Imperial College of Science, Technology and Medicine
Department of Mechanical Engineering

Nuclear thermal hydraulic analysis using coupled CFD and system codes

Antonello Palazzi

Thesis submitted for
the Degree of Doctor of Engineering
of Imperial College London
and for the
Diploma of Imperial College

2018

'*We shall go on to the end [...] We shall never surrender!*'

W. Churchill

Declaration

I herewith certify that all the material in this thesis that is not my own work has been properly acknowledged.

Antonello Palazzi

Acknowledgements

I am sincerely grateful to my supervisors at Imperial College London and at CD-adapco, Dr. Mike Bluck and Dr. Simon Lo, for giving me the opportunity to pursue this research, for their encouragement, enthusiasm and above all, their patience throughout the challenges this work posed me.

I would like to thank CD-adapco for their financial and technical contributions and the opportunity to work in their London office for the last year and a half of the EngD. I would also like to thank the UK Engineering and Physical Science Research Council for providing funding for this work. In particular I am indebted to Dr. Sava Slijepčević of CD-adapco, whose support was invaluable for the outcomes of this work.

I would like to acknowledge my colleagues from the Nuclear Engineering Group at Imperial College London, for providing a cooperative working environment, as well as technical and emotional support throughout all my time here. In particular I am grateful to Anthony, Giovanni and Dave for their constant presence.

I would also like to acknowledge the people of CD-adapco's London office, for making me feel part of a family, in particular Thomas, Max, Marta, Gianluca and Charles.

I am deeply grateful to Lavinia and Paolo, whose support was vital during the lowest emotional points of the past four years.

Last, but by no means least, I would like to thank my parents for their support throughout my entire life and for their understanding.

Abstract

The thermal hydraulic analysis of nuclear reactors is largely performed by what are known as system codes. These codes predict the flows in the complex network of pipes, pumps, vessels and heat exchangers that together form the thermal hydraulic systems of a nuclear reactor. These codes have been used for many decades and are now very well established. Given this long process of refinement, they are able to produce remarkably accurate predictions of plant behaviour under both steady and transient conditions. Modern CFD is able to produce high-quality predictions of flows in complex geometries, but only with the use of large computing resources. It would be impractical to build a CFD model of, for example, the entire primary circuit of a PWR. However, it is possible to model with adequate fidelity much of the primary circuit using a cheaper one-dimensional system code, and it may only be in a limited part of the circuit that full three-dimensional effects are important.

A coupling scheme was developed to couple the CFD software STAR-CCM+ and the system code RELAP5-3D. The structure of the scheme is presented, together with validations for single phase flow in smooth pipes in both transient and steady state cases. Attention was also given to the problem of reconstructing the flow profile at the inlet of the CFD model under the hypothesis of fully developed flow. This problem arises when flow data has to be passed from the one-dimensional system code to the three-dimensional CFD software.

The coupling scheme was then modified to be able to perform multiphase simulations. The PWR subchannel and bundle test (PSBT) benchmark was used to validate the multiphase coupling methodology.

Overall, the results are satisfactory and show how this tool represent a great potential for problems of industrial relevance.

Contents

List of Tables

List of Figures

Nomenclature

Acronyms

API Application programming interface

ATWS Anticipated transient without scram

BFBT BWR full-size fine-mesh bundle tests

BPLU Border-profile lower-upper

BWR Boiling water reactor

CANDU Canada deuterium uranium reactor

CEA Commissariat à l'énergie atomique (French atomic energy commission)

CFD Computational fluid dynamics

CFD4NRS CFD for nuclear reactor safety applications workshop

CL Cold leg

COWS Co-simulation with one-dimensional wrapper solver

DHX Downcomer/Heat exchanger

DNB Departure from nucleate boiling

ECC Emergency core cooling

GMRES Generalised minimal residual

GUI Graphical user interface

HL Hot leg

INL Idaho National Laboratory

LOCA Loss of coolant accident

LOFA Loss of flow accident

LOHS Loss of heat sink

LOOP Loss of offsite power

LWR Light water Reactor

CASL Consortium for Advance Simulation of LWRs

DOE US Department of Energy

MOOSE Multiphysics object-oriented simulation environment

NCCS National center for computational sciences

NEA Nuclear energy agency

OECD Organisation for economic co-operation and development

OO Object-oriented

PG Pressure gradient

PGMRES Preconditioned GMRES

PSBT PWR subchannel and bundle tests

PVM Parallel virtual machine

PWR Pressurised water reactor

RELAP Reactor excursion and leak analysis program

RPV Reactor pressure vessel

SIMPLE Semi-implicit method for pressure linked equations

SNAP Symbolic nuclear analysis package

STS Standard technical specifications

TDJ Time dependent junction

TDV Time dependent volume

UDF User defined function

USNRC United States nuclear regulatory commission

VHTR Very high temperature reactor

WS Wall shear stress

Greek

α_f Liquid phase volume fraction

α_g Gaseous phase volume fraction

α_i Volume fraction of phase i

β Effective delayed neutron fraction

ϵ Absolute roughness height

Γ_f Liquid phase mass flow rate per unit volume

Γ_g Gaseous phase mass flow rate per unit volume

Γ_w Vapour generation resulting from energy exchange in the near-wall thermal boundary layer

Γ_{ig} Vapour generation resulting from energy exchange in the bulk fluid

κ Turbulent kinetic energy

κ_i Turbulent kinetic energy of phase i

κ_j^γ Turbulent kinetic energy of phase j at interface γ

Λ Prompt neutron generation time

λ_i Decay constant of group i

μ	Dynamic viscosity
μ_i	Dynamic viscosity of phase i
μ_t	Turbulent viscosity
$\mu_{t,i}$	Turbulent viscosity of phase i
ν	Kinematic viscosity
ν_i	Kinematic viscosity of phase i
ρ	Density, reactivity
ρ_b	Boron density
ρ_f	Liquid phase density
ρ_g	Gaseous phase density
ρ_i	Density of phase i
ρ_m	Mixture density
Σ_f	Macroscopic fission cross-section
σ_κ	$\kappa - \varepsilon$ model coefficient
σ_ε	$\kappa - \varepsilon$ model coefficient
$\sigma_{t,i}$	Turbulent thermal diffusion Prandtl number of phase i
τ_i	Molecular/collisional stress tensor
τ_i^t	Turbulent stress tensor
τ_w	Wall shear stress
Υ_M	Compressibility modification term
$\Upsilon_{M,i}$	Compressibility modification term of phase i
ε	Turbulent dissipation rate
ε_0	Ambient turbulence value in sources that counteracts turbulence decay

ε_i Turbulent dissipation rate of phase i

φ Neutron flux

ξ Source term

ξ_i Source term of phase i

Roman

A Cross sectional area

B_x Body forces

C Courant number

C Virtual mass coefficient

C_b Boron mass concentration

C_i Delayed neutron precursor concentration in group i

c_p Specific heat

$c_{p,i}$ Specific heat of phase i

$C_{\varepsilon 1}$ $\kappa - \varepsilon$ model coefficient

$C_{\varepsilon 2}$ $\kappa - \varepsilon$ model coefficient

$C_{\varepsilon 3}$ $\kappa - \varepsilon$ model coefficient

D Pipe diameter

$DISS_f$ Liquid phase energy dissipation term

$DISS_g$ Gaseous phase energy dissipation term

E Total energy

E_f Energy per fission

E_i Total energy of phase i

$\mathbf{F}$ Body forces

f Moody's friction factor

f_c Curvature correction factor

f_i Fraction of delayed neutrons of group i, β_i/β

$f_{c,i}$ Curvature correction factor of phase i

FIG Gaseous phase interphase friction coefficient

FIL Liquid phase interphase friction coefficient

FWG Gaseous phase wall friction coefficient

FWL Liquid phase wall friction coefficient

$\mathbf{g}$ Gravity vector

G_b Buoyancy production term

G_k Turbulent production term

$G_{b,i}$ Buoyancy production term of phase i

$G_{k,i}$ Turbulent production term of phase i

H Total enthalpy

h Enthalpy

h'_f Liquid phase specific enthalpy associated with near-wall interface mass transfer

h'_g Gaseous phase specific enthalpy associated with near-wall interface mass transfer

h^*_f Liquid phase specific enthalpy associated with bulk interface mass transfer

h^*_g Gaseous phase specific enthalpy associated with bulk interface mass transfer

H_i Total enthalpy of phase i

h_i Enthalpy of phase i

$h_{\gamma,i}$ Enthalpy of phase i evaluated at the temperature of interface γ

k Thermal conductivity

k_i Thermal conductivity of phase i

k_i^{eff} Effective thermal conductivity of phase i

$\mathbf{M}_i$ Interphase momentum transfer per unit volume

m_{ij} Mass transfer rate from phase i to phase j

m_{ji} Mass transfer rate from phase j to phase i

$\mathbf{n}$ Normal vector

n Neutron density

N_d Number of delayed neutron precursor groups

p Pressure

P_f Total fission power

$\dot{\mathbf{q}}''$ Heat flux vector

Q_i^{γ} Interphase heat transfer rate from interface γ to phase i

Q_{ig} Gaseous phase interface heat transfer rate

Q_{ij} Interphase heat transfer rate from phase j to phase i

Q_{il} Liquid phase interface heat transfer rate

Q_{wg} Gaseous phase wall heat transfer rate

Q_{wl} Liquid phase wall heat transfer rate

$\langle R_f \rangle$ Fission rate

Re Reynolds number

$\mathbb{S}$ Modulus of the mean strain rate tensor

S Surface

$\mathbf{T}$ Viscous stress tensor

T Temperature

T_i Temperature of phase i

t Time

$\mathbf{u}$ Velocity

$\mathbf{u}_G$ Grid velocity

$\mathbf{u}_i$ Velocity of phase i

$\mathbf{u}_j$ Velocity of phase j

U_f Liquid phase specific internal energy

u_f Liquid phase bulk velocity

U_g Gaseous phase specific internal energy

u_g Gaseous phase bulk velocity

u_I Liquid-gas interface velocity

V Volume

$\mathbf{x}^n$ State of the system at time n

$\mathbf{x}$ Position vector

x Axial coordinate

X_n Volume fraction of n-th non-condensable gas

z Axial coordinate

Field operators

$\nabla\cdot$ Divergence operator $\rightarrow \nabla \cdot \mathbf{v} = \dfrac{\partial v_x}{\partial x} + \dfrac{\partial v_y}{\partial y} + \dfrac{\partial v_z}{\partial z}$

∇ Gradient operator $\rightarrow \nabla \mathbf{v} = \dfrac{\partial v_x}{\partial x}\mathbf{i} + \dfrac{\partial v_y}{\partial y}\mathbf{j} + \dfrac{\partial v_z}{\partial z}\mathbf{k}$

$\otimes$ Kronecker product

$\nabla\times$ Rotor operator $\rightarrow \nabla \times \mathbf{v} = \left(\dfrac{\partial v_z}{\partial y} - \dfrac{\partial v_y}{\partial z}\right) \cdot \mathbf{i} + \left(\dfrac{\partial v_x}{\partial z} - \dfrac{\partial v_z}{\partial x}\right) \cdot \mathbf{j} + \left(\dfrac{\partial v_y}{\partial x} - \dfrac{\partial v_x}{\partial y}\right) \cdot \mathbf{k}$

Chapter 1

Introduction

1.1 Motivation

Modelling complex systems with an acceptable level of physical detail while avoiding reliance on an excessive amount of computational power has always been a major challenge. The need for a compromise between model complexity and computational costs led to the development of one-dimensional codes, also known as system codes.

These codes necessarily embody simplified models that solve one-dimensional forms of the conservation equations for mass, energy and momentum, together with empirical correlations for the treatment of various phenomena ranging from frictional pressure drops to multiphase heat and mass transfer. System codes have been used for many decades and thanks to a long refinement process, their predictions are remarkably accurate in most circumstances. Examples of system codes are RELAP5-3D, TRACE and CATHARE for nuclear thermal hydraulics applications, OLGA for oil and gas pipeline calculations and GT-POWER for simulating internal combustion engines.

However, in spite of their high levels of accuracy in most circumstances, the one-dimensional approximations used by system codes also constitute their main limitation. In fact, to capture the behaviour of certain types of flow a full three-dimensional model is required. Examples of

these scenarios include flows in complex geometries and counter current flows of different fluid phases.

In principle, CFD codes provide a valid tool to accurately predict three-dimensional flow characteristics thanks to the higher level of detail in the implementation of physical models, albeit at a higher computational cost. An example of a state-of-the-art CFD simulation can be found in Popov et al. (2012), where a quarter of a Westinghouse-type PWR pressure vessel was modelled in "full" detail. The size of the model was 1.035 billion cells, requiring it be run on two of the NCCS clusters. The runs were made on 400 and 800 cores, taking about 220 minutes to complete 100 iterations on 400 cores and 150 minutes on 800 cores. The simulation only ran in steady state mode and required 1500 iterations to converge to acceptable results. However, it is remarked in the article that to estimate the overall clock time for the 1500 iterations a separate scaling assessment is necessary.

From these considerations it follows that building a full CFD model of a large complex system such as an industrial-size nuclear reactor would be impractical. However, using system codes where the flow can be approximated with high fidelity by a one dimensional model and CFD codes only where a higher level of detail is important would allow the simulation of such large, complex systems in reasonable time.

Computational efficiency is not the only motivation behind the choice of coupling CFD and system codes, however. Nuclear reactors are extremely large and complex systems with very strict and detailed design and review criteria. Examples of this attention to details are the documents of the STS series for operating and new reactors and the standard review plans for LWRs from the USNRC:

- USNRC (2007): Standard review plan for LWRs
- USNRC (2012a): Babcock and Wilcox plants STS
- USNRC (2012f): Westinghouse plants STS
- USNRC (2012b): Combustion Engineering plants STS

- USNRC (2012c): General Electric plants - BWR/4 STS
- USNRC (2012d): General Electric plants - BWR/6 STS
- USNRC (2012e): Westinghouse AP1000 plants STS

The details lost due to the one dimensional modelling of such complex systems lead to a design that can be overly conservative, with a great impact on design assessment procedures as well as building time and costs. The nuclear industry is well known for its slow innovation pace and the preference for well known and tested technologies, even if these are very old.

Computational tools are no exception to this practice. In particular, the need to develop a coordinated strategy to effectively accelerate the use of CFD as a main tool for nuclear system design and safety analysis was only mentioned in a lecture at the CFD4NRS-3 (Lee and Mousseau, 2010) workshop in 2010. Some guidelines on the use of CFD in nuclear reactor safety applications (NEA, 2007) were already published at the time, however they were not comprehensive and focused only on single phase problems.

These issues, together with the computational costs of a full reactor simulation are the main limitations to the use of CFD in nuclear reactor safety applications. Using coupled codes to model nuclear reactors can increase the detail level of computations and at the same time give a reason to accelerate the use of CFD in nuclear reactor safety applications.

1.2 Research hypotheses and objectives

Interesting work on coupling CFD and system codes has been performed on different aspects, from development and validation of different coupling algorithms to their application to various cases. Common considerations that can be identified in all coupling works are the following:

- Definition of variables calculated and exchanged by the two codes
- Conservation of mass and energy

- Frequency and location of information exchange
- Time step control

It is possible to differentiate the coupling paradigms in two main types, as explained in Weaver et al. (2002): Integrated solution matrices and coupled interfaces. In the first case, the solution matrices of the constitutive codes are integrated to generate a unique matrix that provides a numerically stable solution. However, there is no advantage in using this method because semi-implicit numerical schemes are available for both paradigms (Weaver et al., 2002). Furthermore, combining the matrices of the components leads to a rigid program, tailored for particular applications and less able to take advantage of the development of the single components.

The coupled interfaces paradigm allows a more flexible coupling between the codes thanks to the possibility of exchanging data at generic locations without the need of altering the solution matrices of the constitutive codes. This method is not bound to a particular application or to specific versions of the constitutive programs, provided that the information exchange paradigm is not altered in the development phase of the constitutive programs. This paradigm was used for single phase and multiphase coupled simulations, which constitute the foundations of this work.

A coupling technique that relies on domain decomposition instead of domain overlapping (Jeltsov et al., 2013) to be used in thermal hydraulic simulations of single and multiphase flows is proposed. The intention is to provide a tool that can be applied to the analysis of current reactors to better understand reactor behaviour and provide more representative analyses of current systems for safety assessment as well as providing a deeper insight in the development of future designs.

For single phase flows, existing benchmark calculations (Palazzi et al., 2014) are performed to assess the ability of the code to predict correct steady state scenarios; new benchmarks will be presented for transient calculations. For multiphase flows the two main benchmarks available are the BFBT (Neykov et al., 2006) and PSBT (Rubin et al., 2010), both organized by the OECD and the NEA. Five cases from the PSBT benchmark were used to evaluate the

performance of the coupling tool. One of the cases from the PSBT benchmark was evaluated in a closed loop configuration, to better test the stability of the code and to provide a scenario of industrial relevance. The two codes used for this work are STAR-CCM+ and RELAP5-3D. However, the method is general and can be adapted for any CFD/system code pair.

The proposed research objectives are as follows:

- Develop a coupling interface between RELAP5-3D and STAR-CCM+
- Implement an explicit data exchange algorithm for single and multiphase flows
- Demonstrate the coupled solver's capabilities by analysing proof-of-concept cases
- Apply the solver to an industrially relevant case

1.3 Thesis outline

The rest of the thesis is arranged as follows:

- **Chapter 2**: Extensive literature review on code coupling
- **Chapter 3**: Description of the codes that were chosen for this work and the equations they solve
- **Chapter 4**: Detailed description of the coupling scheme used in the simulations
- **Chapter 5**: Results of the validation tests performed to assess the single phase capabilities of the code
- **Chapter 6**: Results of the validation tests performed to assess the multiphase capabilities of the code
- **Chapter 7**: Simulation of a closed loop scenario
- **Chapter 8**: Discussion of results and future work

1.4 Publication list

The following papers have been published or submitted for publication as a result of the work presented in this thesis.

1.4.1 Journal papers

- A. Palazzi, M. J. Bluck, S. Lo, S. Slijepcevic, "*Simulation of single-phase and two-phase flows with coupled CFD and system codes: Steady and transient single-phase flows*", Under review in Nuclear Engineering and Design
- A. Palazzi, S. Lo, M. J. Bluck, "*Simulation of single-phase and two-phase flows with coupled CFD and system codes: The influence of the coupling interface location on the void distribution of a typical PWR subchannel*", Under review in Nuclear Engineering and Design
- A. Palazzi, S. Lo, M. J. Bluck, "*Simulation of single-phase and two-phase flows with coupled CFD and system codes: Simulation of a two phase flow in a forced convection loop heated by a PWR-like rod bundle*", Under review in Nuclear Engineering and Design

1.4.2 Conference proceedings

- A. Palazzi, M. J. Bluck, S. Lo, "*A coupled RELAP5-3D/STAR-CCM+ simulation for the calculation of friction factor in pipes*", Proceedings of ICAPP 2014, **14032**, ANS 2014
- A. Palazzi, M. J. Bluck, S. Lo, S. Slijepcevic "*Coupling RELAP5-3D and STAR-CCM+ for simulations of steady and transient single phase flows*", Proceedings of ICAPP 2016, **15733**, ANS 2016

Chapter 2

Literature review

There has been much effort over recent decades in the development of numerical methods specific to particular physical problems. The majority of effort has focused on the improvement of such techniques in terms of capability and overall performance. Understandably the nature of such methods is dictated by the nature of the precise physical problem. As a result there are a wide range of methods across a similarly wide range of disciplines. Even within a discipline there will be technology drivers that dictate very different approaches.

The majority of work in the development of numerical methods is focused in specific areas and is, as a result, guided by these demands. In contrast, the ultimate practical application of such methods is to whole systems rather than the narrowly focused discipline of the developer. From the perspective of a designer there are a bewildering array of tools, many of which are relevant, but when considered individually, do not address the demands of the design process.

