question is that of validation of the novel multiphysics approaches. Although many of the suggested modules for the coupled tools can be separately validated, the community should aim at direct validation of the multiphysics solvers, in particular on fine-scales.



# Acknowledgements

The research performed in this thesis was financed by Svenskt Kärntekniskt Centrum (SKC) and the support is gratefully acknowledged. Additional grants for traveling were given by Åforsk, Kungliga Vetenskaps- och Vitterhets-Samhället (KVVS) and The Royal Swedish Academy of Science. The computations were performed on resources at Chalmers Centre for Computational Science and Engineering (C3SE) provided by the Swedish National Infrastructure for Computing (SNIC). Furthermore, the Swedish Research Council (Vetenskapsrådet) is acknowledged for the financing of the Development of Revolutionary and Accurate Methods for Safety Analyses of Future and Existing Reactors (DREAM4SAFER) framework grant (contract number C0467701), related to Papers III and VII in the thesis.

First, I would like to express my gratitudes to my supervisor Prof. Christophe Demaziere. Your long standing support and belief in all my ideas have been both an encouragement and privilege. I also thank my co-supervisors. Assoc. Prof. Paolo Vinai for your devotion to all my writing and your efforts on making them better. Prof. Srdjan Sasic, thank you for your guidance, encouragement and all insightful discussions on multiphase flow and more. Finally, I thank Prof. Imre Pazsit for being the examiner of the thesis.

I thank all colleagues I had the pleasure to work with during the past five years. Assistant Prof. Henrik Ström for interesting and fruitful work on the intricate two-fluid model. Sebastian Gonzalez-Pintor for insightful thoughts on neutronics and mathematics. Rasmus Andersson for the splendid Master thesis work. Assistant Prof. Gaetano Sardina for the DEM and two-fluid discussions. Prof. Håkan Nilsson for the opportunities of teaching and discussing OpenFOAM. Prof. Hrvoje Jasak for the inspiring talks on CFD and the opportunity to visit Zagreb. Vuko Vukcevic for inspiring discussions on coupled calculations. Ananda Subramani Kannan for the DEM discussions. Assoc. Prof. Julie Gold for the yearly doktorandsamtal.

Furthermore, I thank all my current and past colleagues at Nuclear Engineering for discussions, travel company, floorball and more. Thank you Daniel, Johan and Olov for the lunches. Thanks to all students I had the privilege to teach and

thanks to Assoc. Prof. Anders Nordlund for your support and encouragement in my teaching and the many interesting discussions. In addition, I thank my colleagues at Fraunhofer-Chalmers Research Centre for Industrial Mathematics for engaing work, inspiring discussions and motivation to get the PhD finished.

My warmest thanks to my family: my father for the almost daily encouraging discussions and all shared interests, my mother for the love and support, my sisters and their husbands for all enjoyable moments. Adam your support and discussions was more than once indispensable and your constant encouragement on the writing of this thesis was decisive.

Finally, thank you Noomi and Julia for the inspiration and help with getting priorities in life straight, and thank you Cornelia for your love, support and endless patience!

# Bibliography

- [1] ENDF/B-VII. Data extracted via EXFOR: Experimental Nuclear Reaction Data. URL: <https://www-nds.iaea.org/exfor/servlet/E4sGetTabSect?SectID=2292099&req=2031&PenSectID=7664758>.
- [2] J. Rhodes, K. Smith, D. Lee. "CASMO-5 development and applications". *Proc. ANS Topical Meeting on Reactor Physics (PHYSOR-2006)*. 2006, pp. 10–14.
- [3] D. Knott, A. Yamamoto. "Lattice Physics Computations". *Handbook of Nuclear Engineering*. Ed. by Dan Gabriel Cacuci. Springer US, 2010.
- [4] T. Bahadir, S.-Ö. Lindahl. "Studsviks next generation nodal code SIMULATE-5". *Advances in Nuclear Fuel Management IV (ANFM 2009)* (2009).
- [5] J. Hohorst, S. Polkinghorne, L. Siefken, C. Allison, C. Dobbe. *TMI-2 analysis using SCDAP/RELAP5/MOD3. 1*. Tech. rep. Lockheed Idaho Technologies Co., 1994.
- [6] A. Prosperetti, G. Tryggvason. *Computational Methods for Multiphase Flow*. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 2007.
- [7] R. K. Salko, M Avramova. "CTF theory manual". *The Pennsylvania State University* (2015).
- [8] M. Daeubler, A. Ivanov, B. L. Sjenitzer, V. Sanchez, R. Stieglitz, R. Macian-Juan. "High-fidelity coupled Monte Carlo neutron transport and thermal-hydraulic simulations using Serpent 2/SUBCHANFLOW". *Annals of Nuclear Energy* 83 (2015), pp. 352 –375.
- [9] C. Brennen. *Thermo-Hydraulics of Nuclear Reactors*. Dankat Publishing Company, 2013.
- [10] K. Ivanov, M. Avramova. "Challenges in coupled thermal-hydraulics and neutronics simulations for LWR safety analysis". *Annals of Nuclear Energy* 34.6 (2007), pp. 501–513.
- [11] IAEA. *Best estimate safety analysis for nuclear power plants: Uncertainty evaluation*. Tech. rep. IAEA, Vienna, 2008.
- [12] U. Imke, V. H. Sanchez. "Validation of the subchannel code SUBCHANFLOW using the NUPEC PWR tests (PSBT)". *Science and Technology of Nuclear Installations* (2012).

## Bibliography

---

- [13] D. Bestion. "From the Direct Numerical Simulation to System Codes-Perspective for the Multi-scale Analysis of LWR Thermalhydraulics". *Nuclear Engineering and Technology* 42.6 (2010), pp. 608–619.
- [14] J. C. Ragusa, V. S. Mahadevan. "Consistent and accurate schemes for coupled neutronics thermal-hydraulics reactor analysis". *Nuclear Engineering and Design* 239.3 (2009), pp. 566–579.
- [15] M. Ellis, J Watson, K Ivanov. "Progress in the development of an implicit steady state solution in the coupled code TRACE/PARCS". *Progress in Nuclear Energy* 66 (2013), pp. 1–12.
- [16] M. E. Conner, E. Baglietto, A. M. Elmahdi. "CFD methodology and validation for single-phase flow in PWR fuel assemblies". *Nuclear Engineering and Design* 240.9 (2010), pp. 2088–2095.
- [17] F. Jatuff, F. Giust, J. Krouthen, S. Helmersson, R. Chawla. "Effects of void uncertainties on the void reactivity coefficient and pin power distributions for a 10x10 BWR assembly". *Ann. Nucl. Energy* 33 (2006), pp. 119–125.
- [18] T. Ikehara, Y. Kudo, M. Tamitani, M. Yamamoto. "Effect of Subchannel Void Fraction Distribution on Lattice Physics Parameters for Boiling Water Reactor Fuel Bundles". *J. Nucl. Sci. Technol.* 45.12 (2008), pp. 1237–1251.
- [19] D. R. Gaston. "Physics-based multiscale coupling for full core nuclear reactor simulation". *Annals of Nuclear Energy* 84 (2015), pp. 45–54.
- [20] V. Petrov, B. K. Kendrick, D. Walter, A. Manera, J. Secker. "Prediction of CRUD deposition on PWR fuel using a state-of-the-art CFD-based multi-physics computational tool". *Nuclear Engineering and Design* 299 (2016), pp. 95–104.
- [21] CASL. *Successful expanded prediction of the nature of CRUD found in pressurized water reactor coolant*. (last visited June 19, 2017). URL: <http://www.casl.gov/highlights/crud.shtml>.
- [22] J. Bakosi, M. A. Christon, R. B. Lowrie, L. Pritchett-Sheats, R. Nourgaliev. "Large-eddy simulations of turbulent flow for grid-to-rod fretting in nuclear reactors". *Nuclear Engineering and Design* 262 (2013), pp. 544–561.
- [23] U. Bieder, F. Falk, G. Fauchet. "LES analysis of the flow in a simplified PWR assembly with mixing grid". *Progress in Nuclear Energy* 75 (2014), pp. 15–24.
- [24] E. Merzari, A. Obabko, P. Fischer, N. Halford, J. Walker, A. Siegel, Y. Yu. "Large-scale large eddy simulation of nuclear reactor flows: Issues and perspectives". *Nuclear Engineering and Design* 312 (2017), pp. 86–98.
- [25] J. Cardoni, Rizwan-uddin. "Nuclear Reactor Multi-Physics Simulations with Coupled MCNP5 and STAR-CCM+". *M&C 2011, Rio de Janeiro, Brazil* (2011).
- [26] L. Li, K. Wang. "The first-principle coupled calculations using TMCC and CFX for the pin-wise simulation of LWR". *Proc. Int. Conf. PHYSOR, American Nuclear Society, Knoxville, Tennessee, USA*. 2012.
- [27] A Ivanov, V Sanchez, R Stieglitz, K Ivanov. "High fidelity simulation of conventional and innovative LWR with the coupled Monte-Carlo thermal-hydraulic system MCNP-SUBCHANFLOW". *Nuclear Engineering and Design* 262 (2013), pp. 264–275.

- [28] D. P. Weber. "High-Fidelity Light Water Reactor Analysis with the Numerical Nuclear Reactor". *Nucl. Sci. Eng.* 155.155 (2006), pp. 395–408.
- [29] J. Yan, B. Kochunas, M. Hursin, T. Downar, Z. Karoutas, E. Baglietto. "Coupled Computational Fluid Dynamics and MOC Neutronic Simulations of Westinghouse PWR Fuel Assemblies with Grid Spacers". *NURETH-14, Toronto, Ontario, Canada, September 25-30* (2011).
- [30] B. Kochunas, S. Stimpson, B. Collins, T. Downar, R. Brewster, E. Baglietto, J. Yan. "Coupled Full Core Neutron Transport/CFD Simulations of Pressurized Water Reactors". *PHYSOR 2012, Knoxville, Tennessee, USA, April 15-20* (2012).
- [31] R. Schmidt. "An approach for coupled-code multiphysics core simulations from a common input". *Annals of Nuclear Energy* 84 (2015), pp. 140–152.
- [32] D. Gaston, C. Newman, G. Hansen, D. Lebrun-Grandjean. "MOOSE: A parallel computational framework for coupled systems of nonlinear equations". *Nuclear Engineering and Design* 239.10 (2009), pp. 1768–1778.
- [33] J. Bell, B. Lewis. "CANDU fuel bundle deformation modelling with COMSOL multiphysics". *Nuclear Engineering and Design* 250 (2012), pp. 134–141.
- [34] R. Liu, A. Prudil, W. Zhou, P. K. Chan. "Multiphysics coupled modeling of light water reactor fuel performance". *Progress in Nuclear Energy* 91 (2016), pp. 38–48.
- [35] C. Fiorina, I. Clifford, M. Aufiero, K. Mikityuk. "GeN-Foam: a novel OpenFOAM® based multi-physics solver for 2D/3D transient analysis of nuclear reactors". *Nuclear Engineering and Design* 294 (2015), pp. 24–37.
- [36] C. Newman, G. Hansen, D. Gaston. "Three dimensional coupled simulation of thermomechanics, heat, and oxygen diffusion in UO<sub>2</sub> nuclear fuel rods". *Journal of Nuclear Materials* 392.1 (2009), pp. 6–15.
- [37] J. A. Turner, K. Clarno, M. Sieger, R. Bartlett, B. Collins, R. Pawlowski, R. Schmidt, R. Summers. "The Virtual Environment for Reactor Applications (VERA): Design and architecture". *Journal of Computational Physics* 326 (2016), pp. 544–568.
- [38] M. A. Christon, R. Lu, J. Bakosi, B. T. Nadiga, Z. Karoutas, M. Berndt. "Large-eddy simulation, fuel rod vibration and grid-to-rod fretting in pressurized water reactors". *Journal of Computational Physics* 322 (2016), pp. 142–161.
- [39] J. R. Lee, J. Kim, C.-H. Song. "Synthesis of the turbulent mixing in a rod bundle with vaned spacer grids based on the OECD-KAERI CFD benchmark exercise". *Nuclear Engineering and Design* 279 (2014), pp. 3–18.
- [40] S. Lo, J. Osman. "CFD modeling of boiling flow in PSBT 5 5 bundle". *Science and Technology of Nuclear Installations* (2012).
- [41] T. Sofu, J. Thomas, D. Weber, W. Pointer. "Coupled BWR Calculations with the Numerical Nuclear Reactor Software System". *Joint International Topical Meeting on Mathematics & Computation and Supercomputing in Nuclear Applications*. 2007.
- [42] D. Bestion. "The difficult challenge of a two-phase {CFD} modelling for all flow regimes". *Nuclear Engineering and Design* 279 (2014). {SI} : CFD4NRS-4, pp. 116–125.
- [43] *Univac LARC PROGRAMMING: The Computing Unit*. Remington Rand Univac. 1961.

## Bibliography

---

- [44] *CDC6600*. (last visited June 19, 2017). Wikipedia. URL: [https://en.wikipedia.org/wiki/CDC\\_6600](https://en.wikipedia.org/wiki/CDC_6600).
- [45] D. Loshin. "Chapter 2 - History". *High Performance Computing Demystified*. Ed. by David Loshin. Academic Press, 1994, pp. 13–21.
- [46] *Blue Gene*. (last visited June 19, 2017). Wikipedia. URL: [https://en.wikipedia.org/wiki/Blue\\_Gene](https://en.wikipedia.org/wiki/Blue_Gene).
- [47] P. J. Turinsky. "Advances in multi-physics and high performance computing in support of nuclear reactor power systems modeling and simulation". *Nuclear Engineering and Technology* 44.2 (2012), pp. 103–122.
- [48] H. Sutter. *The Free Lunch Is Over: A Fundamental Turn Toward Concurrency in Software*. (last visited March 3, 2017). Dr. Dobb's Journal, 2005. URL: <http://www.gotw.ca/publications/concurrency-ddj.htm>.
- [49] A. Danowitz, K. Kelley, J. Mao, J. P. Stevenson, M. Horowitz. "CPU DB: Recording Microprocessor History". *Commun. ACM* 55.4 (Apr. 2012), pp. 55–63.
- [50] C. DB. (last visited June 19, 2017). Stanford VLSI Group. URL: <http://cpudb.stanford.edu/processors>.
- [51] T. S. Sites. *Performance Development*. (last visited June 19, 2017). URL: <https://www.top500.org/statistics/perfdevel/>.
- [52] *Intel Math Kernel Library Benchmarks*. Linux package `1_mklb-p_2017.2.015`. URL: <https://software.intel.com/en-us/articles/intel-mkl-benchmarks-suite>.
- [53] J. Demmel, M. Hoemmen, M. Mohiyuddin, K. Yelick. "Avoiding communication in sparse matrix computations". *Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on*. IEEE. 2008, pp. 1–12.
- [54] C. Lomont. *Introduction to Intel Advanced Vector Extensions*, June 2011.
- [55] K. Yukhin. *Intel Advanced Vector Extensions 2015/2016 Support in GNU Compiler Collection. GNU Tools Cauldron 2014*. (last visited June 19, 2017). Intel. 2014. URL: [https://gcc.gnu.org/wiki/cauldron2014?action=AttachFile&do=get&target=Cauldron14\\_AVX-512\\_Vector\\_ISA\\_Kirill\\_Yukhin\\_20140711.pdf](https://gcc.gnu.org/wiki/cauldron2014?action=AttachFile&do=get&target=Cauldron14_AVX-512_Vector_ISA_Kirill_Yukhin_20140711.pdf) (visited on 04/21/2017).
- [56] N. CUDA. *CUDA Zone*. 2017.
- [57] C. Calvin, D. Nowak. "High Performance Computing in Nuclear Engineering". *Handbook of Nuclear Engineering*. Ed. by Dan Gabriel Cacuci. Springer US, 2010.
- [58] MPI Forum. *MPI Forum*. (last visited June 19, 2017). URL: <http://mpi-forum.org/>.
- [59] SIEMENS MDX. *STAR-CCM+*. (last visited June 19, 2017). URL: <http://mdx.plm.automation.siemens.com/star-ccm-plus>.
- [60] ANSYS. *ANSYS Fluent: CFD Simulation*. (last visited June 19, 2017). URL: <http://www.ansys.com/Products/Fluids/ANSYS-Fluent>.
- [61] Gerris. *Gerris Flow Solver*. (last visited June 19, 2017). Apr. 13, 2017. URL: <http://gfs.sourceforge.net/>.

- [62] OpenFOAM Foundation. *The OpenFOAM Foundation*. (last visited June 19, 2017). URL: <https://openfoam.org/>.
- [63] Multiphase Flow Sciences Group at NETL. *MFIX*. (last visited June 19, 2017). URL: <https://mfix.netl.doe.gov/>.
- [64] F. Harlow, J. Fromm. "Dynamics and heat transfer in the von Krmn wake of a rectangular cylinder". *Physics of Fluids* 7.8 (1964), pp. 1147–1156.
- [65] J. H. Ferziger, M. Peric. *Computational methods for fluid dynamics*. 1997.
- [66] COMSOL. *COMSOL CFD Module*. (last visited June 19, 2017). 2017. URL: <https://www.comsol.se/cfd-module>.
- [67] E. Dick. "Introduction to Finite Element Methods in Computational Fluid Dynamics". *Computational Fluid Dynamics*. Ed. by John F. Wendt. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 235–274.
- [68] A. Hébert. *Multigroup Neutron Transport and Diffusion Computations*. Springer-Verlag, 2010, pp. 751–911.
- [69] S. Patankar. *Numerical heat transfer and fluid flow*. CRC press, 1980.
- [70] H Jasak. "Error analysis and estimation in the finite volume method with applications to 1027 fluid flows". PhD thesis. PhD thesis, Imperial College, University of London. 1028, 1996.
- [71] F. Bassi, S. Rebay. "High-order accurate discontinuous finite element solution of the 2D Euler equations". *Journal of computational physics* 138.2 (1997), pp. 251–285.
- [72] Y. Saad, H. A. Van Der Vorst. "Iterative solution of linear systems in the 20th century". *Journal of Computational and Applied Mathematics* 123.1 (2000), pp. 1–33.
- [73] Y. Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.
- [74] Y. Shapira. *Matrix-based multigrid: theory and applications*. Vol. 2. Springer Science & Business Media, 2008.
- [75] H. G. Weller, G Tabor, H. Jasak, C Fureby. "A tensorial approach to computational continuum mechanics using object-oriented techniques". *Computers in physics* 12.6 (1998), pp. 620–631.
- [76] Wikki. *foam-extend-3.1*. (last visited February 16, 2015). 2015. URL: <http://wikki.gridcore.se/foam-extend/foam-extend-3-1-zagreb>.
- [77] OpenFOAM. *OpenFOAM User's Guide. User's Guide*. Version 2.1.1. OpenCFD Limited. OpenFOAM Foundation, 2012.
- [78] OpenFOAM. *OpenFOAM Programmer's Guide. Programmers Guide*. Version 2.0.0. OpenCFD Limited. OpenFOAM Foundation, 2011.
- [79] T. Maric, J. Höpken, K. Mooney. *The OpenFOAM Technology Primer*. Stan Mott, 2014.
- [80] F Moukalled, L Mangani, M Darwish. *The finite volume method in computational fluid dynamics*. Springer, 2016.
- [81] S. Patankar, D. Spalding. "A calculation procedure for heat, mass and momentum transfer in three dimensional parabolic flows". *International Journal of Heat and Mass Transfer* 15 (1972), pp. 1787–1806.

## Bibliography

---

- [82] R. Issa, B Ahmadi-Befrui, K. Beshay, A. Gosman. "Solution of the implicitly discretised reacting flow equations by operator-splitting". *Journal of Computational Physics* 93.2 (1991), pp. 388 –410.
- [83] C. Rhie, W. Chow. "A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation". *AIAA Journal* 21 (1983), pp. 1525–1532.
- [84] J. Ahrens, B. Geveci, C. Law, C. Hansen, C. Johnson. "36-ParaView: An End-User Tool for Large-Data Visualization". *The Visualization Handbook* (2005), p. 717.
- [85] F Juretić, A. Gosman. "Error analysis of the finite-volume method with respect to mesh type". *Numerical heat transfer, part B: fundamentals* 57.6 (2010), pp. 414–439.
- [86] H. Jasak, A. Gosman. "Automatic resolution control for the finite-volume method, Part 2: Adaptive mesh refinement and coarsening". *Numerical Heat Transfer: Part B: Fundamentals* 38.3 (2000), pp. 257–271.
- [87] G. Hansen, S. Owen. "Mesh generation technology for nuclear reactor simulation; barriers and opportunities". *Nuclear Engineering and Design* 238.10 (2008), pp. 2590–2605.
- [88] T. J. Tautges, R. Jain. "Creating geometry and mesh models for nuclear reactor core geometries using a lattice hierarchy-based approach". *Engineering with Computers* 28.4 (Oct. 1, 2012), p. 319.
- [89] K. Podila, Y. Rao. "{CFD} modelling of turbulent flows through 5 5 fuel rod bundles with spacer-grids". *Annals of Nuclear Energy* 97 (2016), pp. 86 –95.
- [90] SALOME Platform. 2017. URL: <http://www.salome-platform.org/>.
- [91] C. Chauliac, J.-M. Aragonés, D. Bestion, D. G. Cacuci, N. Crouzet, F.-P. Weiss, M. A. Zimmermann. "NURESIM–A European simulation platform for nuclear reactor safety: Multi-scale and multi-physics calculations, sensitivity and uncertainty analysis". *Nuclear Engineering and Design* 241.9 (2011), pp. 3416–3426.
- [92] A Rubin, A Schoedel, M Avramova, H Utsuno, S Bajorek, A Velazquez-Lozada. "OECD/NRC Benchmark based on NUPEC PWR subchannel and bundle tests (PSBT), Volume I: Experimental Database and Final Problem Specifications". *US NRC OECD Nuclear Energy Agency* (2010).
- [93] ANSYS® ICEM. *ANSYS ICEM CFD Tutarial Manual*. 2013.
- [94] D. Lebrun-Grandie, J. C. Ragusa, R. Sampath. "Efficient finite element field interpolation for multiphysics applications". *PHYSOR 2014, September 28 - October 3, Japan* (2014).
- [95] Studsvik. *CASMO 4e Manual*. 2009.
- [96] G. Marleau, A. Hebert, R. Roy. *A User Guide for DRAGON*. 2013.
- [97] Los Alamos National Laboratory. *MCNP*. (last visited June 19, 2017). Apr. 20, 2017. URL: <https://mcnp.lanl.gov/>.
- [98] J. E. Hoogenboom, V. A. Khotylev, J. M. Tholammakkil. "Generation of multi-group cross sections and scattering matrices with the Monte Carlo code MCNP5". *Joint International Topical Meeting on Mathematics & Computation and Supercomputing in Nuclear Applications*. 2007, pp. 15–19.

- [99] J. Kuijper, S. Van Der Marck, A Hogenbirk. "Using homogenized macroscopic group cross sections in continuous-energy Monte Carlo neutron transport calculations with MCNP". *proc. M&C+ SNA* (2007), pp. 15–19.
- [100] E. Pettersen, C. Demazière, K. Jareteg, E. Schönfeldt T, B. Lauritzen. "Development of a Monte-Carlo based method for calculating the effect of stationary fluctuations". *M&C 2015, Nashville, Tennessee* (2015).
- [101] J. Leppänen, M. Pusa, T. Viitanen, V. Valtavirta, T. Kaltiaisenaho. "The Serpent Monte Carlo code: Status, development and applications in 2013". *Annals of Nuclear Energy* 82 (2015), pp. 142–150.
- [102] C. Demazière. "Investigation of the bias coming from spectrum corrections in the simulations of nuclear reactor transients". Vol. 6. 2016, pp. 3561–3570.
- [103] M. Aufiero, C. Fiorina, A. Laureau, P. Rubiolo, V. Valtavirta. "Serpent–OpenFOAM coupling in transient mode: simulation of a Godiva prompt critical burst". *Proceedings of M&C+ SNA+ MC* (2015), pp. 19–23.
- [104] V Valtavirta, T Ikonen, T Viitanen, J Leppänen. *Simulating fast transients with fuel behavior feedback using the Serpent 2 Monte Carlo code*. Tech. rep. 2015.
- [105] B. L. Sjenitzer, J. E. Hoogenboom. "A Monte Carlo method for calculation on the dynamic behaviour of nuclear reactors". *Progress in Nuclear Science and Technology* 2 (2011), pp. 716–721.
- [106] A. K. Prinja, E. W. Larsen. "General Principles of Neutron Transport". *Handbook of Nuclear Engineering*. Ed. by Dan Gabriel Cacuci. Springer US, 2010, pp. 427–542.
- [107] G. I. Bell, S. Glasstone. *Nuclear reactor theory*. Vol. 252. Van Nostrand Reinhold New York, 1970.
- [108] E. W. Larsen, J. E. Morel. "Advances in discrete-ordinates methodology". *Nuclear Computational Science*. Springer, 2010, pp. 1–84.
- [109] C. Demazière. *Modelling of nuclear reactors*. Division of Nuclear Engineering, Department of Applied Physics, Chalmers University of Techonology, 2011.
- [110] I. K. Abu-Shumays. "Angular Quadrature for Improved Transport Computations". *Transport Theory and Statistical Physics* 30.2-3 (2001), pp. 169–204.
- [111] M. L. Adams, E. W. Larsen. "Fast iterative methods for discrete-ordinates particle transport calculations". *Progress in nuclear energy* 40.1 (2002), pp. 3–159.
- [112] E. W. Larsen. "Diffusion-synthetic acceleration methods for discrete-ordinates problems". *Transport Theory and Statistical Physics* 13.1-2 (1984), pp. 107–126.
- [113] J. S. Warsa, T. A. Wareing, J. E. Morel. "Krylov iterative methods and the degraded effectiveness of diffusion synthetic acceleration for multidimensional S n calculations in problems with material discontinuities". *Nuclear science and engineering* 147.3 (2004), pp. 218–248.
- [114] R. Panton. *Incompressible flow*. Third edition. Wiley, 2005.
- [115] J. C. Tannehill, D. A. Anderson, R. H. Pletcher. "Computational fluid dynamics and heat transfer". *Series in Computational and Physical Processes in Mechanics and Thermal Sciences* (1997).
- [116] S. B. Pope. *Turbulent flows*. 2001.

## Bibliography

---

- [117] S. Benhamadouche. "On the use of (U)RANS and {LES} approaches for turbulent incompressible single phase flows in nuclear engineering applications". *Nuclear Engineering and Design* 312 (2017). 16th International Topical Meeting on Nuclear Reactor Thermal Hydraulics, pp. 2–11.
- [118] H. K. Versteeg, W. Malalasekera. *An introduction to computational fluid dynamics: the finite volume method*. Pearson Education, 2007.
- [119] L. Davidson. "Fluid mechanics, turbulent flow and turbulence modeling". *Chalmers University of Technology, Goteborg, Sweden (Mar 2017)* (2017).
- [120] H. Elman. "Preconditioning Strategies for Models of Incompressible Flow". English. *Journal of Scientific Computing* 25.1 (2005), pp. 347–366.
- [121] E. Brakkee, P. Wilders. *A domain decomposition method for the advection-diffusion equation*. Tech. rep. Technische Universiteit Delft, 1994.
- [122] P. Lions. "On the Schwarz alternating method. III: a variant for non-overlapping domains". *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations, Houston, Texas, March 20-22*, (1990).
- [123] M. Modest. *Radiative Heat Transfer*. Academic Press, San Diego, USA, 2003.
- [124] K. Jareteg. *Development of an integrated deterministic neutronic/thermal-hydraulic model using a CFD solver*. M.Sc. thesis, Chalmers University of Technology. 2012.
- [125] M. Berrill, K. Clarno, S. Hamilton, R. Pawlowski. *Evaluation of Coupling Approaches*. Tech. rep. CASL-U-2014-0081-000, Oak Ridge National Laboratory, 2014.
- [126] D. A. Knoll, D. E. Keyes. "Jacobian-free Newton–Krylov methods: a survey of approaches and applications". *Journal of Computational Physics* 193.2 (2004), pp. 357–397.
- [127] M. A. Pope, V. A. Mousseau. "Accuracy and efficiency of a coupled neutronics and thermal hydraulics model". *Nuclear Engineering and Technology* 41.7 (2009), pp. 885–892.
- [128] D. G. Anderson. "Iterative procedures for nonlinear integral equations". *Journal of the ACM (JACM)* 12.4 (1965), pp. 547–560.
- [129] A. Toth, C. Kelley, S. Slattery, S. Hamilton, K. Clarno, R Pawlowski. "Analysis of Anderson acceleration on a simplified neutronics/thermal hydraulics system". *Proceedings of MC2015, Nashville, USA* (2015).
- [130] M. T. Calef, E. D. Fichtl, J. S. Warsa, M. Berndt, N. N. Carlson. "Nonlinear Krylov acceleration applied to a discrete ordinates formulation of the k-eigenvalue problem". *Journal of Computational Physics* 238 (2013), pp. 188–209.
- [131] J. Willert, H Park, W. Taitano. "Using Anderson acceleration to accelerate the convergence of neutron transport calculations with anisotropic scattering". *Nuclear Science and Engineering* 181.3 (2015), pp. 342–350.
- [132] W. Liu, L. Zhang, Y. Zhong, Y. Wang, Y. Che, C. Xu, X. Cheng. "Cfd high-order accurate scheme jacobian-free newton krylov method". *Computers & Fluids* 110 (2015), pp. 43–47.
- [133] S. Schunert. "A flexible nonlinear diffusion acceleration method for the S N transport equations discretized with discontinuous finite elements". *Journal of Computational Physics* 338 (2017), pp. 107–136.

- [134] G. Kim, S. Kim, Y. Kim. "Parallelized Unstructured-Grid Finite Volume Method for Modeling Radiative Heat Transfer". *Journal of Mechanical Science and Technology* 19 (2005), pp. 1006–1017.
- [135] S. Plimpton, B. Hendrickson, S. Burns, W. McLendon, L. Rauchwerger. "Parallel Sn Sweeps on Unstructured Grids: Algorithms for Prioritization, Grid Partitioning and Cycle Detection". *Nuclear Science and Engineering* 150 (2005), pp. 267–283.
- [136] Y. Saad. *Numerical Methods for Large Eigenvalue Problems: Revised Edition*. SIAM, 2011.
- [137] R. Stammier, M. Abbate. *Methods of steady state reactor physics in nuclear design*. Academic Press, London, 1983.
- [138] E. Lewis, W. J. Miller. *Computational Methods of Neutron Transport*. John Wiley & Sons, Inc., USA, 1984.
- [139] M. DeHart. "A discrete ordinates approximation to the neutron transport equation applied to generalized geometries". PhD thesis. Texas A&M University, 1992.
- [140] D Ginestar, G Verdú, V Vidal, R Bru, J Marín, J. Munoz-Cobo. "High order backward discretization of the neutron diffusion equation". *Annals of Nuclear Energy* 25.1-3 (1998), pp. 47–64.
- [141] N. Todreas, M. Kazimi. *Nuclear systems I: Thermal hydraulic fundamentals*. Taylor & Francis, Levittown, USA, 1993.
- [142] R. Nigmatulin. "Spatial averaging in the mechanics of heterogeneous and dispersed systems". *International Journal of Multiphase Flow* 5.5 (1979), pp. 353–385.
- [143] R. Lahey, D. Drew. "The three-dimensional time and volume averaged conservation equations of two-phase flow". *Advances in Nuclear Science & Technology* (1989).
- [144] G Yadigaroglu. "CMFD and the critical-heat-flux grand challenge in nuclear thermal-hydraulics–A letter to the Editor of this special issue". *International Journal of Multiphase Flow* 67 (2014), pp. 3–12.
- [145] Y. Sato, B. Niceno. "A depletable micro-layer model for nucleate pool boiling". *Journal of Computational Physics* 300 (2015), pp. 20–52.
- [146] G Yadigaroglu. "CMFD (a brand name) and other acronyms". *International Journal of Multiphase Flow* 29.4 (2003), pp. 719–720.
- [147] G Tryggvason, J Lu. "DNS for multiphase flow model generation and validation". *NURETH-14, Toronto, Ontario, Canada, September 25-30* (2011).
- [148] N. Deen, J. Kuipers. "Direct Numerical Simulation (DNS) of mass, momentum and heat transfer in dense fluid-particle systems". *Current Opinion in Chemical Engineering* 5 (2014), pp. 84–89.
- [149] W. F. Noh, P. Woodward. "SLIC (simple line interface calculation)". *Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics June 28–July 2, 1976 Twente University, Enschede*. Springer. 1976, pp. 330–340.
- [150] C. W. Hirt, B. D. Nichols. "Volume of fluid (VOF) method for the dynamics of free boundaries". *Journal of Computational Physics* 39.1 (1981), pp. 201–225.

## Bibliography

---

- [151] S. Osher, J. A. Sethian. "Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations". *Journal of Computational Physics* 79.1 (1988), pp. 12–49.
- [152] M. Sussman, P. Smereka, S. Osher. "A level set approach for computing solutions to incompressible two-phase flow". *Journal of Computational physics* 114.1 (1994), pp. 146–159.
- [153] S. Unverdi, G. Tryggvason. "A front-tracking method for viscous, incompressible, multi-fluid flows". *Journal of computational physics* 100.1 (1992), pp. 25–37.
- [154] Q. Zeng, J. Cai, H. Yin, X. Yang, T. Watanabe. "Numerical simulation of single bubble condensation in subcooled flow using OpenFOAM". *Progress in Nuclear Energy* 83 (2015), pp. 336–346.
- [155] N. Samkhaniani, M. Ansari. "Numerical simulation of bubble condensation using CF-VOF". *Progress in Nuclear Energy* 89 (2016), pp. 120 –131.
- [156] M. Ishii. "Two-fluid model for two-phase flow". *Multiphase science and technology* 5.1-4 (1990).
- [157] M. Ishii, T. Hibiki. *Thermo-fluid dynamics of two-phase flow*. Springer Science & Business Media, 2010.
- [158] A. Prosperetti. "Ensemble Averaging Techniques for Disperse Flows". *Particulate Flows*. Vol. 98. The IMA Volumes in Mathematics and its Applications. Springer New York, 1998, pp. 99–136.
- [159] H. Weller. *Derivation, Modeling and Solution of the Conditionally Averaged Two-Phase Flow Equations*. Tech. rep. OpenCFD, 2005.
- [160] H. Rusche. "Computational Fluid Dynamics of Dispersed Two-Phase Flows at High Phase Fractions". PhD thesis. Imperial College of Science, Technology & Medicine, 2002.
- [161] D. Lhuillier, T. Theofanous, M.-S. Liou. "Multiphase Flows: Compressible Multi-Hydrodynamics". *Handbook of Nuclear Engineering*. Ed. by Dan Gabriel Cacuci. Springer US, 2010.
- [162] T. Dinh, R. Nourgaliev, T. Theofanous. "Understanding the ill-posed two-fluid model". *Proceedings of the 10th International topical meeting on nuclear reactor thermal-hydraulics (NURETH03)*. 2003.
- [163] H. A. Jakobsen. "Chemical Reactor Modeling". *Multiphase Reactive Flows, Berlin, Germany: Springer-Verlag* (2008).
- [164] G. Espinosa-Paredes. "A Derivation of the Nonlocal Volume-Averaged Equations for Two-Phase Flow Transport". *Science and Technology of Nuclear Installations* 2012 (2012).
- [165] S. Chu, A. Prosperetti. "On flux terms in volume averaging". *International Journal of Multiphase Flow* 80 (), pp. 176 –180.
- [166] M. Z. Podowski, R. M. Podowski. "Mechanistic multidimensional modeling of forced convection boiling heat transfer". *Science and Technology of Nuclear Installations* 2009 (2008).

- [167] H. Anglart, O. Nylund. "CFD application to prediction of void distribution in two-phase bubbly flows in rod bundles". *Nuclear Engineering and Design* 163.1-2 (1996), pp. 81–98.
- [168] T. Hibiki, M. Ishii. "Interfacial area concentration of bubbly flow systems". *Chemical Engineering Science* 57.18 (2002), pp. 3967–3977.
- [169] E. Krepper, D. Lucas, T. Frank, H.-M. Prasser, P. J. Zwart. "The inhomogeneous MUSIG model for the simulation of polydispersed flows". *Nuclear Engineering and Design* 238.7 (2008), pp. 1690–1702.
- [170] B. Selma, R. Bannari, P. Proulx. "Simulation of bubbly flows: Comparison between direct quadrature method of moments (DQMOM) and method of classes (CM)". *Chemical Engineering Science* 65.6 (2010), pp. 1925–1941.
- [171] G. Yeoh, C. Sherman, J. Tu. "On the prediction of the phase distribution of bubbly flow in a horizontal pipe". *Chemical Engineering Research and Design* 90 (2012), pp. 40–51.
- [172] E. Krepper, B. Koncar, Y. Egorov. "CFD modelling of subcooled boiling - Concept, validation and application to fuel assembly design". *Nuclear Engineering and Design* 237 (2007), pp. 716–731.
- [173] E. Krepper, R. Rzehak, C. Lifante, T. Frank. "CFD for subcooled flow boiling: Coupling wall boiling and population balance models". *Nuclear Engineering and Design* 255 (2013), pp. 330–346.
- [174] R. Rzehak, E. Krepper. "CFD for Subcooled Flow Boiling: Parametric Variations". *Science and Technology of Nuclear Installations* (2013).
- [175] D. Marchisio, R. Fox. "Solution of population balance equations using the direct quadrature method of moments". *Aerosol Science* 36 (2005), pp. 43–73.
- [176] D. Marchisio, A. Barresi, G. Baldi, R. Fox. "Comparison between the Classes Method and the Quadrature Method of Moments for multiphase systems". *8th conference "Multiphase flow in industrial plants", Alba, Italy* (2002).
- [177] R. Fox, F. Laurent, M. Massot. "Numerical simulation of spray coalescence in an Eulerian framework: direct quadrature method of moments and multi-fluid method". *Journal of Computational Physics* 227 (2008), pp. 3058–3088.
- [178] D. Ramkrishna. *Population balances: Theory and applications to particulate systems in engineering*. Academic Press, 2000.
- [179] S. A. Issa, P. Weisensee, R. Macin-Juan. "Experimental investigation of steam bubble condensation in vertical large diameter geometry under atmospheric pressure and different flow conditions". *International Journal of Heat and Mass Transfer* 70.0 (2014), pp. 918–929.
- [180] R. Mudde, W. Hartevelde, H. Akker. "Uniform flow in bubble columns". *Industrial & Engineering Chemistry Research* 48.1 (2008), pp. 148–158.
- [181] K. Agrawal, P. Loezos, M. Syamlal, S. Sundaresan. "The role of meso-scale structures in rapid gas-solid flows". *Journal of Fluid Mechanics* 445 (2001), pp. 151–185.
- [182] K. Jareteg, H. Ström, S. Sasic, C. Demazière. "Numerical investigation of instabilities in the two-fluid model for CFD simulations of LWRs". *M&C 2015, Nashville, Tennessee* (2015).



## **Papers I-VIII**



# Paper I

*“Fine-mesh deterministic modeling of PWR fuel assemblies: Proof-of-principle of coupled neutronic/thermalhydraulic calculations”*

K. Jareteg, P. Vinai, C. Demaziére

*Annals of Nuclear Energy 68 (2014), pp. 247-256*





# Fine-mesh deterministic modeling of PWR fuel assemblies: Proof-of-principle of coupled neutronic/thermal–hydraulic calculations



Klas Jareteg <sup>\*</sup>, Paolo Vinai, Christophe Demazière

Division of Nuclear Engineering, Department of Applied Physics, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden

## ARTICLE INFO

### Article history:

Received 10 June 2013

Received in revised form 22 December 2013

Accepted 30 December 2013

Available online 14 February 2014

### Keywords:

Fine-mesh solver

Neutronics

Thermal–hydraulics

Multi-physics

Coupled deterministic nuclear reactor

modeling

## ABSTRACT

This paper investigates the feasibility of developing a fine mesh coupled neutronic/thermal–hydraulic solver within the same computing platform for selected fuel assemblies in nuclear cores. As a first step in this developmental work, a Pressurized Water Reactor at steady-state conditions was considered. The system being simulated has a finite axial size, but is infinite in the radial direction. The platform used for the modeling is based on the open source C++ library OpenFOAM. The thermal–hydraulics is solved using the built-in SIMPLE algorithm for the mass and momentum fields of the fluid, complemented by an equation for the temperature field applied simultaneously to all the regions (i.e. fluid and solid structures). For the neutronics, a two-group neutron diffusion-based solver was developed, with sets of macroscopic cross-sections generated by the Monte Carlo code SERPENT. The meshing of the system was created by the open source software SALOME. Successful convergence of the neutronic and thermal–hydraulic fields was achieved, thus bringing the solution of the coupled problem to an unprecedented level of details. Most importantly, the true interdependence of the different fields is automatically guaranteed at all scales. In addition, comparisons with a coarse-mesh radial averaging of the thermal–hydraulic variables show that a coarse-mesh fuel temperature identical for all fuel pins can lead to discrepancies of up to 0.5% in pin powers, and of several tens of pcm in multiplication factor.

© 2014 Published by Elsevier Ltd.

## 1. Introduction

The current deterministic modeling of light water nuclear reactor systems, both for steady-state and most importantly for transient calculations, rely on the use of different modeling tools, each tackling the modeling of a specific physical field and/or the modeling of a specific scale (Demazière, 2013). The different fields to be tackled are the neutron density field, the flow field, and the temperature field. The first field is usually referred to as “neutronics”, whereas the two last fields are usually referred to as “thermal–hydraulics”.

On the neutronic side, a lattice code is used to pre-generate sets of macroscopic cross-sections as functions of instantaneous and history variables for each fuel assembly individually modeled in a two-dimensional representation of an infinite lattice. Lattice codes usually embed several successive computational steps, starting with self-shielding of the data, pin cell calculations, assembly calculations, and finally spectrum corrections. The modeling is essentially performed using the neutron transport equation, with a rather high resolution of the neutron density field in space, energy, and angle. These pre-generated sets are thereafter fed into a three-dimensional

coarse-mesh core simulator, usually based on a simpler formalism than neutron transport, such as the diffusion approximation. In this modeling stage, the resolution in space, energy, and angle is rather poor, but the actual system configuration is accounted for. For steady-state simulations, such core simulators also account for a simplified representation of the thermal–hydraulics based on a macroscopic, modeling of the mass and energy/enthalpy conservation equations (for single phase systems), complemented by the linear momentum conservation equation (for two-phase systems). In this paper, macroscopic modeling refers to a volume integration of the fields on rather large volumes, typically an axial slice of 15–20 cm of a fuel assembly. The interdependence between the neutronics and thermal–hydraulics is then resolved by iterating between the neutronic and thermal–hydraulic solvers.

In case of complex reactor transients, the simplified thermal–hydraulic modeling is replaced by coupling the neutronic solver to a dedicated macroscopic thermal–hydraulic solver (also called system code), essentially based on solving the mass, linear momentum, and energy/enthalpy conservation equations on coarse control volumes and time bins. This macroscopic approach is actually equivalent to filtering small scale and high frequency phenomena from the balance equations, with these phenomena being artificially reintroduced in empirically-derived closure relationships.

\* Corresponding author. Tel.: +46 317723077.

E-mail address: [klas.jareteg@chalmers.se](mailto:klas.jareteg@chalmers.se) (K. Jareteg).

Even though such approaches have been proven to give reasonable results in most cases, it can be noticed that the different scales are not fully resolved, since the fine mesh neutron transport solver (i.e. lattice code) uses thermal–hydraulic variables not known at the same level of details as the resolution achieved by the neutron transport solver. Due to the fact that the mean free path of thermal neutrons is on the order of a few centimeters, not resolving the neutronic and thermal–hydraulic fields at the same level of sophistication/details might be challenged when considering modern fuel assemblies with strongly heterogeneous designs/conditions.

The modeling of large reactors with a high resolution of the neutronic and thermal–hydraulic variables still appear to be difficult. However, a detailed simulation of all the fields in a restricted part of the system, such as a fuel assembly, might be particularly interesting.

This work discusses some efforts initiated at the Division of Nuclear Engineering, Department of Applied Physics, Chalmers University of Technology, along such directions. More precisely, this paper considers implementing in the open source C++ library OpenFOAM, primarily and typically used for modeling flow dynamics and heat transfer, a fine-mesh neutronic solver, so that the interdependence between the neutronic and thermal–hydraulic fields can be fully resolved with the same degree of detail. Because of the novelty of the approach and complexity of the problem at hand, it was deemed necessary to start the development and implementation of the neutronic solver with the simplest possible model, in the present case the neutron diffusion approximation in a two-energy group structure. On the thermal–hydraulic side, single-phase flow conditions representative of Pressurized Water Reactors (PWR) were assumed. This work should thus be considered as a feasibility study for the development of a fine mesh coupled neutronic/thermal–hydraulic solver, rather than the actual high fidelity modeling of all scales and fields.

The paper is structured as follows. First, the models and methods used in this feasibility study are presented, both for the neutronics and the thermal–hydraulics. Thereafter, the numerical solution procedure and treatment of the coupling between the fields is touched upon. Finally, the application of the developed tool to a PWR fuel assembly at steady-state conditions is shown, with emphasis on the convergence of the fields. Conclusions are then drawn on the applicability and usefulness of the methodology.

## 2. Models and methodology

In this section, the models and methodology used for steady-state coupled calculations are presented. The models tackle neutronics, fluid dynamics and heat transfer.

The work is based on a finite volume method and it makes use of the open source C++ library OpenFOAM® (OpenFOAM, 2012). The library allows implementation of fields, equations and operator discretization using a high-level C++ coding interface. Descriptions of the available mathematical operators, discretization implementations, matrix solvers and the general use of the software is found in OpenFOAM (2011).

### 2.1. Neutronic model

The steady state neutron flux  $\Phi(\mathbf{r}, E)$  is computed using the neutron diffusion approximation (Bell and Glasstone, 1970):

$$-\nabla \cdot (D(\mathbf{r}, E) \nabla \Phi(\mathbf{r}, E)) + \Sigma_T(\mathbf{r}, E) \Phi(\mathbf{r}, E) = \int_0^\infty \Sigma_{s0}(\mathbf{r}, E' \rightarrow E) \Phi(\mathbf{r}, E') dE' + \frac{\chi(E)}{k_{\text{eff}}} \int_0^\infty v \Sigma_f(\mathbf{r}, E') \Phi(\mathbf{r}, E') dE' \quad (1)$$

where the meaning of each quantity is given in the nomenclature in the appendix. The energy dependence is discretized in a multi-group formalism, such that for

$$[E_{\min}, E_{\max}] = \bigcup_{g=1}^G [E_g, E_{g-1}] \quad (2)$$

the neutron diffusion Eq. (1) will read as:

$$-\nabla \cdot (D_g(\mathbf{r}) \nabla \Phi_g(\mathbf{r})) + \Sigma_{T,g}(\mathbf{r}) \Phi_g(\mathbf{r}) = \sum_{g'=1}^G \Sigma_{s0,g' \rightarrow g}(\mathbf{r}) \Phi_{g'}(\mathbf{r}) + \frac{\chi_g^p}{k_{\text{eff}}} \sum_{g'=1}^G v \Sigma_{f,g'}(\mathbf{r}) \Phi_{g'}(\mathbf{r}) \quad (3)$$

Since the aim of the present work is to investigate the feasibility of a coupled neutronic/thermal–hydraulic scheme based on a fine mesh, the neutron diffusion approximation was chosen for the sake of simplicity. To take advantage of the fine three-dimensional resolution of the CFD approach, a neutron transport method will be implemented in a later stage.

#### 2.1.1. Solution methodology of the neutronic model

A detailed implementation of the neutronics algorithm can be seen in Fig. 1. In the first stage, the fission source is computed for all regions (fuel, gap, cladding and moderator) and all energy groups. A linear matrix system of equations is then assembled, one group at a time. This corresponds to eq. (3) for each computational cell, including the contribution from all other groups through the up and down scattering and the fission source. The matrix equation is then solved using Krylov space iterative solvers. This step is performed for all neutron energy groups and corresponds to an inner iteration, i.e. the spatial dependence of the neutron flux for the given energy group is resolved. The new neutron flux is used to compute the effective multiplication factor according to the power iteration method (Nakamura, 1977). Finally the integral power level is renormalized according to a fixed power level which is specified as an input parameter.

#### 2.1.2. Cross-sections and macroscopic neutronic data

In order to generate the groupwise macroscopic neutronic data needed in eq. (3), a methodology based on the Monte Carlo code SERPENT (Leppänen, 2012) was developed. The set of data computed includes the groupwise cross-sections, the diffusion coefficients ( $D_g$ ), the neutron fission yields ( $v_g$ ) and the fission spectra ( $\chi_g$ ).

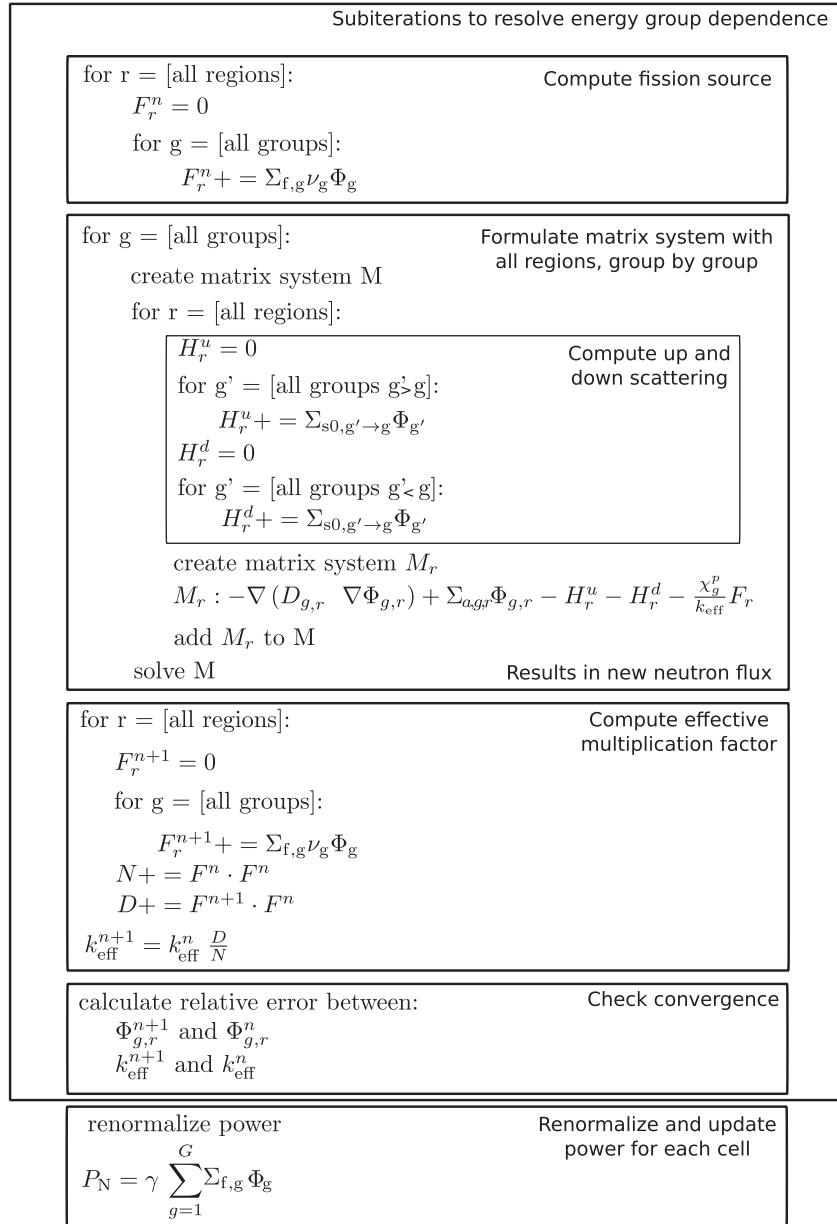
A pin cell that consists of a nuclear fuel rod (including also the gap and the cladding) with the surrounding moderator is considered. Its spatial discretization is shown in Fig. 2, and for each region one set of groupwise macroscopic neutronic parameters is calculated.

Cross-sections are generated for a set of thermophysical state points, specifying temperature and, in the case of the moderator, density. This results in a set of different cross-sections for each region.

An interpolation is performed each time updating the cross-sections, and it will only be determined from the temperature and density in the present computational cell. This means that the dependencies between the cells are neglected, and consequently the assumed temperature profile in the pin cell is not exactly followed. This will need to be addressed in the continuation of this work.

### 2.2. Thermal–hydraulic model

The governing equations for the fluid dynamics are the continuity equation, the momentum equation (i.e. Navier Stokes equation) and an energy equation (Panton, 2005). In this work only single



**Fig. 1.** Overview of the applied methodology for resolving the neutronic field, with one sub-iteration being detailed.

phase flow is modeled, so the moderator will be assumed a liquid at all times.

To retain the influence of small time scale phenomena as turbulence also in the steady-state equations, a Reynolds decomposition is performed. Accordingly, the generic quantity  $f(t)$  (that can be velocity, pressure or enthalpy) is decomposed in a mean value  $F(t)$  and a fluctuating part  $f'(t)$ :

$$f(t) = F(t) + f'(t) \quad (4)$$

with

$$F(t) = \frac{1}{\delta t} \int_t^{t+\delta t} f(t') dt' \quad (5)$$

This results in the time averaged, steady-state equations:

$$\nabla \cdot (\rho \mathbf{U})(\mathbf{r}) = 0 \quad (6)$$

$$\nabla \cdot (\rho \mathbf{U} \otimes \mathbf{U})(\mathbf{r}) = \nabla \cdot \bar{\tau}(\mathbf{r}) - \nabla \cdot \overline{\rho \mathbf{u}' \otimes \mathbf{u}'}(\mathbf{r}) - \nabla P(\mathbf{r}) + \rho(\mathbf{r}) \mathbf{g} \quad (7)$$

$$\nabla \cdot (\rho \mathbf{U} H)(\mathbf{r}) = -\nabla \cdot \overline{\mathbf{q}''(\mathbf{r})} + \overline{\mathbf{q}'''(\mathbf{r})} + \nabla \cdot (\mathbf{U} P)(\mathbf{r}) + \bar{\tau}(\mathbf{r}) : \nabla \otimes \mathbf{U}(\mathbf{r}) - \nabla \cdot \overline{(\rho \mathbf{u}' h')}(\mathbf{r}) \quad (8)$$

where the overbar indicates time-averaging and the quantities  $\mathbf{U}$ ,  $H$  and  $P$  are the mean values of the velocity, the enthalpy and the pressure respectively. The terms including pressure and stress tensor fluctuations are neglected. The stress tensor,  $\bar{\tau}$ , for Newtonian fluids is given by Panton (2005):

$$\tau_{ij} = \mu \left( U_{ij} + U_{ji} - \frac{2}{3} U_{kk} \delta_{ij} \right) \quad (9)$$

The velocity fluctuation term  $\overline{\rho \mathbf{u}' \otimes \mathbf{u}'}$  is modeled on the basis of the Boussinesq assumption, and it can be expressed in terms of the turbulent kinetic energy  $k$  and the mean flow velocity  $\mathbf{U}$  as:

$$-\overline{\rho u_i' u_j'} = \mu_t \left( U_{ij} + U_{ji} - \frac{2}{3} U_{kk} \delta_{ij} \right) - \frac{2}{3} \rho k \delta_{ij} \quad (10)$$

The enthalpy and velocity fluctuation term  $-\nabla \cdot \overline{(\rho \mathbf{u}' h')}$  can be related to the turbulent viscosity as follows (Versteeg and Malasekera, 2007):

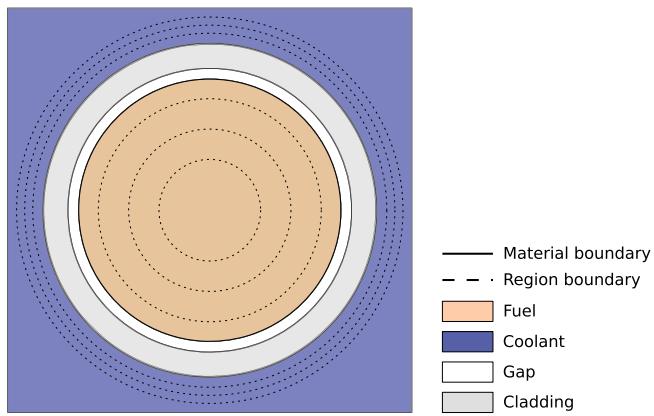


Fig. 2. Geometry for cross-section generation. All boundaries are treated as reflective boundaries. Figure not in scale.

$$-\nabla \cdot (\overline{\rho \mathbf{u} h'})(\mathbf{r}) = \nabla \cdot \left( \frac{\mu_t}{P_r} \nabla h \right) \quad (11)$$

In the current framework, the standard  $k - \epsilon$  turbulence model is applied to determine the turbulent kinetic energy and its dissipation ( $\epsilon$ ) (Ferziger and Peric, 2002):

$$\nabla \cdot (\rho k \mathbf{U}) = \nabla \cdot \left( \mu + \frac{\mu_t}{\sigma_k} \nabla k \right) + \mu_t (\nabla \otimes \mathbf{U}) : (\nabla \otimes \mathbf{U} + (\nabla \otimes \mathbf{U})^T) - \rho \epsilon \quad (12)$$

$$\begin{aligned} \nabla \cdot (\rho \epsilon \mathbf{U}) = \nabla \cdot \left( \mu + \frac{\mu_t}{\sigma_\epsilon} \nabla \epsilon \right) + C_{1\epsilon} \frac{\epsilon}{k} \mu_t (\nabla \otimes \mathbf{U}) : (\nabla \otimes \mathbf{U} + (\nabla \otimes \mathbf{U})^T) \\ - C_{2\epsilon} \rho \frac{\epsilon^2}{k} \end{aligned} \quad (13)$$

with the turbulent viscosity given by:

$$\mu_t = \rho C_\mu \frac{k^2}{\epsilon} \quad (14)$$

The influence of heat transfer by radiation in the gap, is taken into account by providing a separate transport equation for the incident radiative heat flux. The radiative heat transfer equation used in this work is derived from a P1 approximation and integrated over all frequencies (Modest, 2003):

$$-\nabla \cdot \left( \frac{1}{3(K_a - K_s) - C_A K_s} \nabla G \right) = K_a (4E_b(T) - G) \quad (15)$$

where  $K_a$  is the radiation absorption coefficient,  $K_s$  the radiation scattering coefficient,  $C_A$  a model constant and  $E_b(T)$  is the black-body source:

$$E_b(T) = \sigma_{\text{SB}} T^4 \quad (16)$$

where  $\sigma_{\text{SB}}$  is the Stefan–Boltzmann constant. The heat radiation transport influences the general energy equation through a productive term (corresponding to radiation absorption) and a dissipative term (corresponding to black body radiation), with the sum denoted as  $\overline{q'''}_{\text{rad}}$ :

$$\overline{q'''}_{\text{rad}}(T) = K_a (G - 4E_b(T)) \quad (17)$$

The enthalpy Eq. (8) is implemented as a temperature equation. To do so, Fourier's law of conduction and the temperature dependence of the enthalpy are introduced. This yields:

$$(\rho c_p(T)) \mathbf{U} \cdot \nabla T = \beta(T) \mathbf{U} \cdot \nabla P + \nabla \cdot (K_{\text{eff}}(T) \nabla T) + \overline{q'''}_{\text{rad}} \quad (18)$$

where the frictional terms, i.e. terms expressed as  $\bar{\tau}(\mathbf{r}) : \nabla \otimes \mathbf{U}(\mathbf{r})$ , are neglected.

The steady state temperature distribution in the solid regions of the reactor is calculated by Todreas and Kazimi (1993):

$$-\nabla \cdot (K(\mathbf{r}, T) \nabla T(\mathbf{r})) = \overline{q'''}(\mathbf{r}) \quad (19)$$

where the volumetric source term is given by the energy released by fission in the region:

$$\overline{q'''}(\mathbf{r}) = \gamma \sum_{g=1}^G \sum_{f,g} \Phi_g(\mathbf{r}) \quad (20)$$

It is assumed that all energy from fission is dissipated in the same control volume as where the fission occurred, e.g. no smearing from gamma radiation is considered. The solid regions (fuel and cladding) are assumed rigid without any thermal expansion or contraction, consequently no mass or momentum equations are solved for the solid regions.

### 2.2.1. Thermophysical data

The material properties (as density, thermal conductivity, laminar viscosity, specific heat capacity) depend on the material composition, temperature and pressure. In this work, the thermophysical data of the fluids are provided for several temperature values and one pressure value (1.5 MPa for water and 0.1 MPa for the gaseous gap). In the case of solid, temperature dependent thermal conductivity and specific heat capacity are included.

Therefore, these parameters are associated to the mesh cells according to the heterogeneities of the systems and their values are updated during the calculations for each update of the temperature field. The spatial resolution of the thermophysical data is thus identical to the ones of the fields being solved for. The sources of the data used are given with the nomenclature in the appendix.

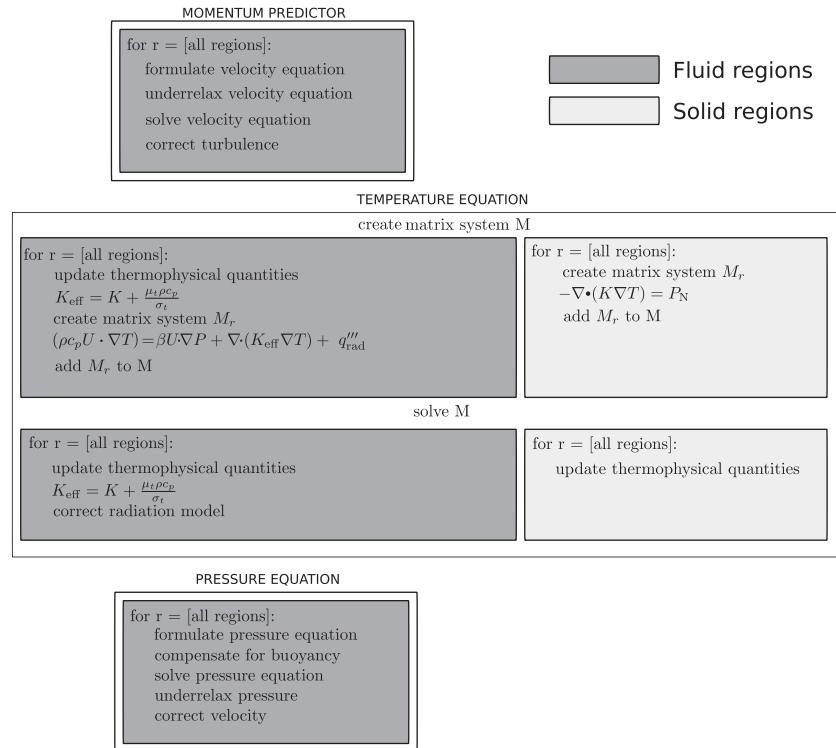
### 2.2.2. Solution methodology of the thermal–hydraulic model

A schematic of the solution procedure is given in Fig. 3. The continuity and momentum equations (eqs. (6) and (7)) are solved using the SIMPLE algorithm (Patankar and Spalding, 1972) as implemented in OpenFOAM. In this iterative algorithm a predicted semi-discretized momentum equation is inserted into the continuity equation, producing an equation for pressure. After the solution of the pressure equation, the momentum equation is processed and the velocity profile is updated. The turbulence equations (eqs. (12) and (13)) and the thermal radiation model are solved using the built-in turbulence routines in OpenFOAM. Under-relaxation is applied to the pressure, velocity and turbulence quantities according to the standard SIMPLE values. For the velocity and turbulent quantities equation systems, Krylov subspace methods are applied, whereas the pressure equation is computed using an algebraic multigrid solver.

The fluid temperature Eq. (18) has been implemented together with the solid temperature Eq. (19) and they are solved for all regions in a combined matrix equation. The full conjugate heat transfer problem is solved accordingly without any iteration between the solid and fluid regions. Again Krylov subspace methods are used. Also, in this case no under-relaxation is necessary.

### 2.3. Numerical solution

The mesh is created with the open source software SALOME (CEA and EDF, 2012), and the neutronic and thermal–hydraulic fields are computed with respect to the same spatial mesh. Although the neutronics does not normally require a grid as fine as the fluid dynamics, the advantage of a common mesh is that no interpolation or cell mapping between the two fields are needed. Moreover, since the methodology is based on an integrated algorithm (i.e. all the equations are implemented in the same software) an efficient coupling scheme is achieved. In fact no external data transfer is necessary so that computational time and memory cost are saved.



**Fig. 3.** Overview of the methodology applied to solve the thermal–hydraulics, with one sub-iteration being detailed.

This approach is different from other recent works, where standalone neutronic and thermal–hydraulic solvers are coupled. An example of this can be seen in Kochunas et al. (2012). In contrast to the mentioned work, a direct transfer of the power in the fuel and temperature field in all regions is allowed. This gives a consistent high-resolution coupling, where transferred fields are as highly resolved as the separate neutronics and thermal–hydraulics solutions. Furthermore, in comparison to the separate code approach, solving the full problem in the same software minimizes the cost of data transfer, avoiding any type of I/O data transfer as used in Kochunas et al. (2012).

The model equations are spatially discretized using the high-level C++ coding interface available in OpenFOAM. In particular, the diffusion terms in eqs. (3), (7), (18) and (19) are treated with a central differencing scheme. The convective term in eq. (7) is treated with an upwind scheme.

#### 2.4. Coupling neutronic and thermal–hydraulic fields

In Light Water Reactors (LWRs), the coolant is also the neutron moderator. Thus, the neutronics and the thermal–hydraulics are strongly coupled. On one hand, the neutron density field determines the distribution of the heat released in the fuel (eq. (20)). On the other hand, the temperature and density profiles calculated from the thermal–hydraulic model affect the macroscopic cross-sections.

The general iterative procedure used to couple the neutronics and the thermal–hydraulics is shown in Fig. 4. The neutron diffusion equation and the temperature equation are each solved in a combined manner. All material regions (claddings, gaps, fuel regions and moderator) are assembled into the same system of equations. The neutron diffusion equation still needs to be iteratively solved, as, for the inner iterations, only one neutron energy group is solved at a time. The fluid momentum, turbulence and pressure equations are solved region by region, one matrix for each gap in each fuel pin and one for the whole moderator, respectively.

The implemented coupled algorithm allows sub-iterations in the neutronics and/or the thermal–hydraulics. The number of sub-iterations is preferably optimized to give better performance in terms of convergence rate.

For the initialization, spatially constant values are applied to all fields. In order to ensure convergence the neutronics is first solved using the constant temperature profile of the thermal–hydraulics. The outer, coupled, iteration is thus solved first with the neutronics as indicated in Fig. 4. For the first outer iteration, the sub-iterations in the thermal–hydraulics do not include the temperature. This ensures that physical pressure and velocity profiles are first achieved, so that the convergence of the problem is somewhat facilitated.

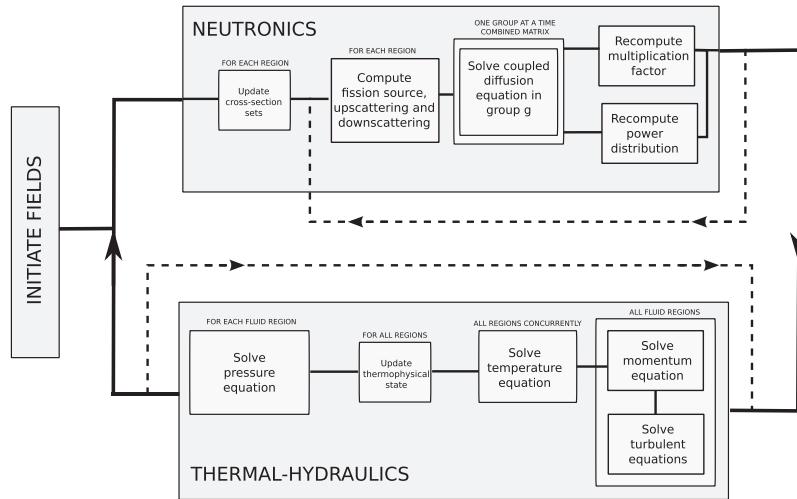
### 3. Application

The developed method was applied to a simplified nuclear fuel assembly at PWR conditions, with only single phase flow for the coolant. First, the system is described, then, the boundary conditions together with other details used for performing the calculations are discussed, finally the results are presented.

#### 3.1. Description of the fuel assembly

The system used for the verification of the current algorithm is a simplified  $5 \times 5$  fuel assembly. The geometry is shown in Fig. 5, and the general characteristics are summarized in Table 1. No spacers are included. A checkerboard pattern of pins with high and low enrichments is considered, with the central fuel pin having a high enrichment.

The radial discretization of a single fuel pin is given in Fig. 6. For the gap, the cladding and the part of the moderator close to the pin, a structured mesh is used. The axial discretization varies with height in order to give better resolution at the inlet and the outlet, where the flow path changes. The single pin mesh is repeated for all pins in the assembly. The total number of cells for each pin with the surrounding moderator is 64,000, split on 80 axial levels.



**Fig. 4.** Overview of the applied methodology to solve the coupled neutronics and thermal–hydraulics, with the outer iteration and sub-iterations indicated by full lines and dashed lines, respectively.

Each computational volume in Fig. 6 is associated with one region in the SERPENT geometry (as seen in Fig. 2). The position is determined from the center of the computational cell. Following the update of the temperature in the system, new cross-sections are calculated using interpolation in the pre-generated sets of macroscopic data. This is done for each cell in the system, based on the temperature solution from the thermal–hydraulic calculations. Accordingly, the cross-sections are allowed to vary with the resolution of the fine mesh.

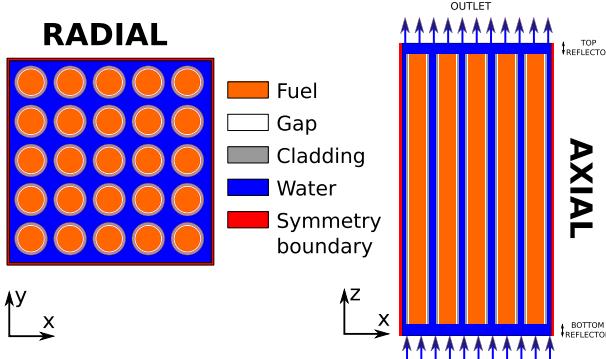
### 3.2. Setup for the calculation

The choice of under-relaxation and convergence criteria for the matrix solvers are reported in Table 2. The boundary conditions of the problem correspond to a realistic PWR case (see Table 3). In the radial direction, symmetry boundary conditions are assumed. This corresponds to an infinite lattice of the  $5 \times 5$  system. For the neutronics and the temperature equation, the region to region boundary conditions are not needed. Since the equations for all regions are solved in one system, the dependencies between the regions will be implicit in the combined, concurrently solved matrix.

The total power of the  $5 \times 5$  assembly was chosen to be 12.5 MW. The neutronics is computed using two energy groups, with the energy boundary at 0.625 eV.

### 3.3. Results

Figs. 7 and 8 show the simulated radial and axial neutron flux for the two energy groups. The expected higher density of thermal

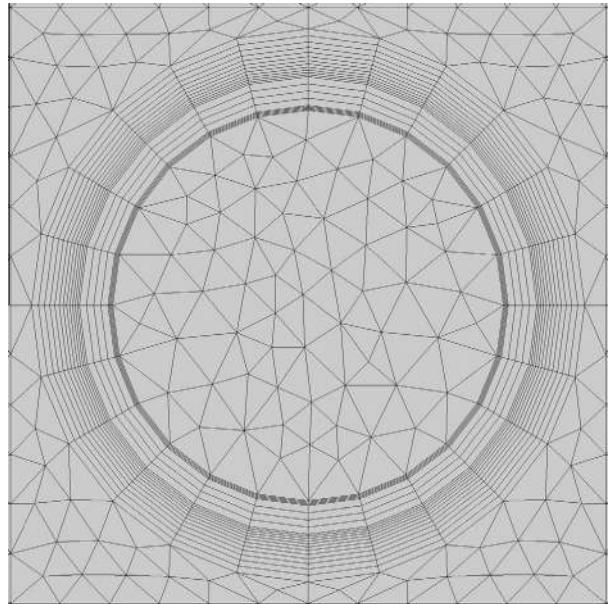


**Fig. 5.** Outline of the geometry and materials used for the  $5 \times 5$  assembly.

**Table 1**

Dimensions and characteristics of the  $5 \times 5$  fuel assembly.

Fuel pin radius	4.1 mm
Cladding inner radius	4.2 mm
Cladding outer radius	4.8 mm
Pitch	12.5 mm
Fuel height	3 m
Bottom reflector height	0.2 m
Top reflector height	0.2 m
Fuel	UOX (2% and 4% enrichment)
Moderator	Water, 1000 ppm boron
Gap	Helium, 0.1 MPa
Cladding	Zircaloy-2



**Fig. 6.** Radial discretization of a single fuel pin.

neutrons in the moderator and higher density of fast neutrons in the fuel are correctly predicted. Axially the fast neutrons peak slightly below mid-elevation, because of the decrease of the moderator density from the bottom to the top of the system.

Figs. 9 and 10 give the radial and axial temperature profiles in the moderator. As seen, the variation of the moderator temperature is captured. The inhomogeneity influences the neutronics through the cross-sections in the moderator.

**Table 2**

Solver settings used for each field, including typical convergence criteria and under-relaxation factors, where 1.0 means no under-relaxation.

Field	Type of solver	Conv. criteria	Under-relaxation
$T$	Gauss-Seidel	$10^{-4}$	1.0
$\Phi$	Stab. biconjugate gradient	$10^{-4}$	1.0
$P$	Precond. biconjugate gradient	$10^{-3}$	0.3
$G$	Precond. biconjugate gradient	$10^{-5}$	1.0
$U$	Precond. stab. biconjugate gradient	$10^{-5}$	0.7
$k$	Precond. stab. biconjugate gradient	$10^{-5}$	0.7
$\epsilon$	Precond. stab. biconjugate gradient	$10^{-5}$	0.7

**Table 3**

Boundary conditions used for the different computed quantities.

Field	Boundary	Type	Value
$T$	Inlet	Constant value	540 K
	Outlet	Zero gradient	
	Region to region	Implicit	
$\Phi_g$	Inlet	Constant value	$0 \text{ m}^{-2} \text{ s}^{-1}$
	Outlet	Constant value	$0 \text{ m}^{-2} \text{ s}^{-1}$
	Region to region	Implicit	
$P$	Inlet	Zero gradient	
	Outlet	Constant value	15.5 MPa
	Fluid to solid	Zero gradient	
$k$	Inlet	Constant	$0.1 \text{ m}^2 \text{ s}^{-2}$
	Outlet	Zero gradient	
	Fluid to solid	Wall function	
$\epsilon$	Inlet	Constant	$2.0 \text{ m}^2 \text{ s}^{-3}$
	Outlet	Zero gradient	
	Fluid to solid	Wall function	
$G$	All	Marshak	
$U$	Inlet	Constant value	$(0, 0, 3) \text{ m s}^{-1}$
	Outlet	Zero gradient	
	Fluid to solid	No slip	$(0, 0, 0) \text{ m s}^{-1}$

Cross-sectional results of the radial temperature is displayed in Fig. 11 for different axial elevations. Every second pin, corresponding to the higher enrichment pins, has a higher temperature, due to more fissions. The chosen spatial discretization allows resolving the radial gradient in the fuel, through the gap and the cladding.

The radial power density profile in the fuel is seen in Fig. 12. The self-shielding in the fuel pins is spatially resolved. Again a difference is captured between the higher and the lower enriched pins, respectively. The calculated axial power density profile peaks below mid-elevation (Fig. 13). As earlier explained, and as highlighted in Fig. 10, this is due to the increase of the moderator temperature from bottom to top, thus leading to better neutron moderation at the core bottom.