Of particular relevance here is nuclear thermal hydraulics. Nuclear thermal hydraulics considers the flow of coolant through a nuclear reactor. At the smallest scales we are concerned with the growth of bubbles and their evolution and their impact on heat transfer mechanisms, whilst at larger scales we are concerned with overall pressure drop and bulk coolant and wall temperatures.

Detailed computational fluid dynamics, typically, though not always, described by finite volume

methods, is able to resolve the small-scale behaviour. In addition fully three-dimensional flows and the effects of turbulence can be addressed. This however comes at great cost in terms of computational resource. The need for those working in thermal hydraulics is to capture the behaviour throughout the entire plant and potentially the core.

It has been and remains for the foreseeable future impossible for computational fluid dynamics to address plant scale phenomena with the required resolution and performance appropriate for the wide range of cases that must be considered by the reactor designer.

For this reason nuclear thermal hydraulics has long made use of systems codes where the detailed three-dimensional small scale phenomena are essentially ignored, their effects taken into account by a combination of averaging and through the use of correlations. Of course in doing so the essential three-dimensional nature of flow and detailed smaller scale effects can be overlooked.

This loss of fidelity leads to the designer making conservative decisions at best and at worst missing potentially crucial features of the behaviour of the system.

The solution to this seeming paradox has long been considered. A means must be sought whereby individual codes can be combined in some way appropriate to their capabilities and performance.

Code coupling has been the subject of study for nearly two decades. It must be emphasised that it is difficult if not impossible to describe an entirely general approach. This is due to the sheer variety of methods involved. In this work we focus on the coupling of computational fluid dynamics with systems codes. The coupling of codes necessarily involves passing data between the domains in which the codes apply. Three broad issues arise; what data is passed, when is data passed and how is data passed.

What data is passed is very much dictated by the demands of the individual components. At the very least sufficient data is required for the problem to be well posed from the perspective of each code.

The means by which data is passed is dependent on the platforms, topology (Client-Server)

and the availability of source code or appropriate APIs.

The question of when data is passed is extremely important. In the context of the coupling of CFD and systems codes, it is usually expressed via terms such as “explicit” and “implicit”. These terms have meanings that are quite specific to code coupling and will be explained in the following section.

2.1 Terminology

Throughout the document the terms *explicit* and *implicit* will be extensively and continuously used. Hence the necessity of explaining how they are to be interpreted. In the subject of numerical methods the term explicit is used to indicate an algorithm that calculates the state of a system at a later time from the state of the system at the current time, as expressed by equation 2.1.

$$\mathbf{x}^{n+1} = f(\mathbf{x}^n) \tag{2.1}$$

Since phenomena like pressure wave propagation are treated explicitly in space, there is a limit concerning the time step dictated by the Courant condition (Eq. 2.2), which can be expressed as:

$$C = \frac{u\Delta t}{\Delta x} \leq C_{max} \tag{2.2}$$

This is known as sonic Courant limit, where $C_{max} = 1$ is the typical value imposed for an explicit algorithm. On the other hand, the term implicit is used to indicate an algorithm that finds a solution by solving an equation involving both the current state of the system and the later one, as expressed by equation 2.3.

$$\mathbf{x}^{n+1} = f(\mathbf{x}^n, \mathbf{x}^{n+1}) \tag{2.3}$$

This technique is not affected by the sonic Courant limit, but the drawback results in a simultaneous spatial coupling of all independent variables due to the use of implicit convected quantities (e.g. void fraction, density).

Halfway between the explicit and the implicit algorithm stands the semi-implicit discretisation technique. The semi-implicit technique implicitly treats sonic phenomena so that the sonic Courant limit can be bypassed in favour of the material Courant limit.

The material Courant limit is a less stringent condition than its sonic counterpart. Liles and Reed (1978) explain that in nuclear reactor safety problems the fluid velocities range from sonic to far subsonic and that their work sought a method that was not limited by the too stringent sonic Courant condition.

Reading the work of Liles and Reed (1978) and the numerical formulation of RELAP5-3D (INL, 2014a), it is possible to deduce that the material Courant limit is dependent on the algorithm used. In fact, Liles and Reed (1978) state that for their semi-implicit algorithm, which is based on the drift-flux method (Iishi and Hibiki (2011) and Tong and Tang (1997)), the material Courant limit takes the following form:

$$C = \frac{u_m \Delta t}{\Delta x} \tag{2.4}$$

Provided that u_m, the mixture velocity, is larger than the relative velocity between the two phases. In RELAP5-3D the material Courant limit is of the following form (INL, 2014a):

$$C = \frac{max(|\alpha_f u_f|, |\alpha_g u_g|)}{max(\alpha_f, \alpha_g)} \frac{\Delta t}{\Delta x} \tag{2.5}$$

The value of the material Courant limit in equation 2.5 is used by RELAP5-3D's semi-implicit solution algorithm. Furthermore, INL (2014c) states that RELAP5-3D is not the best software choice to model wave propagation problems and, in case the user wants to model those problems, to use a software that specialises in that field.

Unlike the typical usage of these terms, for code coupling there are different meanings to consider because they are used to address the way information are exchanged between the two codes.

In general, an explicit coupling method consists of a single data exchange per timestep, after which the global solution is advanced. On the other hand, in an implicit coupling method before the global solution can be advanced, multiple data exchanges per timestep can be performed. A criterion to decide whether the global solution can be advanced is needed when using an implicit method.

In the literature only explicit and semi-implicit coupling method are discussed. The reason for the absence of fully implicit methods can be attributed to the quantities exchanged by the codes that take part in the coupling. In fact, for a method to be implicit, all the quantities calculated by one code should be sent to the other. However, this is not always possible for two reasons:

- The volume of data to be exchanged might be excessively large
- The two codes are modelling different physical phenomena

Generic issues to be addressed when coupling different type of codes are needed to ensure the conservation laws are satisfied. Amongst them:

- Frequency and location of information exchange in the solution procedure
- Which code calculates which terms in the solution scheme
- Definition of variables exchanged by the two codes
- Timestep control

It has been explained (Aumiller et al., 2002) how different coupling techniques can affect the numerical stability of the integrated code, stressing the fact that the most important criteria

to be satisfied are the conservation of mass and energy. Another important statement is that the coupled code should not be characterised by more restrictive stability limits that any of its constitutive programs.

In the following sections uses of explicit and semi-implicit methods for thermal hydraulic coupling between CFD and system codes will be presented.

2.1.1 Explicit coupling

As described by Aumiller et al. (2001), their explicit coupling technique has been implemented by exchanging all the pertinent information at the beginning of the hydraulic solution for each timestep. As a consequence, this allows the two codes to perform their calculations with different timesteps, since the values are exchanged at fixed intervals in the solution, usually at each new step of the code with coarser time step (RELAP5-3D in most of the cases). For simplicity, they run both codes at a fixed time step.

As it is possible to see in figure 2.1 the flow of information is clear: from computational domain A the information stored in volume A1 and A2 are passed to the coupling volume connected to the computational domain B (CB1 and CB2). At the same time information from domain B, stored in the volumes B1 and B2, is passed back to the coupling volumes connected to domain A (CA1 and CA2).

Where:

- CAx, CBx: Coupling volumes connected to computational domains
- Ax, Bx: Computational domain volumes

The “explicitness” stands in the fact that the information passed back and forth are calculated at the same time step for both domains A and B. This means that a very strict Courant condition (sonic Courant condition) is needed in order to also catch all the phenomena concerning pressure wave propagation.

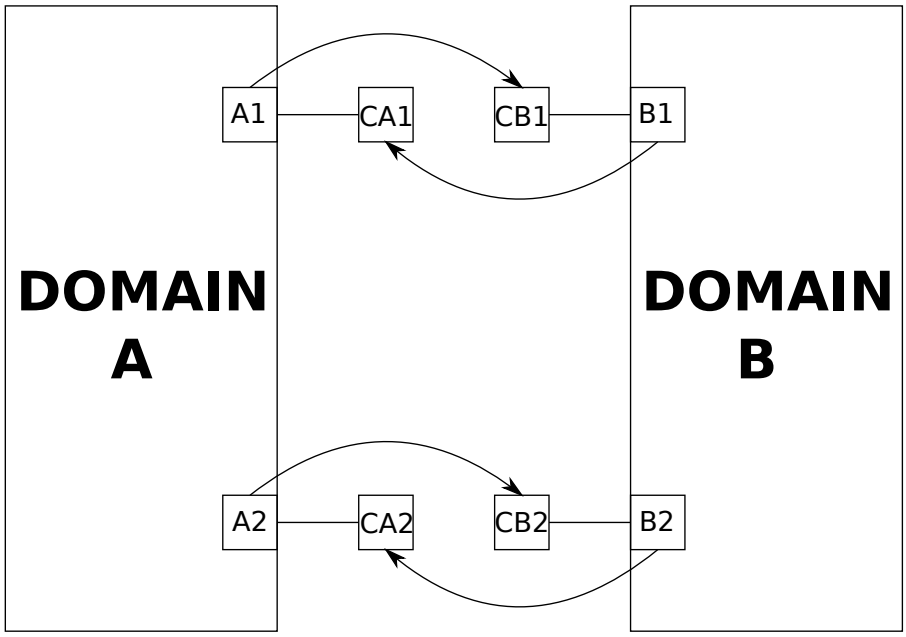


Figure 2.1: Explicit coupling information flow

Using the explicit coupling implementation, momentum is not conserved, due to the fact that both codes do not have all of the required information to correctly calculate the $u\nabla u$ term at the boundary. The error arising from this imperfection will be most significant whenever a significant velocity gradient is present in the coupling location (Aumiller et al., 2001).

2.1.2 Semi-implicit and fully implicit coupling

In the semi-implicit coupling technique (Weaver et al., 2002) the computational domains are identified with a master/slave approach (also known as client/server protocol). The designation of a code as master or slave does not depend on the direction of the fluid flow of the two domains. Both domains have specific roles to ensure that all the required information is provided.

Usually, system codes such as RELAP5-3D, are used as masters whereas single component programs such as CFD codes are most likely to be slaves. It is also pointed out that there are two requirements for a successful implementation of a semi-implicit coupling algorithm:

- Usage of new-time velocities and pressures to bypass the sonic Courant condition
- Same quantities have to be convected by the programs in order to conserve mass and

energy

As a consequence of the first requirement, new-time velocities must depend on new-time pressures on each side of the coupling junction. To satisfy this requirement, changes of pressure in all the volumes of the master process are represented as linear functions of mass and energy flow rates in the coupling junctions.

The coefficients of these linear relations are transmitted to the slave process which, through them, calculates the mass and energy flow rates across the coupling location and the change of pressure of the coupled volume in the master process. The flow rates are then transmitted back to the master process where they are used to compute the changes in pressure consistently with the slave process.

The semi-implicit methodology described in this section is the one implemented by Weaver et al. (2002). It is remarked in the same work that coupling methodologies for passing information between programs can be developed based on any numerical method.

Two further time differencing schemes are mentioned by Weaver et al. (2002): the fully implicit scheme and the SETS method. They are widely used in system codes and both are not restricted by the material Courant limit.

The fully implicit scheme is, as explained before, characterised by using old time parameters only in the representation of the temporal derivative. This is the opposite strategy used in the fully explicit method, where the new time parameters are used in the representation of the time derivative only.

The SETS (Stability-Enhancing Two-Step) method (Mahaffy, 1982) was created with the purpose of removing the restrictions imposed by the material Courant limit without incurring in the computational costs of a fully implicit numerical scheme. The SETS method consist of a standard set of basic mass, momentum and energy conservation equations and a set of stabilising equations for mass, momentum and energy. The results presented by Mahaffy (1982) show that the stabiliser equations of the SETS method add less that 20% to the computational cost

per cell per step of the basic equation set. Compared with the factor of 6 that is introduced by a fully implicit method it is evident why the SETS method is largely used in system codes.

Weaver et al. (2002) considered neither the SETS nor the fully implicit method as a base for their coupling algorithm because they identified problems associated with the conservation of mass and energy. This was due to the fact that old time quantities are not convected by either method, hence exchanges in between iterations of the programs being coupled would be needed to update convected quantities.

2.2 Client-server protocol

The use of coupled codes to analyse the dynamics of fluids in complex systems (e.g. nuclear reactors) is not yet widespread but some interesting work has been done. Most of the publications are in nuclear engineering journals, but use of this technique has been demonstrated for the oil industry as well.

There is no standard procedure on which code or which coupling procedure to use. The only common denominator seems to be the use of socket libraries to exchange data, implementing a client/server protocol. Various approaches can be found in literature:

- PVM: external software that has the function of managing the information exchange between the codes to be coupled
- UDFs: ANSYS FLUENT feature that allows the user to implement new pieces of code and use them in the main CFD package
- Java API: STAR-CCM+ feature similar to UDFs, does not allow the user to implement new pieces of code, but allows him to manipulate the interaction between the existing instructions only
- Co-Simulation API: CD-adapco recently added an API in STAR-CCM+ with the specific purpose of making data exchange between STAR-CCM+ and other programs easier

2.2.1 FLUENT/RELAP5-3D coupling

In the literature, two examples of using this software to couple system codes with CFD codes can be found: the analysis of the hot gas flow in the outlet plenum of a VHTR (Anderson et al., 2008) and single phase mixing studies (Bertolotto et al., 2009).

Analysis of the hot gas flow in the outlet plenum of a VHTR

In this example (Anderson et al., 2008), the system code was used to calculate the different radial flow conditions at the outlet region of the reactor core. These exit conditions were used as the inlet boundary condition for the CFD model of the outlet plenum. The computational strategy adopted involved constructing two separate models for the system code and the CFD code, which would be coupled in a second instance. A standalone steady state simulation for both models was performed.

Then the two models were coupled together from the convergence state, thus reducing the required computational time. The problem analysed seems to be only a steady-state convergence test, but it is not very clear.

Use of User Defined Functions (UDFs)

This kind of simulation (Yan, 2011) is very interesting under both the profiles of coupling technique used and reference case studied. As pointed out by the author, merging two different codes into an integral one needs access to the source code of both programs, whereas interfacing two codes through a third program (PVM) requires a considerable programming effort in order to maintain consistency between the two codes and the access to the source code of the CFD software.

Hence a third strategy has been employed: the use of FLUENT's UDFs. In this approach FLUENT is the master (client) code and RELAP5-3D is the slave (server). This work has been carried out using the following semi-implicit coupling scheme: the pressure boundary condition

used at the outlet of the upstream code is based on the solution at the previous time step. The flow rate information passed to the downstream code is an information based on the current time step. As a result, the solution for new time step depends on both previous and current time step values.

A very important consideration made by the author is the fact that the time step necessary to overcome numerical instabilities in the coupled code simulations is still much smaller than the time step for simulations using the CFD or the system code standalone. As a consequence, the coupled code in its current state is not very practical for slow transient scenarios like SBLOCAs in PWRs, which may take several hours to evolve.

Validation of the coupling strategy has been made through a simple pipe flow simulation (for further reference please see Yan (2011)). The reference case studied are a large scale simulation of a simplified PWR design based on the AP1000 reactor by Westinghouse and a coupled simulation of a gas turbine and a Modular Helium Reactor.

2.2.2 CFX/TRACE coupling

In this example, it is stressed how a system code (TRACE) is not suitable for the particular class of problems which are mixing phenomena, because they are strongly 3D and influenced by turbulence. Two different coupling schemes, explicit and semi-implicit, have been tested with a test case made by two loops connected with a double T-junction.

Here the definition of explicit and semi-implicit is given in a slightly different way from the one that has been pointed out earlier (for further reading please see Bertolotto (2011)).

For this case experimental results were already present and have been used to test the accuracy of the simulations. It is also shown how the coupled system gives better results if compared with the standalone system code simulation, but there is still disagreement if compared with experimental results. This is due to the numerical scheme used in TRACE, strongly affected by numerical diffusion.

2.2.3 STAR-CCM+ coupling

OLGA

The path chosen by the development team of STAR-CCM+ at CD-adapco uses an explicit coupling technique in an embedded tool in the CFD software. Most of the work has been done with the system code OLGA (Xing et al., 2011), used in the oil and gas industry.

Differently from the other coupled simulations it has been pointed out that, both upstream and downstream of the region of interest for the CFD analysis, a 1 m pipe section has been included in the model. This obviously to allow the flow information to have coherent properties when arriving in the interested zones. Differently from the other applications, coupling has been used here to analyse an air-water multiphase system.

Using the slug tracking module present in OLGA the analysis has been performed, with the slug characteristic parameters obtained from OLGA and the forces acting on the bend, liquid hold-up in the bend and pressure on the bend wall were provided by STAR-CCM+. All the results have been compared with experimental data.

RELAP5-3D

A capability test on STAR-CCM+ has been performed at Texas A&M University (Rodriguez, 2012), using STAR-CCM+ for the parts where complex flow patterns were expected and RELAP5-3D for the rest of the system.

According to the author, the STAR-CCM+/RELAP5-3D coupling showed inability to perform flow coupling simulations and to handle closed loop systems. However, thermal coupling simulations were successful and showed results which were congruent to the physical expectations.

In this thesis it is pointed out that the information are communicated between the two codes only when these are set to run transient simulations. It is also mentioned that the code is stable for compressible fluids, but its stability is not as good when incompressible fluids are used.

2.3 Domain overlapping approach

All the previous examples used a domain decomposition approach, which consists of substituting the part of the circuit that has to be modelled with the CFD program. Data are sent from and to the boundaries of the open circuit. In the domain overlapping approach the location of the data transfer is the same, but the two models run simultaneously. The value of the desired variable at the exit of the CFD model is overwritten on the system code model, thus continuing the simulation with the new value.

This approach has been possible thanks to the Java API tool already present in STAR-CCM+ (Jeltsov et al., 2013), which allows the user to build STAR-CCM+ macros. A STAR-CCM+ macro is a Java program that is compiled and executed within the STAR-CCM+ workspace.

Both transient and steady state calculations have been performed with this approach, allowing the authors to use a 2D axisymmetric model for the CFD calculations, normally not available when using the coupling tool embedded in STAR-CCM+ and probably a different strategy in the time step advancement control.

Significant feedbacks between 3D and 1D phenomena are identified via experimental means and captured by the coupled codes. In the future work an algorithm with an adaptive coupling time step and various transients are mentioned.

Another example of this approach can be seen in Baviere et al. (2013), where the two codes being coupled are the system code CATHARE and its CFD counterpart TRIO-U (both developed at CEA). The strategy adopted here is based on using an API implemented in both codes, which looks very similar to the approach followed in Aumiller et al. (2001) and Aumiller et al. (2002).

For the coupling validations a semi-implicit coupling scheme has been used, whereas an explicit coupling scheme was chosen to model the industrially relevant case.

Domain overlapping can be seen used in other contexts, such as multigrid methods for CFD. The basic assumption behind multigrid methods is that for a chosen discretisation there is an iterative solver that converges such that the higher frequency component of the error are

damped faster than the lower frequency ones.

As a consequence, after few initial iterations on the finest grid it would be more efficient to continue iterating over a coarser grid. This is achieved by transferring the fine-grid solution to the coarse-grid solution. The lower frequency components of the error on the fine grid appear as high frequency components on the coarse grid, hence it will reduce rapidly. Once the solution on the coarser grid is obtained, it needs to be transferred back to the finer grid. Since the number of points of the two grids is not the same, the solution needs to be interpolated. Once the interpolation step has finished, some more iterations on the finer grids are needed to eliminate any errors that might have been introduced by transferring the solutions back and forth on the grids of different sizes.

This procedure can be applied to any number of coarseness levels, with any kind of cycling through the various levels (W cycle, F cycle etc.) as explained in Jimack (2007) and Versteeg and Malalasekera (2007).