In addition, convergence of key parameters (i.e. multiplication factor, temperature, pressure, fast flux) are analyzed (Fig. 14). The elapsed time corresponds to the calculational time on a 4 CPU Intel® Core™ i7-3770 K. Parallelization is applied using the built-in MPI functionality in OpenFOAM. Different numbers of sub-iterations for the neutronics and the thermal-hydraulics were tested, and it was found that a high number is beneficial. The case presented here is summarized in Table 4, with a maximum of 100 sub-iterations applied for both the neutronics and thermal-hydraulics. Whereas the neutronics converges in fewer than the maximum allowed sub-iterations for all outer iterations, the thermal-hydraulics need 4 outer iteration to converge in less than 100 sub-iterations. In the first outer iteration the neutronics is allowed to converge indepen-

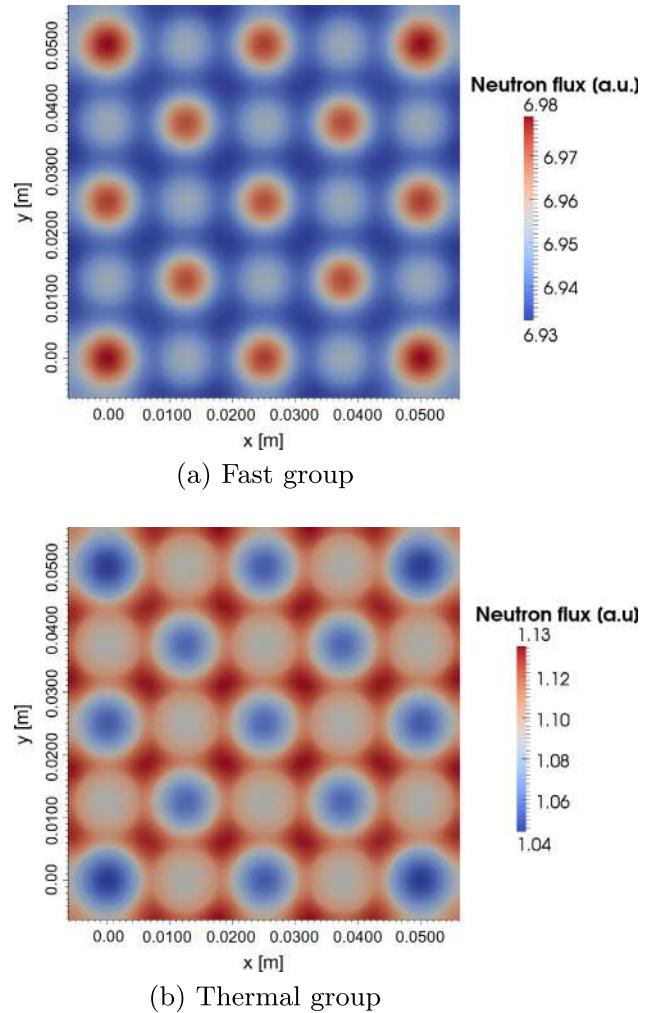


Fig. 7. Radial profiles of the neutron flux at mid-elevation ( $z = 1.7 \text{ m}$ ).

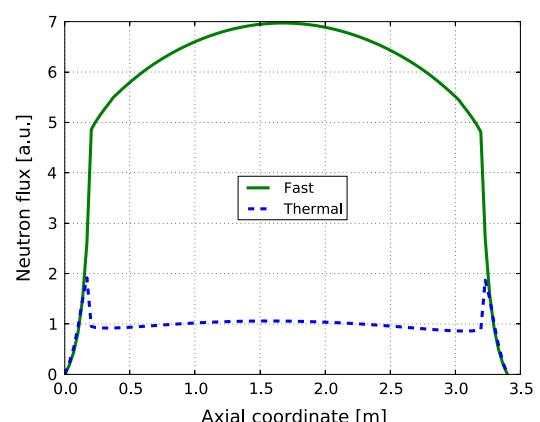


Fig. 8. Axial profile of the neutron flux taken in the middle of the central pin.

dently of the thermal-hydraulics. This is necessary to ensure the convergence of the solution starting from reasonable guesses of all fields. In the second iteration, the temperature is computed on the basis of the pressure and velocities calculated in the first iteration, including the power source estimated from the neutronics. The first fully coupled iteration is number 3. In this iteration, the temperature profile is used to update the cross-sections. The low

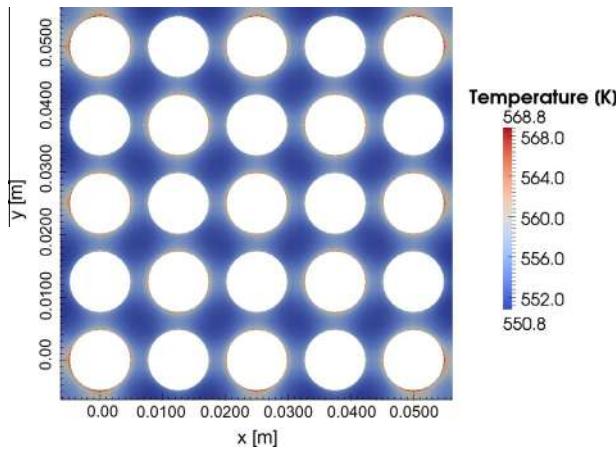


Fig. 9. Radial temperature profile in the moderator.

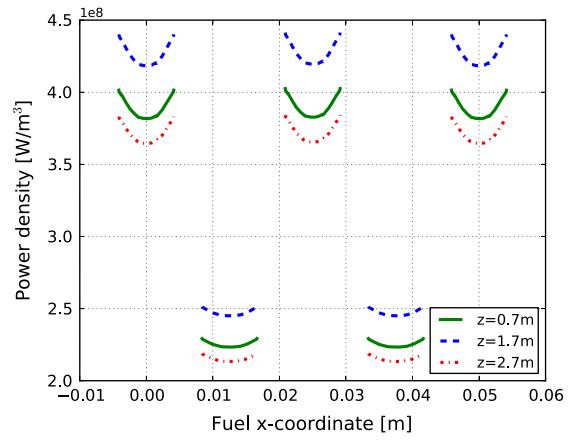


Fig. 12. Radial power density profile in the fuel pins. The profiles are given along lines chosen through the centers of adjacent fuel pins, for different elevations.

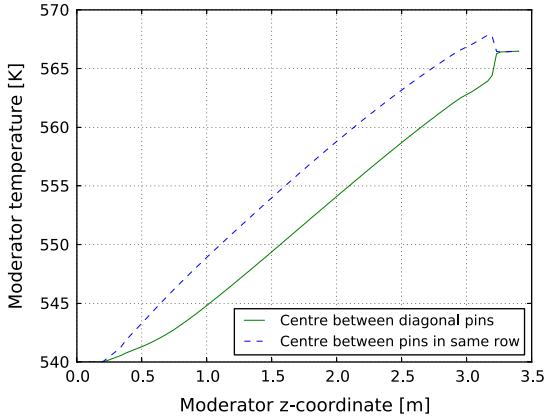


Fig. 10. Axial temperature profile in the moderator. The profiles are given in the moderator at the center between two diagonally adjacent pins (solid line), or two pins in same row (dashed line), respectively.

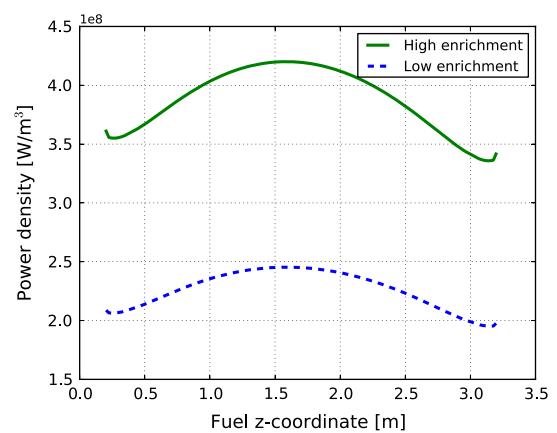


Fig. 13. Axial power density profile in the fuel pins. The profiles are given at the center of a highly enriched fuel pin, and at the center of a low enriched fuel pin, respectively.

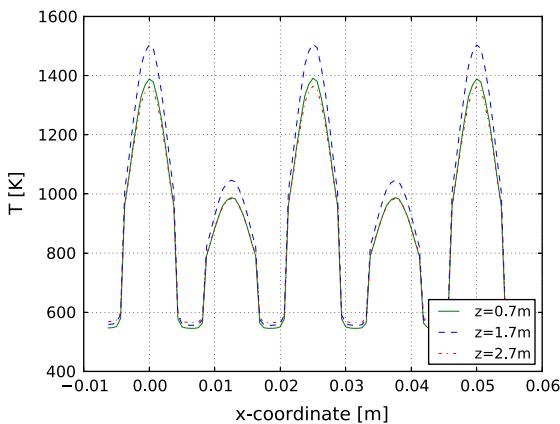


Fig. 11. Radial temperature profile throughout the assembly. The profiles are given along lines chosen through the centers of adjacent fuel pins, for different elevations.

number of outer iterations indicates that the system is loosely coupled for a coolant being in liquid phase.

The computational time is balanced between the neutronics and thermal-hydraulics (see Table 5). This is the result of the same resolution of the calculations used for both parts. The choice of matrix solver has a major influence on the time spent in each part of the code.

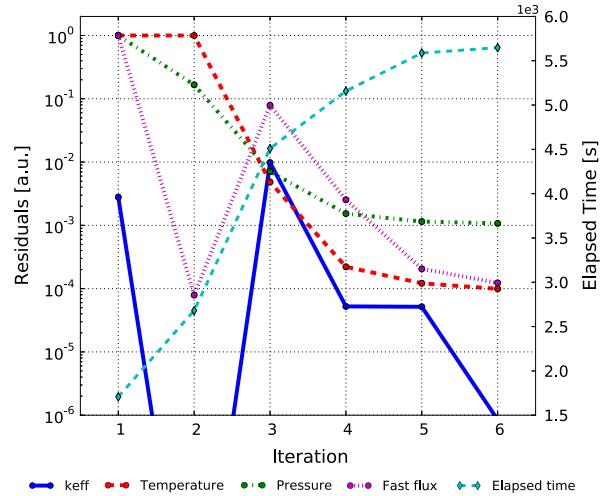


Fig. 14. Convergence with respect to the outer iterations in the coupled methodology (see Fig. 4).

In order to show the impact of the fine-mesh approach, the system was, after full convergence, averaged in radial direction in the moderator and in the fuel pins, respectively. For the fuel pins, two averaging procedures were tested: either an averaging on each of

**Table 4**

Number of sub-iterations per coupled iteration.

Iteration	Neutronics	Thermal-hyd.	Note
1	90	100	Non-coupled iteration
2	1	100	Temperature added in thermal-hydraulics
3	95	100	Cross-sections first updated
4	16	100	
5	8	73	
6	3	9	

the individual fuel pins, or an averaging on all fuel pins together. The full axial resolution (80 levels) was kept. For each axial segment, an averaged enthalpy of the converged results was computed and the cross-sections updated accordingly.

**Table 6** and **Fig. 15** show the results of such computations. Averaging of the moderator has a minor effect, which is anticipated as the radial gradients of temperature in the moderator are very small, as can be seen in **Fig. 9**. For pin by pin averaging of the fuel, the power is increased in the center of the fuel. This is the result of the radial temperature profile (**Fig. 11**) being flattened, leading to a colder fuel at the center of the pins. When all the fuel pins are averaged together, an additional effect can be seen. Namely, the overall temperature in the low-enriched pins increases because of the averaging, whereas the reverse effect can be noticed in the highly enriched pins. Therefore, this outcome points out that fine resolution simulations of the fuel assembly power density can lead to relative differences of more than 0.5% with respect to those obtained

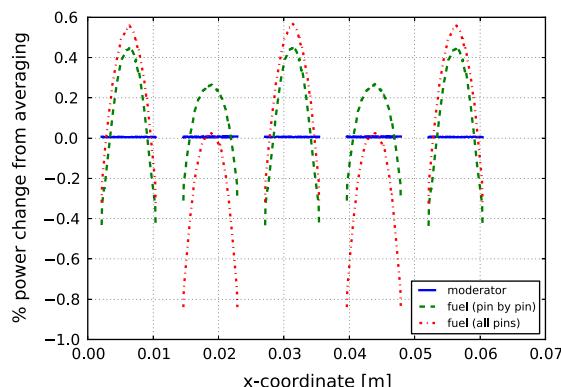
**Table 5**

Amount of time spent for solving each physical field.

Calculation	Elapsed time (s)	% of total
Neutronics	2677	47.8
Momentum predictor	1254	22.4
Temperature equation	925	16.5
Pressure equation	748	13.3

**Table 6**Change in  $k_{\text{eff}}$  as a result of the homogenization in moderator and fuel, respectively.

Averaged cell region	$k_{\text{eff}}$	Change (pcm)
None (reference)	1.17109	
Moderator	1.17109	0
Fuel (pin by pin)	1.17032	-77
Fuel (all pins)	1.16999	-110



**Fig. 15.** Power change due to averaging in moderator and fuel respectively. The relative change in power is computed using the heterogeneous solution as a reference.

with radial averages. Moreover, discrepancies of several tens of pcm are observed in the predicted effective multiplication factor, as reported in **Table 6**.

#### 4. Conclusions

In this paper, a first-of-a-kind coupled neutronic/thermal hydraulic model of a single PWR fuel assembly at steady-state conditions, axially modeled as being finite but radially taken in an infinite lattice, was developed using the open source C++ library OpenFOAM. This model has the advantage of resolving all physical fields with the same level of details, in the same computing platform, and in a truly coupled manner.

Comparisons with a coarse-mesh radial averaging of the thermal-hydraulic variables show that using a coarse-mesh fuel temperature identical for all fuel pins can lead to errors in pin powers of about 0.5% and lead to discrepancies of several tens of pcm in multiplication factor.

Although this study demonstrates that a fine mesh and coupled modeling of fuel assemblies is useful for getting information at an unprecedented level of details, further developments are necessary in order to achieve a high-fidelity modeling. Among others, work is on-going to replace the two-group diffusion based solver by a multi-group discrete ordinates solver, so that a higher resolution in angle and energy of the neutron density field can be achieved. In addition, the development and implementation of a two-phase flow solver targeted at the simulation of Boiling Water Reactor fuel assemblies have been initiated.

The ultimate goal of this approach is to be able to get detailed simulations for selected fuel assemblies in an actual commercial nuclear core. In view of this, a coarser modeling of the surrounding fuel assemblies could be considered. In such a case, the proper treatment of the neutronic boundary conditions in both the fine-mesh and coarse-mesh solver as well as their interdependence need to be particularly looked upon.

#### Acknowledgements

The Swedish Centre for Nuclear Technology (SKC) is acknowledged for financially supporting this PhD project. This work is also part of a larger effort undertaken at Chalmers University of Technology for developing next-generation methods for modeling nuclear reactor systems. In this framework also supported by the Swedish Research Council (VR), the Nordic Thermal-hydraulic Network (NORTHNET), and the Swedish Radiation Safety Authority (SSM), researchers with expertise in neutron transport, fluid

**Table A.1**  
Nomenclature, with description and sources, for the neutronic quantities.

Quantity	Description	Source	Unit
$\Sigma_a$	Absorption cross-section	Monte Carlo	$[\text{m}^{-1}]$
$D$	Diffusion coefficient	Monte Carlo	$[\text{m}]$
$H^d$	Downscattering source term	Calculation	$[\text{m}^{-3} \text{s}^{-1}]$
$\gamma$	Energy per fission		$[3.2 \times 10^{-11} \text{ J}]$
$\Sigma_f$	Fission cross-section	Monte Carlo	$[\text{m}^{-1}]$
$\chi$	Fission neutron spectrum	Monte Carlo	[1]
$F$	Fission source term	Calculation	$[\text{m}^{-3} \text{s}^{-1}]$
$\Sigma_{s0}$	Isotropic scattering cross-section	Monte Carlo	$[\text{m}^{-1}]$
$k_{\text{eff}}$	Multiplication factor	Calculation	[1]
$v$	Neutron fission yield	Monte Carlo	[1]
$P_N$	Power density	Calculation	$[\text{W m}^{-3}]$
$\phi$	Scalar neutron flux	Calculation	$[\text{m}^{-2} \text{s}^{-1}]$
$\Sigma_T$	Total macroscopic cross-section	Monte Carlo	$[\text{m}^{-1}]$
$H^u$	Upscattering source term	Calculation	$[\text{m}^{-3} \text{s}^{-1}]$

**Table A.2**

Nomenclature, with description and sources, for the thermal–hydraulic quantities.

Quantity	Description	Source	Unit
$\rho$	Density	International Atomic Energy Agency (2006) (solids), NIST Webbook (2011) (fluids)	[kg m <sup>-3</sup> ]
$\epsilon$	Dissipation of turbulent kinetic energy	Calculation	[m <sup>2</sup> s <sup>-3</sup> ]
$\mathbf{g}$	Gravitational acceleration		[−9.81 m s <sup>-2</sup> ]
$G$	Incident radiative heat flux	Calculation	[W m <sup>-2</sup> ]
$K_a$	Radiation absorption coefficient	Fink (2000) (UOX), Murphy and Havelock (1976) (Zircaloy)	[m <sup>-1</sup> ]
$K_s$	Radiation scattering coefficient	Fink (2000) (UOX), Murphy and Havelock (1976) (Zircaloy)	[m <sup>-1</sup> ]
$c_p$	Specific heat capacity at constant pressure	International Atomic Energy Agency (2006) (solids), NIST Webbook (2011) (fluids)	[J K <sup>-1</sup> kg <sup>-1</sup> ]
$\tau$	Stress tensor	NIST Webbook (2011)	[N m <sup>-1</sup> ]
$\mathbf{q}''$	Surface heat flux	Calculation	[J m <sup>-2</sup> ]
$T$	Temperature	Calculation	[K]
$K$	Thermal conductivity	International Atomic Energy Agency (2006) (solids), NIST Webbook (2011) (fluids)	[W K <sup>-1</sup> m <sup>-1</sup> ]
$\beta$	Thermal expansion coefficient	Calculation	[K]
$H$	Time averaged specific enthalpy	Calculation	[J kg <sup>-1</sup> ]
$P$	Time averaged pressure	Calculation	[Pa]
$\mathbf{U}$	Time averaged velocity	Calculation	[m s <sup>-1</sup> ]
$k$	Turbulent kinetic energy	Calculation	[m <sup>2</sup> s <sup>-2</sup> ]
$P_r$	Turbulent Prandtl	Demazière and Mattson (2006)	[1]
$q''_{\text{rad}}$	Volumetric heat source thermal radiation	Calculation	[J m <sup>-3</sup> ]

dynamics, heat transfer, and numerical techniques, have gathered forces in a task force called DREAM (Deterministic Reactor Modelling).

## Appendix A. Nomenclature

The following tables summarize the nomenclature, including description and sources of the quantities used or computed. The label “Monte Carlo” means that the quantity is pre-calculated and tabulated, “Calculation” means that this value is calculated in the coupled calculational tool implemented in this work. The literature sources are given as references (see Tables A.1 and A.2).

## References

Bell, G., Glasstone, S., 1970. Nuclear Reactor Theory. Van Nostrand Reinhold Company, New York, USA.

CEA, EDF, 2012. OpenCascade, Salome Platform Documentation, SALOME Platform. <<http://www.salome-platform.org/user-section/online-documentation>>.

Demazière, C., 2013. Multi-physics modelling of nuclear reactors: current practices in a nutshell. *Int. J. Nucl. Energy Sci. Technol.* 7 (4), 288–318.

Demazière, C., Mattson, H., 2006. Determination of the fuel heat transfer dynamics via cfd for the purpose of noise analysis. NPIC&HMIT 2006, Albuquerque, USA, November 12–16.

Ferziger, J., Peric, M., 2002. Computational Methods for Fluid Dynamics. Springer.

Fink, L., 2000. Thermophysical properties of uranium dioxide. *J. Nucl. Mater.*, 279.

International Atomic Energy Agency, 2006. Thermophysical properties database of materials for light water reactors and heavy water reactors. Technical report. <[http://www-pub.iaea.org/MTCD/publications/PDF/te\\_1496\\_web.pdf](http://www-pub.iaea.org/MTCD/publications/PDF/te_1496_web.pdf)>.

Kochunas, B., Stimpson, S., Collins, B., Downar, T.J., Brewster, R., Baglietto, E., Yan, J., 2012. Coupled full core neutron transport/cfd simulations of pressurized water reactors, PHYSOR 2012, Knoxville, Tennessee, USA, April 15–20.

Leppänen, J., 2012. Serpent – a Continuous-energy Monte Carlo Reactor Physics Burnup Calculation Code, VTT Technical Research Centre of Finland.

Modest, M., 2003. Radiative Heat Transfer. Academic Press, San Diego, USA.

Murphy, E., Havelock, F., 1976. Emissivity of zirconium alloys in air in the temperature range 100–400 °C. *J. Nucl. Mater.*, 60.

Nakamura, S., 1977. Computational methods in engineering and science with applications to fluid dynamics and nuclear systems. John Wiley & Sons Inc., New York, USA.

OpenFOAM, 2011. OpenFOAM Programmer’s Guide, OpenCFD Limited.

OpenFOAM, 2012. OpenFOAM User’s Guide, OpenCFD Limited.

Panton, R., 2005. Incompressible Flow, third ed. Wiley.

Patankar, S., Spalding, D., 1972. A calculation procedure for heat, mass and momentum transfer in three dimensional parabolic flows. *Int. J. Heat Mass Transfer* 15, 1787–1806.

NIST Webbook, 2011. Thermophysical Properties of Fluid Systems, <<http://webbook.nist.gov/chemistry/fluid/>>.

Todreas, N., Kazimi, M., 1993. Nuclear Systems I: Thermal Hydraulic Fundamentals. Taylor & Francis, Levittown, USA.

Versteeg, H.K., Malasekera, W., 2007. An Introduction to Computational Fluid Dynamics, second ed. Prentice Hall.

# **Paper II**

*“Influence of an  $S_N$  solver in a fine-mesh neutronics/thermal-hydraulics framework”*

K. Jareteg, P. Vinai, C. Demaziére, S. Sasic

*PHYSOR 2014, September 28 - October 3, Japan (2014)*



# INFLUENCE OF AN $S_N$ SOLVER IN A FINE-MESH NEUTRONICS/THERMAL-HYDRAULICS FRAMEWORK

**Klas Jareteg, Paolo Vinai and Christophe Demazière**

Division of Nuclear Engineering, Department of Applied Physics

Chalmers University of Technology, Gothenburg, Sweden

[klas.jareteg@chalmers.se](mailto:klas.jareteg@chalmers.se), [vinai@chalmers.se](mailto:vinai@chalmers.se), [demaz@chalmers.se](mailto:demaz@chalmers.se)

**Srdjan Sasic**

Division of Fluid Dynamics, Department of Applied Mechanics

Chalmers University of Technology, Gothenburg, Sweden

[srdjan@chalmers.se](mailto:srdjan@chalmers.se)

## ABSTRACT

In this paper a study on the influence of a neutron discrete ordinates ( $S_N$ ) solver within a fine-mesh neutronic/thermal-hydraulic methodology is presented. The methodology consists of coupling a neutronic solver with a single-phase fluid solver, and it is aimed at computing the two fields on a three-dimensional (3D) sub-pin level. The cross-sections needed for the neutron transport equations are pre-generated using a Monte Carlo approach. The coupling is resolved in an iterative manner with full convergence of both fields. A conservative transfer of the full 3D information is achieved, allowing for a proper coupling between the neutronic and the thermal-hydraulic meshes on the finest calculated scales. The discrete ordinates solver is benchmarked against a Monte Carlo reference solution for a two-dimensional (2D) system. The results confirm the need of a high number of ordinates, giving a satisfactory accuracy in  $k_{\text{eff}}$  and scalar flux profile applying  $S_{16}$  for 16 energy groups. The coupled framework is used to compare the  $S_N$  implementation and a solver based on the neutron diffusion approximation for a full 3D system of a quarter of a symmetric, 7x7 array in an infinite lattice setup. In this case, the impact of the discrete ordinates solver shows to be significant for the coupled system, as demonstrated in the calculations of the temperature distributions.

**Key Words:** **coupled neutronics/thermal-hydraulics, discrete ordinates method, fine-mesh, sub-pin cross-section generation**

## 1. INTRODUCTION

The simulation of a Light Water Reactor requires different fields of physics to be considered. In the nuclear reactor core, the interwoven dependence of the neutron density, of the properties of the coolant and of the properties of the fuel pins constitutes a challenging task, from the point of view of both the modelling and numerical algorithms. Often a "divide-and-conquer" strategy is applied to simplify the problem and make it feasible to solve. Therefore, instead of following an implicit approach, the multiphysics problem is split in separate parts and solved sequentially. To regain the dependencies between the different fields, an explicit coupling is introduced. Commonly, an

iterative procedure is used to update the interdependencies. Although some multiphysics aspects are recovered, the iterative approach often comes with a price. Only macroscopic dependencies are re-introduced, whereas the different physical phenomena were potentially solved at a meso- or even microscale level. Such and other simplified schemes can lead to inconsistent coupling or a coupling that does not resolve the small scales.

Recently, additional focus has been given to direct methodologies where the multiphysics problem is solved in a more tightly coupled manner. Such a work is often characterized by the use of modern algorithms and massively parallel computers. The type of issues tackled is not only related to neutronic/thermal-hydraulic coupling, but also to other phenomena, e.g. fluid-structure interaction, microscopic modelling of fuel behaviour, and others [1]. MOOSE, developed at INL [2], is one example of a framework targeting a wide range of reactor applications. For this and similar codes the problems are typically solved in an implicit manner, often using non-linear methodologies. Other works have aimed more specifically at the neutronic/thermal-hydraulic coupling, using a combination of different softwares with a less integrated approach. Examples of this kind of strategies include coupling DeCart and StarCCM+ [3] and coupling MCNP and StarCCM+ [4].

In addition, the multiphysics modelling should aim at different level of details depending on the application. Recent work focused on a wide range of topics, such as microscopic development of fuel material properties [5], influence of fuel spacer grids on the fluid flow [6] or full-core coupled neutronics and thermal-hydraulics [3]. Resolving the full system, including the mentioned and some other phenomena, in a fully coupled solver is still out of reach, and represents a future major challenge [7].

Furthermore, the interplay between neutronics and thermal-hydraulics is of interest for an accurate understanding of phenomena at the level of both the core and the fuel assemblies. In fact, high-resolution predictions of parameters for the fuel assembly and fuel pin (e.g. temperature distributions) can provide useful information from a design and safety perspective. Again, this resolution is not affordable at the full-core level, but can contribute to important local information within a fuel bundle.

In this research project, efforts are addressed to develop a framework for treating the neutronics and thermal-hydraulics on a fine-mesh in a coupled manner. This is achieved with an iterative scheme, but applied directly on the fine-mesh. Therefore information at the smaller scales is preserved and a tighter coupling of the problem is allowed. In our previous work [8, 9], a system of a limited number of fuel pins was solved using an integrated fully 3D coupled tool. This was done by modelling single phase flow conditions, the heat transfer within the fuel pins and the coolant, and the neutron transport in a multigroup diffusion framework.

The current paper presents a further step in which the diffusion approximation is replaced with a fine, unstructured mesh discrete ordinates method. The underlying theory, discretization and implementation of  $S_N$  are described in Section 2, while the previously implemented single-phase flow methodology is briefly outlined in Section 3. The method and algorithm used for the iterative coupling scheme are given in Section 4. The  $S_N$  solver is tested against a 2D case (Section 5.1), and it is then applied to a 3D coupled problem (Section 5.2).

## 2. NEUTRON TRANSPORT METHODOLOGY

### 2.1. Discrete Ordinates Method

In our previous work the neutronics was based on a multigroup diffusion solver [8]. In order to resolve the angular dependence of the neutron flux, a higher order method in angle is required. The neutron transport equation is here solved using the discrete ordinates method. The theory of the method is well described and analyzed in the literature (e.g [10, 11]), and the transport equation for one ordinate and one energy group can be written:

$$\Omega_m \cdot \nabla \Psi_{m,g} + \Sigma_{T,g} \Psi_{m,g} = S_{m,g} + \frac{1}{k} F_{m,g} \quad (1)$$

where standard notations are used. The scattering source term,  $S_{m,g}$ , is for anisotropic scattering given by:

$$S_{m,g} \equiv \sum_{l=0}^L (2l+1) \sum_{m'}^M P_l(\Omega_m \cdot \Omega_{m'}) w_{m'} \sum_{g'=1}^G \Sigma_{s,l,g' \rightarrow g} \Psi_{m',g'} \quad (2)$$

where  $P_l$  are the Legendre polynomials and  $w_{m'}$  are the ordinate weights. The fission source term  $F_{m,g}$  is given by:

$$F_{m,g} \equiv \chi_g \sum_{m'}^M w_{m'} \sum_{g'=1}^G \nu_{g'} \Sigma_{f,g'} \Psi_{m',g'} \quad (3)$$

The scalar flux is given by the weighted sum of all the ordinates, such that:

$$\Phi_g = 4\pi \sum_m^M w_m \Psi_{m,g} \quad (4)$$

using standard notations for all quantities.

#### 2.1.1. Quadrature sets

Given the equation (1), a set of ordinates and related fluxes  $\Psi_{m,g}$  is used to approximate the angular flux. For each ordinate, a direction and weight must be determined, and the set could be chosen based on different criteria. As exemplified and analyzed elsewhere in literature (e.g. [12]) the selection of the set has a major influence on the accuracy of  $k_{\text{eff}}$  and the scalar flux solution, in particular for heterogeneous systems. As pointed out in [13], an optimal quadrature set applicable to all cases can not be predicted, but will depend on the type of problem. The weights and directions employed in this work correspond to the level symmetric set as presented in [14].

#### 2.1.2. Expansion on spherical harmonics

As seen from eq. (2), when calculating the scattering contribution to one ordinate  $m$ , all other ordinates have an influence. In general, each scattering source term  $S_{m,g}$  requires  $(L \times M \times G)$

calculations. Considering that the group-to-group scattering is identically zero for most of the pairs of groups, the computational burden is somewhat lower. Still, as the connectivity of the mesh is typically on the order of 10, the left hand side of eq. (1) requires much fewer calculations than the right-hand side.

To reduce the number of summations for the scattering source, a projection on the spherical harmonics is used. The formulation is based on the real spherical harmonics functions as given in [14]. As fewer expansion coefficients are required for the spherical harmonics functions, fewer summations are needed on the right-hand side. Further, for a lower  $L$ , a lower spherical harmonics expansion can be used, again reducing the number of calculations. Also, if applying an acceleration to the lowest expansion using e.g. diffusion synthetic acceleration (DSA), it is desirable to formulate the right-hand side using an expansion on the spherical harmonics. Here, such an acceleration has not been implemented.

### 2.1.3. Discretization and implementation

In the present work the finite volume method (FVM) is applied to discretize all equations. Using the FVM, only the streaming term in eq. (1) will give a transfer between the different computational cells. Integrating eq. (1) over the current cell gives:

$$\int_V \Omega_m \cdot \nabla \Psi_{m,g} dV + \int_V \Sigma_{T,g} \Psi_{m,g} dV = \int_V S_{m,g} dV + \int_V \frac{1}{k} F_{m,g} dV \quad (5)$$

where  $V$  corresponds to the control volume (the cell in the mesh). The streaming term is transformed to a sum over the surfaces using Gauss theorem which results in:

$$\int_V \Omega_m \cdot \nabla \Psi_{m,g} dV = \int_S \Omega_m \Psi_{m,g} \cdot \mathbf{n}_f dS = \sum_f \Psi_{m,g,f} \Omega_m \cdot \mathbf{n}_f A_f \quad (6)$$

where index  $f$  corresponds to a face value and  $\mathbf{n}_f$  corresponds to the unit outward normal to face  $f$ . A more detailed description of the finite volume method and the general discretization procedure can be found in [15].

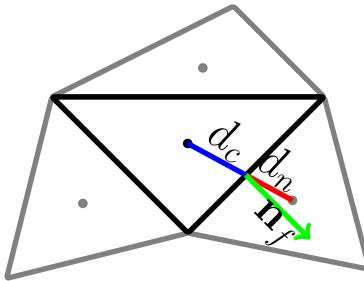
In a structured mesh, higher order discretization schemes can be achieved using not only neighbour cell values but also second neighbours. One example of a higher order scheme would be the diamond differencing scheme. Based on the face-based computational molecule used in the applied FVM implementation of unstructured meshes, the use of second neighbour values is more complicated. In the case of such an face-based discretization strategy, only the first neighbour, directly sharing a face with the considered cell, is available. This is illustrated in Figure 1. To determine the face value of the flux ( $\Psi_{m,g,f}$ ), an interpolation using only the cell value and the first neighbour value is used. This interpolation can be written:

$$\Psi_{m,g,f} = f \Psi_{m,g}^C + (1 - f) \Psi_{m,g}^N \quad (7)$$

where  $C$  denotes the current cell and  $N$  denotes a neighbouring cell. Choosing the interpolation scheme depending on the current ordinate ( $\Omega_m$ ), we can write:

$$f = \begin{cases} 1 & \text{if } \mathbf{n}_f \cdot \Omega_m \geq 0, \\ 0 & \text{if } \mathbf{n}_f \cdot \Omega_m < 0 \end{cases} \quad (8)$$

which corresponds to a first order step differencing scheme. While the step differencing scheme will be unconditionally stable, higher order differencing scheme will only be stable under certain condition [16].



**Figure 1.** Finite volume discretization for an unstructured mesh

#### 2.1.4. Sweep algorithm

The solution algorithm for each direction and each group is based on a sweep over all cells in the mesh. This corresponds to a Gauss-Seidel solver with the cells ordered such that when updating the current cell, all upstream cells are already updated. Thus, to update eq. (1) for a specific direction and energy group, a single sweep over all cells is necessary.

The sweeping order of the unstructured mesh is calculated at the start of the simulation. Since in the present case the domain is decomposed and parallelization used, the sweeping order is determined separately for each decomposed domain (i.e. each CPU in the parallel runs) and each direction. Algorithms to find a sweeping order are found in e.g. [17], and are based on iteratively finding cells in which all upstream faces are updated, allowing the downstream faces to be calculated. The mesh used consists of strictly convex cells, either prisms or hexahedrals, and thus any cyclic dependencies are avoided. Furthermore, the domains are split in strictly convex domains, avoiding cyclic dependencies between the different domains.

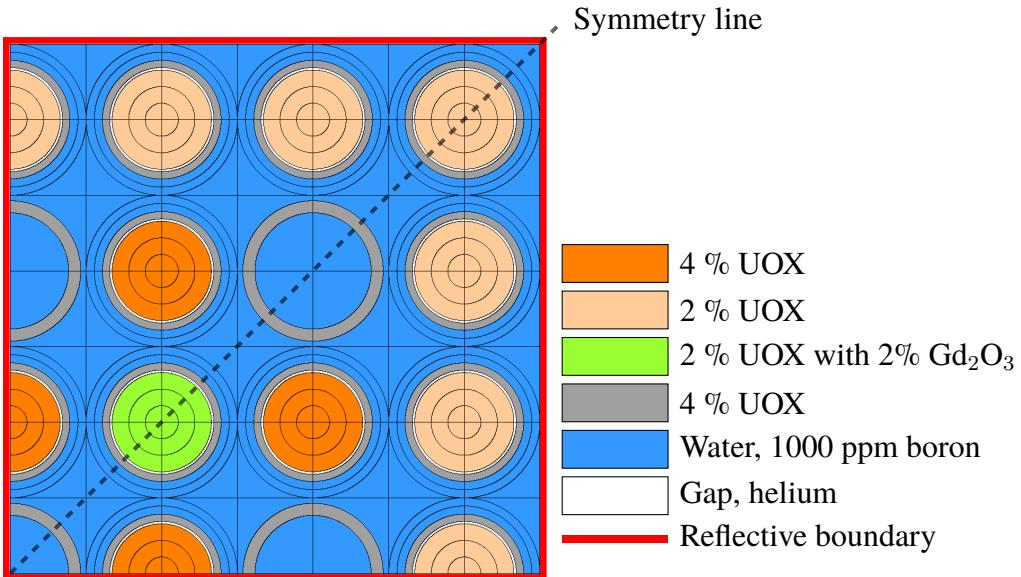
To conserve the convergence of the domain decomposed case, a modified order of updating the different directions is needed. This can be achieved by solving different directions at different CPUs, concurrently. Such scheme is out of scope for the current coupled solver. Instead all CPUs solve the same direction simultaneously. This introduces a minor penalty for the parallelized case, not further analyzed.

## 2.2. Cross-section Generation

The cross-sections are generated in a two-dimensional system. The horizontal geometry corresponds directly to the assembly simulated, whereas an infinite system is assumed in the axial direction. In the current methodology the cross-sections are pre-generated, and they are tabulated according to the position in the horizontal plane and the thermophysical state point. In the present case of single phase fluid, only temperature is used to determine the thermophysical state point.

Since the pressure drop over a fuel assembly is small, the pressure influence on the density in the moderator is of minor importance, and a fixed pressure is used when calculating the material properties.

To carry-out discretization in space, the horizontal plane is split into a number of regions. The union of those regions covers the simulated system completely, without any overlap, and for each of them separate homogenized cross-sections and macroscopic data are generated. The azimuthal angle is divided by a multiple of four, and each material region is further divided in a certain number of radial regions. Generating the data with these splits allows for a sub-pin-cell resolution in the cross-sections. An example of the chosen discretization of the geometry can be seen in Figure 2. The different colors correspond to different material properties, as explained in the legend. Each enclosed area will have a separate table of homogenized cross-sections and macroscopic data.

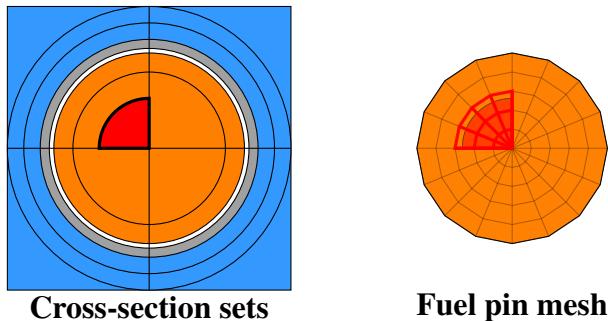


**Figure 2.** Discretized geometry for generation of cross-sections.

All the macroscopic parameters are calculated using the Monte Carlo code Serpent [18]. The necessary input for Serpent is built with a separate written Python application. This pre- and post-processing code reads a configuration file specifying necessary discretization, geometry and lattice pattern. The preprocessing part of the code produces a Serpent input with each discrete region implemented as one volume and universe for cross-section homogenization. After Serpent is run, the post-processing part of the code analyzes the output and generates the tables of data for each geometrical region. Further options are introduced in the utility code, so that one can control material types, temperature profiles for all material regions, number of particles and generations simulated, among others.

The same pre- and post-processing code also provides the geometrical description used to determine which cells in the unstructured mesh (for the coupled calculations) will fall in to each cross-section region. Such sets handle the mapping between the cross-section geometry and the unstructured mesh. This allows for a fully automatic transfer of information, hiding the mapping to the user. A sketch example of how such mapping is done is seen in Figure 3. The red area corresponds to a region in the cross-section discretization. This region is mapped to the example fuel pin mesh,

used in the coupled calculations. All cells with centres within the red region will be using the cross-sections from the table corresponding to this region.



**Figure 3.** Example of mapping between cross-section discretization and an example mesh for the fuel pin.

### 2.2.1. Cross-section interpolation

From the mapping, each cell is pointed to the cross-section table for the cross-section region it belongs to. Given the correct table, each cell in the unstructured mesh will use the local (cell) temperature to interpolate between the existing temperatures. This is done each time the temperature has been recalculated in the thermal-hydraulic solver (further described in Section 4). At the point of performing the interpolation, the global temperature and neutron flux profiles are both disregarded. This means that, although generated for the full influence of the complete horizontal geometry (as in Figure 2) and temperature profile, the cross-sections are used locally, independently of the surrounding conditions. The influence of such an approximation will be considered in later work.

Both the mapping of cells to cross-sections regions and the interpolation in the tables are done in the coupled code responsible for all deterministic calculations. As a result, a high-resolution set of cross-sections is achieved for a small cost in terms of CPU-time.

## 3. SINGLE-PHASE FLUID DYNAMICS AND HEAT TRANSFER

The single-phase fluid dynamics is solved by applying the continuity, momentum and energy equations formulated from first principles, and using a fine-mesh method. In the fluid regions, the flow is assumed incompressible and the pressure and velocity are resolved using the SIMPLE algorithm [19], with the addition of buoyancy and expansion of the heated fluid. All material data are temperature dependent, including the viscosity and thermal conductivity. However, the change in density is only considered in the fluid regions. In the presented methodology, the thermal expansion in the fuel is disregarded.

The energy conservation equation is implemented not only in the fluid regions, but also in the solid regions. In order to solve these equations using a monolithic approach, all energy equations

are formulated in terms of the temperature. From this, all regions can be implicitly coupled in a single matrix system. The implicit approach avoids iteration for the conjugate heat transfer problem between the fuel pins and the moderator.

As the requirements for the resolution of the meshes are different for the neutronics and the thermal-hydraulics, the fluid dynamics and heat transfer are solved on their own meshes. This has implications for the transfer of information between the neutronics and thermal-hydraulics. Such issues are discussed further in Section 4.2.

A full description of the models and the implementation of the single-phase fluid dynamics and the heat transfer can be found elsewhere [8].

## 4. COUPLING IMPLEMENTATION AND MULTIPHYSICS ASPECTS

For this work, we use OpenFOAM-1.6-ext[20] to implement, discretize and solve the multiphysics problem. The software provides general structures for solving partial differential equations (PDEs), which are discretized with the finite volume method. The systems of linear equations can be solved using intrinsic matrix solvers and full parallel support is implemented using domain decomposition and MPI.

OpenFOAM-1.6-ext is an open-source software, with the C++ source code freely distributed. To succeed with a high performance coupled multiphysics framework, the availability and modifiability of the source is of major importance. OpenFOAM-1.6-ext is modular, based on the object-orientation paradigm complemented by the use of templates. The extensions made for the neutronics, especially for  $S_N$ , are added as new modules, using a similar structure as in the original source code.

### 4.1. Coupled Algorithm

In this work the steady-state coupled problem is computed using the iterative algorithm displayed in Figure 4. To ensure a realistic power profile, the neutronics, solved by the power iteration method, is first converged. This first solution is based on the initial guess of the temperature in the system. In an equivalent manner, the pressure and velocity are allowed to converge in two outer iterations before the temperature equation is added.

Full convergence is reached by ensuring that the residuals are lower than a set of criteria in both the neutronics and thermal-hydraulics. The residual error is checked when entering the neutronic and thermal-hydraulic modules, respectively. Due to the large number of equations solved, many different tolerances are applied. Tuning such parameters can substantially improve the performances of the iterative algorithm.

Based on the present implementation and under the currently run conditions, there is no need to apply under-relaxation between the neutronics and thermal-hydraulics. However, for the pressure

and velocity coupling an under-relaxation is necessary. Such behavior is typical for the SIMPLE algorithm. Yet, no error will be transferred to the neutronics since the fluid pressure and velocity couplings are fully converged in each thermal-hydraulic iteration.

## 4.2. Mesh to Mesh Interpolation

As different meshes are used for the thermal-hydraulics and the neutronics, accurate and conservative transfer of shared quantities is of vital importance. Considering that the finite volume method is applied, an algorithm conserving each quantity based on the cell volume average is needed. To achieve this we apply a mesh intersection algorithm. This method is based on finding the volumetric overlap between cells in the different meshes. The cell intersections are computed at the start of the simulation.

Given that a cell  $i$  in the first mesh overlaps with a set of cells  $J$  in the second mesh, a conservative transfer from the second to the first mesh is achieved using volumetric intersections such that:

$$c_i = \sum_{j \in J} \frac{c_j I_{ij} V_j}{V_i} \quad (9)$$

where  $c$  is some extensive property to be transferred,  $I_{ij}$  is the volumetric intersection of cell  $j$  with cell  $i$  and  $V_i$  and  $V_j$  are the cell volumes of cell  $i$  and cell  $j$ , respectively.

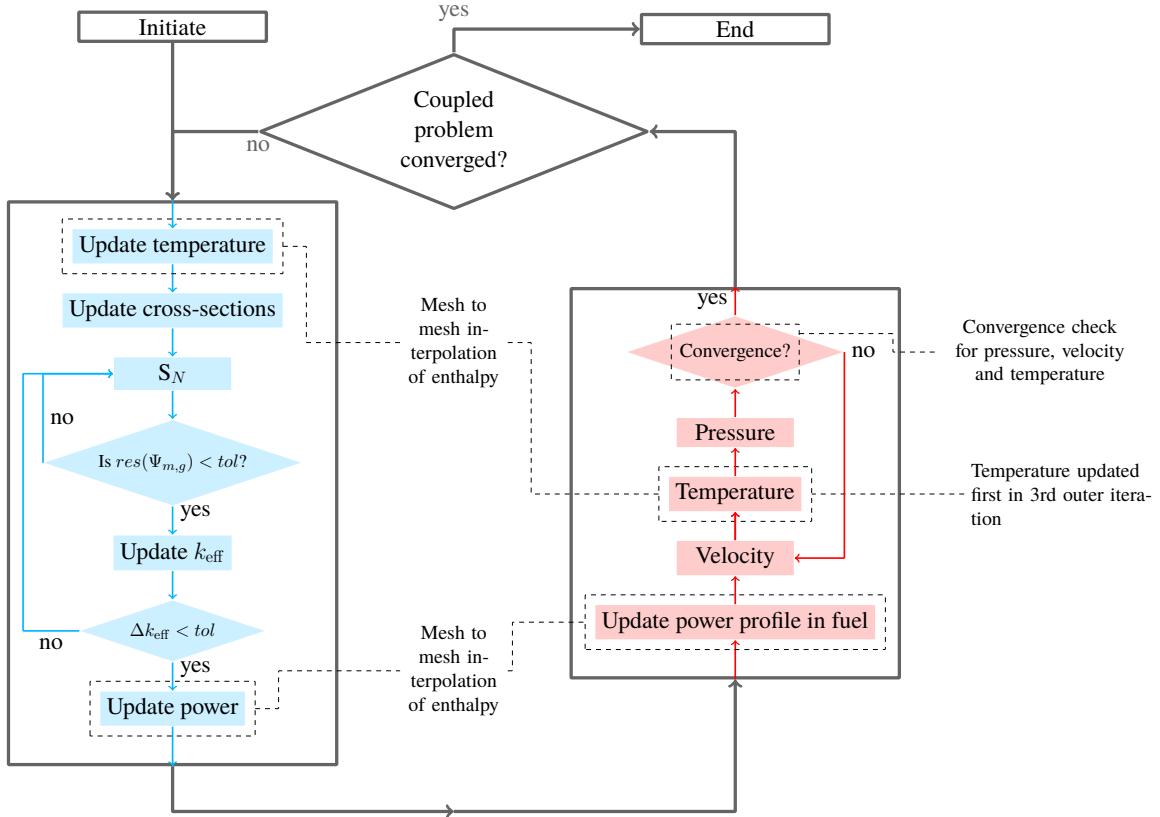
The intersection calculation algorithm is implemented for arbitrary convex polygons in the horizontal plane, allowing an extruded mesh in the axial direction. The calculation is applied on each decomposed domain separately and causes no significant extra burden in terms of computational time. This algorithm allows for two different meshes to be applied with a fully automatic mesh mapping between them.

## 5. RESULTS

### 5.1. Benchmarking the Discrete Ordinates Implementation

First the discrete ordinates solver is applied to a two-dimensional benchmark case. The test case is defined by the material composition and geometry already presented in Figure 2. The mesh applied for the neutronics is presented in Figure 5, and is created by a Python application, with a format consistent with OpenFOAM. The cross-sections are generated for an isothermal system at 540 K. Even though a system with constant temperature is considered, there are large variations of the cross-sections because of the heterogeneity of the system.

Reference solutions are extracted from Serpent while producing the cross-sections. A uniform grid detector, with 16x16 cells per pin cell is used to record the scalar flux. It should be noted that the discrete ordinates solution is calculated with a discrete set of energy groups (2, 4, 8 or 16) whereas the Serpent reference solution is using the internal continuous approach.



**Figure 4.** Coupled algorithm for the neutronics and the thermal-hydraulics.

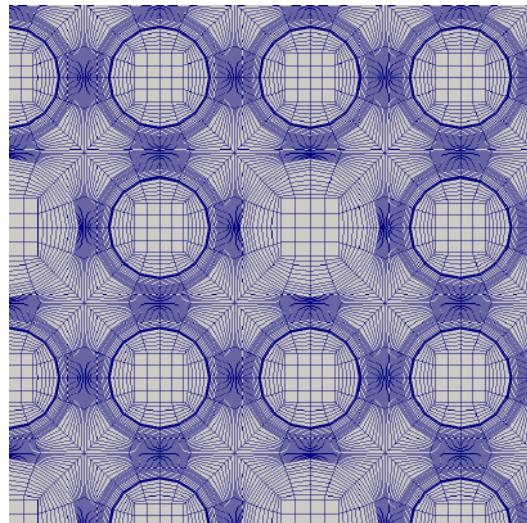
Table I summarizes the different test cases and the results. Considering first the error in  $k_{\text{eff}}$ , a better result is reached increasing the number of ordinates. However, as seen in comparing  $S_8$  and  $S_{16}$ , this increase in order does not give any major difference. Increasing the number of groups up to eight gives a better  $k_{\text{eff}}$  as compared to the reference calculation. The results for 16 groups are similar to the results for eight groups.

The flux error is calculated along the symmetry line given in Figure 2 as the average error compared to the reference solution:

$$\Delta\Phi_g = \sqrt{\frac{\sum_{i=1}^N (\Phi_{g,i} - \Phi_{g,i}^{\text{ref}})^2}{N}} \quad (10)$$

where  $i$  is taken over all the Serpent detector cells along the symmetry line, with a corresponding point from the  $S_N$  solution. In total  $N$  points are compared with  $N = 56$  for the 16x16 detector cells per pin cell.

In contrast to the error in  $k_{\text{eff}}$ , the flux error in the lowest energy group decreases when going from  $S_8$  to  $S_{16}$ . This can be explained by the so-called ray effect, which gives unphysical artifacts in the scalar flux profiles. These artifacts appear due to the inability of the quadrature set to represent the true angular flux [21]. The effect is clearly distinguishable for the low orders, whereas for  $S_{16}$  it is much less significant, as shown in Figure 6. In the cases of 8 and 16 groups, the largest error in flux is seen for the highest energy group ( $g = 1$ ). To further decrease the flux error a higher number of



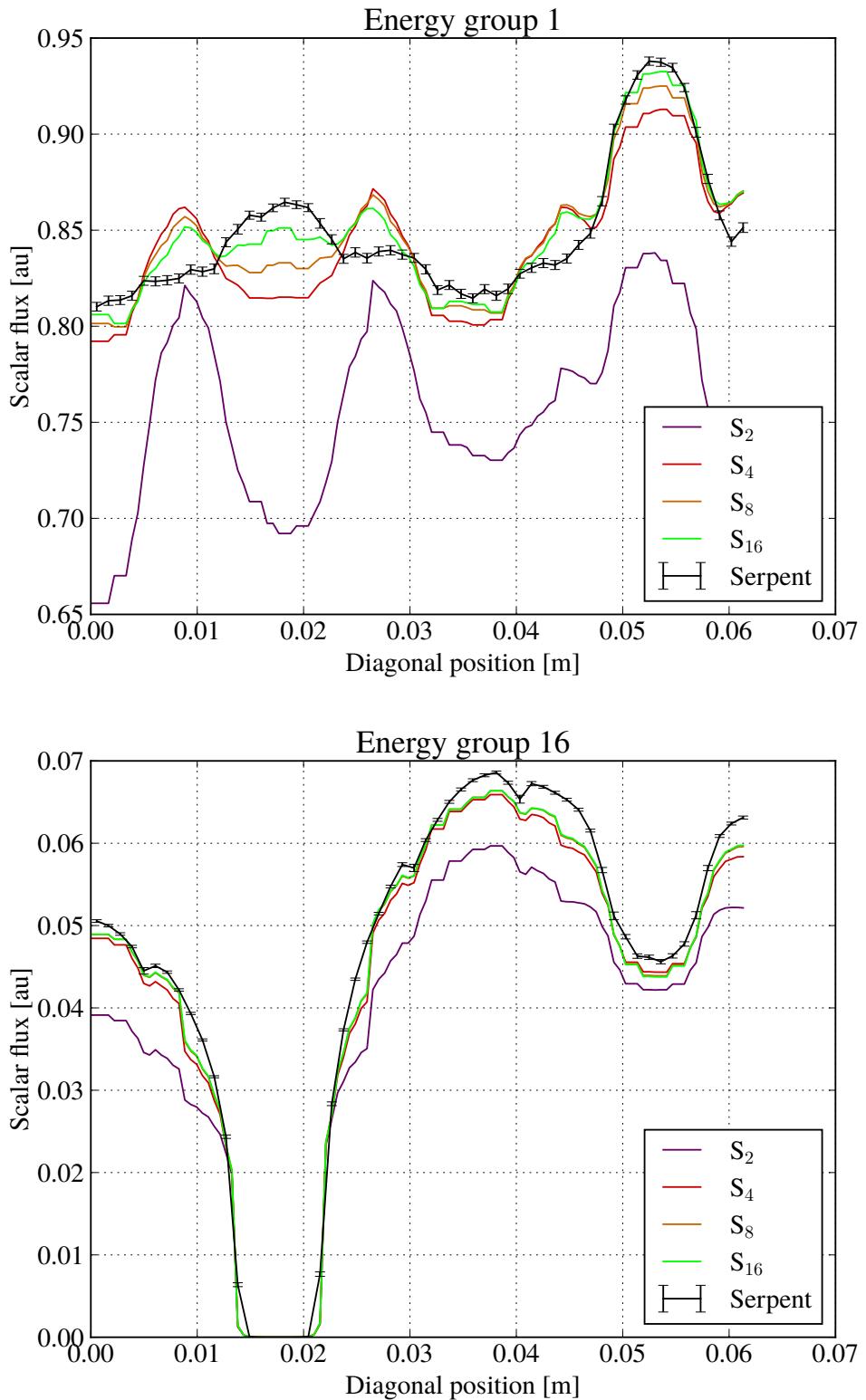
**Figure 5.** Horizontal mesh discretization of a 7x7 pin system, using quarter symmetry.

ordinates are needed, as to better reproduce the angular flux.

## 5.2. Application of $S_N$ in the Coupled Framework

The discrete ordinate solver was also coupled to the thermal-hydraulic solver according to the scheme described in Section 4. To assess the influence of the transport solver on the fine-mesh system, the coupled calculations are run on a system with 1 m in active core height and with 0.2 m of top and bottom reflectors. The horizontal meshes are seen for the thermal-hydraulics and the neutronics separately in Figure 7. In the axial direction there are a total of 140 layers in the thermal-hydraulics mesh and 30 layers in the neutronics mesh. In total 172000 cells are used in the neutronics mesh and 1.28 M cells in the moderator.

The cross-sections are generated for a set of temperature profiles to be able to simulate the temperature variation in the system. The profiles are chosen to correspond to a set of different power profiles in the assembly. A radial profile is assumed in the fuel pins.



**Figure 6.** Comparison of different number of ordinates plotted along the symmetry line (see Figure 2) for the highest ( $g = 1$ ) and lowest energy group ( $g = 16$ ). The error bars on the Serpent solution indicates the error corresponding to  $1\sigma$ .

**Table I.** Comparison of the error in  $k_{\text{eff}}$  and scalar flux for different quadrature orders and number of groups for the 2D reference system. The Serpent reference solution gives  $k_{\text{eff}} = 0.93130 \pm 0.00009$ , where the error is given for  $1\sigma$ .

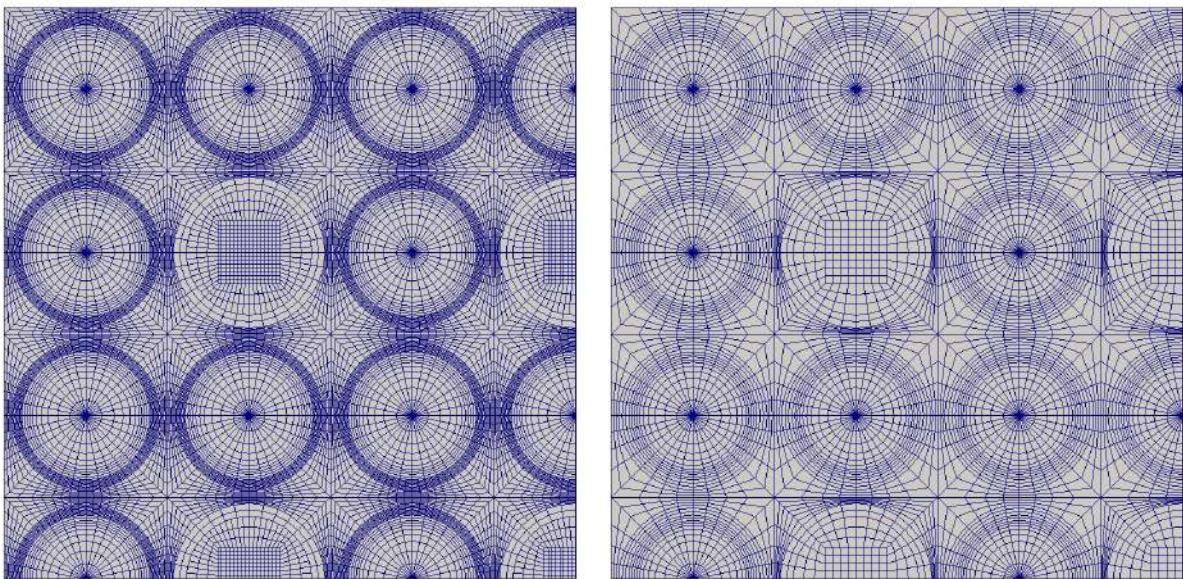
G	N	$k_{\text{eff}}$	$\Delta k_{\text{eff}}$ [pcm]	$\Delta\Phi_1$	$\Delta\Phi_G$
2	2	0.91262	-1868	0.03100	0.01957
2	4	0.91934	-1197	0.00640	0.00875
2	8	0.91868	-1263	0.00635	0.00777
2	16	0.91800	-1330	0.00730	0.00780
4	2	0.91134	-1996	0.09874	0.09032
4	4	0.91972	-1158	0.02280	0.03658
4	8	0.92001	-1129	0.01621	0.03146
4	16	0.91990	-1140	0.01086	0.03117
8	2	0.92230	-900	0.10246	0.02185
8	4	0.92833	-298	0.02418	0.00844
8	8	0.92861	-270	0.01727	0.00718
8	16	0.92852	-278	0.01153	0.00710
16	2	0.92178	-953	0.09978	0.00810
16	4	0.92775	-355	0.02318	0.00306
16	8	0.92808	-322	0.01684	0.00258
16	16	0.92796	-335	0.01160	0.00255

**Table II.** CPU time for the 3D cases. All cases were parallelized using 32 CPUs.

Energy groups	Methodology	Wall-clock time [s]	Total CPU-time [h]
4	Diffusion	5998	53.3
4	$S_8$	9631	85.6
8	Diffusion	6039	53.7
8	$S_8$	15434	137.2

Calculations done with different approximations are compared, namely diffusion theory in 4 and 8 energy groups and  $S_N$  theory in 4 and 8 energy groups. The applied CPU times are given in Table II. To characterize the influence on the coupled framework, not only the neutronics behaviour should be compared, but also the results for the thermal-hydraulics, showing the implicit dependence on the neutronic method used. As an example, the calculated temperature distributions for the different cases can be seen in Figure 8. The relative temperature variation shows significant discrepancies for the diffusion solver in comparison to the transport solver. Primarily the fuel temperature is much lower when using the neutron diffusion solver for the coupled calculations and thus the diffusion-based results are not conservative.

## 6. CONCLUSION



**Figure 7.** Horizontal meshes for thermal-hydraulics (left) and neutronics (right) for the 3D case.

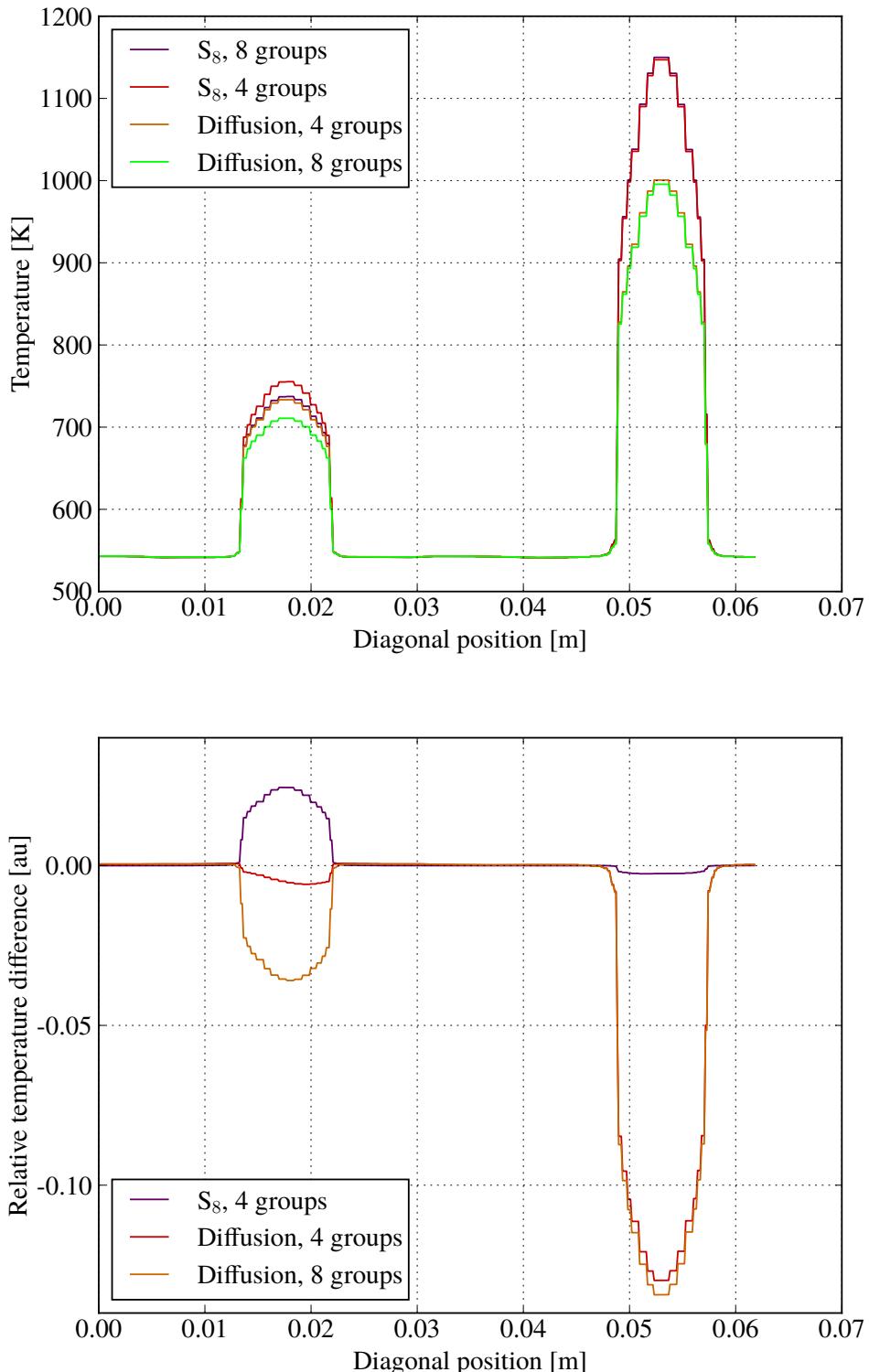
We presented a solver based on the discrete ordinates method within a fine-mesh coupled framework for neutronics and thermal-hydraulics. The implementation was described together with the procedure to generate the cross-sections by making use of a Monte Carlo method. The iterative algorithm was illustrated and accompanied by an explanation of the field interpolation, fully conserving the transferred data at the fine-mesh level.

Applying the  $S_N$  solver to a two dimensional, strongly heterogeneous system using the fine-mesh uncoupled neutronic solution procedure identifies the need of a large ordinate set as well as a large number of neutron groups. The results from the three dimensional, coupled calculations emphasizes the difference in using a neutron transport solver method in comparison to the diffusion method, also for the combined neutronics/thermal-hydraulics problem. A significant underestimation of the fuel temperature is seen for the diffusion solver in comparison to the discrete ordinates solver.

The influence of the full-core neutronics will be investigated in the future, so that proper boundary conditions can be modelled for the small system solved for in our test problem. Future research is planned on complementing the single-phase fluid solver with a two-phase solver, aimed at performing the coupled analysis with a fine-mesh resolved void fraction.

## ACKNOWLEDGMENTS

The Swedish Center for Nuclear Technology (SKC) is acknowledged for financially supporting this PhD project. This work is also part of the DREAM task force (Deterministic Reactor Modeling) at Chalmers University of Technology. The computations were performed on resources at Chalmers Centre for Computational Science and Engineering (C3SE) provided by the Swedish National Infrastructure for Computing (SNIC).



**Figure 8.** Temperature profile and relative temperature difference at mid-elevation along the horizontal symmetry line. The relative difference is calculated using  $S_8$  with 8 groups as reference.

## REFERENCES

- [1] P. J. Turinsky. “Advances in multi-physics and high performance computing in support of nuclear reactor power systems modeling and simulation.” *Nuclear Engineering and Technology*, **44**(2) (2012).
- [2] D. Gaston *et al.* “MOOSE: A parallel computational framework for coupled systems of non-linear equations.” *Nuclear Engineering and Design*, **239**(10): pp. 1768–1778. URL <http://www.sciencedirect.com/science/article/pii/S0029549309002635> (2009).
- [3] B. Kochunas *et al.* “Coupled Full Core Neutron Transport/CFD Simulations of Pressurized Water Reactors.” *PHYSOR 2012, Knoxville, Tennessee, USA, April 15-20* (2012).
- [4] J. Cardoni and Rizwan-uddin. “Nuclear Reactor Multi-Physics Simulations with Coupled MCNP5 and STAR-CCM+.” *M&C 2011, Rio de Janeiro, Brazil* (2011).
- [5] C. Newman, G. Hansen, and D. Gaston. “Three dimensional coupled simulation of thermomechanics, heat, and oxygen diffusion in UO<sub>2</sub> nuclear fuel rods.” *Journal of Nuclear Materials*, **392**(1): pp. 6–15 (2009).
- [6] J. Yan *et al.* “Coupled Computational Fluid Dynamics and MOC Neutronic Simulations of Westinghouse PWR Fuel Assemblies with Grid Spacers.” *NURETH-14, Toronto, Ontario, Canada, September 25-30* (2011).
- [7] K. Smith and B. Forget. “Challenges in the Development of High-Fidelity LWR Core Neutronics Tools.” *M&C 2013, Sun Valley, Idaho* (2013).
- [8] K. Jareteg, P. Vinai, and C. Demazière. “Investigation of the possibility to use a fine-mesh solver for resolving coupled neutronics and thermal-hydraulics.” *M&C 2013, Sun Valley, Idaho* (2013).
- [9] K. Jareteg, P. Vinai, and C. Demazière. “Fine-mesh deterministic modeling of PWR fuel assemblies: Proof-of-principle of coupled neutronic/thermal-hydraulic calculations.” *Annals of Nuclear Energy*, **68**(0): pp. 247 – 256. ISSN 0306-4549. URL <http://dx.doi.org/http://dx.doi.org/10.1016/j.anucene.2013.12.019> (2014).
- [10] E. W. Larsen and J. E. Morel. “Advances in discrete-ordinates methodology.” In: *Nuclear Computational Science*, (pp. 1–84). Springer. URL [http://link.springer.com/chapter/10.1007/978-90-481-3411-3\\_1](http://link.springer.com/chapter/10.1007/978-90-481-3411-3_1) (2010).
- [11] M. L. Adams and E. W. Larsen. “Fast iterative methods for discrete-ordinates particle transport calculations.” *Progress in nuclear energy*, **40**(1): pp. 3–159. URL <http://www.sciencedirect.com/science/article/pii/S0149197001000233> (2002).
- [12] J. Jarrell, T. Evans, and G. Davidson. “Discrete ordinate quadrature selection for reactor-based eigenvalue problems.” *M&C 2013, Sun Valley, Idaho* (2013).
- [13] I. K. Abu-Shumays. “Angular Quadrature for Improved Transport Computations.” *Transport Theory and Statistical Physics*, **30**(2-3): pp. 169–204. URL <http://dx.doi.org/10.1081/TT-100105367> (2001).

- [14] A. Hébert. *Multigroup Neutron Transport and Diffusion Computations*. Springer-Verlag. ISBN 978-0-387-98130-7. URL [http://dx.doi.org/10.1007/978-0-387-98149-9\\_8](http://dx.doi.org/10.1007/978-0-387-98149-9_8) (2010).
- [15] J. Ferziger and M. Peric. *Computational Methods for Fluid Dynamics*. Springer (2002).
- [16] H. K. Versteeg and W. Malasekera. *An introduction to Computational Fluid Dynamics*. Prentice Hall, second edition edition. ISBN 978-0-13-127498 (2007).
- [17] S. Plimpton *et al.* “Parallel Sn Sweeps on Unstructured Grids: Algorithms for Priorization, Grid Partitioning and Cycle Detection.” *Nuclear Science and Engineering*, **150**: pp. 267–283 (2005).
- [18] J. Leppänen. *Serpent - a Continuous-energy Monte Carlo Reactor Physics Burnup Calculation Code*. VTT Technical Research Centre of Finland (2012).
- [19] S. Patankar and D. Spalding. “A calculation procedure for heat, mass and momentum transfer in three dimensional parabolic flows.” *International Journal of Heat and Mass Transfer*, **15**: pp. 1787–1806 (1972).
- [20] OpenFOAM Extend Project. *OpenFOAM 1.6-ext*. OpenFOAM Extend Project. URL <http://sourceforge.net/projects/openfoam-extend/> (2012).
- [21] M. DeHart. *A discrete ordinates approximation to the neutron transport equation applied to generalized geometries*. Ph.D. thesis, Texas A&M University (1992).



## **Paper III**

*“Behaviour and Stability of the Two-Fluid Model for Fine-Scale Simulations of Bubbly Flow in Nuclear Reactors”*

H. Ström, S. Sasic, K. Jareteg, C. Demazi'ere

*International Journal of Chemical Reactor Engineering 13.4 (2015), pp. 449-459*



Henrik Ström\*, Srdjan Sasic, Klas Jareteg and Christophe Demazière

# Behaviour and Stability of the Two-Fluid Model for Fine-Scale Simulations of Bubbly Flow in Nuclear Reactors

DOI 10.1515/ijcre-2014-0171

**Abstract:** In the present work, we formulate a simplistic two-fluid model for bubbly steam-water flow existing between fuel pins in nuclear fuel assemblies. Numerical simulations are performed in periodic 2D domains of varying sizes. The appearance of a non-uniform volume fraction field in the form of meso-scales is investigated and shown to be varying with the bubble loading and the domain size, as well as with the numerical algorithm employed. These findings highlight the difficulties involved in interpreting the occurrence of instabilities in two-fluid simulations of gas-liquid flows, where physical and unphysical instabilities are prone to be confounded. The results obtained in this work therefore contribute to a rigorous foundation in on-going efforts to derive a consistent meso-scale formulation of the traditional two-fluid model for multiphase flows in nuclear reactors.

**Keywords:** two-phase flow, bubbly flow, gas-liquid flow, nuclear reactors

## 1 Introduction

The modeling of the current fleet of nuclear reactors traditionally relies on macroscopic approaches modeling different fields of physics, namely neutron transport, fluid dynamics, and heat transfer, among others. The reason behind the choice of macroscopic models lies in the fact that the verification and validation of models being used by the nuclear industry is a lengthy and

expensive process that involves many actors, such as research institutes, code manufacturers, plant and fuel manufacturers, and safety regulatory bodies. The industry is thus heavily relying on capitalization, i.e. on the continuous improvement of codes and models, rather than on developing of entirely new modeling approaches. In this respect, the advancement of computer resources in the 1970s resulted in the start-up of large developmental projects in the simulation of nuclear reactors. Due to the size of such systems though, and because of the rather limited computing power available at that time, macroscopic models represented the only possible modeling approach. In the area of two-phase flow, this resulted in the development of the two-fluid model at a macroscopic level, where the high-frequency and small scale filtered phenomena were artificially introduced via the use of experimentally-derived closure relationships.

The present work investigates the performance of the two-fluid model on smaller scales than what is typically used in the nuclear industry today. In the investigations reported hereafter, we formulate a simplistic two-fluid model for bubbly steam-water flow. Numerical simulations are performed in periodic 2D domains of varying sizes, investigating the spontaneous emergence of non-uniform bubble volume fraction fields in the form of meso-scales. The results obtained in this work contribute to a rigorous foundation in on-going efforts to derive a formulation of the traditional two-fluid model for multiphase flows in nuclear reactors that takes the effects of meso-scales into account.

It is well known, particularly from the literature on gas-solid flows, that solutions to the conventional two-fluid model exhibit the emergence of meso-scale structures at sufficiently high volume fractions of the dispersed phase. These structures can be seen as non-uniformities in the obtained volume fraction fields and thus represent structures whose characteristic sizes are of the order of several (more than ten) dispersed phase diameters. Falling in between the micro-scales (determined by the size of the dispersed phase) and the macro-scales

\*Corresponding author: **Henrik Ström**, Division of Fluid Dynamics, Department of Applied Mechanics, Chalmers University of Technology, SE-412 96 Göteborg, Sweden, E-mail: henrik.strom@chalmers.se

**Srdjan Sasic**, Division of Fluid Dynamics, Department of Applied Mechanics, Chalmers University of Technology, SE-412 96 Göteborg, Sweden

**Klas Jareteg, Christophe Demazière**, Division of Nuclear Engineering, Department of Applied Physics, Chalmers University of Technology, SE-412 96 Göteborg, Sweden

(determined by the size of the bounding geometry), these structures are termed meso-scale structures. The formation of these structures is due to clustering of the dispersed phase and takes place because of inherent instabilities of the system. Meso-scale structures may therefore manifest also under uniform flow conditions in unbounded domains. The occurrence of meso-scale structures in numerical simulations of two-phase flow using the two-fluid model was first investigated by Agrawal et al. (2001) and has since then received great interest. Investigations of instabilities and filtering approaches for taking the effects of unresolved meso-scale structures into account in coarse-grid two-phase simulations has also been investigated in great detail, mostly for gas-solid flows (Agrawal et al. 2001; Zhang and VanderHeyden 2001; Benyahia and Sundaresan 2012; Wang, van der Hoef, and Kuipers 2009) but also more recently for gas-liquid flows (Yang et al. 2011). It has been shown that the appearance of instabilities related to clustering and voidage formation in two-phase systems requires neither macroscopic shear nor the presence of boundaries (Agrawal et al. 2001). In fact, it is sufficient that there is an interaction between the inertia of the dispersed phase, gravity and the interphase drag (Agrawal et al. 2001).

The motion and stability of bubbly suspensions have been studied on single bubbles in periodic 2D- and 3D-domains (i.e. representing an infinite regular array of rising bubbles) (Sankaranarayanan and Sundaresan 2002; Sankaranarayanan et al. 2002), and it is known that the uniform bubbling state loses stability at some critical bubble volume fraction (Joshi et al. 2001; Mudde, Harteveld, and van den Akker 2009). Freely evolving swarms of near-spherical two-dimensional bubbles were also shown to produce meso-scale structures, although they rise faster than a regular array of bubbles at the same volume fraction (Esmaeeli and Tryggvason 1998).

Several different scenarios have been proposed for the underlying physical mechanisms responsible for a transition from a homogeneous to a heterogeneous bubbly flow. Sankaranarayanan and Sundaresan (2002) found that the hindered motion of the dispersed phase gives rise to vertically travelling waves (just as in gas-solid flows) whereas cooperative motion of the dispersed phase tends to create columnar structures. They also showed that a lift force is not necessary to trigger an instability. On the other hand, Lucas, Prasser, and Manera (2005) and Lucas et al. (2006) concluded that the lift force is of utmost importance, and that a positive sign of the lift force coefficient practically acts as a stability criterion. Finally, Monahan and Fox (2007) attributed the loss of stability to yet another mechanism – that

bubble wakes are suppressed as the gas holdup increases, which triggers the instabilities due to a decreased effective viscosity. Even though the experiments of Mudde, Harteveld, and van den Akker (2009) show that uniform bubbly flows are indeed unstable, theoretical works seem to predict lower critical gas fractions for the transition to unstable behaviour. We find this observation very interesting, as we believe that it hints at some of the difficulties in interpreting the occurrence of instabilities in two-fluid model predictions of gas-liquid flows.

In addition to the physical reasons for the unstable behaviours observed in gas-liquid two-phase flow, there are namely also well-documented mathematical difficulties associated with the two-fluid model as such. The most commonly used version of the two-fluid model for gas-liquid flows is based on an assumption of pressure equilibrium between the phases, which leads to a non-hyperbolic ill-posed model in the inviscid limit (Dinh, Nourgaliev, and Theofanous 2003; Yström 2001; Prosperetti and Satrape 1990; Ransom and Hicks 1984, 1988). Although it has been argued that the ill-posedness arises when local discontinuities (i.e. interfaces) are homogenized by an averaging procedure, there is still no general consensus on whether the non-hyperbolicity of the equation set reflects the presence of physical instabilities or whether instabilities are artificially augmented by an inadequate mathematical representation (Dinh, Nourgaliev, and Theofanous 2003; Prosperetti and Satrape 1990). In practice, the appearance of instabilities in numerical solutions to the two-fluid model is often suppressed by the use of excessively dissipative numerical schemes or by regularization of the physical model (Prosperetti and Satrape 1990; Pokharna, Mori, and Ransom 1997). Unfortunately, little is known about the effect of such measures on the simulation of large-scale flow oscillations over long times (Prosperetti and Satrape 1990; Pokharna, Mori, and Ransom 1997; Stewart 1986). Understanding the uncertainties inherent in these types of simulations is absolutely critical in an effort to use numerical results obtained with a two-fluid model at fine resolution to derive appropriate relations for the effects of unresolved meso-scale structures at a coarser resolution. Most notably, there is a significant risk that solutions of an ill-posed model might suffer from excessive numerical diffusion and/or exhibit unphysical instabilities (Dinh, Nourgaliev, and Theofanous 2003), which could inadvertently be transferred along to the meso-scale closures.