Jimack (2007) further explains how parallel implementations of multigrid methods are achieved through a geometric decomposition of the computational domain. This feature is remarkably similar to the domain decomposition in code coupling. In parallel multigrid methods, each processor is responsible of implementing the solution algorithm on its subdomain, making use of additional cells situated to either side of the subdomain to ensure communication at various stages of the multigrid algorithm.

In domain decomposition for coupled codes the same things happen: the codes taking part in the coupled simulations are the dual component of the processors, the additional cells to either side of the subdomains are represented by the storage allocated for the data to be exchanged between the two codes. Lastly, the muultigrid algorithm running on each processor is the solver algorithm of the coupled codes taking part in the simulation.

In essence, parallel multigrid algorithms present both features for domain decomposition and overlapping.

2.4 Summary

A description of explicit and semi-implicit methods, together with the master/slave (or client/server) approach was given.

Coupling terminology concerning data exchange mechanism clashed with the one of numerical methods when mentioning explicit and semi-implicit methods. In code coupling an explicit data exchange consists of data being exchanged between the codes once per timestep, after which the global solution is advanced. On the other hand, in an implicit coupling method multiple data exchanges can be performed before the global solution can be advanced. A criterion to decide whether the global solution can be advanced is needed when using a semi-implicit method.

In spite of a clash in terminology, a duality between coupling methods and numerical methods emerged: In fact, explicit methods are easier to implement but can be unstable. On the other hand, semi-implicit methods offer a greater stability albeit at a higher computational cost.

A peculiarity of the coupling methodologies is the division between domain decomposition and domain overlapping. The results given by both approaches are satisfactory, however the domain overlapping methodology seems more complex to implement.

Albeit no coupling methodology can yet be considered as the industry standard, the master/slave (client/server) approach, together with the use of APIs or third party software (e.g. PVM), seem to be a common denominator within the methods presented in this chapter.

Using an API with a client server approach was chosen as the coupling method for this work. The codes used in the coupling will be presented in chapter 3 whereas details on the coupling methodology will be presented in chapter 4.

Chapter 3

Overview of the codes used for coupling

The first step needed to develop a coupling tool is to choose which code to couple. For this work the choice fell on RELAP5-3D and STAR-CCM+.

CD-adapco is one of the major companies in CFD for nuclear applications. Among the vast number of projects that use STAR-CCM+, CD-adapco's flagship product, one is worth mentioning for its first-of-a-kind nature: the simulation of flow inside a PWR vessel with full fuel assemblies details (Popov et al., 2012).

As explained in chapter 1, this work consisted in a state-of-the-art simulation of a quarter of a Westinghouse-type PWR vessel, where meticulous attention was given to all of the vessel's internal features. The size of the CFD model was 1.035 billion cells, requiring two of the clusters of the NCCS to run in reasonable times. Because of the complexity of the model, only steady state analyses were performed.

This, together with CD-adapco being the sponsor of this work, made the choice of STAR-CCM+ as the CFD code for the coupled analyses obvious.

The widespread acceptance of RELAP5-3D, which can arguably be considered the de-facto standard code for nuclear thermal hydraulic analyses, together with existing coupling capabil-

ities between STAR-CCM+ and RELAP5-3D (Palazzi et al. (2014) and CD-adapco (2016)) made the choice fall onto RELAP5-3D as the system code for the coupling work.

In this chapter the STAR-CCM+ and RELAP5-3D are presented and a brief description of the general features, together with the basic equations, the numerics of each code and considerations on boundary conditions for coupled simulations is given.

3.1 Overview of RELAP5-3D

The RELAP suite has been developed at INL under sponsorship of the US Department of Energy, the USNRC, members of the International Code Assessment and Applications Program, members of the Code Applications and Maintenance Program and members of the International RELAP5 User Group (INL, 2014a). The code is used widely for simulation of steady and transient behaviour of LWRs. Among these scenarios there are LOCA, ATWS, LOFA, LOOP, LOHS, station blackout and turbine trips. However, it can also be used for a wide variety of hydraulic and thermal transient not necessarily related to nuclear systems. RELAP5-3D is the latest software of the suite and it was released in 1995. It is written in FORTRAN 95 with a top-down structure. The three main blocks that constitute RELAP5-3D are shown in fig. 3.1.

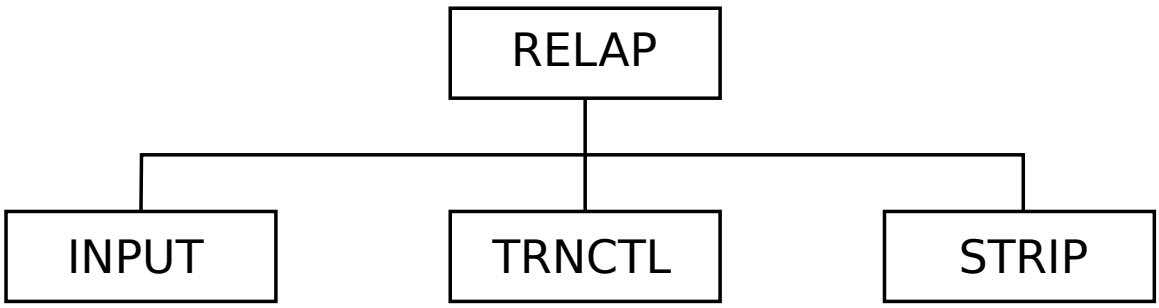


Figure 3.1: RELAP5-3D constitutive blocks

The input block processes all the information from the inputs and prepares the data required for all the program options. The strip block is used to extract data from output files. The transient/steady-state block (TRNCTL) handles all the options related to transient/steady-state calculations. The transient/steady-state block also presents a top-down structure (Fig. 3.2).

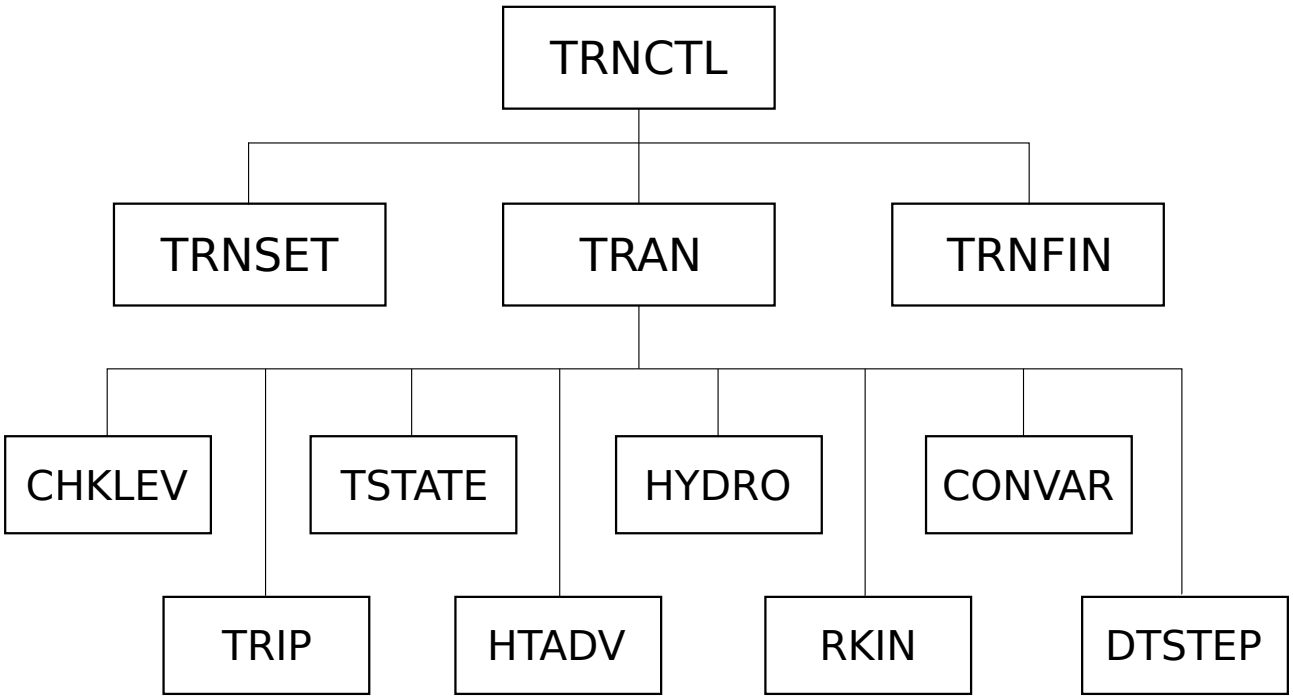


Figure 3.2: Transient/steady-state block structure - Adapted from INL (2014a)

The TRNSET block finalises the links between data blocks, sets up arrays to control the matrix solution and returns computer memory that is not needed. The TRAN block controls the time advancement of the solution. After the advancements are terminated the TRNFIN block releases space for the dynamic data blocks that are no longer needed. The TRAN block is where nearly all the execution time is spent and it is comprised of several modules:

- CHKLEV: level module, controls the movement of two phase mixture levels between volumes
- TRIP: trip system module, evaluates logical statements for components
- TSTATE: equation of state boundary volume module, calculates the thermodynamic state of the fluid in each boundary volume
- HTADV: heat structure module, advances the heat transfer solution
- HYDRO: hydrodynamic model, advances the hydrodynamic solution
- RKIN: reactor kinetics module, advances the core reactor kinetics

- CONVAR: control system module, provides capabilities of simulating control systems typically used in hydrodynamic systems

- DTSTEP: time step control module, determines time step size and if the transient advancement should be terminated

3.1.1 Main components

A RELAP5-3D simulation is based on a one-dimensional model of the transient flow for a steam-water non-condensable mixture. A physical system is simulated by constructing a network of volume components connected by junction components.

As will be explained in section 3.1.5, volume components represent mass and energy control volumes whereas junction components represent momentum control volumes, thus generating a staggered grid.

In RELAP5-3D every component has its own unique characteristics, however hydrodinamic components can be categorised in the following two main groups:

- Volumes

- Junctions

Components are specified by writing an input file called "deck" consisting of a set of cards, a terminology that has been inherited from the first versions of the program, which used decks of punch cards. In appendix C an accurate description on how to construct a RELAP5-3D input deck is given.

A brief description of the general features of each group will be given hereafter. A more detailed description of how they are used in modelling a real problem is given in section 3.3.

3.1.2 Volumes

Volumes can be seen as stream tubes having inlet and outlet junctions and an associated direction which is positive from the inlet to the outlet.

Volume orientation defines which face is the inlet and which face is the outlet: for example, in a volume with a positive vertical elevation change, its inlet is at the lowest elevation of the volume. In case the vertical elevation change is negative, the inlet is at the highest elevation of the volume instead.

Each volume component is characterised by the following base quantities:

- Length
- Volume
- Flow area
- Vertical angle
- Wall roughness
- Hydraulic diameter
- Flags
- Initial conditions

Flags are used to activate particular modules or to specify different settings (e.g. whether boron is present or not or the type of fluid in the volume)

Different volume components have different specific quantities. For example, a TDV component, which is used to impose pressure or temperature boundary conditions, has the possibility of imposing a time-dependent pressure or temperature on the system. On the other hand, a pipe component can be subdivided up to 99 volumes.

An example of a TDV in an input deck can be seen in the following listing.

```
//          NAME      TYPE
1110000     pres_bc   tmdpvol

//          AREA    LENGTH    VOLUME
1110101     1.      0.        10.
//          HORIZONTAL ANGLE     VERTICAL ANGLE     ELEVATION CHANGE
1110102     0.                   0.                 0.
//          HYDRAULIC DIAMETER     FLAGS
1110103     0.                     0

//          TDV DATA CONTROL FLAG
1110200     3
//          TIME     PRESSURE     TEMPERATURE
1110201     0.       147.e5       450.
```

Listing 3.1: TDV example

3.1.3 Junctions

Junctions are the components that connect the various volumes in the network representing the modelled thermal hydraulic system.

Junctions, similarly to volumes, have a positive or negative direction depending on the volumes they connect to each side. In fact, the main characteristic of junctions is the presence of the volume id of the upstream and downstream components. The direction of junctions and, consequently, of the flow, is positive for the fluid going from the upstream component to the downstream component.

Each junction is characterised by the following base quantities:

- Upstream volume id
- Downstream volume id
- Flags
- Flow area
- Initial conditions

Flags, similarly to volumes, are used to activate particular modules or specify different settings (e.g. whether the volumes at the extremities of the junction are connected by a cross section with an abrupt change or which choke model to use during calculations).

Different junction components have different specific quantities. For example, a TDJ component, which is used to impose velocity or mass flow rate boundary conditions has the possibility to impose a time-dependent velocity or mass flow rate. On the other hand, a valve component can be associated to a logical variable, called trip, in order to evaluate whether the valve is open or closed.

An example of a TDJ in an input deck can be seen in the following listing.

```
//          NAME      TYPE
1120000     inlet     tmdpjun

//          FROM          TO              AREA
1120101     111010000     113000000       0.5

//          TDJ DATA CONTROL FLAG
1120200     1
//          TIME    LIQUID M_FLOW    GAS M_FLOW    INTERFACE VELOCITY
1120201     0.      20.              12.5          0.
```

Listing 3.2: TDJ example

3.1.4 Basic equations

RELAP5-3D solves eight equations for eight primary dependent variables, namely pressure, gas internal energy, liquid internal energy, void fraction, liquid velocity, gas velocity, non-condensable quality and boron density. In case boron and non-condensables are not present the number of equation reduces to six. The secondary dependent variables used in the equations are: liquid density, gas density, liquid temperature, gas temperature, saturation temperature and non-condensable mass fraction in non-condensable gas phase. The basic equations used by RELAP5-3D are presented hereafter. The meaning of the symbols is obvious and will not be explained in the text. However, if clarifications are needed, an exhaustive nomenclature is presented at the beginning of this manuscript. This chapter does not intend to substitute

RELAP5-3D's theory guide. More information are available on the official documentation (INL, 2014a).

Mass continuity

- Gas phase

$$\frac{\partial}{\partial t}(\alpha_g \rho_g) + \frac{1}{A}\frac{\partial}{\partial x}(\alpha_g \rho_g u_g A) = \Gamma_g \tag{3.1}$$

- Liquid phase:

$$\frac{\partial}{\partial t}(\alpha_f \rho_f) + \frac{1}{A}\frac{\partial}{\partial x}(\alpha_f \rho_f u_f A) = \Gamma_f \tag{3.2}$$

- Non-condensables in gas phase

$$\frac{\partial}{\partial t}(\alpha_g \rho_g X_n) + \frac{1}{A}\frac{\partial}{\partial x}(\alpha_g \rho_g X_n u_g A) = 0 \tag{3.3}$$

- Boron in liquid phase

$$\frac{\partial \rho_b}{\partial t} + \frac{1}{A}\frac{\partial}{\partial x}(\rho_b u_f A) = 0 \tag{3.4}$$

$$\rho_b = \alpha_f \rho_f C_b \tag{3.5}$$

Momentum balance

- Gas phase

$$\alpha_g \rho_g A \frac{\partial u_g}{\partial t} + \frac{1}{2}\alpha_g \rho_g A \frac{\partial u_g^2}{\partial x}$$

$$= -\alpha_g A \frac{\partial p}{\partial x} + \alpha_g \rho_g B_x A - \alpha_g \rho_g A(\mathbf{FWG}) u_g + \Gamma_g A(u_I - u_g) \tag{3.6}$$

$$-\alpha_g \rho_g A(\mathbf{FIG})(u_g - u_f) - C\alpha_g \alpha_f \rho_m A \left[\frac{\partial (u_g - u_f)}{\partial t} + u_f \frac{\partial u_g}{\partial x} - u_g \frac{\partial u_f}{\partial x}\right]$$

- Liquid phase

$$\alpha_f \rho_f A \frac{\partial u_f}{\partial t} + \frac{1}{2}\alpha_f \rho_f A \frac{\partial u_f^2}{\partial x}$$

$$= -\alpha_f A \frac{\partial p}{\partial x} + \alpha_f \rho_f B_x A - \alpha_f \rho_f A(\mathbf{FWF}) u_f + \Gamma_g A(u_I - u_f) \tag{3.7}$$

$$-\alpha_f \rho_f A(\mathbf{FIF})(u_f - u_g) - C\alpha_f \alpha_g \rho_m A \left[\frac{\partial (u_f - u_g)}{\partial t} + u_g \frac{\partial u_f}{\partial x} - u_f \frac{\partial u_g}{\partial x}\right]$$

The force terms on the right sides of the momentum equations are, respectively, the pressure gradient, the body force (i.e., gravity and pump head), wall friction, momentum transfer due to interface mass transfer, interface frictional drag and force due to virtual mass.

The terms **FWF** and **FWG** are the wall frictional drag coefficients. They are linear with respect to velocity and are the product of the friction factor, the frictional reference area per unit volume and the magnitude of the fluid bulk velocity. Their units are s^{-1}.

The terms **FIF** and **FIG** are the interphase frictional drag coefficients. Either the drift flux model or the drag coefficient model is used, according to the flow regime. The formulation for the interphase frictional depends on which model is used. More details on the formulations are available on INL (2014a). Their units are s^{-1}.

The virtual mass coefficient C is a function of the flow regime. A value of 0.5 is appropriate for bubbly or dispersed flows, whereas a value of 0 is appropriate for stratified flows.

Energy balance

- Gas phase

$$\frac{\partial}{\partial t}\alpha_g \rho_g U_g + \frac{1}{A}\frac{\partial}{\partial x}\alpha_g \rho_g U_g u_g A = -p\frac{\partial \alpha_g}{\partial t} - \frac{p}{A}\frac{\partial}{\partial x}\alpha_g u_g A + Q_{wg} + Q_{ig} + \Gamma_{ig} h_g^* + \Gamma_w h_g' + \mathbf{DISS}_g \tag{3.8}$$

- Liquid phase

$$\frac{\partial}{\partial t}\alpha_f\rho_f U_f + \frac{1}{A}\frac{\partial}{\partial x}\alpha_f\rho_f U_f u_f A = -p\frac{\partial \alpha_f}{\partial t} - \frac{p}{A}\frac{\partial}{\partial x}\alpha_f u_f A + Q_{wl} + Q_{il} - \Gamma_{ig}h_f^* - \Gamma_w h_f' + \mathbf{DISS}_f \tag{3.9}$$

The terms $\mathbf{DISS}_f$ and $\mathbf{DISS}_g$ are the sums of the dissipation effects due to wall friction, pumps and turbines. The dissipation effects due to interface mass transfer, interface friction, and virtual mass are neglected since these terms are small in magnitude in the energy equation. In the mass and momentum equations, interface mass transfer, interface friction, and virtual mass are important and are not neglected. Their units are Wm^{-3}. The wall friction dissipations are defined as follows:

$$\mathbf{DISS}_f = \alpha_f\rho_f\,\mathbf{FWF}\,u_f^2 \tag{3.10}$$

$$\mathbf{DISS}_g = \alpha_g\rho_g\,\mathbf{FWF}\,u_g^2 \tag{3.11}$$

Heat conduction

RELAP5-3D uses the integral form of the heat conduction equation.

$$\int\int\int_V \rho\, c_p(T,\mathbf{x})\frac{\partial T(\mathbf{x},t)}{\partial t}\,dV = \int\int_S k\,(T,\mathbf{x})\nabla T(\mathbf{x},t)\cdot\mathbf{n}\,dS + \int\int\int_V \xi(\mathbf{x},t)\,dV \tag{3.12}$$

Reactor kinetics

- Time evolution of the neutron population

$$\frac{dn(t)}{dt} = \frac{\rho(t)-\beta}{\Lambda}n(t) + \sum_{i=1}^{N_d}\lambda_i C_i(t) + \xi \tag{3.13}$$

- Time evolution of the neutron precursors concentration

$$\frac{dC_i(t)}{dt} = \frac{\beta f_i}{\Lambda} n(t) - \lambda_i C_i(t) \tag{3.14}$$

- Neutron flux

$$\varphi(t) = n(t)V \tag{3.15}$$

- Fission rate

$$\langle R_f \rangle = V \Sigma_f \varphi(t) \tag{3.16}$$

- Fission power

$$P_f = E_f \langle R_f \rangle \tag{3.17}$$

3.1.5 Numerics

RELAP5-3D implements a semi-implicit[1] time advancement hydrodynamic model discretised with the finite difference method. The method is first order accurate in both time and space.