The aim of the current work is to investigate the performance of the two-fluid model for nuclear reactor applications. More specifically, the aim is to assess the dependence of the behaviour and stability of the two-

fluid model on the specifications of the computational setup and the case properties in a vein similar to that of previous studies related to gas-solid flows (Agrawal et al. 2001; Zhang and VanderHeyden 2001; Benyahia and Sundaresan 2012; Wang, van der Hoef, and Kuipers 2009). The results obtained in this work may therefore form a rigorous foundation in on-going efforts to devise a meso-scale formulation of the traditional two-fluid model for multiphase flows in nuclear reactors, as the two-fluid model used here will be subsequently extended to also allow descriptions of other forms of momentum exchange, as well as heat and mass transfer.

## 2 Modelling

This work investigates the occurrence of instabilities, fluctuations and deviations from the uniform state in two-fluid model simulations of bubbly flow in nuclear reactors. Consequently, the two-fluid model employed must be made simple enough to enable the direct observation of the effects of each feature added to the model. It is therefore assumed that the flow of interest can be represented by spherical, rigid bubbles occupying a certain volume fraction in a continuous liquid. The momentum exchange between the two phases is assumed to be dominated by the drag force. The flow is isothermal and there is no mass transfer between the phases. There is no coalescence or breakup of bubbles. These assumptions are not adequate for a real nuclear reactor, but are deemed necessary to make the formulation of a two-fluid model relevant for the steam-water flow that is stripped of all complexities except for those inherent in the two-fluid model.

### 2.1 Simplistic two-fluid model

The two-fluid model formulated for a continuous liquid phase and a mono-sized dispersed bubbly phase, under the set of restrictive assumptions noted previously, is presented in the following. The continuity equation for the dispersed phase becomes:

$$\frac{\partial}{\partial t}(\alpha_b \rho_b) + \nabla \cdot (\alpha_b \rho_b \mathbf{u}_b) = 0 \quad (1)$$

The continuous phase volume fraction field is obtained from the condition that the sum of the volume fractions everywhere must be equal to unity:

$$\alpha_l = 1 - \alpha_b \quad (2)$$

In this notation,  $l$  represents the liquid phase and  $b$  the bubbly phase.

The momentum balance equations become:

$$\begin{aligned} \frac{\partial}{\partial t}(\alpha_k \rho_k \mathbf{u}_k) + \nabla \cdot (\alpha_k \rho_k \mathbf{u}_k \mathbf{u}_k) = \\ - \alpha_k \nabla p + \nabla \cdot [\alpha_k \mu_k (\nabla \mathbf{u}_k + \nabla \mathbf{u}_k^T)] + \alpha_k \rho_k \mathbf{g} + K(\mathbf{u}_q - \mathbf{u}_k) \end{aligned} \quad (3)$$

Here,  $k$  is either  $l$  or  $b$ , and  $q$  is the other phase (not currently represented by  $k$ ). The pressure  $p$  is shared by both phases, and the viscosity used is the sum of the turbulent viscosity (when a turbulence model is used) and the molecular viscosity of the continuous phase.

Two things should be emphasized in relation to (eq. (3)): Firstly, there is no discrete phase pressure and thus no repulsive term that prevents overpacking of the discrete entities. This choice is in line with the current ambition to perform a transparent investigation with as few complications as possible. At relatively low discrete phase volume fractions, such as in this work, overpacking is generally not an issue and this approach can be justified *a posteriori* by investigations of the obtained volume fraction fields. At higher volume fractions, a dispersed phase pressure would be needed to prevent overpacking, but such models are known to be sensitive to the choices of model parameters (Benyahia and Sundaresan 2012). In this work, the isotropic contribution to the discrete phase stresses thus come from the shared pressure field. Secondly, the deviatoric contribution to the discrete phase stresses is here obtained from the product of the continuous phase viscosity with the discrete phase velocity gradients. In similar previous studies, other approaches have been suggested for the treatment of these terms, such as neglecting them altogether (Benyahia and Sundaresan 2012) or to assume them to be identical to the continuous phase deviatoric stresses (Zhang and VanderHeyden 2001). As the effect of the deviatoric contribution to the discrete phase stresses is related to the growth rate of disturbances in the flow field (the discrete phase viscosity determines the length scale of the dominant instability), but not to the existence of disturbances as such (Anderson, Sundaresan, and Jackson 1995), the treatment of these terms is deemed acceptable for the purpose of the current work. Furthermore, it has been shown for gas-solid flows that the meso-scale stresses dominate over the molecular stresses (Agrawal et al. 2001).

The momentum exchange coefficient  $K$  is obtained using the following relation:

$$K = \alpha_l \alpha_b \frac{18 \mu_l}{d_b^2} \frac{C_D Re_p}{24} \quad (4)$$

Here,  $C_D$  is evaluated using the Morsi and Alexander drag law (Morsi and Alexander 1972), which is applicable for spherical particles over a wide range of particle Reynolds numbers (from the Stokes flow regime up to above 10,000).

The two-fluid model used here is somewhat simpler than a conventional two-fluid model for nuclear applications. We therefore stress that there has been no loss of generality from the simplifications involved with respect to the occurrence of instabilities. The origin of the mathematical instabilities lies in the one-pressure formulation and the energy equations, if included, do not affect the hyperbolicity of the equation set (Ransom and Hicks 1984, 1988). Furthermore, the model is comprehensive enough to capture at least some of the possible physical instabilities known to exist in gas-liquid flows (Agrawal et al. 2001; Sankaranarayanan and Sundaresan 2002). In addition, the main emphasis in the current work is not on wall-bounded systems, where velocity gradients develop and may affect the stability of a uniformly bubbling suspension via the bubble lift forces (Sankaranarayanan and Sundaresan 2002; Lucas, Prasser, and Manera 2005; Lucas et al. 2006). Instead, we aim to study whether loss of stability is possible in a periodic domain.

The two-fluid model derived in this way and applied to a periodic domain is well-posed locally in time if the initial data are smooth, but is known to possess a medium to high wavenumber instability. In practice, a smooth solution to such a two-fluid model is therefore exponentially unstable. Linearization around an assumed smooth solution, followed by a freezing of coefficients, yields that the exponential growth rate of the instabilities is (to the first order) (Gudmundsson 2005):

$$r = \frac{(|\mathbf{u}_r|)^2 \alpha_b \alpha_l (\rho_l \mu_b^2 \alpha_l + \rho_b \mu_l^2 \alpha_b)}{(\mu_l \alpha_b + \mu_b \alpha_l)^3} \quad (5)$$

Although it is still unclear how well this result transfers to the original non-linear problem (Gudmundsson 2005; Keyfitz 2001), it is deemed valuable in the assessment of the growth rate observed in our simulations. It has also been shown that, even though there is an exponential instability, the solutions can still be bounded if they become highly oscillatory or if they form shock-like structures (Gudmundsson 2005; Keyfitz 2001; Kreiss and Yström 2002). In practice, the numerical results obtained will be influenced also by the algorithms and discretization schemes used in the solution procedure (Coquel et al. 1997). It is the purpose of this work to investigate the sensitivity of the obtained solutions to both the physics

(as specified by the model itself and its initial and boundary conditions) and to the numerics.

## 2.2 Standard k- $\epsilon$ model

The bubbly flow inside a nuclear reactor is highly turbulent, which typically motivates the inclusion of a turbulence model in the computational framework (Bestion 2012). On the other hand, if large-scale vortical structures are expected to be a more significant source of velocity fluctuations than bubble- or shear-induced turbulence (as should indeed be expected for a fully periodic bubbly flow at low to moderate bubble loading), a turbulence model is generally not needed (Ojima et al. 2014). In this work, we compare simulations using a turbulence model (Section 3.5) with simulations where no turbulence model is employed (Sections 3.1–3.4). In line with the aim to keep the current investigation transparent and simplistic, the standard k- $\epsilon$  model (Launder and Spalding 1972) is chosen. Note that the model is applied to the mixture and not for the individual phases. However, it should be stressed here that due to the large density difference between the phases, the modeled turbulence will be most significantly affected by the liquid phase. Even though turbulence modeling in dense, dispersed two-phase flows is a highly complex issue, this approach is deemed appropriate given the aim and scope of the current work.

## 2.3 Computational cases

The pressure-velocity coupling is handled via the phase-coupled SIMPLE algorithm (Vasquez and Ivanov 2000). In the discretization of the convective terms of all balance equations, the third-order accurate, bounded QUICK scheme is used with structured hexahedral meshes. The diffusion terms are discretized using a second-order accurate central-differencing scheme.

The geometry chosen is a fully periodic 2D system similar to that used by Benyahia and Sundaresan (2012). The domain height is four times its width (0.1 × 0.4 m), and gravity acts downwards in the vertical direction. The weight of the carrier and the dispersed phase in the domain is balanced exactly by a prescribed pressure drop in the vertical direction. This system is not typical of the flow channels in nuclear cores, but designed to represent a situation where the flow is unbounded. The material properties are assumed constant. The spatial discretization, unless otherwise noted, is the finest one used by Benyahia and Sundaresan (2012), i.e. 64 by 256.

The trends reported in this work were confirmed to be mesh-independent with this resolution.

The choice to use a 2D domain is based on the need to reduce the computational cost when performing many simulations over long times. It should therefore be stressed that physical meso-scale structures, as defined here, arise as the continuous phase will bypass clusters of the dispersed phase more easily than it will flow through a completely homogenized system. This effect therefore appears in both two- and three-dimensional numerical simulations, albeit with quantitative differences (Agrawal et al. 2001). Furthermore, instabilities of unphysical origin are known to manifest already in 1D simulations (Pokharna, Mori, and Ransom 1997).

Since the purpose of the current work is to investigate the possible onset of instabilities inherent in the two-fluid formulation applied to typical nuclear reactor cases, we are not primarily interested in whether such instabilities eventually develop at all, but rather whether they develop over length and time scales relevant to the thermo-hydraulics of nuclear reactors. Taking 4 m as a representative height of a nuclear reactor (Demazière 2013) and 2 m/s as a representative liquid velocity (Anglart et al. 1997; Ustinenco et al. 2008), we obtain a macroscopic time scale of 2 sec. Hence, in theory it should be sufficient to study the system for a period of 2 sec. On the other hand, all simulations are started at time zero with a uniform volume fraction field, and it is likely that the time needed for instabilities to manifest will be the longest for such initial conditions. In order not to neglect instabilities that would develop within a short time frame only if the initial conditions are favorable, we therefore choose to study a longer time period of 20 sec. This time period corresponds to more than 100 passages through the periodic computational domain in the streamwise direction for the mean flow.

As a means to quantify the magnitude of the meso-scale structures, we define a global, time-resolved uniformity index  $\Phi(t)$  such that:

$$\Phi(t) = \frac{\alpha_{q,\max}(t) - \alpha_{q,\min}(t)}{\alpha_{q,\text{avg}}} \quad (6)$$

where  $\alpha_{q,\max}$  is the current maximum volume fraction of phase  $q$  in the solution domain at time  $t$ ,  $\alpha_{q,\min}$  is the corresponding current minimum volume fraction and  $\alpha_{q,\text{avg}}$  is the domain-average value of the volume fraction. With this definition, a value of  $\Phi$  equal to zero corresponds to a uniform volume fraction field and complete phase separation would correspond to  $\Phi = 1/\alpha_{q,\text{avg}}$ . The occurrences of non-zero values of  $\Phi$  with time are indicative of the appearance of a non-uniform volume fraction field.

## 3 Results and discussion

### 3.1 Case #1 – Gas-solids case

We first investigate the same gas-solids flow as the one that was simulated by Benyahia and Sundaresan (2012). The case specification is presented in Table 1. The simulation is started with a uniform volume fraction field of the dispersed phase and advanced in time with a time step of  $10^{-4}$  s. The simulation results for the volume fraction field are shown in Figure 1. Using the results from Benyahia and Sundaresan (2012) as a reference solution, it is verified that our numerical setup can reproduce the qualitative behavior of this gas-solids flow, including the appearance of meso-scale structures and fluctuations in the slip velocity, and we use this setup as a basis for going towards bubbly flow relevant to nuclear reactors.

**Table 1:** Gas-solids case.

System size (width $\times$ height), m	0.1 $\times$ 0.4
Fluid density, $\rho_f$ kg m $^{-3}$	1.3
Fluid viscosity, $\mu_f$ Pa s	$1.8 \times 10^{-5}$
Particle density, $\rho_p$ kg m $^{-3}$	1,500
Particle diameter, $d_p$ $\mu\text{m}$	75
Average particle volume fraction	0.05

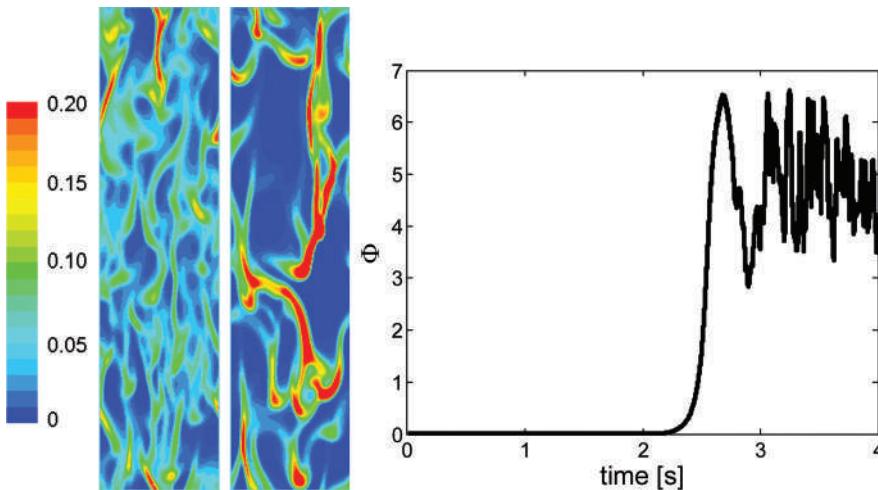
### 3.2 Case #2–gas-liquid case 1

As a first test for a gas-liquid system, we adjust Case #1 so that the terminal velocity of a bubble is similar to that of the solid particles. Under the simplified assumption of Stokes flow around the dispersed phase, the new bubble size can be found from:

$$d_b = d_p \sqrt{\frac{\mu_l}{\mu_g} \frac{|\rho_s - \rho_g|}{|\rho_b - \rho_l|}} \quad (7)$$

We then rerun Case #1 with the material properties of both phases representative of a gas-liquid flow (cf. Table 2). The properties are chosen to reflect relevant orders of magnitude, and could represent either an air-water system or a steam-water system (with the steam being either saturated or superheated). All other parameters are kept the same.

The resulting bubble diameter for this case is 0.68 mm, which is a relevant bubble size for the onset of bubble creation at sub-cooled boiling conditions (Anglart et al.

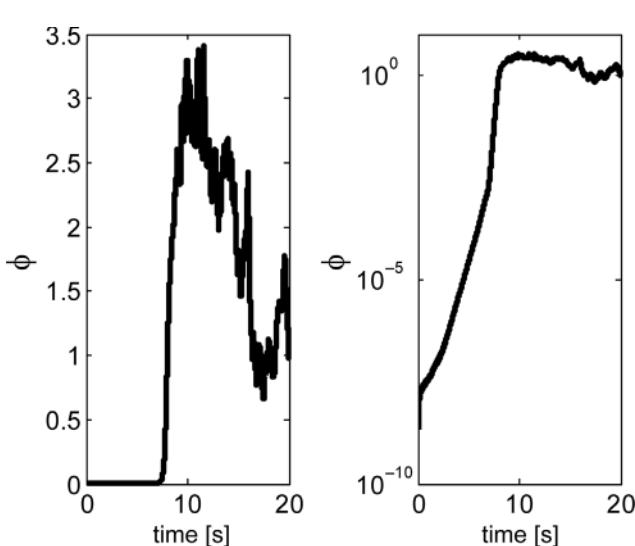


**Figure 1:** Snapshots of discrete phase volume fraction at  $t = 2.5$  s (left) and  $3.5$  s (middle). Blue and red indicate dilute and dense (volume fraction of 0.2 and higher) flow regions. To the right: Time-resolved uniformity index for Case #1.

**Table 2:** Gas-liquid case.

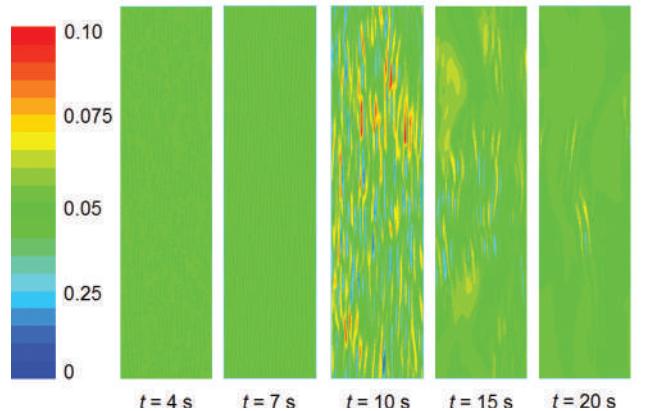
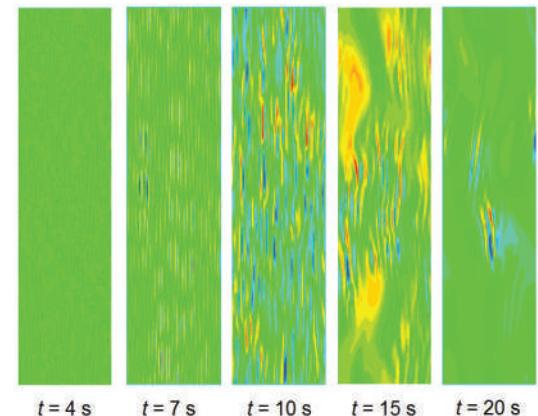
Fluid density, $\rho_f$ kg m $^{-3}$	1,000
Fluid viscosity, $\mu_f$ Pa s	$1 \times 10^{-3}$
Particle density, $\rho_p$ kg m $^{-3}$	1

1997). The behavior of the time-resolved uniformity index is shown in Figure 2. There are clear non-zero values after a few seconds of real time. A visual inspection of the volume fraction field (Figure 3) reveals meso-scale structures in the form of very thin vertically stretched zones of bubble-rich and bubble-lean areas.



**Figure 2:** Time-resolved uniformity index for Case #2. To the left: linear scale, to the right: semi-log scale.

Soon after the emergence of the “striped” structures in Figure 3, the time-resolved uniformity index starts to



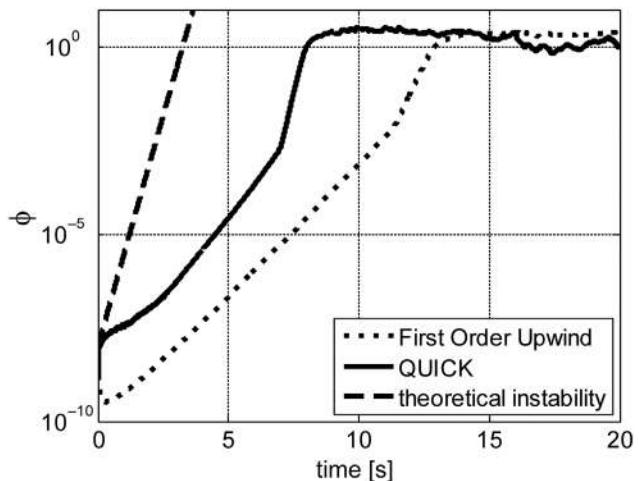
**Figure 3:** Snapshot of the discrete phase volume fractions in Case #2. Top: scaled by the maximum and minimum values observed in each snapshot. Bottom: scaled by the maximum and minimum values observed throughout the simulation (colour legend to the left). Streaks of high (red) and low (blue) volume fraction emerge slowly as a “striped” pattern and can be clearly discerned after 10 sec.

decrease slowly and the meso-scale structures become less sharp and develop a smoother, somewhat undulating character. These phenomena are attributed to the

exponential growth of small initial disturbances (in the uncertainty pertaining to the numerical precision). The solution eventually goes through a maximally unstable state, where gradients become so large that they will start to become smoothed out again by diffusive mechanisms. These diffusive mechanisms originate both from the physical model and from the numerical procedure. In the present model, it is primarily the viscous stresses that tend to stabilize disturbances (Arai 1980), but also the momentum exchange between the phases could play a similar role (Stewart 1979). There are several steps in the numerical solution procedure that influence stability, most notably the discretization of the convective terms in (eqs (1) and (3)) (Toumi 1996 and Prosperetti 2003). After the peak in the instability, the solution remains non-homogeneous throughout the rest of the simulation.

These observations are consistent with the previous literature investigations of the stability of the two-fluid model: the current formulation is unstable, but its solutions are still bounded. It is also interesting to note that the (material and case-specific) properties of this bubbly flow are very similar to one of the examples investigated by Sankaranarayanan and Sundaresan (2002). For this system, they found that the bubbles remain nearly spherical and the bubbly flow loses stability due to a growth of vertically traveling wavefronts having no horizontal structure. This mode is associated with the inertia of relative motion between the two phases and is thus the same as the dominant instability mode in typical gas-solid flows. The lift force, which is not taken into account in the present work, plays an important role in the loss of stability through a different mode and is thus not required for these instabilities to manifest (Sankaranarayanan and Sundaresan 2002). On the contrary, the addition of a lift force or a discrete-phase pressure to the model would act so as to stabilize the system under these conditions (Sankaranarayanan and Sundaresan 2002; Gudmundsson 2005; Coquel et al. 1997).

The effect of the numerical procedure to obtain the solution on the behavior and the stability of the solution itself is illustrated in Figure 4, where the results are compared for two simulations that differ only in the choice of either a lower-order scheme (First Order Upwind) or a higher-order scheme (QUICK) in the discretization of the convective terms in all balance equations. The First Order Upwind scheme is known to be robust but to give rise to false (numerical) diffusion, which should generally be expected to increase the stability and delay the transition to the unstable state on the same mesh. These are also the inferences that indeed can be drawn from Figure 4. Additionally, it is observed that neither scheme produces a growth rate as large as the theoretical one predicted by



**Figure 4:** Time-resolved uniformity index for Case #2: the effect of using a lower-order (First Order Upwind) or a higher-order (QUICK) discretization scheme for the convective terms in (eqs (1) and (3)). The theoretical growth rate for the instabilities obtained from (eq. (5)) is also plotted for comparison.

(eq. (5)). It is also concluded that the effect of the choice of the numerical solution procedure on the behavior of the system is significant, and it should thus be emphasized that the exact time-history obtained (as shown in Figure 3) is unlikely to reflect a true physical behavior of the system, but should be interpreted rather as a display of the unstable character of a bubbly two-phase flow and the complex interplay between the design of the mathematical model and the choice of numerical algorithms dedicated to solving it.

Ransom and Hicks (1984, 1988) use the term “unphysical instabilities” to differentiate unbounded instabilities from bounded instabilities (which they refer to as “physical instabilities”), and they argue that mathematical models of physical instabilities (e.g. interface instabilities in two-phase flows) will start to exhibit unphysical behavior when the physics included in the model is insufficient (Ransom and Hicks 1984, 1988). They propose that instabilities will result from the inadequate pressure equilibrium assumption inherent in the single-pressure two-fluid model, but that these instabilities may remain bounded because of the combined effects of, for example, viscous stresses and numerical diffusion. These speculations agree with our observations. As the simulations are advanced in time, the minuscule numerical round-off errors in the predicted pressure, velocity and volume fraction fields tend to grow with time throughout the domain.

Given the fact that the current formulation of the two-fluid model for a bubbly gas-liquid flow exhibits these characteristics, we will now investigate the sensitivity of this behavior to changes in the case specification

(e.g. changes to the bubble loading and the domain size) and to the inclusion of a turbulence model.

### 3.3 Case #3 – Gas-liquid case 2

It is known that a uniformly bubbling suspension loses stability at bubble loadings of a few percent or more, and that the suggested critical bubble loadings span a wide range (cf. Sankaranarayanan and Sundaresan (2002), Joshi et al. (2001), Mudde, Harteveld and van den Akker (2009)). However, the simplistic formulation of the two-fluid model that is under investigation here is known to be unstable for all non-zero bubble loadings (Gudmundsson 2005). Numerical simulations are performed to investigate the behavior of the volume fraction fields predicted by this two-fluid model for three different average bubble loadings: 1, 5 and 10%. The results are shown in Figure 5. It is clear that the growth rate of the initial instabilities is a function of the bubble loading, and that the growth rate increases with increasing loading. The non-uniformities in the case with 1% bubble loading are still extremely small after 20 sec, but a growth is still clearly there. The behavior at 5 and 10% bubble loading are qualitatively similar, with the difference that stability is lost earlier with the higher bubble loading.

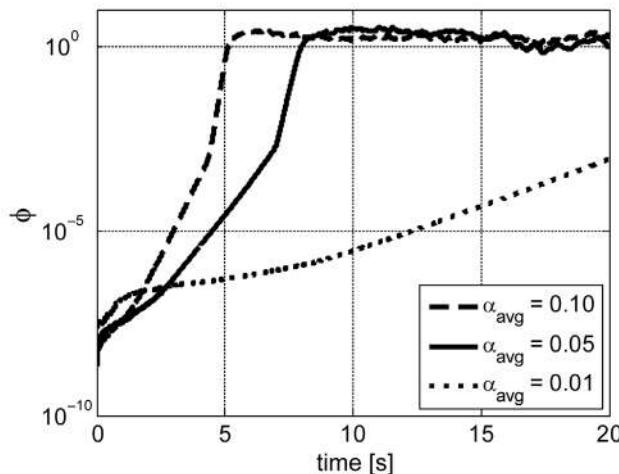


Figure 5: Time-resolved uniformity index for Case #3.

### 3.4 Case #4 – Nuclear reactor case 1

The bubbly flow in a nuclear reactor occurs in between the fuel rods and is therefore geometrically restricted in the horizontal direction. Representative values of the distance between the centers of the fuel rods are in the range 1–3 cm, with the distance between the outer radii of two consecutive fuel pins being approximately 3–8 mm

(Anglart et al. 1997; Ustinenco et al. 2008). In other words, the characteristic length of the bounding geometry in a nuclear reactor case is approximately an order of magnitude smaller than in the previous computational cases. At the same time, the variation in the typical bubble size can be very significant in the vertical direction. The initial creation of bubbles at sub-cooled boiling conditions, however, results in small, spherical bubbles of 0.15–1.5 mm in diameter (Anglart et al. 1997), but bubble diameters up to 5 mm are relevant (Ustinenco et al. 2008). The resulting ratio between the discrete phase diameter and the length scale characterizing the bounding geometry is thus of the order of 0.015–0.625 in a nuclear reactor, whereas it was 0.0068 in the previous computational case. In other words, for the investigations to be fully relevant to nuclear reactor applications, it is necessary to also study higher values of  $d_p/L$ . Here, we choose to scale the computational domain so that it becomes 1 cm wide (maintaining the aspect ratio) and to maintain a bubble diameter of 0.68 mm, yielding a value of  $d_p/L$  equal to 0.068. This value is in the correct range and represents a significant increase from the previous cases. Finally, we also change the boundary conditions on the vertical sides from periodic to free-slip walls. The presence of a wall introduces the sought-after geometrical limitation on the meso-scale structures, and the free-slip boundary condition allows us to probe the influence from this restriction without imposing sharp gradients in the velocity field that are likely to have an additional influence of their own on the instabilities.

The time-resolved non-uniformity index obtained with a 1 cm wide, horizontally bounded geometry with free-slip walls is displayed in Figure 6. The instabilities are there but the non-uniformity index remains lower

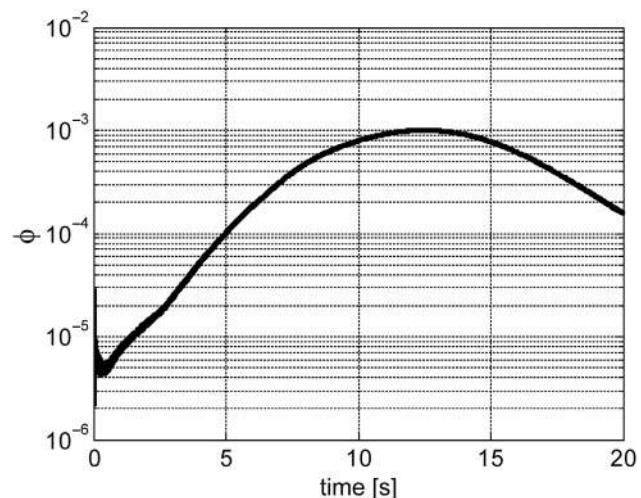


Figure 6: Time-resolved uniformity indices for Case #4: a small (1 cm) domain with free-slip walls.

than  $10^{-3}$  for the first 20 sec for this case. The effects of imposing a no-slip boundary condition are shown in Figure 7 for the original 10 cm wide domain. The large gradients in the velocity field that result from the no-slip boundary condition trigger the instabilities much faster than for the fully periodic case. Here, meso-scale structures appear already within the first 3 sec. The trend with respect to the domain size is as expected: the instabilities are less pronounced in a smaller (1 cm) domain, but their magnitude are still larger when employing no-slip boundary conditions than with free slip.

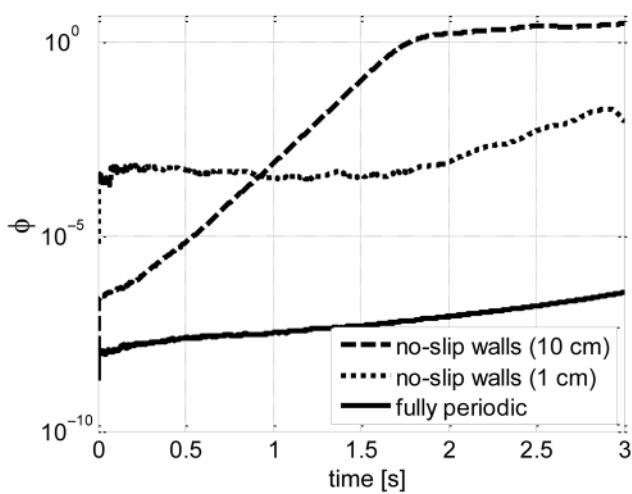


Figure 7: Time-resolved uniformity indices for Case #4: the effect of no-slip walls.

### 3.5 Case #5 – Nuclear reactor case 2

As described previously, the two-fluid model is typically used together with a Reynolds-Averaged Navier-Stokes (RANS)-based turbulence model in the simulations of nuclear reactors. The most important effect of adding a RANS model to these computations is the introduced change in the effective viscosity. Consequently, we revisit the previous nuclear reactor cases with the addition of the Standard  $k-\epsilon$  turbulence model.

The Standard  $k-\epsilon$  model predicts energy-containing turbulent eddies of approximately 1.8 cm in the bounded 10 cm domain. The ratio of the turbulent to the dynamic viscosity of the liquid becomes of the order of several hundreds, implying that the effective viscosity is significantly increased, as expected. In comparison, large eddy simulations of the turbulent liquid (single-phase) flow in a three-dimensional mesh of identical resolution with the dynamic Smagorinsky subgrid model yields turbulent viscosities of the same order of magnitude as the molecular

viscosity or lower. These large eddy simulations hence suggest that the mesh resolution is almost sufficient for a direct numerical simulation of the turbulence.

The effect on the meso-scale structures from the addition of the Standard  $k-\epsilon$  model is to effectively dampen out all fluctuations, and the non-uniformity indices (not shown) remain very small throughout the simulations. This observation is not so strange after all, since it has been known for decades that the addition of an “artificial viscosity” in the numerical algorithm employed to solve the two-fluid model helps dampen out high-frequency oscillations and therefore assists in achieving numerical stability (Ransom and Hicks 1984; Ishii and Mishima 1984).

It should be stressed here that the physical relevance of turbulence as an inhibitor of the emergence of meso-scale structures is very dubious. The two-fluid model employed here is derived by averaging over the spatial and temporal resolution employed in the computational setup. The meso-scale structures then appear when employing the two-fluid model with adequate resolution; that is, when using a resolution that is significantly finer than the meso-scales. Such a numerical simulation presupposes that the velocities used in (eqs (1) and (3)) are obtained with the same resolution as the volume fraction fields. A RANS-based turbulence model, on the other hand, is derived by averaging out all of the turbulent velocity fluctuations. Combining the two-fluid model with the Standard  $k-\epsilon$  model therefore results in an attempt to solve for fluctuations in the volume fraction fields without accounting for the fluctuating components of the phase velocities. Consequently, RANS-based turbulence modeling can only be considered compatible with the two-fluid model if the flow is steady or quasi-steady (i.e. there is a clear separation of scales between the mean flow variations and the turbulence and two-phase intermittency) (Bestion 2012).

It is interesting to relate the current results to those obtained when applying some variant of a two-fluid model together with a RANS-based turbulence model to a bubble plume in a flat geometry (Sokolichin and Eigenberger 1999; Pfleger et al. 1999; Mudde and Simonin 1999). If the depth of such a domain is neglected and the geometry described as two-dimensional, the turbulent viscosity becomes much too high and the unsteady motion of the plume is suppressed entirely. In three-dimensional simulations of the thin geometry, the effective viscosity however decreases and a dynamic behavior can be recovered. Of specific interest here is the fact that Sokolichin and Eigenberger (1999), who did not use a full two-fluid model, obtained good agreement

with the experimental data of Becker, Sokolichin, and Eigenberger (1994), whereas Mudde and Simonin (1999), who used a single-pressure two-fluid model, obtained close agreement with the same experimental data only after incorporating the effect of virtual mass into the interfacial momentum transfer term. These results are very interesting in the light of the fact that the virtual mass terms are generally small but affect the hyperbolicity, and hence the stability, of the two-fluid model employed (Lahey et al. 1980).

In summary, employing the Standard  $k-\epsilon$  model to take the effects of turbulence into account produced high effective viscosities that damped out the occurrence of meso-scale structures. This numerical experiment highlights the difficulties in describing turbulent two-phase flow using a combination of averaged mathematical models (i.e. a two-fluid model and a RANS model) that are averaged on different length scales. In a mesh refinement study of such a model, small-scale fluctuations in the volume fraction fields that are permitted by the model as such are prevented to appear, as the corresponding velocity field cannot be retrieved. In effect, the correlated fluctuations between the dispersed phase volume fraction field and the continuous phase stress gradient are neglected in the momentum exchange (Igci et al. 2008). The details of the interaction between turbulence and the dispersed phase meso-scale structures must therefore be studied using more comprehensive mathematical frameworks. If meso-scale structures are deemed important for the overall behavior of bubbly flows in nuclear reactors, more work will be needed in the derivation of models for the unresolved fluctuations and their cross-correlations. It is to be expected that more sophisticated turbulence models will be necessary for such simulations (e.g. LES or possibly URANS). It is therefore interesting to note that such turbulence models do not in general work well with dissipative discretization schemes, as these tend to dampen out the resolved turbulent fluctuations and therefore limit the accuracy of the underlying subgrid-scale model. Such issues, which are manageable in single-phase flows, will continue to pose challenges to simulations of gas-liquid flows with a two-fluid model until the complex boundaries and interplay between physical and unphysical instabilities are fully understood.

## 4 Conclusions

In the present work, we formulate a simplistic two-fluid model for the bubbly steam-water flow existing between

the fuel pins in nuclear assemblies. The appearance of non-uniform volume fraction fields is investigated and shown to be a function of the bubble loading as well as the domain size. The combination of the two-fluid model with a RANS-based turbulence model is shown to dampen the instabilities and to prevent the non-uniform fields from emerging.

In conclusion, the findings in this work highlight the importance of a consistent filtering approach in the treatment of the volume fraction field and the turbulence in simulations of two-phase flows in nuclear reactors.

**Funding:** This research was conducted with funding from the Swedish Research Council (Vetenskapsrådet) as a part of the Development of Revolutionary and Accurate Methods for Safety Analyses of Future and Existing Reactors (DREAM4SAFER) framework grant (contract number C0467701). The computations were partly performed on resources at Chalmers Centre for Computational Science and Engineering (C3SE) provided by the Swedish National Infrastructure for Computing (SNIC).

## References

1. Agrawal, K., Loezos, P.N., Syamlal, M., Sundaresan, S., 2001. The role of meso-scale structures in rapid gas-solid flows. *Journal of Fluid Mechanics* 445, 151–185.
2. Anderson, K., Sundaresan, S., Jackson, R., 1995. Instabilities and the formation of bubbles in fluidized beds. *Journal of Fluid Mechanics* 303, 327–66.
3. Anglart, H., Nylund, O., Kurul, N., Podowski, M.Z., 1997. CFD prediction of flow and phase distribution in fuel assemblies with spacers. *Nuclear Engineering and Design* 177, 215–228.
4. Arai, M., 1980. Characteristics and stability analyses for two-phase flow equation systems with viscous terms. *Nuclear Science and Engineering* 74, 77–83.
5. Becker, S., Sokolichin, A., Eigenberger, G., 1994. Gas-liquid flow in bubble columns and loop reactors: PART II. Comparison of detailed experiments and flow simulations. *Chemical Engineering Science* 49, 5747–5762.
6. Benyahia, S., Sundaresan, S., 2012. Do we need sub-grid scale corrections for both continuum and discrete gas-particle flow models? *Powder Technology* 220, 2–6.
7. Bestion, D., 2012. Applicability of two-phase CFD to nuclear reactor thermohydraulics and elaboration of best practice guidelines. *Nuclear Engineering and Design* 253, 311–321.
8. Coquel, F., El Amine, K., Godlewski, E., Perthame, B., Rascle, P., 1997. A numerical method using upwind schemes for the resolution of two-phase flows. *Journal of Computational Physics* 136, 272–88.
9. Demazière, C., 2013. Multi-physics modelling of nuclear reactors: current practices in a nutshell. *International Journal of Nuclear Energy Science and Technology* 7, 288–318.

10. Dinh, T.N., Nourgaliev, R.R., Theofanous, T.G., 2003. Understanding the ill-posed two-fluid model. Proceedings of the 10<sup>th</sup> International Topical Meeting on Nuclear Reactor Thermal Hydraulics NURETH-10), Seoul, Korea.
11. Esmaeeli, A., Tryggvason, G., 1998. Direct numerical simulation of bubbly flows. Part 1. Low Reynolds number arrays. *Journal of Fluid Mechanics* 377, 313–345.
12. Gudmundsson, R.L., 2005. A numerical study of the two-fluid models for dispersed two-phase flow. PhD thesis, KTH Royal Institute of Technology, Stockholm, Sweden.
13. Igci, Y., Andrews, A.T. IV, Sundaresan, S., Pannala, S., O'Brien, T., 2008. Filtered two-fluid models for fluidized gas-particle suspensions. *AIChE Journal* 54, 1431–1448.
14. Ishii, M., Mishima, K., 1984. Two-fluid model and hydrodynamic constitutive relations. *Nuclear Engineering and Design* 82, 107–126.
15. Joshi, J.B., Deshpande, N.S., Dinkar, M., Phanikumar, D.V., 2001. Hydrodynamic stability of multiphase reactors. *Advances in Chemical Engineering* 26, 1–130.
16. Keyfitz, B.L., 2001. Mathematical properties of nonhyperbolic models for incompressible two-phase flow. *Proceedings of the 4<sup>th</sup> International Conference on Multiphase Flow*, New Orleans, USA.
17. Kreiss, H.-O., Yström, J., 2002. Parabolic problems which are ill-posed in the zero dissipation limit. *Mathematical and Computer Modelling* 35, 1271–1295.
18. Lahey, R.T. Jr. Cheng, L.Y., Drew, D.A., Flaherty, J.E., 1980. The effect of virtual mass on the numerical stability of accelerating two-phase flows. *International Journal of Multiphase Flow* 6, 281–294.
19. Launder, B.E., Spalding, D.B., 1972. *Lectures in Mathematical Models of Turbulence*, Academic Press, London.
20. Lucas, D., Krepper, E., Prasser, H.-M., Manera, A., 2006. Investigations on the stability of the flow characteristics in a bubble column. *Chemical Engineering & Technology* 29, 1066–1072.
21. Lucas, D., Prasser, H.-M., Manera, A., 2005. Influence of the lift force on the stability of a bubble column. *Chemical Engineering Science* 60, 3609–3619.
22. Monahan, S.M., Fox, R.O., 2007. Linear stability analysis of a two-fluid model for air-water bubble columns. *Chemical Engineering Science* 62, 3159–3177.
23. Morsi, S.A., Alexander, A.J., 1972. An investigation of particle trajectories in two-phase flow systems. *Journal of Fluid Mechanics* 55, 193–208.
24. Mudde, R.F., Hartevelde, W.K., van den Akker, H.E.A., 2009. Uniform flow in bubble columns. *Industrial & Engineering Chemistry Research* 48, 148–158.
25. Mudde, R. F., Simonin, O., 1999. Two- and three-dimensional simulations of a bubble plume using a two-fluid model. *Chemical Engineering Science* 54, 5061–5069.
26. Ojima, S., Hayashi, K., Hosokawa, S., Tomiyama, A., 2014. Distributions of void fraction and liquid velocity in airwater bubble column. *International Journal of Multiphase Flow* 67, 111–121.
27. Pfleger, D., Gomes, S., Gilbert, N., Wagner, H.-G., 1999. Hydrodynamic simulations of laboratory scale bubble columns fundamental studies of the Eulerian-Eulerian modelling approach. *Chemical Engineering Science* 54, 5091–5099.
28. Pokharna, H., Mori, M., Ransom, V.H., 1997. Regularization of two-phase flow models: a comparison between numerical and differential approaches. *Journal of Computational Physics* 134, 282–295.
29. Prosperetti, A., 2003. Two-fluid modelling and averaged equations. *Multiphase Science and Technology* 15, 181–192.
30. Prosperetti, A., Satrape, J.V., 1990. Stability of two-phase flow models, in: Joseph, D.D. et al. (Eds.), *Two Phase Flows and Waves* 98–117, Springer-Verlag, New York.
31. Ransom, V.H., Hicks, D.L., 1984. Hyperbolic two-pressure models for two-phase flow. *Journal of Computational Physics* 53, 124–151.
32. Ransom, V.H., Hicks, D.L., 1988. Hyperbolic two-pressure models for two-phase flow revisited. *Journal of Computational Physics* 75, 498–504.
33. Sankaranarayanan, K., Shan, X., Kevrekidis, I.G., Sundaresan, S., 2002. Analysis of drag and virtual mass forces in bubbly suspensions using an implicit formulation of the Lattice Boltzmann method. *Journal of Fluid Mechanics* 452, 61–96.
34. Sankaranarayanan, K., Sundaresan, S., 2002. Lift force in bubbly suspensions. *Chemical Engineering Science* 57, 3521–3542.
35. Sokolichin, A., Eigenberger, G., 1999. Applicability of the standard  $k-\epsilon$  turbulence model to the dynamic simulation of bubble columns: Part I. Detailed numerical simulations. *Chemical Engineering Science* 54, 2273–2284.
36. Stewart, H.B., 1979. Stability of two-phase flow calculation using two-fluid models. *Journal of Computational Physics* 33, 259–270.
37. Stewart, H.B., 1986. On the mathematics of multifield flow. *Journal of Mathematical Analysis and Applications* 114, 241–251.
38. Toumi, I., 1996. An upwind numerical method for two-fluid two-phase flow models. *Nuclear Science and Engineering* 123, 147–168.
39. Ustinenko, V., Samigulin, M., Ioilev, A., Lo, S., Tentner, A., Lychagin, A., Razin, A., Girin, V., Vanyukov, Ye., 2008. Validation of CFD-BWR, a new two-phase computational fluid dynamics model for boiling water reactor analysis. *Nuclear Engineering and Design* 238, 660–670.
40. Vasquez, S.A., Ivanov, V.A., 2000. A phase coupled method for solving multiphase problems on unstructured meshes. *Proceedings of ASME FEDSM'00*, ASME 2000 Fluids Engineering Division Summer Meeting, Boston.
41. Wang, J., van der Hoef, M.A., Kuipers, J.A.M., 2009. Why the two-fluid model fails to predict the bed expansion characteristics of Geldart A particles in gas-fluidized beds: a tentative answer. *Chemical Engineering Science* 64, 622–625.
42. Yang, N., Wu, Z., Chen, J., Wang, Y., Li, J., 2011. Multi-scale analysis of gas-liquid interaction and CFD simulation of gas-liquid flow in bubble columns. *Chemical Engineering Science* 66, 3212–3222.
43. Yström, J., 2001. On two-fluid equations for dispersed incompressible two-phase flow. *Computing and Visualization in Science* 4, 125–135.
44. Zhang, D.Z., VanderHeyden, W.B., 2001. High-resolution three-dimensional numerical simulation of a circulating fluidized bed. *Powder Technology* 116, 133–141.



## **Paper IV**

*“Coupled fine-mesh neutronics and thermal-hydraulics - modeling and implementation for PWR fuel assemblies”*

K. Jareteg, P. Vinai, S. Sasic, C. Demaziére

*Annals of Nuclear Energy 84 (2015), pp. 244-257*





## Coupled fine-mesh neutronics and thermal-hydraulics – Modeling and implementation for PWR fuel assemblies



Klas Jareteg <sup>a,\*</sup>, Paolo Vinai <sup>a</sup>, Srdjan Sasic <sup>b</sup>, Christophe Demazière <sup>a</sup>

<sup>a</sup>Division of Nuclear Engineering, Department of Applied Physics, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden

<sup>b</sup>Division of Fluid Dynamics, Department of Applied Mechanics, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden

### ARTICLE INFO

#### Article history:

Received 31 March 2014

Accepted 20 January 2015

Available online 18 March 2015

#### Keywords:

Fine-mesh solver

Neutronics

Thermal-hydraulics

Parallelization

Coupled deterministic nuclear reactor modeling

### ABSTRACT

In this paper we present a fine-mesh solver aimed at resolving in a coupled manner and at the pin cell level the neutronic and thermal-hydraulic fields. Presently, the tool considers Pressurized Water Reactor (PWR) conditions. The methods and implementation strategy are such that the coupled neutronic and thermal-hydraulic problem is formulated in a fully three-dimensional (3D) and fine mesh manner, and for steady-state situations. The solver is built on finite volume discretization schemes, matrix solvers and capabilities for parallel computing that are available in the open source C++ library foam-extend-3.0. The angular neutron flux is determined with a multigroup discrete ordinates method ( $S_N$ ), solved by a sweeping algorithm. The thermal-hydraulics is based on Computational Fluid Dynamics (CFD) models for the moderator/coolant mass, momentum, and energy equations, together with the fuel pin energy equation. The multiphysics coupling is solved by making use of an iterative algorithm, and convergence is ensured for both the separate equations and the coupled scheme. Since all the equations are implemented in the same software, all fields can be directly accessed in such a manner that external transfer and external mapping are avoided. The parallelization relies on a domain decomposition which is shared between the neutronics and the thermal-hydraulics. The latter allows to exchange the coupled data locally on each CPU, thus minimizing the data transfer. The code is tested on a quarter of a  $15 \times 15$  PWR fuel lattice. The results show that convergence is successfully reached, and correct physical behaviors of all fields can be achieved with a reasonable computational effort.

© 2015 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

### 1. Introduction

To simulate the behavior of a nuclear reactor core, multiple fields of physics need to be considered. The distribution of the neutrons determine the amount of energy released by fission in the fuel. In a Light Water Reactor (LWR), the released energy is conducted through the solid fuel pins to the conjugate liquid (or vapor) water, and it is removed by the forced water flow. The water is not only acting as coolant, but also as moderator for the neutrons. In turn, the density of the water couples to the neutron distribution. Besides, the fuel temperature, which depends on the power and the coolant conditions, gives another feedback to the macroscopic neutron cross-sections via the Doppler effect. Other phenomena such as thermal expansion of the solid fuel and other core structures, fluid–structure interaction and material properties also impact the behavior of the core.

Furthermore, the reactor core is a multiscale system, with the scales ranging from the core width and height to the atomic ones governing nuclear reactions. For the neutronic solvers, the multiscale problem has typically been solved using a multistage procedure: the macroscopic cross-sections are generated in advance with a high order lattice solver and are employed for full core calculations performed with a low order coarse mesh solver. For thermal-hydraulics an equivalent multistage scheme can be seen in the use of subchannel codes and lower dimensional system codes, although such codes are not sequentially applied.

The multiphysics and multiscale can be tackled using a wide variety of schemes and methodologies. A splitting approach is often utilized, which consists of separate methodologies and codes for each field of physics and scale. The dependencies could then be regained using an *a posteriori* coupling, applying iterative schemes (for an overview see e.g. [Ivanov and Avramova, 2007](#)). Due to the split schemes, the couplings are usually only retrieved at the coarsest level, whereas the multiphysics coupling will take place at multiple scales. Therefore, these schemes might be inconsistent. This issue has recently gained a renewed interest, where the use of

\* Corresponding author. Tel.: +46 317723077.

E-mail address: [klas.jareteg@chalmers.se](mailto:klas.jareteg@chalmers.se) (K. Jareteg).

more computational power has allowed for direct coupling at finer scales, and thus leading to more integrated approaches to the multiphysics problem (see e.g. [Gaston et al., 2009](#)). Such methodologies typically rely on tightly coupled solvers, using implicit, non-linear techniques to handle the couplings. In other works, fine-mesh multiphysics couplings have been achieved using multiple codes, with a varying level of sophistication for the couplings (see e.g. [Kochunas et al., 2012](#); [Hamilton et al., 2013](#)).

The large increase in computer resources has also allowed new applications of numerical simulations in the reactor core. Examples include integrated approaches for direct simulation of fuel material properties ([Newman et al., 2009](#)) and numerical simulations of resolved grid-to-rod fretting in a fuel assembly ([Bakosi et al., 2013](#)). High-fidelity simulations of a nuclear reactor that include all the mentioned aspects are challenging from a modeling perspective and are beyond the computational capacity available today. Instead, different approaches are still required for the different parts of the multiscale problem.

However, such insight does not imply that smaller scales (as compared to the core global scale) could be discarded. On the contrary, from a higher resolution coupled simulation, the models and correlations used for the coarse scale couplings can be evaluated and improved. In the view of the earlier statement, there are incentives to work with the coupled neutronic and thermal-hydraulic problem already on the fine-mesh level. In fact, a more detailed prediction of the interplay between the two fields can provide a better understanding of, e.g., the local temperature and power distributions in the fuel pins, as well as the conjugate heat transfer between the fuel and the coolant. Such and other fuel assembly parameters and aspects are important both from economical and resource utilization perspectives, and also from safety considerations.

The goal of this article is to describe a fine-mesh simulation framework that has been developed at Chalmers University of Technology, and that aims at reproducing the coupling between neutronics and thermal-hydraulics within PWR fuel assemblies. In particular, the solver can handle steady-state problems with respect to fine meshes, and it includes models for single-phase fluid dynamics, heat transfer in the solid and fluid regions, and neutron transport. We aim to give both specific example and also a more general overview of the key points for such a framework. The full coupled problem is solved using a single code approach, allowing fine-mesh direct multiphysics coupling.

The paper is structured as follows. The implications of the fine-mesh approach are described in Section 2, highlighting some of the key concepts and main issues to be tackled. The specific models used for the neutronics, the fluid dynamics and heat transfer are given in Sections 3 and 4, respectively. In Section 5, we describe the treatment in our methodology of the coupling and data transfer between the different fields. In Section 6, strategies for the parallelization of the coupled system are outlined and discussed. The described models and implementations are then tested on a simplified  $15 \times 15$  PWR system and reported in Section 7. Finally, a summary and a number of important conclusions are given in Section 8.

## 2. Fine-mesh considerations

As mentioned above, the target of this work is to couple the neutronic and thermal-hydraulic fields on a sub-pin level. Following the choice of resolution, limitations and approximations will be imposed for other scales. With the chosen level of detail, full core calculations are extremely computationally expensive. Instead, the fine-mesh approach is applied to the fuel assembly calculations, where the computational burden is still relatively large, but can be handled with the available computer resources.

The application of a fine-mesh approach gives a possibility of a full three-dimensional representation of the fuel assembly. This fact is beneficial for two main reasons. First, the geometrical complexity of the fuel assembly can be taken in account with a better resolution. Second, physical phenomena can be better captured, avoiding one dimensional (1D) or two dimensional (2D) approximations that may be questionable.

To preserve the geometry on the sub-pin cell scales in an efficient manner, an unstructured computational mesh is required. This allows for a correct representation of the fuel pins and other structural parts and avoids homogenization of the simulated system. The choice of resolution of the mesh influences the discretization method applied to models and equations. In this work the discretization is based on the Finite Volume Method (FVM). FVM allows unstructured meshes to be used, and ensures local conservation on each cell ([Ferziger and Peric, 2002](#)). It is also a well proven method for neutronics and fluid problems, and is a standard practice for Computational Fluid Dynamics (CFD) codes.

To handle the fine-mesh requirements, the use of unstructured meshes, and the coupling of multiple fields of physics, a versatile computational code is necessary. Such a tool could consist of multiple specialized softwares, combined using an external coupling (see e.g. [Cardoni and Rizwan-uddin, 2011](#); [Kochunas et al., 2012](#)). An advantage of this scheme is the possibility to use a set of validated tools, limiting the amount of new code to be implemented and tested. However, the external transfer approach introduces not only an expensive overhead, but it also limits the resolution of the coupling and imposes constraints on the coupling algorithms. An alternative to the split software scheme is to include all fields of physics and the coupling in the same code (see e.g. [Gaston et al., 2009](#)), which is the approach followed in this work. This allows for a coupling directly on the finest scale, avoiding the computational and methodological penalties associated with external or a posteriori coupling schemes. Further benefits will be discussed in Section 6, where the integrated approach is described, and also used to handle the parallelization in an efficient manner.

To benefit from an existing code, we use the open source C++ library foam-extend-3.0 as a base for the coupled code. foam-extend-3.0 is a fork of OpenFOAM®, earlier named OpenFOAM®-dev and OpenFOAM®-ext ([Wikki, 2014](#)). The software gives access to a high performance library with a flexible code with many different applications. In particular, full availability of the source code allows for extending the code at any level, including implemented equations, physical models as well as for the linear solvers and the discretization schemes.

As a consequence of the chosen resolution and using a 3D representation of the system, the computational requirements for this kind of effort can not be overlooked. The fine-mesh approach must be solved using high performance computations (HPC), including fast programming languages, efficient algorithms and use of parallelization to efficiently tackle the problem. Furthermore, the problem is parallelized using the Message Passing Interface (MPI) ([MPI Forum, 2009](#)) as implemented in foam-extend-3.0 (further described in Section 6).

## 3. Neutronics for the sub-pin cell calculations

For the type of system and resolution under study, the selected neutronics solver methodology needs to handle the strong material heterogeneities and the angular dependence. In the case of the sub-pin neutronic calculations, the diffusion approximation is not an accurate choice. Instead a full transport methodology is required. Here, we use the discrete ordinates method, formulated for a general unstructured mesh and using the finite volume method to discretize the equations. We solve the steady state neutronic eigenvalue problem using the power iteration method.

### 3.1. $S_N$ method

The discrete ordinates method is based on solving the neutron transport equation on a set of directions (or ordinates). The equation to be solved for each separate direction is given by (Larsen and Morel, 2010):

$$\Omega_m \cdot \nabla \Psi_{m,g} + \Sigma_{T,g} \Psi_{m,g} = S_{m,g} + \frac{1}{k} F_{m,g} \quad (1)$$

where the anisotropic scattering source term is given by:

$$S_{m,g} = \sum_{l=0}^L (2l+1) \sum_{m'=1}^M P_l(\Omega_m \cdot \Omega_{m'}) w_{m'} \sum_{g'=1}^G \Sigma_{s,l,g' \rightarrow g} \Psi_{m',g'} \quad (2)$$

which is formulated in terms of the Legendre polynomials ( $P_l$ ) and the ordinate weights ( $w_{m'}$ ). The fission source term in terms of the angular flux reads:

$$F_{m,g} = \chi_g \sum_{m'=1}^M w_{m'} \sum_{g'=1}^G v_{g'} \Sigma_{f,g'} \Psi_{m',g'} \quad (3)$$

The scalar flux can then be retrieved by performing the weighted sum over all the different directions such that:

$$\Phi_g = 4\pi \sum_m^M w_m \Psi_{m,g} \quad (4)$$

The fission source with respect to the scalar flux will read:

$$F_{m,g} = \frac{\chi_g}{4\pi} \sum_{g'=1}^G v_{g'} \Sigma_{f,g'} \Phi_{g'} \quad (5)$$

Standard notations are employed for all quantities.

We assume the most general case with a fully anisotropic scattering. Thus, in each inner iteration while solving Eq. (1) (i.e. for each energy group ( $g$ ) and each ordinate  $m$ ), the full dependence of all other groups and directions will enter through the scattering term  $S_{m,g}$ . To reduce the computational cost of evaluating the scattering source, an expansion of the angular flux on real spherical harmonics is used, such that:

$$S_{m,g} = \sum_{g'=1}^G \sum_{l=0}^L (2l+1) \sum_{r=-l}^l R_{l,r} \phi_{g,l,r} S_{s,l,g' \rightarrow g} \quad (6)$$

where  $\phi_{g,l,r}$  are the expansion coefficients and  $R_{l,r}$  are the real spherical harmonics as given in Hébert (2010).

The choice of directions ( $\Omega_m$ ) and weights ( $w_m$ ) for  $S_N$  can potentially have a large influence on the accuracy of the solution, as discussed by e.g. Abu-Shumays (2001). The directions and weights in this work are based on the level symmetric quadrature set as given in Hébert (2010).

### 3.2. Solution and parallelization of $S_N$

Due to the nature of Eq. (1), the discretized equation can be solved with a sweeping algorithm that corresponds to the solution of a lower diagonal matrix using a Gauss–Seidel method, and based on the ordinate direction  $\Omega_m$ . For an unstructured (as well as a structured) mesh such sweeping order can be found starting from an inlet boundary and iteratively transversing the cells of the mesh (Plimpton et al., 2005). Cyclic dependencies are avoided if only convex cells are used, which is the case in this work. A brief outline of the applied sweep order methodology is given in Fig. 1.

When the problem is parallelized by splitting the space in different domains (as further discussed in Section 6.3), the sweeping order also needs to be parallelized. In order to conserve the single sweep over the mesh, each separate domain must wait for neighboring upstream domains to finish their sweeps. To still utilize

the time waiting, the different domains should optimally start with different directions. Such algorithm is not implemented in the current work. Instead all domains are concurrently sweeping for the same direction. This will introduce a penalty for the parallelized version of the discrete ordinates solver. However, the used algorithm is easier both to implement and to maintain in terms of generalization and implementation.

Much effort has been spent on the development of acceleration techniques for the discrete ordinates method (see e.g. Larsen and Morel, 2010). Nevertheless, no acceleration is used in this work, and the problem is instead solved by iteratively updating the directions, group by group.

An outline of the applied solution methodology is given in Fig. 1. During the initialization, the decomposed mesh is read by each CPU, whereafter the sweeping order is determined. Finally, the cross-section sets are determined (see Section 3.3). Each neutronic iteration starts with an update of the cross-sections based on the current temperature distribution in the system. Thereafter an outer iteration is used to update the eigenvalue problem, using an inner iteration to update the angular flux. After convergence of the outer iteration, the power profile is updated. Optionally, we use a diffusion solver during the first iteration. This allows an approximate fission source and criticality value to be calculated with a smaller computational cost, which can be used to increase the acceleration and convergence of the  $S_N$  solver.

### 3.3. Cross-sections and macroscopic data

To close Eq. (1), cross-sections and macroscopic data are needed. Such data are here generated using a Monte Carlo approach. We use the software Serpent (Leppänen, 2012), to create a set of two-dimensional condensed and homogenized cross-sections. The details of this approach can be found in Jareteg et al. (2014b).

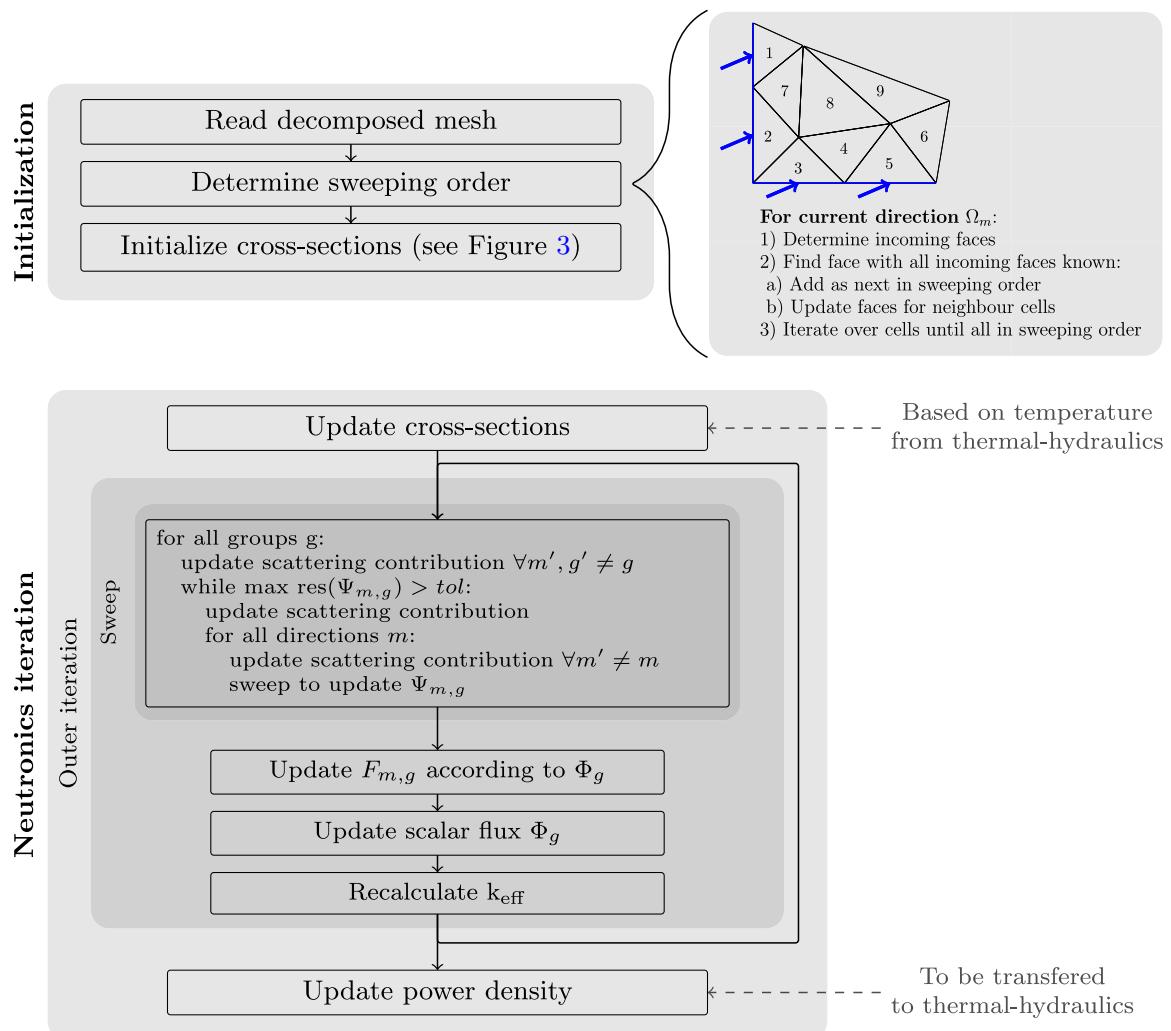
To match the resolution of the system, sub-pin cell dependent cross-sections are needed. The data are thus generated for regions split azimuthally as well as radially and tabulated according to position. Furthermore each specific radial and azimuthal region is run for a set of temperatures, as to give a temperature-dependent macroscopic data in all regions of the fuel.

The radial and azimuthal discretization used in this work can be seen for a quarter of a  $15 \times 15$  fuel pin array in Fig. 2. The regions displayed in Fig. 2 are automatically mapped to an unstructured mesh of the deterministic calculations (as exemplified in Fig. 3). In this manner the code is kept very general, and more physics and different geometrical structures can be easily added later.

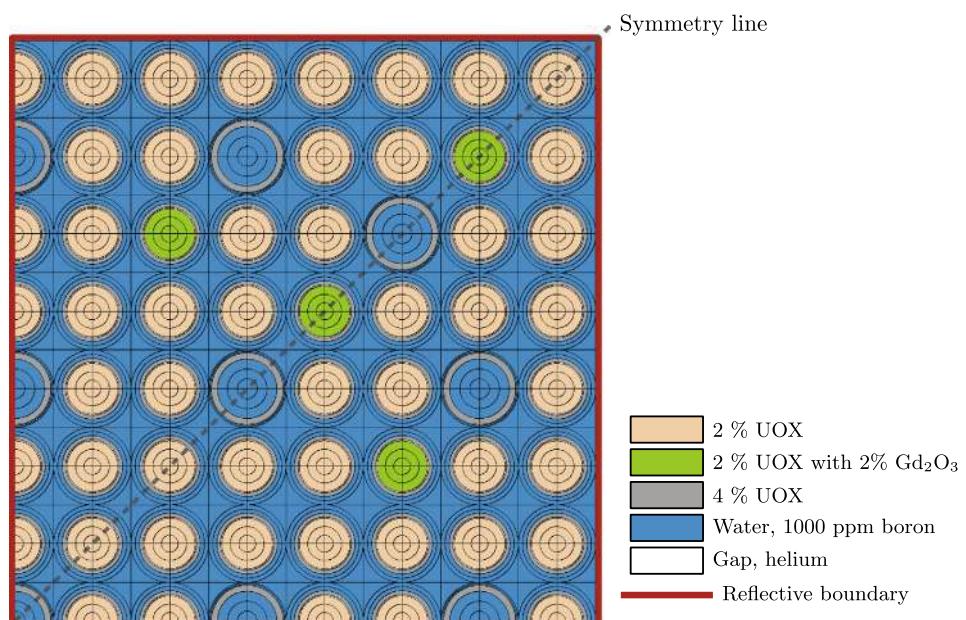
The scheme followed is given in Fig. 3. The first part, covering the generation of the cross-sections, takes place before the coupled deterministic calculations, and it provides geometrical sets and cross-section tables for each region as displayed in Fig. 2. The produced sets are then used to calculate a mapping based on the geometrical information generated by the Python script and the cell centers in the unstructured mesh. Once a mapping is determined for each cell in each CPU, the correct table can be read. Each time the cross-sections are updated (as seen in Fig. 1), the mapping information is re-used, and the cross-sections updated locally for each cell.

## 4. Thermal-hydraulic model for the fine-mesh calculations

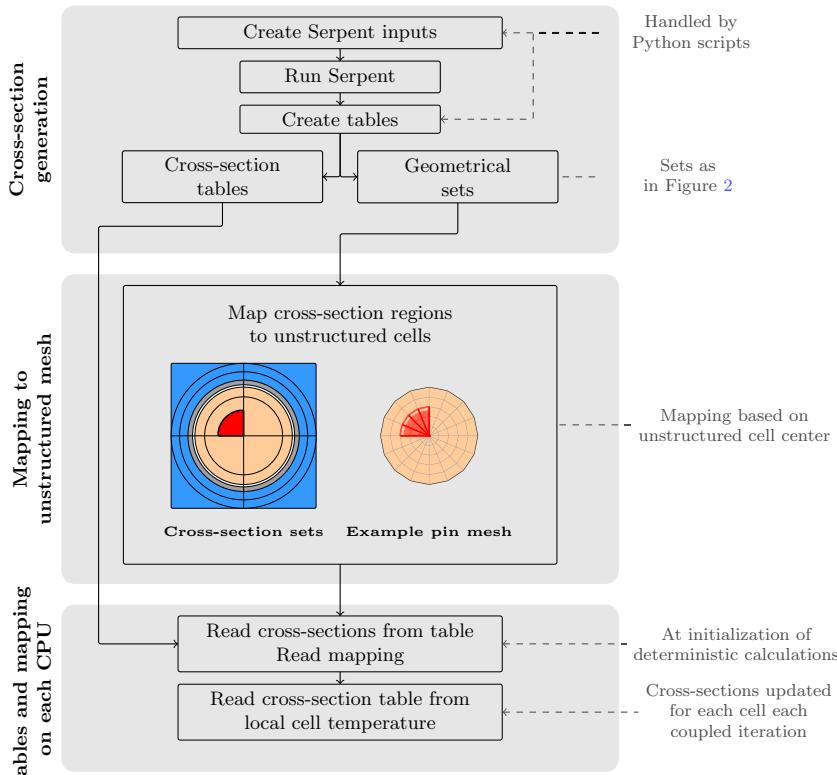
In this work the thermal-hydraulic problem is solved using mass, momentum and energy equations, all formulated from first principles. The equations and the models applied are aimed at the sub-pin cell resolution, and are consistent with the neutronic calculations. This means that the moderator gradients of velocity, density and temperature between the fuel pins and the temperature gradient within the fuel pins must all be resolved. The



**Fig. 1.** Applied algorithm for the discrete ordinates method.



**Fig. 2.** Fuel pin discretization in horizontal plane, using 4 azimuthal and 8 radial regions per pin cell, in total 1775 regions.



**Fig. 3.** Mapping scheme for the cross-sections.

approach followed here is thus more representative for CFD than for typical lower dimensional codes, which are to a larger extent based on empirical correlations.

Using a CFD methodology introduces new type of equations and solvers, both with the aim to reduce the reliance on measured or approximated quantities and to reach a higher resolution. As an example, in this work a two equation turbulence model is solved on the same mesh resolution as the momentum equation. This, in combination with the wall boundary conditions, allows for the pressure drop over the channel to be directly calculated, avoiding the use of pressure drop correlations.

The work is limited to pure single phase liquid systems, i.e. at this stage the existence of multiple phases encountered in Boiling Water Reactors (BWRs) and in the subcooled boiling region in PWRs are not taken into account. Such gas-liquid heated problem is however the focus of the next developmental stage.

Even though a single phase methodology is applied, the heat transfer from the fuel pins will still lead to a change in density in the moderator and thus also influence the momentum and mass conservation equations. This is accounted for by solving the full conjugate heat transfer problem, including the moderator, the cladding encapsulation, the gaps and the fuel pins, in one fully coupled system.

#### 4.1. Pressure–velocity solver

The momentum equation is formulated for the steady state problem and by time-averaging the Reynolds decomposed velocity. This results in a filtered momentum equation such that:

$$\nabla \cdot (\rho \mathbf{U} \otimes \mathbf{U}) = \nabla \cdot \bar{\bar{\tau}} - \nabla \cdot \overline{\rho \mathbf{u}' \otimes \mathbf{u}'} - \nabla P + \rho \mathbf{g} \quad (7)$$

with the mean velocity given by  $\mathbf{U}$ , the fluctuating contributions given by  $\mathbf{u}'$  and with the density  $\rho$ , the pressure  $P$  and the gravity

g. To close the momentum equation, the stress tensor ( $\tau_{ij}$ ) is modeled as (Panton, 2005):

$$\tau_{ij} = \mu \left( U_{ij} + U_{j,i} - \frac{2}{3} U_{k,k} \delta_{ij} \right) \quad (8)$$

where  $\mu$  is the viscosity. The velocity fluctuation term is related to the stress tensor and a kinetic viscosity according to:

$$-\overline{\rho u'_i u'_j} = \mu_t \left( U_{ij} + U_{ji} - \frac{2}{3} U_{kk} \delta_{ij} \right) - \frac{2}{3} \rho k \delta_{ij} \quad (9)$$

where  $k$  is the turbulent kinetic energy which is computed based on a two-equation turbulence model, here formulated as (see e.g. Ferziger and Peric, 2002):

$$\begin{aligned} \nabla \cdot (\rho k \mathbf{U}) &= \nabla \cdot \left( \mu + \frac{\mu_t}{\sigma_k} \nabla k \right) + \mu_t (\nabla \otimes \mathbf{U}) \\ &\quad : (\nabla \otimes \mathbf{U} + (\nabla \otimes \mathbf{U})^T) - \rho \epsilon \end{aligned} \quad (10)$$

where  $\sigma_k$  is a model constant and with the turbulent dissipation  $\epsilon$  calculated by the second equation:

$$\begin{aligned} \nabla \cdot (\rho \epsilon \mathbf{U}) &= \nabla \cdot \left( \mu + \frac{\mu_t}{\sigma_\epsilon} \nabla \epsilon \right) + C_{1\epsilon} \frac{\epsilon}{k} \mu_t (\nabla \otimes \mathbf{U}) \\ &\quad : (\nabla \otimes \mathbf{U} + (\nabla \otimes \mathbf{U})^T) - C_{2\epsilon} \rho \frac{\epsilon^2}{k} \end{aligned} \quad (11)$$

where  $\sigma_\epsilon$ ,  $C_{1\epsilon}$  and  $C_{2\epsilon}$  are model constants. After solving the turbulent kinetic energy and the turbulent dissipation, Eqs. (10) and (11), the turbulent kinetic viscosity in Eq. (9) is computed as:

$$\mu_t = \rho C_\mu \frac{k^2}{\epsilon} \quad (12)$$

Finally, the time-averaged, steady state mass equation reads:

$$\nabla \cdot (\rho \mathbf{U}) = 0 \quad (13)$$

The momentum and mass equations are used to solve for the coupled pressure and velocity distribution in the moderator. Such a formulation is typical for incompressible flow CFD calculations (see e.g. Ferziger and Peric, 2002). To resolve the steady state coupling, the SIMPLE algorithm is applied (Patankar and Spalding, 1972). This algorithm relies on first solving a momentum predictor step based on Eq. (7), resulting in an estimated velocity field. The momentum equation is then reformulated and inserted in the mass conservation equation, now solved for the pressure field, based on the predicted velocity field. Finally, the momentum equation is updated from the newly calculated pressure field.

#### 4.2. Conjugate heat transfer

Since the moderator acts also as the coolant, the thermal coupling to the fuel pins must be taken into account. The flow pattern has an impact on the coolant ability to extract the heat at the wall of the cladding and the heating of the moderator causes a change in the density leading to a change in the flow.

The regional dependence of the conjugate heat transfer can be solved either based on an iterative approach or a monolithic, implicit approach. Using the iterative approach, each region or a group of regions are computed separately. The dependence is then handled by imposing alternating Dirichlet and Neumann boundary conditions. For the system here simulated, consisting of a quarter of a  $15 \times 15$  fuel assembly, with the gap and cladding explicitly modeled, in total 184 different material regions are encountered. The high number of regions makes the system unattractive to solve in an iterative manner. If, instead, formulating and computing the full system in an implicit manner, the couplings between the regions are treated implicitly in the equation system. Such an approach, applied in this work, avoids the iterations between the regions.

We express energy conservation in terms of temperature. This allows to directly couple the equations on the faces of the cells at the region boundaries. The moderator temperature conservation equation is derived from the energy conservation equation considering steady state conditions only and applying a Reynolds decom-

position to filter the rapid fluctuations. The procedure followed is further described in (Jareteg et al., 2014a), and the final equation will read:

$$(\rho c_p(T) \mathbf{U} \cdot \nabla T) = \beta(T) \mathbf{U} \cdot \nabla P + \nabla \cdot (K_{\text{eff}}(T) \nabla T) \quad (14)$$

where  $c_p$  is the specific heat capacity,  $\beta$  is the thermal expansion coefficient and  $K_{\text{eff}}$  is the effective thermal conductivity that includes the heat transfer enhancement from turbulence.

The temperature equation for the solid regions is derived by applying Fourier's law to the heat transfer equation, such that:

$$-\nabla \cdot (K(T) \nabla T) = q''' \quad (15)$$

where  $K$  is the thermal conductivity of the material and  $q'''$  is the power density, that is only non-zero in the fuel and is given by the energy released per fission and the computed neutron flux. The heat transfer in the gap is solved using Eq. (15), neglecting the radiation heat transfer.

#### 4.3. Solver methodology

An outline of the combined pressure–velocity and conjugate heat transfer algorithm is given in Fig. 4. In the first stage the momentum predictor and the turbulence equations are solved for the moderator. This is followed by the implicit solution of the conjugate heat transfer problem between all solid material regions and the moderator. Given the new temperature field, all thermo-physical properties are updated. Finally, the effect of buoyancy is accounted for using the Boussinesq approximation (Ferziger and Peric, 2002), the pressure equation is solved and the momentum is calculated according to the new pressure. The scheme presented in Fig. 4 corresponds to one sub-iteration in the thermal-hydraulics.

In the applied methodology no explicit thermal expansion in the fuel is treated. The influence of such an approximation will be considered in later work.