It is possible for the user to select a nearly-implicit[1] time advancement scheme, however this option is only suitable for steady-state cases and quasi-steady (i.e. slowly varying) transients. Both semi- and nearly-implicit schemes BPLU solver (INL, 2014a) to calculate the solution of the hydrodynamic system.

Optionally the user can select a Harwell direct solver and a Krylov solver, the GMRES method preconditioned with the BPLU, called PGMRES. The latter was designed for very large problems. The only terms that are evaluated implicitly are the velocities in mass and energy transport terms, the pressure gradient in the momentum equations and interface momentum and mass exchange terms.

[1]The meaning is that of traditional numerical methods.

The two-fluid differential equations for the hydrodynamic model are formulated in terms of volume- and time-averaged parameters of the flow. Phenomena depending on transverse gradients such as friction and heat transfer are formulated in terms of bulk properties using empirical correlations for the transfer coefficients.

These differential equations have complex characteristic roots, thus giving the system a partially elliptic behaviour. This constitutes an ill-posed problem. Several stabilising techniques are used to ensure the problem is well posed numerically.

Among these there is the introduction of artificial viscosity terms in the numerical formulation of the equations to damp the high frequency spatial components of the solution, the selective implicit evaluation of spatial gradients at the new time, donor formulation for mass and energy fluxes and donor-like formulation for momentum fluxes. The term *donor-like* means that the momentum flux formulation consists of a centred formulation for the spatial velocity gradients to which a numerical viscosity term is added.

The discretised equations for the hydrodynamic model are based on a "room and doors" concept. The accumulation rate of mass and energy is equated to the rate of mass and energy entering the control volume through the cell boundaries minus the rate of mass and energy leaving the volume plus the source terms.

Mass and energy properties are averaged over the volume and require knowledge of velocities at the boundaries of these volumes. It is convenient to define velocities through control volumes for momentum fluxes centred on the boundaries of mass and energy control volumes. Each control volume has a constant cross section. With these formulation control volumes for mass and energy represent the "rooms" of the system whereas the control volumes for the velocity represent the "doors" (Fig. 3.3).

This approach results in a staggered spatial mesh where scalar properties are defined in the rooms and vector quantities are defined on the upstream doors. This translates into an upwind differencing formulation.

In RELAP5-3D mass and energy control volumes are represented by "volume" components

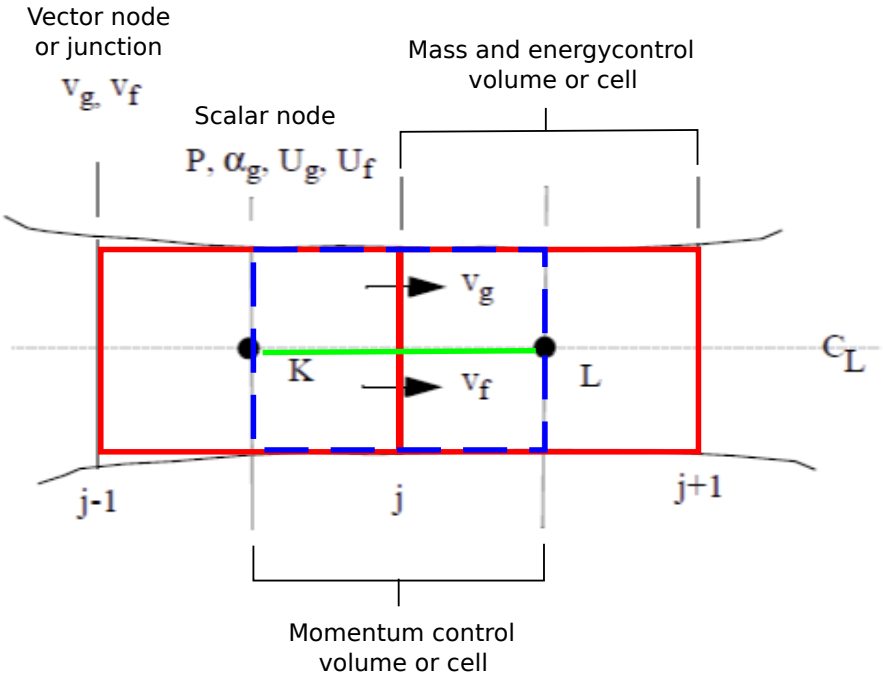


Figure 3.3: Rooms and doors concept - Adapted from INL (2014a)

(e.g. pipe, annulus, single volume etc.) whereas momentum control volumes are represented by junction components (time dependent junction, single junction etc.).

Calculations of heat transfer across solid boundaries of hydrodynamic volumes are performed through a RELAP5-3D component called heat structure. This component has general modelling capabilities and is able to represent several scenarios, including fuel pins or plates heated electrically or through nuclear reactions, heat transfer across steam generator tubes, pipe walls and vessel walls. Heat structures can be represented in rectangular, cylindrical or spherical geometry and, differently from hydrodynamic components, they present also a radial dimension.

Finite differences are used to discretise heat structures. Each radial interval can use a different spacing, a different material and a different spatial dependence of the internal heat source. Time dependence of the heat source is obtained by either using the reactor kinetics module or from heat source tables. Boundary conditions allow symmetry or insulation conditions, heat transfer correlations and tables of various quantities (e.g. heat flux, power etc.) versus time.

Heat structures allow the use of several models to represent various scenarios that can occur inside a fuel pin. These models are listed below.

- Gap conduction model
- Radiation enclosure model
- Conduction enclosure model
- Metal-water reaction model
- Cladding deformation model

RELAP5-3D uses the Crank-Nicolson method to solve the heat transfer equation.

3.2 Overview of STAR-CCM+

STAR-CCM+ is CD-adapco's flagship product, "*an entire engineering process for solving problems involving flow (or fluids or solids), heat transfer and stress*". The software is used in a vast spectrum of applications, spanning from nuclear to electronics. Since the range of problems and physics phenomena STAR-CCM+ can model is broad, this overview will present only those models relevant to applications in nuclear thermal-hydraulics.

3.2.1 General features

STAR-CCM+ is based on object-oriented programming technology. It relies on a client-server architecture that minimises the effort of the user during the all the phases of the simulation, from meshing to post-processing. The code uses a finite volume and body fitted mesh approach for the domain discretisation and allows the user to perform steady-state and transient calculations.

3.2.2 STAR-CCM+ models

Given the large number of models implemented by STAR-CCM+, providing a description for each model would be cumbersome and out of scope. Instead, only the equations of the main

models used in the simulations will be presented hereafter. When necessary, a distinction between single and multiphase flows will be presented. The equations presented hereafter were adapted from CD-adapco (2016), Versteeg and Malalasekera (2007), Prosperetti and Tryggvason (2009) and Pope (2010).

An important mention is required for the grid velocity, which is present in all the equations of sections 3.2.3 and 3.2.4. STAR-CCM+ offers moving mesh capabilities, which allow for the user to specify an overlapping region between two grids: a static one and a moving one. The grid velocity is the velocity of the moving grid.

In this work the moving mesh capabilities were not used, however, being one of the distinguishing features of STAR-CCM+, it was deemed important to mention it when presenting the equations of the program.

A possible application in the nuclear field could be the analysis of flow in fuel assemblies when the control rods are lowered following a reactor scram.

3.2.3 Conservation laws of fluid motion

Mass continuity

- Single phase flows

$$\frac{\partial}{\partial t}\rho + \nabla \cdot [\rho(\mathbf{u} - \mathbf{u}_G)] = \xi \tag{3.18}$$

- Multiphase flows

$$\frac{\partial}{\partial t}\alpha_i\rho_i + \nabla \cdot [\alpha_i\rho_i(\mathbf{u}_i - \mathbf{u}_G)] = \sum_{j\neq i}(m_{ij} - m_{ji}) + \xi_i \tag{3.19}$$

$$\sum_i \alpha_i = 1 \tag{3.20}$$

Momentum balance

- Single phase flows

$$\frac{\partial}{\partial t}\rho\mathbf{u} + \nabla\cdot[\mathbf{u}\otimes(\mathbf{u}-\mathbf{u}_G)] = -\nabla p + \nabla\cdot\mathbf{T} + \mathbf{F} \tag{3.21}$$

- Multiphase flows

$$\begin{aligned}\frac{\partial}{\partial t}\alpha_i\rho_i\mathbf{u}_i + \nabla\cdot[\alpha_i\rho_i\mathbf{u}\otimes(\mathbf{u}_i-\mathbf{u}_G)] = \\ -\alpha_i\nabla p + \alpha_i\rho_i\mathbf{g} + \nabla\cdot\left[\alpha_i\left(\tau_i+\tau_i^t\right)\right] + \mathbf{M}_i + \mathbf{F}_i + \xi_i + \sum_{i,j}\left(m_{ij}\mathbf{u}_j - m_{ji}\mathbf{u}_i\right)\end{aligned} \tag{3.22}$$

Energy balance

- Single phase flows

$$\frac{\partial}{\partial t}\rho E + \nabla\cdot[\rho H(\mathbf{u}-\mathbf{u}_G)] + \nabla\cdot[\mathbf{u}_G p] = -\nabla\cdot[\dot{\mathbf{q}}''] + \nabla\cdot[\mathbf{T}\cdot\mathbf{u}] + \mathbf{F}\cdot\mathbf{u} + \xi \tag{3.23}$$

$$E = H - p/\rho \tag{3.24}$$

$$H = h + |\mathbf{u}^2|/2 \tag{3.25}$$

- Multiphase flows

$$\begin{aligned}&\frac{\partial}{\partial t}\alpha_i\rho_i E_i + \nabla\cdot[\alpha_i\rho_i H_i(\mathbf{u}_i-\mathbf{u}_G)] + \nabla\cdot[\alpha_i\mathbf{u}_G p] = \\ &\nabla\cdot\left[\alpha_i k_i^{eff}\Delta T_i\right] + \nabla\cdot[\mathbf{T}_i\cdot\mathbf{u}_i] + \mathbf{F}_i\cdot\mathbf{u}_i + \xi_i + \sum_{j\neq i}Q_{ij} + \sum_{\gamma}Q_i^{\gamma} + \sum_{j\neq i}\left(m_{ij}-m_{ji}\right)h_{\gamma,i}\end{aligned} \tag{3.26}$$

$$k_i^{eff} = k_i + \frac{\mu_{t,i} c_{p,i}}{\sigma_{t,i}} \tag{3.27}$$

$$E_i = H_i - p/\rho_i \tag{3.28}$$

$$H_i = h_i + |\mathbf{u}_i|/2 \tag{3.29}$$

$$h_i(T_i) = h_i^{REF} + \int_{T_i^{REF}}^{T_i} c_{p,i}(\theta) d\theta \tag{3.30}$$

3.2.4 Turbulence models

STAR-CCM+ provides several turbulence models. In this work only the realizable $\kappa - \varepsilon$ model was used, hence only that formulation will be presented. For more details it is advised to read CD-adapco (2016) and Pope (2010).

Transport equation for κ

- Single phase flows

$$\frac{\partial}{\partial t}\rho\kappa + \nabla \cdot [\rho\kappa (\mathbf{u} - \mathbf{u}_G)] = \nabla \cdot \left[\left(\mu + \frac{\mu_t}{\sigma_\kappa} \right) \nabla\kappa \right] + f_c G_k + G_b - \rho [(\varepsilon - \varepsilon_0) + \Upsilon_M] + \xi \tag{3.31}$$

- Multiphase flows

$$\frac{\partial}{\partial t}\alpha_i\rho_i\kappa_i + \nabla \cdot [\alpha_i\rho_i\kappa_i (\mathbf{u}_i - \mathbf{u}_G)] = \nabla \cdot \left[\alpha_i \left(\mu_i + \frac{\mu_{t,i}}{\sigma_k} \right) \nabla\kappa_i \right] + \alpha_i f_{c,i} G_{k,i} + \alpha_i G_{b,i} - \alpha_i\rho_i [(\varepsilon_i - \varepsilon_0) + \Upsilon_{M,i}] + \xi_i + \sum_{j \neq i} \left(m_{ij}\kappa_j^\gamma - m_{ji}\kappa_i \right) \tag{3.32}$$

Transport equation for ε

- Single phase flows

$$\frac{\partial}{\partial t}\rho\varepsilon + \nabla\cdot\left[\rho\varepsilon\left(\mathbf{u}-\mathbf{u}_G\right)\right] = \nabla\cdot\left[\left(\mu+\frac{\mu_t}{\sigma_\varepsilon}\right)\nabla\varepsilon\right] + f_c C_{\varepsilon 1}\mathbb{S}\,\varepsilon + \frac{\varepsilon}{\kappa}C_{\varepsilon 1}C_{\varepsilon 3}G_b - \frac{\varepsilon}{\kappa+\sqrt{\nu\varepsilon}}C_{\varepsilon 2}\rho\left(\varepsilon-\varepsilon_0\right)+\xi \quad (3.33)$$

- Multiphase flows

$$\frac{\partial}{\partial t}\alpha_i\rho_i\varepsilon_i + \nabla\cdot\left[\alpha_i\rho_i\varepsilon_i\left(\mathbf{u}_i-\mathbf{u}_G\right)\right] = \nabla\cdot\left[\alpha_i\left(\mu_i+\frac{\mu_{t,i}}{\sigma_{\varepsilon,i}}\right)\nabla\varepsilon_i\right] + \alpha_i f_{c,i}C_{\varepsilon 1}\mathbb{S}\,\varepsilon + \alpha_i\frac{\varepsilon_i}{\kappa_i}C_{\varepsilon 1}C_{\varepsilon 3}G_{b,i} - \alpha_i\frac{\varepsilon_i}{\kappa_i+\sqrt{\nu_i\varepsilon_i}}C_{\varepsilon 2}\rho_i\left(\varepsilon_i-\varepsilon_0\right)+\xi+\sum_{j\neq i}\left(m_{ij}\varepsilon_j^{\gamma}-m_{ji}\varepsilon_i\right) \quad (3.34)$$

3.2.5 Pressure-velocity coupling

The SIMPLE algorithm (Patankar, 1980) was used to control the overall solution. The algorithm implemented by STAR-CCM+ can be summarised as follows (CD-adapco, 2016):

1. Set the boundary conditions
2. Compute the reconstruction gradients of velocity and pressure
3. Compute velocity and pressure gradients
4. Solve the discretised momentum equation
5. Compute the uncorrected mass fluxes
6. Solve the pressure correction equations
7. Update the pressure field under-relaxing the solution
8. Update the boundary pressure corrections

9. Correct the face mass fluxes
10. Correct the cell velocities
11. Update densities due to pressure changes
12. Free all temporary storage

3.3 Boundary conditions in coupled simulations

Differently from CFD codes RELAP5-3D uses volume and junction components also to impose boundary conditions. Usually, three components are used to model boundary conditions: TDV, TDJ and single junction.

A single junction is the most basic junction component. It is characterised only by the base quantities presented in section 3.1.3.

TDVs are used to specify either pressure outlet boundaries or fluid conditions at injection sites. When used to model a pressure outlet boundary the TDV is attached to the rest of the computational model through a single junction (SJ), usually at a location where the fluid is leaving the domain. In these cases, TDVs actively interact with the rest of the model, meaning that if the pressure of the volume adjacent to the TDV is less than the pressure the TDV specifies, there will be inflow towards the model. For example, a TDV can be used to control the pressure of a secondary circuit: in this way all the components that would need to be modelled accurately to reproduce coherent boundary conditions (pumps, valves, branches, etc.) are substituted by an equivalent boundary condition provided by the TDV. Essentially the TDV should be visualised as an infinite fluid source or sink. In fact, as explained in INL (2014c), a TDV is essentially used to model pressure outlets and mass sources or sinks across the system.

On the other hand, when modelling inlet boundaries TDVs are always connected to the rest of the model through TDJs. By doing so it is ensured that the TDV provides the proper fluid

conditions at the rate defined by the TDJ. It is important to remark that the pressure specified by the TDV in this case is ignored by the rest of the model, as the flow rate of the entering fluid is specified by the TDJ. Dually with the example presented in the previous paragraph, a TDV can be used to impose the inlet temperature of a secondary circuit.

TDJs are used to specify inflow or outflow at any location in the model. The TDJ can specify either velocity or mass flow rate. When used in combination with a TDV for an inlet boundary, the pressure inside the TDV (which is upstream of the TDJ), is not changed by the code. In fact, in TDVs used to specify inlet conditions, pressure is needed only to specify a coherent thermodynamic state for the fluid entering the system. This peculiarity of RELAP5-3D dictated the choices concerning data exchanges.

Essentially, a TDJ can be visualised as a pump. In fact, following the example presented with the TDV description, a TDJ should be placed after the inlet TDV to provide an inflow of mass with the physical specifications imposed by the TDV.

The prefix "time dependent" is inaccurate. Originally, fluid conditions could be specified only as a function of problem time, hence the name. Now however it is possible to vary the fluid conditions as a function of any problem variable the user desires.

Figure 3.4 shows how TDVs, TDJs and SJs can be visualised in use for a typical boundary condition setup. Further examples can be found in figures 4.1 and 7.2.

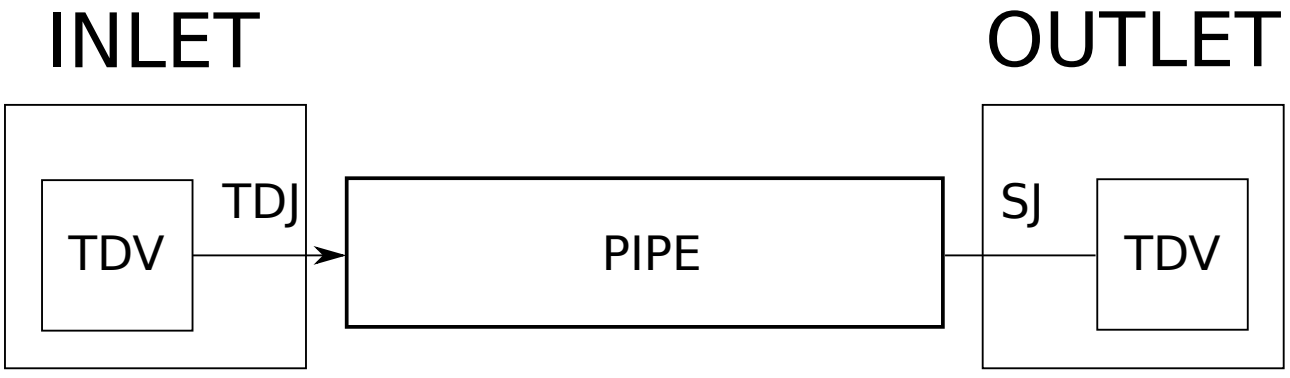


Figure 3.4: Usage of TDV, TDJ and SJ in inlet and outlet boundaries

In STAR-CCM+ boundary conditions are more straightforward. In fact, each type of boundary has its own set of variables that need to be used depending on the models chosen by the user.

In a coupled simulation awareness about the type of boundary used in each code is very important to ensure that all the quantities needed for the simulation to run successfully are exchanged between the codes.

Albeit coupling interfaces are a part of the coupled computational domain, they actually consist of two boundaries, one from RELAP5-3D and one from STAR-CCM+ that exchange specific data. This is a direct consequence of COWS relying on domain decomposition rather than domain overlapping setup to achieve the coupling between RELAP5-3D and STAR-CCM+.

From the point of view of each single code, the coupling interface is simply a standard boundary, hence the need for awareness of its type when building a coupled interface. Usually, at a coupling interface there is a code that provides an outlet boundary and the other that provides an inlet boundary.

The main assumption is that in the domain providing the outlet boundary there is outflow and in the domain providing the inlet boundary there is inflow. The data exchange between these two boundaries has to be coordinated such that the quantities are passed between the inlet and outlet boundaries to ensure that conservation laws are satisfied.

In COWS, the variables chosen to be exchanged between the two codes are mass flow rates, pressures, temperatures and void fraction. The choice of these variables comes directly from the equations implemented in the two codes. INL (2014a) explains that in nuclear reactor simulation with system codes the most important quantities to conserve are mass and energy. Momentum conservation is not considered as crucial because of the large number of momentum sources and sinks present in a nuclear reactor. This is the main reason behind the choice of exchanging mass flow rates instead of velocities. The choice for temperature instead of enthalpy or internal energy was dictated by the implementation of boundary conditions in STAR-CCM+. In fact, it is not possible to specify energies or enthalpies at inlets, only temperatures. As a consequence, additional care has been take in ensuring that the fluid properties were coherent between the two codes, such that the energy balance would not be violated. Void fraction is the only advected scalar supported at the moment in COWS and it is exchanged between the two codes only when a multiphase simulation is run. The need for pressure to be exchanged

is dictated by the use of domain decomposion as a coupling strategy. As previously explained, a coupling interface is composed by two boundaries, an outlet and an inlet, provided by the codes being coupled. At outlets for flow simulations, pressure is normally imposed as a boundary condition, hence its choice as a variable to be exchanged.

In this type of exchange usually pressure is sent from the code providing the inlet boundary whereas information about flow rates, temperatures and advected scalars are sent from the code providing the outlet boundary.

This setup, described in more detail in chapter 4, ensures a coherent exchange of variables that satisfies conservation laws.

3.4 Summary

In this chapter an overview on the codes used for the coupling, as well as the motivation for choosing such codes was given.

STAR-CCM+ is CD-adapco's flagship product and was chosen because it is becoming one of the major CFD packages used in nuclear applications. It was used to simulate flow inside a PWR vessel with full fuel assemblies details (Popov et al., 2012), which was computationally very expensive. In fact, the model was 1.035 billion cells, requiring two clusters of the NCCS to run in reasonable time.

RELAP5-3D can be considered the de-facto standard code for nuclear thermal hydraulic analyses. Coupling capabilities with STAR-CCM+ existed already, thus making the choice fall on RELAP5-3D as the system code for this work.

It was deemed necessary to provide a more detaild description of the structure of RELAP5-3D and its main components (volumes and junctions), because of its niche application field, nuclear thermal hydraulics. A description on the structure of a RELAP5-3D input deck is given in appendix C.

The equations that each code implements were presented, together with an explanation on how two standard boundaries, one per code, interact to form a coupled boundary.

Chapter 4

Coupling methodology

In chapter 3 an overview of RELAP5-3D and STAR-CCM+ was given. It was chosen to implement an explicit coupling method named COWS using STAR-CCM+'s Co-simulation API after an evaluation of the strengths and weaknesses of the various approaches presented in chapter 2. This decision was also motivated by the fact that access to either of the source codes was not granted.

In this chapter a description of STAR-CCM+'s own coupling methodology with RELAP5-3D is provided, followed by a thorough description of the structure of COWS, from both an algorithmic and a software engineering perspective.

4.1 STAR-CCM+ existing coupling tool

In STAR-CCM+, coupling to RELAP5-3D is achieved through two entities called zones and flow zones (CD-adapco, 2016). Zones are wall boundaries specified in the CFD model, needed to exchange thermal data. A flow zone is a mass flow boundary, needed to exchange flow data. An average is performed at the zones and/or flow zones in order to pass a single value from STAR-CCM+ to RELAP5-3D. Data is exchanged at each RELAP5-3D time step. STAR-CCM+ can be run at a smaller time step than RELAP5-3D. The data exchanged is the following:

- Pressure exchanged through the RELAP5-3D time dependent volume (TDV) component
- Mass flux exchanged through the RELAP5-3D junction component
- Temperature is advected through the RELAP5-3D TDV component

It is very important to remark that TDV components take active part in a RELAP5-3D simulation; however, they are used only to impose boundary conditions on the system. The latter is not allowed to change any value specified on TDV components.

TDV components are volumes where a coherent thermodynamic state must be specified. Usually this translates into the specification of pressure and temperature of the fluid. TDV components are used to specify the fluid conditions at both inlet and outlet. It is the TDJ component that specifies the mass flow rate in case of flows that are not driven by a pressure difference in the inlet and outlet boundaries.

In flows where mass flow rate is specified explicitly by a TDJ component, for the inlet boundary the TDV specifies the temperature of the fluid entering the system and for the outlet boundary it specifies the pressure of the system. For further details it is advised to read INL (2014a).

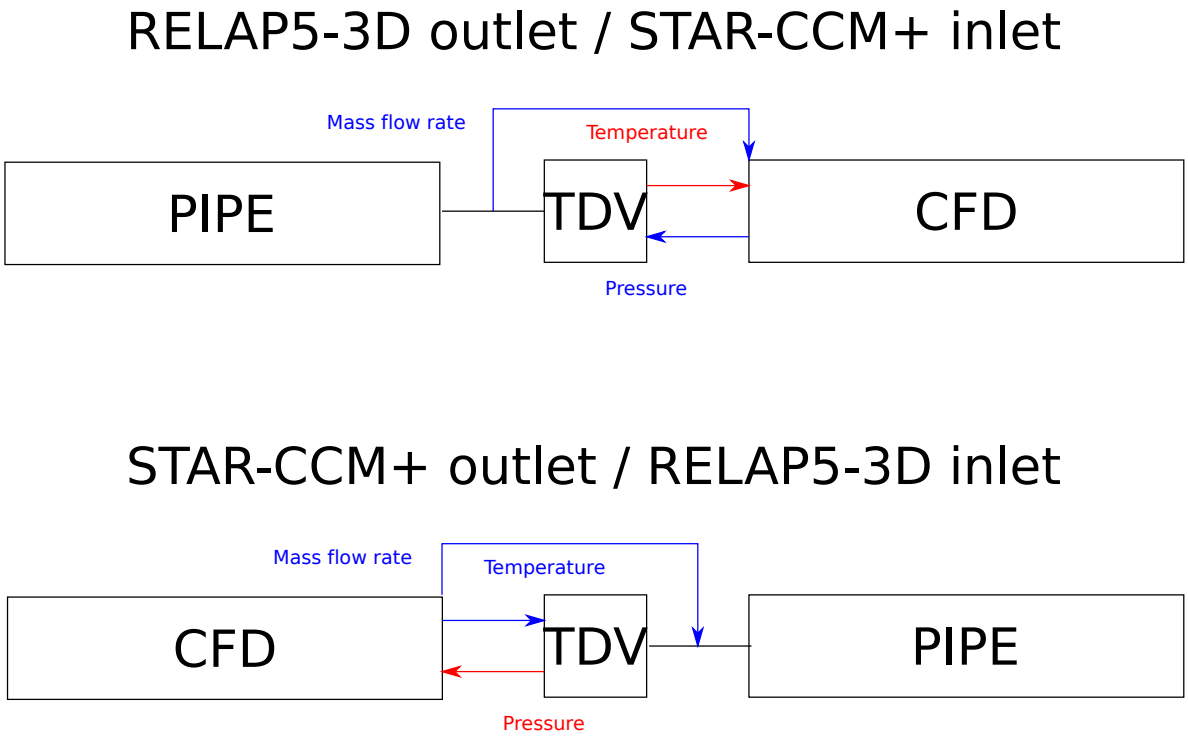


Figure 4.1: Errors in the data exchange paradigm implemented in STAR-CCM+

From this brief explanation it is evident that this implementation is not reliable because in boundaries where pressure has to be passed from RELAP5-3D to STAR-CCM+ the value being passed is always taken from the TDV component. This is not correct because the pressure should be taken from the system rather than the component that specifies the boundary condition. The same argument is valid for temperature. These cases are shown in figure , where the value that should be passed from the system but is taken from the TDV is highlighted in red. This incorrect implementation will not yield good results, as shown in chapter 5.

4.2 COWS: Co-simulation with One-dimensional Wrapper Solver

Previous work showed how the coupling tool present in STAR-CCM+ yields accurate results only in a certain configuration (Palazzi et al., 2014). At the time, access to the source code of either software was not available, hence a different strategy was adopted. STAR-CCM+'s Co-Simulation API (CD-adapco, 2015) and RELAP5-3D's "strip" option (INL, 2014a) were used to extract the data needed for a coupled simulation.

RELAP5-3D's "strip" option is an execution mode that generates a text file containing the variables specified in a specific input deck at the desired time. These variables are taken from RELAP5-3D's output file.

The Co-Simulation API was developed by CD-adapco with the purpose to offer a way of interfacing third party codes with STAR-CCM+. It consists of a library written in C++ that provides a connection to STAR-CCM+, thanks to which it is possible for data to be extracted from and sent to STAR-CCM+. The library has an interface written in C, so that the Co-Simulation API can be used with programs written in both C and FORTRAN.

The data involved in the exchage is stored in particular variables in STAR-CCM+ called field functions. In the third party code both the variables extracted from and sent to STAR-CCM+ have to be registered by the third party code. Then they have to be associated to local variables

as well so that they can be used in the third party code.

Before the Co-Simulation API it was possible to connect third party codes to STAR-CCM+ through its Java API. By doing so, the Java language could be used to write a macro to interface third party codes to STAR-CCM+. However, this would have a negative effect on runtime, slowing the performance of STAR-CCM+.

The Co-Simulation API was used to control the data transfer and the time synchronisation of the two simulations. An object oriented approach was used to develop a RELAP5-3D wrapper-solver which, together with the Co-Simulation API constitutes the COWS tool (Fig. 4.2).

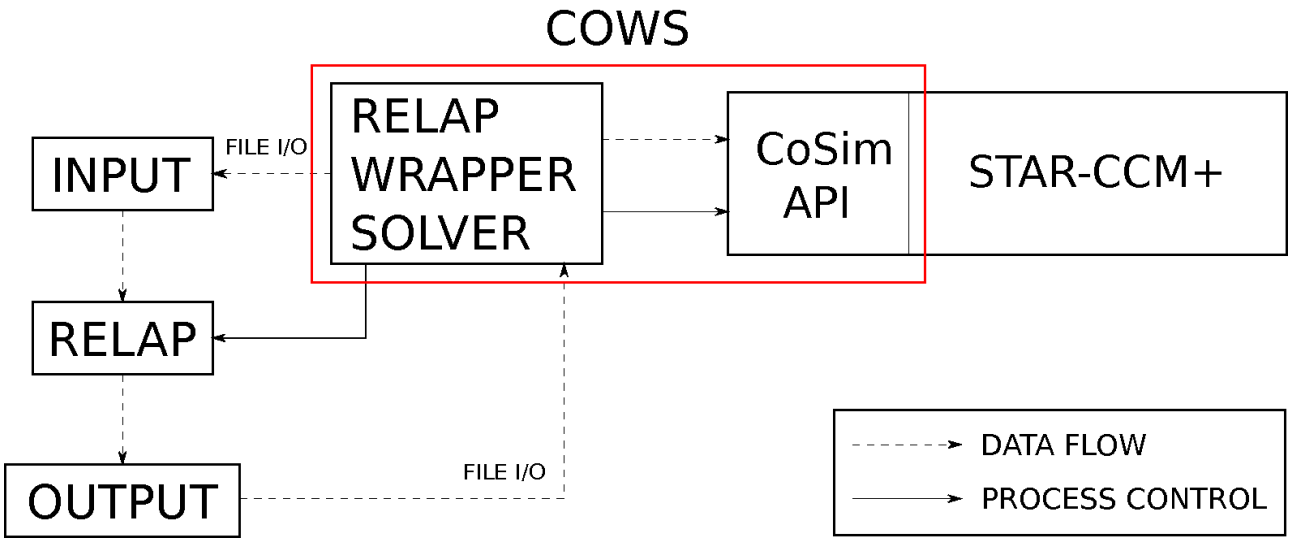


Figure 4.2: COWS tool

To run a coupled simulation, the user only needs to create the starting input deck for RELAP5-3D and the input file for COWS. When these input files are ready, COWS can be started and only then STAR-CCM+ can be connected to COWS to start the coupled simulation. When started, COWS identifies RELAP5-3D's executable path, input deck and coupled boundaries, and then instantiates RELAP5-3D for the first step. At this point, the relevant data for STAR-CCM+ is stripped from the RELAP5-3D output file and applied as boundary conditions to STAR-CCM+, which runs for one step as well. After STAR-CCM+ finishes its calculations for the current step, the relevant data for RELAP5-3D is extracted by COWS and used to generate a new input deck for RELAP5-3D. The system is thus ready to start the next step. This step advancement scheme is represented in fig. 4.3. RELAP5-3D needs to be stopped at

each iteration to extract the data to be used as boundary conditions in STAR-CCM+. This is achieved by using RELAP5-3D's restart feature.

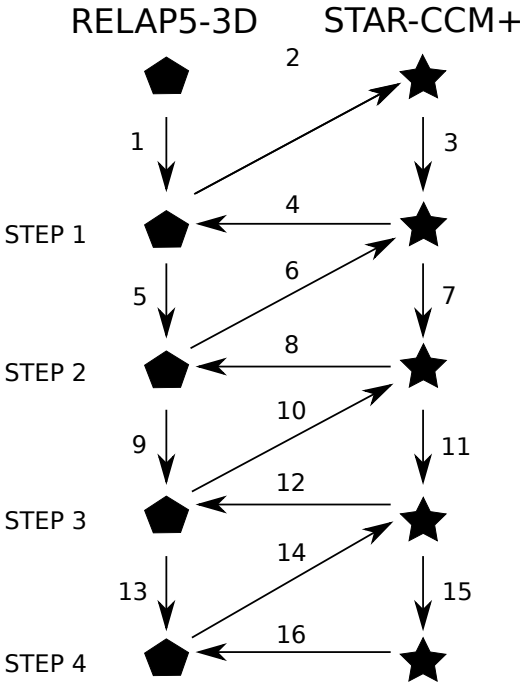


Figure 4.3: Step advancement scheme

File I/O information transfer was unavoidable because the source code of RELAP5-3D was not available. However it was minimised to avoid unnecessary lags. COWS relies on an explicit data exchange paradigm. In this context the definition of explicit data exchange paradigm can be summarised by the concept of data being exchanged only once per step. A more detailed discussion on the topic can be found in specific literature (Aumiller et al. (2001) and Weaver et al. (2002)).

It is important to note that the current implementation of COWS only supports an explicit data exchange paradigm which, in transient calculations, can lead to oscillatory behaviour and/or lags in the coupled solution (Aumiller et al. (2001) and chapter 5). Further information on coupling algorithms can be found in Aumiller et al. (2002).

Aumiller et al. (2001), Aumiller et al. (2002) and Weaver et al. (2002) show how an explicit algorithm can lead to an oscillatory behaviour during coupled simulations. Unfortunately, the Co-Simulation API only allowed for an explicit algorithm to be implemented, because of the API's embryonic development stage. In fact, the API did not allow for data to be exchanged

during the inner iterations of a STAR-CCM+ timestep, only at the beginning of each timestep. This constraint forced COWS to be implemented with an explicit coupling algorithm. For this reason, the data sets chosen from the PSBT benchmark to validate COWS for multiphase flows were taken from the steady state group instead of the transient group.

4.3 Data exchanged

The variables passed between the two codes change in relation to which boundaries are coupled. Pressure is always passed from the inlet of a downstream section of the domain to the outlet of an upstream section, independently of which code is downstream. Mass flow rate is always passed from the outlet of an upstream section of the domain to the inlet of a downstream section, independently from which code is upstream. The temperature of the fluid is convected upwind, meaning that the upstream code passes the temperature downstream. If STAR-CCM+ is modelling the upstream part of the domain, the surface average of the temperature is passed to RELAP5-3D. The reason for this can be found in the formulation of RELAP5-3D, where it is assumed that the bulk fluid properties are uniform over the cross section of the pipe (INL, 2014a). Table 4.1 summarises the variable exchanged at different interfaces for single phase flows.

	STAR-CCM+ inlet	**STAR-CCM+ outlet**	**RELAP5-3D inlet**	**RELAP5-3D outlet**
p	Passed	Received	Passed	Received
$\dot{m}$	Received	Passed	Received	Passed
T	Advected			

Table 4.1: Variables exchanged at interfaces - Single phase flows

For multiphase flows the set of variables exchanged is the same as the single phase variables. Mass flow rate and temperature of each phase are exchanged, together with the volume fraction of each phase. Table 4.2 summarises the variables exchanged at different interfaces for multiphase flows.

This choice of exchange variables ensures mass and energy conservation, provided that the fluid

	STAR-CCM+ inlet	**STAR-CCM+ outlet**	**RELAP5-3D inlet**	**RELAP5-3D outlet**
$\mathbf{p}$	Passed	Received	Passed	Received
$\dot{\mathbf{m}}_\mathbf{i}$	Received	Passed	Received	Passed
$\mathbf{T}_\mathbf{i}$	Advected			
$\alpha_\mathbf{i}$	Advected			

Table 4.2: Variables exchanged at interfaces - Multiphase flows

properties of the two codes are set to the same values. Momentum conservation is a well-known issue (Aumiller et al. (2001), Aumiller et al. (2002) and Weaver et al. (2002)) but is considered to be of secondary importance compared to conservation of mass and energy. The reason for this is the presence of a large number of momentum sources and sinks in systems like a nuclear reactor (INL, 2014a).

4.3.1 Periodic boundary conditions

An important remark concerning the data exchange between the two codes is needed: The one-dimensional nature of RELAP5-3D requires the data coming from STAR-CCM+ to be averaged before use by RELAP5-3D. The averaging operation can be performed easily.

However, when data is coming from RELAP5-3D to STAR-CCM+ some issues arise, due to the absence of direct information about transversal gradients and turbulence in RELAP5-3D. To overcome these limitations, which are intrinsic characteristics of every system code, assumptions on the transversal profiles of the inlet variables of the STAR-CCM+ model are required.

A simple solution consists in adding an inlet region of a length that allows the flow to develop fully before entering the domain of interest to the calculations. Alternatively, surface maps of the desired variables can be used as multipliers for the single values extracted from the system code. However, the respective drawbacks to these two alternatives are an increase in computational costs and the need of a database of the desired inlet quantities at different flow regimes respectively.

Two of the most significant limitations of system codes are the inability to represent turbulence explicitly and the lack of information about non-axial profiles. To overcome these issues it is

possible to add a further piece of domain at the inlet of the CFD model to allow the flow to develop fully. Alternatively, normalised desired profiles can be used as multipliers for the single values taken from the system code.

The main drawback of the first method is that adding a portion of domain might increase the computational costs. The second method allows for the reconstruction of the profile of the desired quantity (e.g. velocity, temperature, turbulent kinetic energy etc.) without the need of additions to the existing computational domain. However, a database of all the quantities needed at different flow regimes is necessary.

Using STAR-CCM+'s periodic boundary interfaces it is possible to capture the benefits of both methods while minimising the drawbacks. To do this, a small domain, henceforth referred to as periodic domain, is added to the existing simulation (fig. 4.4). The inlet and outlet of the periodic domain are then connected with each other through a periodic boundary interface. This type of interface represents a cyclic repeat of information across the boundaries, thus effectively approximating a repeating geometry (CD-adapco, 2015).