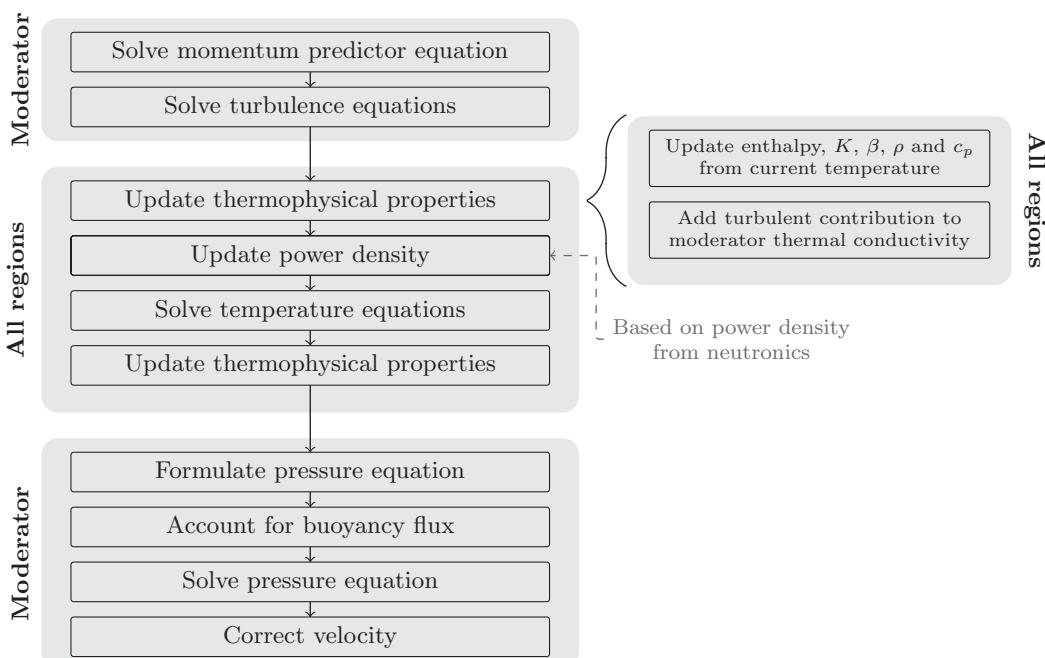


Fig. 4. Thermal-hydraulics solver methodology.

## 5. Multiphysics treatment and coupled aspects

Based on the separate models for the neutronics and thermal-hydraulics presented in the previous Sections 3 and 4, the aspects related to coupling of those phenomena will be discussed hereafter. Focus will be given on the discretization of the separate fields, on the scheme used for a consistent and conservative spatial mapping and on the iterative algorithm for resolving the multiphysics dependencies.

### 5.1. Mesh resolutions

In order to calculate accurate solutions for all separate fields of the problem, we should apply different computational meshes. Considering, for example, the different characteristic length scales present in the neutron angular flux and the near wall turbulent flow in the moderator, using the same mesh for all fields is not an efficient solution procedure. Instead of a common resolution, the mesh for each separate field should be optimized for the type of physics encountered. In the example mentioned, the near wall flow will (based on our models) require a finer mesh close to the fuel pins than for the neutron flux, so that a comparable precision can be obtained for the simulated system.

The application of different meshes also allows the computational effort to be optimized. Again, it would be a waste of resources to perform the neutronic calculations directly on the fluid flow mesh. Each part of the problem will thus be discretized with a mesh in such a way that a consistent level of detail is reached for the coupled problem. In this work the mesh for the moderator is chosen to have twice the number of cells in the azimuthal direction as compared to that for the solid heat transfer and the neutronics, as displayed in Fig. 5. In general, the validity of mesh resolution should be judged both from the grid dependence of the separate fields as well as from the grid dependence for the coupled problem.

The foam-extend-3.0 internal mesh format applied in this work primarily relies on mesh faces with two connecting cells. However, when solving the conjugate heat transfer problem, a single temperature equation is desired even in the case that two moderator faces meet a single cladding face (see Fig. 5). Since the discretization of the solid regions is more coarse than the moderator discretization, a matrix level mapping of the faces is required. This is handled with the general grid interface routines available in foam-extend-3.0 (Jasak, 2009).

Whereas the specific implementation and choice of resolution and mesh will depend on the chosen framework, the importance of such elements should be stressed. Again, a larger flexibility as well as a better consistency, are reached using a single framework for all calculations, as exemplified here by the seamless use of different mesh discretizations in the solid regions and the moderator region.

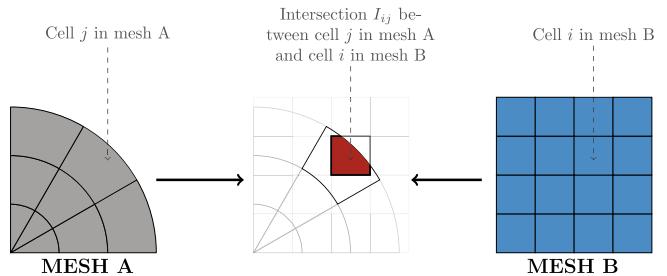


Fig. 6. Example of mapping of two overlapping meshes.

### 5.2. Consistent spatial mapping

As a consequence of using multiple meshes, a consistent and conservative mapping scheme is needed for overlapping meshes. As regards the mapping of the power density from the neutronic to the thermal-hydraulic fuel pin mesh, the correct energy released by fission is needed in the heat transfer problem given by Eq. (15). In the same manner, a correct thermophysical state must be mapped from the thermal-hydraulic to the neutronic mesh, in order to accurately update the cross-sections.

The mapping scheme here implemented is based on finding volumetric overlaps between cells in the different meshes. We apply a collocated finite volume method, where the cell values are calculated for all quantities. Since these cell values are considered constant over the cell, an exact and conservative mapping can be achieved if the volumetric overlaps between the cells are known. An example is seen in Fig. 6. Given a cell  $j$  in mesh A and a cell  $i$  in mesh B, the overlap is calculated using a polygon intersection algorithm in the horizontal plane and a direct overlap in the axial direction. The polygonal intersection algorithm relies on the Sutherland–Hodgman algorithm (Sutherland and Hodgman, 1974), as earlier implemented in foam-extend-3.0 (Page et al., 2010). The calculation of the overlap in the axial direction relies on all cells having parallel faces in the axial direction. This is assured by using only hexahedron and prism elements.

Given the calculated volumetric intersections  $I_{ij}$  for all cells  $j$  intersecting with cell  $i$ , an extensive property  $c$  can be transferred from mesh A to mesh B as follows:

$$c_i = \sum_j \frac{c_j I_{ij} V_j}{V_i} \quad (16)$$

where  $V_i$  and  $V_j$  are the cell volumes of cell  $i$  and cell  $j$ , respectively.

The implemented scheme is fully automatic and the calculations are performed at the initialization of the code, completely removing any hard coding or manual calculation of the mappings between the neutronic and the thermal-hydraulic meshes. This also means that the meshes can be changed without any further intervention in the code or in any input files. In order for the map-

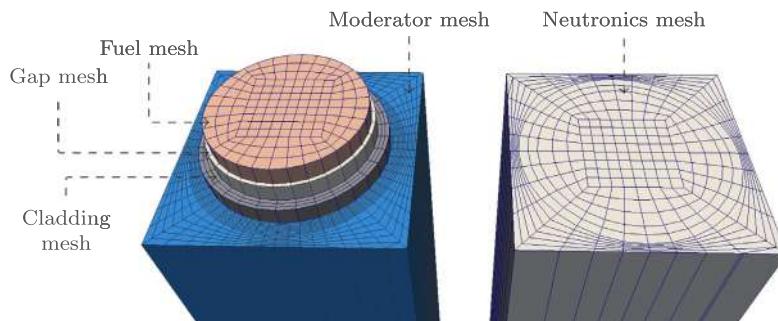


Fig. 5. Horizontal mesh discretization exploded for the thermal-hydraulics (left) and the neutronics (right).

ping to occur correctly, the material boundaries should, however, be preserved between all the meshes. If this is not the case, the transferred quantity will not be interpreted correctly in the receiving cells.

In the chosen mapping scheme based on intersection calculations and not on point interpolation, only extensive quantities should be transferred. Since temperature is not an extensive quantity, the enthalpy is used in the transfer from the thermal-hydraulic cells to the neutronic cells.

The mapping utility also opens for other fields of physics to be coupled to the problem, all potentially using separate meshes. This can further be exploited to introduce different scales of the same physics. A coarse neutron solver can be applied on its own mesh, then automatically coupled to the fine-mesh solver.

### 5.3. Coupling scheme

The multiphysics coupling can be resolved using either a fully implicit technique, an explicit approach or a combination of both. Whereas the fully implicit techniques are based on combining the fields in a non-linear problem, the fully explicit techniques will rely on iteration between the different fields. Potentially a non-linear scheme will increase the convergence rate and possibly also reach convergence for systems not possible to solve in a segregated manner (Gaston et al., 2009). However, the implicit approach will also increase the memory usage, and will require implementation of good preconditioners for each separate field of physics (see e.g. Zou et al., 2013).

We apply an explicit approach in this work, where we solve each field in a separate manner. However, the coupled dependencies are fully resolved, iterating between the different physics. In the steady state problem solved in this paper, it will be shown that reaching a global convergence is not a major problem. Further, the number of iterations between the thermal-hydraulics and the neutronics are few, and thus the incentive for the non-linear approaches are limited. In the presented case, acceleration of the neutronics and thermal-hydraulics problems separately would be of higher priority as future developments.

As we address each part separately, the independence between the modules is increased, which also simplifies the development of each separate module and gives a larger freedom to the solution procedure of each field. Nevertheless, in future development, approaching transient simulations and adding more physical phenomena, an outermost non-linear solver might be beneficial or even necessary.

The iterative procedure followed in this work can be seen in Fig. 7. After initialization of all fields the neutronics is solved first.

Given the new power profile the thermal-hydraulics is then solved, resulting in a new temperature and enthalpy field.

During the first full iteration, only the diffusion equation neutronics is solved (as described in Section 3.2). In the second iteration, the thermal-hydraulics is processed, without the temperature equation, but only pressure and velocity. Finally, in the third iteration all fields are calculated. Consequently, the fourth iteration is the first iteration with the cross-sections updated according to a new temperature profile. Such a procedure allows the problem to converge in a more stable manner, not starting the full coupled problem only with the initial guesses.

Sub-iterations are used both in the neutronics and thermal-hydraulics. In earlier work such sub-iterations were avoided (Jareteg et al., 2013). It was, however, found that an overall faster convergence was reached if all fields were allowed to converge within each outermost coupled iteration (Jareteg et al., 2013). Yet, a maximum number of sub-iterations are deployed for both neutronics and thermal-hydraulics, since full convergence is still not necessary until the full coupled problem has converged.

## 6. Efficiency and scalability; implementation and parallelization

Due to a large number of degrees of freedom in the fine-mesh calculations, a severe computational cost needs to be tackled already for a single fuel assembly. Consequently, the code implementation and the use of high performing computer languages are both of significant importance. This is true for the separate models as well as for the coupling scheme. Furthermore, we must make it possible for the full problem to be parallelized in a scalable manner.

### 6.1. Model and equation implementation

The separate parts of the multiphysics problem are all implemented using foam-extend-3.0. The library foam-extend-3.0 is based entirely on C++ programming language and contains a large set of solvers for different type of problems in fluid mechanics as well as heat transfer. For the implementation of the neutronics, the existing internal structure is used for the equation discretization, for the equation solvers, and for all general structures such as matrices and meshes.

When selecting a framework, or writing a completely new, the performance of the code is not the only point of concern. In fact, the code should be easy to extend and it should be written in such a way that it is simple to maintain. Also from this aspect foam-extend-3.0 is a seemingly viable choice. The code uses modern

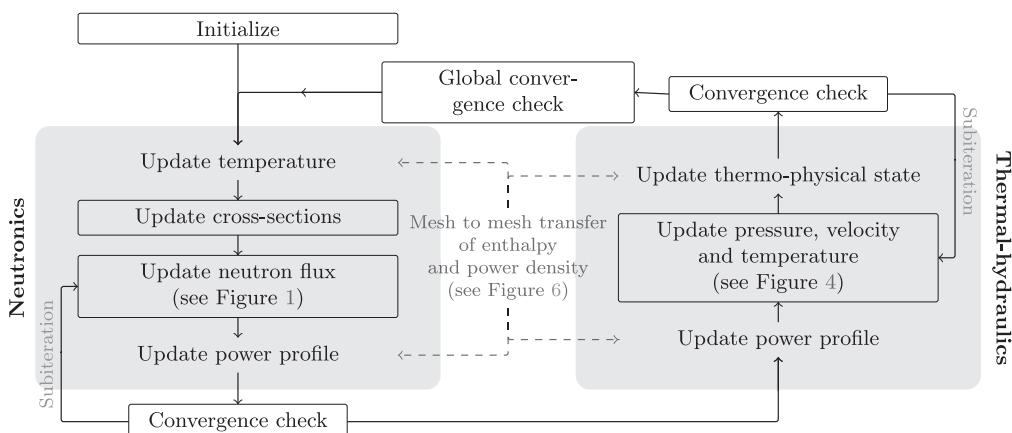


Fig. 7. Iterative scheme applied for the coupling of the thermal-hydraulics and the neutronics.

C++ functionalities such as templates and polymorphism to allow for easy and convenient extensions and modifications to the code.

The code foam-extend-3.0 contains a high level equation format, which makes possible a fast implementation of additional equations. This is used for the heat transfer equations, (14) and (15), whereas the pressure and velocity solver is based directly on the existing solvers. For the cross-section formalism, the mapping algorithm, and the sweeps for the discrete ordinates method, more extensive coding work was needed.

## 6.2. Coupling implementation

For the multiphysics couplings, different potential approaches were described in Section 5.3. The implementation of the coupling method could be based on different schemes. Irrespective of whether an implicit, non-linear coupling or an iterative segregated approach is used, two setups could be followed. Either all equations are solved in the same code entity (a single code approach) or multiple different softwares (multiple codes approach) address different subsets of the equations.

Using the multiple codes implementation, an external transfer of information is necessary to exchange the coupled states. In the present case, the power density would be passed from the neutronics to the thermal-hydraulics, while the thermophysical state would go in the reverse direction. In many cases such data transfer can severely reduce the efficiency of the code and take a major part of the program execution time (see e.g. Yan et al., 2011).

Further, the possibilities to obtain efficient parallel couplings between the neutronics and thermal-hydraulics are limited. In many approaches two different softwares are used, each performing its own parallelization. The communication is in such a case often handled by a central application, running on a single CPU, and therefore all the information must be gathered to and shared via this CPU. An example of the multiple codes scheme can be seen in Fig. 8. Typically, a coupling application is placed between the different codes, handling the transfer of data (see e.g. Kochunas et al., 2012). This type of coupling is especially limiting for transient calculations where a high number of transfers between the codes is necessary.

In the single code approach, that is chosen in the present work, all parts of the problem are solved in the same code. This spares not only the external transfer but raises possibilities for a more efficient parallelization. Avoiding any external transfer of data via IO also permits a higher resolution of the coupling since the overhead is basically removed. An example of such a scheme is given in Fig. 9. The transfer of information can now be done by direct memory access. The iterative algorithm can also be switched between

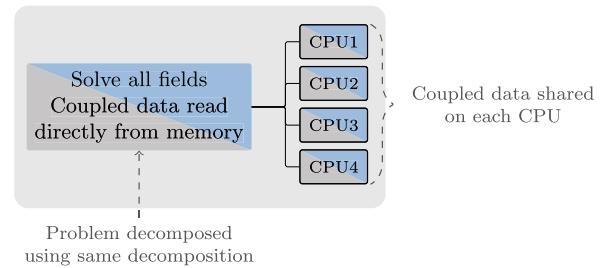


Fig. 9. Example of single code coupling scheme avoiding external transfer.

the different fields without the overhead cost of the multiple codes approach, which is also typically needed for the time stepping in a transient calculation.

## 6.3. Parallelization by shared domain decomposition

To parallelize the problem, different strategies could be followed. The domain could be split in different spatial regions, corresponding to a domain decomposition, or the problem could be split in its different fields of physics each solved on a separate set of CPUs, corresponding to a functional decomposition (Calvin and Nowak, 2010). Equivalent to multiple codes parallelized on separate CPUs (as in Fig. 8 and discussed in Section 6.2), the coupled data information transfer in a functional decomposition becomes an obstacle. If the parallelization is instead based on a common domain decomposition for all fields, as displayed in Fig. 9, the exchange of information will take place locally for each CPU, minimizing the amount of data that needs to be updated between CPUs. Such an argument is of particular importance since the fine-mesh direct coupling results in a large quantity of data to be shared.

Applying the domain decomposition method, only the data at the domain interfaces need to be exchanged during the solution of the equations. While the foam-extend-3.0 structure for parallelization is employed, the information transfer is handled according to a zero-halo layer approach, i.e. there is no overlapping between the decomposed domains. In practice, the boundary data exchange occurs only at the point of the sparse matrix solution, or prior to a sweep in the neutronics (corresponding to prior to “sweep to update  $\Psi_{mg}$ ” in Fig. 1).

To get the same decomposition for the neutronic and thermal-hydraulic meshes, the geometry is in this case decomposed along planes parallel to  $x$ ,  $y$  and  $z$ -axis of the system. In general, the decomposition should be done such that the amount of work to

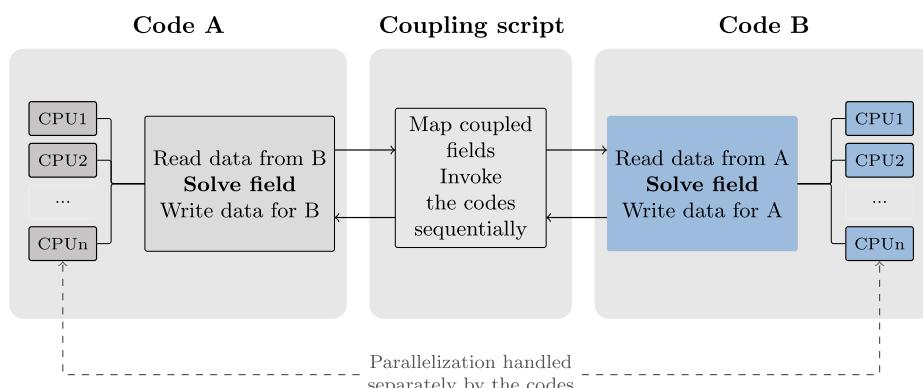
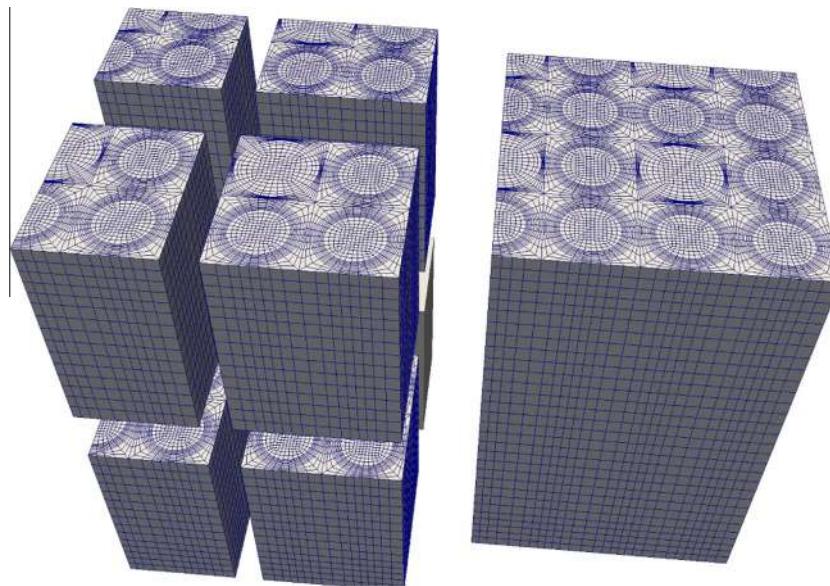


Fig. 8. Example of data transfer in the multiple codes approach.



**Fig. 10.** Example of decomposition for a quarter of a  $7 \times 7$  pins lattice.

**Table 1**

Geometry specification for the simulated assembly, with control rod guide tube values in brackets.

Fuel pin radius	0.41 cm
Cladding inner radius	0.43 cm (0.48 cm)
Cladding outer radius	0.49 cm (0.58 cm)
Pitch	1.25 cm
Fuel height	100 cm
Bottom reflector	20 cm
Top reflector	20 cm

**Table 4**

Neutronic parameters.

$N$ (Eq. (1))	8
$G$ (Eq. (1))	8
$L$ (Eq. (6))	2
Total power	400 kW

**Table 2**

Mesh specification for the simulated assembly.

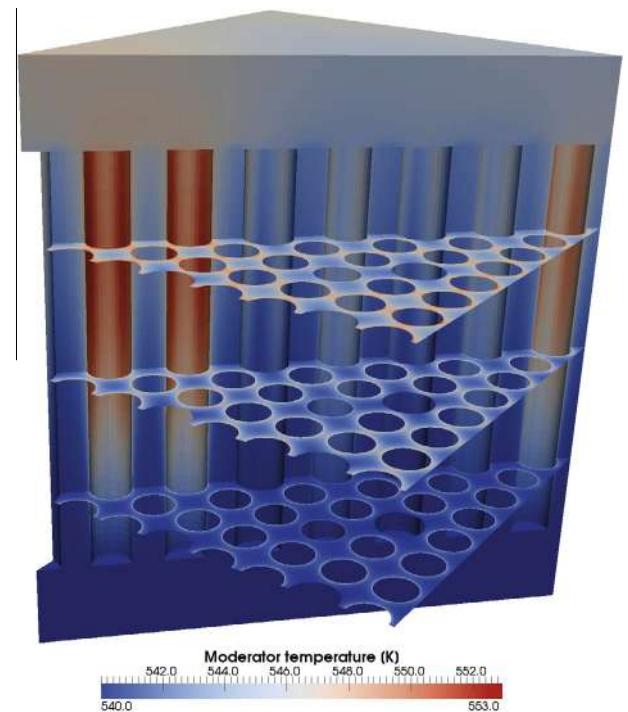
Region	Number of cells
Moderator	6,088,000
Fuel (per pin)	8000
Cladding (per pin)	4800
Gap (per pin)	1600
Neutronics	798,000

**Table 3**

Selected boundary conditions and inlet conditions.

Quantity	Initial condition	Inlet	Outlet
Temperature	540 K	540 K	Zero gradient
Pressure	15 MPa	Zero gradient	15 MPa
Velocity	2 m/s $\hat{z}$	2 m/s $\hat{z}$	Zero gradient
Neutron angular flux	1.0	Outgoing direction: zero gradient Incoming direction: fixed value zero	

be done by each CPU is equivalent. Since the loading will potentially be different for different equations, there must be a compromise. The shared decomposition restricts all used meshes to have faces along the chosen lines for splitting, so that every cell in all meshes exists entirely on a single CPU. In the case of the regular pattern of a fuel bundle this condition is not a major restriction. An example



**Fig. 11.** Moderator temperature at three horizontal planes, with the axial dependence at a diagonal cut in the background.

of the decomposition can be seen in Fig. 10 for a  $7 \times 7$  fuel pins system. The example system is split between the fuel pins and at mid-elevation.

## 7. Application and results

To illustrate the capabilities of the methods and implementations described above, we apply the developed code to a quarter of a  $15 \times 15$  fuel lattice, with reduced height with respect to a fuel assembly. The steady-state problem is solved for conditions typical of a PWR.

### 7.1. Geometry and mesh

The geometry of the simulated system is specified in [Table 1](#), and the material composition in the horizontal plane corresponds to the presented case in [Fig. 2](#). No spacers are included in the simulated system. Reflectors are modeled at the bottom and the top of the assembly, here consisting of moderator only without any other structural parts, as the inlet orifice and the top nozzle.

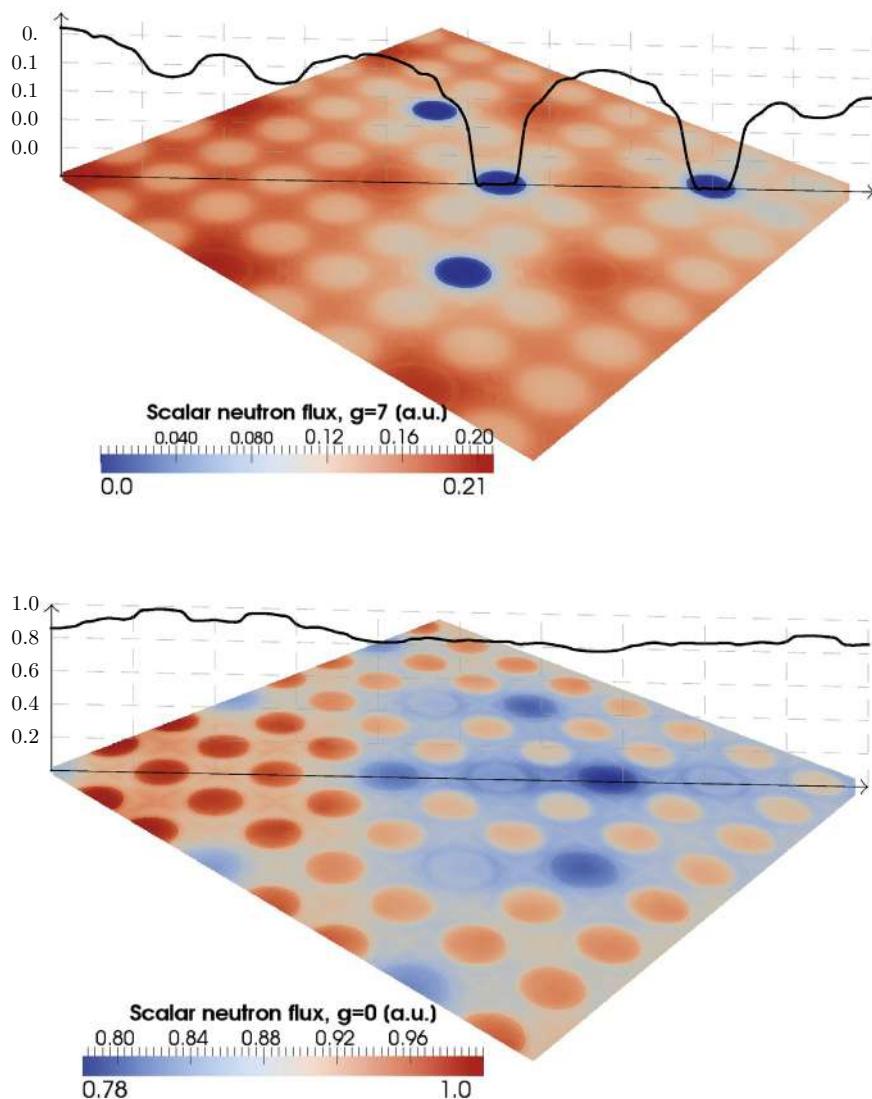
The applied mesh is summarized in [Table 2](#). All meshes consists of prism and hexahedron elements only. The mesh is generated using an in-house developed software, which utilizes the repeating lattice structure of the assembly to generate a discretization with the same mesh characteristics for all pin cells. The generated mesh

is written in a foam-extend-3.0 mesh format. The developed application also generates and prepares all boundary and initial conditions for all fields calculated.

### 7.2. Boundary and initial conditions

The axial boundary conditions are given as inlet and outlet boundary conditions, specified for all fields at the lower faces of the bottom reflector mesh and at the upper faces of the top reflector mesh. The conditions are specified as fixed value (Dirichlet) or zero gradient (homogeneous Neumann) conditions, with a special treatment for the case of the angular neutron flux. In the horizontal direction, reflective boundary conditions are used for all fields.

For the angular neutron flux ( $\Psi_{mg}$ ), a special implementation of the reflective boundary conditions is needed. Using the symmetry of the level symmetric weights and directions for  $S_N$ , a reflected direction can always be found for planes with normals parallel to the Cartesian coordinate axes. Thus for an incoming direction, the corresponding (i.e. reflected) outgoing direction is determined and the outgoing value of the flux is directly applied as an inlet condition for the incoming direction.



**Fig. 12.** Scalar flux at mid-elevation for the fast group ( $g = 0$ , bottom) and the thermal group ( $g = 7$ , top).

Constant initial conditions are used in all fields, but, as described in Section 5.3, during the first iterations not all fields are solved. Selected boundary and initial conditions are given in Table 3.

### 7.3. Domain decomposition

The system is decomposed into 64 parts, using 3 cutting planes in each of the Cartesian directions and in such a way that each cutting plane is along faces in all of the meshes, as described in Section 6.3. No automatic load balancing is applied. However, due to the computational burden of the discrete ordinates method, the neutronics mesh is the most important to balance, with the aim of an even computational effort on all CPUs.

The choice of decomposition is not unique and to optimize the performance not only an even distribution of cells should be the target. In our case with an axially coarser and horizontally finer mesh, also the cutting plane direction will have a major significance on the performance. Since only face values will need to be transferred, as few faces as possible should be transferred. Consequently, from a data transfer point of view, in our case it is better to cut in planes with normals  $\hat{x}$  and  $\hat{y}$ .

### 7.4. Neutronic parameters

The settings for the neutron transport solver are given in Table 4. Eight discrete energy groups are used, based on the group structure from CASMO-4 (Studsvik Scandpower, 2009). We apply  $S_8$ , which corresponds to 80 discrete directions for each energy group, for a total of 640 neutron flux fields. A larger set of tests for the discrete ordinates, applying the same code was performed and reported elsewhere (Jareteg et al., 2014b). The total power is

the power integrated over all the fuel pins, and is employed to normalize the scalar and angular neutron fluxes.

### 7.5. Results

The presented case was run on 64 Nehalem CPUs (Intel® Xeon® E5520, 2.27 GHz), divided on 8 computational nodes, with a total wall-clock running time of 48,000 s ( $\approx 14$  h).

The moderator temperature distribution is shown in Fig. 11. As can be seen, both horizontal and axial heterogeneous distributions are achieved. The maximum axial temperature occurs at the top of the lattice, just below the top reflector, and the maximal horizontal temperature at the points where the distance between the pins is smallest.

Fig. 12 gives the scalar flux at mid-elevation for the fast and the thermal group. The flux profile along the symmetry line at mid-elevation is also given. The lowest energy group flux has strong minima in the pins which partially consist of burnable absorber, and maxima in the empty fuel rod channels. In contrast, the highest energy flux has minima in the water channels.

Considering both the slices and the line plots, some ray effect can be found. Such artifacts are typical for the discrete ordinates methods, and arise from the inability to reconstruct the angular flux with a set of few ordinates (Lewis and Miller, 1984). The easiest remedy against this is to increase the order of the method, which will however also increase the computational time.

The convergence of the coupled system, with residuals for each separate equation, is given in Fig. 13. The diagram displays the first eight outer iterations (each corresponding to a full loop in Fig. 7), with the convergence for the subiterations.

As explained in Section 5.3, during the first iteration only the diffusion neutronics is solved, whereas in the second iteration

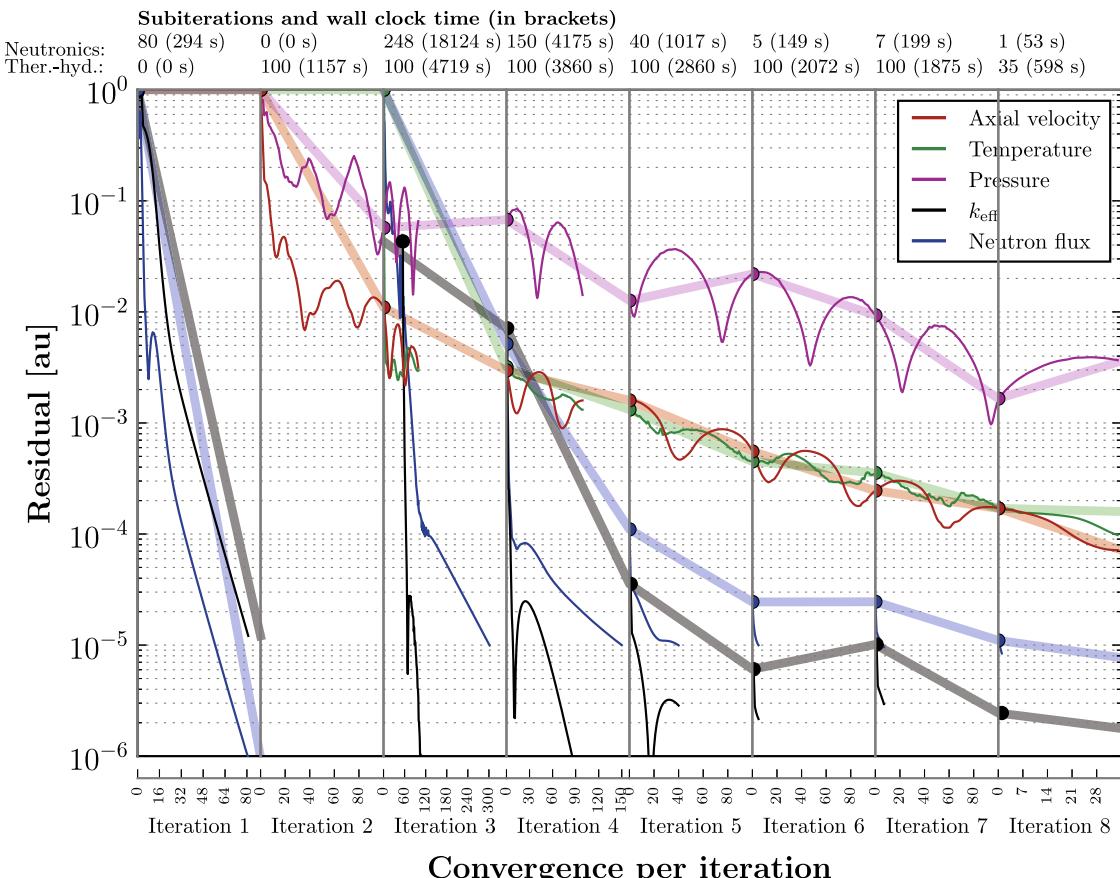


Fig. 13. Convergence results for the coupled system, with outer iteration convergence as opaque broader lines and the corresponding inner iterations as thinner lines.

the pressure and velocity are calculated. In the third iteration the  $S_N$  solver, as well as the temperature equation, are added and during the fourth iteration the first update of the cross-sections occurs. As seen from the figure, the interdependence is close to convergence after the seventh iteration. From iteration nine a single  $S_N$  sweep occurs for each outer iteration, and the total change in  $k_{eff}$  is only 10 pcm from iteration 8 to iteration 50. The thermal-hydraulics solver takes in total 50 iterations to converge. However, much less than 100 subiterations for most of the iterations are required. The slower convergence of the thermal-hydraulics solver is a property of the applied CFD-algorithm, and is not connected to the coupled solver.

From Fig. 13, it is shown that the largest effort is spent for the first neutronic iteration including  $S_N$  (Iteration 3). After this iteration the thermal-hydraulics requires the majority of the time. Summing over all iterations, the thermal-hydraulic and neutronic calculations take 45.1% and 54.1% of the total time, respectively. For such comparison it should also be noted that the thermal-hydraulic mesh is in the presented case finer than the neutronic mesh (see Table 2) and also that the computational time for the neutronics depends on the chosen number of energy groups and directions.

For the thermal-hydraulics the subiteration convergence is not monotonic. Instead, a periodic behavior was observed, where the axial velocity and the pressure residuals decrease and increase out of phase. Again, such a behavior is a property of the algorithms and matrix solvers applied and not any artifact of the coupling.

## 8. Summary and conclusion

We present a fine-mesh solver for the coupled neutronic and thermal-hydraulic problem that can be applied to PWR sub-pin level calculations. The work focuses on the methods and the implementation strategy of such a framework.

The use of HPC and fully parallelized solvers are both pointed out as key issues. The fine-mesh approach results in a large number of computational cells, relying on the use of computational clusters to solve the problem in a feasible time. The implementation of the decomposition necessary for the parallelization is described, along with a scheme minimizing the amount of data transfer by keeping all fields of physics in the same spatial region on the same CPU.

The methodology is implemented as a standalone application based on the open source C++ library foam-extend-3.0. It includes a neutronic solver based on the discrete ordinates method, and a thermal-hydraulic solver based on the mass, momentum and energy equations, complemented by a turbulence model. Both fields of physics are solved using the same simulation framework. This feature is essential to directly formulate the coupling on the fine-mesh level. In addition, a fully conservative mesh mapping scheme is included, and it aims at exchanging coupled data between different meshes using the finite volume methodology.

The fine-mesh solver is tested for the case of a quarter of a  $15 \times 15$  fuel pin lattice, and a converged coupled solution is reached. Physically correct dependencies are obtained for the simulated variables and the results confirm that the present coupled scheme works. It also shows that the algorithms and implementations are efficient enough to produce converged results on a fine-mesh within 14 h using 64 CPUs. The convergence behavior of the coupled solver points out that there is much room for improvement of the separated models for neutronics and thermal-hydraulics. However, the convergence of the multiphysics problem demonstrates that the presented iterative scheme is working well to resolve the coupled dependencies.

Many interesting and challenging areas need to be investigated, including a future extension to two-phase flow simulations, so that

BWR cases and departure from nucleate boiling in PWRs can also be considered. Such an extension requires not only to formulate and implement accurate two-fluid models, but it also poses a new type of coupled problem, including new conjugate heat transfer regimes, and an anticipated stronger feedback to the neutronics. Furthermore, other fields of physics could be fit in the same framework, including explicit thermal expansion, fluid–structure interaction and others.

In the presented simulation, reflective boundary conditions are used for all fields in the horizontal direction, disregarding the global dependence of all fields. Future work could tackle this approximation by either running a larger case using the presented framework (thus realized by investing more computational effort) or by coupling multiple scales of resolution, only adding a coarse layer to the simulation. Whereas the first approach is easier but more computationally expensive, the second method is more challenging but it has the potential for a better use of the resources at hand.

## Acknowledgments

The Swedish Center for Nuclear Technology (SKC) is acknowledged for financially supporting this PhD project. This work is carried out within the DREAM (Deterministic Reactor Modelling) framework at Chalmers University of Technology. The computations were performed on resources at Chalmers Centre for Computational Science and Engineering (C3SE) provided by the Swedish National Infrastructure for Computing (SNIC).

## References

- Abu-Shumays, I.K., 2001. Angular quadrature for improved transport computations. *Transport Theory Stat. Phys.* 30 (2–3), 169–204, URL <<http://dx.doi.org/10.1081/TT-100105367>>.
- Bakosi, J., Christon, M.A., Lowrie, R.B., Pritchett-Sheets, L.A., Nourgaliev, R.R., 2013. Large-eddy simulations of turbulent flow for grid-to-rod fretting in nuclear reactors. *Nucl. Eng. Des.* 262, 544–561, ISSN 0029-5493.
- Calvin, C., Nowak, D., 2010. High Performance Computing in Nuclear Engineering. Springer Science+Business Media, Url: <[http://dx.doi.org/10.1007/978-0-387-98149-9\\_12](http://dx.doi.org/10.1007/978-0-387-98149-9_12)>.
- Cardoni, J., Rizwan-uddin, 2011. Nuclear Reactor Multi-physics Simulations with Coupled MCNP5 and STAR-CCM+. M&C 2011, Rio de Janeiro, Brazil.
- Ferziger, J., Peric, M., 2002. *Computational Methods for Fluid Dynamics*. Springer.
- Gaston, D., Newman, C., Hansen, G., Lebrun-Grandié, D., 2009. Moose: a parallel computational framework for coupled systems of nonlinear equations. *Nucl. Eng. Des.* 239 (10), 1768–1778, URL <<http://www.sciencedirect.com/science/article/pii/S0029549309002635>>.
- Hamilton, S., Clarno, K., Berrill, M., Evans, T., Davidson, G., Lefebvre, R., Sampath, R., Hansel, J., Ragusa, J., Josey, C., 2013. Multiphysics Simulations for LWR Analysis. M&C 2013, Sun Valley, Idaho.
- Hébert, A., 2010. Multigroup Neutron Transport and Diffusion Computations. Springer-Verlag, URL <[http://dx.doi.org/10.1007/978-0-387-98149-9\\_8](http://dx.doi.org/10.1007/978-0-387-98149-9_8)>.
- Ivanov, K., Avramova, M., 2007. Challenges in coupled thermal hydraulics and neutronics simulations for LWR safety analysis. *Ann. Nucl. Energy* 34 (6), 501–513, URL <<http://dx.doi.org/10.1016/j.anucene.2007.02.016>>.
- Jareteg, K., Vinai, P., Demazière, C., 2013. Investigation of the Possibility to Use a Fine-mesh Solver for Resolving Coupled Neutronics and Thermal-hydraulics. M&C 2013, Sun Valley, Idaho.
- Jareteg, K., Vinai, P., Demazière, C., 2014a. Fine-mesh deterministic modeling of PWR fuel assemblies: proof-of-principle of coupled neutronic/thermal hydraulic calculations. *Ann. Nucl. Energy* 68 (0), 247–256, URL <<http://www.sciencedirect.com/science/article/pii/S0306454914000036>>.
- Jareteg, K., Vinai, P., Sasic, S., Demazière, C., 2014b. Influence of an  $S_N$  Solver in a Fine-mesh Neutronics/Thermal-hydraulics Framework. Submitted to PHYSOR 2014, September 28–October 3, Japan.
- Jasak, H., 2009. General Grid Interface – Theoretical Basis and Implementation. NUMAP-FOAM Summer School, Zagreb, 2–15 September, 2009.
- Kochunas, B., Stimpson, S., Collins, B., Downar, T., Brewster, R., Baglietto, E., Yan, J., 2012. Coupled Full Core Neutron Transport/CFD Simulations of Pressurized Water Reactors. PHYSOR 2012, Knoxville, Tennessee, USA, April 15–20.
- Larsen, E.W., Morel, J.E., 2010. Advances in discrete-ordinates methodology. In: Nuclear Computational Science. Springer, pp. 1–84, URL <[http://link.springer.com/chapter/10.1007/978-90-481-3411-3\\_1](http://link.springer.com/chapter/10.1007/978-90-481-3411-3_1)>.
- Leppänen, J., 2012. Serpent – a Continuous-energy Monte Carlo Reactor Physics Burnup Calculation Code, VTT Technical Research Centre of Finland.

Lewis, E., Miller, W.J., 1984. Computational Methods of Neutron Transport. John Wiley & Sons, Inc., USA.

MPI Forum, 2009. MPI: A Message-Passing Interface Standard. <<http://www.mpi-forum.org/>>.

Newman, C., Hansen, G., Gaston, D., 2009. Three dimensional coupled simulation of thermomechanics, heat, and oxygen diffusion in UO<sub>2</sub> nuclear fuel rods. *J. Nucl. Mater.* 392 (1), 6–15.

Page, M., Beaudoin, M., Giroux, A.M., 2010. Steady-state capabilities for hydro turbines with OpenFOAM. *IOP Conf. Ser.: Earth Environ. Sci.* 12, 012076, URL <<http://dx.doi.org/10.1088/1755-1315/12/1/012076>>.

Panton, R., 2005. Incompressible Flow, third ed. Wiley.

Patankar, S., Spalding, D., 1972. A calculation procedure for heat, mass and momentum transfer in three dimensional parabolic flows. *Int. J. Heat Mass Transfer* 15, 1787–1806.

Plimpton, S., Hendrickson, B., Burns, S., McLendon, W., Rauchwerger, L., 2005. Parallel Sn sweeps on unstructured grids: algorithms for prioritization, grid partitioning and cycle detection. *Nucl. Sci. Eng.* 150, 267–283.

Studsvik Scandpower, 2009. CASMO-4.

Sutherland, I.E., Hodgman, G.W., 1974. Reentrant polygon clipping. *Commun. ACM* 17 (1), 32–42, URL <<http://doi.acm.org/10.1145/360767.360802>>.

Wikki, 2014. Foam-extend-3.0. (last visited March 9, 2014). URL: <<http://wikki.gridcore.se/foam-extend>>.

Yan, J., Kochunas, B., Hursin, M., Downar, T., Karoutas, Z., Baglietto, E., 2011. Coupled Computational Fluid Dynamics and MOC Neutronic Simulations of Westinghouse PWR Fuel Assemblies with Grid Spacers. NURETH-14, Toronto, Ontario, Canada, September 25–30.

Zou, L., Peterson, J., Zhao, H., Zhang, H., Andrs, D., Martineau, R., 2013. Solving Implicit Multi-mesh Flow and Conjugate Heat Transfer Problems with RELAP-7. M&C 2013, Sun Valley, Idaho.

# Paper V

*“Development and test of a transient fine-mesh LWR multiphysics solver in a CFD framework”*

K. Jareteg, R. Andersson, C. Demaziére

*M&C 2015, Nashville, Tennessee (2015)*



# DEVELOPMENT AND TEST OF A TRANSIENT FINE-MESH LWR MULTIPHYSICS SOLVER IN A CFD FRAMEWORK

**Klas Jareteg, Rasmus Andersson, and Christophe Demazière**

Chalmers University of Technology, Department of Applied Physics  
SE-412 96 Gothenburg, Sweden

klas.jareteg@chalmers.se; rasande@student.chalmers.se; demaz@chalmers.se

## ABSTRACT

We present a framework for fine-mesh, transient simulations of coupled neutronics and thermal-hydraulics for Light Water Reactor (LWR) fuel assemblies. The framework includes models of single-phase fluid transport for the coolant and conjugate-heat transfer between the coolant and the fuel pins, complemented by a neutronic solver. The thermal-hydraulic models are based on a CFD approach, resolving the pressure and velocity coupling via an iterative algorithm. Similarly, the neutronics is formulated in a fine-mesh manner with resolved fuel pins. The neutronic and thermal-hydraulic equations are discretized and solved in the same numerical framework (foam-extend-3.1). A test case of a quarter of a fuel pin is used to test the transient behavior of the code for a set of different initial reactivities. The same geometry is used to simulate a decrease of the inlet temperature, which demonstrates the response both in the CFD and the neutronics for an increase in reactivity. Furthermore, a system of  $7 \times 7$  fuel pins is simulated with the same inlet temperature decrease and we present the temporal development of the temperature as well as an analysis of the heterogeneities captured by the fine-mesh approach. The solver is shown to capture the transient multiphysics couplings and demonstrates the numerical and computational applicability based on the presented cases.

*Key Words:* Coupled neutronics/thermal-hydraulics, CFD, nuclear reactor multiphysics, transient analysis, fine-mesh

## 1 INTRODUCTION

The core of a LWR constitutes a major challenge from a modeling point of view. A wide range of physical phenomena, such as fluid transport, heat transfer, neutron transport and the dependencies on the material properties, govern the behavior of the core. In addition to the multiphysics perspective, the reactor core is also a multiscale system, with scales ranging from the atomic scales to the global behavior on the core scale. To accurately model the system, all those scales must be taken into account.

The multiphysics problem of the reactor core has typically been resolved by only modeling the larger scales, ultimately solving for the global behavior of the core with smaller scales information provided by pre-computed data or empirical correlations [1]. The neutronic solver is commonly based on piece-wise homogenized fuel assemblies, where the finer scales are not fully recovered but accounted for when generating macroscopic, i.e. assembly-wise, parameters. In a similar manner, the thermal-hydraulic solver is often applied to a number of channels corresponding to the

number of fuel assemblies or less. Again, the microscopic scales are not directly treated. Although successfully applied to transient safety calculations, such a coupled computational scheme can only capture the multiphysics on a coarse scale. The sub-fuel assembly couplings are disregarded, or at the most only recovered by the use of empirical relations.

The development of computational power has in recent years led to an increased use of Computational Fluid Dynamic (CFD) calculations for nuclear reactor applications. CFD has been applied to a large range of problems including both single- and two-phase flow calculations. Phenomena considered include Grid-to-Rod fretting [2], thermal mixing due to spacers [3], two-phase flow in fuel assemblies [4] and many more. The fine-mesh approaches, typical for the mentioned calculations, allow for solving the problems using first principles and thus avoid or limit the use of empirical relations and macroscopic models.

Recently an increased interest in doing coupled neutronic and thermal-hydraulic calculations on fine scales has been seen. Instead of the coarse-mesh multiphysics, the different fields are combined with a higher resolution of the coupling terms. Although moderately fine resolution calculations of coupled CFD and neutronics have been applied to full core problems [5], the computational burden of such calculations is still severe, especially compared to routine reactor core calculations. In contrast to the full core calculations, algorithms and models aimed at and applicable on one or a few fuel assemblies can give valuable insight to the coupled physics and still be solved with a moderate computational cost. Such sub-core scale simulations are still of relevance and importance for the global core problem, e.g. spacer CFD calculations can aid the design process and thus lead to improved designs of fuel assemblies. Also for the coupled neutronics and thermal-hydraulics problem, fuel assembly calculations could provide interesting results considering aspects such as the interplay between the local moderator density and the neutronic fields in Boiling Water Reactor (BWR) fuel assemblies. It should be noted that the core-wise, global dependencies still need to be taken into account as boundary conditions in the fine-mesh approach and a multiscale approach is thus required. Although here focused on CFD and neutronics, other areas of coupled fine-mesh calculations are also of interest, e.g. high resolution simulations of fuel performance analysis [6]. In the same manner that CFD calculations allow for higher resolution and fewer correlations and can produce correlations for coarser methods, the resolved coupled problems could potentially be used to refine the coarse coupled models and give new insights for optimal design of nuclear fuel assemblies.

In some coupling methodologies the fine-mesh calculations have been achieved by combining existing CFD and neutronic solvers (e.g. [7, 8]). On the other hand, some presented methodologies are based on an integrated coupling, where all fields of physics are solved in the same solver (e.g. [9]). Whereas coupling of existing software has intrinsic benefits, especially considering verification and validation, there are also major drawbacks, as further discussed in [10]. One of the main benefits of using a single code for the whole multiphysics problem is the advantage in computational performance. Using a single software or single code allows for a direct sharing of all coupled quantities and also permits parallelization schemes shared between the neutronics and the thermal-hydraulics. The advantage in computational efficiency is particularly important for transient multiphysics calculations where a large number of information exchanges or data transfers between separate codes could otherwise limit the performance. As a result of the computational

burden and novelty of the field, few transient fine-mesh methodologies are present in the literature.

In this paper we present a fine-mesh coupled neutronic and thermal-hydraulic solver, implemented as an integrated single solver. The solver is based on a deterministic, finite volume, fine-mesh framework and the code is parallelized based on domain decomposition. The capabilities include a neutronic diffusion solver as well as CFD algorithms for single-phase flow calculations and conjugate heat transfer between fuel and coolant. Furthermore, the geometry is recovered on an unstructured grid, and each separate material region (fuel, gap, cladding and moderator) are separated, resulting in a heterogeneous system. Whereas earlier work based on the same framework has been successfully applied to coupled fine-mesh steady-state problems [9, 10], we here focus on the transient multiphysics problem.

The presented methodology and simulations are focused on the coupling algorithm and its implications for a fine-mesh LWR transient system. We investigate the convergence properties as well the computational costs associated with the multiphysics problem. The transient solver is applied to an example case including a small lattice of fuel pins with single phase liquid water as the moderator. The solver is successfully applied to sub- and supercritical initial conditions and to time-dependent inlet temperature variations. The solver allows calculating the time-dependent and space-dependent neutronic and thermal-hydraulic fields, including temperature, flow properties, power distribution as well as the neutron flux.

## 2 METHOD

In this section, we present the method used to solve the neutronics and thermal-hydraulics, followed by an overview of the coupling algorithm. To achieve a consistent coupled solver, it is of importance to select methods which result in comparable spatial and temporal precision and accuracy. In the presented approach, an unstructured grid, finite volume approach is followed for both the neutronics and thermal-hydraulics.

### 2.1 Neutronic solver

For the neutronics, we apply a multigroup diffusion method to resolve the spatial resolution, temporal resolution and the energy dependence of the neutron distribution. It has previously been shown that the diffusion approximation is inferior to a discrete ordinates approach for a resolved, heterogeneous fuel pin lattice [11]. However, since the focus of the present paper is primarily on the coupling to the CFD approach, a diffusion solver is applied in the first attempt to perform the transient fine-mesh simulations. The time-dependent multigroup diffusion equation for the neutron flux ( $\Phi_g$ ) is given by

$$\begin{aligned} \frac{1}{v_g} \frac{\partial \Phi_g(\mathbf{r}, t)}{\partial t} &= \nabla \cdot D_g(\mathbf{r}, t) \nabla \Phi_g(\mathbf{r}, t) - \Sigma_{T,g}(\mathbf{r}, t) \Phi_g(\mathbf{r}, t) + \sum_{g'=1}^G \Sigma_{s0,g' \rightarrow g}(\mathbf{r}, t) \Phi_{g'}(\mathbf{r}, t) \\ &+ (1 - \beta) \chi_g^p \sum_{g'=1}^G \nu_{g'}(\mathbf{r}, t) \Sigma_{f,g'}(\mathbf{r}, t) \Phi_{g'}(\mathbf{r}, t) + \chi_g^d \sum_{i=1}^I \lambda_i C_i(\mathbf{r}, t), \quad g = 1, \dots, G, \end{aligned} \quad (1)$$

with the precursor concentrations ( $C_i$ ) calculated as

$$\frac{dC_i(\mathbf{r}, t)}{dt} = \beta_i \sum_{g'=1}^G \nu_{g'}(\mathbf{r}, t) \Sigma_{f,g'}(\mathbf{r}, t) \Phi_{g'}(\mathbf{r}, t) - \lambda_i C_i(\mathbf{r}, t), \quad i = 1, \dots, I, \quad (2)$$

and where standard notations are employed for all quantities. The equations are coupled through the neutron flux as well as the precursor concentrations. Furthermore, both Eq. (1) and Eq. (2) are coupled to the thermal-hydraulics via the dependence of the cross-sections on the temperature of the fuel and the density of the moderator. Based on the finite volume method, we discretize the diffusion term using a second order linear Gauss approach. It should be noted that the precursor equations are ordinary differential equations, and thus solved without any operator spatially discretized.

The required cross-sections and macroscopic parameters are pre-generated from lower dimensional simulations (2D) based on the Monte Carlo software Serpent [12]. There are multiple benefits of utilizing a pre-generated, temperature and density dependent table of cross-sections. First, whereas the diffusion solver is computationally cheap, the Monte Carlo solver requires considerably larger computational efforts. This implies that a pre-generated table, applied throughout the whole simulated transient, is more efficient than separate Monte Carlo runs for each time-step. Second, the Monte Carlo method is general in terms of the assembly geometry. This is useful for the resolved fuel pin lattices here simulated, where sub-pin resolution of the cross-sections are required. The cross-section generation is described in further detail elsewhere [10].

Given the cross-sections, the macroscopic parameters and the temperature, an iterative algorithm is applied to resolve the precursor and flux dependencies in Eqs. (1)-(2). The scheme is presented in the left hand side of Figure 1. After convergence of the energy and spatial dependence of the diffusion equations (Eq. 1), only a single sweep through the delayed precursor equations (Eq. 2) is necessary. An outer neutronics iteration is performed to ensure that the dependencies between the flux and neutron precursors are resolved. Whereas more sophisticated algorithms to resolve the dependence on the neutron precursors exist, the presented algorithm is used as a first and intuitive approach for the multiphysics solver.

## 2.2 Thermal-hydraulic solver

The thermal-hydraulic solver is based on a CFD approach for time-dependent and heated single-phase flow. The equations solved in the fluid include the mass continuity equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{U}) = 0, \quad (3)$$

the Navier-Stokes equations

$$\frac{\partial(\rho \mathbf{U})}{\partial t} + \nabla \cdot (\rho \mathbf{U} \otimes \mathbf{U}) = \nabla \cdot \boldsymbol{\tau} - \nabla P + \rho \mathbf{g} \quad (4)$$

and a temperature equation

$$\frac{\partial(\rho c_p T)}{\partial t} + \rho c_p \mathbf{U} \cdot \nabla T = \beta \mathbf{U} \cdot \nabla P + \nabla \cdot (K \nabla T) + q'''. \quad (5)$$

The pressure and velocity coupling is resolved by a standard PISO approach (see [13]). Furthermore, a Reynolds averaging is performed on all the equations, followed by the formulation of turbulence equations to close the set of averaged equations. In the presented simulations, we use the Standard  $k - \epsilon$  turbulence model with wall functions for all simulations.

For the solid regions, the temperature equation corresponds to

$$\rho c_p \frac{\partial T}{\partial t} = \nabla \cdot K \nabla T + q''', \quad (6)$$

where the volumetric source term ( $q'''$ ) corresponds to the energy released from fission in the fuel, as provided by the power density calculated in the neutronic solver. The volumetric source term is zero in the cladding. The energy equations for all material regions are formulated in terms of temperature. This allows for an implicit solution methodology. Instead of solving the temperature equation in each region separately and iterating between the regions, a common matrix is formulated. This allows all regions to be solved for at once, without iteration and it thus considerably accelerates the solver.

Given the volumetric heat source ( $q'''$ ), an iterative algorithm is applied to resolve the pressure and velocity dependence, the temperature equation and the coupling to the thermo-physical state. The iterative scheme is presented on the right-hand side of Figure 1. It should be noted that an update of the thermo-physical properties influences all material properties in the fluid equations ( $c_p, \rho, \beta, K, \tau$ ) as well as the neutronic cross-sections. We account for the temperature dependence of all mentioned fluid quantities by interpolation in thermo-physical tables. Due to the small axial variation of pressure over the core in a typical PWR, the pressure dependence is not accounted for in the thermo-physical data. An outer thermal-hydraulics iteration is performed to ensure the convergence of the coupling between pressure, velocity and temperature in all regions.

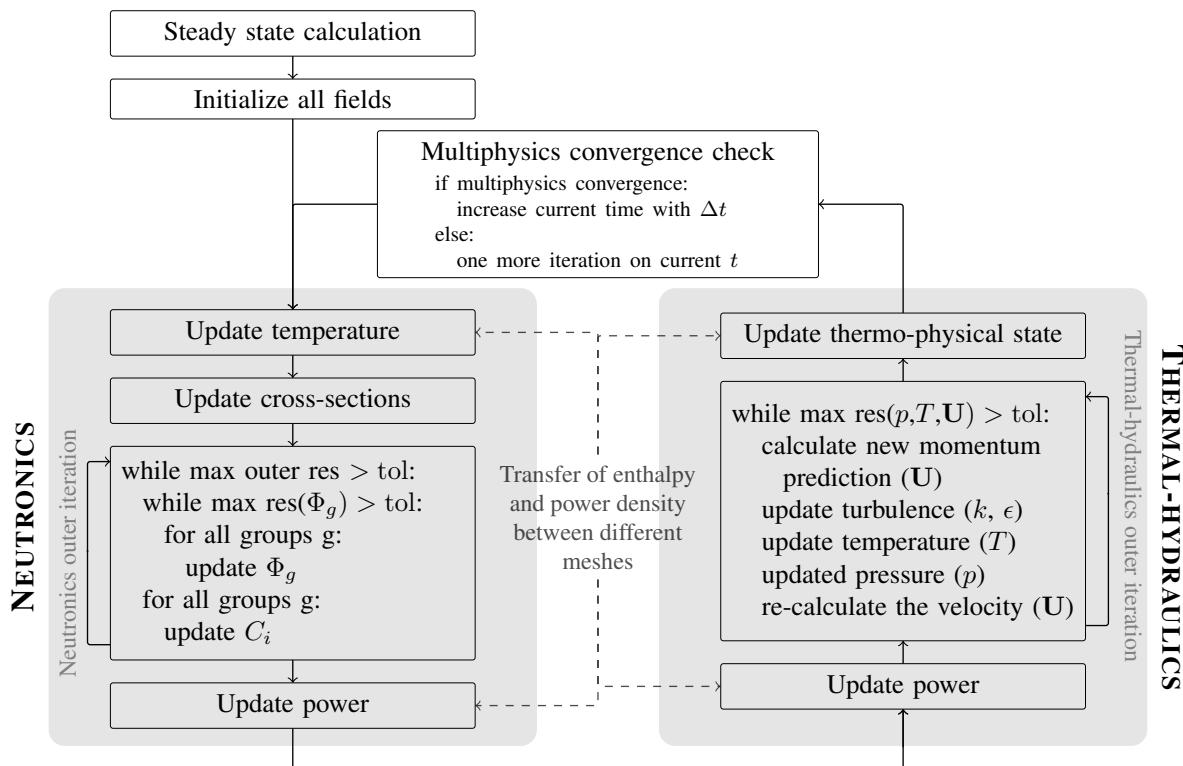
### 2.3 Coupling methodology

We apply an iterative scheme in space and time to resolve the coupling between the neutronics and the thermal-hydraulics. Recently, much attention has been given to implicitly coupled, nonlinear approaches (e.g. [6]). Such coupling schemes have potential advantages in terms of convergence rate for strongly coupled problems and it is feasible to use longer time steps with the same temporal accuracy. However, if only a few multiphysics iterations are required, the segregated methods will in many cases be less computationally expensive and more intuitive than a nonlinear approach. Furthermore, a segregated scheme often has the advantage of simplicity in contrast to the implicit approaches for which mathematically formalized interdependencies between the different fields of physics are required. For the segregated approach a black-box approach can be applied, solving each field separately but based on the coupled data. The argument of simplicity is particularly strong for complex methods, where the coupling to other fields is not easily formulated. From a CFD point of view, segregated schemes are commonly used for both pressure-velocity coupling and also for the coupling to the turbulence models. It should be emphasized that a fine-mesh coupling is still achieved since the fully resolved fields are used between the two different modules, i.e. neutronics and thermal-hydraulics.

In order to avoid an explicit scheme in time, iteration must be applied within each time step to resolve the dependencies in the segregated approach. We apply a multiphysics iteration for which the coupled convergence of the neutronics and thermal-hydraulics is evaluated. Two alternative time-schemes are evaluated: either based on a shared fixed time step or a shared time step negotiated between the neutronics and thermal-hydraulics. For the thermal-hydraulic solver, a maximum Courant number is typically applied. In the presented cases we use a fixed time-step, where the maximum Courant number never exceeds 0.3.

To best optimize the computational cost, separate meshes are used for the different fields of physics. This allows a finer mesh to be used in areas of large gradients for each separate field without applying the same refinement for all resolved quantities. The use of different meshes requires computational algorithms for mapping the meshes for the transfer of common data such as the temperature field. A fully automatic algorithm based on the volumetric overlap between different cells is used in the presented methodology [10].

The coupling scheme is outlined in Figure 1. As seen from the figure, the multiphysics iteration is performed until convergence for each time step. No underrelaxation is performed in either neutronics, thermal-hydraulics or for the multiphysics coupling.



**Figure 1. Iterative scheme applied to solve neutronics and thermal-hydraulics coupling.**

## 2.4 Implementation and code framework

The coupled solver is based on foam-extend-3 [14], formerly known as OpenFOAM®-1.6-ext. The code is an open-source, C++ library for solving CFD problems. The code is written in a generic, object-oriented manner and new fields of physics can be added. For the present paper, existing fluid solvers were extended to allow multiple material regions and to solve the temperature equation. For the neutronics, a previously developed solver [9, 10] was further extended to handle time-dependent simulation cases.

## 3 CASE DESCRIPTION AND NUMERICAL TESTS

To evaluate the validity of the coupled scheme and to present the functionality and performance of the implemented solver, two different test cases are presented. The first case, consisting of a quarter of a single fuel pin, is used to perform a reactivity analysis during an inlet temperature ramp as well as with constant boundary conditions for a set of different boron concentrations. The second test case, consisting of a quarter of a  $7 \times 7$  lattice, is used to investigate the coupled scheme for a larger case and to give an account of the heterogeneities that are resolved with the fine-mesh methodology.

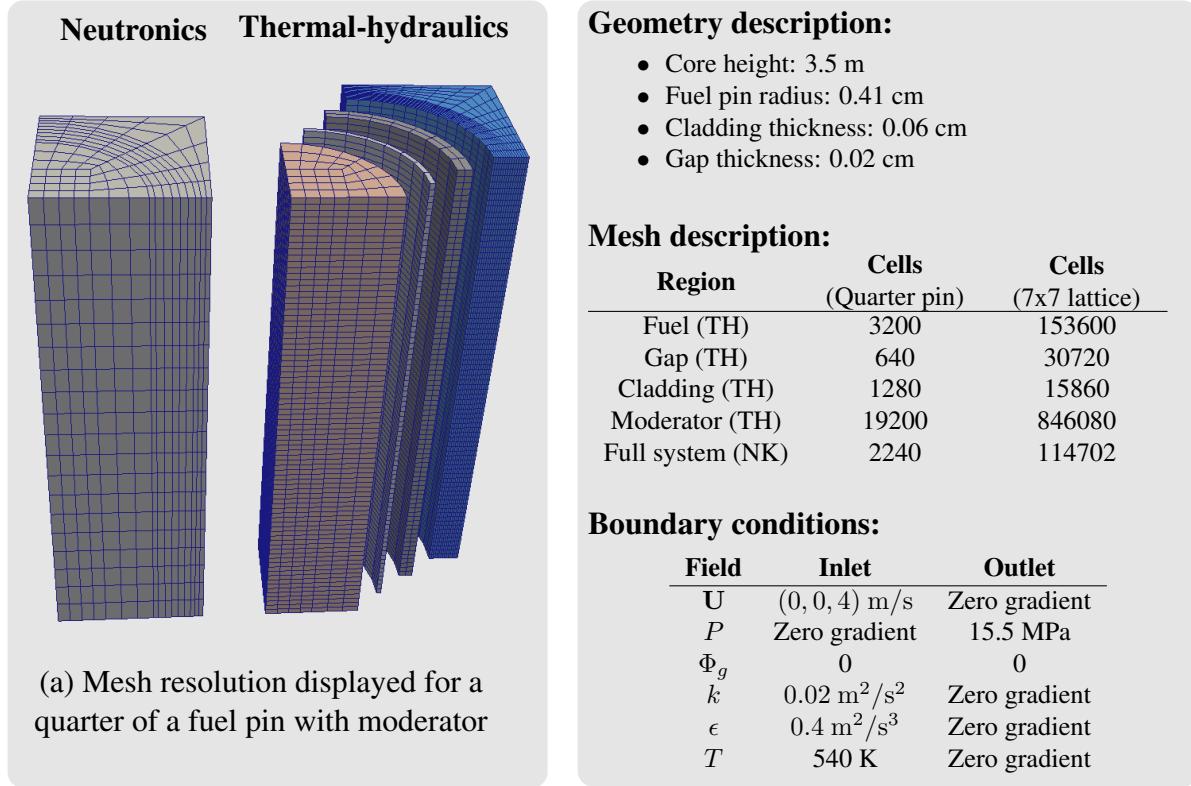
### 3.1 Geometry and boundary conditions

The geometrical regions and the mesh for the quarter fuel pin system is presented in Figure 2. The boundary conditions represent PWR conditions and the initial conditions are calculated by running a steady-state solver (earlier presented [10]) normalized with a total power condition. The power condition corresponds to a Hot Zero Power (HZP), i.e., corresponding to a low total power with a warm coolant. In the horizontal direction fully reflective boundary conditions are used. The conditions presented in Figure 2 are applied also to the  $7 \times 7$  lattice. The horizontal material regions of the  $7 \times 7$  case are displayed in Figure 3. The presented cases include no axial reflectors. Four neutronic energy groups and six delayed neutron precursor groups are used for all the presented results.

### 3.2 Case 1: Variation of boron concentration

In order to demonstrate the behavior of the solver for different initial reactivities, the boron concentration in the coolant is increased in five steps. This is actuated by generating cross-sections sets with different concentrations. In accordance with the methodology (see Figure 1), the initial state is determined by a criticality calculation. Probed values of the power density corresponding to each boron concentration are presented in Figure 4 for 10 simulated seconds.

As displayed in the figure, all cases except the 700 ppm case starts with a fast transient in the power density. This is explained by a prompt jump due to the initial reactivity as calculated from the  $k_{\text{eff}}$  eigenvalue problem in the steady-state solver. For 400 ppm, 500 ppm and 600 ppm an initial positive reactivity is seen. On the other hand, the 800 ppm steady state solution gives a negative



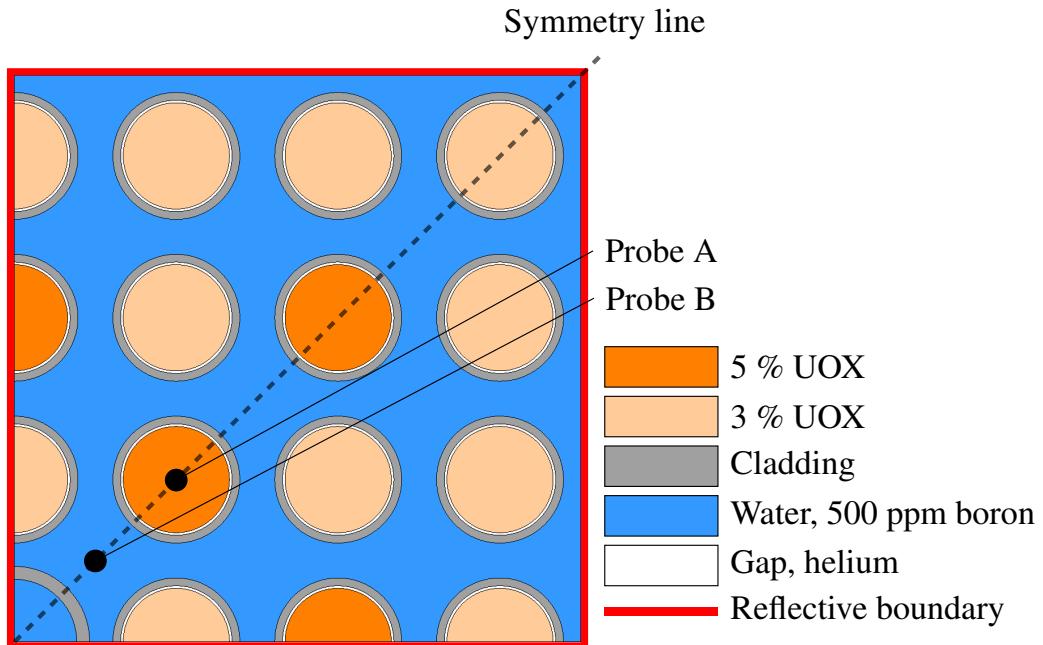
**Figure 2. Geometry and mesh description for the two systems used in the simulations. Index TH indicates meshes used for thermal-hydraulics, whereas NK indicates the mesh used for neutronics.**

initial reactivity. For the 700 ppm case, the steady-state calculation resulted in  $k_{\text{eff}} = 1.00002$  and accordingly there is no initial jump in the reactivity. Figure 4 also presents the reactivity ( $\rho$ ) at ten time-steps for the 700 ppm case. The reactivity values are calculated posterior to the transient calculation by applying the steady-state solver with all thermal-hydraulics data frozen to the values as calculated by the transient solver.

Due to the temperature feedback in the fuel and the moderator, the initially supercritical cases first rapidly increase in power, later to asymptotically approach a constant power density. The small overshots in the power density profile confirm slightly different time scales for the feedback of the neutronics and the thermal-hydraulics. The initial fast increase in power is after some tenths of a second somewhat lowered by the slightly postponed increase in temperature.

### 3.3 Case 2: Inlet temperature variation

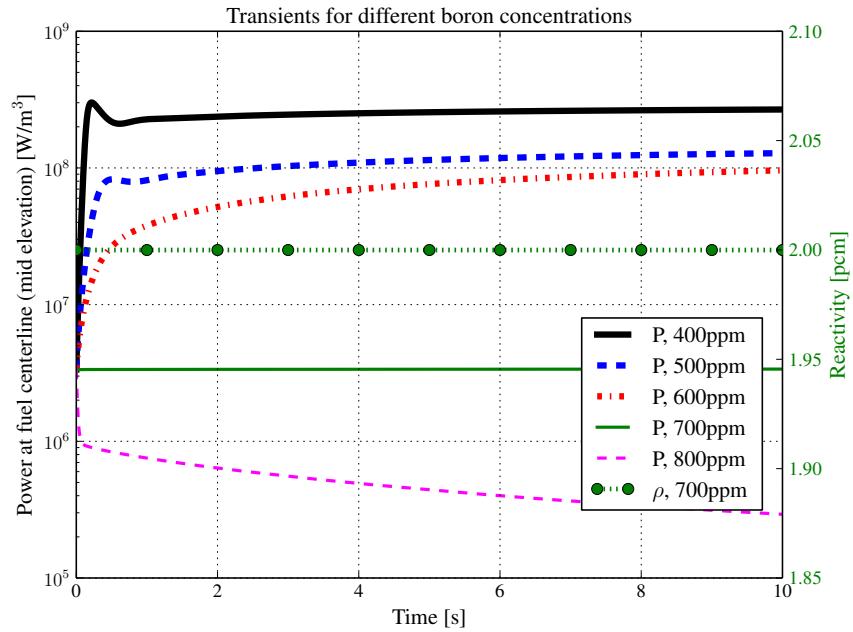
In addition to the variation in boron concentration, we perform an inlet temperature variation for all previously presented cases. The temperature at the inlet is at  $t = 2$  s lowered with a constant rate from 550 K to 540 K at  $t = 3$  s. The resulting power density probes are presented in Figure 5.



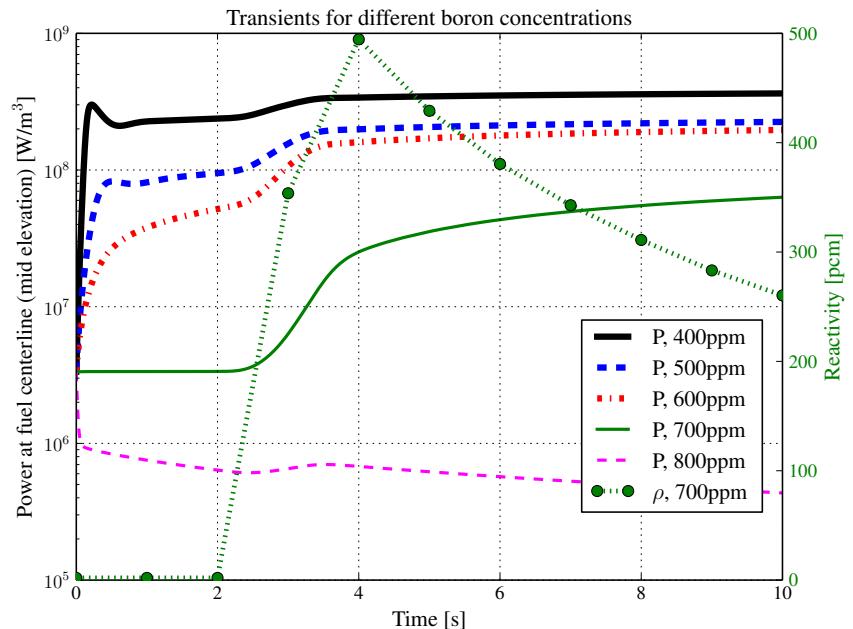
**Figure 3. Horizontal geometry for a quarter of a  $7 \times 7$  system, including horizontal probe locations used for post-processing.**

Whereas the first two seconds of the simulations are identical to Figure 4, the temperature ramped condition results in an increase in reactivity for all of the cases due to the enhanced moderation and the lowered fuel temperature. Considering the 700 ppm case, it is seen that the power density increases with more than one magnitude from the start to the end of the simulation. As the presented data correspond to mid-elevation, the effect in power is somewhat delayed due to the transport of the colder moderator. The initially large reactivity increase due the inlet temperature ramp is damped out after the transient boundary condition ends and the system asymptotically approaches a new steady state.

The presented case is not primarily aimed at explaining the physics of the presented system, but at demonstrating the coupling capability and at illustrating the influence of the CFD thermal-hydraulics on the fine-mesh neutronics. Given the complexity of the coupled calculations, also for the small test case presented above, further studies on aspects such as mesh-resolution, time-step dependence and coupled convergence are required.



**Figure 4.** Power density at the fuel centerline (mid-elevation) for five different boron concentrations, initiated from steady-state calculations, with the reactivity displayed for ten time-steps for 700 ppm.

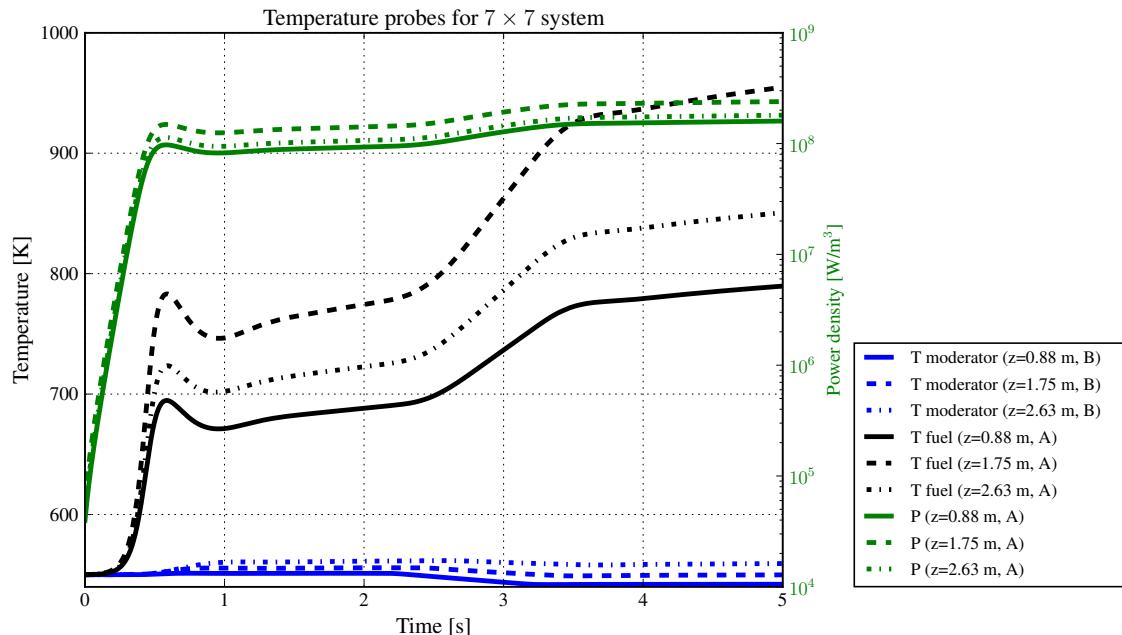


**Figure 5.** Power density at fuel centerline (mid-elevation) for an inlet temperature ramp from 550 K to 540 K between  $t = 2.0$  s and  $t = 3.0$  s, with the reactivity displayed for ten time-steps for the 700 ppm case.

### 3.4 Case 3: $7 \times 7$ lattice

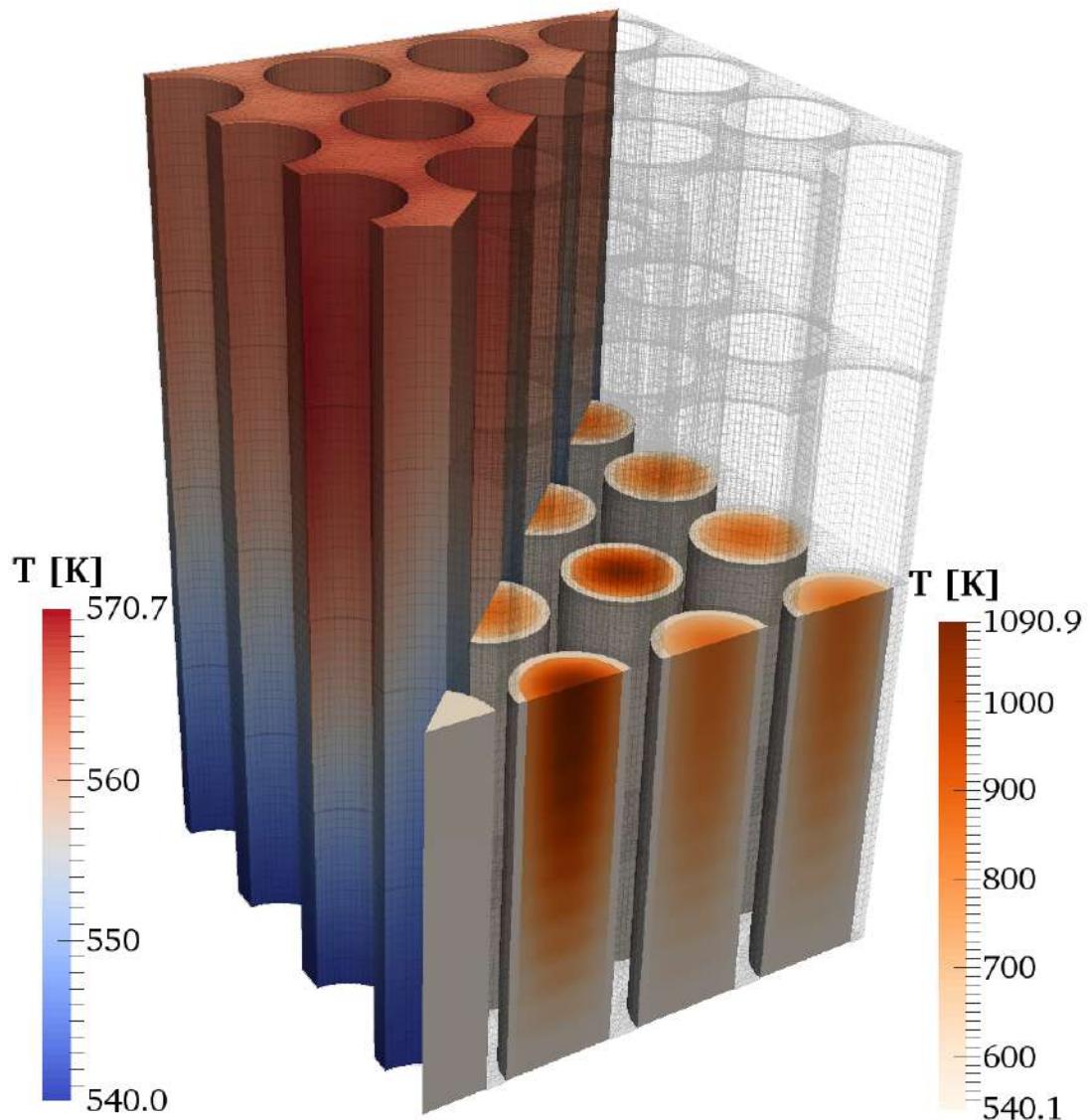
In the third case, we use a quarter of the  $7 \times 7$  lattice of fuel pins depicted in Figure 3. The simulations are run for a single boron concentration (500 ppm). The initial values are again retrieved from a steady-state calculation and the simulation is run with a constant time-step of  $\Delta t = 1$  ms. We apply the same inlet boundary condition as presented for Case 2, i.e., a ramp of the inlet temperature.

Figure 6 presents the temporal development of the fuel temperature, power density and moderator temperature for probe positions A and B (mid-elevation and with horizontal position shown in Figure 3). As displayed in the figure, the system is initially supercritical and accordingly there is a rapid increase in the fuel power density. At the beginning of the inlet temperature drop, the fuel temperature still slowly increases following the initial supercritical state. However, as the temperature of the moderator decreases the power further increases and the rate of the fuel temperature increase is larger. Complementary to the probed data, Figure 7 presents a three-dimensional view



**Figure 6.** Time-dependent temperature profiles for fuel and moderator and power density for fuel for positions *a* and *b* (Figure 3), respectively.

of the moderator and fuel temperature. Whereas the fuel temperature is strongly varying in the axial plane, only a small horizontal variation is seen at the top of the moderator. The higher power density of the higher enriched pins is evident both in the fuel and the moderator temperature.



**Figure 7. Moderator temperature to the left of the symmetry plane and fuel temperature to the right of the symmetry plane, cut at mid-elevation at  $t = 3$  s.**

In Figure 8, we present horizontal slices of the moderator temperature and the fuel power density (Figures 8a and 8c) and the relative difference in maximum and minimum temperature and power density (Figures 8b and 8d). The relative difference represents a measure of the heterogeneity in the two fields at mid-elevation, and displayed for each second throughout the 10 s of simulation. From the figure it is seen that there is initially a small heterogeneity in the moderator temperature, which almost doubles in magnitude following the decrease of the inlet temperature. On the contrary, the relative difference in magnitude of the power density is initially large and only increases with a small magnitude. The large difference in the maximum and minimum power density, seen throughout the simulation, is a result of the two different enrichments used in the presented case (see Figure 3).

The computational effort and the time used by the different parts of the code for the  $7 \times 7$  case are presented in Table I. The calculations were performed on 16 CPUs \* and the total wall-clock time was 59 hours. As seen from the table, the thermal-hydraulic part of the solver accounts for 89% of the CPU-time, where most of the time is spent on the solution of the pressure equation in the moderator. For most of the time-steps, a single multiphysics iteration is required and for the majority ( $\approx 70\%$ ) of these multiphysics iterations less than 3 thermal-hydraulic iterations are required. Still, the first few iterations, where the reactivity and change in power density is large, a much higher number of thermal-hydraulic outer iterations are required. This suggests that the solver is less efficient with more severe transients. To further assess the computational performance of the different parts of the solver, more comparative studies are required. However, for the current setup, the thermal-hydraulics is shown to be the most expensive field. Still, the coupled fine-mesh simulations are shown to be computationally feasible for short transients at assembly level, even on as few as 16 CPUs.

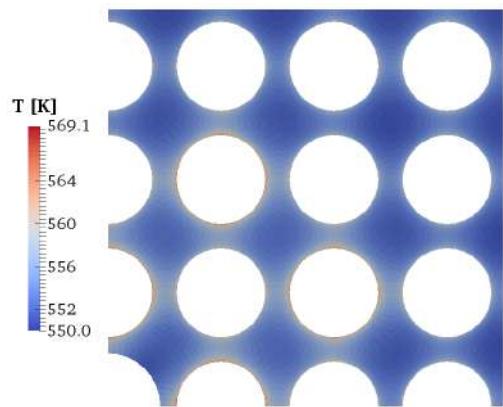
**Table I. Wall-Clock Time (WCT) for the  $7 \times 7$  parallelized on 16 CPUs.**

Field	WCT [s]	Relative WCT
Neutronics	23488 s	0.11
Thermal-hydraulics	189946 s	0.89
<b>Total</b>	<b>216599 s</b>	

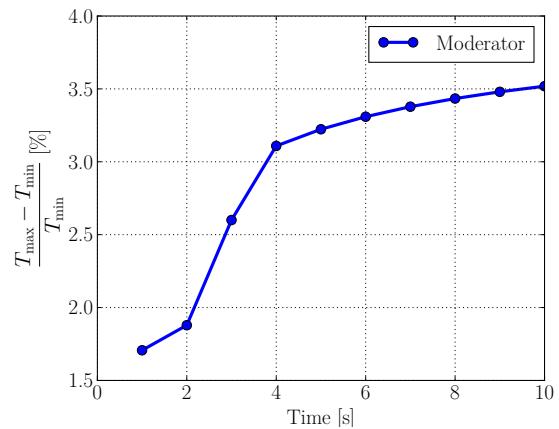
#### 4 CONCLUSIONS

In this paper, we have presented a framework for fine-mesh transient simulations for neutronics and thermal-hydraulics. The implemented solver includes single-phase coolant transport, conjugate heat transfer between the fuel pins and the coolant and a multigroup diffusion method to calculate the neutron density distribution in the system. An iterative scheme for the multiphysics couplings was outlined. The solver was applied to a small test system of a quarter of a fuel pin. A set of different boron concentrations were tested and resulted in cases of different initial reactivities which consequently lead to different power densities. The system was also tested for a time-dependent inlet temperature boundary condition, which was successfully resolved in all fields in space as well as time. Furthermore, we presented a system of a quarter of a  $7 \times 7$  lattice of fuel

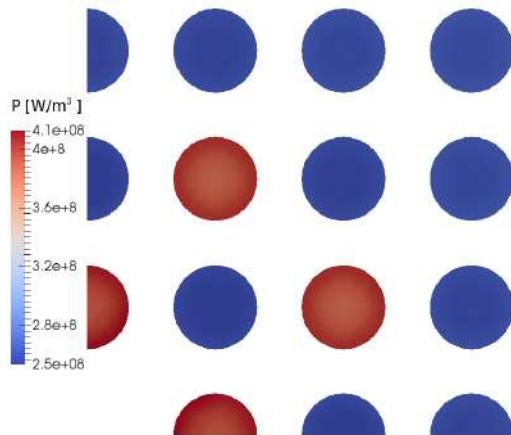
\* Nehalem CPUs (Intel® Xeon® E5520, 2.27GHz), 8 processors per computational node



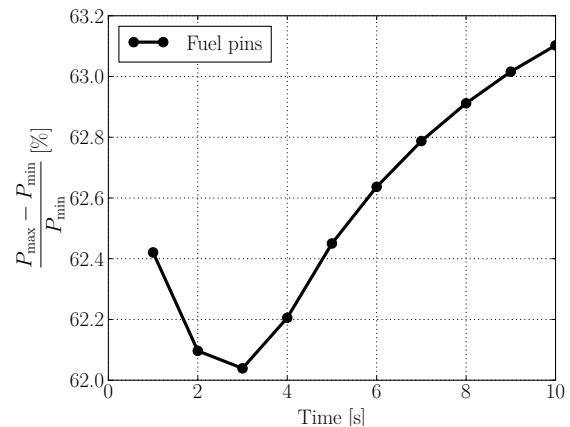
**(a) Moderator temperature at mid-elevation ( $t = 3$  s)**



**(b) Time-dependent heterogeneity of temperature in the moderator.**



**(c) Fuel power density at mid-elevation ( $t = 3$  s)**



**(d) Time-dependent difference between max and min power at mid-elevation.**

**Figure 8. Axial slices of moderator temperature and power density with corresponding time development of max and min values at mid-elevation.**

pins, including two different enrichments in U-235. The larger system was run with the same inlet temperature transient and the results were analyzed and presented in terms of the time development of the temperature in the domain as well as of the horizontal moderator and power density heterogeneity, as captured by the fine-mesh methodology. Whereas the currently presented cases are relatively homogeneous in terms of the thermal-hydraulic parameters, the spatial heterogeneities are expectedly larger for cases with more heterogeneous physics, like resolved two-phase flow, or heterogeneous geometries, e.g., from inclusion of resolved spacers. However, the presented results are examples of the current capabilities of the code and the future potential to achieve detailed spatial information for transient simulations within fuel bundles. The computational cost is assessed and the thermal-hydraulic CFD solver is shown to be almost an order of magnitude more expensive than the neutron diffusion solver for the presented  $7 \times 7$  case. Future extensions of the framework will be aimed at the evaluation of the multiphysics influence of increased turbulence from spacers and the influence from the insertion of control rods. Further verification and validation of the framework are also necessary, as well as the integration of the influence of the core global conditions on the simulated sub-system.

## ACKNOWLEDGMENTS

The project was carried out within DREAM (Deterministic REActor Modelling), which is a multi-disciplinary task force existing at Chalmers University of Technology aimed at developing next-generation high-fidelity modelling techniques for nuclear reactors in an integrated viewpoint, with focus on the multi-scale and multi-physics aspects of the systems. The Swedish Center for Nuclear Technology (SKC) is acknowledged for financially supporting K. Jareteg. The computations were performed on resources at Chalmers Centre for Computational Science and Engineering (C3SE) provided by the Swedish National Infrastructure for Computing (SNIC).

## 5 REFERENCES

- [1] C. Demazière, “Multi-physics modelling of nuclear reactors: current practices in a nutshell,” *International Journal of Nuclear Energy Science and Technology*, **7**, 4, pp. 288–318 (2013).
- [2] J. Bakosi, M. A. Christon, R. B. Lowrie, L. A. Pritchett-Sheats, and R. R. Nourgaliev, “Large-eddy simulations of turbulent flow for grid-to-rod fretting in nuclear reactors,” *Nuclear Engineering and Design*, **262**, pp. 544–561 (2013).
- [3] J. R. Lee, J. Kim, and C.-H. Song, “Synthesis of the turbulent mixing in a rod bundle with vaned spacer grids based on the OECD-KAERI CFD benchmark exercise,” *Nuclear Engineering and Design*, **279**, pp. 3 – 18 (2014), SI : CFD4NRS-4.
- [4] W. Jaeger, J. P. Manes, U. Imke, J. J. Escalante, and V. S. Espinoza, “Validation and comparison of two-phase flow modeling capabilities of CFD, sub channel and system codes by means of post-test calculations of BFBT transient tests,” *Nuclear Engineering and Design*, **263**, pp. 313 – 326 (2013).
- [5] B. Kochunas et al., “Coupled Full Core Neutron Transport/CFD Simulations of Pressurized Water Reactors,” *PHYSOR 2012, Knoxville, Tennessee, USA, April 15-20* (2012).

- [6] J. Hales et al., "Advanced multiphysics coupling for LWR fuel performance analysis," *Annals of Nuclear Energy*, *in press* (2014).
- [7] J. Yan et al., "Coupled Computational Fluid Dynamics and MOC Neutronic Simulations of Westinghouse PWR Fuel Assemblies with Grid Spacers," *NURETH-14, Toronto, Ontario, Canada, September 25-30* (2011).
- [8] J. W. T. Volkan Seker and T. J. Downar, "Reactor Physics Simulations with Coupled Monte Carlo Calculation and Computational Fluid Dynamics," *ICENES-2007* (2007).
- [9] K. Jareteg, P. Vinai, and C. Demazière, "Fine-mesh deterministic modeling of PWR fuel assemblies: Proof-of-principle of coupled neutronic/thermalhydraulic calculations," *Annals of Nuclear Energy*, **68**, pp. 247 – 256 (2014).
- [10] K. Jareteg, P. Vinai, S. Sasic, and C. Demazière, "Coupled fine-mesh neutronics and thermal-hydraulics - modeling and implementation for PWR fuel assemblies," *Submitted to Annals of Nuclear Energy, Special Issue: "Multi-Physics Modeling of LWR Static and Transient Behavior"* (2014).
- [11] K. Jareteg, P. Vinai, S. Sasic, and C. Demazière, "Influence of an  $S_N$  solver in a fine-mesh neutronics/thermal-hydraulics framework," *PHYSOR 2014, September 28 - October 3, Japan* (2014).
- [12] J. Leppänen, *Serpent - a Continuous-energy Monte Carlo Reactor Physics Burnup Calculation Code*, VTT Technical Research Centre of Finland, 2012.
- [13] H. K. Versteeg and W. Malalasekera, *An introduction to Computational Fluid Dynamics*, Prentice Hall, second edition edition, (2007).
- [14] OpenFOAM extensions, "foam-extend-3.0," 2014, <http://sourceforge.net/projects/openfoam-extend/>.

## **Paper VI**

*“A numerical framework for bubble transport in a sub-cooled fluid flow”*

K. Jareteg, S. Sasic, P. Vinai, C. Demaziére

*Journal of Computational Physics 345 (2017), pp. 373-403*





# A numerical framework for bubble transport in a subcooled fluid flow



Klas Jareteg <sup>a,\*</sup>, Srdjan Sasic <sup>b</sup>, Paolo Vinai <sup>a</sup>, Christophe Demazière <sup>a</sup>

<sup>a</sup> Division of Subatomic and Plasma Physics, Department of Physics, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden

<sup>b</sup> Division of Fluid Dynamics, Department of Applied Mechanics, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden

## ARTICLE INFO

### Article history:

Received 1 March 2016

Received in revised form 2 March 2017

Accepted 17 May 2017

Available online 22 May 2017

### Keywords:

Direct quadrature method of moments

Two-fluid

CFD

PBE

Condensation

## ABSTRACT

In this paper we present a framework for the simulation of dispersed bubbly two-phase flows, with the specific aim of describing vapor–liquid systems with condensation. We formulate and implement a framework that consists of a population balance equation (PBE) for the bubble size distribution and an Eulerian–Eulerian two-fluid solver. The PBE is discretized using the Direct Quadrature Method of Moments (DQMOM) in which we include the condensation of the bubbles as an internal phase space convection. We investigate the robustness of the DQMOM formulation and the numerical issues arising from the rapid shrinkage of the vapor bubbles. In contrast to a PBE method based on the multiple-size-group (MUSIG) method, the DQMOM formulation allows us to compute a distribution with dynamic bubble sizes. Such a property is advantageous to capture the wide range of bubble sizes associated with the condensation process. Furthermore, we compare the computational performance of the DQMOM-based framework with the MUSIG method. The results demonstrate that DQMOM is able to retrieve the bubble size distribution with a good numerical precision in only a small fraction of the computational time required by MUSIG. For the two-fluid solver, we examine the implementation of the mass, momentum and enthalpy conservation equations in relation to the coupling to the PBE. In particular, we propose a formulation of the pressure and liquid continuity equations, that was shown to correctly preserve mass when computing the vapor fraction with DQMOM. In addition, the conservation of enthalpy was also proven. Therefore a consistent overall framework that couples the PBE and two-fluid solvers is achieved.

© 2017 Elsevier Inc. All rights reserved.

## 1. Introduction

The physics of dispersed multiphase flows involves a wide variety of phenomena and exhibits a great complexity in terms of flow patterns, phase distributions and turbulent structures. For instance, in the case of gas–liquid systems, dispersed vapor bubbles interact with a continuous liquid phase, and can exchange momentum via forces due to drag, lift, turbulent motion, and other possible effects. Furthermore, if evaporation and condensation take place as during boiling water processes, the two phases can also exchange mass, with additional contributions to the mass, momentum and energy balance of the system. In these conditions, distribution, size, shape and velocity of the bubbles play a relevant role.

\* Corresponding author.

E-mail address: [klas.jareteg@chalmers.se](mailto:klas.jareteg@chalmers.se) (K. Jareteg).

## Nomenclature

### Latin letters

<i>a</i>	Source term for <i>w</i> equation
<i>b</i>	Source term for $\eta$ equation
<i>C</i>	Condensation rate
$d_{43}$	Mean diameter
<b>F</b>	Interfacial momentum force
<i>f</i>	Average number density
<i>h</i>	Enthalpy
$h_{if}$	Interfacial heat transfer
$H_{lg}$	Heat of evaporation
$\bar{\mathbf{I}}$	Unit tensor
<b>J</b>	Momentum transfer due to phase change
<i>k</i>	Turbulent kinetic energy
<b>M</b>	Interfacial momentum transfer
<i>m</i>	Bubble mass
<i>N</i>	Number of abscissas/classes
<i>Nu</i>	Nusselt number
<i>P</i>	Pressure
<i>Pr</i>	Prandtl number
<b>q</b>	Conduction heat flux
$\mathbf{q}^t$	Turbulent heat flux
<b>r</b>	Spatial coordinate
$Re_b$	Reynolds particle number
<i>S</i>	Source for PBE
<i>T</i>	Temperature
<i>t</i>	Time
<b>U</b>	Phase weighted velocity
$\mathbf{U}_p$	Phase velocity
$\mathbf{U}_r$	Relative velocity between phases
<i>w</i>	Weight of abscissa/class
<b>x</b>	Internal coordinates

### Greek letters

$\alpha$	Phase volume fraction
$\beta_{ij}$	Aggregation kernel
$\varepsilon$	Dissipation of turbulent kinetic energy
$\eta$	Abscissa times weight
$\Gamma$	Mass exchange source term
$\gamma$	Thermal diffusivity
$\Lambda$	Interfacial energy transfer
$\mu$	Viscosity
$\mu^t$	Turbulent viscosity
$\nu$	Kinematic viscosity
$\rho$	Density
$\bar{\tau}$	Viscous stress tensor
$\bar{\tau}^t$	Turbulent stress tensor
$\theta$	Breakage kernel
$\xi$	Bubble size
$\zeta$	Fragment distribution function

### Subscripts

agg	Term due to aggregation
bre	Term due to breakage
<i>C</i>	Term due to condensation
<i>g</i>	Gas phase
<i>i</i>	Abscissa number
<i>j</i>	Class number
<i>k</i>	Moment
<i>l</i>	Liquid phase
<i>p</i>	Current phase
<i>q</i>	Other phase
wall	Wall quantity

Boiling water flows are important for a wide range of industrial applications, such as nuclear reactors and steam generators. In fact they can be used for an efficient extraction and transport of energy, since the heat transfer can reach relatively high values in the sub-cooled boiling and saturated nucleate boiling regimes.

Yet, precise and accurate predictions of such systems still represent a major challenge. Although a Lagrangian approach would resolve in detail the interaction between the single bubbles and the liquid phase, it is not computationally feasible. As an alternative, a spatial averaging procedure can be applied, and the vapor and liquid phases are accordingly treated like interpenetrating continua within an Eulerian macroscopic arrangement [1]. As a result of averaging, the microscopic dynamics of the bubbles is lost, and the mass, momentum and energy exchanges between the separate bubbles and the surrounding liquid are not explicitly modeled. Whereas the inter-phase transfers occur on a local, microscopic level, the macroscopic approach requires a set of closure relations to describe the averaged behavior of the bubbles. Such closures need to be formulated from the available information, and can thus only be given through averaged parameters.

The resulting equations in Eulerian form can be solved at a reasonable computational cost, but have drawbacks. For example, Eulerian models for subcooled boiling have been developed (e.g., see [2]), but their accuracy was shown to be limited (e.g., see [3]). To overcome this kind of shortcomings, remedies were introduced in terms of additional equations to retrieve the distributions of some crucial microscopic properties of the bubbles, such as the size, velocity and material composition. The distribution of these parameters can be specified in terms of a probability density function (PDF), whose evolution can be described from a Population Balance Equation (PBE). The parameters of the PDF represent the internal phase space of the PBE, i.e. one or more of the aforementioned microscopic properties, while the position of the bubbles is associated to the physical external phase space. The PBE establishes a differential relationship for the internal and the external phase space [4].

In a great majority of cases, the shape of the PDF cannot be determined in a closed analytic form, especially not *a priori*. Instead, a discretized closure model is sought. One of these approaches is the method of classes (CM), used in the MUSIG model [5,6]. Accordingly, a single-parameter (mono-variate) PDF describes the distribution of bubble sizes with a fixed set of classes, and it is combined with closure models for bubble interactions. MUSIG has been applied to subcooled boiling flow, proving an increased accuracy of the vapor phase distribution as compared to pure Eulerian–Eulerian approaches [7].

A different way of discretizing the PDF is followed with the Method of Moments (MOM) [8], in which a closed set of moments of the PDF are calculated. Further development of MOM are the Quadrature Method of Moments (QMOM) [9] and the Direct Quadrature Method of Moments (DQMOM) [10]. In agreement with the latter two, the PDF is approximated by a set of quadrature points, whose weights and abscissas can be related to the moments of the PDF. A clear advantage with QMOM and DQMOM over CM is that, for the same accuracy, the computational burden is reduced, since a smaller set of moments is needed in comparison to the set of classes [11,12].

In the case of DQMOM the discrete moments are derived from transport equations describing the spatial and temporal evolution of the abscissas and weights of the quadrature. Such a method is well suited for Computational Fluid Dynamics (CFD) frameworks and is therefore an attractive formulation to determine bubble distribution parameters. DQMOM has previously been combined with two-fluid solvers and used for bubbly flow in different applications, including soot formation in combustion [13] and adiabatic bubbly flow [11]. For this purpose, DQMOM typically replaces the two-fluid continuity equation, so that the solution of the PBE is coupled to the momentum conservation equations.

One of the potential strengths of DQMOM is its applicability also for multivariate distributions [10,14]. However, in bubbly flow simulations, DQMOM has usually been applied to a mono-variate problem based on the bubble size distribution, and the velocity of the bubbles specified by a combined vapor momentum equation [15] or by a separate momentum equation for each abscissa [16]. For bi-variate problems the Conditional Quadrature Method of Moments (CQMOM) was also tested, for example, bi-variate PDFs were estimated with respect to bubble size and composition [17] and with respect to bubble size and velocity [18]. In contrast to DQMOM, CQMOM is not formulated in terms of transport equations, but relies on an inversion algorithm for the moments of the quadrature set. The CQMOM algorithm is less tractable for a direct incorporation into CFD software, though.

The aim of the current work is to investigate the capability of DQMOM for simulating bubbles in a subcooled liquid. This is particularly pertinent to the modeling of subcooled boiling. In the example of a heated tube, with an inlet forced flow of subcooled liquid that can undergo boiling, the vapor bubbles are first generated at the heated wall. The bubbles can then detach from the wall and move to the sub-cooled bulk. Finally, saturate nucleate boiling develops when the liquid becomes saturated. In the specific stage of bubbles that depart from the wall towards the bulk, different physical mechanisms come into play and must be considered, including aggregation, breakage, and reduction of bubbles due to condensation in the subcooled liquid.

As mentioned above, DQMOM was studied for some applications of bubbly flow, for which breakage, aggregation and evaporation of bubbles were treated. In this paper, we introduce an extended DQMOM-based framework that can deal with the condensation process. We derive a formulation of the condensation term as an internal phase space convection which directly corresponds to the shrinkage rate of the bubbles.

Despite the possible advantages of DQMOM, earlier research also pointed out that failure due to initial and boundary conditions and numerical difficulties such as abscissa crossing, might arise when solving the system of transport equations for the weights and abscissas [19]. We therefore investigate the robustness and the convergence of the proposed DQMOM algorithm. Special attention is given to the smallest size of the condensing bubbles for which the PDF may tend to collapse and lead to numerical problems. Also, we characterize discontinuities in the PDF that are due to the disappearance of bubbles in regions without any additional creation of vapor. Such conditions can be found where the subcooled bulk of flow contains a very low void, i.e. the vapor fraction is small.

Furthermore, the abscissas derived with DQMOM are compared to the classes applied in the MUSIG method. This aspect emphasizes a significant difference between the two methods: the abscissas from DQMOM can dynamically vary, while the classes in MUSIG are static. For the subcooled conditions that were considered in this work, a relatively wide range of bubble sizes can occur as the bubbles leave the wall and gradually condense in the subcooled bulk. Expectedly, it is impractical to recover such a span of bubble sizes from a distribution based on fixed classes. DQMOM has a major advantage for such conditions because the dynamic abscissas allow different size distributions to be resolved in different parts of the domain. In the paper we quantify this effect for different numbers of abscissas and classes.

Another specific aim in this paper is to compare the computational performance of the proposed DQMOM with the MUSIG method in the specific scenario under study. The evaluation is carried out in terms of computational time required to achieve the same precision for both methods. The outcome of this analysis suggests that, in terms of both computational cost and resolution of the bubble size distribution, DQMOM can be a more advantageous option for enhancing Eulerian models of bubbly flow in subcooled boiling conditions.

It should further be noted that the PBE equations were integrated with two-fluid two-phase conservation equations in Eulerian form. Whereas the PBE model describes the bubble size distribution, the mass, momentum and energy exchanges between the two phases still rely on an Eulerian approach. In the case of phase change, the conservation equations are explicitly coupled to the PBE via the mass exchange terms. In this work the contribution to this coupling comes from the shrinking bubbles (i.e., from the condensation). In particular, we propose a coupled formulation where the vapor continuity equation is replaced by the DQMOM equations. The latter are used to compute the vapor fraction and the phase change source terms which are important for the pressure-velocity coupling in the momentum equations. Furthermore, we specify an expression of the energy equations based on enthalpy, that is fully consistent with the momentum equations, and that was shown to be critical for the implementation.

The paper is structured as follows. The DQMOM and the Eulerian frameworks are described in Section 2. The solution algorithms are illustrated in Section 3. A set of numerical experiments are defined in Section 4 and the corresponding results are presented and discussed in Section 5. Finally, a summary and a conclusion are given in Section 6.

## 2. Model formulation

In the following section, we formulate a framework for two-phase flow that consists of bubbles dispersed in a subcooled liquid. We define a PBE for the bubbles that include a term for condensation as internal coordinate (Section 2.1) and derive a solution scheme based on DQMOM model (Section 2.2). In Section 2.3 we outline the MUSIG model that is used for the purpose of comparison. Then we provide the two-fluid Eulerian model that is coupled to DQMOM and MUSIG (Section 2.4). Finally, in Section 2.5 we describe how the bubbles are modeled at the wall.

### 2.1. PBE for bubbles in a subcooled liquid flow

In its most general form the PBE for a distribution of the average number density,  $f(\mathbf{x}, \mathbf{r}, t)$ , of bubbles can be written [4]:

$$\frac{\partial f(\mathbf{x}, \mathbf{r}, t)}{\partial t} + \nabla_{\mathbf{x}} \cdot \left( \frac{\partial \mathbf{x}(\mathbf{x}, \mathbf{r}, t)}{\partial t} f(\mathbf{x}, \mathbf{r}, t) \right) + \nabla_{\mathbf{r}} \cdot (\mathbf{U}(\mathbf{x}, \mathbf{r}, t) f(\mathbf{x}, \mathbf{r}, t)) = S(\mathbf{x}, \mathbf{r}, t), \quad (1)$$

where  $\mathbf{x}$  denotes a vector of internal coordinates,  $\mathbf{r}$  denotes the external coordinates,  $\mathbf{U}$  denotes the convection velocity in physical space and  $S$  denotes a source term. In the current paper the distribution depends on the bubble size (internal space), position (physical space) and time. A mono-variate PBE based on the bubble size  $\xi$  and with dependence on the common vapor velocity field  $\mathbf{U}$  can then be written:

$$\frac{\partial f(\xi, \mathbf{r}, t)}{\partial t} + \frac{\partial}{\partial \xi} \left( \frac{\partial \xi(\mathbf{r}, t)}{\partial t} f(\xi, \mathbf{r}, t) \right) + \nabla_{\mathbf{r}} \cdot (\mathbf{U}(\mathbf{r}, t) f(\xi, \mathbf{r}, t)) = S(\xi, \mathbf{r}, t). \quad (2)$$

It should be noted that there are two contributions to the convection; an internal convection corresponding to the rate of change in bubble size ( $\partial_t \xi(\mathbf{r}, t)$ ) and a physical space convection due to the velocity of the bubbles ( $\mathbf{U}$ ). This is primal for the formulation of the condensation, since the time-dependent variation of bubble size can be directly related to the condensation rate via the internal phase space convection:

$$\frac{\partial \xi(\mathbf{r}, t)}{\partial t} = C(\xi, \mathbf{r}, t). \quad (3)$$

Thus no additional contribution from the condensation to the source term is required. A similar approach is followed by Fox et al. [14], but for the treatment of droplet evaporation.

As regards the source term  $S$  in eq. (2), it consists of the contributions from aggregation and breakage of the bubbles. Therefore the closure of the term is given by an aggregation kernel and a breakage kernel, that depend on the size of the individual bubbles involved in the specific processes.

### 2.2. DQMOM model

DQMOM is based on a quadrature approximation of the bubble distribution function:

$$f(\xi; \mathbf{x}, t) \approx \sum_{i=1}^N w_i(\mathbf{x}, t) \delta(\xi - \xi_i(\mathbf{x}, t)), \quad (4)$$

where  $N$  denotes the number of abscissas ( $\xi_i$ ) and the associated weights ( $w_i$ ). According to the general procedure of DQMOM [10], the quadrature approximation is inserted in the PBE (eq. (2)), followed by a moment transform:

$$\mathcal{F}_k(\dots) = \int_0^\infty \xi^k \dots d\xi. \quad (5)$$

The procedure leads to a set of closed transport equations for the weights and abscissas:

$$\frac{\partial w_i}{\partial t} + \nabla \cdot (\mathbf{U}_g w_i) = a_i, \quad (6)$$

and

$$\frac{\partial w_i \xi_i}{\partial t} + \nabla \cdot (\mathbf{U}_g w_i \xi_i) = b_i, \quad (7)$$

in which  $a_i$  and  $b_i$  are source terms for the transport of  $w_i$  and  $w_i\xi_i$ , respectively; and  $\mathbf{U}_g$  is the vapor velocity. For equation (7) it is convenient to use a new variable defined as the product of weight and abscissa as:

$$\eta_i = w_i\xi_i, \quad (8)$$

The abscissas can then be obtained after solving eq. (7) as:

$$\xi_i = \frac{\eta_i}{w_i}. \quad (9)$$

To close the system we compute  $2N$  moments of the PBE, i.e.  $N$  for the weights and  $N$  for the abscissas. The source terms  $a_i$  and  $b_i$  for eqs. (6)–(7) are determined from the linear system of the  $N$  moment transforms of eq. (2), which takes the form:

$$\sum_i^N \left[ \xi_i^k a_i - k \xi_i^k a_i + k \xi_i^{k-1} b_i \right] = \mathcal{F}_k \left( S(\xi_i) - \nabla_\xi \cdot \left( \frac{\partial \xi(\mathbf{r}, t)}{\partial t} f(\xi, \mathbf{r}, t) \right) \right) \quad (10)$$

where  $k$  is the order of the moment. The left-hand side of eq. (10) is the result of the moment transform applied to the time-dependent and the physical space convection term as described in detail by [10]. Since the moment transform is linear, we may consider the transform of the internal convection, the source from aggregation and the source from breakage separately.

It should be noted that for high-order spatial schemes, the discretization of the convection terms in eqs. (6) and (7) lead to unrealizable moments [20] and, thus, unphysical distributions. It can be shown that only first-order finite volume schemes are guaranteed to be realizable [21]. As a remedy for the numerical diffusion associated with the low order scheme the fully-conservative DQMOM (FC-DQMOM), which treats also the convection terms as source terms, is adopted. This can be demonstrated to elude some of the effects of numerical diffusion on the weights and abscissas [17]. In this paper, we follow the approach in [22] and apply a first order convection scheme in physical space. Although the convection term may still induce a non-negligible numerical diffusion, the scheme proposed for the condensation term is consistent with FC-DQMOM and has the advantage of limiting such an issue.

### 2.2.1. Moment transform of the condensation term

From eqs. (5) and (10), the condensation term is written as a source term  $S_C$ :

$$S_{C,k}(\mathbf{r}, t) = - \int_0^\infty \xi^k \frac{\partial}{\partial \xi} (C(\xi, \mathbf{r}, t) f(\xi, \mathbf{r}, t)) d\xi \quad (11)$$

As proposed by Fox et al. [14], the derivative in  $\xi_i$  that appears in this kind of integrals can be eliminated by performing an integration by parts. From eq. (11) we then obtain:

$$S_{C,k} = \delta_{k0} C(0) f(0) + \int_0^\infty k \xi^{k-1} C(\xi) f(\xi) d\xi, \quad (12)$$

where  $\delta_{k0} C(0) f(0)$  corresponds to the internal phase space flux of particles of zero size and the explicit dependencies on space and time have been omitted for brevity. The Kronecker delta,  $\delta_{k0}$ , arises due to the regularity condition which implies that the number density must go to zero at infinite size [4]. Inserting the quadrature approximation given by eq. (4) yields:

$$S_{C,k} = \delta_{k0} C(0) f(0) + \int_0^\infty k \xi^{k-1} C(\xi) \sum_i^N w_i \delta(\xi - \xi_i) d\xi, \quad (13)$$

After the calculation of the integral, we retrieve

$$S_{C,k} = \delta_{k0} C(0) f(0) + \sum_i^N k w_i \xi_i^{k-1} C(\xi_i). \quad (14)$$

The second term of the right-hand side is closed by specifying a proper model for the condensation of the bubbles, while the first term is generally unknown. Fox et al. [14] propose a closure for the latter based on a set of ratio constraints between the weights of the quadrature and using one more moment.

The first term on the right-hand side is not zero for  $k = 0$ , and it represents the flux of condensing bubbles at zero size. In fact neither the number density  $f(0)$  nor the condensation rate  $C(0)$  can be assumed to be necessarily zero. The vapor bubbles dispersed in a subcooled liquid, shrink more and more because of the condensation, and eventually reach zero-size and disappear, thus giving a non-zero number density. As regards the condensation rate, it is typically estimated

from experimental correlations, and increases with smaller bubble sizes. In the current work, the contribution from such a term is not included. Instead, we propose that, when the simulated bubbles have reached a certain small size (or volume), they are automatically removed from the system, which corresponds to the assumption that an instantaneous and complete condensation occurs at that threshold. This procedure also has an advantage from another aspect. Since the rate of condensation quickly accelerates for smaller and smaller bubbles, prohibitively small time-steps may be needed in order to capture the behavior of the bubbles approaching zero-size. A threshold for the very small bubbles, allows to also avoid this kind of issues. More details on this point are discussed in Section 3.3.

It should be noted that for the outlined example of independent abscissas and the treatment of continuously shrinking bubbles, DQMOM has a major advantage over MUSIG. The latter method can not exactly represent the scenario described above, because the condensation will take a discontinuous form due to the use of explicit size classes.

### 2.2.2. Aggregation and breakage

The aggregation and breakage kernels required to close the system of equations (eq. (10)) are based on the derivations from [23]. The final, transformed, form of the source terms due to aggregation reads as:

$$S_{\text{agg},k}^B = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\xi_i^3 + \xi_j^3)^{k/3} \beta_{ij} w_i w_j \quad (15)$$

and

$$S_{\text{agg},k}^D = \sum_{i=1}^N \sum_{j=1}^N \xi_i^k \beta_{ij} w_i w_j \quad (16)$$

where  $B$  denotes the birth of bubbles,  $D$  the death of bubbles and  $\beta_{ij}$  the aggregation kernel. In the same manner, the contributions from the breakage read as:

$$S_{\text{bre},k}^B = \sum_{i=1}^N \zeta_i^k \theta_i w_i \quad (17)$$

and

$$S_{\text{bre},k}^D = \sum_{i=1}^N \zeta_i^k \theta_i w_i, \quad (18)$$

where  $\theta_i$  and  $\zeta_i^k$  correspond to the breakage kernel and the fragment distribution function, respectively. The models for the aggregation and breakage kernels, and for the fragment distribution function are given in Section 4.1.2.

### 2.3. MUSIG model

In order to investigate the DQMOM model, we compare it to another model derived from the MUSIG method. The MUSIG method is also based on a discretization procedure, but it makes use of fixed classes of bubble sizes. The MUSIG model with condensation applied in this work, is obtained from the one published by Lucas et al. [24]. Accordingly, eq. (1) is written in terms of bubble mass,  $m$ , instead of bubble size (compare eq. (2)), being:

$$\frac{\partial f(m, \mathbf{r}, t)}{\partial t} + \frac{\partial}{\partial m} \left( \frac{\partial m(\mathbf{r}, t)}{\partial t} f(m, \mathbf{r}, t) \right) + \nabla_{\mathbf{r}} \cdot (\mathbf{U}(\mathbf{r}, t) f(m, \mathbf{r}, t)) = S(m, \mathbf{r}, t). \quad (19)$$

The conservation equation for each bubble class  $j$  of mass  $m_j$  is achieved by integrating eq. (19) between the class lower and upper bounds denoted by  $m_{j-1/2}$  and  $m_{j+1/2}$  such that:

$$f_j(\mathbf{r}, t) = \int_{m_{j-1/2}}^{m_{j+1/2}} f(m, \mathbf{r}, t) dm. \quad (20)$$

We stress that the boundaries for each class  $m_{j-1/2}$  and  $m_{j+1/2}$  are fixed and a priori determined, which is further discussed in Section 2.3.1. In order to resemble a standard multi-fluid model, the number density  $f_j$  is written in terms of the corresponding bubble mass  $m_j$ , void fraction  $\alpha_{g,j}$  and gas density  $\rho_g$  as follows:

$$f_j = \frac{\alpha_{g,j} \rho_g}{m_j}. \quad (21)$$

Combining eqs. (19)–(21), we thus get:

$$\frac{\partial \alpha_{g,j} \rho_g}{\partial t} + \nabla \cdot (\alpha_{g,j} \rho_g \mathbf{U}_g) = S_{c,j} + S_{bre,j} + S_{agg,j}, \quad (22)$$

where the source terms  $S_{c,j}$ ,  $S_{bre,j}$  and  $S_{agg,j}$  include the contributions from condensation, breakage and aggregation, respectively.

It should be noted that Lucas et al. [24] apply multiple vapor momentum equations via a so-called inhomogeneous MUSIG model, whereas the DQMOM-based framework we developed consists of vapor continuity equations for the different bubble sizes together with a single momentum equation. The extension of both the DQMOM and MUSIG models to the inhomogeneous formulation is beyond the scope of the current article.

### 2.3.1. Condensation

The condensation term is discretized into fixed classes based on the procedure described above, i.e. by integrating the condensation term between the lower and upper bounds. Following such an integration, the source contribution due to condensation is given as the difference of bubbles introduced from the next class ( $j+1$ ), and the bubbles shrinking and leaving for a smaller size ( $j-1$ ) such that:

$$S_{c,j} = \frac{\Gamma_{g \rightarrow l,j} m_j}{m_j - m_{j-1}} - \frac{\Gamma_{g \rightarrow l,j+1} m_j}{m_{j+1} - m_j} \quad (23)$$

where  $\Gamma_{g \rightarrow l,j}$  is the mass exchange source term from gas to liquid due to condensation. The latter is computed using the condensation rate per size  $C(\xi_j)$ :

$$\Gamma_{g \rightarrow l,j} = -3 \frac{\alpha_{g,j} \rho_g C(\xi_j)}{\xi_j}, \quad (24)$$

where  $\xi_j$  is the size corresponding to  $m_j$  for the given density. The full derivation procedure is described in detail by Lucas et al. [24].

MUSIG inherently contains a drawback in the resolution of the PDF due to the fixed classes, which will later be shown to limit the accuracy for small number of classes, i.e. a coarse discretization of the continuous bubble size space.

Alternative formulations of the method of classes allow the class boundaries to be moved (see Kumar and Ramkrishna [25]). The benefit of these types of formulations is that a continuous size change in the bubble is more accurately captured, since discrete jumps in the phase space is not introduced in the PBE. However, such a procedure introduces additional difficulties since we need to dynamically modify the bin structure in order to accommodate new bubbles, with potential issues for the treatment of very small bubbles (similar to the discussion in Section 3.3). In addition, the spatial convection of the bubbles would require the development of a complex algorithm so that different class boundaries in different parts of the domain, can be taken in account. This would be of particular importance for the modeling of wall boiling phenomena. All together, dynamic sizes would risk the stability of the fixed sizes of the currently applied MUSIG method.

### 2.3.2. Aggregation and breakage

The breakage and aggregation sources are discretized for the  $N$  classes according to the formulation by Yeoh and Tu [26] where the birth and death contributions due to aggregation read:

$$S_{agg,j}^B = \frac{1}{2} \sum_k^N \sum_l^N \beta_{kl} \alpha_j \alpha_k \rho_g^2 \frac{(m_k + m_l)}{m_l m_k} \chi_{jkl} \quad (25)$$

and

$$S_{agg,j}^D = m_j \sum_k^N \beta_{ij} \frac{\alpha_j \alpha_k \rho_g^2}{m_k}. \quad (26)$$

The coalescence matrix  $\chi_{jkl}$  is defined as:

$$\chi_{jkl} = \begin{cases} 1 & \text{if } m_k + m_l = m_j \\ 0 & \text{else,} \end{cases} \quad (27)$$

which is applicable for the special case of  $m_j$  exactly equal to  $m_k + m_l$ . Such a condition is only fulfilled for cases where equal widths are used for the masses (i.e.  $m_j = j \Delta m + m_0$ ) as discussed by Kumar and Ramkrishna [27].

For general distributions of the masses (i.e. not necessarily uniformly discretized masses),  $\chi_{jkl}$  needs to be formulated in such a manner that mass (or volume) is conserved. A formulation that returns unity if the size  $m_j$  is close to the sum of the aggregate mass  $m_k + m_l$ , inexorably introduces an error when, e.g.,  $m_k + m_l$  lies between  $m_j$  and  $m_{j+1}$ . In the current work we apply a linear interpolation, weighting the contributions to the sizes  $j$  and  $j+1$  where  $m_j < m_k + m_l < m_{j+1}$ .

The source terms due to breakage is given by:

$$S_{bre,j}^B = \sum_k^N \zeta_k \theta_k \rho_g \alpha_k \quad (28)$$

and

$$S_{bre,j}^D = \theta_j \rho_g \alpha_j. \quad (29)$$

Again the fragment distribution function of the birth term eq. (28) needs to be formulated similar to  $\chi_{jkl}$  in order to conserve mass.

The same kernel models are applied for both DQMOM and MUSIG, later described in Section 4.1.2.

#### 2.4. Two-fluid model

The closure of the system of equations (6)–(7) for the DQMOM requires the vapor velocity ( $\mathbf{U}_g$ ) and the thermophysical state of the flow. In view of this, a two-fluid model based on the mass, momentum and energy conservation equations is introduced. The mass conservation equation for the phase  $p$  reads as [28]:

$$\frac{\partial \alpha_p \rho_p}{\partial t} + \nabla \cdot (\alpha_p \rho_p \mathbf{U}_p) = \Gamma_p. \quad (30)$$

$\alpha_p$  is volumetric fraction for phase  $p$ ,  $\mathbf{U}_p$  is the velocity,  $\rho_p$  is the density and  $\Gamma_p$  is the net mass transfer to this phase, i.e.:

$$\Gamma_p = -\Gamma_{p \rightarrow q} + \Gamma_{q \rightarrow p}. \quad (31)$$

In the above equation  $\Gamma_{p \rightarrow q}$  is the phase change from the phase  $p$  to phase  $q$ . When bubbles are dispersed within a subcooled liquid flow, there is no evaporation and only condensation, so that  $\Gamma_{l \rightarrow g}$  is zero. This condition is typical of the cases considered in the current work.

As a result of applying a PBE, the mass conservation equation (30) is not used for determining the phase fraction. However such an equation is still needed as constraint in solving the momentum and enthalpy equations.

The momentum conservation equation is given by:

$$\frac{\partial \alpha_p \rho_p \mathbf{U}_p}{\partial t} + \nabla \cdot (\alpha_p \rho_p \mathbf{U}_p \mathbf{U}_p) = -\nabla \cdot (\alpha_p (\bar{\tau}_p + \bar{\tau}_p^t)) - \nabla(\alpha_p P) + \alpha_p \rho_p \mathbf{g} + \mathbf{M}_p + \mathbf{J}_p \quad (32)$$

where  $\bar{\tau}_p$  is the viscous stress tensor,  $\bar{\tau}_p^t$  the turbulent stress tensor, and  $P$  the pressure shared between the phase.  $\mathbf{M}_p$  is the interfacial momentum transfer and  $\mathbf{J}_p$  the momentum transfer due to phase change. The latter is a combination of the loss of the phase  $p$  and the gain of the second phase  $j$ , i.e.:

$$\mathbf{J}_p = -\Gamma_{p \rightarrow q} \mathbf{U}_p + \Gamma_{q \rightarrow p} \mathbf{U}_p \quad (33)$$

Finally, the enthalpy equation can be written as:

$$\frac{\partial \alpha_p \rho_p h_p}{\partial t} + \nabla \cdot (\alpha_p \rho_p h_p \mathbf{U}_p) = -\nabla \cdot \alpha_p (\mathbf{q} + \mathbf{q}^t) + \frac{D(\alpha_p P)}{Dt} - \mathbf{U}_p \cdot \nabla \cdot (\alpha_p \bar{\tau}_p^t) + \alpha_p \bar{\tau}_p : \nabla \mathbf{U}_p + \Lambda_p \quad (34)$$

where  $h_p$  is the enthalpy,  $\mathbf{q}_p$  is the conduction heat flux,  $\mathbf{q}_p^t$  is the turbulent heat flux and  $\Lambda_p$  is the interfacial energy transfer defined as

$$\Lambda_p = -\Gamma_{p \rightarrow q} h_p + \Gamma_{q \rightarrow p} h_p \quad (35)$$

The dissipation source terms ( $\mathbf{U}_p \cdot \nabla \cdot (\alpha_p \bar{\tau}_p^t)$  and  $\alpha_p \bar{\tau}_p : \nabla \mathbf{U}_p$ ) are here neglected, as it can be shown that the size of these terms is much smaller than the typical heat flux and phase enthalpy exchange for heated flows.

In addition, we solve a two-equation turbulence model for the turbulent kinetic energy ( $k$ ) and the turbulent energy dissipation ( $\varepsilon$ ) in the liquid. The turbulence of the liquid phase will influence the velocity profiles and also the mixing conditions for the bubbles. In the presented framework the choice of the turbulence model is not of primary focus, however its impact will be investigated in future studies.

##### 2.4.1. Constituent relationships

A number of constituent relationships are necessary to close the conservation equations (30)–(34). We choose a set of closure laws that are commonly used in simulations of two-fluid two-phase flow.

First, the thermo-physical variables are read from lookup tables, including density ( $\rho_p$ ), viscosity ( $\mu_p$ ), and temperature ( $T_p$ ). By providing the current value of the enthalpy and the pressure, two tables are used for saturated conditions and sub-saturated conditions, providing the data for the vapor phase and the subcooled liquid phase, respectively.

For the stress tensor, a Newtonian fluid is assumed and the following relationship is valid:

$$\bar{\bar{\tau}}_p = -\mu_p \left( \nabla \mathbf{U}_p + (\nabla \mathbf{U}_p)^T - \frac{2}{3} \bar{\bar{\mathbf{I}}} \nabla \cdot \mathbf{U}_p \right). \quad (36)$$

For the liquid turbulent contribution ( $\bar{\bar{\tau}}_p^t$ ), the Boussinesq approximation in the form given by Ferziger and Peric [29] can be used:

$$\bar{\bar{\tau}}_p^t = \mu_p^t \left( \nabla \mathbf{U}_p + (\nabla \mathbf{U}_p)^T - \frac{2}{3} \bar{\bar{\mathbf{I}}} \nabla \cdot \mathbf{U}_p \right) \quad (37)$$

The term  $\mu^t$  for the liquid and vapor phases can be determined with the  $k - \varepsilon$  model as:

$$\mu_l^t = C_\mu \frac{k^2}{\varepsilon} \quad (38)$$

and

$$\mu_g^t = C^t \mu_l^t \quad (39)$$

where  $C^t$  is a model constant.

For the conduction and turbulent heat flux, Fourier's law of conduction in terms of enthalpy is used in combination with a constitutive model for the turbulent contribution such that

$$-\nabla \cdot \alpha_p (\mathbf{q} + \mathbf{q}^t) = -\nabla \cdot \alpha_p ((\gamma_p + \mu_p) h_p), \quad (40)$$

where  $\gamma_p$  is the thermal diffusivity for phase  $p$ .

The mass interfacial terms  $\Gamma_p$  are results of the DQMOM (or MUSIG) calculations and are further discussed in Section 3.6. The momentum interfacial transport terms are discussed in Section 4.1.3.

## 2.5. Treatment of the bubbles at the wall

In typical applications of subcooled boiling the vapor will appear due to phase change at the superheated walls of the system. The rate of vaporization is also influenced by the local conditions of the liquid phase adjacent to the wall. An overview of suitable models that can be employed to reproduce this phenomenon can be found in Yeoh and Tu [26].

In the current work, the process of vaporization of the liquid at the wall is disregarded. A priori bubble size distributions are instead assumed at the heated wall of the physical system under investigation. Thus the focus is entirely on the influence of the PBE on the prediction of the vapor distribution in the sub-cooled liquid flow.

Some studies from the open literature followed a so-called 'first cell layer' strategy. Accordingly, a source of vapor fraction is added in the computational nodes closest to the wall [30]. However, such an approach might cause unphysical values for conditions with high vapor content [31]. In the case of subcooled boiling, the technique can be expected reasonable since the vapor fraction is low.

Alternatively to the previous technique, the vapor generated at the wall could be mimicked by specifying a wall flux of bubbles. For such an approach one additional aspect must be clarified. In a CFD framework based on the Finite Volume Method (FVM), the flux of bubbles depends on the vapor velocity interpolated to the faces of the computational cells. The velocity of the component normal to the wall is however zero by definition. Thus, a flux of bubbles needs to be specified in terms of a velocity not acquired from the momentum equation. Furthermore it should not contribute to the momentum balance of the system. After the integration over the control volumes that is required in FVM, the effect of such a flux of bubbles is equivalent to a first cell source term. Despite this, the approach has benefits for the DQMOM scheme as explained below.

### 2.5.1. Wall conditions for the DQMOM-based model

For DQMOM the first cell layer approach is not directly achievable. Since the distribution of bubbles computed according to the wall model might be different from the first cell layer PDF, a mismatch can then occur. This issue is particularly evident when bubbles change in size during their transport through the system. The PDF of bubbles is solved for by making use of DQMOM and can not be enforced in the cell layer close to the wall. Thus, the mentioned technique would constrain the boundary distribution to be projected on to the first cell PDF.

An alternative is to fix the distribution of sizes  $\xi_i$  in the first cell layer. This approach was however found numerically unstable. Since  $\xi_i$  is computed based on the transport equations (6) and (7), it cannot be easily controlled. In addition to this, the distribution of abscissas would need to be reset after each time step, since the weights also require to be re-calculated. Such a solution scheme is nevertheless not viable because of the associated numerical problems that can arise.

To allow for an arbitrary bubble distribution at the wall, we specify all the quantities relevant to DQMOM (i.e.,  $\xi_i$ ,  $w_i$  and  $\eta_i$ ) as boundary conditions, in the form of fluxes at the wall. The DQMOM transport equations is then written as:

$$\frac{\partial w_i}{\partial t} + \nabla \cdot ((\mathbf{U}_g + \mathbf{U}_{\text{wall}}) w_i) = a_i \quad (41)$$

and

$$\frac{\partial \eta_i}{\partial t} + \nabla \cdot ((\mathbf{U}_g + \mathbf{U}_{\text{wall}}) \eta_i) = b_i, \quad (42)$$

where the wall flux only gives a contribution at the heated wall. The proposed formulation has the benefit of defining the desired boundary contribution and to let it interact with the PDF from the remaining system domain via eqs. (41) and (42). Thus, no posterior tweaking of  $\xi_i$  or  $w_i$  is required.

### 2.5.2. Wall conditions in the MUSIG-based model

In contrast to the dynamic abscissas in DQMOM, the static classes in MUSIG allow to introduce the vapor from a wall bubble size distribution in a more straightforward manner. In this case, the required wall distribution is projected on the static MUSIG classes, thus it results in a single wall flux for each class. As for DQMOM, the appropriate boundary value of the void fraction for each size should be chosen such that the flux at the boundary times the void fraction gives a source term equivalent to the desired vapor source at the wall.

We also compare the wall flux approach with a first cell layer one, where an additional volumetric source term  $S_b$  of vapor that can take in account the possible contribution from the boiling, is added as follows:

$$\frac{\partial \alpha_{g,j} \rho_{g,j}}{\partial t} + \nabla \cdot (\alpha_{g,j} \rho_g \mathbf{U}_g) = S_{c,j} + S_{\text{bre},j} + S_{\text{agg},j} + S_b, \quad (43)$$

with  $S_b$  that is non-zero only for the first cell layer close to the wall.

## 3. Solution algorithms

In this section we describe and analyze the solution algorithms applied for DQMOM (Section 3.1) and MUSIG (Section 3.2). We also discuss the issues specifically related to rapidly condensing bubbles and the potential remedies to such issues (Section 3.3). Furthermore, we comment on some of the specificities associated with the phase change due to condensation in the discretization of the momentum conservation equation (Section 3.4) and the enthalpy conservation equation (Section 3.5). Finally, we outline the coupling algorithm for DQMOM/MUSIG and the two-fluid solver (Section 3.6).

### 3.1. DQMOM solution algorithm

We first explain the iterative algorithm applied to the DQMOM equations, as presented in [Algorithm 1](#) (left). The main building blocks of the algorithm computes the source terms for eqs. (6) and (7), solves the same equations and finally determines the void fraction and the mass transfer source term from the computed weights ( $w_i$ ) and abscissas ( $\xi_i$ ).

The first part of the algorithm (lines 2–20) consists of a cell-wise computation of the contributions from the aggregation, breakage and condensation terms. In the same scope, the cell-wise square matrix system (eq. (10)) is inverted to compute the source terms for the transport equations of  $w_i$  and  $\eta_i$ , where  $\eta_i = w_i \xi_i$ . It should also be noted that in line 15, a check is performed on the sanity of the source terms  $a_i$  and  $b_i$  for regions of low void. This condition is discussed further in Section 3.3.

Following the cell-wise computations of  $a_i$  and  $b_i$ , the transport equations are closed and can then be solved to update  $w_i$  and  $\xi_i$ . To ensure that the solution to the coupled system of equations for  $w_i$  and  $\eta_i$  has converged, the maximum initial residual is computed and compared to a tolerance  $\varepsilon$  (line 1). Due to the coupling of the abscissas and weights in the calculations of the aggregation and breakage kernels and in the condensation source term, such an iterative procedure must be used.

From a computational point of view, the cell-wise calculation of the source terms  $a_i$  and  $b_i$  is the most expensive part of the algorithm. For a typical case (as presented in Section 5.3) with  $N = 2$ , lines 2–20 take approximately 75% of the CPU time, and, for  $N = 4$ , more than 90%. Still, it is inevitable to re-calculate the source terms after each solution of the transport equations (lines 21–24) since all models are affected by the updated abscissas and weights.

An alternative algorithm would be to re-calculate the cell-wise source terms inside the loop on lines 21–24. This choice gives a slightly more implicit dependence on the other  $i$ , because  $a_i$  and  $b_i$  are updated after each solution of the transport equations. On the other hand, the extra computational effort would be expected to be worth only if the system of equations (6) and (7) is strongly coupled. As will be shown in Section 5.3.3, this is not the case when the time step is governed by the Courant limit. Instead, we use the approach in [Algorithm 1](#) with the fewest possible number of updates of the cell-wise coefficients.

At the convergence of the DQMOM transport equations, we calculate the void fractions associated with each abscissa (line 28) and finally evaluate the source term for the phase change ( $\Gamma_{g \rightarrow l}$ ). To preserve mass conservation, the LHS of eq. (30) is explicitly calculated and used for  $\Gamma_{g \rightarrow l}$ . This has the benefit of ensuring that the total source term for the vapor fraction  $\alpha_g$  is consistent with the PBE method.

```

1 while res > ε do
2   //Compute source terms for transport equations
3   for c ∈ mesh cells do
4     for k = 1 to 2N do
5       Sa,k ← aggregation source from eqs. (15) and (16)
6       Sb,k ← breakage source from eqs. (17) and (18)
7       Sc,k ← condensation form eq. (14)
8       Stot,k = Sa,k + Sb,k + Sc,k
9     end
10    A: Square matrix size 2N × 2N
11    A ← Compute as LHS of eq. (10)
12    x ← Solution to Ax = Stot
13    for i = 1 to N do
14      ac,i = xc,i //Source term eq. (6)
15      bc,i = xc,i+N //Source term eq. (7)
16      //Ensure positive source terms for low
17      αc
18      if αi,c < εα then
19        ac,i = min(ac,i, 0)
20        bc,i = min(bc,i, 0)
21    end
22    //Solve transport equations for wi and ηi
23    for i = 1 to N do
24      wi ← from solution of eq. (6)
25      ηi ← from solution of eq. (7)
26    end
27    res ← maximum equation residual for wi and ηi ∀i
28  end
29  //Calculate void fraction and phase change
30  for i = 1 to N do
31    αi ← compute from current ξi and wi
32  end
33  Γg,l ← calculate from mass conservation eq. (30)
34
35  Γg,l ← update mass transfer term ∀j
36  while res > ε do
37    //Iterate over classes
38    for j ∈ N do
39      //Compute source terms
40      Sc,j ← condensation source term form (23)
41      Sa,j ← aggregation source from eqs. (15) and
42      (16)
43      Sb,j ← breakage source from eqs. (28) and (29)
44      //Solve transport equation for αg,j
45      αg,j ← from solution of eq. (22)
46
47      //Update maximum residual
48      if average(αg,j) > αthres then
49        res ← update maximum residual
50      end
51      //Optional under-relaxation
52      relax(αg,j)
53      //Ensure a positive αg,j
54      αg,j = max(αg,j, αMIN)
55    end
56    //Compute total gas fraction
57    α ← sum of αg,j ∀j
58  end
59  //Calculate phase change
60  Γg,l ← calculate from mass conservation eq. (30)

```

**Algorithm 1.** DQMOM (left) and MUSIG (right) solution algorithms.

### 3.2. MUSIG solution algorithm

For MUSIG we deploy a similar iterative approach as for DQMOM, presented in [Algorithm 1](#) (right). Again, the algorithm is such that the convergence of the coupled equations for the different classes is ensured within each time-step. Within each sub-iteration, i.e. starting with line 2, an iteration is performed over the  $N$  classes. For each class  $j$  and iteration, all the source terms are recomputed (lines 4–6). A more explicit, and potentially more stable, scheme would be to compute the source terms for all class  $j$  prior to the for loop (i.e. before line 3). However, by computing the source term according to the current void fractions, we take advantage of a faster propagation of the updated fields for each class. The chosen scheme requires a strict convergence criterion in order to ensure that the problem is fully solved before continuing to the next time step.

Following the update of the source terms, we compute  $\alpha_{g,j}$  according to the conservation equation. The maximum residual from all classes is continuously updated within the loop and is used as a criterion for the sub-iteration convergence. It should however be noted that the residual for a class  $j$  is accounted for only if there is a presence of bubbles. Such a threshold criterion is beneficial: in fact, if no bubble and source terms for class  $j$  are involved, the related conservation equation (22) will be trivial and the initial matrix of the residuals for that class will not contribute with any valuable information.

An under-relaxation of the current  $\alpha_{g,j}$  is optionally performed during the first one or two iterations. Such an under-relaxation might accelerate the convergence of the coupled system of transport equations. The final sub-iterations during each time step is not performed with under-relaxation, thus ensuring full convergence within each time step. Finally, after convergence of all the classes, the mass conservation equation is used to compute the source term for the liquid phase.

In terms of computational performance, the implementation of the source terms is the most crucial part. In particular, the computation of the aggregation source term from eq. (25), is expensive to compute as a double sum is required. Similar to the cost for DQMOM as described in Section 3.1, the cost of the double sum increases as  $N$  increases. For a direct implementation of eq. (25) as a double sum where each of the two terms are evaluated, the aggregation requires almost 98% of the computational time in the case of MUSIG-50. To optimize the algorithm, the number of operations for the aggregation term was minimized by moving the constant terms outside the sum and by first computing  $\chi_{jkl}$ . Therefore,

this version of the scheme led to a significant reduction of the computational burden needed for the evaluation of the aggregation term (which is now less than 80% of the total computational time for MUSIG-50).

The implementations of the MUSIG and DQMOM solvers are done in the same computational framework using the same type of data structures and implementation schemes. Furthermore, the conservation equations are solved based on the same discretization routines and sparse matrix solvers. We optimized both codes with respect to CPU-usage for the different functions. Nevertheless, a direct comparison between codes is often difficult due to the major influence of very specific details of the methods, as indicated for the case of the aggregation source term above, as well as architecture specific details which might benefit one of the two methods.

### 3.3. Treatment of small bubble sizes in DQMOM

As discussed in Section 2.2.1 the condensation process progresses with the volume reduction of the bubbles, which finally fully condense to liquid. Since DQMOM handles the size change of the bubbles in a continuous manner, the shrinkage of the bubbles can cause numerical problems, approaching small values. The mathematical relationship between the bubble size and the condensation rate is such that negative sizes could even be reached.

In particular, negative  $w_i$  and  $\eta_i$  could be obtained for critical cells, thus unphysical conditions could lead to the failure of either the DQMOM solver or the two-fluid solver. In theory, this problem could be avoided, if the time step is selected according to the largest condensation rate and the smallest bubble size. In practice, this option leads to prohibitively small time steps that make the calculations unfeasible.

Instead of directly limiting the time step we discuss two possible modifications of the method, which are to:

- Limit the shrinkage of the smallest bubbles, by modifying the condensation rate. Numerical experiments for this kind of remedy were found to be unstable and sensitive to the threshold size of bubbles applied, and still prone to negative local negative values of  $w_i$  or  $\eta_i$ .
- Filter the source terms  $a_i$  and  $b_i$  after the solution of the cell-wise linear system eq. (10). By guaranteeing positive values of  $a_i$  and  $b_i$  in all the computational domain, and given a bounded discretization scheme for the advection term, then  $w_i$  and  $\xi_i$  are always positive.

For low void fractions and small bubble sizes in a sub-cooled flow, the liquid phase is not significantly affected by the vapor phase, so it is possible to decouple the population balance and the vapor velocity. Based on this observation, it is reasonable to discard the contributions of the very low void fractions. Furthermore, numerical assessments showed that the second option is by far the most numerically robust approach. The filtering is then implemented in [Algorithm 1](#), lines 15–18.

It should be noted that the same kind of numerical issues will not appear for MUSIG. Due to the static sizes, extremely small bubble sizes are not directly resolved. On the other hand, this can not be accounted as an advantage of the MUSIG algorithm since the numerical issues are only avoided due to the limited resolution of the small bubbles.

### 3.4. Mass and momentum equations

For the two-fluid formulation our main concerns are to achieve: numerical stability in the bulk of the flow domain, where  $\alpha_g \rightarrow 0$  as discussed in Section 3.3; and a mass-conservative solver. For the first purpose, the momentum equation discretization is based on a phase-intensive formulation. The derivation follows the procedure proposed by [32], and the important steps are repeated here for completeness. In addition, we also include the formulation of the contribution from the phase change (i.e.  $\Gamma_p$ ) and the enthalpy equation.

The phase-intensive momentum equation is derived by subtracting  $\mathbf{U}_p$  times eq. (30) from eq. (32) and results in:

$$\alpha_p \rho_p \frac{\partial \mathbf{U}_p}{\partial t} + \alpha_p \rho_p \mathbf{U}_p \nabla \cdot \mathbf{U}_p = -\nabla \cdot (\alpha_p (\bar{\tau}_p + \bar{\tau}_p^t)) - \nabla(\alpha_p P) + \alpha_p \rho_p \mathbf{g} + \mathbf{M}_p + \Gamma_{q \rightarrow p}(\mathbf{U}_p - \mathbf{U}_p) \quad (44)$$

When the latter is divided with  $\alpha_p \rho_p$ , the left-hand side is independent of the phase fraction, and it is thus posed in a more stable form. For the right-hand side it is seen that, for regions where  $\alpha_p \rightarrow 0$ , a singularity occurs. Potential numerical problems can be avoided if each of the momentum and phase exchange terms are formulated as bounded values, as discussed, e.g., by Rusche [33].

In the next step, a combined, mass-conservative pressure equation is formulated. For this purpose we write a combined mass equation derived from the sum of eq. (30) for both phases such that

$$\nabla \cdot \mathbf{U} = \frac{\Gamma_g}{\rho_g} + \frac{\Gamma_l}{\rho_l} - \mathbf{U}_g \alpha_g \frac{\nabla \rho_g}{\rho_g} - \mathbf{U}_l \alpha_l \frac{\nabla \rho_l}{\rho_l} - \frac{\alpha_g}{\rho_g} \frac{\partial \rho_g}{\partial t} - \frac{\alpha_l}{\rho_l} \frac{\partial \rho_l}{\partial t}, \quad (45)$$

where

$$\mathbf{U} = \alpha_g \mathbf{U}_g + \alpha_l \mathbf{U}_l. \quad (46)$$

The pressure equation is then retrieved from the momentum equation (44) and inserted in the combined mass equation (45). By including the mass exchange terms and the compressibility due to density variation from heating, the pressure equation directly fulfills the mass conservation equation.

Under adiabatic conditions and for incompressible flow, the right-hand side of the mass conservation equation (45) is zero. However, this is not true for heated flow with phase change. The right-hand side of eq. (45) consists of the phase change (terms 1–2) and the variation of density for the heated flow (terms 3–6). In order to achieve a conservative solver for the conditions with change of phase, the correction terms are of major importance.

The momentum and pressure equations are discretized according to a standard collocated grid CFD approach (see, e.g., [29]). This means that the advective terms, such as the second term in eq. (44), are based on an interpolated face flux

$$\phi_p = \{\mathbf{U}_p\}, \quad (47)$$

where  $\{\dots\}$  denotes an interpolation from the cell centers to face centers. An accurate reconstruction of the face fluxes is a crucial aspect both for the accuracy and the mass conservation of the solver.

### 3.5. Enthalpy equation

In the same manner as for the momentum equation, the enthalpy equation is written in a phase-intensive formulation. For this purpose eq. (30) times the phase enthalpy is subtracted from eq. (34), for each phase, so that we obtain:

$$\alpha_p \rho_p \frac{\partial h_p}{\partial t} + \alpha_p \rho_p \mathbf{U}_p \cdot \nabla h_p = -\nabla \cdot (\alpha_p \gamma_p \nabla h_p) - \frac{D(\alpha_p P)}{Dt} + \Lambda_p - h_p \Gamma_p \quad (48)$$

From a discretization point of view, this formulation has the benefit that face fluxes needed for the second term on the left-hand side, are the same as the ones computed from the momentum equation.

In contrast, the first formulation of the enthalpy conservation equation (eq. 34) requires the interpolated facial flux of the product of  $\alpha_p \rho_p \mathbf{U}_p$ , i.e.  $\{\alpha_p \rho_p \mathbf{U}_p\}$ . However, we cannot generally assume that:

$$\{\alpha_p \rho_p \mathbf{U}_p\} = \{\rho_p\} \{\rho_p\} \{\mathbf{U}_p\} \quad (49)$$

and since  $\{\mathbf{U}_p\}$  is already used for the formulation of the momentum, pressure,  $w_i$  and  $\eta_i$  equations, it is also required to use the phase-intensive formulation of the enthalpy equation. In Section 5.4.3 the advantage of eq. (48) over the direct discretization of eq. (34) is clarified and proved in more details.

### 3.6. Combining the DQMOM model with the two-fluid model

The combined framework of the DQMOM (or MUSIG) and the two-fluid model couples via the vapor and liquid phase fractions  $\alpha_g$  and  $\alpha_l$  ( $= 1 - \alpha_g$ ), the vapor velocity  $\mathbf{U}_g$  and the variables used in the condensation model. As described above, DQMOM (or MUSIG) is used to calculate the vapor transport in the system, implicitly corresponding to the transport of  $\alpha_g$  in the case of eq. (30). The velocity is used in the transport equations for the weights and the weights times the abscissas in DQMOM, and for the void fraction for each class in MUSIG.

The overall algorithm for the coupled solver is presented in [Algorithm 2](#). The time step is optionally computed based on the Courant number of the largest of either the velocity field or the relative velocity. As discussed in Section 3.3, an additional limitation on the time step could be based on the rate of condensation. However, we do not include such a condition in the algorithm because of the filtering operation described in Section 3.3, that is used for preventing too short time steps related to small bubbles. Furthermore, in order to isolate the effects of different aspects of the algorithm, only a fixed time step is used in the results presented below.

Once the void fraction and size distribution have been calculated, we update the size-dependent interfacial forces. In the following steps (lines 6–12), a velocity predictor is computed and inserted into the combined continuity equation (45) to form the combined pressure equation. The pressure equation is optionally solved  $n_p$  times to account for non-orthogonality updates in the Laplacian operator [29]. The other iteration, started in line 6, assures that the physical pressure and velocity profiles are retrieved before re-computing the thermophysical data and the enthalpy equation for the liquid. This is especially important when using a pressure-dependent table for the thermo-physical properties: a non-converged pressure may cause a non-physical set of values, and endangers the convergence of the entire algorithm.

Finally, the enthalpy equation for the liquid is solved and the turbulence equations are evaluated to compute new turbulent quantities. In the current study, the vapor is assumed to be at saturated conditions, so the vapor enthalpy is not solved.

## 4. Case descriptions

To verify the functionality of the framework, we consider a set of test cases in a rectangular domain, with a vertical subcooled liquid flow in which vapor bubbles are transported.

In the first case, we apply constant and linear condensation rates (Section 4.1.1) and 3 aggregation kernels and 2 breakage kernels (Section 4.1.2), in a single-cell test case solving only the DQMOM/MUSIG models. Under such conditions, we

```

1  $\Delta t \leftarrow$  Fixed time step or Courant number limited
2 //Update of void fraction ( $\alpha_i$ ) and mass transfer ( $\Gamma_g$ )
3  $\alpha_i, \Gamma_{g \rightarrow l} \leftarrow$  DQMOM/MUSIG solver
4 //Compute interfacial force contributions based on  $\alpha_i$  and  $\xi_i$ 
5 for  $i = 1$  to  $N$  do
6   |  $\mathbf{F}_d, \mathbf{F}_l, \mathbf{F}_{vm}, \mathbf{F}_t \leftarrow$  Contributions from bubble size  $i$ 
7   | 5 end
8   | //Solve the pressure and velocity equation with  $n_c$  corrects
9   | for  $i = 1$  to  $n_c$  do
10    |   |  $U_g^{\text{pre}}, U_l^{\text{pre}} \leftarrow$  Velocity predictors based on eq. (44)
11    |   | //Use  $n_p$  pressure correctors
12    |   | for  $m = 1$  to  $n_p$  do
13    |   |   |  $p \leftarrow U_g^{\text{pre}}, U_l^{\text{pre}}$  inserted in eq. (45)
14    |   | end
15    |   |  $U_g, U_l \leftarrow$  Correction based on  $p$ 
16 end
17  $h_l \leftarrow$  Calculate liquid enthalpy equation (48)
18 Update thermo-physical state based on  $h_l$  and  $p$ 
19  $k, \varepsilon \leftarrow$  Turbulence model

```

**Algorithm 2.** Two-fluid solution algorithm and coupling to DQMOM.

only study the kinetics of the PBE methods, including neither the two-fluid coupling nor the spatial dependence of the abscissas/class transport. The simplified single-cell system is thus used to verify the correctness of the implementation of the DQMOM/MUSIG algorithms.

For the second case, we define a horizontally periodic domain and a fictitious condensation rate that is independent of the flow properties. Under such conditions, we are able to decouple the population balance method (DQMOM/MUSIG) from the two-fluid solver while still including the spatial dependence. Accordingly, we can compare the performance of the DQMOM solver with respect to the MUSIG formulation, without the influence of the two-fluid solver, but with the influence of a convective fluid velocity.

In the third case, the system is confined by vertical walls that induce velocity gradients in the liquid phase. We further apply a condensation rate-based on an empirical relation from the literature, which introduces a coupling between the flow conditions and the rate of condensation of the bubbles. Therefore, we are able to assess the implementation of the mass, momentum and enthalpy equations within the global algorithm, as well as the influence of the two-fluid solver on the bubble size distribution.

Finally, in the fourth case we impose a vapor distribution at the vertical walls in order to mimic a wall boiling condition, as described in Section 2.5. The purpose of it is to investigate further the enthalpy model.