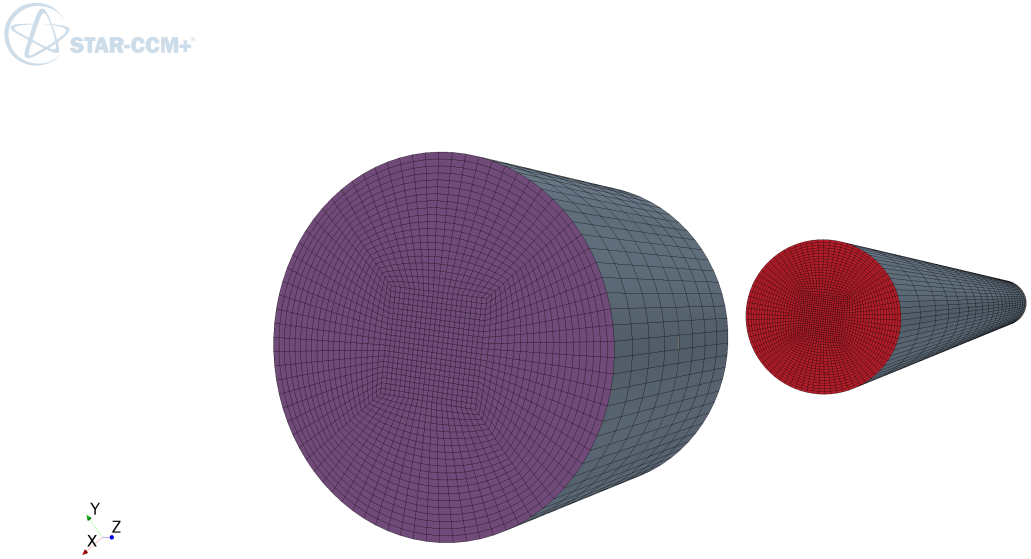


Figure 4.4: Periodic domain (front) and computational domain (back)

Mass flow rate or cyclic pressure drop can be used as the parameter needed from the periodic boundary interface to perform calculations. The physical parameters needed for the actual simulation can be mapped in any cross section of the periodic domain before being passed to

the inlet of the main computational domain.

The extension of this technique for the treatment of multiphase flows is, in principle, simple. In fact, it would be necessary to also reconstruct the maps of the variables associated with the additional phase. This technique could not be used for multiphase flows however, because STAR-CCM+ currently does not allow the use of periodic boundary interfaces with a specified mass flow rate or pressure drop of the flow. As a consequence, uniform inlet profiles for temperature, void fraction and mass flow rate were used for STAR-CCM+ in the coupled simulations of multiphase flows.

4.4 Software engineering structure details

COWS was developed in C++ following an OO paradigm because the code resulting from this approach is easier to maintain, extend and modify. Some software design patterns, namely singleton, factory method and state, were used. A brief description of the main characteristic of the OO methodology and of the design patterns used can be found in appendices A and B respectively.

COWS is hosted in CD-adapco's repositories and can be accessed by contacting the author or through a direct request to CD-adapco.

4.4.1 Main components

`Options` class

The `Options` class parses the options from the commandline parameters and stores them. It is a singleton. It has a `Creator` class that creates the correct `Wrapper` class by delegating the task to a `Creator` subclass implemented inside each subclass of `Wrapper`. The implementation with the `Creator` class can be considered as a variant of the factory method pattern. It holds a static list of `Creator` objects.

`Wrapper` class

The `Wrapper` class is the base class for the one-dimensional code wrappers. It is an abstract class and provides an interface for all the wrappers. Most of the functionalities present in the `Wrapper` interface are implemented in its subclasses, which are singletons. Every subclass has a static initialisation method that register itself inside a list object inside the `Options` class. Every subclass of `Wrapper` also implements a subclass of the `Creator` class inside the `Options` class for the creation of the correct wrapper. The interface of the `Wrapper` class implements the following methods:

- `appendNewData`

This function appends the data of the latest step to the global output file.

- `clean`

This function cleans the local directory from temporary files generated by the RELAP5-3D run.

- `generateRestartDeck`

This function creates a RELAP5-3D input deck suited to be used in restart mode. The deck is built with the values passed by STAR-CCM+ as boundary conditions.

- `generateStripDeck`

This function creates a RELAP5-3D input deck suited to be used in strip mode. This input deck will be referred as the strip deck. The strip deck contains the names of the variables that will be extracted from RELAP5-3D and passed to STAR-CCM+.

- `instantiate`

This function runs RELAP5-3D for one step.

- `parseStripFile`

When RELAP5-3D is run in strip mode it generates a text file containing the variables indicated in the strip deck in a particular format. The text file containing the RELAP5-3D coupling variables will be referred as the strip file. This function parses the strip file and extracts the variables.

- `populateIncomingFields`

This function populates the fields that will be sent to STAR-CCM+ with the values extracted from RELAP5-3D.

- `registerCoSimulationFields`

This function binds the fields inside STAR-CCM+ to the ones used in COWS. It needs to be called only once at the beginning of the coupled simulation.

`OneDCodeState` class

The `OneDCodeState` class is the base class for the state pattern that identifies the state of the system code. Different options are used to run the system code depending on its state. Three different states are taken into account and they are represented by the following subclasses:

- `OneDCodeInitialState`
- `OneDCodeRestartState`
- `OneDCodeStripState`

The `OneDCodeState` is held by the `OneDCodeCommand` class.

`CoupledBoundary` class

The `CoupledBoundary` class is the base class for the different type of boundaries that the user wants to couple. Its derived classes are the following:

- `CcmInlet` for coupled boundaries associated with an inlet boundary in STAR-CCM+
- `CcmOutlet` for coupled boundaries associated with an outlet boundary in STAR-CCM+
- `CcmWall` for coupled boundaries associated with a wall boundary in STAR-CCM+

It is an abstract class and provides an interface for all the coupled boundaries.

`OneDComponent` class

The `OneDComponent` class is the base class for the different types of components that are part of a coupled boundary. Its derived classes are the following:

- `HydraulicComponent` for system code components where coupled scalar variables have to be extracted from
- `ScalarComponent` for system code components dedicated to provide pressure and/or temperature boundary conditions
- `VectorComponent` for system code components dedicated to provide or extract mass flow boundary conditions
- `ThermalComponent` for system code components dedicated to heat sources or sinks

For every code supported by COWS every class derived from `OneDComponent` has at least one subclass that represent a component modelled in the system code. For RELAP5-3D the subclasses are as follows:

- `Pipe` inherits from `HydraulicComponent`
- `TimeDependentVolume` inherits from `ScalarComponent`
- `Junction` inherits from `VectorComponent`
- `HeatStructure` inherits from `ThermalComponent`

`LibraryLoader` class

The `LibraryLoader` class is provided with the tutorials of STAR-CCM+ and provides the function callbacks to the Co-Simulation API. It is a singleton.

`CouplingAlgorithm` class

The `CouplingAlgorithm` class coordinates the advancement of the coupled simulation. When it is first instantiated it creates the correct `Wrapper` subclass and uses it throughout the simulation.

4.4.2 Interaction of objects during program execution

When COWS is launched, every subclass of `Wrapper` register itself inside the `Options` class' static list of `Creator` objects.

The `Options` class is the first one to be instantiated by COWS. After the command line parameters have been parsed and stored inside the `Options` class, the `CouplingAlgorithm` class is instantiated, where the correct subclass of `Wrapper` is created.

The creation process of the correct `Wrapper` object consists of the creation of all the `OneDComponent` objects and all the `CoupledBoundary` objects involved in the coupled simulation; using the `LibraryLoader` object to load the CoSimulation API library and register the variables to be exchanged by the two codes on the server started by the CoSimulation API.

At this point the initialisation process is over and the coupled simulation can start. All the operations described hereafter are executed by the correct subclass of `Wrapper`, thus every method mentioned in the following paragraphs is implemented by every subclass of `Wrapper`.

The system code is instantiated with the `instantiate` method, which puts the `OneDCodeCommand` class in the state `OneDCodeInitialState`. After this instantiation the state of `OneDCodeCommand` is changed to `OneDCodeStripState`. At each iteration, the text files generated by the system code are appended to a global file with the method `appendNewData`. COWS creates a strip file in a format suitable for the specific system code based on the variables that need to be exchanged thanks to the method `generateStripDeck`. The system code is instantiated with the `instantiate` method once more to generate a text file with the variables needed for the exchange. The state of `OneDCodeCommand` is changed to `OneDCodeRestartState` after this instantiation. The text files that are no longer needed are deleted with the `clean` method. The variables to be passed to STAR-CCM+ are parsed with the method `parseStripFile` from the text file generated by the most recent instantiation of the system code and sent to the awaiting STAR-CCM+ simulation. At this stage, COWS is idle, waiting for STAR-CCM+ to send its data.

The data received from STAR-CCM+ is used as boundary conditions for the simulation, which is run for a number of iterations. After this, the data for the exchange is sent from STAR-CCM+ to the awaiting COWS process. When COWS receives the data it updates the existing `OneDComponent` objects with the method `populateIncomingFields` and then generates a restart deck for the system code with the method `generateStripDeck`.

Finally, the simulation time is advanced and the cycle restarts.

4.5 Computational resources

All the simulations presented in the following chapters were run on this machine:

Item	Quantity
Manufacturer	Dell Inc.
Model	Precision Workstation T5500
OS	CentOS Linux 7.2
RAM	48 GB
CPU count	2
CPU model	Intel Xeon X5660
CPU clock	2.80 GHz
Core count	12 (6 per CPU)
Architecture	x64
Video controller	NVIDIA GF100GL [Quadro 4000], 64 MB

Table 4.3: Computational resources

4.6 Summary

COWS was described in this chapter. COWS was developed in C++ following an OO paradigm because the code resulting from this approach is easier to maintain, extend and modify. Some software design patterns, namely singleton, factory method and state, were used. COWS relies on STAR-CCM+'s Co-Simulation API and RELAP5-3D's strip options.

The Co-Simulation API was developed by CD-adapco with the purpose to offer a way of interfacing third party codes with STAR-CCM+. It consists of a library written in C++ that provides a connection to STAR-CCM+, thanks to which it is possible for data to be extracted from and sent to STAR-CCM+. The library has an interface written in C, so that the Co-Simulation API can be used with programs written in both C and FORTRAN.

It is important to note that at the moment COWS only supports an explicit data exchange paradigm. The choice of an explicit coupling algorithm was dictated by the embryonic development stage of the STAR-CCM+ Co-Simulation API.

Two sets of variables, one for single-phase and one for multiphase flows, were chosen to satisfy mass and energy, provided that the fluid properties of the two codes are set to the same values.

Momentum conservation is a well-known issue but is considered to be of secondary importance compared to conservation of mass and energy. The reason for this is the presence of a large number of momentum sources and sinks in systems like a nuclear reactor.

The one-dimensional nature of RELAP5-3D requires the data coming from STAR-CCM+ to be averaged before being used by RELAP5-3D. However, when data is coming from RELAP5-3D to STAR-CCM+ some issues arise, due to the absence of direct information about transversal gradients and turbulence in RELAP5-3D. To overcome these limitations, which are intrinsic characteristics of every system code, assumptions on the transversal profiles of the inlet variables of the STAR-CCM+ model are required.

Using periodic boundary conditions in STAR-CCM+ allows to overcome the problem of mapping information coming from a one-dimensional domain onto a three dimensional domain. Periodic boundary interfaces represents a cyclic repeat of information across the boundaries, thus effectively approximating a repeating geometry.

Chapter 5

Single phase coupling

COWS, the tool developed to couple STAR-CCM+ and RELAP5-3D using STAR-CCM+'s Co-simulation API, was validated with different scenarios and its performance was compared to STAR-CCM+'s own coupling tool for RELAP5-3D.

Two main cases were simulated: a steady state scenario where Moody's friction factor was used as a performance benchmark and a transient scenario where COWS was tested against RELAP5-3D standalone simulations.

This chapter shows how STAR-CCM+'s own coupling tool for RELAP5-3D is able to perform only a limited set of simulations, more specifically those cases where only the inlet of the CFD model is coupled to RELAP5-3D. In fact, in cases where the outlet of the CFD model is coupled to RELAP5-3D the results do not match, even for the steady state case. For this reason the model was not tested against the more complicated scenarios.

The initial development of COWS focused on creating a program that did not have such shortcomings, thus allowing coupled simulations of pipe sequences or loops for single phase flows. This chapter shows how the implementation described in chapter 4 achieves this objective for both steady and transient flows.

In addition, the use of periodic boundary conditions, a feature present in STAR-CCM+, allows the reconstruction of two-dimensional fully developed profiles from quantities sent from

RELAP5-3D, which are one-dimensional. These two-dimensional profiles are then used as boundary conditions for the CFD model, as shown at the end of this chapter.

5.1 Model description

Hydraulic analyses in both steady state and transient scenarios were performed to evaluate the capabilities of COWS.

For the steady state cases Moody's friction factor was evaluated at $Re = 10^4$, $Re = 10^5$ and $Re = 10^6$ for flows in circular pipes at a temperature of 350 K and an outlet pressure of 101325 Pa. No heat source is present.

When the location for the friction factor evaluation was inside the domain modelled with STAR-CCM+, the friction factor was calculated in two different ways, using the pressure gradient (Eq. 5.1) and the wall shear stress (Eq. 5.2) (Palazzi et al., 2014):

$$f_{PG} = -\frac{D(\partial p/\partial z)}{0.5\rho u^2} \tag{5.1}$$

$$f_{WS} = \frac{4\tau_w}{0.5\rho u^2} \tag{5.2}$$

Equations (5.1) and (5.2) are referred as PG and WS evaluation, respectively. When the location for the friction factor evaluation was inside the domain modelled with RELAP5-3D, its value was taken directly from the output file. The values from RELAP5-3D, WS and PG evaluations were compared with Haaland's equation for the friction factor (Genic et al., 2011):

$$\frac{1}{\sqrt{f}} = -1.8 \log_{10}\left(\frac{6.9}{Re}\right) \tag{5.3}$$

For transient calculations flow in smooth pipes at a temperature of 350 K, a pressure of 70 bar and a hydraulic regime of $Re = 10^5$ was chosen as a validation benchmark. No heat source

is present. A time dependent mass flow rate was imposed at the inlet of the full domain and pressure trends were evaluated at three different timesteps: 0.1 s, 0.01 s and 0.001 s.

RELAP5-3D's capabilities of handling turbulence are intrinsic, thanks to the many experimental correlations implemented in the code. On the other hand STAR-CCM+ requires specification of turbulence related parameters. The Realizable $k-\varepsilon$ turbulence model with the two-layer wall treatment was used. The turbulence model parameters were set to the default values given by STAR-CCM+.

5.2 STAR-CCM+'s own coupling tool results

It was deemed necessary to assess the coupling capabilities of the existing tool implemented in STAR-CCM+ before attempting to develop an in-house coupling interface.

5.2.1 Additional features of the model

A coupled analysis of single phase flow in a circular pipe was performed. The pipe was 1.5 m long and its diameter was 10 mm. Moody's friction factor was used to evaluate the performance of the coupled tool. The evaluation point for the friction factor was set at 1.4 m from the inlet boundary, far enough from both inlet and outlet to avoid boundary effects.

The pipe was divided into two parts: the upstream part, which was 1.005 m long and the downstream part which was 0.495 m long. This layout ensured that the point at which the results were evaluated was always inside the downstream domain (fig. 5.1).

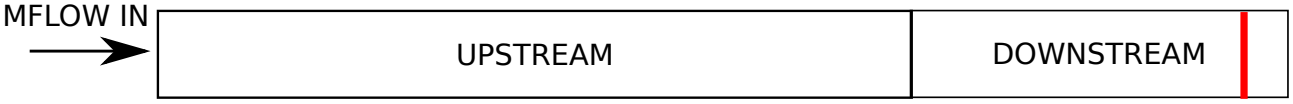


Figure 5.1: Computational domain

Rodriguez (2012) states that STAR-CCM+'s own coupling tool is not able to perform closed loop simulations, but information on where the code fails to perform are not given. For this

reason, before attempting a closed loop or a multiple system code/CFD domain sequence simulation, STAR-CCM+'s tool is tested on a simple upstream/downstream domain decomposition.

Three cases were studied: The first case consisted of modelling the pipe walls as smooth, whereas in the second and third case the roughness of the walls was set to $20\,\mu m$ and $50\,\mu m$ respectively, hence the need for correlations that take into account wall roughness.

In addition to the evaluation methods presented in section 5.1, three more correlations will be used for the analysis of the results:

- Zigrang-Sylvester (INL, 2014a)

$$\frac{1}{\sqrt{f}} = -2\log_{10}\left\{\frac{\epsilon}{3.7D} + \frac{2.51}{Re}\left[1.14 - 2\log_{10}\left(\frac{\epsilon}{D} + \frac{21.25}{Re^{0.9}}\right)\right]\right\} \tag{5.4}$$

- Nikuradse (Cebeci and Bradshaw, 1977)

$$\frac{1}{\sqrt{f}} = 2\log_{10}\left(\frac{D}{2\epsilon}\right) + 1.74 \tag{5.5}$$

- Haaland correlation modified to allow the treatment of rough pipes (Genic et al., 2011)

$$\frac{1}{\sqrt{f}} = -1.8\log_{10}\left[\left(\frac{\epsilon}{3.7D}\right)^{1.11} + \frac{6.9}{Re}\right] \tag{5.6}$$

The Zigrang-Sylvester and Haaland correlations are explicit approximations of the Colebrook-White formula (Genic et al. (2011) and INL (2014a)). The Zigrang-Sylvester correlation is the one implemented by RELAP5-3D and the Haaland correlation is used as a further comparison. The Colebrook-White formula originates from experiment on standard commercial pipes. Nikuradse's formula originates from experiments on non-commercial pipes with uniform wall roughness (Idel'chik, 1966). Furthermore, Nikuradse's formula is valid only in the "fully turbulent region" of Moody's friction factor chart (Cebeci and Bradshaw, 1977).

It follows that Nikuradse's formula and the family of formulae deriving from Colebrook-White

are essentially covering different behaviours, which cannot be distinguished with just one roughness parameter.

The different behaviour is evident when comparing Moody's chart (fig. 5.3) with the equivalent diagram for uniform sand-grain roughness (fig. 5.2). Between the curve for smooth pipes and the fully turbulent region the friction factor is a function of the relative roughness only. Compared with non-uniform roughness, the friction factor for uniform sand-grain roughness has a steeper decrease at the beginning and then increases to approximately the same value of the one in Moody's chart. The difference in behaviour at intermediate Reynolds numbers is due to whether the roughness is uniform across the pipe walls or not.