Complementary to the framework, in Section 4.1 we discuss the closure relations used for condensation, the interfacial forces and the bubble aggregation and breakage kernels. The details of the geometries and the boundary conditions of the four cases are described in Sections 4.3–4.6.

#### 4.1. Closure relationships

To close the coupled framework of the population balance solvers and the two-fluid solver, constitutive relations are needed. These include momentum interfacial terms as well as condensation models. Furthermore, consistent models for the aggregation and breakage are required for DQMOM and MUSIG. The aim with the selection of models is to evaluate the framework as a whole and to compare the computational efficiency and ability to resolve the size distribution of bubbles of DQMOM against MUSIG. In view of this, the optimal choice of models and parameters along with the capability of reproducing experimental data, are not the primary concerns.

##### 4.1.1. Condensation

The condensation model relates the bubble shrinkage rate  $C(\xi)$  to the flow properties and the thermophysical state. We first formulate a model based on a constant or linearly varying condensation rate such that

$$C(\xi) = R_1 \xi + C_1 \quad (50)$$

where  $R_1$  is the rate of change of the shrinkage rate and  $C_1$  is a constant. A controlled condensation uncoupled from the thermophysical local properties and the flow conditions gives the opportunity to evaluate the performance of DQMOM in comparison to MUSIG, disregarding the influence of the two-fluid solver.

In Cases 2 and 3, we let the condensation rate depend on the local flow and the thermo-physical properties. From an energy balance between bubbles whose volume decreases in time, and the surrounding subcooled liquid, it is possible to estimate the condensation rate as:

$$C(\xi) = -\frac{2h_{if}}{\rho_g} \frac{(T_g - T_l)}{H_{lg}}, \quad (51)$$

where  $h_{if}$  is the interfacial heat transfer coefficient. This coefficient  $h_{if}$  can be related to the definition of Nusselt number (Nu):

$$h_{if} = \frac{Nu K_l}{\xi}. \quad (52)$$

Several empirical correlations are available in literature for determining Nu under condensation (e.g., see [34]). In the current work, the Ranz–Marshall correlation [35] is selected:

$$Nu = 2 + 0.6Re_b^{0.50}Pr^{0.33}, \quad (53)$$

where the Reynolds number is given as:

$$Re_b = \max\left(\frac{U_r \xi}{\nu_l}, 1.0\right). \quad (54)$$

It can be noted that the condensation rate is inversely proportional to the square root of the bubble size diameter. As a result of this, condensing bubbles will experience an increasingly faster shrinkage, which might cause the type of numerical issues discussed in Section 3.3.

Although the available relations for the condensation rate are generally valid only within certain ranges (e.g., see [34]), we apply the model regardless of the bubble sizes and the flow conditions.

#### 4.1.2. Aggregation and breakage

In the current work, a set of generic kernels are applied [36]. For the aggregation terms we apply a constant kernel

$$\beta_{ij} = \beta_0, \quad (55)$$

a sum kernel

$$\beta_{ij} = \beta_0 \frac{\xi_i^3 + \xi_j^3}{L^3}, \quad (56)$$

and a hydrodynamic kernel

$$\beta_{ij} = \beta_0 \frac{(\xi_i + \xi_j)^3}{L^3}, \quad (57)$$

where  $L$  is a normalization constant on the order of the initial mean particle size. For breakage we apply a constant kernel

$$\theta_i = \theta_0, \quad (58)$$

and a power law

$$\theta_i = \theta_0 \left(\frac{\xi_i}{L}\right)^y \quad (59)$$

where  $y$  is an exponent set to  $\frac{1}{2}$  for this investigation. For the latter two kernels, a symmetric breakage fragmentation function is applied (as derived for DQMOM in Marchisio et al. [23]).

In this context, other strategies can be used. For instance, kernels with different constants and physics can be implemented to accurately capture various bubble phenomena, e.g. by following the approach proposed by Luo and Svendsen [37]. However, difficulties can be encountered in a proper choice of the model parameters for capturing experimental data. As an alternative, kernels can be derived from fitting methods [38]. For the purpose of the current study, the selection of generic kernels can be considered satisfactory on covering the general dependence on particle size [36].

#### 4.1.3. Momentum interfacial terms

The interfacial forces are the links between the bubble size distribution and the two-fluid framework. Given the information of the distribution, the total force that acts on each bubble size and the corresponding vapor fraction, can be decomposed in contributions of different nature. Four interfacial terms are considered in this paper, namely the drag force, the lift force, the virtual mass force and the turbulent dispersion force. The specific choices are not separately studied for the specific application, but meant to be a representative set of models for the purpose of evaluating the proposed framework.

First, the drag force is based on the Schiller–Naumann correlation [39], and calculated as

$$\mathbf{F}_d = \frac{3C_d \rho_l \alpha_g \|\mathbf{U}_g - \mathbf{U}_l\|}{4\xi} (\mathbf{U}_g - \mathbf{U}_l), \quad (60)$$

where the drag coefficient  $C_d$  is computed as

$$C_d = \frac{24}{\text{Re}_b} \left( 1 + 0.15 \text{Re}_b^{0.687} \right), \quad (61)$$

with the bubble Reynolds number given by eq. (54). It should be noted that the Reynolds number is proportional to the bubble size and thus the drag coefficient is inversely proportional to the bubble size. Second, the lift force is calculated as

$$\mathbf{F}_l = -C_l \rho_l \alpha_g (\mathbf{U}_g - \mathbf{U}_l) \times \nabla \times \mathbf{U}_l. \quad (62)$$

The lift coefficient ( $C_l$ ) is in general a function of flow parameters and the particle size [40]. In our case we apply a constant lift force coefficient of  $C_l = -0.3$  which is reasonable for the range of bubble sizes of interest. Third, the virtual mass force is calculated as

$$\mathbf{F}_{vm} = -C_{vm} \alpha_g \rho_l \left( \frac{D\mathbf{U}_g}{Dt} - \frac{D\mathbf{U}_l}{Dt} \right), \quad (63)$$

with the virtual mass coefficient  $C_{vm} = 0.5$ . Fourth, the turbulent dispersion force is calculated by the Bertodano model [41]

$$\mathbf{F}_t = -\rho_l k_l \nabla \alpha_g. \quad (64)$$

The applied turbulence model is of major importance for the simulations. A wide range of alternatives can be found in literature. However, as mentioned in Section 2.4, a  $k - \varepsilon$  model is used, and the study of the impact of the different turbulence models is beyond the scope of the work.

#### 4.2. Initial and boundary size distributions

For the initial and boundary bubble size distributions, two alternatives are used. For Case 1 and 2, a mono-sized distribution is applied as a verification case for the condensation model implementation. For all other cases the PDF is computed based on a normal distribution, characterized by the mean size ( $\mu$ ) and the standard deviation ( $\sigma$ ). Given the latter two characteristics, the choice of quadrature points for DQMOM still contains a degree of freedom. Therefore, for the initial and boundary distributions, we use the Gauss–Hermite quadrature to compute the initial bubble sizes of the normal distribution.

For MUSIG a maximum bubble size is assumed. This size must be large enough to allow the bubbles growing due to aggregation, to be represented. In a general case this can be ensured by post-processing the size distribution and to verify that the number of bubbles in the largest bubble class is small enough. Given the upper boundary and the number of classes ( $N$ ), a uniform distribution of the bubble diameter is applied. Such a uniform (or close to uniform) distribution is typically seen in the literature (see, e.g., [42,38]). This partitioning does not give a uniform distribution in the bubble volumes and it also includes a discretization error in the represented distribution.

We compute the mean diameter

$$d_{43} = \frac{\sum_{i=1}^N \xi_i^4 w_i}{\sum_{i=1}^N \xi_i^3 w_i}, \quad (65)$$

as a single characteristic of the bubble size distribution. For the cases applying a normal size as initial and/or boundary conditions, the initial  $d_{43}$  is the same for all the cases independent of the number of abscissas or classes.

#### 4.3. Case 1: single cell case

In the first case we apply the DQMOM and MUSIG solvers to a single cell system, without coupling to the two-fluid solver and without any influence of convection in physical space. The kinetics of the methods are compared for different

**Table 1**

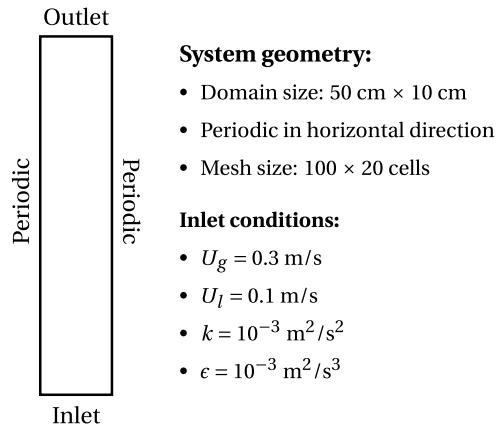
Kernels compared for the single-cell test case. For all cases a normal initial bubble distribution is applied with  $\mu = 7$  mm and  $\sigma = 2$  mm. For the aggregation cases the MUSIG sizes are discretized in the range 0–30 mm and for the breakage cases in the range 0–14 mm. For all kernels  $L = 10$  mm.

Model	Parameters
Constant aggregation	$\beta_0 = 10^{-5} \text{ s}^{-1}$
Sum aggregation	$\beta_0 = 10^{-5} \text{ s}^{-1}$
Hydrodynamic aggregation	$\beta_0 = 10^{-5} \text{ s}^{-1}$
Constant breakage	$\theta_0 = 10 \text{ s}^{-1}$
Power law breakage	$\theta_0 = 10 \text{ s}^{-1}$

**Table 2**

Condensation models applied for the single-cell test case with an initial mono size bubble distributions with  $\mu = 7$  mm and for the normal initial bubble distribution  $\mu = 7$  mm and  $\sigma = 2$  mm. The MUSIG sizes are discretized in the range 0–14 mm. For the exponential kernel  $L = 10$  mm.

Initial distribution	Model	Parameters
Mono	Constant cond.	$C_1 = -4.0 \text{ mm/s}, R_1 = 0.0 \text{ s}^{-1}$
	Linear cond.	$C_1 = -11.0 \text{ mm/s}, R_1 = 1.0 \text{ s}^{-1}$

**Fig. 1.** Geometry and boundary conditions for the rectangular, horizontally periodic system applied in Case 2.

kernels as summarized in Table 1 and for different condensation models as presented in Table 2. Therefore, we isolate the effect of aggregation, breakage and condensation, so that the performance of the two methods can be evaluated.

For the condensation cases with a mono size distribution, an analytical particle size can be computed and thus directly used to verify the implementation. In addition, we investigate the conservation properties of the kernels by analyzing the change in the vapor fraction.

#### 4.4. Case 2: comparison of DQMOM and MUSIG with convection

The purpose of this case is to investigate how the number of bubble sizes introduced in MUSIG corresponds to the number of abscissas in DQMOM under the influence of spatial convection.

In addition, the system is simulated both with and without aggregation and breakage, so that the interplay of the condensation term with the other source terms in the PBE can be investigated. In the current case we apply the constant kernels for both aggregation and breakage. To compare the different methodologies for the prediction of the bubble size distribution, the coupling to the two-fluid framework is limited by a horizontally periodic system, with a constant and a linearly varying condensation rate.

The simulated cases are presented in Table 3, from the simplest one with a mono-size distribution, without aggregation and breakage, to the more complex ones. We compare the solvers for the rectangular domain and the boundary conditions as presented in Fig. 1. For all the cases, liquid and vapor water are considered and all thermophysical data are temperature as well as pressure dependent [43]. As a result of the low void fractions considered in these test cases and of the relative small size of the simulated system, the influence of the thermophysical variables is limited.

**Table 3**

Case descriptions including inlet bubble size distributions and the applied source terms for Case 2.

Case	Inlet distribution	Condensation	Aggregation	Breakage
1a	Mono-size, 7.0 mm	Constant, $-2.0 \text{ mm/s}$	None	None
1b	Normal, $\mu = 10 \text{ mm}$ $\sigma = 2 \text{ mm}$	Constant, $-2.0 \text{ mm/s}$	None	None
1c	Normal, $\mu = 10 \text{ mm}$ $\sigma = 2 \text{ mm}$	Linear, $R_1 = 0.2 \text{ s}^{-1}$ , $C_1 = -3.0 \text{ mm/s}$	None	None
1d	Normal, $\mu = 10 \text{ mm}$ $\sigma = 2 \text{ mm}$	None	$\beta_0 = 5 \times 10^{-5} \text{ s}^{-1}$	None
1e	Normal, $\mu = 10 \text{ mm}$ $\sigma = 2 \text{ mm}$	None	None	$\theta_0 = 1 \times 10^{-2} \text{ s}^{-1}$
1f	Normal, $\mu = 10 \text{ mm}$ $\sigma = 2 \text{ mm}$	Linear, $R_1 = 0.2 \text{ s}^{-1}$ , $C_1 = -3.0 \text{ mm/s}$	$\beta_0 = 5 \times 10^{-5} \text{ s}^{-1}$	$\theta_0 = 1 \times 10^{-2} \text{ s}^{-1}$

#### 4.5. Case 3: coupling to two-fluid model

In the third case, the coupling to the two-fluid solver is introduced through a condensation model which is based on the flow properties as given in eq. (51). Furthermore, the system is confined by walls with no-slip conditions for the liquid phase and slip conditions for the vapor phase. Again we compare the efficiency of the PBE formulations in terms of the number of classes and abscissas.

The system is thus the same as in Fig. 1 except for applying vertical walls. The inlet boundary profiles for the phase velocities, the abscissas, and the weights are all assumed parabolic with the maximum values corresponding to the data in Fig. 1. For the inlet, we again specify a normal distribution of bubbles with  $\mu = 7 \text{ mm}$  and  $\sigma = 2 \text{ mm}$ . In this case, constant aggregation and breakage are always applied (with  $\beta_0 = 10^{-4}$  and  $\theta_0 = 10^{-4}$ ).

Complementary to the comparison of the bubble size distributions, we also show the influence of sub-iterations in Algorithm 1, which is an important evaluation of the convergence of the methods. Finally, we study the pure two-fluid aspects namely the mass conservation depending on the flux formulation for the pressure equation. The modified face flux used for the pressure equation in eq. (45) is compared to a solver without the terms due to phase change, i.e.:

$$\nabla \cdot \mathbf{U} = -\mathbf{U}_g \alpha_g \frac{\nabla \rho_g}{\rho_g} - \mathbf{U}_l \alpha_l \frac{\nabla \rho_l}{\rho_l} - \alpha_g \frac{\partial \rho_g}{\partial t} - \alpha_l \frac{\partial \rho_l}{\partial t}. \quad (66)$$

This allows to highlight the need for modifications of the standard two-fluid algorithm with respect to the mass exchange terms.

#### 4.6. Case 4: generation of bubbles at the wall

In the final case, we use the same geometry and conditions as in the previous case except for the void which is inserted at the wall, in order to reproduce the vapor generation that could occur under wall boiling conditions. Such a boundary condition allows us to demonstrate the capability to resolve the transport and the dynamics of bubbles from the wall to the bulk of the flow.

At the wall, we again assume a normal distribution of bubbles with  $\mu = 7 \text{ mm}$  and  $\sigma = 2 \text{ mm}$ . Due to the inflow at the inlet, we also need to specify a bubble size distribution and the associated void at the inlet. For this purpose we use the same parameters as the wall condition, but with a significantly lower total void fraction ( $10^{-5}$ ). The same conditions for aggregation and breakage (with  $\beta_0 = 10^{-4}$  and  $\theta_0 = 10^{-4}$ ) are applied in all the simulations.

As regards the enthalpy equation, we evaluate the importance of the flux formulation by comparing the original and the phase-intensive versions (see Section 3.5). Furthermore, we demonstrate potential numerical issues that can arise in DQMOM for colliding distributions, and lead to spatial discontinuities in the abscissas.

### 5. Numerical results and discussion

#### 5.1. Case 1

The results for condensation in a single-cell are presented in Table 4. As regards the mono-size distribution, an analytical solution for the final mean diameter  $d_{43}$  is computed by substituting the condensation rate as defined by eq. (50) into eq. (3), and integrating the resulting expression. Thus we obtain:

$$\xi(t) = \frac{1}{R} \left( (R\xi_0 + C_1)e^{Rt} - C_1 \right). \quad (67)$$

In the case of mono-dispersed bubbles of initial mean size  $d_{43} = 7 \text{ mm}$  and a simulation time of 0.5 s,  $\xi$  is calculated to be equal to 5 mm and 4.41 mm for the constant and linear condensation rates specified in Table 2, respectively. The DQMOM model predicts the analytical solution for all the number of abscissas considered. This is expected as the use of condensation only decouples the different abscissas; so a continuous decrease in size should capture the exact solution.

**Table 4**

Comparison of the mean bubble size ( $d_{43}$ ) and the void fraction ( $\alpha$ ) after 0.5 s simulation in a system corresponding to a single cell with an initial mono size distribution and with condensation models according to Table 2.

Model	Solver	$N$	$d_{43}$ [m]	$\alpha$ [-]
Constant	DQMOM	2	$5.00 \times 10^{-3}$	$1.82 \times 10^{-2}$
		3	$5.00 \times 10^{-3}$	$1.82 \times 10^{-2}$
		4	$5.00 \times 10^{-3}$	$1.90 \times 10^{-2}$
	MUSIG	10	$5.29 \times 10^{-3}$	$1.74 \times 10^{-2}$
		30	$5.11 \times 10^{-3}$	$1.79 \times 10^{-2}$
		50	$5.07 \times 10^{-3}$	$1.80 \times 10^{-2}$
		100	$5.03 \times 10^{-3}$	$1.81 \times 10^{-2}$
		150	$5.02 \times 10^{-3}$	$1.81 \times 10^{-2}$
Linear	DQMOM	2	$4.41 \times 10^{-3}$	$1.25 \times 10^{-2}$
		3	$4.41 \times 10^{-3}$	$1.25 \times 10^{-2}$
		4	$4.41 \times 10^{-3}$	$1.25 \times 10^{-2}$
	MUSIG	10	$5.28 \times 10^{-3}$	$1.19 \times 10^{-2}$
		30	$4.82 \times 10^{-3}$	$1.22 \times 10^{-2}$
		50	$4.67 \times 10^{-3}$	$1.23 \times 10^{-2}$
		100	$4.54 \times 10^{-3}$	$1.23 \times 10^{-2}$
		150	$4.49 \times 10^{-3}$	$1.24 \times 10^{-2}$

**Table 5**

Comparison of two breakage kernels in terms of the mean bubble size ( $d_{43}$ ), the void fraction ( $\alpha$ ) and the corresponding relative change in void fraction ( $\Delta\alpha$ ) after 0.5 s simulation in a system corresponding to a single computational cell.

Model	Solver	$N$	$d_{43}$ [m]	$\alpha$ [-]	$\Delta\alpha$ [-]
Constant	DQMOM	2	$2.12 \times 10^{-3}$	$4.93 \times 10^{-2}$	$-1.47 \times 10^{-2}$
		3	$2.48 \times 10^{-3}$	$4.93 \times 10^{-2}$	$-1.29 \times 10^{-2}$
		4	$2.48 \times 10^{-3}$	$4.93 \times 10^{-2}$	$-1.21 \times 10^{-2}$
		5	$2.47 \times 10^{-3}$	$4.92 \times 10^{-2}$	$-1.18 \times 10^{-2}$
		6	$2.46 \times 10^{-3}$	$4.89 \times 10^{-2}$	$-1.12 \times 10^{-2}$
	MUSIG	10	$3.52 \times 10^{-3}$	$1.38 \times 10^{-2}$	$-7.24 \times 10^{-1}$
		30	$2.75 \times 10^{-3}$	$4.16 \times 10^{-2}$	$-1.68 \times 10^{-1}$
		50	$2.58 \times 10^{-3}$	$4.73 \times 10^{-2}$	$-5.32 \times 10^{-2}$
		100	$2.51 \times 10^{-3}$	$4.96 \times 10^{-2}$	$-7.93 \times 10^{-3}$
		150	$2.50 \times 10^{-3}$	$4.99 \times 10^{-2}$	$-2.46 \times 10^{-3}$
Exponential	DQMOM	2	$4.36 \times 10^{-3}$	$4.99 \times 10^{-2}$	$-2.31 \times 10^{-3}$
		3	$4.36 \times 10^{-3}$	$4.99 \times 10^{-2}$	$-2.03 \times 10^{-3}$
		4	$4.35 \times 10^{-3}$	$4.99 \times 10^{-2}$	$-2.00 \times 10^{-3}$
		5	$4.34 \times 10^{-3}$	$4.99 \times 10^{-2}$	$-2.24 \times 10^{-3}$
		6	$4.34 \times 10^{-3}$	$4.98 \times 10^{-2}$	$-2.73 \times 10^{-3}$
	MUSIG	10	$4.31 \times 10^{-3}$	$3.95 \times 10^{-2}$	$-2.09 \times 10^{-1}$
		30	$4.32 \times 10^{-3}$	$4.99 \times 10^{-2}$	$-8.54 \times 10^{-4}$
		50	$4.33 \times 10^{-3}$	$5.00 \times 10^{-2}$	$-5.80 \times 10^{-5}$
		100	$4.33 \times 10^{-3}$	$5.00 \times 10^{-2}$	$-2.00 \times 10^{-6}$
		150	$4.33 \times 10^{-3}$	$5.00 \times 10^{-2}$	$4.00 \times 10^{-6}$

For MUSIG the results converge towards the correct solution and the discrepancies in the mean diameter for MUSIG-50 with constant condensation are less than 2%. The discrepancies in the final void fraction (last column in Table 4) are smaller than the ones in the mean diameter. Based on the convergence towards the analytical solution, the results for the mono size distribution verify that condensation is correctly implemented.

For the constant and power law breakage kernels, the comparison between MUSIG and DQMOM is presented in Table 5. The results show a good agreement in terms of the mean diameter between MUSIG and DQMOM. For the constant breakage model, DQMOM with 3 abscissas already provides values that are similar to the ones with 6 abscissas, indicating the strength of the dynamic abscissas. For MUSIG, 100 classes give a result close to the one using 150 classes. The difference between the number of abscissas/classes is smaller for the exponential kernel.

The comparisons between the different aggregation kernels are shown in Table 6. Again, there is a good agreement between MUSIG and DQMOM, with the same general trend of rapid convergence in the number of abscissas for DQMOM and a slower convergence for MUSIG.

The last two columns of Tables 5 and 6 show the void fraction and its relative change. These quantities are a direct measure of the conservativeness of the methods. For a non-conservative formulation of the aggregation and breakage kernels a significant change in  $\alpha$  would be seen. From both tables, the aggregation and breakage kernels give a very small

**Table 6**

Comparison of three aggregation kernels in terms of the mean bubble size ( $d_{43}$ ), the void fraction ( $\alpha$ ) and the corresponding relative change in void fraction ( $\Delta\alpha$ ) after 0.5 s simulation in a system corresponding to a single computational cell. For the relative change in void fraction ( $\Delta\alpha$ ) the value for MUSIG was smaller than  $10^{-6}$  which corresponds to the precision of the data stored.

Model	Solver	$N$	$d_{43}$ [m]	$\alpha$ [-]	$\Delta\alpha$ [-]
Constant	DQMOM	2	$9.00 \times 10^{-3}$	$5.00 \times 10^{-2}$	$-1.56 \times 10^{-4}$
		3	$9.06 \times 10^{-3}$	$5.00 \times 10^{-2}$	$-1.80 \times 10^{-4}$
		4	$9.08 \times 10^{-3}$	$5.00 \times 10^{-2}$	$-1.98 \times 10^{-4}$
		5	$9.08 \times 10^{-3}$	$5.00 \times 10^{-2}$	$-2.20 \times 10^{-4}$
		6	$9.08 \times 10^{-3}$	$5.00 \times 10^{-2}$	$-2.42 \times 10^{-4}$
	MUSIG	10	$8.94 \times 10^{-3}$	$5.00 \times 10^{-2}$	-
		30	$9.08 \times 10^{-3}$	$5.00 \times 10^{-2}$	-
		50	$9.09 \times 10^{-3}$	$5.00 \times 10^{-2}$	-
		100	$9.09 \times 10^{-3}$	$5.00 \times 10^{-2}$	-
		150	$9.10 \times 10^{-3}$	$5.00 \times 10^{-2}$	-
Hydrodynamic	DQMOM	2	$1.73 \times 10^{-2}$	$4.98 \times 10^{-2}$	$-3.13 \times 10^{-3}$
		3	$1.80 \times 10^{-2}$	$4.98 \times 10^{-2}$	$-3.52 \times 10^{-3}$
		4	$1.80 \times 10^{-2}$	$4.98 \times 10^{-2}$	$-3.96 \times 10^{-3}$
		5	$1.79 \times 10^{-2}$	$4.98 \times 10^{-2}$	$-4.42 \times 10^{-3}$
		6	$1.79 \times 10^{-2}$	$4.97 \times 10^{-2}$	$-4.84 \times 10^{-3}$
	MUSIG	10	$1.64 \times 10^{-2}$	$5.00 \times 10^{-2}$	-
		30	$1.74 \times 10^{-2}$	$5.00 \times 10^{-2}$	-
		50	$1.76 \times 10^{-2}$	$5.00 \times 10^{-2}$	-
		100	$1.76 \times 10^{-2}$	$5.00 \times 10^{-2}$	-
		150	$1.76 \times 10^{-2}$	$5.00 \times 10^{-2}$	-
Sum	DQMOM	2	$9.06 \times 10^{-3}$	$5.00 \times 10^{-2}$	$-2.52 \times 10^{-4}$
		3	$9.18 \times 10^{-3}$	$5.00 \times 10^{-2}$	$-3.40 \times 10^{-4}$
		4	$9.19 \times 10^{-3}$	$5.00 \times 10^{-2}$	$-4.30 \times 10^{-4}$
		5	$9.20 \times 10^{-3}$	$5.00 \times 10^{-2}$	$-5.40 \times 10^{-4}$
		6	$9.20 \times 10^{-3}$	$5.00 \times 10^{-2}$	$-6.92 \times 10^{-4}$
	MUSIG	10	$9.04 \times 10^{-3}$	$5.00 \times 10^{-2}$	-
		30	$9.20 \times 10^{-3}$	$5.00 \times 10^{-2}$	-
		50	$9.22 \times 10^{-3}$	$5.00 \times 10^{-2}$	-
		100	$9.23 \times 10^{-3}$	$5.00 \times 10^{-2}$	-
		150	$9.23 \times 10^{-3}$	$5.00 \times 10^{-2}$	-

relative change. In particular, the construction of the MUSIG algorithm as described in Section 2.3.2 leads to a conservative formulation despite the discrete nature of the method.

In general, Case 1 shows that the aggregation and breakage kernels yield comparable results for MUSIG and DQMOM and that the condensation models are correctly implemented as compared to the analytical value of the size change. As will be later seen, the discrepancy between MUSIG and DQMOM is for many cases larger.

## 5.2. Case 2

The comparison between the DQMOM and MUSIG approaches, when spatial convection is included, is summarized in Table 7 and in Fig. 2. The table includes the computational time, the change of the mean diameter, while the Figure shows the axial dependence of the mean diameter. In addition, the PDFs for the different number of abscissas and classes are presented in Fig. 3.

For the mono-size distribution and constant condensation rate in Case 2a, Table 7 and Fig. 2a report that DQMOM with three and four abscissas give the same  $\Delta d_{43}$ . Such a result is again anticipated because of the absence of interaction between the different sizes (i.e. no aggregation and breakage) and the exclusion of size-dependence in the applied condensation model. Nevertheless, this reveals an advantage of the dynamic abscissas in DQMOM as the constant condensation rate can be reproduced by the continuous decrease in the bubble sizes. In contrast, the MUSIG approach converges towards the  $\Delta d_{43}$  for DQMOM, but the use of 150 classes still gives a 3% discrepancy in the size change.

The inability of MUSIG to reproduce the continuous size change is also shown in Fig. 3a, where the distributions are presented in terms of the void (i.e.  $\alpha_g$ ) associated with each abscissa or class.

In the present case, it is expected that all the bubbles are identical at a certain axial position because they are only affected by condensation at a constant rate. Whereas the distributions from DQMOM are closely grouped around a single value, the MUSIG results provide a larger variety of bubble sizes due to a diffusivity in the internal phase space convection, caused by the static and discrete classes. In detail, any small concentration in bubbles will induce a non-zero condensation source term within every time step; thus a slightly diffused distribution is obtained as bubbles are condensed towards

**Table 7**

Relative computational time (**WCT [au]**) and absolute change in the mean diameter ( $\Delta d_{43}$ ) between inlet and outlet along the center line for Cases 2a–2f. The computational time is similar for all cases whereas the change in  $d_{43}$  between cases a–f is a result of the applied coefficients as listed in [Table 3](#).

(a)				(b)			
Solver	<i>N</i>	WCT [-]	$\Delta d_{43}$ [m]	Solver	<i>N</i>	WCT [-]	$\Delta d_{43}$ [m]
DQMOM	2	1.0	−3.71e−04	DQMOM	2	1.0	−3.05e−04
	3	1.1	−3.70e−04		3	1.2	−2.95e−04
	4	1.3	−3.70e−04		4	1.4	−2.90e−04
	10	1.0	−1.74e−04		5	1.8	−2.88e−04
	30	1.6	−3.00e−04		MUSIG	10	1.1
	50	2.2	−3.29e−04		30	1.6	−1.49e−04
MUSIG	100	4.4	−3.53e−04		50	2.2	−2.38e−04
	150	7.8	−3.59e−04		100	4.4	−2.56e−04
					150	7.8	−2.71e−04
							−2.75e−04
(c)				(d)			
Solver	<i>N</i>	WCT [-]	$\Delta d_{43}$ [m]	Solver	<i>N</i>	WCT [-]	$\Delta d_{43}$ [m]
DQMOM	2	1.0	−1.33e−04	DQMOM	2	1.0	8.50e−04
	3	1.1	−1.17e−04		3	1.2	8.68e−04
	4	1.4	−1.09e−04		4	1.4	8.73e−04
	5	1.7	−1.05e−04		5	1.7	8.77e−04
	10	1.1	−1.77e−05		MUSIG	10	1.1
MUSIG	30	1.6	−6.90e−05		30	1.6	9.57e−04
	50	2.2	−8.03e−05		50	2.3	9.78e−04
	100	4.3	−8.78e−05		100	4.6	9.80e−04
	150	7.6	−9.09e−05		150	8.0	9.81e−04
(e)				(f)			
Solver	<i>N</i>	WCT [-]	$\Delta d_{43}$ [m]	Solver	<i>N</i>	WCT [-]	$\Delta d_{43}$ [m]
DQMOM	2	1.0	−3.50e−06	DQMOM	2	1.0	7.40e−04
	3	1.2	−3.36e−06		3	1.2	7.72e−04
	4	1.4	−3.43e−06		4	1.4	7.82e−04
	5	1.8	−3.73e−06		5	1.7	7.87e−04
	10	1.1	−3.30e−06		MUSIG	10	1.1
MUSIG	30	1.7	−3.35e−06		30	1.6	9.36e−04
	50	2.3	−3.43e−06		50	5.2	9.18e−04
	100	4.5	−3.41e−06		100	13.6	9.12e−04
	150	7.9	−3.35e−06		150	28.1	9.06e−04
							9.04e−04

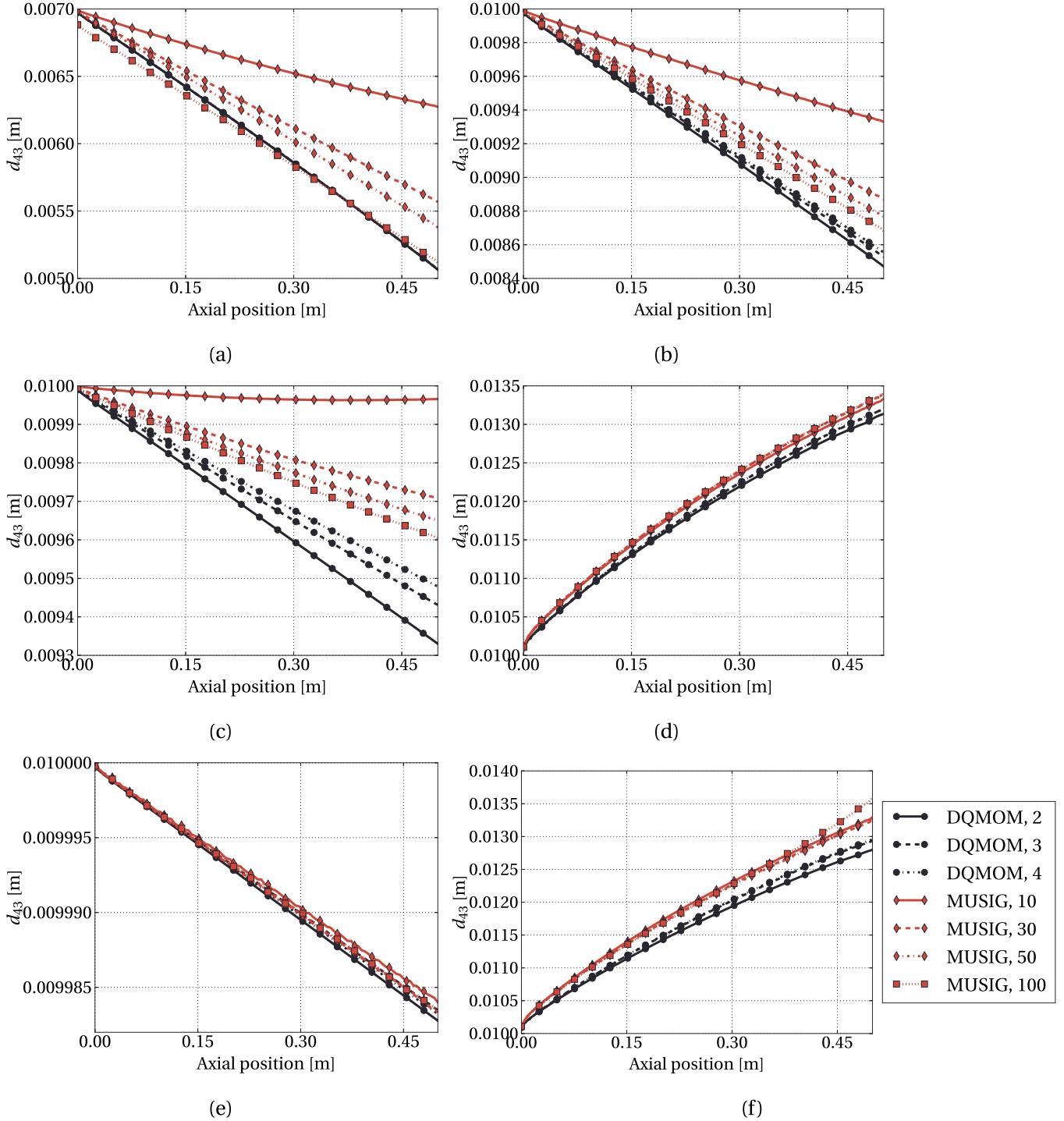
smaller sizes. However, for a larger number of classes the effect is less pronounced, where the using of 100 classes results in the most narrow distribution. In order to avoid the collapse of the DQMOM algorithm, a minor perturbation of the sizes was introduced, as visible through the small spread in the size distributions.

In Case 2b, the constant condensation model is applied for normally distributed bubble sizes. In contrast to Case 2a, there is now an expected difference between the DQMOM simulations as a higher number of abscissas better resolve the distribution in sizes. However, as seen from Figs. 2b and 3b, the result of four abscissas is already well-captured with three abscissas. For MUSIG the same trend as in Case 2a is again seen, with an underestimation of the shrinkage  $\Delta d_{43}$  for the cases with few classes as seen in [Table 7](#).

For Case 2c, we introduce a condensation rate that varies linearly with bubble size. A similar trend as for Case 2a and 2b is again seen. DQMOM-3 closely captures the size distribution of the bubbles as produced for DQMOM-4, which can be interpreted as the method is close to convergence. From [Table 7](#) the relative computational time for MUSIG with 150 classes is 5 times larger than that for DQMOM-4. This clearly shows the efficiency of DQMOM in comparison to MUSIG, for a similar precision.

For Case 2c, [Table 7](#) displays that MUSIG-100 is close to the results for MUSIG-150. It is however seen that DQMOM and MUSIG seems to converge to different values for the change in the mean diameter. The discrepancies between the MUSIG and the DQMOM simulations are due to: i) the inability of MUSIG to represent the continuous character of condensation; and ii) the discretization error related to the static approach for describing the bubble sizes. This effect is even more evident for Case 3, where the condensation changes more rapidly (see discussion in Section 5.3.1).

The separate effect of aggregation (Figs. 2d and 3d) indicates that MUSIG can capture the phenomenon in a relatively better way. As regards the simulation of Case 2e with only breakage (Fig. 2e and 3e), DQMOM again gives comparable

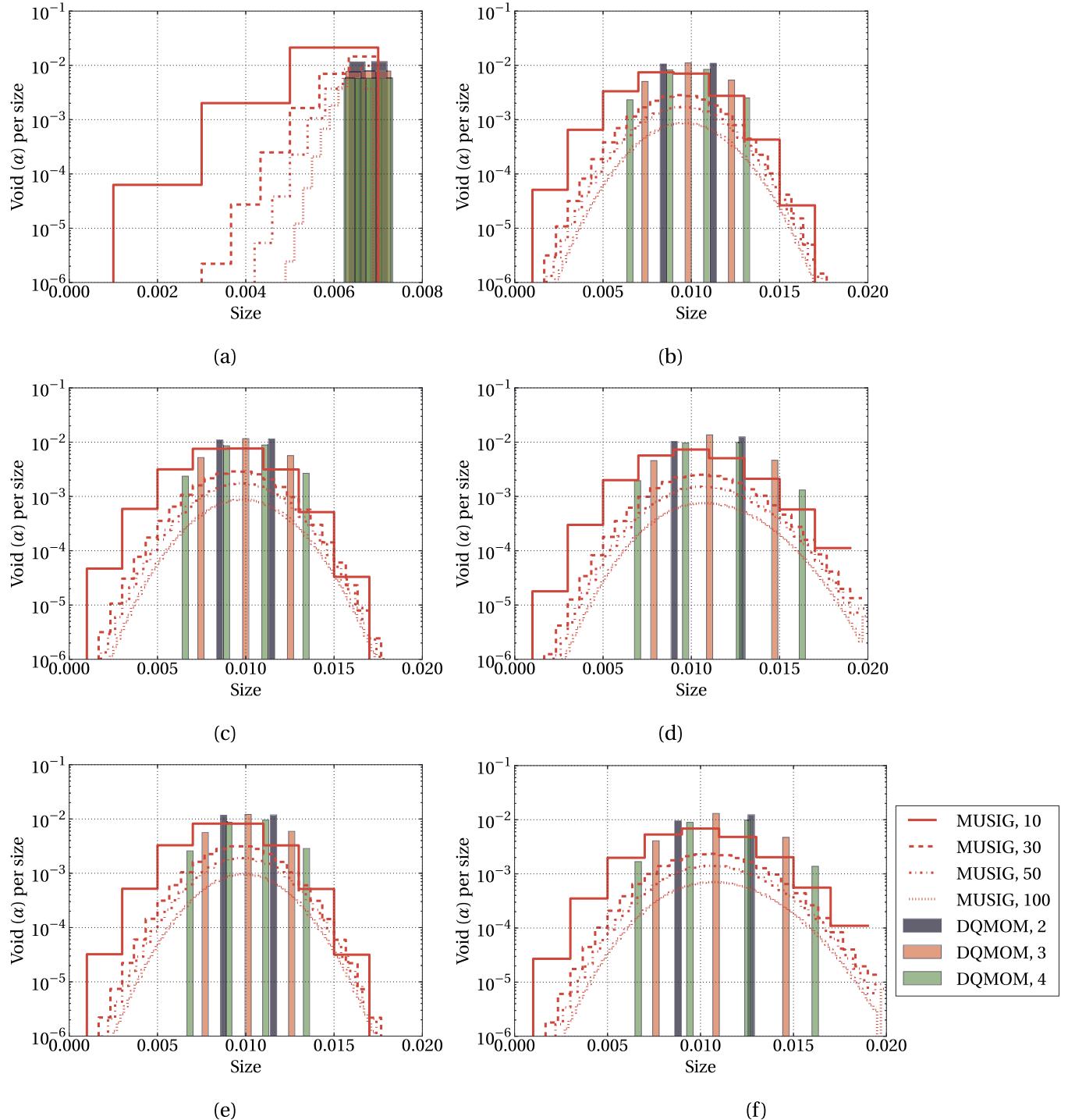


**Fig. 2.** Axial dependence of mean diameter ( $d_{43}$ ) for Cases 2a–2f sampled at the center line of the system at  $t = 5$  s.

outcomes with 2 abscissas and with 4 abscissas. Similarly, MUSIG gives convergence with few classes. Nevertheless, the effect of breakage is relatively small because of the current choice of the breakage kernel and the associated constant.

In the last Case 2f, we include aggregation, breakage and condensation at the same time. The results (Figs. 2f and 3f) are similar to Case 2d, since the contribution from the aggregation kernel is dominant. In fact, the change of mean diameter for Case 2c (linear condensation rate) is about one order smaller than the one for Case 2d (aggregation), as reported in Table 7.

Overall, the analysis of Case 2 demonstrates that the DQMOM and MUSIG methodologies for the condensation problem give close results. Furthermore, the continuous approach of DQMOM allows to capture the dynamics of condensation with few abscissas, while MUSIG can require a relatively large number of static classes for accurate simulations, with a higher computational cost. Also, consistent results for the breakage and aggregation kernels are retrieved from both models.



**Fig. 3.** Bubble size distributions in terms of the vapor fractions displayed for Cases 2a–2f at  $t = 5$  s and axial position 0.1 m at the center line of the system. Cases 2a–2c correspond to condensation only and the PDF are shifted towards smaller bubbles. For Case 2d, based on aggregation only, a shift towards larger bubbles is seen.

### 5.3. Case 3

In the presented Case 2 the influence from the two-fluid solver is minimized and the coupled effects are thus less pronounced. Following the pure verification of the handling of the condensation in DQMOM, we now demonstrate and evaluate the influence of the coupling to the remaining of the two-fluid framework. In section 5.3.1 we again illustrate the influence of the number of abscissas/classes but now for the confined, shear-dominated flow with an empirical model for the condensation rate. In Section 5.3.2 we demonstrate the importance of the flux derivation for the conservation of mass in the two-fluid framework. Finally, in Section 5.3.3 we evaluate the performance of the framework in terms of choice of time step and iterative schemes (as discussed in Section 3.1 and 3.2).

**Table 8**

Comparison of change of the mean diameter ( $\Delta - d_{43}$ ) and the void fraction ( $\Delta\alpha_g$ ) from the inlet to  $z = 0.1$  m along the center line of the system. **WCT** corresponds to the relative computational time for each case as normalized against DQMOM-2.

Solver	$N$	WCT [-]	$\Delta\alpha_s$ [-]	$\Delta d_{43}$ [m]
alphaEqn	1	0.6	$-3.55 \times 10^{-2}$	0.0
DQMOM	2	1.0	$-3.03 \times 10^{-2}$	$-3.85 \times 10^{-4}$
	3	1.5	$-3.01 \times 10^{-2}$	$1.09 \times 10^{-4}$
	4	2.4	$-3.01 \times 10^{-2}$	$1.88 \times 10^{-4}$
	5	3.6	$-3.00 \times 10^{-2}$	$1.35 \times 10^{-4}$
	6	5.7	$-3.00 \times 10^{-2}$	$1.84 \times 10^{-4}$
MUSIG	10	0.9	$-3.41 \times 10^{-2}$	$1.42 \times 10^{-3}$
	20	1.3	$-3.44 \times 10^{-2}$	$1.04 \times 10^{-3}$
	30	1.7	$-3.45 \times 10^{-2}$	$8.79 \times 10^{-4}$
	40	2.2	$-3.45 \times 10^{-2}$	$7.93 \times 10^{-4}$
	50	2.8	$-3.46 \times 10^{-2}$	$7.39 \times 10^{-4}$
	100	7.0	$-3.47 \times 10^{-2}$	$6.26 \times 10^{-4}$

### 5.3.1. DQMOM and MUSIG coupled to the two-fluid solver

We present in [Table 8](#) the relative computational time, the change in the mean diameter, and the void fraction, over the first 0.1 m along the symmetry line. Accordingly, the change of  $d_{43}$  as calculated in MUSIG is initially larger than DQMOM. When the entire length is analyzed, the average bubble size and void fraction however decrease much faster for DQMOM (see [Fig. 4](#)). The faster decrease in size is due to the fact that the condensation model is inversely proportional to the bubble size: the smaller the bubbles, the quicker the condensation. This is also shown by the acceleration of the shrinkage in [Fig. 4b](#).

Furthermore, the bubble size distributions for MUSIG with 10, 50 and 100 classes and DQMOM with 2 and 4 abscissas are displayed in [Fig. 5](#). The different dynamics of the two methods is well illustrated considering the axial development of the bubble size distribution. Close to the inlet ( $z = 0.025$  m, [Fig. 5a](#)) the distributions are similar for DQMOM and MUSIG and resemble the initial normal distribution in sizes. However, the maximum sizes of the DQMOM cases are significantly smaller than the ones estimated with MUSIG. Due to the size-dependent condensation this has a direct impact on the total condensation and, correspondingly, the size change of the bubbles. Along the next axial levels,  $z = 0.05$  m to  $z = 0.200$  m ([Figs. 5b to 5d](#)), the differences of the bubbles associated to the maximum void become larger and larger. Such discrepancies are also reflected on the calculated void fraction, as can be seen in [Table 8](#).

In [Fig. 4b](#), it is also notable how the change in mean diameter is dampened between  $z = 0.3$  m and  $z = 0.4$  m. This is caused by the modeling artifact introduced to deal with small bubbles and discussed in [Section 3.3](#). In fact the source terms  $a$  and  $b$  in the DQMOM framework are filtered for cells with vapor fraction below a threshold limit. The void fraction at  $z = 0.3$  m ([Fig. 4a](#)) is already so small that the influence of the vapor phase on the continuous liquid phase is negligible.

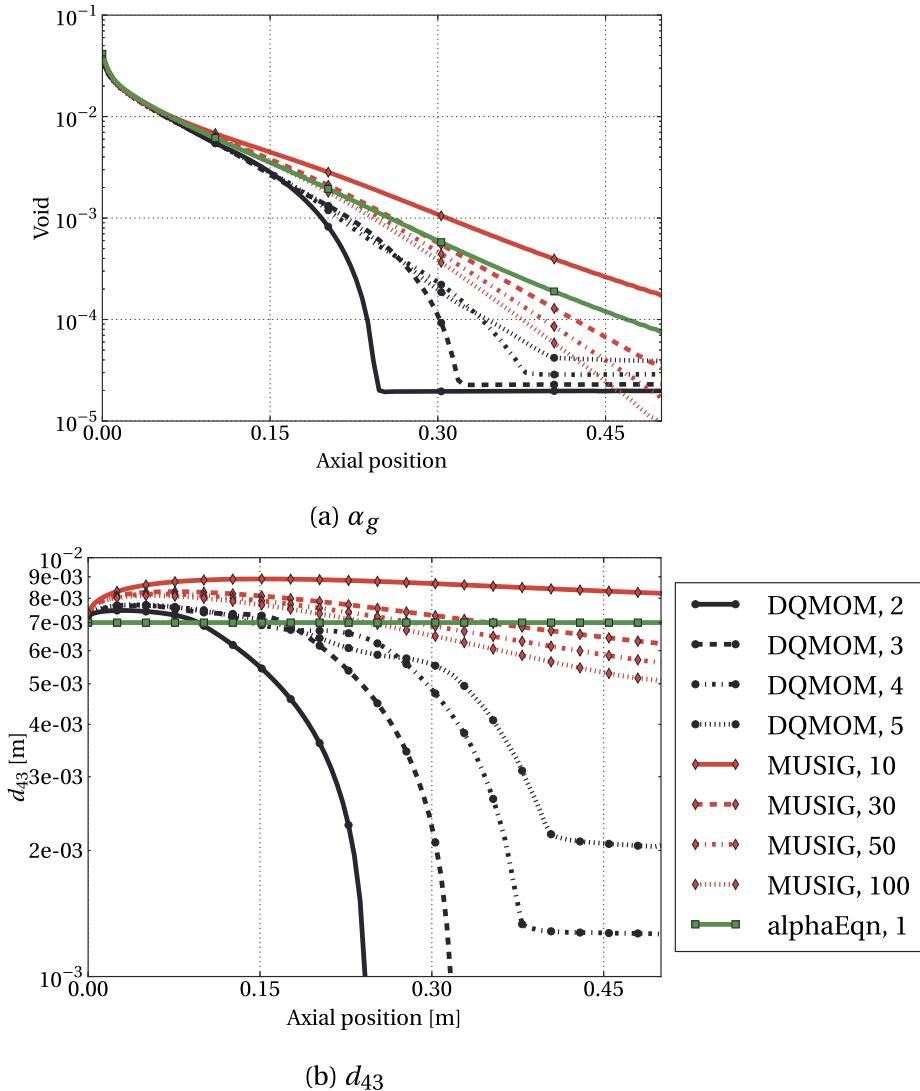
For illustrative purposes, we also include the calculation of a single void fraction equation over the axial axis. In principal, this simple model corresponds to MUSIG in the limit of a single bubble class. The change in the axial void fraction is similar to the MUSIG results with few classes (see the green line against the red lines in [Fig. 4a](#)). The average bubble size is trivially constant throughout the domain (see green line in [Fig. 4b](#)). The precision and accuracy of the single vapor fraction equation must be considered with respect to the change of the bubble size distribution. For small changes in size the single static size will potentially perform sufficiently well. On the other hand, for large changes in the distribution, a larger discrepancy is anticipated as the size used in the condensation model (eq. (51)) introduces an error.

### 5.3.2. Flux formulation with phase change

To evaluate the importance of the pressure equation and mass flux formulation discussed in [Section 3.4](#), we analyze the conservation of mass for both DQMOM and MUSIG. The mass conservation is computed as the difference between the boundary flux at the inlet and outlet and the current mass in the system. The results are presented in [Fig. 6](#) and clearly emphasize the importance of the formulation of the flux. While a conservative solver is directly achieved by performing a consistent derivation of the pressure equation, a wrong formulation leads to a significant loss of mass in the system. Then it is necessary to specifically target the coupled framework as a whole. Besides, the gradient of the loss of mass in the system is the same for DQMOM and MUSIG, which points out that the issue on the mass conservation is associated with the coupling to the two-fluid algorithm, and not with the PBE formulations.

### 5.3.3. Influence of sub-iterations and time steps

In [Table 9](#), the simulation of each case is performed with and without a maximum of 20 sub-iterations. For illustrative purposes, the solvers are forced to use only a single sub-iteration for the latter case. The comparison shows that the computed values for the change of void fraction and the average bubble size often differ, although the magnitude of the difference is small. In particular, the iterative scheme plays a more relevant role for the longer time-steps: in fact the



**Fig. 4.** Vapor fraction ( $\alpha_g$ ) and average bubble size ( $d_{43}$ ) along the axial centerline compared between MUSIG and DQMOM and for a void fraction equation (alphaEqn) with a single static class. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

difference between the iterated and non-iterated results with time-step  $5.0 \times 10^{-3}$  s are larger than the ones for time-step  $5.0 \times 10^{-4}$  s or  $1.0 \times 10^{-3}$  s.

Then, the DQMOM calculations suggests that a longer time-step with iterations allows for a similar precision, with a lower computational cost, with respect to a shorter time-step with only one iteration. For instance, DQMOM-2 with time-step  $5.0 \times 10^{-3}$  s and sub-iteration, gives a  $\Delta d_{43}$  comparable to the one estimated with time-step  $5 \times 10^{-4}$  s and the computational time is significantly reduced. The results also stress that the sub-iterations can be beneficial for the shortest time-step since the mean number of iterations is found to be around 3.

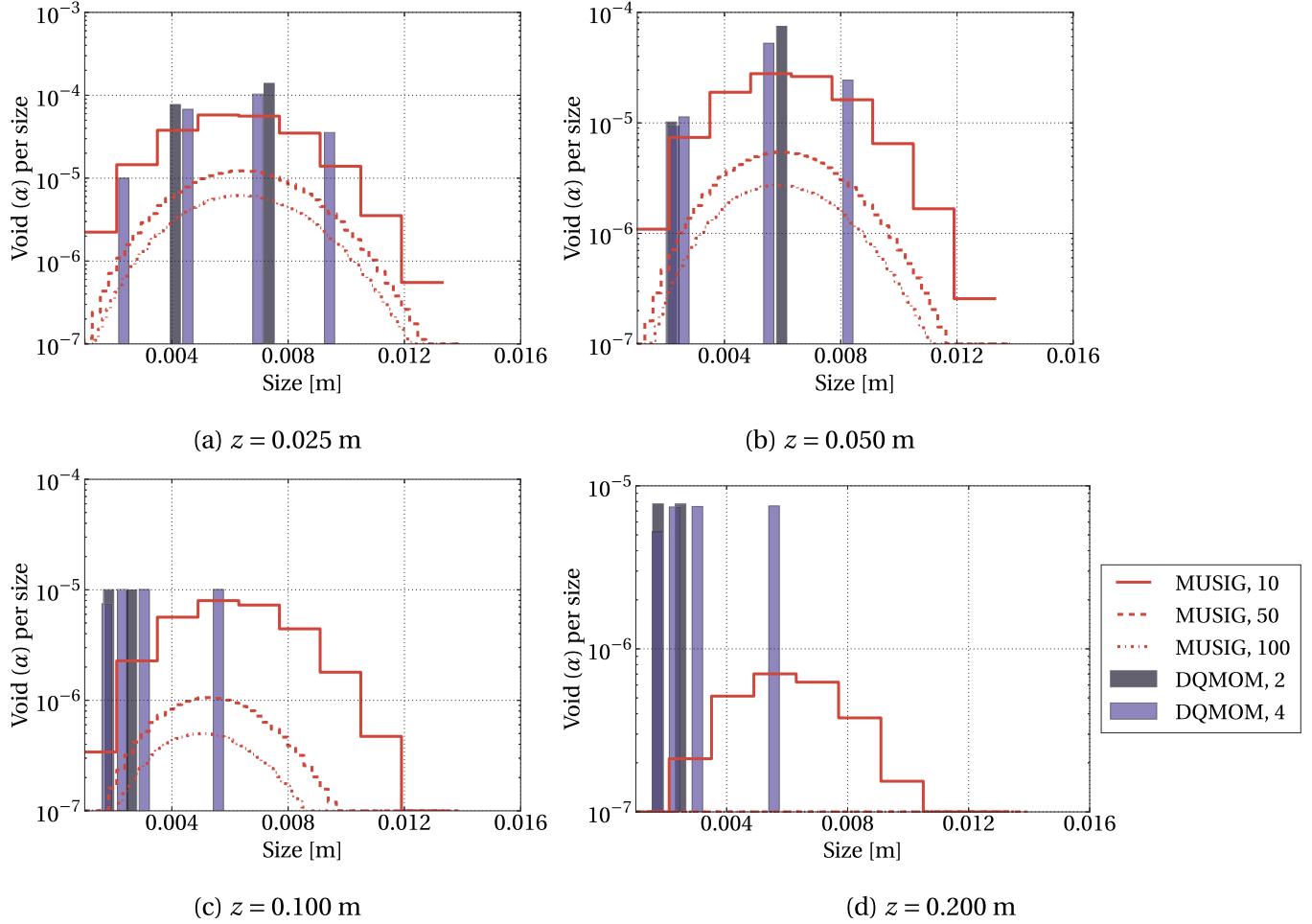
The two cases for MUSIG-10 and MUSIG-30, with time-step  $5 \times 10^{-3}$  s, reach the maximum number of sub-iterations and also a high average number of sub-iterations. This is likely due to the relatively large maximum Courant number, but also reflects the complexity as the couplings between the different sizes are not resolved within the maximum 20 sub-iterations.

Finally, it should be noted that the short simulation time ( $t = 0.05$  s) ensures that the simulation is still evolving in time. This should be compared to the results in Table 8 where the system has reached a pseudo steady-state. For the latter case, the differences between performing sub-iterations and just sweeping once through all class/sizes are less pronounced. However, as indicated by Table 9, we need the sub-iterations to accurately simulate the transient evolution.

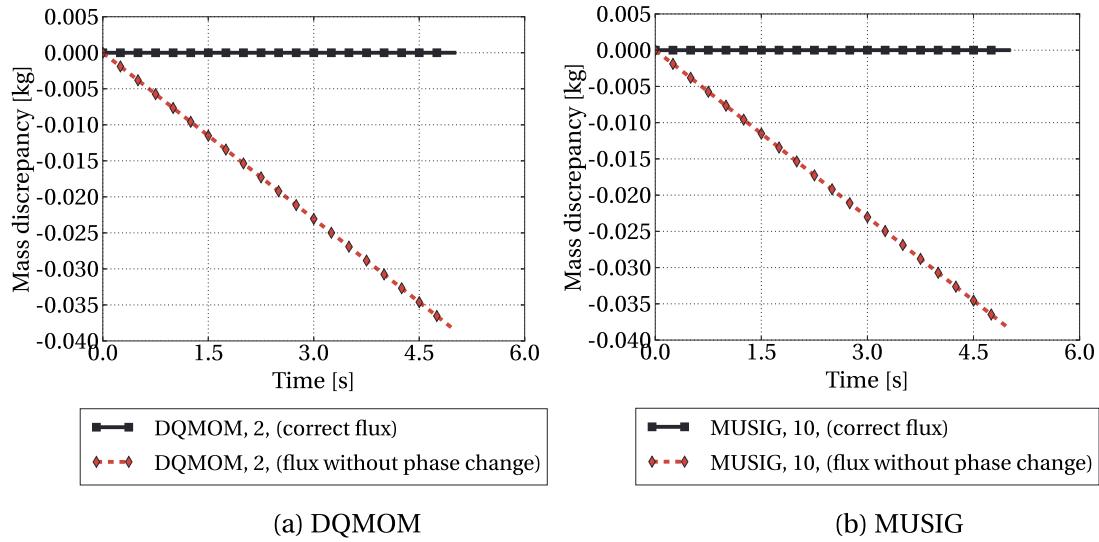
#### 5.4. Case 4

##### 5.4.1. Effect of the wall convective velocity

As presented in Section 2.5, a bubble flux formulation is used to simulate the effect of bubbles leaving the heated wall. Accordingly, a convective velocity is required and is combined with a wall bubble size distribution in order to give the desired void fraction and bubble size distribution for the current time step. To evaluate this scheme we simulate three dif-



**Fig. 5.** Bubble size distributions for Case 3 presented for four axial positions at centerline.



**Fig. 6.** Comparison of flux formulations evaluated in terms of the mass conservation in the system, where the mass discrepancy is computed as the difference between mass flux at the inlet and outlet and the volume integrated mass. The two-dimensional system has a fictitious 1 cm extension in the third direction and the results are therefore presented with the unit kg/cm.

ferent wall convective velocities, as presented in Fig. 7 for DQMOM and MUSIG. For each of the velocities the corresponding bubble distributions are computed, so that the same void fraction is introduced for all the three cases. The results in the figure verify the independence of the convective velocity as such: indeed, the mean diameter and the vapor fraction along

**Table 9**

Relative change in void fraction ( $\Delta\alpha_g$ ) and mean diameter ( $\Delta d_{43}$ ) computed for  $z = 0.03$  m at the centerline of the system for  $t = 0.05$  s. The **Max** correspond to the maximum number of sub-iterations applied in any time step of the simulation and **Mean** correspond to the average number of sub-iterations. **Max U, Co** is the maximum Courant number computed based on the relative velocity between the phases. The wall clock time (**WCT**) is normalized against DQMOM-2.

Solver	$N$	$\Delta t$ [s]	<b>Max</b>	<b>Max U, Co</b>	<b>WCT</b> [-]	$\Delta\alpha_g$ [-]	$\Delta d_{43}$ [-]	<b>Max</b>	<b>Mean</b>
DQMOM	2	5.0e-04	1	0.09	8.1	-0.878	-0.050	1	1.0
			20	0.09	9.9	-0.879	-0.051	3	3.0
		1.0e-03	1	0.17	4.3	-0.875	-0.049	1	1.0
			20	0.17	5.3	-0.879	-0.050	3	3.0
		5.0e-03	1	0.85	1.0	-0.857	-0.040	1	1.0
			20	0.85	1.4	-0.872	-0.044	5	4.5
	4	5.0e-04	1	0.08	15.3	-0.881	-0.034	1	1.0
			20	0.08	27.6	-0.883	-0.034	3	3.0
		1.0e-03	1	0.17	7.8	-0.879	-0.033	1	1.0
			20	0.17	16.5	-0.882	-0.033	4	3.3
		5.0e-03	1	0.84	1.7	-0.859	-0.026	1	1.0
			20	0.84	6.7	-0.875	-0.024	10	6.7
MUSIG	10	5.0e-04	1	0.08	8.4	-0.959	-0.009	1	1.0
			20	0.08	9.7	-0.960	-0.009	4	3.2
		1.0e-03	1	0.17	4.3	-0.956	-0.005	1	1.0
			20	0.17	5.3	-0.956	-0.004	6	4.0
		5.0e-03	1	0.85	1.0	-0.923	0.028	1	1.0
			20	0.84	2.3	-0.929	0.032	20	18.5
	30	5.0e-04	1	0.08	12.6	-0.960	-0.032	1	1.0
			20	0.08	18.7	-0.961	-0.032	4	3.1
		1.0e-03	1	0.17	6.8	-0.956	-0.027	1	1.0
			20	0.17	11.1	-0.957	-0.026	5	4.0
		5.0e-03	1	0.84	1.6	-0.919	0.015	1	1.0
			20	0.84	6.7	-0.930	0.013	20	16.0

the horizontal direction at the mid-elevation, are relatively insensitive to it. So one can conclude that a higher wall bubble velocity does not contribute significantly to the momentum balance in the system.

Furthermore, we include the result of a conventional first cell source term for MUSIG, i.e. addition of the void in the first cell via a source term. The results confirm that the effect of the wall flux formulation is equivalent to the first cell source term.

For the cell source term, discussed in Section 2.5, mesh-independence is not typically achieved for a wall source boiling condition. In Fig. 8, we perform a mesh-independence analysis in terms of the vapor fraction distribution and the mean diameter. The change in the mean diameter is limited. For MUSIG,  $d_{43}$  seems to converge for a finer mesh (Fig. 8b), whereas for DQMOM there is a small residual discrepancy between the finest mesh resolutions (Fig. 8a).

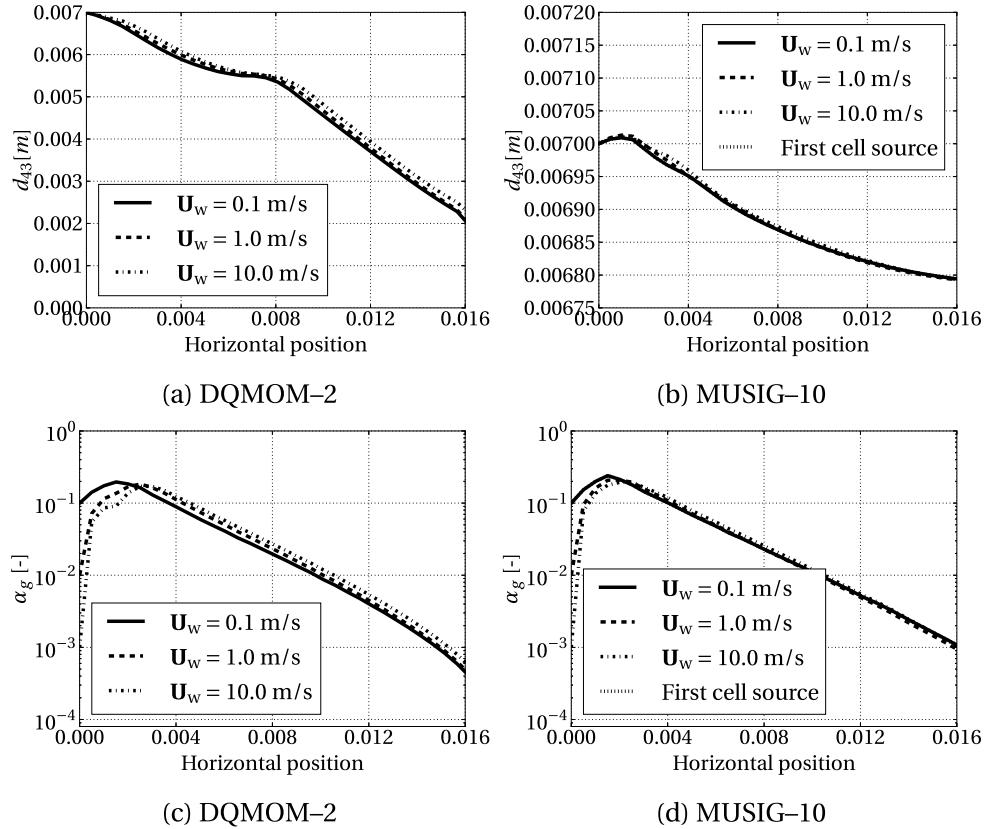
As regards the horizontal void distribution, the different mesh resolutions have a more significant impact. The same trends can be seen for both DQMOM (Fig. 8c) and MUSIG (Fig. 8d). In order to fully discern this type of behavior, a comparison with experimental data or direct numerical simulations would be required.

The wall flux model is also susceptible to the mesh: a different formulation would be necessary to achieve mesh independence for the case of void injected at the wall.

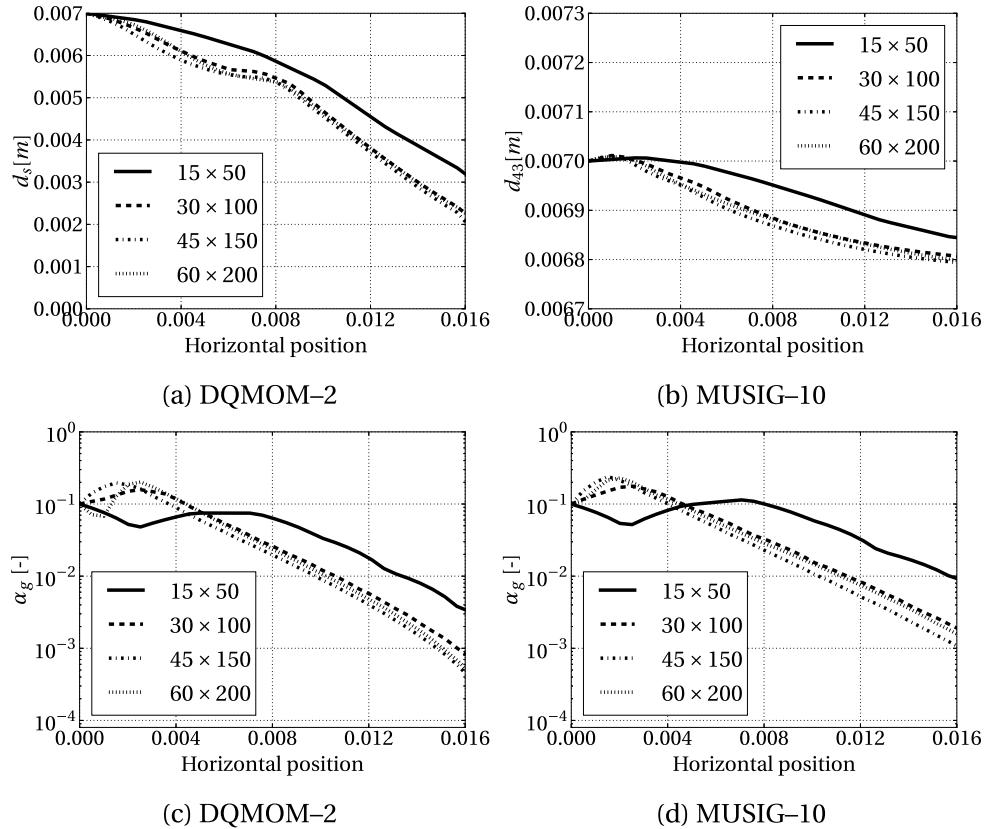
#### 5.4.2. Collisions between bubble size distributions

The complexity and issues associated with the wall condition are further shown in Fig. 9, where the void fraction (on the left) and the mean diameter (on the right) are given with respect to the full simulation domain. The higher concentrations of bubbles are primarily found close to the wall, and, along the horizontal direction, the maximum value is located slightly off the wall. The bubble sizes are slightly larger at the inlet and at the wall, as a direct result of the chosen inlet and wall distributions of bubble sizes. Despite the large bubble at the inlet, the void fraction was however kept low in this part of the system. When the bubbles move away from the wall to the sub-cooled liquid flow, they can condense, break, aggregate: the net effect is a shrinkage of the sizes (see the plot on the right side of Fig. 9, from the dark red area close to the wall to the blue area).

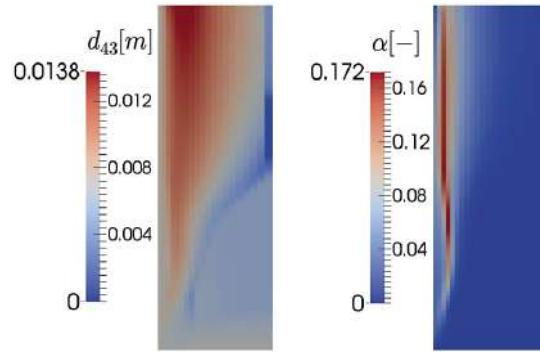
The analysis of the mean diameter points out the issues with colliding size distributions. In fact the bubbles from the wall and from the inlet will interfere and mix with each others, over a relatively large range of sizes. This can cause numerical issues in the solution of the DQMOM transport equations for the weights and abscissas. Due to the threshold that was introduced in the algorithm and that removes the bubbles with very small sizes and keep bubbles with non-negligible sizes (see Section 3.3 and Fig. 4a), discontinuities are limited and the computational scheme is more stable and robust.



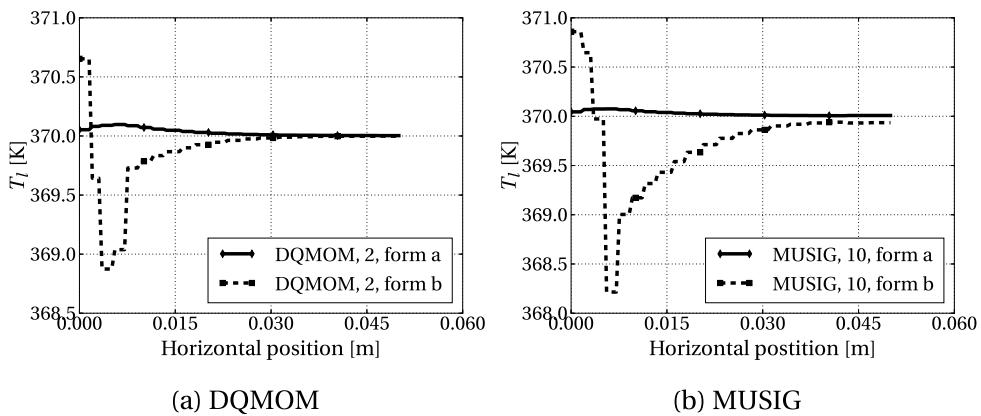
**Fig. 7.** Comparison of different wall bubble convection velocities.



**Fig. 8.** Mesh dependence study for the condition of bubbles entering at the wall compared for MUSIG and DQMOM (2D case). The legend indicates the horizontal  $\times$  axial mesh resolution. A boundary condition of  $\alpha_g = 0.1$  with a fictitious wall velocity of 0.1 m/s is assumed for all cases.



**Fig. 9.** Example of void and size distribution represented by  $d_{43}$  and  $\alpha_g$  for the full domain. The system is scaled in axial direction for presentation purposes. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)



**Fig. 10.** Enthalpy formulation comparison for DQMOM (left) and MUSIC (right), where *form a* corresponds to the discretization consistent with the momentum equation and *form b* corresponds to the flux calculation relying on the equality given by eq. (49).

#### 5.4.3. Enthalpy flux formulation

We also verify the formulation of the enthalpy fluxes that are needed for the convection terms in eq. (48). To do so, we compare the discretization based on eq. (48) (labeled *form a*), and the discretization of the enthalpy equation that relies on eq. (49) (labeled as *form b*). Fig. 10 shows the horizontal liquid temperature of the system, which directly corresponds to the enthalpy. An enthalpy flux that is not consistent with the momentum flux, as in the case of *form b*, gives unphysical profiles of the temperature. Using the consistent *form a*, the correct behavior is predicted: the temperature can increase because of the vapor condensation and it never goes below the inlet temperature.

## 6. Conclusions

A framework for simulations of bubbles in a sub-cooled liquid flow is formulated and implemented. It consists of a DQMOM model for describing the bubble size distribution under condensation, breakage and aggregation, and an Eulerian-Eulerian two-fluid flow model. The framework is characterized in terms of the key aspects of DQMOM and of the coupling to the two-fluid solver. For this purpose, a set of numerical test cases are analyzed. To verify and highlight the features of the DQMOM model, a comparison is also carried out against a model based on a different method for the evolution of the bubble size distribution (MUSIC).

The dynamic abscissas associated with DQMOM are demonstrated to be advantageous for the continuous shrinkage associated with the bubbly flow in subcooled conditions. The results suggest that, similar to other non-heated flow applications of DQMOM, only a few abscissas are required to reproduce the bubble size distribution, in contrast to the typically 30 or more classes needed in MUSIC. Furthermore, the dynamic calculation of the abscissas is shown to allow bubble distributions of very different range of sizes to be resolved in different parts of the domain.

We stress the importance of an iterative scheme for resolving the non-linear dependencies between the different abscissas for DQMOM (or the different classes for MUSIC). Typically, a few iterations, within each time step, are necessary to reach convergence of all the equations. Besides, it is possible to employ larger time-steps, with a consequent reduction of computational costs. We find that a scheme with only one iteration per time-step, induced an error, although relatively small in terms of the change in void fraction. However, it is expected that the benefit of using sub-iterations can be even more important for cases with more challenging transient conditions.

Furthermore, the DQMOM and MUSIG approaches are contrasted in terms of computational times. The DQMOM solver is shown to be between 3 and 10 times faster for a comparable precision of the calculated bubble size distributions. Finally, we demonstrated that the overall solver relies on a consistent formulation, so that mass and enthalpy are correctly preserved.

In addition to these types of numerical investigations, validation studies are needed so that a better understanding of the capabilities and limitations of these methods can be achieved from the analysis of experimental data.

## Acknowledgements

The Swedish Center for Nuclear Technology (SKC) is acknowledged for financially supporting this PhD project. This work is carried out within the DREAM (Deterministic Reactor Modeling) task force at Chalmers University of Technology. The computations were performed on resources at Chalmers Center for Computational Science and Engineering (C3SE) provided by the Swedish National Infrastructure for Computing (SNIC).