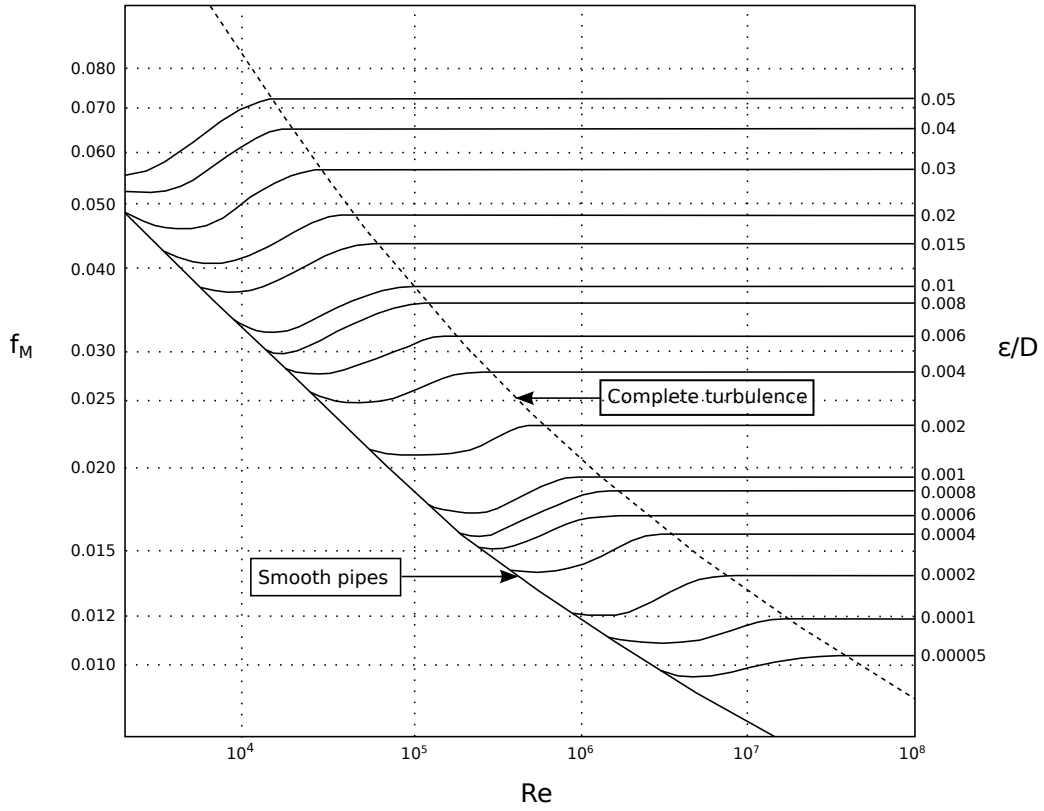


Figure 5.2: Uniform sand-grain roughness friction factor diagram - Adapted from Idel'chik (1966)

STAR-CCM+ models roughness using an equivalent sand-grain roughness parameter for its wall treatment models (CD-adapco, 2016). Using an equivalent sand-grain roughness implies that the wall roughness is uniform throughout the computational domain, as explained when Nikuradse's formula was introduced. This configuration is typical of non-commercial pipes with uniform wall roughness (Idel'chik, 1966), hence will have a behaviour similar to the one in fig. 5.2.

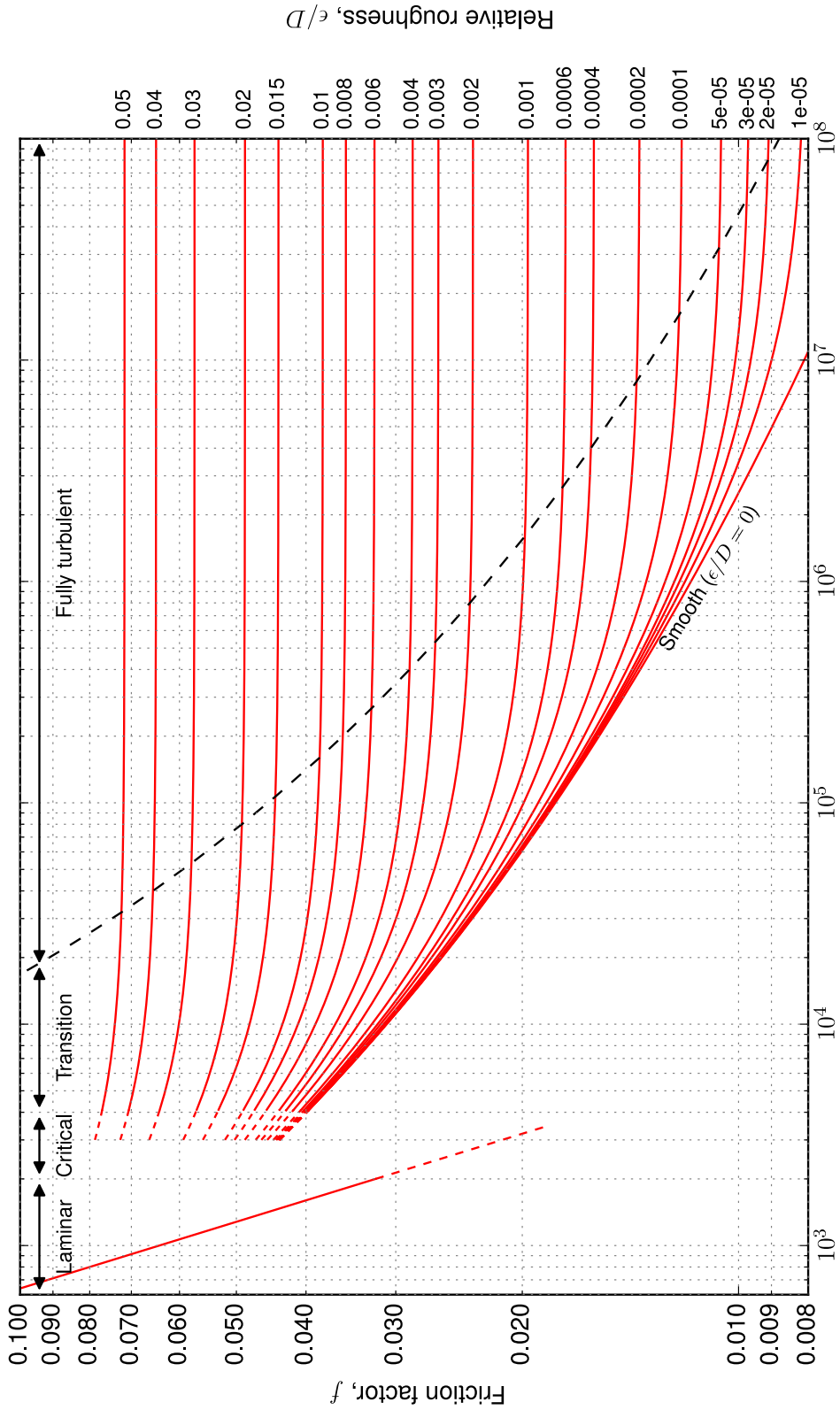


Figure 5.3: Moody chart - Adapted from Hyams (2012)

In STAR-CCM+, the equivalent sand-grain roughness parameter is used when modelling the velocity distribution in turbulent boundary layers. The inner region of the boundary layer can be split up into three sublayers.

- Viscous sublayer
- Logarithmic layer
- Buffer layer

In general, STAR-CCM+ models the effect of the wall roughness by moving the logarithmic region of the boundary layer closer to the wall. For a more detailed description it is advised to read CD-adapco (2016). The dimensionless velocity distribution in the logarithmic layer is modelled as:

$$u^+ = \frac{1}{\kappa} \ln \left(\frac{E}{f} y^+ \right) \tag{5.7}$$

Where:

$$\begin{aligned} y^+ &= \frac{y u_\tau}{\nu} \\ u_\tau &= \sqrt{\tau_w / \rho} \\ u^+ &= \frac{u}{u_\tau} \end{aligned} \tag{5.8}$$

And the default values of the other coefficients are:

- $\kappa = 0.42$
- $E = 9.0$

The roughness function f is unity for smooth walls and for rough walls is computed as:

$$f = \begin{cases} 1 & R^+ \leq R^+_{smooth} \\ \left[B \left(\dfrac{R^+ - R^+_{smooth}}{R^+_{rough} - R^+_{smooth}} \right) + CR^+ \right]^a & R^+_{smooth} < R^+ < R^+_{rough} \\ B + CR^+ & R^+ \geq R^+_{rough} \end{cases} \tag{5.9}$$

Where:

$$\begin{aligned} R^+ &= \frac{\epsilon u_\tau}{\nu} \\ a &= \sin \left[\frac{\pi}{2} \frac{\log(R^+/R^+_{smooth})}{\log(R^+_{rough}/R^+_{smooth})} \right] \end{aligned} \tag{5.10}$$

And the default values of the other coefficients are:

- $B = 0$
- $C = 0.253$
- $R^+_{smooth} = 2.25$
- $R^+_{rough} = 90$

The coefficient B was included for completeness as the original formulation includes it.

These considerations, as well as the correlations will be used in the following sections to analyse the results.

When the wall is rough, the values from the simulations and from the correlations differ. This is due to the wall treatment used used by STAR-CCM+, which relies on a different method to incorporate roughness in the calculations.

Correlations are explicit approximations of the Colebrook-White formula, which gives good results for non-uniform roughness, because it is derived from experiments on commercial pipes (Idel'chik, 1966).

Nikuradse's formula was needed because STAR-CCM+ uses equivalent sand-grain roughness height.

As already mentioned, Nikuradse's formula is valid only in the "fully turbulent region" of Moody's chart. It follows that Colebrook-White's and Nikuradse's correlations are actually covering different behaviours, which are indistinguishable if using a single roughness parameter.

5.2.2 STAR-CCM+ downstream

The first set of results to be analysed are with STAR-CCM+ downstream. Under the assumption of fully developed flow, PG and WS evaluations should yield the same value. In fact, from a balance of forces on a pipe of infinitesimal length and the previous assumption of fully developed flow it follows that:

$$-\frac{\partial p}{\partial z} = 4\frac{\tau_w}{D} \tag{5.11}$$

Multiplying everything by the pipe diameter D and dividing everything by the fluid's kinetic energy $0.5\rho u^2$, equation 5.11 becomes:

$$-\frac{D(\partial p/\partial z)}{0.5\rho u^2} = 4\frac{\tau_w}{0.5\rho u^2} \tag{5.12}$$

Which becomes, recalling equations (5.1) and (5.2):

$$f_{PG} = f_{WS} \tag{5.13}$$

This can also be seen comparing the results of the PG and WS evaluations of the friction factor (fig. 5.4).

Since there is no difference between PG and WS evaluations, the results presented hereafter will make use of the PG evaluation, when calculations of the friction factor were performed

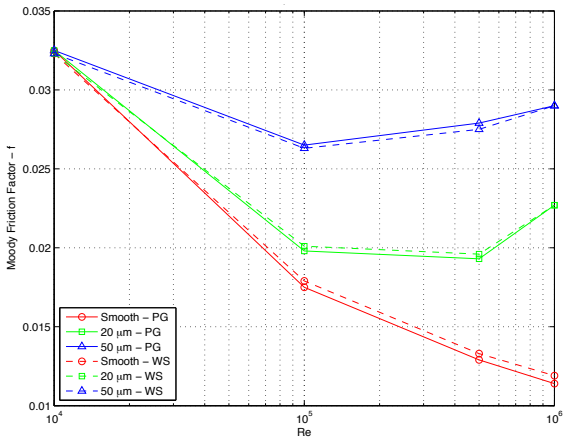


Figure 5.4: Friction factor comparison - PG vs WS method

with STAR-CCM+.

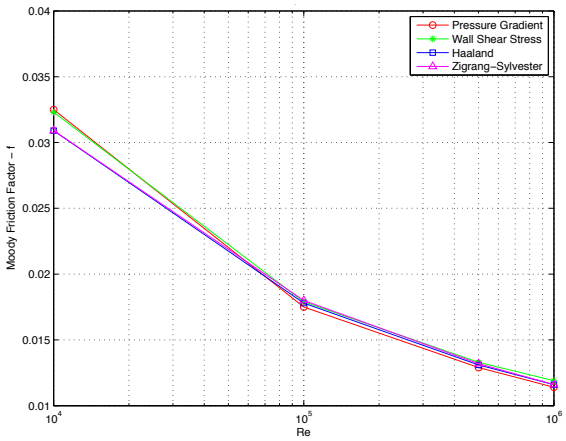
At low Reynolds number the results collapse to the value of the friction factor for smooth pipes. Increasing the Reynolds number causes the values of the friction factor to decrease with different slopes and, in the case of rough pipes, they increase again, at almost the same value given by the correlations (fig. 5.5). The results from the coupled simulations agree with Nikuradse's correlation when approaching the fully turbulent region.

Fig. 5.6 shows the comparison between the coupled simulations and the standalone RELAP5-3D and STAR-CCM+ simulations. The results of the coupled simulations match almost perfectly the standalone STAR-CCM+ simulations (fig. 5.6a). However, there is a discrepancy with the standalone RELAP5-3D results (fig. 5.6b) due to the different way the two codes model wall roughness. As expected, this discrepancy is present only in the transition region.

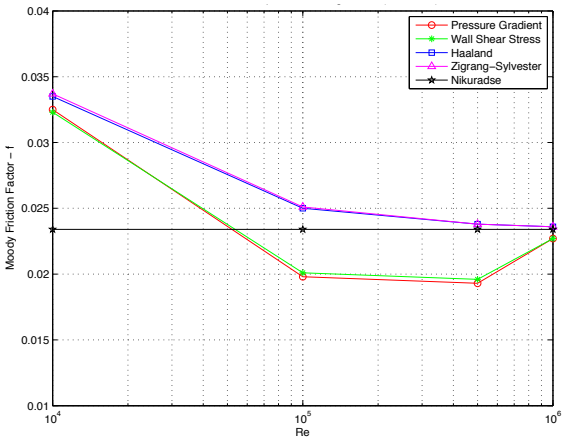
5.2.3 STAR-CCM+ upstream

When STAR-CCM+ was used to model the upstream part of the pipe the results of the coupled simulations do not match with the results of the standalone simulations (fig. 5.7). However, the simulations yield similar results for high Reynolds numbers.

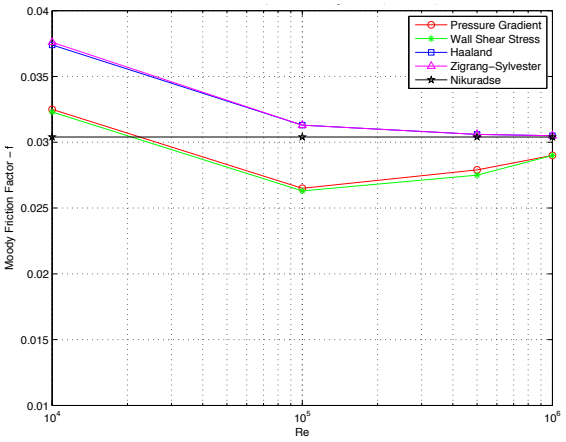
The same behaviour can be seen when comparing the results of the coupled simulations with the various correlations 5.8.



(a) Smooth walls

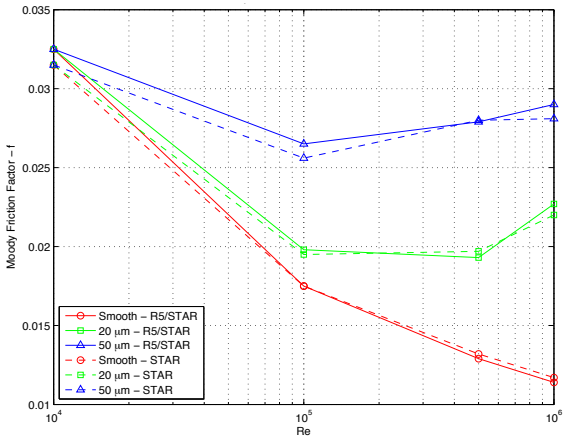


(b) Rough walls, 20 μm

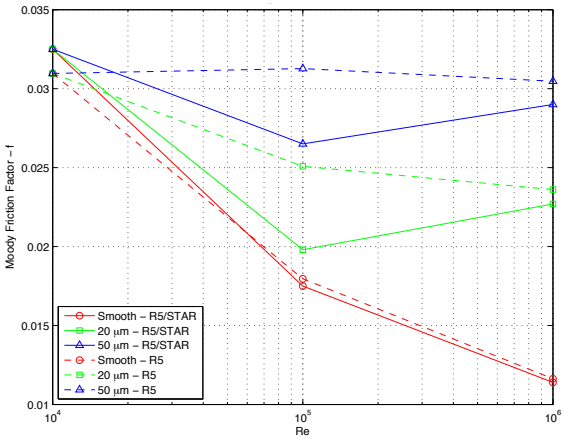


(c) Rough walls, 50 μm

Figure 5.5: Coupled simulations vs correlations



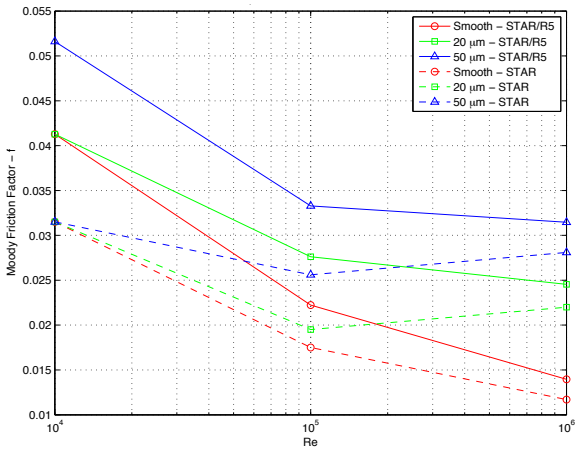
(a) Coupled vs. STAR-CCM+



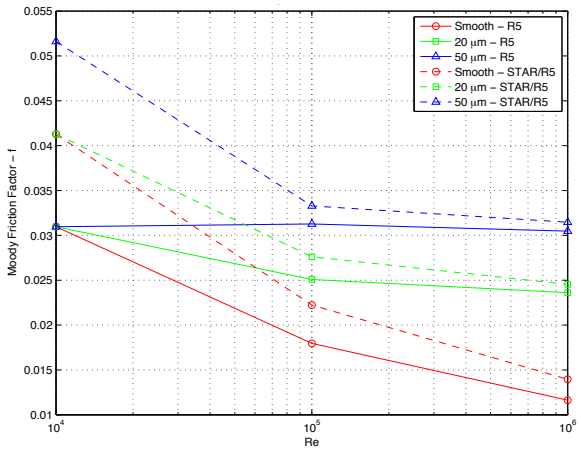
(b) Coupled vs. RELAP5-3D

Figure 5.6: Coupled vs standalone simulations

Fig. 5.9 shows the comparison between the coupled simulations with STAR-CCM+ both upstream and downstream.

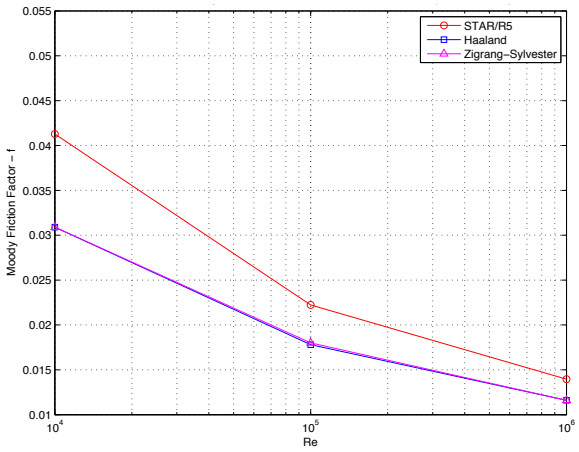


(a) STAR-CCM+

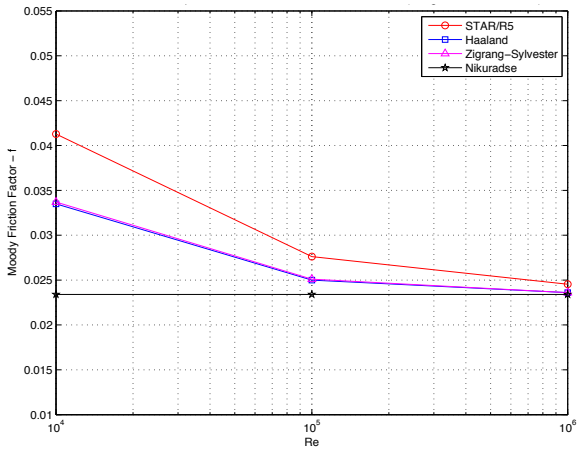


(b) RELAP5-3D

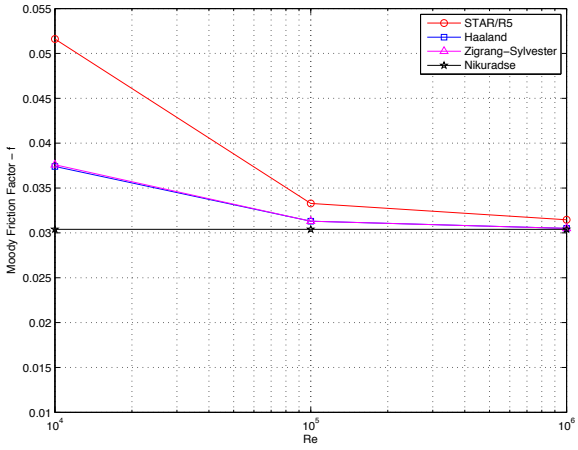
Figure 5.7: Coupled vs standalone simulations



(a) Smooth walls



(b) Rough walls, $20\,\mu m$



(c) Rough walls, $50\,\mu m$

Figure 5.8: Coupled simulations vs correlations

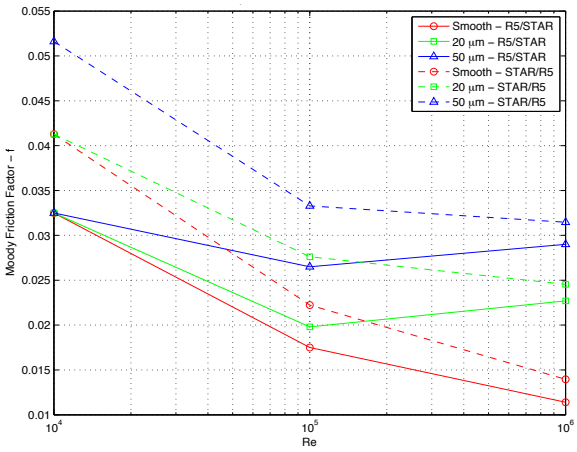


Figure 5.9: Coupled simulations comparison

5.2.4 Considerations

The results presented in the previous sections show that the coupling tool implemented in STAR-CCM+ is not suited to perform simulations of closed loop/pipe sequence scenarios. In fact, when STAR-CCM+ precedes RELAP5-3D in the domain sequence, the results are not reliable. Access to the source code of either program was not granted however, as explained in section 4.1, the data exchange paradigm implemented by the STAR-CCM+ tool is not correct. This incorrect data exchange paradigm is deemed to be the cause of the discrepancies observed.