## References

- [1] A. Prosperetti, G. Tryggvason, *Computational Methods for Multiphase Flow*, Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 2007.
- [2] D. Shaver, S. Antal, M. Podowski, Modeling and analysis of interfacial heat transfer phenomena in subcooled boiling along PWR coolant channels, in: NURETH-15, Pisa, Italy, May 12–16, 2013.
- [3] E. Krepper, B. Koncar, Y. Egorov, CFD modelling of subcooled boiling – concept, validation and application to fuel assembly design, *Nucl. Eng. Des.* 237 (2007) 716–731.
- [4] D. Ramkrishna, *Population Balances: Theory and Applications to Particulate Systems in Engineering*, Academic Press, 2000.
- [5] S. Lo, *Application of the MUSIG Model to Bubbly Flows*, AEAT-1096, AEA Technol., 1996.
- [6] S. Lo, Some recent developments and applications of CFD to multiphase flows in stirred reactors, in: Proc. AMIF-ESF Workshop on Computing Methods for Two-Phase Flow, Aussois, France, January 12–14, 2000.
- [7] G. Yeoh, J. Tu, Two-fluid and population balance models for subcooled boiling flow, *Appl. Math. Model.* 30 (2006) 1370–1391.
- [8] H. Hulbert, S. Katz, Some problems in particle technology, *Chem. Eng. Sci.* 19 (1964) 555–574.
- [9] R. McGraw, Description of aerosol dynamics by the quadrature method of moments, *Aerosol Sci. Technol.* 27 (1997) 255–265.
- [10] D. Marchisio, R. Fox, Solution of population balance equations using the direct quadrature method of moments, *Aerosol Sci.* 36 (2005) 43–73.
- [11] B. Selma, R. Bannari, P. Proulx, Simulation of bubbly flows: comparison between direct quadrature method of moments (DQMOM) and method of classes (CM), *Chem. Eng. Sci.* 65 (2010) 1925–1941.
- [12] D. Marchisio, A. Barresi, G. Baldi, R. Fox, Comparison between the classes method and the quadrature method of moments for multiphase systems, in: 8th Conference “Multiphase Flow in Industrial Plants”, Alba, Italy, 2002.
- [13] A. Zucca, D. Marchisio, A. Barresi, R. Fox, Implementation of the population balance equation in CFD codes for modelling soot formation in turbulent flames, *Chem. Eng. Sci.* 61 (2006) 87–95.
- [14] R. Fox, F. Laurent, M. Massot, Numerical simulation of spray coalescence in an Eulerian framework: direct quadrature method of moments and multi-fluid method, *J. Comput. Phys.* 227 (2008) 3058–3088.
- [15] G. Yeoh, C. Sherman, J. Tu, On the prediction of the phase distribution of bubbly flow in a horizontal pipe, *Chem. Eng. Res. Des.* 90 (2012) 40–51.
- [16] L. Silva, R. Damian, P. Lage, Implementation and analysis of numerical solution of the population balance equation in CFD packages, *Comput. Chem. Eng.* 32 (2008) 2933–2945.
- [17] A. Buffo, M. Vanni, D. Marchisio, R. Fox, Multivariate quadrature-based moments methods for turbulent polydisperse gas–liquid systems, *Int. J. Multiph. Flow* 50 (2013) 41–57.
- [18] C. Yuan, R. Fox, Conditional quadrature method of moments for kinetic equations, *J. Comput. Phys.* 230 (2011) 8216–8246.
- [19] D. Marchisio, R. Fox, *Computational Models for Polydisperse Particulate and Multiphase Systems*, Cambridge University Press, 2013.
- [20] V. Vikas, Z. Wang, A. Passalacqua, R. Fox, Realizable high-order finite-volume schemes for quadrature-based moment methods, *J. Comput. Phys.* 230 (2011) 5328–5352.
- [21] O. Desjardins, R. Fox, P. Villedieu, A quadrature-based moment method for dilute fluid–particle flows, *J. Comput. Phys.* 227 (2008) 2514–2539.
- [22] A. Buffo, M. Vanni, D. Marchisio, Multidimensional population balance model for the simulation of turbulent gas–liquid systems in stirred tank reactors, *Chem. Eng. Sci.* 70 (2012) 31–44.
- [23] D. Marchisio, R. Vigil, R. Fox, Quadrature method of moments for aggregation-breakage processes, *J. Colloid Interface Sci.* 258 (2003) 322–334.
- [24] D. Lucas, T. Frank, C. Lifante, P. Zwart, A. Burns, Extension of the inhomogeneous MUSIG model for bubble condensation, *Nucl. Eng. Des.* 241 (2011).
- [25] S. Kumar, D. Ramkrishna, On the solution of population balance equations by discretization – III. Nucleation, growth and aggregation of particles, *Chem. Eng. Sci.* 52 (1997) 4659–4679.
- [26] G. Yeoh, J. Tu, *Computational Techniques for Multiphase Flows*, Elsevier Ltd., 2010.
- [27] S. Kumar, D. Ramkrishna, On the solution of population balance equations by discretization – I. A fixed pivot technique, *Chem. Eng. Sci.* 51 (1996) 1311–1332.
- [28] M. Ishii, T. Hibiki, *Thermo-Fluid Dynamic Theory of Two-Phase Flow*, second edition, Springer, 2011.
- [29] J. Ferziger, M. Peric, *Computational Methods for Fluid Dynamics*, Springer, 2002.
- [30] N. Kurul, M. Podowski, Multidimensional effects in forced convection subcooled boiling, in: Proceedings of the 9th International Heat Transfer Conference, Jerusalem, Israel, 1990.
- [31] R. Rzehak, E. Krepper, CFD for subcooled flow boiling: parametric variations, *Sci. Technol. Nucl. Install.* 2013 (2013) 687494, <http://dx.doi.org/10.1155/2013/687494>.
- [32] H. Weller, Derivation, Modeling and Solution of the Conditionally Averaged Two-Phase Flow Equations, Technical report, OpenCFD, 2005.
- [33] H. Rusche, *Computational Fluid Dynamics of Dispersed Two-Phase Flows at High Phase Fractions*, Ph.D. thesis, Imperial College of Science, Technology & Medicine, 2002.
- [34] S.A. Issa, P. Weisensee, R. Macián-Juan, Experimental investigation of steam bubble condensation in vertical large diameter geometry under atmospheric pressure and different flow conditions, *Int. J. Heat Mass Transf.* 70 (2014) 918–929.
- [35] W.E. Ranz, W.R. Marshall Jr, Evaporation from drops. Parts I & II, *Chem. Eng. Prog.* 48 (1952) 141–146, 173–180.
- [36] M. Vanni, Approximate population balance equations for aggregation-breakage processes, *J. Colloid Interface Sci.* 221 (2000) 143–160.
- [37] H. Luo, H. Svendsen, Theoretical model for drop and bubble breakup in turbulent dispersions, *AIChE J.* 42 (1996) 1225–1233.
- [38] E. Krepper, D. Lucas, T. Frank, H.-M. Prasser, P.J. Zwart, The inhomogeneous {MUSIG} model for the simulation of polydisperse flows, *Nucl. Eng. Des.* 238 (2008) 1690–1702.

- [39] L. Schiller, A. Naumann, A drag coefficient correlation, *Z. Ver. Dtsch. Ing.* 77 (1935) 318–320.
- [40] T. Hibiki, M. Ishii, Lift force in bubbly flow systems, *Chem. Eng. Sci.* 62 (2007) 6457–6474.
- [41] M. Lopez de Bertodano, Turbulent Bubbly Flow in a Triangular Duct, Ph.D. thesis, Rensselaer Polytechnic Institute, Troy, New York, 1991.
- [42] F. Bertola, J. Grundseth, L. Hagesaether, C.A. Dorao, H. Luo, K.W. Hjarbo, H.F. Svendsen, M. Vanni, G. Baldi, H.A. Jakobsen, Numerical analysis and experimental validation of bubble size distributions in two-phase bubble column reactors, *Multiph. Sci. Technol.* 17 (2005) 123–145.
- [43] NIST, Thermophysical Properties of Fluid Systems, National Institute of Standards Technology, 2011.



## **Paper VII**

*“On the dynamics of instabilities in two-fluid models for  
bubbly flows”*

K. Jareteg, H. Ström, S. Sasic, C. Demaziére

*Chemical Engineering Science 170 (2017), pp. 184-194*





## On the dynamics of instabilities in two-fluid models for bubbly flows



Klas Jareteg <sup>a,\*</sup>, Henrik Ström <sup>b</sup>, Srdjan Sasic <sup>b</sup>, Christophe Demazière <sup>a</sup>

<sup>a</sup> Division of Subatomic and Plasma Physics, Department of Physics, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden

<sup>b</sup> Division of Fluid Dynamics, Department of Applied Mechanics, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden

### HIGHLIGHTS

- Instabilities in bubbly gas-liquid flows are investigated.
- A shared-pressure two-fluid model is applied to low bubble loadings.
- Inclusion of virtual mass force leads to change in the nature of the system.
- Physical phase heterogeneities follow the numerically-triggered instabilities.
- Implications for using two-fluid models for predicting bubbly flows are discussed.

### ARTICLE INFO

#### Article history:

Received 30 August 2016

Received in revised form 31 January 2017

Accepted 31 March 2017

Available online 4 April 2017

#### Keywords:

Two-fluid

Virtual mass

Bubbly flow

Phase instabilities

### ABSTRACT

In this paper we look at instabilities in bubbly gas-liquid flows and investigate the emergence and characteristics of phase heterogeneities. We apply a shared-pressure two-fluid model to low bubble loadings and demonstrate the existence of persistent gas fraction instabilities of a characteristic size larger than the applied computational grid. In particular, we investigate the influence of a virtual mass effect on the stability of the two-fluid model and we demonstrate a change in the emergence and the dynamics of the phase heterogeneities. The change is accounted to a difference in the degree of hyperbolicity due to the inclusion of the virtual mass force. Furthermore, the results indicate that an initial instability, concluded as numerical in its character, evolves into a state with a physical character of the heterogeneities. We discuss implications of the existence and dynamics of the heterogeneities and the importance of the numerical behavior for interpretation of the results. In particular, we argue that underlying characteristics of the model cannot and should not be concealed with additional sub-models (such as momentum exchange terms) but must be acknowledged in the analysis of results from the two-fluid model for bubbly flows.

© 2017 Elsevier Ltd. All rights reserved.

### 1. Introduction

Bubbly gas-liquid flows are important for many industrial processes due to their advantageous characteristics of heat and mass transfer. However, the complexity of the flow and the extensive range of flow regimes in combination with large industrial devices make computational modeling of such systems a major challenge. For full scale simulations, the computational burden makes it unfeasible to use Direct Numerical Simulation (DNS)-like methods where the interface between the two-phases is directly tracked or reconstructed. Examples of the latter include the volume of fluid method (VOF) (Noh and Woodward, 1976; Hirt and Nichols, 1981), the level set method (LS) (Osher and Sethian, 1988;

Sussman et al., 1994) or front tracking (Unverdi and Tryggvason, 1992). As a consequence of the system sizes, typically much larger than the length scales relevant for a single bubble or gas entity, it is necessary to rely on a simplified representation, such as the two-fluid model (Lahey and Drew, 1989; Ishii, 1990; Prosperetti and Tryggvason, 2007).

In the two-fluid method, both phases are described in an Eulerian frame of reference. The fluids, in the present case gas and liquid, are treated as interpenetrating continua that coexist in every computational volume. The proportions of the respective phases are described as a volume fraction and the flow properties are assumed homogeneous for each phase in each discrete cell. Due to such a local homogenization of the flow, information about the interface between the phases is discarded. For a bubbly flow, this means that the local characteristics such as the size of bubbles is not predicted and that the dynamic behavior of the two phases is also only recovered in an average sense.

\* Corresponding author.

E-mail address: [klas.jareteg@chalmers.se](mailto:klas.jareteg@chalmers.se) (K. Jareteg).

Additionally, the governing equations of the two-fluid model are typically derived under the assumption of a slow variation in space of the phasic properties, for example the volume fractions of the phases (Lahey and Drew, 1989). Such a requirement is an attempt to reach the separation of scales, where the void fraction fluctuations should not rapidly change on the scale of the computational mesh. The assumption of a slow variation is a major drawback when it comes to the applicability of the two-fluid model. An assumption of small gradients, often neglected in practice, limits the validity of the model to bubbly flows of low bubble loadings. For flow regimes with higher gas fractions, such as slug or churn flow, the computational cells would have to be enlarged to the extent that no relevant fully dimensional resolution could be achieved. For some applications, a coarse mesh and one dimensional (1D) conservation equations are of relevance to compute macroscopic system properties (Prosperetti and Tryggvason, 2007). In this paper we shall consider low ranges of the void fraction (i.e. the gas fraction) in an attempt to fulfill the discussed criteria.

Although the phases are represented in a spatially averaged sense, the dynamic behavior of the phase fractions and velocities is potentially important both for mass and heat transfer applications. It is thus of interest to accurately capture possible variations and also the phenomena that contribute to the appearance of non-uniform distributions of the void fraction. We will refer to such a non-uniform state in the void distributions as heterogeneities, i.e. heterogeneous in terms of the spatially averaged phase fraction fields. In contrast to fully resolved interfaces in VOF or LS, the two-fluid formulation can only capture meso-scale fluctuations, here used to denote heterogeneities larger than the computational cell but smaller than the system size. The meso-scales are throughout the paper significantly larger than the actual bubble size, and thus in accordance with the requirement of slow variations over the averaging volume.

From experiments, it is well established that initially homogeneous bubbly flows can become heterogeneous at high enough bubble loadings (Mudde et al., 2008). The physical mechanisms responsible for this transition are however not yet fully understood. There have been several attempts to identify those mechanisms based on mathematical or numerical analyses of two-fluid models, resulting in a range of possible, and sometimes even conflicting, suggestions for routes leading to an unstable behavior in the sense of fluctuating values of the phase fractions or velocities (Sankaranarayanan and Sundaresan, 2002; Lucas et al., 2005, 2006; Monahan and Fox, 2007b; Yang et al., 2007; Chen et al., 2009; Yang et al., 2010). We will refer to the term instability for the cause and transition of the homogeneous to the heterogeneous void fraction distribution.

Complementary to the theoretical studies of flow regime transitions, many authors have attempted to capture the experimentally demonstrated change from uniform to heterogeneous flow based on simulations. Notably, Monahan et al. (2005) simulated the experiments by Harteveld (2005) with a variety of momentum exchange terms and proposed the need for a large number of terms to be accounted for to accurately capture the transition. As noted in another paper from the same group (Monahan and Fox, 2007a), the simulations can lack stability (in the sense of reaching a physical and convergent solution) in the limit of small bubbles, which is particularly interesting as the two-fluid model is derived under the assumption of sufficiently small bubbles and slow variations relative to the averaging scales. As made evident from the referenced simulations, not all properties of the two-fluid model are well understood. This is especially pertinent for the dynamic behavior of heterogeneities in fully-dimensional (3D) flow simulations, where an excessive use of additional model terms is likely to significantly contribute with a diffusivity, in effect an excessive vis-

cosity, and thus overshadow potential fluctuations of phase fractions.

In relation to the discussed numerical issues, it is known that the degree of hyperbolicity affects the numerical stability of a two-fluid model (Drew et al., 1979; Lahey et al., 1980; Dinh et al., 2003) and it is therefore of interest for the current investigation of the dynamics of the two-fluid model. In formulations based on 1D conservation equations, issues with instabilities have been seen for models with no viscosity (Lhuillier et al., 2010). As a remedy, a mathematical or numerical regularization may be applied to achieve hyperbolicity (Dinh et al., 2003). A numerical regularization is, in its simplest form, induced from a coarse spatial discretization which results in much numerical diffusion as discussed by Pokharna et al. (1997). The need for viscosity (physical or numerical) is confirmed by linear stability analysis based on simplified models, where it can be shown that such terms enhance the stability of the short wave lengths (Arai, 1980).

Another way to deal with the model instabilities is to include specific momentum exchange terms directly aimed to stabilize the solution in the numerical sense. An example of this is to include the virtual mass force in the formulation of the governing equations. The virtual mass force corresponds to the force exerted on a moving object immersed in a fluid when it accelerates relative to its surrounding, and hence must also accelerate some of the surrounding fluid. Although the effect of that force may be of little significance to the final results, the virtual mass force can have a profound effect on the numerical behavior of the problem (Lahey et al., 1980; Toumi and Kumbaro, 1996). Theoretical studies on 1D models for various formulations of the virtual mass force confirm that hyperbolicity is obtained, but typically only for a sufficiently low void fraction (Prosperetti and Satrape, 1990). Such a finding is again of interest and importance for the two-fluid model applied to low bubble loadings and small bubbles as it potentially affects the dynamics of the heterogeneities. As relates to the dynamics of bubbly flow, studies have shown that the virtual mass force is crucial for accurate predictions of transient phenomena such as a bubble plume oscillation (Mudde and Simonin, 1999; León-Becerril et al., 2002). As such, an inclusion of the virtual mass force has multiple advantages, both improving numerical characteristics and the predictability of transient behavior.

In effect, the two-fluid model is typically accompanied with a turbulence model. The turbulence model enhances stability of the two-fluid model due to existence of a significant turbulent viscosity. However, to rely on this approach is not straightforward if the two-fluid model is applied to a successively refined mesh, as – in contrast to single-phase turbulence – meso-scale instabilities in disperse two-phase flows typically originate from the very small scales, which are increasingly well resolved as the cells become smaller (Agrawal et al., 2001; Ström et al., 2015). Furthermore, turbulence models, such as Reynolds-averaged Navier-Stokes (RANS) models, are often applied to the continuous phase, not necessarily taking the effect of the dispersed flow into account. There is still no consensus on how to adapt well-established single-phase two-equation turbulence models to properly account for complex two-phase phenomena, such as bubble-induced turbulence (Rzehak and Krepper, 2013). In relation to the virtual mass force, Lhuillier et al. (2013) argue that it needs to be combined with a model for the turbulent velocity fluctuations in order to guarantee hyperbolicity. In a similar manner, Stewart (1979) demonstrate that the two-fluid equations are well-behaved given a large enough momentum exchange between the phases and a coarse enough mesh. However, it should be emphasized that such a finding does not guarantee that the underlying equations are stable, but rather that the model is well-behaved on a coarse mesh without addressing the underlying ill-posedness.

In addition to forces for the momentum exchange and the turbulence model, some authors propose a two-fluid formulation with separate pressures for each phase (Ransom and Hicks, 1984; Lee et al., 1998; Chung et al., 2000). This is in contrast to the widely used formulation of a single, shared, pressure, also applied in this paper. Although the interpretation of the dispersed phase pressure is not fully clear, different two-pressure formulations are shown to add additional diffusivity, thus stabilizing the two-fluid model (Dinh et al., 2003). The numerical smearing is further confirmed by Munkejord (2010) who shows that, for a dual pressure approach with a short relaxation time between the phases, the system of equations shows similar ill-posedness as the single-pressure two-fluid model. In the current scope we are primarily concerned with the effect of the momentum exchange, and as the numerical properties were discussed to be similar to the single-pressure model, the two-pressure approach is not considered. Further, we note that a specific bubble phase pressure (Spelt and Sangani, 1997) is occasionally applied. Similar to the above reasoning on additional momentum exchange mechanisms, we will show that an excessive number of forces is not likely desirable for the studies of the underlying numerical properties of the two-fluid model.

We aim in the current paper to qualitatively unite the experimentally proved bubble instabilities with the extensive 1D analyses and formulations of the two-fluid methodology in the presented 3D simulations. Such an aim involves the analysis of the combination of issues related to the lack of hyperbolicity, the experimentally observed instabilities as well discerning the implications of conventional use of the two-fluid model involving a large number of sub-models.

We investigate the capability of the two-fluid model based on a shared-pressure formulation to resolve the dynamics in a volume-averaged bubbly flow. We address the question of whether the observed instabilities are physical or purely numerical by contrasting the numerical results obtained with two similar two-fluid models for the canonical problem of an initially homogeneous periodic bubbly flow. The main difference between the two models is their degree of hyperbolicity, as one accounts for the effects of the virtual mass force whereas the other does not. This work thus extends the analysis of Lahey et al. (1980) on the virtual mass force effects on the numerical stability of two-phase flows, to the unsteady problem of a transition from a homogeneous to a heterogeneous flow.

In detail, we apply the two-fluid model on a fully periodic 3D system with initially uniform fields for both the phase fractions and velocities. As such, the system includes no initial spatial variations and no gradients from walls. We characterize the growth and existence of instabilities in the void fraction field by extracting time series of relevant fields as well as computing a global index for the heterogeneity in the system as the solution is advanced in time.

The analysis is challenging in the sense that the studied models are shown to exhibit numerical instabilities, typically both unwanted and inadvertent from a physical point of view. At the same time, these models are used to investigate the stability of bubbly flow systems. Our aim is to demonstrate whether the instabilities and fluctuations seen in the simulations are physical. We attempt to discern the different stages of the instabilities by studying the spatial distribution and the length scales of the fluctuations in combination with a temporal analysis of the global heterogeneity in the system.

The paper is structured as follows. We introduce the governing equations and the implementation of the two-fluid model in Section 2. The 3D system and the initial and boundary conditions are defined in Section 3. In Section 4, we present numerical results that illustrate the cases with the virtual mass force included or excluded. Finally, in Section 5 we give a conclusion regarding the

feasibility to recover physical heterogeneities within the presented type of the two-fluid model.

## 2. Two-fluid model

In this section, we present a two-fluid model for the simulation of bubbly flows. We give the governing equations and also briefly describe the implementation with a specific focus on the virtual mass force. We will show that, in spite of the attempted simplicity, the solver exhibits complex instabilities in the phase fraction distributions (Section 4).

### 2.1. Conservation equations

The derivation of the conservation equations follows a standard procedure, where the local instantaneous Navier-Stokes equations are typically time and/or space averaged (Lahey and Drew, 1989; Ishii and Hibiki, 2010; Prosperetti and Tryggvason, 2007). Although some authors used ensemble averaged equations (e.g. Zhang and Prosperetti, 1997), the majority of implementations reported in the literature relies on a volume averaging approach, especially for the closure of the interfacial momentum transfer (as discussed by Jakobsen (2008)).

The formulation applied in this work closely follows the procedure outlined by Weller (2005), where the conservation equations are ensemble averaged, but with the mentioned volumetric average for the momentum exchange terms. The continuity equation is given by

$$\frac{\partial \alpha_i \rho_i}{\partial t} + \nabla \cdot (\alpha_i \rho_i \mathbf{U}_i) = 0, \quad (1)$$

where  $\rho_i$  is the density,  $\alpha_i$  the phase fraction and  $\mathbf{U}_i$  is the velocity. In all equations  $i$  indicates the phase. In the applied framework, the continuity equation is reformulated to

$$\frac{\partial \alpha_g \rho_g}{\partial t} + \nabla \cdot (\alpha_g \rho_g \mathbf{U}_c) + \nabla \cdot (\alpha_g (1 - \alpha_g) \rho_g (\mathbf{U}_g - \mathbf{U}_l)) = 0 \quad (2)$$

where  $\mathbf{U}_c$ , the mean velocity, is given by

$$\mathbf{U}_c = \alpha_g \mathbf{U}_g + \alpha_l \mathbf{U}_l. \quad (3)$$

The formulation of the continuity equation in terms of the mean velocity results in an additional term which is meant to help the coupling of the phases and result in bounding for the extremes  $\alpha_g = 0$  or  $\alpha_g = 1$  (Weller, 2005; Rusche, 2002). In the latter work it is also noted that the additional non-linearity introduced in Eq. (2) in comparison to a straightforward implementation, like Eq. (1), might lead to difficulties in convergence. A direct remedy for such issues is however to apply sufficiently low Courant numbers.

In a similar way to the continuity equation, the momentum conservation equation on a standard form, such that

$$\frac{\partial \alpha_i \rho_i \mathbf{U}_i}{\partial t} + \nabla \cdot (\alpha_i \rho_i \mathbf{U}_i \mathbf{U}_i) = -\nabla \cdot (\alpha_i (\bar{\tau}_i + \bar{\tau}_i^t)) - \alpha_i \nabla (P) + \alpha_i \rho_i \mathbf{g} + \mathbf{M}_i, \quad (4)$$

is re-written in a phase-intensive manner

$$\alpha_i \rho_i \frac{\partial \mathbf{U}_i}{\partial t} + \alpha_i \rho_i \mathbf{U}_i \nabla \cdot \mathbf{U}_i = -\nabla \cdot (\alpha_i (\bar{\tau}_i + \bar{\tau}_i^t)) - \alpha_i \nabla (P) + \alpha_i \rho_i \mathbf{g} + \mathbf{M}_i \quad (5)$$

where  $\bar{\tau}_i$  denotes the viscous stress tensor,  $\bar{\tau}_i^t$  the turbulent stress tensor,  $P$  the shared pressure and the interfacial momentum transfer is given by  $\mathbf{M}_i$ . The viscous stress tensor is given by:

$$\bar{\tau}_i = -\mu_i \left( \nabla \mathbf{U}_i + \nabla \mathbf{U}_i^T - \frac{2}{3} \bar{\mathbf{I}} \nabla \cdot \mathbf{U}_i \right) \quad (6)$$

where the fluids are assumed to be Newtonian. Eq. (5) is derived by combining the continuity equation of the respective phase to Eq. (4). The direct advantage of Eq. (5) over Eq. (4) is that it is possible to solve also for the extremes, i.e.  $\alpha_g = 0$  or  $\alpha_g = 1$ .

For the turbulent contribution to the stress tensor, we note that the bubbly flow will see different sources to the velocity fluctuations, namely large-scale vortical structures, bubble-induced turbulence and shear-induced turbulence (Ojima et al., 2014a,b; Tomiyama and Shimada, 2001). For the case of low gas fractions, Ojima et al. (2014a) showed that, for dilute bubbly flows, the large-scale vortical structures are the main contributor to the turbulent kinetic energy. Furthermore, Monahan et al. (2005) argue that a multiphase turbulent model is of doubtful validity for a very dilute flow, as applied in this paper. As an example, Becker et al. (1994) showed that a typical  $k - \epsilon$  RANS-based model overpredicts the turbulent viscosity, which has also been shown to smear the fine-scale structures of interest for the emergence of instabilities (Jareteg et al., 2015). Based on the previous remarks in combination with a very fine resolution of the mesh and with the aim to distinguish the effect of the virtual mass effect, we perform all simulations without a turbulence model, i.e.  $\bar{\tau}_i^0 = 0$ .

In further detail, the approach followed here is thus to apply an implicit LES type of methodology, where the turbulent kinetic energy is not explicitly modeled. Instead a fine grid is applied in an attempt to resolve the large-scale vortical structures as discussed above, and the numerical dissipation is used in lieu of the physical dissipation (Boris et al., 1992). It should be noted that the final form of the conservation equations are identical for the case of time-averaging (for a RANS-based method) or from a volume-average (for a LES-like interpretation). The difference between the two mentioned averages lies with the interpretation of the terms related to fluctuations.

As regards the modeling of the effect of the bubble-induced turbulence, different alternatives have been proposed in the literature including a turbulent dispersion force (Bertodano, 1998), a drift velocity model (Bel Fdhila and Simonin, 1992) or a turbulent contribution to the added mass (Ali et al., 2011). Also for the latter, the effect is essentially introducing an additional gas dispersion. Arguably, there are cases where such a contribution is essential. However, for the purpose of the current study such effects are not studied as our purpose is not primarily to accurately predict the flow but investigate the characteristics of the two-fluid model.

It should be noted that, in the case of incompressible phases and adiabatic conditions, we can extract the densities  $\rho_i$  out of the spatial and temporal derivatives. However such an assumption is not directly necessary following the procedure above which can be used also for the case of phase change and compressible conditions. However, in the current paper, we limit ourselves to isothermal simulations. As a result, no energy equations are solved and the phasic densities and other thermophysical properties are assumed constant. Ransom and Hicks (1984) showed that the energy equations do not affect the numerical characteristics of the two-phase equations.

## 2.2. Interfacial momentum transfer

The interfacial momentum transfer ( $\mathbf{M}_i$ ) is decomposed into individual contributions, based on a Lagrangian approach with bubbles (see e.g. Crowe et al. (1998)). For the Eulerian-Eulerian approach, based on a volumetric calculation of a body force, the assumption of disperse flow, i.e. discrete bubbles, relies on that a sufficiently low gaseous fraction is simulated. With higher fraction and larger structures of gas, the closures for single bubbles are no longer valid. Based on this, we focus on a low gaseous fraction regime, with a maximum of 10% volumetric gas.

The current work is concerned with the effect of the virtual mass force on the stability of the solver and emergence of heterogeneities in the gas fraction field. With this scope in mind, we focus on the drag and virtual mass forces only. The drag force is formulated as

$$\mathbf{M}_{g,drag} = -\frac{3C_d\rho_l\alpha_g\alpha_l\|\mathbf{U}_g - \mathbf{U}_l\|}{4d_b}(\mathbf{U}_g - \mathbf{U}_l), \quad (7)$$

where  $d_b$  is the bubble size and  $C_d$  is the drag coefficient which is calculated based on the particle Reynolds number,  $Re_b$ , according to the Schiller-Naumann correlation (Schiller and Naumann, 1935):

$$C_d = \frac{24}{Re_b} \left( 1 + 0.15Re_b^{0.687} \right). \quad (8)$$

The virtual mass force is modeled as:

$$\mathbf{M}_{g,vm} = -C_{vm}\alpha_g\alpha_l\rho_l\left(\frac{DU_g}{Dt} - \frac{DU_l}{Dt}\right), \quad (9)$$

where the coefficient  $C_{vm}$  is in general modeled with a gas fraction,  $\alpha_g$ , dependence (Pauchon and Banerjee, 1986; Zuber, 1964; Mokeyev, 1977; Van Wijngaarden and Jeffrey, 1976) and a size dependence (Sankaranarayanan et al., 2002). As concerns the gas fraction dependence, the models generally agree on  $C_{vm} = 0.5$  in the limit of low fractions. Accordingly, in the current paper we apply a fixed value of  $C_{vm} = 0.5$ . It should be noted that a simplified 1D stability analysis of the two-fluid model with a virtual mass force included indicate that the stability properties has a certain dependence on  $C_{vm}$ , where a fixed value with no fraction dependence is shown to have the largest stability region (Prosperetti and Satrape, 1990).

Prior to the description of the system and presenting the results, we emphasize the implications of size dependence in the virtual mass and drag correlations. In the derivation of the closure of the momentum interface exchange between the phases in the two-fluid model, we lose the information about the bubble size distribution. As a remedy for this, the open literature contains a large range of population balance methods (PBMs), where the direct quadrature method of moments (DQMOM, see e.g. Marchisio and Fox (2005, 2010, 2013)) and the multiple size group method (MUSIG, Lo (1996)) are notable examples. Whereas such methods attempt to handle the effect of bubble aggregation and breakage, the general implementation of the interfacial forces is the same, i.e. based on a volumetrically averaged source term. For the current work, we discard the additional complexity of a bubble size distribution, but emphasize that the mathematical characteristics of the equations are presumably not significantly altered from the additional information of the bubble sizes, although this is a field for future research.

## 2.3. Numerics and implementation

The conservation equations are implemented, discretized and solved within the CFD software OpenFOAM (Weller et al., 1998). The implementation of the two-fluid solver is an adaptation of the PISO algorithm (Issa, 1986), with a momentum predictor step, and with a combined continuity equation for both phases that is used for the shared pressure equation. Further details of the solver for similar applications are found elsewhere (Jareteg et al., 2015).

## 3. System description

### 3.1. Geometry and initial conditions

All simulations are performed on a fully periodic domain of dimensions  $0.1 \text{ m} \times 0.4 \text{ m} \times 0.1 \text{ m}$ , where the  $x-z$  plane is

**Table 1**

Thermophysical parameters and initial conditions for both phases.

Liquid density, constant $\rho_l$	1000 kg/m <sup>3</sup>
Gas density, constant $\rho_g$	1 kg/m <sup>3</sup>
Liquid viscosity, constant $\mu_l$	10 <sup>-3</sup> Pa s
Gas viscosity, constant $\mu_g$	10 <sup>-3</sup> Pa s
Bubble size, constant $d_b$	0.68 mm
Void fraction, uniform initial condition $\alpha_g$	0.05
Liquid velocity, uniform initial condition	(0, 0, 0) m/s
Gas velocity, uniform initial condition	(0, 0, 0) m/s
Gauge pressure, uniform initial condition	0 Pa

horizontal and the  $y$ -axis is aligned with the gravitational acceleration. The calculations are performed on two different mesh resolutions,  $32 \times 64 \times 32$  and  $48 \times 96 \times 48$  denoted *Coarse* and *Fine*, respectively. It should be noted that the domain should be large enough not to produce spatially self-correlating results. In the current case, the domain size was confirmed large enough by studying the cross-correlation of the void fraction field.

The thermophysical parameters together with the initial conditions are presented in Table 1. Additionally, for the pressure a jump condition is applied between the axial top and bottom boundaries. The purpose of such a condition is to outbalance the effect of gravity, avoiding a downward acceleration of the flow, and thus allowing an axially consistent periodic flow. The magnitude of the jump is computed to exactly balance the weight of the gas and liquid phases. As a result, the magnitude of the jump term is different for different bubble loadings. For the velocity, initially flat and equal values are used for both phases. The assumption of no initial drift velocity between the phases has been compared with simulations with an initial difference in phase velocity and the general trend as regards the void fraction were found to be the same. The bubbles are assumed mono-dispersed and, as discussed in Section 2.2, the effects of aggregation or breakage are not considered.

In the last Section (4.3) of the results, three different average gas fractions are compared. We emphasize that for all simulations a completely flat gas distribution is used as the initial condition. Accordingly, we let the instabilities be triggered by the numerics (i.e. limited floating precision or small perturbations induced from not solving the matrix systems to infinitely low residuals).

### 3.2. Numerical schemes and time step choice

For the presented results, second order schemes are used in space, whereas an explicit scheme is used for time discretization. We strive to limit the influence of the time discretization and apply a very small time step ( $\Delta t = 50 \mu\text{s}$ ), which results in low Courant number (on the order of  $10^{-3}$  or lower).

For the spatial accuracy, an increase of the resolution is less straightforward. As the spatial averaging volume approaches the size of the simulated bubbles, the underlying theoretical assumptions of a slowly varying field is no longer valid (Nigmatulin, 1979; Lahey and Drew, 1989).

On the other hand, Prosperetti (2003) argues that, although the validity of the smallest resolved scales are doubtful, the concept and mechanism of a grid refinement is profound. As such, the refinement will lead to less dissipation from the spatial discretization, and thus it is of interest in the evaluation of the applicability of the two-fluid model to simulate void fraction fluctuations. Such an urge to minimize dissipation is confirmed by Oey et al. (2003), who showed that a too excessive numerical dissipation will render the system steady, smearing the transient behavior of the two-fluid model. As discussed above, the extreme of a coarse mesh can potentially render the system hyperbolic. However, since this is

achieved entirely through numerical diffusivity, there is no gain in the quest of the dynamic behavior of the two-fluid model.

In addition, we note that some authors, e.g. discussed by Ma et al. (2015), argue that the requirement of separation of scales must be enforced only during the derivation of the conservation equations. With such a reasoning the resulting set of PDEs can be solved on an arbitrarily fine mesh. However, in the same context, Jakobsen (2008) suggests that the interpretation of the solution is strictly limited to the smallest scales considered in the formulation of the closure laws. In the same manner, we argue that the use of a very fine mesh is doubtful for a system of equations that is closed with an interfacial closure model aimed for significantly larger averaging volumes. For the derivation of the two-fluid equations to be valid, we are strictly limited to resolutions where we can rely on separation of scales. For application on computational grids where such a separation is not achieved, the conservation equations would in principle contain additional terms due to non-local effects (see e.g. Espinosa-Paredes (2012)).

For the two mesh resolutions applied in this paper, the bubble diameter per cell size in axial direction corresponds to 0.05 (coarse mesh) and 0.1 (fine mesh). In terms of the underlying assumptions of separation of scale, this is a reasonable choice as a sufficiently smooth variation of flow fields is expected. The chosen mesh resolution is also in line with the reasoning of Stewart (1979), who argues that the resolution should be coarser than the bubble size characteristic for the momentum exchange.

### 3.3. Data extraction and post-processing

To analyze the emergence and persistence of spontaneous heterogeneities we extract time resolved values of the gaseous volume fraction field, the velocity for both phases and the interfacial momentum forces. Geometrically resolved data for a smaller number of time steps are stored for visual representation of the system.

Additionally, we compute a time-resolved uniformity index, defined as

$$\Phi(t) = \frac{\alpha_{g,\max} - \alpha_{g,\min}}{\alpha_{g,\text{ave}}}, \quad (10)$$

where  $\alpha_{g,\max}$  and  $\alpha_{g,\min}$  are the instantaneous maximum and minimum gas fractions for all cells and  $\alpha_{g,\text{ave}}$  is the constant domain average void fraction. Such a parameter gives a global measure of the phase fraction heterogeneity, and is used to assist in discussion on the emergence and character of the instabilities.

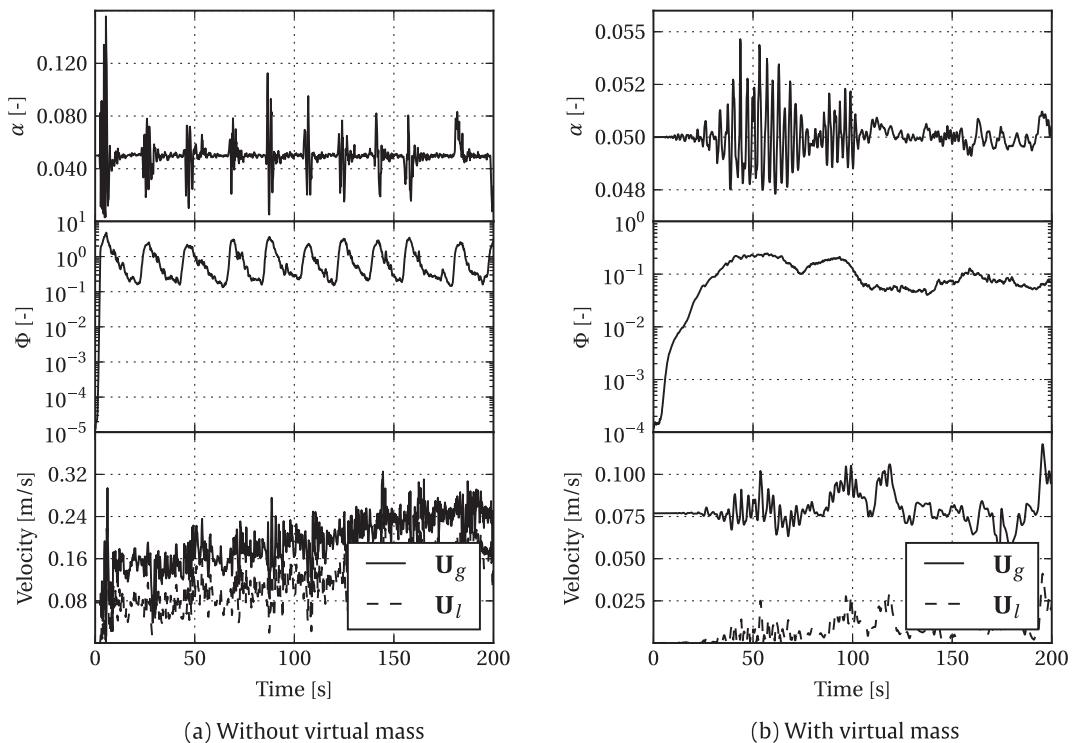
## 4. Numerical results

In this section we present results from the simulations of a fully periodic system. We note that all simulated cases, to a varying degree, exhibit instabilities such that the initially flat distributions of all properties evolve to a heterogeneous state. We demonstrate that multiple stages are seen in the simulations, and we discuss the validity of the presented results from a physical point of view.

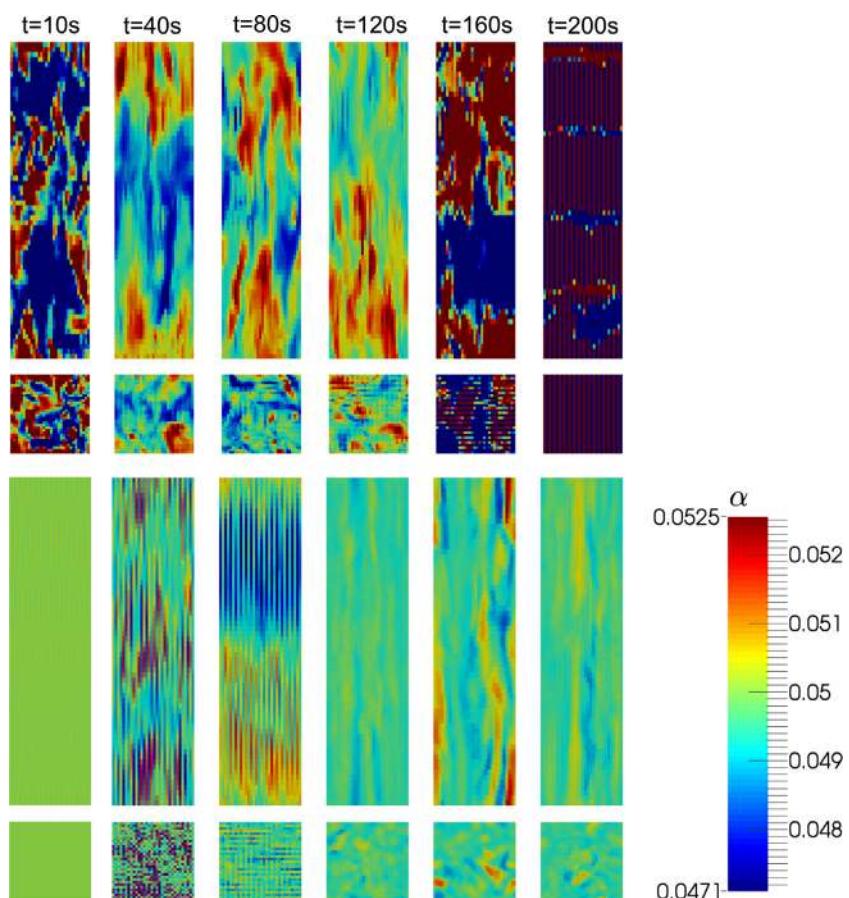
It should be emphasized that the presented data all correspond to a single realization for the given combination of different bubble loadings and included forces. Due to the random character of the instabilities, a different set of settings or a different floating point perturbation will lead to a different realization of the system. The cases presented should thus be considered as representative and of qualitative relevance.

### 4.1. Void fraction and velocity time series

Fig. 1 presents time series of the void fraction, the time-resolved uniformity index and the phase velocities, with and without the



**Fig. 1.** Temporal development of the void fraction field (top), the uniformity index (middle) and the magnitude of the velocity field for both phases (bottom). Both cases exhibit an initial transient in the void fraction and the uniformity index, although significantly faster for the case without the virtual mass force. The simulations are performed on the coarse mesh with the initial condition  $\alpha = 0.05$ .



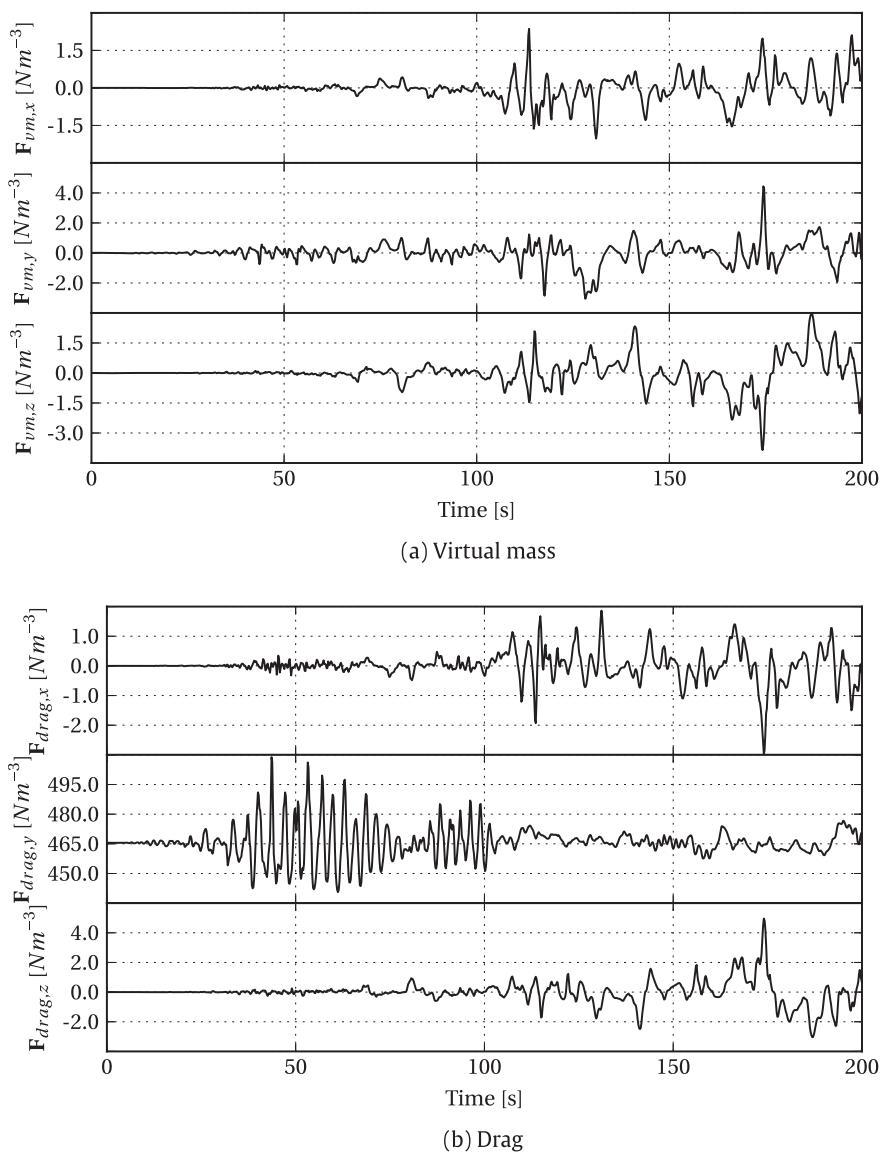
**Fig. 2.** The gas fraction field displayed for 6 time steps (as indicated at the top of the figure) with the case of no virtual mass (top two rows) and virtual mass included (bottom two rows), displayed in the  $x - y$  plan (rectangular figures) and the  $x - z$  plane (square figures). All figures are presented for the same color legend which corresponds to the extremes for the upper row (no virtual mass) at  $t = 120$  s.

virtual mass force. It should be noted that whereas the time-resolved uniformity index is computed globally, the void fraction and velocity profiles are extracted locally from the centre of the fully periodic domain. Furthermore the data is presented with different axes for the different cases in order to clearly visualize the qualitative dynamic behavior of the presented quantities. As a first analysis, we note that the case without the virtual mass force exhibits significantly larger magnitudes of both the void fraction peaks and the time-resolved uniformity index. In addition the velocity for the case without virtual mass (bottom right in Fig. 1) displays a slight acceleration in the velocity of both phases, whereas the drift velocity is quantitatively similar for both cases.

For the case without the virtual mass force we see repeated bursts in the void fraction as well as the time-resolved uniformity index. The results suggest that the system sees a growing instability which decays over approximately 25 s and then grows again. The results make evident that the system is inherently unstable and indicates a confirmation of the previous theoretical discussion, namely that the two-fluid solver with a drag model only lacks hyperbolicity. Furthermore, the oscillating behavior of  $\Phi$  proposes that the observed instability has a numerical character, at least in the vicinity of the outbreaks.

The results with the virtual mass force show different characteristics in terms of both  $\alpha$  and  $\Phi$ . Whereas an initial growth in  $\Phi$  is seen, which is qualitatively similar to the case without the virtual mass force, no regrowth is observed after the maximally unstable state. Instead a different stage is entered after approximately 100 simulated seconds, where the fields exhibit a more smooth variation. The second stage is from a qualitative point of view considered as more physical in its appearance due to its smooth variation the extracted data, which is further strengthened by Fig. 2 as discussed below.

As regards the described drift in the velocity, primarily for the case without the virtual mass (Fig. 1a, bottom), we note that the largest disturbances in the extracted velocity data occur concurrently with the ruptures in the void fraction and the time-resolved uniformity index. The change in the baseline velocity is thus interpreted as a result of the unstable state, where the velocity increases. Since the system is driven by an axially fixed pressure jump, there is no explicit mechanism to drive the system back to the initial zero liquid velocity. On the other hand, for the case including the virtual mass (Fig. 1b, bottom), an on average constant velocity is seen.



**Fig. 3.** Cartesian components of the virtual mass force (top) and the drag force (bottom).

Complementary to Fig. 1, a visualization of the void fraction field is presented in Fig. 2. For the cases without virtual mass, corresponding to the two upper rows in Fig. 2, we see that for example at  $t = 200$  s a striped, non-physical, state coincides with a sudden eruption in  $\Phi$ . Again, this points to an instability of a numerical type, i.e. lack of hyperbolicity, rather than a physical fluctuation. It should be noted that the width of the stripes correspond to a single cell width. The same conclusion is drawn from the horizontal slice, square figures in the second row, where again a striped pattern of a single computational cell is seen.

In contrast, the results for the case with the virtual mass force, the bottom two rows in Fig. 2, an initial striped pattern (up to  $t = 80$  s) is followed by a physically looking instability, larger than the cell size but smaller than the vertical and horizontal size of the domain. Figs. 1 and 2 combined confirm a process in two stages; an initial instability of a numerical character is followed by a meso-scale instability that persists. We also emphasize the significant difference in the dynamics between the cases, where the case with virtual mass exhibits a smoother evolution of all fields.

#### 4.2. Momentum interface transfer

In Fig. 3, we present a comparison between the magnitude of the virtual mass and drag forces. The analysis reveals that the axial component ( $y$ -direction) of the virtual mass (Fig. 3a, middle) is small in comparison to the drag (Fig. 3b, middle). For the other components the forces are on the same order of magnitude. When compared to the time-resolved uniformity index for the same case (Fig. 1b, middle), it is seen that the virtual mass force increases in magnitude primarily after  $t = 100$  s. This coincides with the change in void characteristics (Fig. 1b, top) and further emphasizes the presence of a second stage in the simulations.

It is notable that, even though the size of the virtual mass force is small in the early stage of the simulation, it is still profound for the growth of the instability as the dynamics is significantly changed, compare Figs. 1a (middle) and 1b (middle).

#### 4.3. Void comparison

Next we study the difference between different initial, uniform, void fraction levels. The results are presented in Fig. 4. For the cases with the virtual mass force the dynamics of the growth of the instability are changed. The three realizations indicate a slower growth of the heterogeneity for a lower gas fraction. A similar trend is seen for the case without the virtual mass although it is less clear. For the case without the virtual mass force the repeated

bursts in the uniformity index are seen for an initial uniform void fraction of 5% and 10% but not for 1%. For the latter case a single peak over approximately 30 simulated seconds is instead seen, whereafter  $\Phi$  continually decreases. The latter findings indicate that the model is less prone to the instabilities for lower void fractions.

For a more decisive and quantitative analysis of the relative dynamics of the different bubble loadings, a larger set of realizations would be required. Due to a considerable computational burden of such an analysis, we qualitatively conclude that a variation with the void fraction is seen and that for the case with the virtual mass force, the stage of the proposed physical instability is reached for all the presented void fractions.

#### 4.4. Grid refinement study

We perform a mesh refinement study using the two different mesh resolutions defined in Section 3.1. In both cases, the results are very similar in terms of the magnitude of the void fraction fluctuation and the time-resolved uniformity index. A minor change in the dynamics of the growth of the heterogeneity is seen for the case with the virtual mass (Fig. 5b). However, to quantitatively evaluate the differences between the cases a larger number of realizations of the transient would be necessary.

The results suggest that the refinement has not induced any significant change in the nature of the instabilities. Arguably this means that potentially smaller scale variations in the case of the finer grid do not induce any significant contribution to the meso-scale fluctuations. Although such an argument is only indicatively supported, it is interesting in terms of our quest for a two-fluid model to study the growth rate of heterogeneities.

#### 4.5. On the nature of instabilities

As a final discussion we consider the nature of the presented instabilities and the resulting heterogeneities in the flow; are the emerging meso-scale structures a correct representation of physical fluctuations or a mere result of numerical artifacts? From the results of the second stage of the cases with the virtual mass force, we see a qualitative behavior which has a physical character with continuous phase fractions and velocities and with the persisting heterogeneities. On the other hand, we note that all such results are triggered by the artificial state, with a clearly unphysical character. We are thus not able to conclude that the actual physics is recovered as the initial conditions for the more physically appearing state are not realistic.

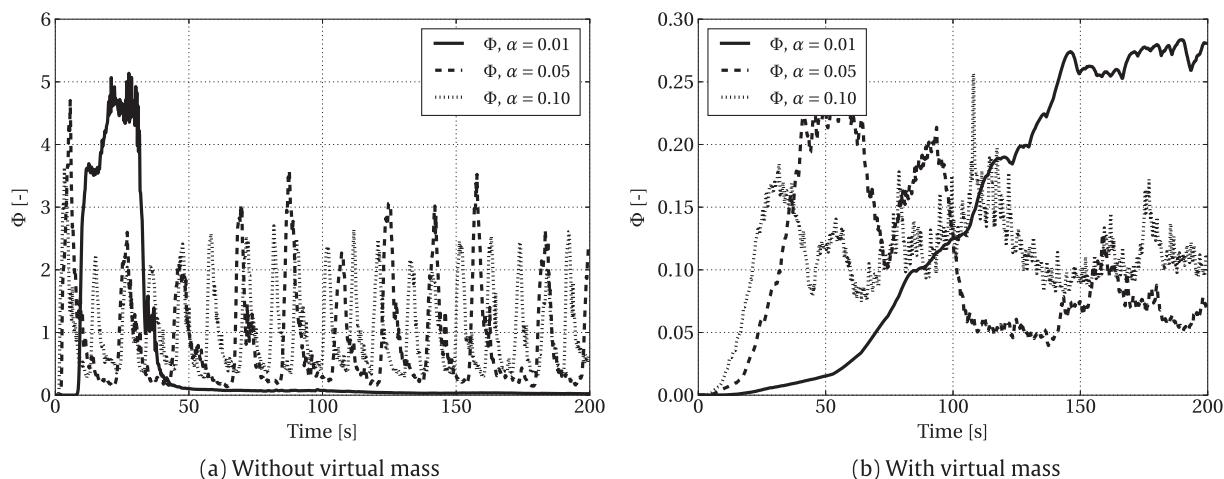
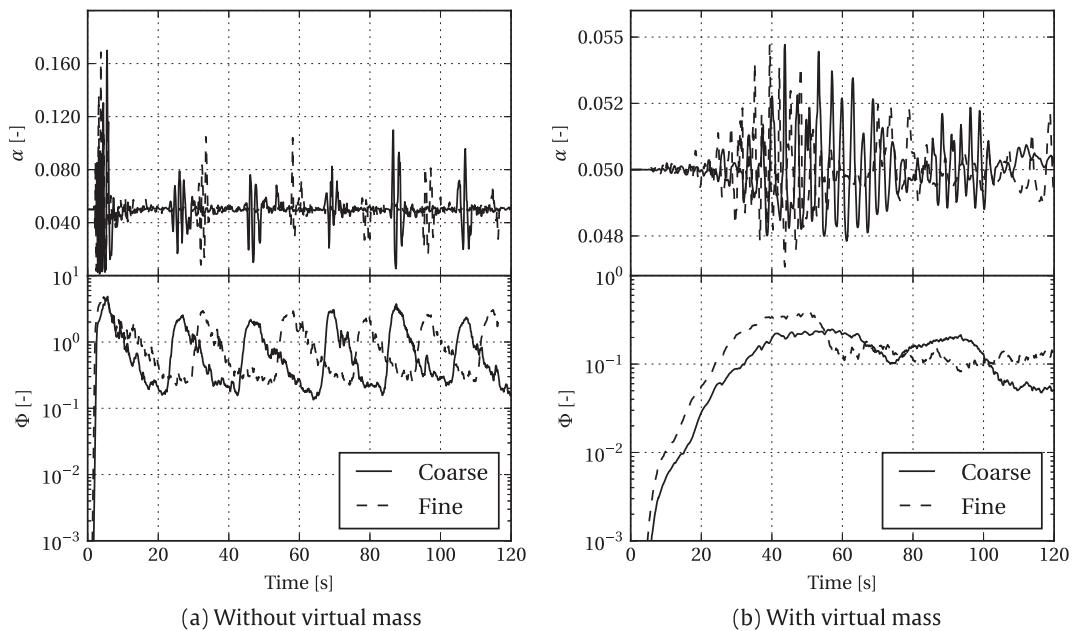


Fig. 4. Temporal development of the time-resolved uniformity index for three different initial, uniform, bubble loadings.



**Fig. 5.** Temporal development of the void fraction field (top) and the uniformity index (bottom) for two different mesh resolutions with the initial condition  $\alpha = 0.05$ .

From an equation point of view, the flat initial distributions of all fields cannot generate any fluctuations as all spatial gradients are zero. However, as discussed, a small round-off error or small residual in the fields is enough to trigger the instability. Interestingly, there is no clear mechanism to drive the system back to a homogeneous state (in principle due to the lack of mass diffusion in the phase fraction equations, Eq. (2)). Still, the finite and discrete representation of the spatial operators will in practice tend to smear the solution. However, as is clear from the re-appearing numerical instabilities for the case with drag only (Fig. 1a, top) the emergence of more physical meso-scale heterogeneities is in itself not a cure for the underlying ill-posedness. The relaxation towards a more flat distribution in space (i.e.  $\Phi$  decreasing) only reaches a certain point, whereafter the system bounces back to a more heterogeneous distribution via the numerically artificial state.

It is also notable that for our reduced model description, based on the drag and virtual mass forces only and without an excessive viscosity from long time-steps or a RANS model, the instabilities are clearly visible. In contrast, a model with enough viscosity contribution from forces or turbulence models would potentially hide the instabilities. In line with this, an additional number of forces to fit experiments can never be taken directly as an example of a correct capture of the physics, but rather as a mean to hide the underlying instability or to suppress it just long enough that a resemblance to the behavior observed in experiments may be obtained. We argue that the two-fluid model issues can be suppressed and tuned, but never fully cured by additional momentum exchanges. This argument is supported by the fact that momentum exchange terms derived strictly based on mathematical arguments (i.e. to ensure hyperbolicity at all conditions) without additional suppression of the inherent instabilities through overly diffusive effects are known to create problems with physically inconsistent model predictions (Vazquez-Gonzalez et al., 2016).

Finally, we argue that the above presented reasoning of the validity of the presented data leads to two possible scenarios. The first scenario is that we deem all the results unphysical due to the numerical trigger of the presented heterogeneous state. Such a standpoint would imply that the two-fluid model structure is inherently dangerous and cannot be trusted. This conclusion would

have far-reaching consequences for all current and future research concerned with averaged descriptions of bubbly gas-liquid flows, as it would imply that modelers should always have to prove the well-posedness of their two-fluid model before any results are analyzed. Alternatively, we might argue that the second stage of the above presented data has a valid physical character, irrespective of the way in which it was triggered. Such a standpoint would however inevitably lead to challenges in the understanding of the results for studies of the stability of bubbly flows with this type of model. It would be necessary to find a solid theoretical basis for judging when the influence of the unphysical trigger has decayed enough. We propose that, in line with the first scenario, due to the unphysical trigger the quantitative values cannot be directly trusted, although we interpret the fluctuations in the second stage as physical in their character. In this view, a shared-pressure two-fluid model can be used to investigate the qualitative dynamics of instabilities in bubbly flows, although it is still an open question whether this type of model will ever allow for a fully quantitative investigation.

## 5. Conclusions

We have studied the emergence and existence of heterogeneities due to instabilities in bubbly gas-liquid flows simulated with a two-fluid model. We have shown that the inclusion of the virtual mass force significantly changes the growth rate of the void heterogeneities, even though the magnitude of that force relative to the drag force is initially small which proposes a change in the characteristics of the equations. After an initial growth of heterogeneities, associated with the numerical instability of the model, a transition is revealed to a different gas fraction behavior of a more physical character. For the cases including the virtual mass force, the smoother evolution and a physically appearing character is associated with a change in the hyperbolicity of the problem.

We have shown that the variation of the initial gas fraction results in a change in the dynamics of the growth of the heterogeneities and that the mesh resolution study indicates that the revealed meso-scale are in character not dependent on the computational grid. The latter suggests that it is possible to reach a certain mesh independence for the formulated equations, although

the underlying two-phase flow does not at all exhibit a separation of scales.

We interpret the demonstrated fluctuations, in particular after the transitions from the initially unstable gas fraction behavior, as physical in their character and thus of fundamental importance for an accurate description of the two-phase flow. However, the fluctuations are triggered by a numerically artificial state and we propose that the quantitative values can thus not be directly trusted. We have emphasized that the effect of a turbulence model entirely based on a viscosity contribution is prone to, unphysically, dampen the presented fluctuations. We conclude that the suggested change in hyperbolicity has an important impact on the physics of the two-phase flow based on the 3D two-fluid model. Furthermore we propose that a model with a large number of momentum exchange forces cannot be a proof of correct physics, but merely an example of a model which hides the underlying instability of the equations.

## Acknowledgements

This research was conducted with funding from the Swedish Research Council (Vetenskapsrådet) as a part of the Development of Revolutionary and Accurate Methods for Safety Analyses of Future and Existing Reactors (DREAM4SAFER) framework grant (contract number C0467701). The computations were performed on resources at Chalmers Centre for Computational Science and Engineering (C3SE) provided by the Swedish National Infrastructure for Computing (SNIC). The Swedish Center for Nuclear Technology (SKC) is acknowledged for financially supporting K. Jareteg.

## References

Agrawal, K., Loezos, P., Syamlal, M., Sundaresan, S., 2001. The role of meso-scale structures in rapid gas-solid flows. *J. Fluid Mech.* 445, 151–185.

Ali, R., Jamel, C., Ghazi, B., Line, A., 2011. Gas dispersion in air-lift reactors: contribution of the turbulent part of the added mass force. *AIChE J.* 57 (12), 3315–3330.

Arai, M., 1980. Characteristics and stability analyses for two-phase flow equation systems with viscous terms. *Nucl. Sci. Eng.* 74 (2), 77–83.

Becker, S., Sokolichin, A., Eigenberger, G., 1994. Gas-liquid flow in bubble columns and loop reactors: Part II. Comparison of detailed experiments and flow simulations. *Chem. Eng. Sci.* 49 (24), 5747–5762.

Bel Fdila, R., Simonin, O., 1992. Eulerian prediction of a turbulent bubbly flow downstream of a sudden pipe expansion. In: 6th Workshop on TwoPhase Flow Predictions.

Bertodano, L., 1998. Two-fluid model for two-phase turbulent jets. *Nucl. Eng. Des.* 179 (1), 65–74.

Boris, J., Grinstein, F., Oran, E., Kolbe, R., 1992. New insights into large eddy simulation. *Fluid Dyn. Res.* 10 (4), 199–228.

Buffo, A., Vanni, M., Marchisio, D., Fox, R., 2013. Multivariate quadrature-based moments methods for turbulent polydisperse gas-liquid systems. *Int. J. Multiphase Flow* 50, 41–57.

Chen, J., Yang, N., Ge, W., Li, J., 2009. Modeling of regime transition in bubble columns with stability condition. *Ind. Eng. Chem. Res.* 48, 290–301.

Chung, M.S., Chang, K.S., Lee, S.J., 2000. Wave propagation in two-phase flow based on a new hyperbolic two-fluid model. *Numer. Heat Transfer: Part A: Appl.* 38 (2), 169–191.

Crowe, C., Sommerfeld, M., Tsuji, Y., 1998. *Multiphase Flows with Particles and Droplets*.

Dinh, T., Nourgaliev, R., Theofanous, T., 2003. Understanding the ill-posed two-fluid model. In: Proceedings of the 10th International Topical Meeting on Nuclear Reactor Thermal-Hydraulics (NURETH03).

Drew, D., Cheng, L., Lahey, R., 1979. The analysis of virtual mass effects in two-phase flow. *Int. J. Multiphase Flow* 5 (4), 233–242.

Espinosa-Paredes, G., 2012. A derivation of the nonlocal volume-averaged equations for two-phase flow transport. *Sci. Technol. Nucl. Install.* 2012. Article ID 890815.

Harteveld, W., 2005. *Bubble Columns: Structures or Stability?* PhD. thesis. Delft Univ. of Technology, The Netherlands.

Hirt, C.W., Nichols, B.D., 1981. Volume of fluid (VOF) method for the dynamics of free boundaries. *J. Comput. Phys.* 39 (1), 201–225.

Ishii, M., 1990. Two-fluid model for two-phase flow. *Multiphase Sci. Technol.* 5 (1–4).

Ishii, M., Hibiki, T., 2010. *Thermo-Fluid Dynamics of Two-Phase Flow*. Springer Science & Business Media.

Issa, R., 1986. Solution of the implicitly discretised fluid flow equations by operator-splitting. *J. Comput. Phys.* 62 (1), 40–65.

Jakobsen, H.A., 2008. *Chemical Reactor Modeling. Multiphase Reactive Flows*. Springer-Verlag, Berlin, Germany.

Jareteg, K., Ström, H., Sasic, S., Demazière, C., 2015. Numerical investigation of instabilities in the two-fluid model for CFD simulations of LWRs. In: M&C 2015, Nashville, Tennessee.

Lahey, R., Cheng, L., Drew, D., Flaherty, J., 1980. The effect of virtual mass on the numerical stability of accelerating two-phase flows. *Int. J. Multiphase Flow* 6 (4), 281–294.

Lahey, R., Drew, D., 1989. The three-dimensional time and volume averaged conservation equations of two-phase flow. *Adv. Nucl. Sci. Technol.* 20, 1–69.

Lee, S.J., Chang, K.S., Kim, K., 1998. Pressure wave speeds from the characteristics of two fluids, two-phase hyperbolic equation system. *Int. J. Multiphase Flow* 24 (5), 855–866.

León-Becerril, E., Cockx, A., Liné, A., 2002. Effect of bubble deformation on stability and mixing in bubble columns. *Chem. Eng. Sci.* 57 (16), 3283–3297.

Lhuillier, D., Chang, C.H., Theofanous, T.G., 2013. On the quest for a hyperbolic effective-field model of disperse flows. *J. Fluid Mech.* 731, 184–194.

Lhuillier, D., Theofanous, T., Liou, M.S., 2010. *Handbook of Nuclear Engineering*. Springer Science+Business Media LLC. In: Cacuci, Dan Gabriel (Ed.). Springer US, Boston, MA, pp. 1813–1912.

Lo, S., 1996. Application of the MUSIG Model to Bubbly Flows. AEAT-1096, AEA Technology.

Lucas, D., Krepper, E., Prasser, H.M., Manera, A., 2006. Investigations on the stability of the flow characteristics in a bubble column. *Chem. Eng. Technol.* 29 (9), 1066–1072.

Lucas, D., Prasser, H.M., Manera, A., 2005. Influence of the lift force on the stability of a bubble column. *Chem. Eng. Sci.* 60 (13), 3609–3619.

Ma, T., Ziegenhein, T., Lucas, D., Krepper, E., Fröhlich, J., 2015. Eulereuler large eddy simulations for dispersed turbulent bubbly flows. *Int. J. Heat Fluid Flow* 56, 51–59.

Marchisio, D., Fox, R., 2005. Solution of population balance equations using the direct quadrature method of moments. *Aerosol Sci.* 36, 43–73.

Mokeyev, Y.G., 1977. Effect of particle concentration on their drag and induced mass. *Fluid Mech. Sov. Res.* 6, 161.

Monahan, S.M., Fox, R.O., 2007a. Effect of model formulation on flow-regime predictions for bubble columns. *AIChE J.* 53 (1), 9–18.

Monahan, S.M., Fox, R.O., 2007b. Linear stability analysis of a two-fluid model for air-water bubble columns. *Chem. Eng. Sci.* 62 (12), 3159–3177.

Monahan, S.M., Vitankar, V.S., Fox, R.O., 2005. CFD predictions for flow-regime transitions in bubble columns. *AIChE J.* 51 (7), 1897–1923.

Mudde, R., Harteveld, W., Van den Akker, H., 2008. Uniform flow in bubble columns. *Ind. Eng. Chem. Res.* 48 (1), 148–158.

Mudde, R., Simonin, O., 1999. Two- and three-dimensional simulations of a bubble plume using a two-fluid model. *Chem. Eng. Sci.* 54, 5061–5069.

Munkejord, S.T., 2010. A numerical study of two-fluid models with pressure and velocity relaxation. *Adv. Appl. Math. Mech.* 2, 131–159.

Nigmatulin, R., 1979. Spatial averaging in the mechanics of heterogeneous and dispersed systems. *Int. J. Multiphase Flow* 5 (5), 353–385.

Noh, W.F., Woodward, P., 1976. SLIC (simple line interface calculation). In: *Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics June 28–July 2, 1976 Twente University, Enschede*. Springer, pp. 330–340.

Oey, R., Mudde, R., Van den Akker, H., 2003. Sensitivity study on interfacial closure laws in two-fluid bubbly flow simulations. *AIChE J.* 49 (7), 1621–1636.

Ojima, S., Hayashi, K., Hosokawa, S., Tomiyama, A., 2014a. Distributions of void fraction and liquid velocity in air–water bubble column. *Int. J. Multiphase Flow* 67, 111–121.

Ojima, S., Hayashi, K., Tomiyama, A., 2014b. Effects of hydrophilic particles on bubbly flow in slurry bubble column. *Int. J. Multiphase Flow* 58, 154–167.

Osher, S., Sethian, J.A., 1988. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.* 79 (1), 12–49.

Pauchon, C., Banerjee, S., 1986. Interphase momentum interaction effects in the averaged multifield model: Part I: Void propagation in bubbly flows. *Int. J. Multiphase Flow* 12 (4), 559–573.

Pokharkar, H., Mori, M., Ransom, V.H., 1997. Regularization of two-phase flow models: a comparison of numerical and differential approaches. *J. Comput. Phys.* 134 (2), 282–295.

Prosperetti, A., 2003. Two-fluid modelling and averaged equations. *Multiphase Sci. Technol.* 15 (1–4).

Prosperetti, A., Satrape, J.V., 1990. Stability of two-phase flow models. In: *Two Phase Flows and Waves*. Springer, pp. 98–117.

Prosperetti, A., Tryggvason, G., 2007. *Computational Methods for Multiphase Flow*. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA.

Ransom, V.H., Hicks, D.L., 1984. Hyperbolic two-pressure models for two-phase flow. *J. Comput. Phys.* 53 (1), 124–151.

Rusche, H., 2002. *Computational Fluid Dynamics of Dispersed Two-Phase Flows at High Phase Fractions*. Ph.D. thesis. Imperial College of Science, Technology & Medicine.

Rzehak, R., Krepper, E., 2013. CFD modeling of bubble-induced turbulence. *Int. J. Multiphase Flow* 55, 138–155.

Sankaranarayanan, K., Shan, X., Kevrekidis, I., Sundaresan, S., 2002. Analysis of drag and virtual mass forces in bubbly suspensions using an implicit formulation of the lattice Boltzmann method. *J. Fluid Mech.* 452, 61–96.

Sankaranarayanan, K., Sundaresan, S., 2002. Lift force in bubbly suspensions. *Chem. Eng. Sci.* 57 (17), 3521–3542.

Schiller, L., Naumann, Z., 1935. A drag coefficient correlation. *Z. Ver. Deutsch. Ing.* 77, 318–320.

Selma, B., Bannari, R., Proulx, P., 2010. Simulation of bubbly flows: Comparison between direct quadrature method of moments (DQMOM) and method of classes (CM). *Chem. Eng. Sci.* 65 (6), 1925–1941.

Spelt, P.D., Sangani, A.S., 1997. Properties and averaged equations for flows of bubbly liquids. *Appl. Sci. Res.* 58 (1–4), 337–386.

Stewart, H.B., 1979. Stability of two-phase flow calculation using two-fluid models. *J. Comput. Phys.* 33 (2), 259–270.

Strööm, H., Sasic, S., Jareteg, K., Demazière, C., 2015. Behaviour and stability of the two-fluid model for fine-scale simulations of bubbly flow in nuclear reactors. *Int. J. Chem. React. Eng.* 13 (4), 449–459.

Sussman, M., Smereka, P., Osher, S., 1994. A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys.* 114 (1), 146–159.

Tomiyama, A., Shimada, N., 2001. A numerical method for bubbly flow simulation based on a multi-fluid model. *J. Press. Vess. Technol.* 123 (4), 510–516.

Toumi, I., Kumbaro, A., 1996. An approximate linearized riemann solver for a two-fluid model. *J. Comput. Phys.* 124 (2), 286–300.

Unverdi, S., Tryggvason, G., 1992. A front-tracking method for viscous, incompressible, multi-fluid flows. *J. Comput. Phys.* 100 (1), 25–37.

Van Wijngaarden, L., Jeffrey, D., 1976. Hydrodynamic interaction between gas bubbles in liquid. *J. Fluid Mech.* 77 (01), 27–44.

Vazquez-Gonzalez, T., Llor, A., Fochesato, C., 2016. Ransom test results from various two-fluid schemes: Is enforcing hyperbolicity a thermodynamically consistent option? *Int. J. Multiphase Flow* 81, 104–112.

Weller, H., 2005. Derivation, Modeling and Solution of the Conditionally Averaged Two-Phase Flow Equations Technical Report. OpenCFD.

Weller, H.G., Tabor, G., Jasak, H., Fureby, C., 1998. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Comput. Phys.* 12 (6), 620–631.

Yang, N., Chen, J., Ge, W., Li, J., 2010. A conceptual model for analyzing the stability condition and regime transition in bubble columns. *Chem. Eng. Sci.* 65 (1), 517–526.

Yang, N., Chen, J., Zhao, H., Ge, W., Li, J., 2007. Explorations on the multi-scale flow structure and stability condition in bubble columns. *Chem. Eng. Sci.* 62 (24), 6978–6991.

Zhang, D., Prosperetti, A., 1997. Momentum and energy equations for disperse two-phase flows and their closure for dilute suspensions. *Int. J. Multiphase Flow* 23 (3), 425–453.

Zuber, N., 1964. On the dispersed two-phase flow in the laminar flow regime. *Chem. Eng. Sci.* 19 (11), 897–917.



## **Paper VIII**

*“Development of a point-kinetic verification scheme for nuclear reactor applications”*

C. Demaziére, V. Dykin, K. Jareteg

*Journal of Computational Physics 339 (2017), pp. 396-411*





# Development of a point-kinetic verification scheme for nuclear reactor applications



C. Demazière <sup>\*</sup>, V. Dykin, K. Jareteg

Division of Subatomic and Plasma Physics, Department of Physics, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden

## ARTICLE INFO

### Article history:

Received 11 November 2016

Received in revised form 1 March 2017

Accepted 10 March 2017

Available online 15 March 2017

### Keywords:

Computational verification and validation

Time-dependent neutron transport

Frequency-dependent neutron transport

Diffusion theory

Neutron fluctuations

Neutron noise

Point-kinetics

## ABSTRACT

In this paper, a new method that can be used for checking the proper implementation of time- or frequency-dependent neutron transport models and for verifying their ability to recover some basic reactor physics properties is proposed. This method makes use of the application of a stationary perturbation to the system at a given frequency and extraction of the point-kinetic component of the system response. Even for strongly heterogeneous systems for which an analytical solution does not exist, the point-kinetic component follows, as a function of frequency, a simple analytical form. The comparison between the extracted point-kinetic component and its expected analytical form provides an opportunity to verify and validate neutron transport solvers. The proposed method is tested on two diffusion-based codes, one working in the time domain and the other working in the frequency domain. As long as the applied perturbation has a non-zero reactivity effect, it is demonstrated that the method can be successfully applied to verify and validate time- or frequency-dependent neutron transport solvers. Although the method is demonstrated in the present paper in a diffusion theory framework, higher order neutron transport methods could be verified based on the same principles.

© 2017 Elsevier Inc. All rights reserved.

## 1. Introduction

The modelling of nuclear power plants relies on complex codes and models capable of resolving the interdependence between several fields of physics. This is particularly true when modelling nuclear reactor cores where a tight coupling between the transport of neutrons, fluid dynamics, and heat transfer exists [1]. The codes used by the nuclear industry, before getting licensed by the safety authorities, must go through a lengthy process of verification and validation. Using common terminology in computer simulations, the verification process aims at verifying the proper implementation of a given mathematical model whereas the validation process targets at demonstrating the correctness of the chosen mathematical model to represent the actual physics [2–4].

The verification of computer models is most often performed by comparing the results of the simulations to analytical or semi-analytical solutions [5–7]. Being able to obtain analytical or semi-analytical solutions usually imposes to drastically reduce the actual complexity of the system. This could require considering homogeneous systems or piece-wise homogeneous systems. The validation, on the other hand, could be carried out by assessing the accuracy of the simulations versus either experimental data or results of other already qualified computer models [8–10].

<sup>\*</sup> Corresponding author.

E-mail address: [demaz@chalmers.se](mailto:demaz@chalmers.se) (C. Demazière).

The behaviour of neutrons inside a nuclear core is governed by the neutron transport equation, which reads, using standard notations, as [11]:

$$\begin{aligned} & \frac{1}{v(E)} \frac{\partial}{\partial t} \psi(\mathbf{r}, \bar{\Omega}, E, t) + \bar{\Omega} \cdot \bar{\nabla} \psi(\mathbf{r}, \bar{\Omega}, E, t) + \Sigma_T(\mathbf{r}, E, t) \psi(\mathbf{r}, \bar{\Omega}, E, t) \\ &= \int_0^{\infty} \int_{(4\pi)} \Sigma_s(\mathbf{r}, \bar{\Omega}' \rightarrow \bar{\Omega}, E' \rightarrow E, t) \psi(\mathbf{r}, \bar{\Omega}', E', t) d\Omega' dE' \\ &+ \frac{(1-\beta)\chi^p(E)}{4\pi} \int_0^{\infty} v(E') \Sigma_f(\mathbf{r}, E', t) \phi(\mathbf{r}, E', t) dE' + \chi^d(E) \lambda C(\mathbf{r}, t) \end{aligned} \quad (1)$$

and

$$\frac{\partial C}{\partial t}(\mathbf{r}, t) = \beta \int_0^{\infty} v(E') \Sigma_f(\mathbf{r}, E', t) \phi(\mathbf{r}, E', t) dE' - \lambda C(\mathbf{r}, t) \quad (2)$$

where for the sake of simplicity, one group of precursors  $C(\mathbf{r}, t)$  of delayed neutrons was assumed. Because of their integro-differential nature, the fact that the neutron angular and scalar fluxes depend on many variables, and that the systems usually modelled are large and strongly heterogeneous, the equations above cannot be used for solving large scale engineering problems, as the ones encountered when considering commercial nuclear reactors. Simplifications are thus introduced in the following manner<sup>1</sup> [1,11]:

- Eq. (1) is integrated on all solid angles.
- Isotropic scattering in the laboratory system is assumed and a transport correction of the scattering macroscopic cross-sections is introduced to recover some level of anisotropy in the scattering.
- Proportionality between the neutron current density vector and the gradient of the scalar neutron flux is considered (Fick's law).

The resulting equations, also known as the diffusion equations, can then be written as:

$$\begin{aligned} & \frac{1}{v(E)} \frac{\partial \phi}{\partial t}(\mathbf{r}, E, t) = \nabla \cdot [D(\mathbf{r}, E, t) \nabla \phi(\mathbf{r}, E, t)] - \Sigma_T(\mathbf{r}, E, t) \phi(\mathbf{r}, E, t) + \int_0^{\infty} \Sigma_{s0}(\mathbf{r}, E' \rightarrow E, t) \phi(\mathbf{r}, E', t) dE' \\ &+ \chi^p(E)(1-\beta) \int_0^{\infty} v(E') \Sigma_f(\mathbf{r}, E', t) \phi(\mathbf{r}, E', t) dE' + \chi^d(E) \lambda C(\mathbf{r}, t) \end{aligned} \quad (3)$$

and

$$\frac{\partial C}{\partial t}(\mathbf{r}, t) = \beta \int_0^{\infty} v(E') \Sigma_f(\mathbf{r}, E', t) \phi(\mathbf{r}, E', t) dE' - \lambda C(\mathbf{r}, t) \quad (4)$$

Finally, the dependence on energy appearing in the above equations is handled by integrating the different quantities with respect to the energy on small energy bins, referred to as energy groups. This results in the following equations:

$$\begin{aligned} & \frac{1}{v_g} \frac{\partial \phi_g}{\partial t}(\mathbf{r}, t) = \nabla \cdot [D_g(\mathbf{r}, t) \nabla \phi_g(\mathbf{r}, t)] - \Sigma_{T,g}(\mathbf{r}, t) \phi_g(\mathbf{r}, t) + \sum_{g'} \Sigma_{s0,g' \rightarrow g}(\mathbf{r}, t) \phi_{g'}(\mathbf{r}, t) \\ &+ \chi_g^p(1-\beta) \sum_{g'} v_{g'} \Sigma_{f,g'}(\mathbf{r}, t) \phi_{g'}(\mathbf{r}, t) + \chi_g^d \lambda C(\mathbf{r}, t) \end{aligned} \quad (5)$$

and

$$\frac{\partial C}{\partial t}(\mathbf{r}, t) = \beta \sum_{g'} v_{g'} \Sigma_{f,g'}(\mathbf{r}, t) \phi_{g'}(\mathbf{r}, t) - \lambda C(\mathbf{r}, t) \quad (6)$$

<sup>1</sup> The diffusion approximation can be more rigorously introduced by first deriving the  $P_1$  approximation of the neutron transport equation, from which diffusion theory is obtained. The interested reader is referred to e.g. Refs. [1,11].

where the indexes  $g$  and  $g'$  refer to the energy groups. Most codes used by the industry for modelling time-dependent neutron transport rely on the diffusion approximation, i.e. Eqs. (3) and (4), rather than on the exact formulation in pure transport given by Eqs. (1) and (2). The transformation of the above equations from a continuous energy representation to a discrete energy representation must be carried out while preserving the actual reaction rates. For the sake of completeness, it should also be noted that the above equations are seldom solved for the actual geometry of the system. Rather, a process of local volume averaging is introduced to reduce the complexity of the problem. As for the averaging with respect to energy, the volume averaging must preserve the reaction rates. Obtained energy- and volume-averaged data in the above equations represents a tedious and complex task, which should be carried out in such a way that the reaction rates are preserved during the averaging. This requires the application of different sequential modelling steps, first at the fuel cell level and then at the fuel assembly level. Such steps aim at obtaining a good enough guess of the solution with a high resolution in space, angle, and energy but on a small part of the computational domain. This solution is thereafter used to average the data on a coarser grid with respect to space, angle, and energy. Because the lower resolution of the data appearing in the balance equations, the entire computational domain can be considered. This new set of data is then used to resolve the actual interdependence between spatial regions that was not considered at the previous modelling steps. The description of the details of such engineering methodologies is beyond the scope of this paper. We refer instead to Ref. [1] for the interested reader.

In addition to the process of averaging, the data appearing in Eqs. (5) and (6) strongly depend on the local instantaneous conditions in the core (such as fuel temperature, control rod positions, soluble poison concentration and coolant density, among others) as well as on their local histories (burnup inclusive). Such conditions cannot be determined when only a small part of the computational domain is modelled while generating the space-, angle-, and energy-averaged data. Such data are thus estimated for a large range of possible values of the instantaneous and history variables determined in advance. When the full core solution is being determined, the value of such variables is determined on-the-fly either as user-defined boundary conditions or internally calculated by the code using adequate modules (thermal-hydraulic module for estimating the fuel temperature and coolant density, boron transport module for estimating the concentration of the soluble poison, etc.).

The simulations of commercial nuclear reactors from a neutron transport viewpoint thus represent a far from trivial task, both from a modelling perspective and from an analyst (i.e. code user) perspective. There is thus a strong incentive in developing techniques that allow to verify that a core simulator behaves as expected.

In the present paper, we propose an innovative method to both verify and validate time- or frequency-dependent neutron transport solvers based on some basic reactor physics properties the solution to given problems should retain. It relies on the estimation of the response of a reactor core to stationary fluctuations. Such a response is made of two parts: the point-kinetic component and the deviation from point-kinetics (also referred to as the fluctuations of the shape function). As is well known in reactor dynamics, the point-kinetic component has a simple analytical expression as a function of the frequency of the perturbations imposed to the system [11]. Such a simple analytical expression always exists even for strongly heterogeneous systems. As will be explained in the remaining of this paper, there is nevertheless a pathological case where under given conditions the point-kinetic component might be identically equal to zero due to an applied zero reactivity perturbation and the methodology presented in this paper thus cannot be followed. Extracting from the results of computer simulations the point-kinetic component and comparing it to its analytical expression thus allows checking both the proper implementation of the neutron transport models and verifying that the physics is adequately modelled. In this respect, the proposed method falls in the category of both code verification and validation.

To the knowledge of the present authors, extracting the point-kinetic component of a reactor system response possibly significantly deviating from point-kinetics and comparing the frequency-dependence of the extracted response to its analytical expression was not earlier proposed as a method for code verification and validation. Even if heterogeneous systems behave in a point-kinetic manner in only a few very specific cases, the extraction of the point-kinetic component is always possible (apart from the pathological case mentioned above).

The paper is structured as follows. First, the derivation of the point-kinetic component in a diffusion theory framework will be recalled and the analytical expression of such a component derived. Thereafter, to better illustrate the powerfulness of the proposed method, the most challenging cases will be identified. This will be done by considering a homogeneous system, for which the exact solution to the problem can be estimated in a semi-analytical manner and compared to its point-kinetic component. Cases where the system response significantly deviates from point-kinetics will be used as the basis of computer simulations performed for strongly heterogeneous systems. Two in-house codes, one relying on a time-domain approach and the other on the frequency-domain, will be put on the test bench and will demonstrate the advantages of the proposed method.

## 2. Point-kinetic approximation

### 2.1. Derivation of the point-kinetic approximation

Although the point-kinetic approximation can be derived directly from the neutron transport equation, we present its derivation hereafter in diffusion theory and for 2 neutron energy groups, which represent the typical modelling framework

used by most time- or frequency-dependent core simulators [12–14]. In such a framework, the governing balance equations read as [11]:

$$\begin{aligned} \frac{1}{v_1} \frac{\partial \phi_1}{\partial t}(\mathbf{r}, t) &= \nabla \cdot [D_1(\mathbf{r}) \nabla \phi_1(\mathbf{r}, t)] + \nu \Sigma_{f,2}(\mathbf{r}, t)(1 - \beta) \phi_2(\mathbf{r}, t) \\ &+ \lambda C(\mathbf{r}, t) + [\nu \Sigma_{f,1}(\mathbf{r}, t)(1 - \beta) - \Sigma_{a,1}(\mathbf{r}, t) - \Sigma_{rem}(\mathbf{r}, t)] \phi_1(\mathbf{r}, t) \end{aligned} \quad (7)$$

$$\frac{1}{v_2} \frac{\partial \phi_2}{\partial t}(\mathbf{r}, t) = \nabla \cdot [D_2(\mathbf{r}) \nabla \phi_2(\mathbf{r}, t)] - \Sigma_{a,2}(\mathbf{r}, t) \phi_2(\mathbf{r}, t) + \Sigma_{rem}(\mathbf{r}, t) \phi_1(\mathbf{r}, t) \quad (8)$$

and

$$\frac{\partial C}{\partial t}(\mathbf{r}, t) = \beta [\nu \Sigma_{f,1}(\mathbf{r}, t) \phi_1(\mathbf{r}, t) + \nu \Sigma_{f,2}(\mathbf{r}, t) \phi_2(\mathbf{r}, t)] - \lambda C(\mathbf{r}, t) \quad (9)$$

Eqs. (7) and (8) represent the time-dependent neutron balance equations in the fast and thermal energy groups, respectively, whereas Eq. (9) expresses the time-dependent balance for the precursors of delayed neutrons. For the sake of simplicity, we also assume that the diffusion coefficients appearing in Eqs. (6) and (7) are time-independent. It was demonstrated in Ref. [15] that neglecting the time-dependence of the diffusion coefficients greatly simplifies the procedure for estimating the time-dependent flux whereas it only introduces a marginal error.

Eqs. (7)–(9) can be more compactly written as:

$$\begin{bmatrix} \frac{1}{v_1} \frac{\partial \phi_1}{\partial t}(\mathbf{r}, t) \\ \frac{1}{v_2} \frac{\partial \phi_2}{\partial t}(\mathbf{r}, t) \end{bmatrix} = [\nabla \cdot \mathbf{D}(\mathbf{r}) \nabla - \mathbf{M}(\mathbf{r}, t) + \mathbf{F}(\mathbf{r}, t)] \times \begin{bmatrix} \phi_1(\mathbf{r}, t) \\ \phi_2(\mathbf{r}, t) \end{bmatrix} + \begin{bmatrix} \lambda C(\mathbf{r}, t) \\ 0 \end{bmatrix} \quad (10)$$

and

$$\frac{\partial C}{\partial t}(\mathbf{r}, t) = \beta [\nu \Sigma_{f,1}(\mathbf{r}, t) \nu \Sigma_{f,2}(\mathbf{r}, t)] \times \begin{bmatrix} \phi_1(\mathbf{r}, t) \\ \phi_2(\mathbf{r}, t) \end{bmatrix} - \lambda C(\mathbf{r}, t) \quad (11)$$

with

$$\mathbf{D}(\mathbf{r}) = \begin{bmatrix} D_1(\mathbf{r}) & 0 \\ 0 & D_2(\mathbf{r}) \end{bmatrix} \quad (12)$$

$$\mathbf{M}(\mathbf{r}, t) = \begin{bmatrix} \Sigma_{a,1}(\mathbf{r}, t) + \Sigma_{rem}(\mathbf{r}, t) & 0 \\ -\Sigma_{rem}(\mathbf{r}, t) & \Sigma_{a,2}(\mathbf{r}, t) \end{bmatrix} \quad (13)$$

$$\mathbf{F}(\mathbf{r}, t) = \begin{bmatrix} \nu \Sigma_{f,1}(\mathbf{r}, t) & \nu \Sigma_{f,2}(\mathbf{r}, t) \\ 0 & 0 \end{bmatrix} \quad (14)$$

Eqs. (7)–(9) represent the system of equations that reactor core simulators solve, from which the spatial and temporal distribution of the neutron flux and precursors of delayed neutrons can be estimated.

In order to obtain the point-kinetic approximation, all time-dependent terms in the previous equations, generically denoted as  $X(\mathbf{r}, t)$ , are expressed as sums between their mean values  $X_0(\mathbf{r})$  and their fluctuations  $\delta X(\mathbf{r}, t)$  around the mean values (also referred to as “noise”):

$$X(\mathbf{r}, t) = X_0(\mathbf{r}) + \delta X(\mathbf{r}, t) \quad (15)$$

where one supposes that  $\delta X(\mathbf{r}, t) \ll X_0(\mathbf{r})$ . Further, assuming a factorization of the space-dependent neutron flux into an amplitude factor  $P(t)$  and a shape function  $\psi_k(\mathbf{r}, t)$ ,  $k = 1, 2$ , one can write [11]:

$$\begin{bmatrix} \phi_1(\mathbf{r}, t) \\ \phi_2(\mathbf{r}, t) \end{bmatrix} = P(t) \times \begin{bmatrix} \psi_1(\mathbf{r}, t) \\ \psi_2(\mathbf{r}, t) \end{bmatrix} \quad (16)$$

with the following normalization condition:

$$\frac{\partial}{\partial t} \int \left[ \frac{1}{v_1} \phi_{1,0}^+(\mathbf{r}) \psi_1(\mathbf{r}, t) + \frac{1}{v_2} \phi_{2,0}^+(\mathbf{r}) \psi_2(\mathbf{r}, t) \right] d\mathbf{r} = 0 \quad (17)$$

In the previous equation,  $\phi_{k,0}^+(\mathbf{r})$ ,  $k = 1, 2$  is the adjoint eigenfunction, given by the solution of the following equation [11]:

$$[\nabla \cdot \mathbf{D}(\mathbf{r}) \nabla - \mathbf{M}_0^T(\mathbf{r})] \times \begin{bmatrix} \phi_{1,0}^+(\mathbf{r}) \\ \phi_{2,0}^+(\mathbf{r}) \end{bmatrix} = -\frac{1}{k_{eff}} \mathbf{F}_0^T(\mathbf{r}) \times \begin{bmatrix} \phi_{1,0}^+(\mathbf{r}) \\ \phi_{2,0}^+(\mathbf{r}) \end{bmatrix} \quad (18)$$

where the superscript  $T$  denotes the transpose operator.

The generic expression given by Eq. (15), when applied to the neutron flux, leads to:

$$\begin{bmatrix} \phi_1(\mathbf{r}, t) \\ \phi_2(\mathbf{r}, t) \end{bmatrix} = \begin{bmatrix} \phi_{1,0}(\mathbf{r}) \\ \phi_{2,0}(\mathbf{r}) \end{bmatrix} + \begin{bmatrix} \delta\phi_1(\mathbf{r}, t) \\ \delta\phi_2(\mathbf{r}, t) \end{bmatrix} \quad (19)$$

Because of Eq. (16), one can also write that:

$$\begin{cases} P(t) = P_0 + \delta P(t) \\ \begin{bmatrix} \psi_1(\mathbf{r}, t) \\ \psi_2(\mathbf{r}, t) \end{bmatrix} = \frac{1}{P_0} \begin{bmatrix} \phi_{1,0}(\mathbf{r}) \\ \phi_{2,0}(\mathbf{r}) \end{bmatrix} + \begin{bmatrix} \delta\psi_1(\mathbf{r}, t) \\ \delta\psi_2(\mathbf{r}, t) \end{bmatrix} \end{cases} \quad (20)$$

where  $\phi_{k,0}(\mathbf{r})$ ,  $k = 1, 2$  corresponds to the static flux, given by the solution of the following equation:

$$[\nabla \cdot \mathbf{D}(\mathbf{r}) \nabla - \mathbf{M}_0(\mathbf{r})] \times \begin{bmatrix} \phi_{1,0}(\mathbf{r}) \\ \phi_{2,0}(\mathbf{r}) \end{bmatrix} = -\frac{1}{k_{eff}} \mathbf{F}_0(\mathbf{r}) \times \begin{bmatrix} \phi_{1,0}(\mathbf{r}) \\ \phi_{2,0}(\mathbf{r}) \end{bmatrix} \quad (21)$$

The mean value of the shape function in Eq. (20) is determined by the initial condition:

$$\begin{bmatrix} \phi_1(\mathbf{r}, 0) \\ \phi_2(\mathbf{r}, 0) \end{bmatrix} = \begin{bmatrix} \phi_{1,0}(\mathbf{r}) \\ \phi_{2,0}(\mathbf{r}) \end{bmatrix} = P_0 \times \begin{bmatrix} \psi_1(\mathbf{r}, 0) \\ \psi_2(\mathbf{r}, 0) \end{bmatrix} \quad (22)$$

If the deviation from the mean of the perturbed quantity is small compared to the mean, i.e.  $\delta X(\mathbf{r}, t) \ll X_0(\mathbf{r})$ , second-order terms can be neglected in the balance equations. The fluctuations in the neutron flux are thus given by:

$$\begin{bmatrix} \delta\phi_1(\mathbf{r}, t) \\ \delta\phi_2(\mathbf{r}, t) \end{bmatrix} = \begin{bmatrix} \delta\phi_1^{pk}(\mathbf{r}, t) \\ \delta\phi_2^{pk}(\mathbf{r}, t) \end{bmatrix} + P_0 \times \begin{bmatrix} \delta\psi_1(\mathbf{r}, t) \\ \delta\psi_2(\mathbf{r}, t) \end{bmatrix} \quad (23)$$

where the point-kinetic term is given by:

$$\begin{bmatrix} \delta\phi_1^{pk}(\mathbf{r}, t) \\ \delta\phi_2^{pk}(\mathbf{r}, t) \end{bmatrix} = \frac{\delta P(t)}{P_0} \times \begin{bmatrix} \phi_{1,0}(\mathbf{r}) \\ \phi_{2,0}(\mathbf{r}) \end{bmatrix} \quad (24)$$

Obtaining the point-kinetic equations is then carried out in the following manner. First, the dynamic equations (7) and (8) are multiplied by the adjoint eigenfunctions  $\phi_{1,0}^+(\mathbf{r})$  and  $\phi_{2,0}^+(\mathbf{r})$ , respectively. The static adjoint equations given by Eq. (18) are multiplied by the time-dependent neutron flux in its factorized form given by Eq. (16). The two resulting sets of equations are then subtracted from each other. Finally, integrating over the whole reactor volume leads to, after making use of the normalization condition given by Eq. (17):

$$\begin{cases} \frac{dP}{dt}(t) = \frac{\rho(t) - \beta}{\Lambda(t)} P(t) + \lambda c(t) \\ \frac{dc}{dt}(t) = \beta \frac{P(t)}{\Lambda(t)} - \lambda c(t) \end{cases} \quad (25)$$

with

$$\begin{cases} \Lambda(t) = \frac{1}{F(t)} \int \left[ \frac{1}{v_1} \phi_{1,0}^+(\mathbf{r}) \psi_1(\mathbf{r}, t) + \frac{1}{v_2} \phi_{2,0}^+(\mathbf{r}) \psi_2(\mathbf{r}, t) \right] d\mathbf{r} \\ c(t) = \frac{\int \phi_{1,0}^+(\mathbf{r}) C(\mathbf{r}, t) d\mathbf{r}}{\Lambda(t) F(t)} \\ F(t) = \int [\nu \Sigma_{f,1}(\mathbf{r}, t) \phi_{1,0}^+(\mathbf{r}) \psi_1(\mathbf{r}, t) + \nu \Sigma_{f,2}(\mathbf{r}, t) \phi_{2,0}^+(\mathbf{r}) \psi_2(\mathbf{r}, t)] d\mathbf{r} \end{cases} \quad (26)$$

and  $\rho(t)$  is the reactivity, that can be evaluated using first-order perturbation theory as [11]:

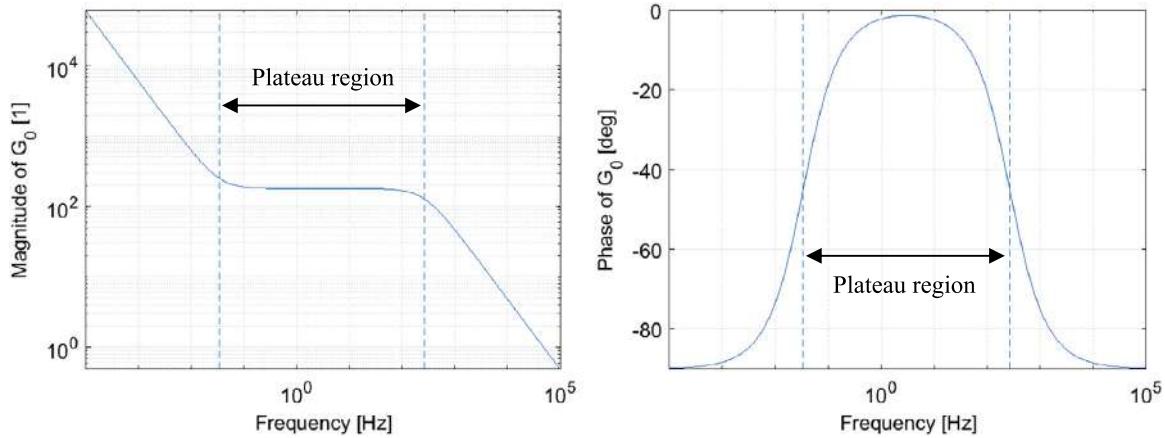
$$\begin{aligned} \rho(t) = & \frac{1}{F(t)} \int \{ [\delta \nu \Sigma_{f,1}(\mathbf{r}, t) - \delta \Sigma_{a,1}(\mathbf{r}, t) - \delta \Sigma_{rem}(\mathbf{r}, t)] \phi_{1,0}^+(\mathbf{r}) \psi_1(\mathbf{r}, t) \\ & + \delta \nu \Sigma_{f,2}(\mathbf{r}, t) \phi_{2,0}^+(\mathbf{r}) \psi_2(\mathbf{r}, t) + \delta \Sigma_{rem}(\mathbf{r}, t) \phi_{2,0}^+(\mathbf{r}) \psi_1(\mathbf{r}, t) - \delta \Sigma_{a,2}(\mathbf{r}, t) \phi_{2,0}^+(\mathbf{r}) \psi_2(\mathbf{r}, t) \} d\mathbf{r} \end{aligned} \quad (27)$$

Eq. (25) simplifies in the static case into:

$$\begin{cases} \rho_0 = 0 \\ \lambda c_0 = \beta \frac{P_0}{\Lambda_0} \end{cases} \quad (28)$$

If one further assumes that  $\delta \Lambda(t) \ll \Lambda_0$ , subtracting Eq. (28) from Eq. (25) and performing a temporal Fourier-transform leads to:

$$\frac{\delta P(\omega)}{P_0} = G_0(\omega) \delta \rho(\omega) \quad (29)$$



**Fig. 1.** Amplitude (left figure) and phase (right figure) of the point-kinetic component of the zero-power reactor transfer function.

where

$$G_0(\omega) = \frac{1}{i\omega(\Lambda_0 + \frac{\beta}{i\omega + \lambda})} \quad (30)$$

where  $i$  is represents the basic imaginary unit. It was verified in the numerical examples presented in Section 3 that the fluctuations in the mean generation time  $\delta\Lambda(t)$  are more than two orders of magnitude smaller than their mean value  $\Lambda_0$ . In the framework of linear theory, the only approximation used in the derivation of point-kinetics is expressed by  $\delta\Lambda(t) \ll \Lambda_0$ , which is thus a valid approximation.

Eq. (30) represents the point-kinetic zero-power reactor transfer function of the critical system. In the literature, this expression is often referred to as the zero-power or open-loop reactor transfer function, since the effect of thermal-hydraulic feedback is disregarded [11]. In this paper and for the sake of clarity, this transfer function is more explicitly labelled as the *point-kinetic* zero-power reactor transfer function. The reason for this additional precision lies with the fact that expression (30) indeed relies on a point-kinetic behaviour of the system, as Eqs. (23) and (24) demonstrate. The spatial dependence of the amplitude of the point-kinetic component is given by the static flux and no phase shift exists between two distant spatial points, as can be seen from Eq. (24). On the contrary, nothing can be said about the spatial dependence of the remaining component  $P_0\delta\psi_k(\mathbf{r}, t)$ ,  $k = 1, 2$  in Eq. (23) and the corresponding phase shift between two distant spatial points.

As will be shown in Section 2.3, a zero-power system might behave in a non-point kinetic manner. The amplitude and phase of the point-kinetic zero-power transfer function is represented in Fig. 1. In this figure, the so-called “plateau region”, obtained for frequencies in the range  $[\lambda; \beta/\Lambda_0]$ , is also plotted. In this region, the amplitude is nearly constant and equal to  $1/\beta$ , whereas the phase delay is close to zero.

It is remarkable that the point-kinetic zero-power transfer function has a simple analytical expression with respect to the frequency dependence. The extraction of the point-kinetic component from the fluctuations of the neutron flux at different frequencies thus provides a direct means of verifying and validating a core simulator and more generally any time- or frequency-dependent neutron transport solver.

## 2.2. Estimation of the point-kinetic component of the neutron fluctuations calculated from neutron transport solvers

If the fluctuations in neutron flux, i.e.  $\delta\phi_k(\mathbf{r}, t)$ ,  $k = 1, 2$ , are known, the corresponding point-kinetic component can also be estimated in the following manner. Combining Eq. (23) with Eq. (24), multiplying in a scalar manner with the vector  $[\phi_{1,0}^+(\mathbf{r})/v_1 \ \phi_{2,0}^+(\mathbf{r})/v_2]$  and integrating on the entire reactor volume results in:

$$\int \left[ \frac{1}{v_1} \phi_{1,0}^+(\mathbf{r}) \delta\phi_1(\mathbf{r}, t) + \frac{1}{v_2} \phi_{2,0}^+(\mathbf{r}) \delta\phi_2(\mathbf{r}, t) \right] d\mathbf{r} = \frac{\delta P(t)}{P_0} \int \left[ \frac{1}{v_1} \phi_{1,0}^+(\mathbf{r}) \phi_{1,0}(\mathbf{r}) + \frac{1}{v_2} \phi_{2,0}^+(\mathbf{r}) \phi_{2,0}(\mathbf{r}) \right] d\mathbf{r} + P_0 \int \left[ \frac{1}{v_1} \phi_{1,0}^+(\mathbf{r}) \delta\psi_1(\mathbf{r}, t) + \frac{1}{v_2} \phi_{2,0}^+(\mathbf{r}) \delta\psi_2(\mathbf{r}, t) \right] d\mathbf{r} \quad (31)$$

One notices from Eq. (20) that Eq. (17) also leads to:

$$\frac{\partial}{\partial t} \int \left[ \frac{1}{v_1} \phi_{1,0}^+(\mathbf{r}) \delta\psi_1(\mathbf{r}, t) + \frac{1}{v_2} \phi_{2,0}^+(\mathbf{r}) \delta\psi_2(\mathbf{r}, t) \right] d\mathbf{r} = 0 \quad (32)$$

or alternatively

$$\int \left[ \frac{1}{v_1} \phi_{1,0}^+(\mathbf{r}) \delta\psi_1(\mathbf{r}, t) + \frac{1}{v_2} \phi_{2,0}^+(\mathbf{r}) \delta\psi_2(\mathbf{r}, t) \right] d\mathbf{r} = \text{constant} \quad (33)$$

Because of the initial condition given by Eq. (22) and resulting in  $\delta\psi_k(\mathbf{r}, 0) = 0$ ,  $k = 1, 2$ , the constant on the right hand-side of Eq. (33) is identically equal to zero, i.e.:

$$\int \left[ \frac{1}{v_1} \phi_{1,0}^+(\mathbf{r}) \delta\psi_1(\mathbf{r}, t) + \frac{1}{v_2} \phi_{2,0}^+(\mathbf{r}) \delta\psi_2(\mathbf{r}, t) \right] d\mathbf{r} = 0 \quad (34)$$

Using this condition in Eq. (31) finally gives:

$$\frac{\delta P(t)}{P_0} = \frac{\int [\frac{1}{v_1} \phi_{1,0}^+(\mathbf{r}) \delta\phi_1(\mathbf{r}, t) + \frac{1}{v_2} \phi_{2,0}^+(\mathbf{r}) \delta\phi_2(\mathbf{r}, t)] d\mathbf{r}}{\int [\frac{1}{v_1} \phi_{1,0}^+(\mathbf{r}) \phi_{1,0}(\mathbf{r}) + \frac{1}{v_2} \phi_{2,0}^+(\mathbf{r}) \phi_{2,0}(\mathbf{r})] d\mathbf{r}} \quad (35)$$

or in the frequency domain:

$$\frac{\delta P(\omega)}{P_0} = \frac{\int [\frac{1}{v_1} \phi_{1,0}^+(\mathbf{r}) \delta\phi_1(\mathbf{r}, \omega) + \frac{1}{v_2} \phi_{2,0}^+(\mathbf{r}) \delta\phi_2(\mathbf{r}, \omega)] d\mathbf{r}}{\int [\frac{1}{v_1} \phi_{1,0}^+(\mathbf{r}) \phi_{1,0}(\mathbf{r}) + \frac{1}{v_2} \phi_{2,0}^+(\mathbf{r}) \phi_{2,0}(\mathbf{r})] d\mathbf{r}} \quad (36)$$

Dividing the above formula by  $\delta\rho(\omega)$  should allow retrieving the point-kinetic zero-power reactor transfer function of the reactor, according to Eq. (29). Ideally, the evaluation of the resulting expression should be identical to the one based on Eq. (30) directly. Contrary to Eq. (30), the use of Eq. (36) does not rely on the assumption of a point-kinetic behaviour of the reactor. Rather, the neutron fluctuations  $\delta\phi_k(\mathbf{r}, \omega)$ ,  $k = 1, 2$  might significantly deviate from a point-kinetic behaviour. In fact, these fluctuations are in the most general case induced by any type of perturbations, either localized or spatially-distributed. The determination of the full space-dependence of such fluctuations could be considered as estimating the response of the reactor to given types of perturbations or its zero-power transfer function. Still, the space-dependence of the neutron fluctuations might significantly deviate from the one of the static flux (or in short from point-kinetics), thus justifying the earlier distinction between the zero-power reactor transfer function and its point-kinetic component.

To make the comparison between the point-kinetic component of the response of the system and its expected theoretical expression, the following additional quantities are required:

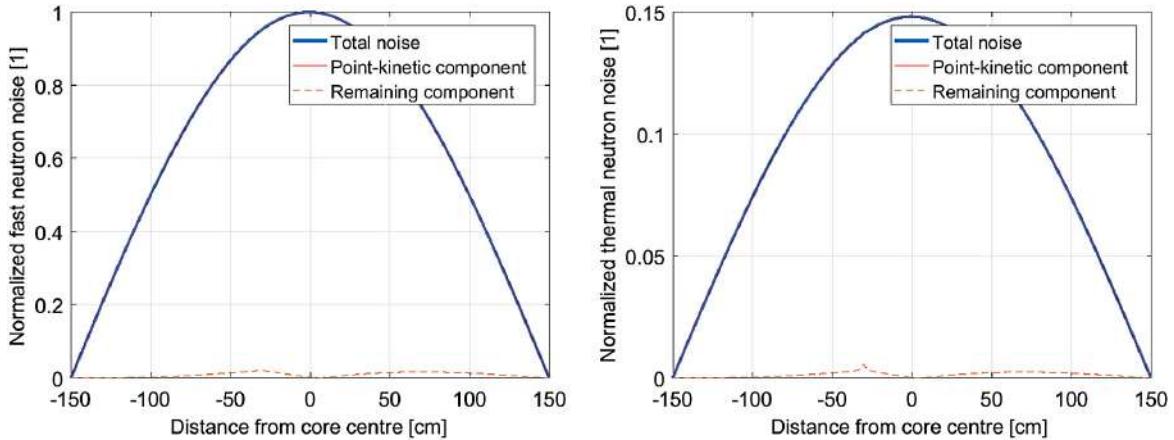
- For the theoretical expression, the static fluxes  $\phi_{k,0}(\mathbf{r})$ ,  $k = 1, 2$  and the adjoint eigenfunctions  $\phi_{k,0}^+(\mathbf{r})$ ,  $k = 1, 2$  [required for estimating the neutron generation time  $\Lambda_0$  in Eq. (30)].
- For the evaluation of the numerical expression [i.e. using Eqs. (36) and (27), the latter being taken in the frequency domain], the static fluxes  $\phi_{k,0}(\mathbf{r})$ ,  $k = 1, 2$ , the adjoint eigenfunctions  $\phi_{k,0}^+(\mathbf{r})$ ,  $k = 1, 2$ , and the neutron fluctuations  $\delta\phi_k(\mathbf{r}, \omega)$ ,  $k = 1, 2$ .

In the former case, no result of time- or frequency-dependent simulations corresponding to the applied perturbation is necessary.

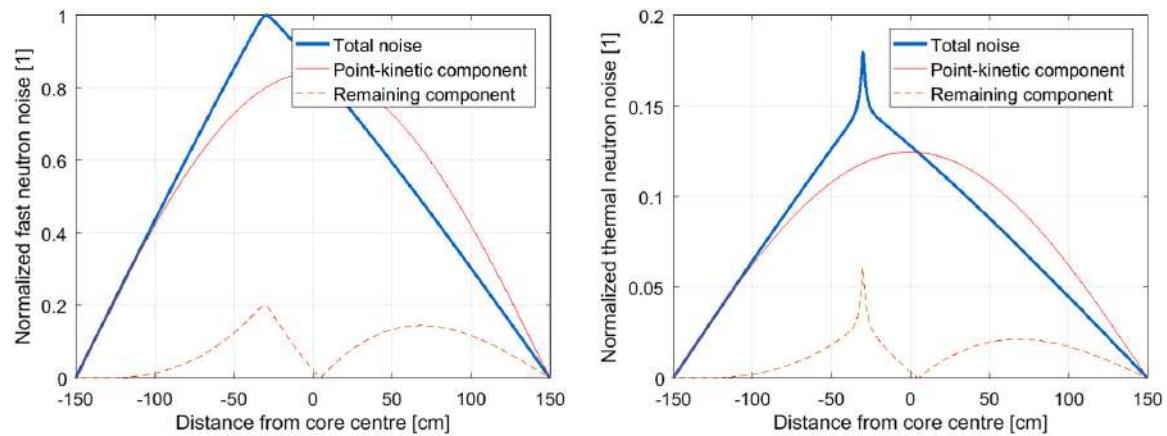
### 2.3. Expected deviation of the reactor response from point-kinetics

To test the present methodology and identify challenging situations, a one-dimensional homogeneous system representative of a bare reactor with 3 m as extrapolated size was considered in two-group diffusion theory, and the perturbation applied to the system was assumed to be a Dirac point-like perturbation in the fast energy group located 30 cm away from the core centre. In this situation, all necessary quantities, i.e. the static fluxes, the adjoint eigenfunctions, and the induced neutron noise, can be estimated fully analytically, or semi-analytically. The full solution can be found elsewhere, e.g. in [16]. The point-kinetic solution can also be estimated using either of the two procedures highlighted above and compared to the full solution. Such comparisons are represented in Figs. 2–4. In these figures, the amplitude of the complex-valued space-dependence of the total neutron noise  $\delta\phi_k(\mathbf{r}, \omega)$ ,  $k = 1, 2$ , of its point-kinetic component (which is proportional to  $\phi_{k,0}(\mathbf{r})$ ,  $k = 1, 2$ ) and of the difference between the two is given. It can be noticed from these figures that for low frequencies (frequencies smaller than the lower bound  $\lambda$  of the plateau region of the point-kinetic zero-power reactor transfer function), the response of the system follows a point-kinetic behaviour. This is explained by the fact that for decreasing frequencies, the transfer function  $G_0(\omega)$  diverges, as can be seen from Eq. (30) and Fig. 1, and the point-kinetic response becomes overwhelming. Such a large point-kinetic component results from the ability of the system to follow the applied driving perturbation without any time delay between spatially-distant points. For a non-zero reactivity perturbation, the neutron flux will thus be predominantly governed by its point-kinetic component, as Eqs. (23) and (24) demonstrate. On the other hand, the cases of high frequency perturbations (frequencies larger than the upper bound  $\beta/\Lambda_0$  of the plateau region of the point-kinetic zero-power reactor transfer function) are of special interest for the proposed methodology, since at those frequencies, the reactor response significantly deviates from point-kinetics.

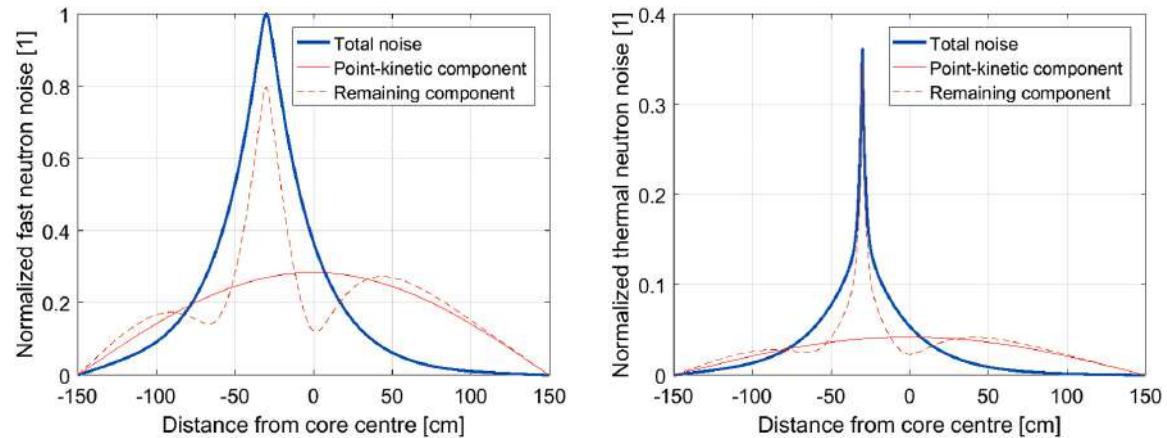
It is also known from the literature that the deviation from point-kinetics is largest for localized perturbations, as compared to spatially-distributed perturbations [16]. Therefore, applying the proposed methodology to high-frequency localized perturbations represents the most challenging case, for which being able to recover the point-kinetic component might be difficult. Therefore, the methodology was precisely tested in such situations for spatially non-homogeneous systems, thus requiring the use of time-dependent or frequency-dependent numerical solvers. The details of such investigations are reported in the following section.



**Fig. 2.** Comparisons between the amplitude of the neutron noise and its point-kinetic component in the case of a point-like perturbation located at  $-30$  cm from the core centre and at a frequency of  $1$  mHz (comparisons in the fast and thermal groups on the left and right figures, respectively).



**Fig. 3.** Comparisons between the amplitude of the neutron noise and its point-kinetic component in the case of a point-like perturbation located at  $-30$  cm from the core centre and at a frequency of  $1$  Hz (comparisons in the fast and thermal groups on the left and right figures, respectively).

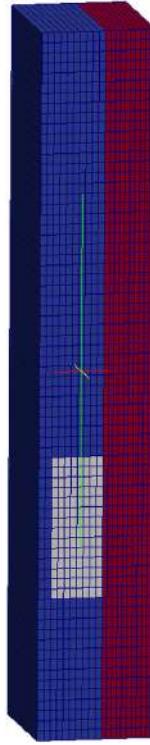


**Fig. 4.** Comparisons between the amplitude of the neutron noise and its point-kinetic component in the case of a point-like perturbation located at  $-30$  cm from the core centre and at a frequency of  $1$  kHz (comparisons in the fast and thermal groups on the left and right figures, respectively).

### 3. Using point-kinetics for code verification and validation

In this section, the feasibility of the proposed method highlighted above is tested. As identified earlier in Section 2.3, the cases of high-frequency localized perturbations are particularly challenging, since the reactor significantly deviates from point-kinetics. Three cases are thus presented hereafter: the case of a so-called localized absorber of variable strength, in the time domain and frequency domain, respectively, and the case of a travelling perturbation in the frequency domain.

In all cases, typical heterogeneous systems representative of commercial pressurized water reactors are chosen for performing the calculations. Two sets of in-house codes are used for this demonstration: a time-domain code for fuel assembly



**Fig. 5.** Representation of the system modelled for the case of a localized absorber of variable strength in the time domain. The blue colour corresponds to the fuel region, the red colour to the moderator region, and the grey colour to the region on which the perturbation was applied. The depicted system is compressed in the axial direction for illustration purposes. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

modelling [17,18] and a frequency-domain code for core simulations [19]. For each of the two codes, the static fluxes, the adjoint eigenfunctions, and the induced neutron noise are all estimated. The full space-dependence of the neutron noise can thus be estimated, and from it the point-kinetic component of the zero-power reactor transfer function evaluated. Such an evaluation is then directly compared to the theoretical expression of the transfer function.

The frequency domain approach allows directly performing the simulations at given frequencies. The modelling results in complex quantities, from which the magnitude and phase can be determined. In the time domain approach, on the other hand, the perturbation needs to be defined as a sine function having the desired frequency. Depending on the frequency of the perturbation, the system response might first experience transient conditions. Typically, after a few periods, the system reaches stationary conditions. Fitting a sine function to the local response of the system then allows estimating the magnitude and phase of the local response.

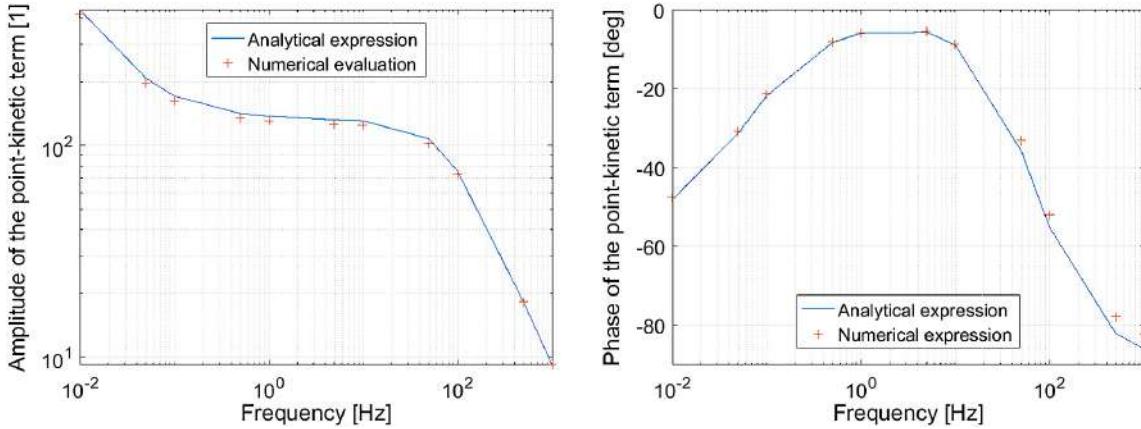
### 3.1. Case of a localized absorber of variable strength in the time domain

In this first case, a two-dimensional slab is modelled. The system has a finite height of 50 cm, with zero flux boundary conditions at the top and bottom of the system. In the finite radial direction, the system is made of two regions: a fuel region of width 0.45 cm and a moderator region of 0.25 cm. Reflective boundary conditions are applied in this radial direction, whereas the system is infinite in the transverse radial direction. A perturbation in a rectangular domain of width 0.35 cm and height 10 cm, placed 10 cm from the bottom of the fuel region is considered. Fig. 5 gives a representation of the system and of the region  $V_P$  on which the perturbation is applied. The perturbation is furthermore expressed as fluctuations of the total macroscopic cross-section in the thermal group. Since the simulations are performed in the time-domain, the fluctuations in the cross-section follow a user-defined sinusoidal temporal dependence as:

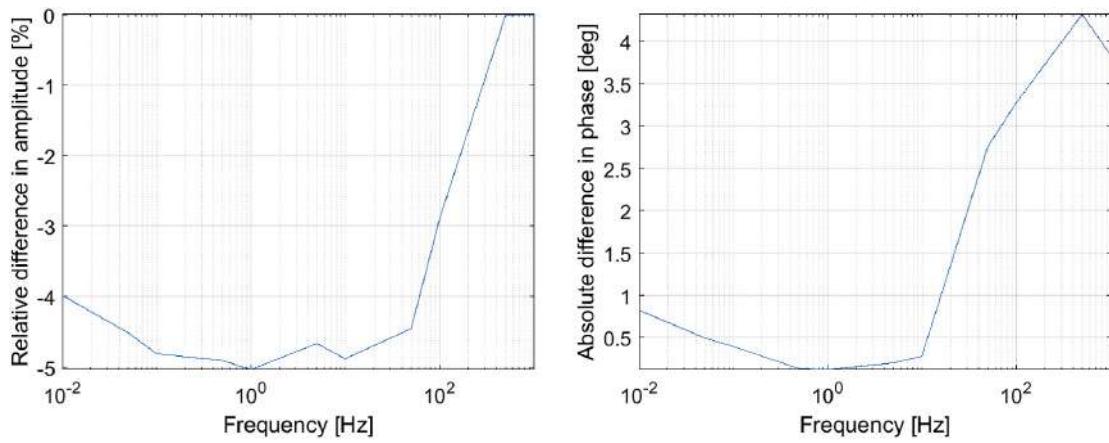
$$\Sigma_{T,2}(\mathbf{r}, t) = \Sigma_{T,2,0}(\mathbf{r}) + A \sin(\omega t) \quad \text{for } \mathbf{r} \in V_P \quad (37)$$

Although the perturbation is applied on the region  $V_P$  having a spatial spread, the perturbation only occupies a fraction of the entire spatial domain. Deviations from point-kinetics are thus expected, especially at high frequencies. The amplitude of the perturbation, which is homogeneous on  $V_P$ , is set to 0.1% of the unperturbed total macroscopic cross-section.

The frequency-dependence of the point-kinetic component of the zero-power reactor transfer function is represented in Fig. 6 computed per the methodology earlier highlighted and then compared to the expected analytical expression. The numerical evaluation is based on the actual neutron noise induced by the localized perturbations. The theoretical expression, on the other hand, is independent of the perturbations applied to the system. As can be seen in this figure, recovering the point-kinetic component of the induced neutron noise leads to a correct estimation of the transfer function,



**Fig. 6.** Frequency-dependence of the amplitude (left figure) and phase (right figure) of  $\delta P(\omega)/[P_0\delta\rho(\omega)]$  as estimated from the two-dimensional time-dependent simulation in the case of an absorber of variable strength and compared to the analytical expression.



**Fig. 7.** Frequency-dependence of the relative difference in amplitude (left figure) and absolute difference in phase (right figure) in  $\delta P(\omega)/[P_0\delta\rho(\omega)]$  between the numerical evaluation and the analytical expression, as estimated from the two-dimensional time-dependent simulation in the case of an absorber of variable strength.

even at high frequencies. Although the point-kinetic component is several orders of magnitude smaller at high frequencies than at the plateau region-frequencies, the estimation of the point-kinetic component is still correct. The relative difference in magnitude and absolute difference in phase, respectively, between the numerical evaluation of the zero-power reactor transfer function and its analytical expression as a function of frequency, generically expressed as:

$$\text{relative difference in } \|X(\omega)\| = \frac{\|X_{num}(\omega)\| - \|X_{ana}(\omega)\|}{\|X_{ana}(\omega)\|} \quad (38)$$

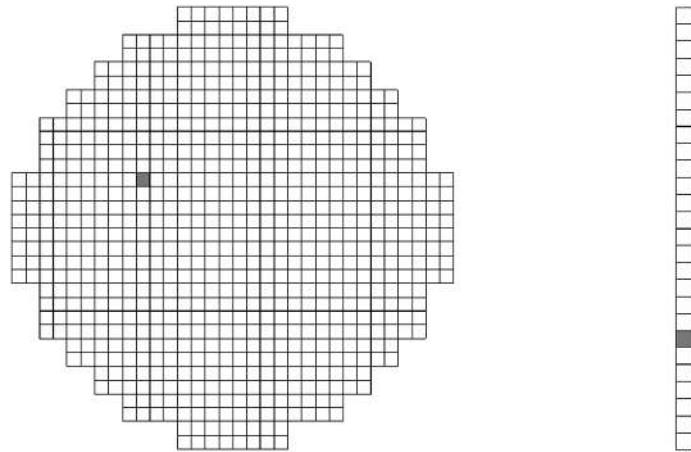
and

$$\text{absolute difference in phase of } X(\omega) = \text{phase of } X_{num}(\omega) - \text{phase of } X_{ana}(\omega), \quad (39)$$

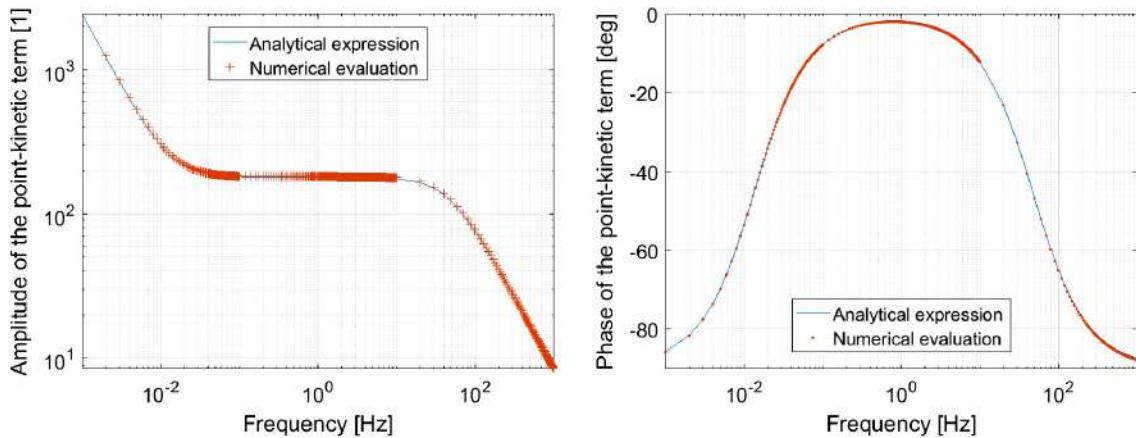
respectively, are represented in Fig. 7.  $X_{num}$  represents the numerical evaluation of  $\delta P(\omega)/[P_0\delta\rho(\omega)]$  and  $X_{ana}$  represents the corresponding analytical expression. One notices that the largest deviations in absolute values are observed at low frequencies for the magnitude (amounting to 5%) and at high frequencies for the phase (amounting to nearly 4.5 deg). Such relative deviations are small and are most likely related to the fitting procedure used to estimate the magnitude and phase of the point-kinetic component from the time-dependent simulations. It can thus be concluded from this first test case that retrieving the point-kinetic component of the zero-power reactor transfer function is possible. In the present case, it also demonstrates that the in-house time-dependent tool gives a faithful estimation of the expected analytical transfer function, thus providing a further verification and validation of the tool.

### 3.2. Case of a localized absorber of variable strength in the frequency domain

This second case is, in essence, similar to the first test case earlier presented. The main difference lies with the fact that the simulations are directly performed in the frequency domain and that the calculations are carried out for an entire reactor core. The system being considered is typical of a pressurized water reactor, and is modelled by a Cartesian coarse mesh grid of radial size 32 × 32 and of axial size 26, as depicted in Fig. 8. The active core region is of radial size 30 × 30 and of axial size



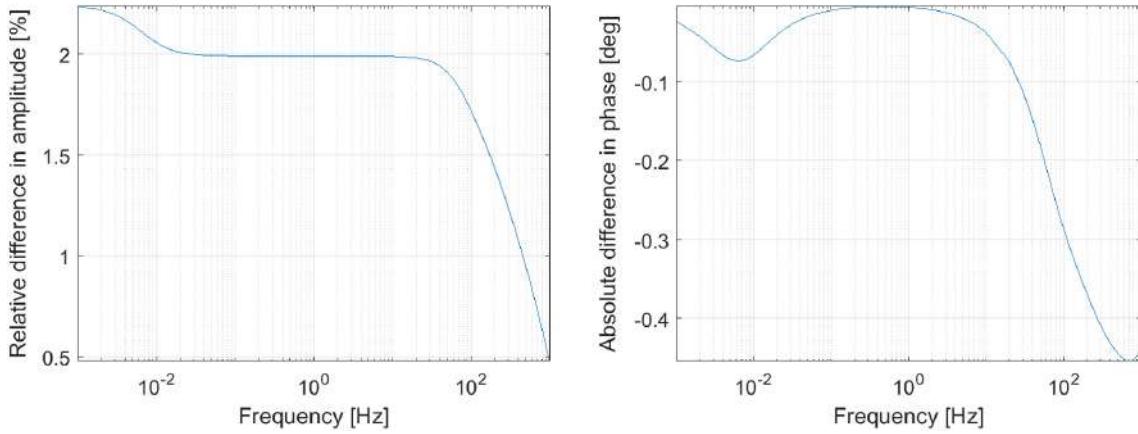
**Fig. 8.** Representation of the system modelled for the case of a localized absorber of variable strength in the frequency domain (radial layout on the left, axial layout on the right). The grey colour corresponds to the node on which the perturbation was applied. The depicted system is compressed in the axial direction for illustration purposes.



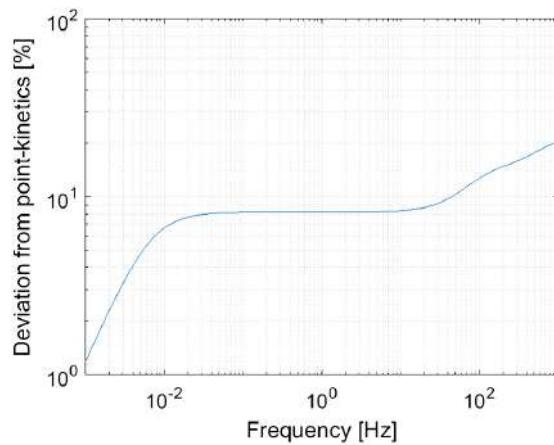
**Fig. 9.** Frequency-dependence of the amplitude (left figure) and phase (right figure) of  $\delta P(\omega)/[P_0\delta\rho(\omega)]$  as estimated from the three-dimensional frequency-dependent core simulations in the case of an absorber of variable strength and compared to the analytical expression.

24 and is surrounded by a reflector being modelled explicitly. Zero flux boundary conditions are applied at the periphery of the system. Each elementary node, assumed to be spatially homogeneous, has for dimensions 10.75 cm in both radial directions and 15.24 cm in the axial direction. A white noise point-like perturbation located radially in one of the quarters of the core and axially in its lower half is considered, as can be seen in Fig. 8. This perturbation is furthermore expressed as fluctuations of the macroscopic absorption cross-section in the thermal group. Since the calculations are performed in the frequency domain directly after removal of the static solution, only the fluctuations need to be resolved. Because of the inhomogeneous nature of the set of differential equations obtained, the amplitude of the perturbation simply acts as a scaling factor to the solution and is thus unimportant. The definition of the perturbation thus reduces to a non-zero value in the node where the perturbation exists and the amplitude of the solution is thus proportional to the amplitude of the perturbation applied.

The frequency-dependence of the point-kinetic component of the zero-power reactor transfer function is represented in Fig. 9 and compared to the analytical expression. The relative difference in amplitude and absolute difference in phase, respectively, as evaluated in Eqs. (38) and (39), are given in Fig. 10. As for the simulations performed in the time domain presented in Section 3.1, excellent agreement is reached between the evaluation based on the actual neutron noise induced by the localized perturbations and the theoretical expression. The maximum deviation is slightly above 2% for the magnitude and  $-0.5$  deg for the phase. Such deviations are smaller than the ones reported in Section 3.1. This is explained by the fact that no fitting procedure as the one used in the time-domain is required, since the simulations are directly performed in the frequency domain, thus directly providing amplitude and phase of all computed quantities. It is remarkable that the theoretical expression is independent of the perturbations applied to the system, whereas the numerical estimation relies on the actual perturbation applied to the system. Very good agreement can even be noticed at high frequencies, for which



**Fig. 10.** Frequency-dependence of the relative difference in amplitude (left figure) and absolute difference in phase (right figure) in  $\delta P(\omega)/[P_0\delta\rho(\omega)]$  between the numerical evaluation and the analytical expression, as estimated from the three-dimensional frequency-dependent core simulations in the case of an absorber of variable strength and compared to the analytical expression.



**Fig. 11.** Relative deviation of the reactor response from point-kinetics as evaluated from Eq. (40) in the case of an absorber of variable strength in the frequency domain.

the contribution of the point-kinetic component is small compared to the actual neutron noise. This can be seen in Fig. 11, where the deviation from point kinetics is estimated according to the following integral measure:

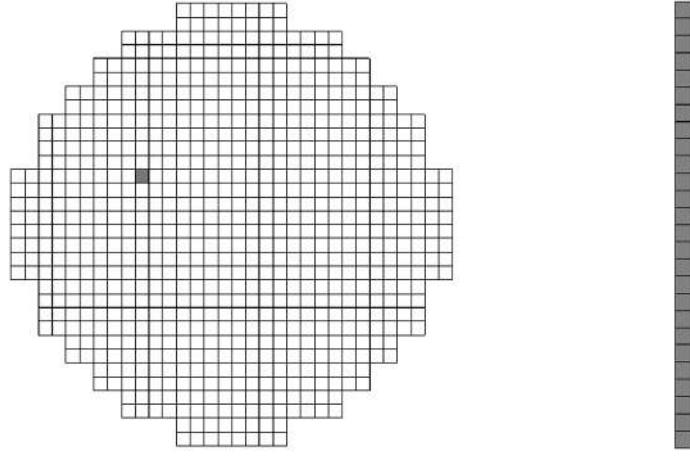
$$dev(\omega) = \left\| \int_V \frac{\delta\phi_1(\mathbf{r}, \omega) + \delta\phi_2(\mathbf{r}, \omega) - G_0(\omega) \times [\phi_{1,0}(\mathbf{r}) + \phi_{2,0}(\mathbf{r})] \times \delta\rho(\omega)}{G_0(\omega) \times [\phi_{1,0}(\mathbf{r}) + \phi_{2,0}(\mathbf{r})] \times \delta\rho(\omega)} d\mathbf{r} \right\| \quad (40)$$

It is observed in this figure that the reactor deviates from point-kinetics already at plateau frequencies. The retrieval of the point-kinetic component and its successful comparison with the analytical expression demonstrates that the proposed method is a powerful tool for verification and validation.

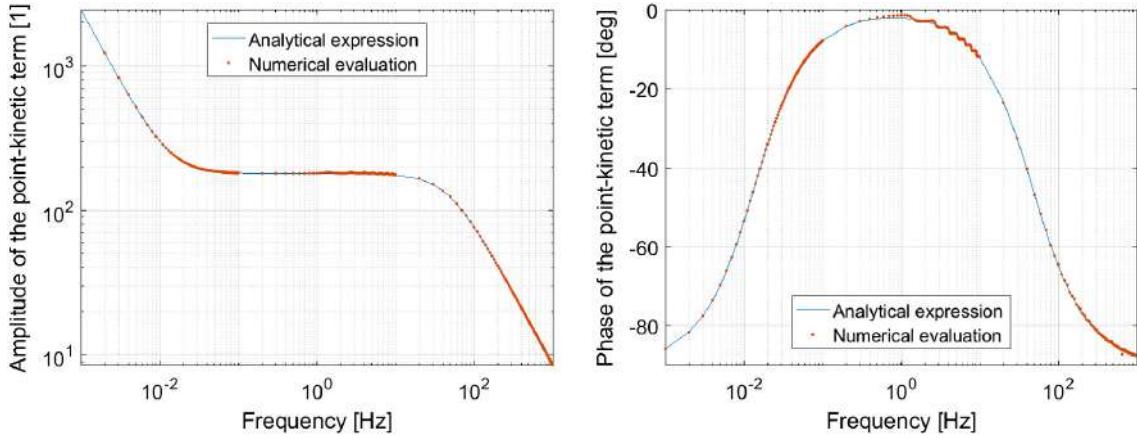
### 3.3. Case of a travelling perturbation in the frequency domain

In this third and last case, a propagating perturbation assumed to travel upwards in one of the fuel assemblies with the coolant, having a flow velocity  $v$ , is considered. The system being considered is identical to the one used in the previous case, i.e. a pressurized water reactor modelled by a Cartesian coarse mesh grid of radial size  $32 \times 32$  and of axial size 26, as depicted in Fig. 12. The active core region is of radial size  $30 \times 30$  and of axial size 24 and is surrounded by a reflector being modelled explicitly. Zero flux boundary conditions are applied at the periphery of the system. Each elementary node, assumed to be spatially homogeneous, has for dimensions 10.75 cm in both radial directions and 15.24 cm in the axial direction. The perturbation is assumed to modify the removal macroscopic cross-section. In the time-domain, such a perturbation would be expressed as:

$$\delta\Sigma(x_0, y_0, z, t) = \delta\Sigma\left(x_0, y_0, 0, t - \frac{z}{v}\right) \quad (41)$$



**Fig. 12.** Representation of the system modelled for the case of a travelling perturbation in the frequency domain (radial layout on the left, axial layout on the right). The grey colour corresponds to the nodes on which the perturbation was applied. The depicted system is compressed in the axial direction for illustration purposes.



**Fig. 13.** Frequency-dependence of the amplitude (left figure) and phase (right figure) of  $\delta P(\omega)/[P_0 \delta \rho(\omega)]$  as estimated from the three-dimensional frequency-dependent core simulations in the case of a travelling perturbation and compared to the analytical expression.

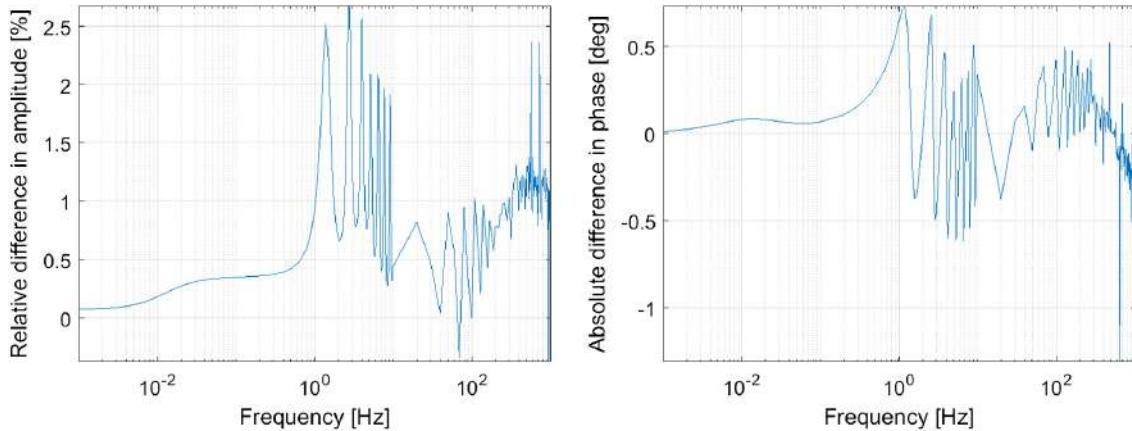
or in the frequency-domain as:

$$\delta \Sigma(x_0, y_0, z, \omega) = \delta \Sigma(x_0, y_0, 0, \omega) \exp\left(-\frac{i\omega z}{v}\right) \quad (42)$$

where  $(x_0, y_0)$  represents the radial location of the fuel assembly in which the perturbation is applied, and  $z$  corresponds to the axial elevation within the core. As for the case of the absorber of variable strength detailed in Section 3.2, the noise source is assumed to be a white noise source and is this time applied at the inlet of the fuel channel. The main difference compared to the previous case lies with the fact that the noise source is spread over the entire length of the core (but still radially localized) and most importantly that the phase varies linearly with height. In terms of definition of the perturbation, this means that the amplitude of the perturbation is axially homogeneous, whereas the phase is changing.

The frequency-dependence of the point-kinetic component of the zero-power reactor transfer function is represented in Fig. 13 using the methodology earlier highlighted and compared to the expected analytical expression. The relative difference in amplitude and absolute difference in phase, respectively, as evaluated in Eqs. (38) and (39), are given in Fig. 14. As for the absorber of variable strength, the evaluation is based on the actual neutron noise induced by the applied perturbations for the former case, whereas in the latter case, the theoretical expression is independent of the perturbations applied to the system. It can be seen from the figure that recovering the point-kinetic component of the induced neutron noise leads to a correct estimation of the transfer function. The relative deviation in amplitude is at maximum around 2.5% and the absolute deviation in phase amounts to nearly 1 deg in absolute value. Although the observed deviations are comparable to the ones obtained for the absorber of variable strength, some “wobbling” of the numerical solution around the expected smooth analytical solution can be seen.

The relative deviation of the reactor response from point-kinetics, calculated using the same integral measure as for the case of the absorber of variable strength – see Eq. (40) –, is represented in Fig. 15. One notices that the deviation from point-kinetics is also comparable to the one corresponding to the absorber of variable strength.



**Fig. 14.** Frequency-dependence of the relative difference in amplitude (left figure) and absolute difference in phase (right figure) in  $\delta P(\omega)/[P_0\delta\rho(\omega)]$  between the numerical evaluation and the analytical expression, as estimated from the three-dimensional frequency-dependent core simulations in the case of a travelling perturbation and compared to the analytical expression.

The “wobbling” of the numerical solution cannot thus be explained by a loss of point-kinetic behaviour of the system. Rather, it is the result of the linear phase of the applied perturbations leading to a smaller reactivity effect. This can be noticed from Eq. (27), which written in the frequency domain and only retaining first-order terms leads to:

$$\delta\rho(\omega) = \frac{\int [-\delta\Sigma_{rem}(\mathbf{r}, \omega)\phi_{1,0}^+(\mathbf{r})\phi_{1,0}(\mathbf{r}) + \delta\Sigma_{rem}(\mathbf{r}, \omega)\phi_{2,0}^+(\mathbf{r})\phi_{1,0}(\mathbf{r})]d\mathbf{r}}{\int [\nu\Sigma_{f,1,0}(\mathbf{r})\phi_{1,0}^+(\mathbf{r})\phi_{1,0}(\mathbf{r}) + \nu\Sigma_{f,2,0}(\mathbf{r})\phi_{1,0}^+(\mathbf{r})\phi_{2,0}(\mathbf{r})]d\mathbf{r}} \quad (43)$$

The phase varying linearly along the height of the channel implies that for high enough frequencies, the phase will spread over the entire interval  $[0; 2\pi]$  and for even higher frequencies, the interval will be covered several times along the height. This results in partial compensation of the reactivity contribution when integrating along the height of the system. Since the reactivity appears in the denominator of  $\delta P(\omega)/[P_0\delta\rho(\omega)]$ , small values of the reactivity will lead to less reliable estimations of the transfer function. This is more clearly demonstrated in Fig. 16, where an analysis of the results is presented. In this figure, the extracted point-kinetic component and its relative deviation from its analytical expression are plotted, together with the relative amplitude factor  $\delta P(\omega)/P_0$  and the reactivity effect  $\delta\rho(\omega)$ . The ratio between the two last quantities corresponds to the extracted point-kinetic component. One notices from these figures that the amplitude of the reactivity effect starts to significantly decrease at around 0.6–0.8 Hz, which also corresponds to the frequency range for which the reliability of the method starts to slightly deteriorate. Compared to Fig. 15, one also notices that the loss of the point-kinetic behaviour is already appreciable at 0.1 Hz, i.e. at lower frequencies. Therefore, the slight deterioration in the reliability of the method cannot be solely related to the loss of point-kinetic behaviour of the system, but also to a reactivity perturbation having a much lower amplitude.

If the applied perturbation had a reactivity effect exactly equal to zero, the point-kinetic component of the induced neutron noise would be identically equal to zero, as Eqs. (23), (24) and (29) demonstrate, and the method proposed in this paper would break down.

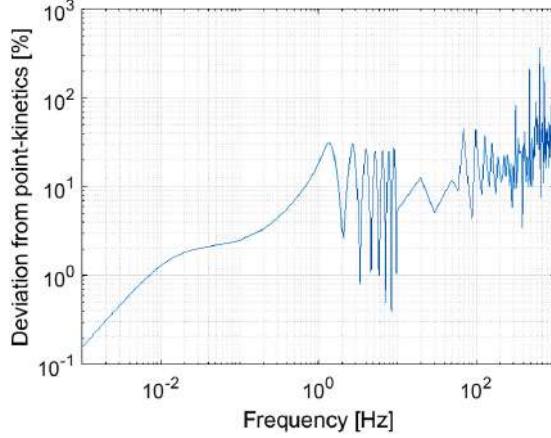
Despite the apparent slight deterioration of the reliability of the method at higher frequencies for the case of the travelling perturbation, the relative deviation between the extracted point-kinetic component and its expected analytical expression is about a few percent. This is comparable with the small deviations noticed in the case of a localized absorber in both the time- and frequency-domain. Such small deviations are marginal, demonstrating that the method can be faithfully used to assess the validity of core simulators.

#### 4. Discussion and conclusions

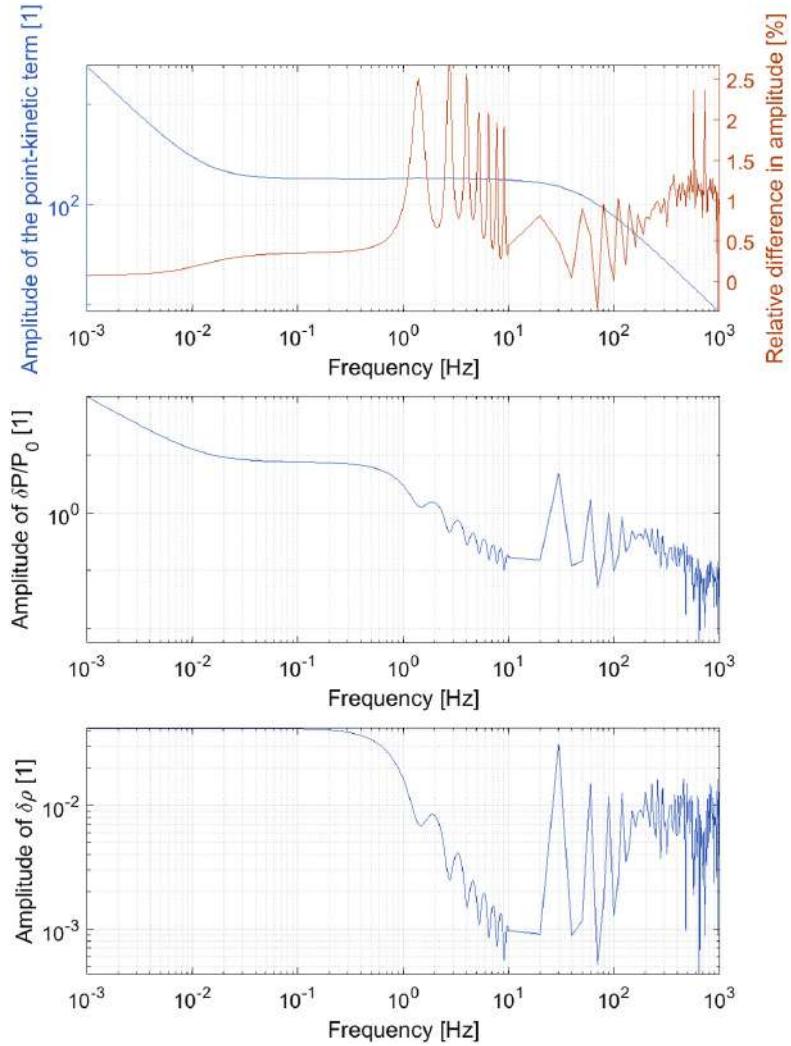
In this paper, a methodology was presented to estimate from the results of calculations of three-dimensional frequency domain-based and time domain-based neutronic core simulators the point-kinetic component of the neutron fluctuations induced by given perturbations. Correspondingly, the zero-power transfer function can be estimated by renormalizing this component to the reactivity effect of the applied perturbations.

This paper demonstrated that even in challenging situations when the reactor response significantly deviates from point-kinetics, such as extremely localized perturbations and high frequencies, such a point-kinetic component is retrievable and could be successfully used to benchmark core simulators. This method represents one of the very few methodologies where the results of three-dimensional calculations of a heterogeneous system can be compared with an analytical expression, thus offering an opportunity to further verify and validate such codes.

Nevertheless, this method also assumes that the point-kinetic component of the induced neutron fluctuations does exist. Perturbations resulting in exact reactivity compensation lead to a point-kinetic component identically equal to zero, for



**Fig. 15.** Relative deviation of the reactor response from point-kinetics as evaluated from Eq. (40) in the case of a travelling perturbation.



**Fig. 16.** Plotting of the amplitude of: the extracted point-kinetic component and its relative deviation from its analytical expression (upper figure), the relative amplitude function (middle figure), and the applied reactivity perturbation (lower figure) in the case of a travelling perturbation.

which the method will fail. Close to zero reactivity effects are also expected to lead to slightly less reliable estimations of the point-kinetic component of the transfer function, as the case of a travelling perturbation demonstrated in this work.

When applying the method, the analyst can consider any type of stationary perturbations, i.e. he/she does not need to only consider situations where point-kinetics is applicable. Although the method was tested in this study by direct perturbations of the macroscopic cross-sections, other types of perturbations can be introduced, such as perturbations of the thermal-hydraulic variables. The only restriction lies with the necessity to define perturbations having a non-zero reactivity

effect. The method also requires scanning a frequency range of the applied perturbations, so that the actual dependence on frequency of the point-kinetic zero-power reactor transfer function can be determined and compared to its expected analytical expression. If the frequency-dependence of the extracted point-kinetic component does not follow its expected analytical expression, a modelling issue (bug) in the code used for the calculations is to be suspected.

Extension to neutron transport methods of higher order than diffusion theory and a generalization of the methodology to any number of energy groups will be the subject of future publications. Available multi-group time-dependent neutron transport solvers that can estimate the adjoint function of the static flux will be used to test the applicability of the proposed method. Due to its generic nature, this verification and validation framework could be applied to any type of core simulators.

## Acknowledgements

The third author's involvement in this work was funded by the Swedish Centre for Nuclear Technology (SKC).

## References

- [1] C. Demazière, Multi-physics modelling of nuclear reactors: current practices in a nutshell, *Int. J. Nucl. Energy Sci. Technol.* 7 (4) (2013) 288–318.
- [2] B.W. Boehm, *Software Engineering Economics*, Prentice-Hall, Saddle River, NJ, 1981.
- [3] W.L. Oberkampf, T.G. Trucano, Verification and validation benchmarks, *Nucl. Eng. Des.* 238 (2008) 716–743.
- [4] W.L. Oberkampf, T.G. Trucano, C. Hirsch, Verification, validation, and predictive capability in computational engineering and physics, *Appl. Mech. Rev.* 57 (5) (2004) 345–384.
- [5] B. Ganapol, Analytical benchmarks for nuclear engineering applications: case studies in neutron transport theory, Tech. rep. OECD/NEA, Paris, 2008.
- [6] A. Talamo, Numerical solution of the time dependent neutron transport equation by the method of the characteristics, *J. Comput. Phys.* 240 (2013) 248–267.
- [7] T.A. Brunner, J.P. Holloway, Two-dimensional time dependent Riemann solvers for neutron transport, *J. Comput. Phys.* 210 (2005) 386–399.
- [8] S. Pelloni, P. Grimm, D.R. Mathews, J.M. Paratte, Validation of light water reactor calculation methods and JEF-1 based data libraries by TRX and BAPL critical experiments, *Nucl. Technol.* 94 (1) (1991) 15–27.
- [9] D.P. Weber, et al., High-fidelity light water reactor analysis with the Numerical Nuclear Reactor, *Nucl. Sci. Eng.* 155 (3) (2007) 395–408.
- [10] M.A. Smith, E.E. Lewis, B.C. Na, Benchmark on deterministic 2-D MOX fuel assembly transport calculations without spatial homogenization, *Prog. Nucl. Energy* 45 (2–4) (2004) 107–118.
- [11] G.I. Bell, S. Glasstone, *Nuclear Reactor Theory*, Van Nostrand Reinhold Company, New York, USA, 1970.
- [12] T. Kozlowski, A. Wysocki, I. Gajev, Y. Xu, T. Downar, K. Ivanov, J. Magedanz, M. Hardgrove, J. March-Leuba, N. Hudson, W. Ma, Analysis of the OECD/NRC Oskarshamn-2 BWR stability benchmark, *Ann. Nucl. Energy* 67 (2014) 4–12.
- [13] A. Dokhane, H. Ferroukhi, A. Pautz, Validation studies and interpretation of the Oskarshamn-2 1999 stability event with SIMULATE-3K, *Ann. Nucl. Energy* 96 (2016) 344–353.
- [14] C. Demazière, I. Pázsit, Numerical tools applied to power reactor noise analysis, *Prog. Nucl. Energy* 51 (2009) 67–81.
- [15] V. Larsson, C. Demazière, Comparative study of 2-group P1 and diffusion theories for the calculation of the neutron noise in 1D 2-region systems, *Ann. Nucl. Energy* 36 (10) (2009) 1574–1587.
- [16] I. Pázsit, C. Demazière, Noise techniques in nuclear systems, in: D. Cacuci (Ed.), *Handbook of Nuclear Engineering*, vol. 3, Springer, 2010.
- [17] K. Jareteg, P. Vinai, S. Sasic, C. Demazière, Coupled fine-mesh neutronics and thermal-hydraulics – modeling and implementation for PWR fuel assemblies, *Ann. Nucl. Energy* 84 (2015) 244–257.
- [18] K. Jareteg, R. Andersson, C. Demazière, Development and test of a transient fine-mesh LWR multiphysics solver in a CFD framework, in: Proc. Joint Int. Conf. Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method, MC2015, Nashville, TN, USA, April 19–23, 2015, American Nuclear Society, LaGrange Park, IL, 2015, on CD-ROM.
- [19] C. Demazière, CORE SIM: a multi-purpose neutronic tool for research and education, *Ann. Nucl. Energy* 38 (12) (2011) 2698–2718.