On the other hand, when STAR-CCM+ follows RELAP5-3D in the domain sequence, the results differ in the region of $10^4 < Re < 10^6$ because of the different way of modelling roughness. RELAP5-3D uses the Zigrang-Sylvester formula, whereas STAR-CCM+ uses a roughness function to modify the logarithmic region of the turbulent boundary layer.

For a 3-loop, 900 MWe PWR, Anglart (2011) gives the following operating parameters for the primary circuit:

- Pressure: 155 bar
- Mass flow rate: 13245 kg/s
- RPV inlet temperature: 286 $^{\circ}C$
- RPV outlet temperature: 323.2 $^{\circ}C$

- RPV inlet density: 753.61 kg/m^3
- RPV outlet density: 671.56 kg/m^3
- RPV inlet dynamic viscosity: $9.4 \cdot 10^{-5}$ $Pa \cdot s$
- RPV outlet dynamic viscosity: $7.92 \cdot 10^{-5}$ $Pa \cdot s$
- Hot leg inner diameter: 0.736 m
- Cold leg inner diameter: 0.698 m

Performing an arithmetic averaging of inner diameters, densities and dynamic viscosities it is possible to estimate the Reynolds number for the primary circuit of the reactor, which is approximately $90 \cdot 10^6$. This value is almost two orders of magnitude higher than the largest case considered in the steady state analyses. As explained in section 5.2.1, in the "fully turbulent" region the friction factor is a function of the wall roughness only. For a Reynolds number relevant for PWRs, like the one calculated before it is possible to see from figures 5.3 and 5.2 that most curves fall into the "fully turbulent" region. Consequently, the differences in how the roughness is modelled are irrelevant for real reactor applications.

5.3 COWS results

The results presented hereafter show how the implementation of COWS has overcome the issues encountered with the native coupling tool implemented in STAR-CCM+.

5.3.1 Steady state analyses

The first set of results to be analysed will be those derived from the steady state calculations. The flow domain consists of three parts: one part modelled with STAR-CCM+ "bounded" by two parts modelled with RELAP5-3D. The length of all the pipe parts modelled with either

Figure 5.10: STAR-CCM+/RELAP5-3D domain subdivision

RELAP5-3D or STAR-CCM+ was 1 m; the diameter was 0.01 m. For these tests RELAP5-3D was used to model the extremities of the pipe (Fig. 5.10).

As a consequence the pressure value of 101325 Pa was imposed at the downstream RELAP5-3D outlet boundary. The value of the friction factor for RELAP5-3D was taken in the RELAP5-3D subdomain downstream of the STAR-CCM+ section, 10 cm from the outlet boundary (blue line in Fig. 5.10). The value of the friction factor for STAR-CCM+ was taken at a distance of 10 cm from the end of the STAR-CCM+ subdomain (red line in Fig. 5.10), using both PG and WS evaluations. Fig 5.11 shows that the PG evaluation slightly underestimates Moody's friction factor. In spite of this, the results match almost perfectly.

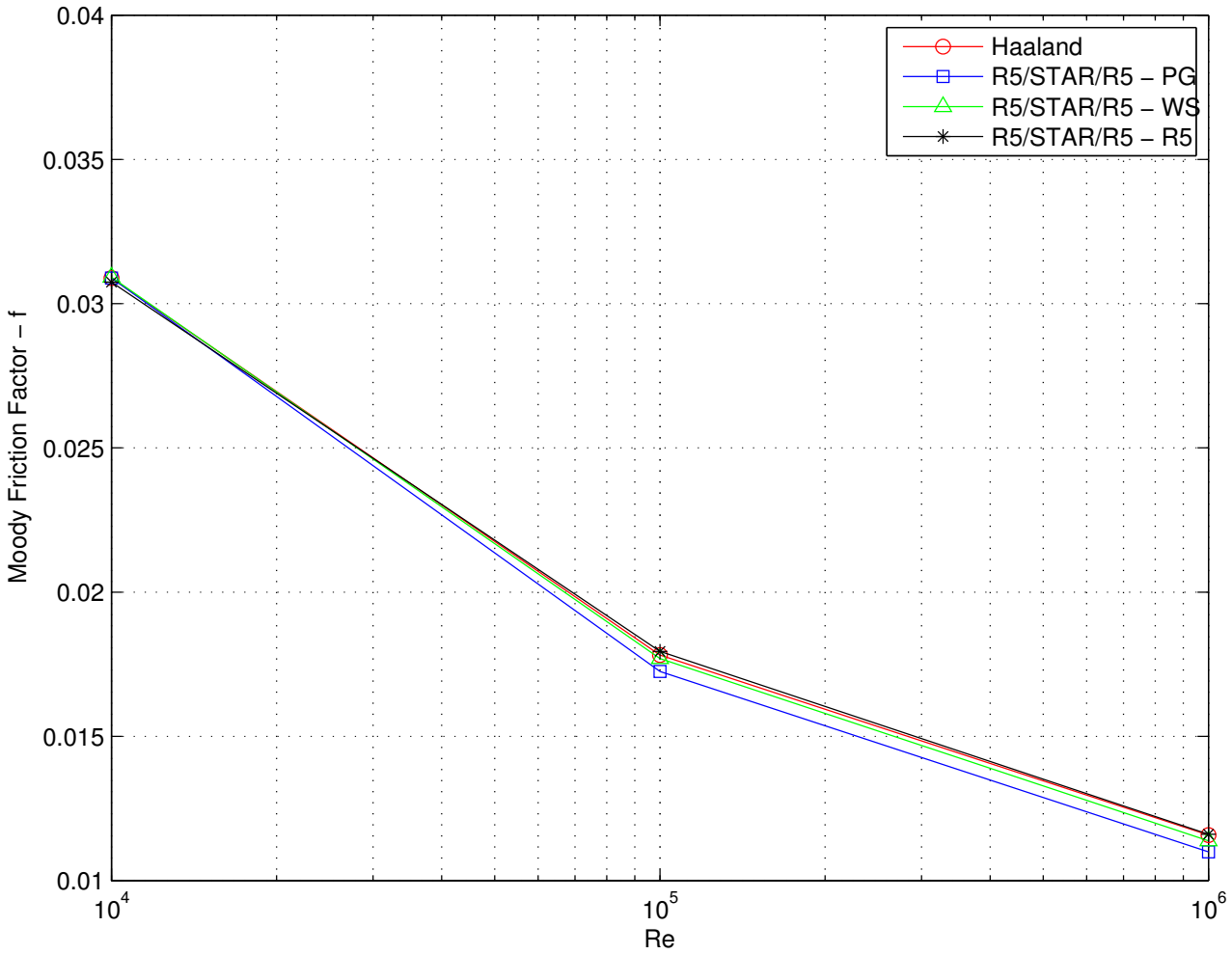


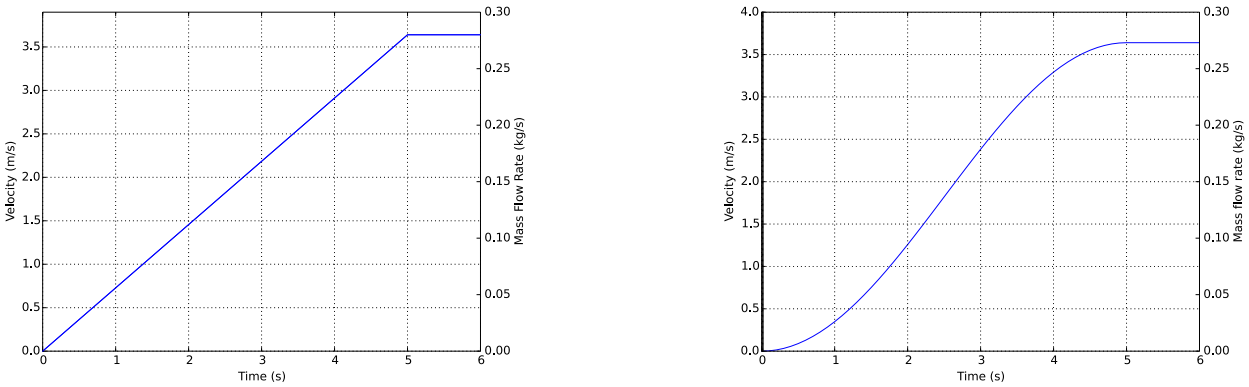
Figure 5.11: Moody's friction factor

5.3.2 Transient analyses

For the transient test cases the same domain as the steady state calculations was used (Fig. 5.10). A pressure of 70 bar was applied to the outlet of the full domain for the transient flow analysis. Fig. 5.12 shows the location of each interface between the two codes.

Figure 5.12: Coupling interfaces location

Two different inlet boundary conditions were applied when analysing transient behaviour: a ramp (Fig. 5.13a) and a cosine (Fig. 5.13b). These two cases were chosen to evaluate the performance of COWS in the presence of singularities (ramp) and in the absence of singularities (cosine). All the results are compared with standalone RELAP5-3D simulations.



(a) Ramp (b) Cosine

Figure 5.13: Transient types

The simulations were run exchanging data with different frequencies, namely every 0.1 s, 0.01 s and 0.001 s. RELAP5-3D will automatically adapt the timestep used in the calculation based on the limits imposed by the Courant number of the simulation. This RELAP5-3D configuration consists of a uniform discretisation with 67 cells of size of 0.015 m for each of the pipe segments upstream and downstream of the STAR-CCM+ segment, thus allowing a maximum time step of $4.12 \cdot 10^{-3}$ s. This value was calculated using eq. 2.5, which for single phase flows simplifies to:

$$C = \frac{u_f \Delta t}{\Delta x} \tag{5.14}$$

The velocity used to calculate the maximum timestep is the maximum velocity imposed as inlet, identifiable in fig. 5.13 as the plateau between 5 and 6 seconds, corresponding to a value of 3.64 m/s. STAR-CCM+ uses a fully implicit time advancement scheme, hence there is no need to calculate the limitations on the timestep imposed by the Courant limit.

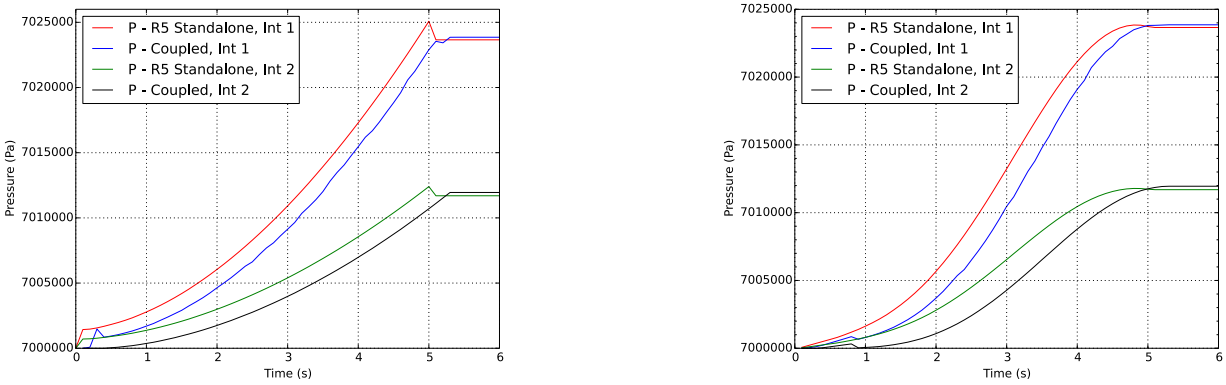
At this point, two remarks on transient simulation are needed. Firstly, momentum imbalance is a well known issue in code coupling because the two codes do not have the all the information required to calculate the $u\nabla u$ term at the pressure boundary locations (Aumiller et al. (2001) and Weaver et al. (2002)) and it was the case for these simulations as well.

Secondly, RELAP5-3D was stopped at each iteration and then started again with RELAP5-3D's restart option when the new input deck was ready. This choice, forced by the unavailability of the source code, led to some instabilities. The cause of these instabilities was possibly due to a bug present in RELAP5-3D: when a simulation is run from start to finish without using the restart option it could yield different results from the same simulation that is stopped and started with the restart option at each step. This problem was identified by Idaho National Laboratory (INL) and put on the improvement list for RELAP5-3D's next release (Mesina and Anderson, 2014).

Data exchange every 0.1 s

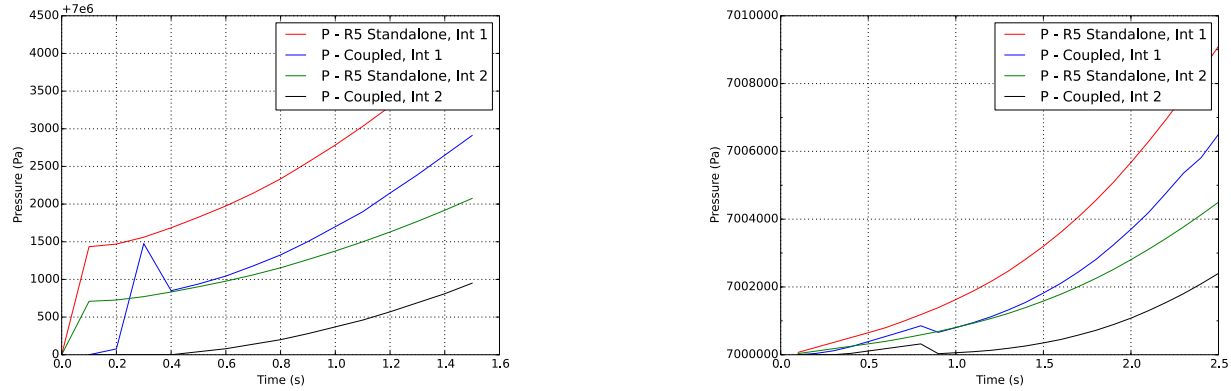
The coupled simulations yield good results in this configuration. However it is possible to see from fig. 5.14 that there is a lag in the coupled results when compared to the standalone RELAP5-3D.

The results present oscillatory behaviour in two locations: the first at the beginning of the simulation and the second around 5 s. Details of the initial oscillations are presented in fig. 5.15 and details of the oscillations around 5 s are presented in fig. 5.16 and fig. 5.17.



(a) Ramp (b) Cosine

Figure 5.14: Pressure at coupling interfaces - Data exchanged every 0.1 s

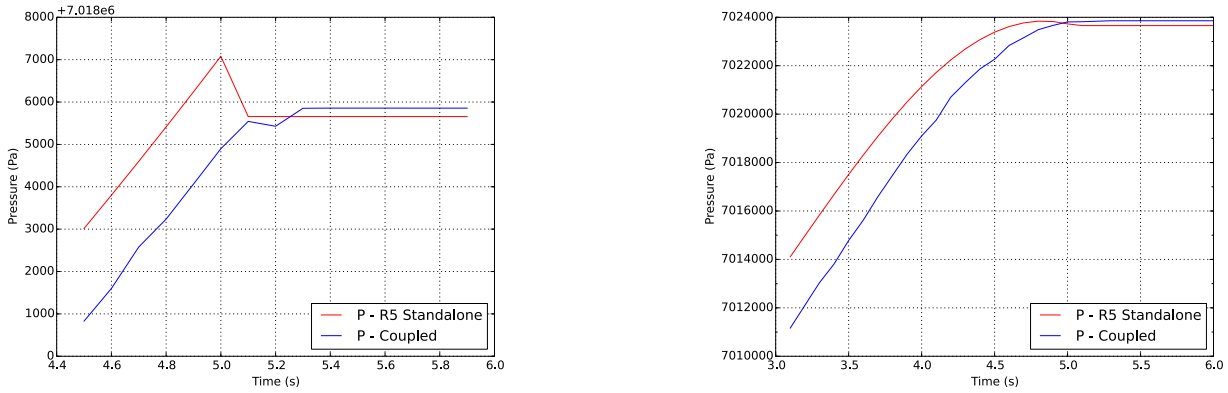


(a) Ramp (b) Cosine

Figure 5.15: Initial pressure transient - Data exchanged every 0.1 s

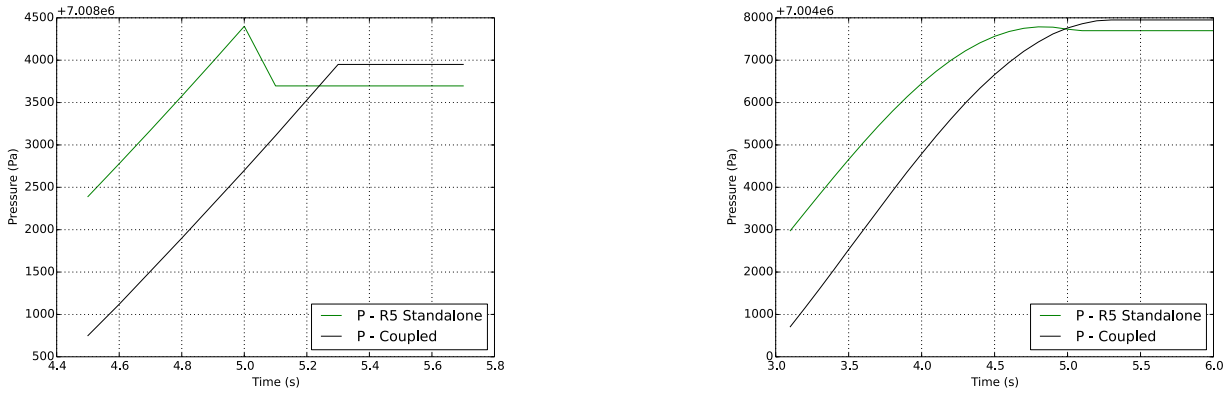
When the inlet mass flow rate is ramped, the standalone RELAP5-3D simulation shows that the initial transient lasts approximatley 0.1 s; in the coupled simulation the oscillation has identical duration but for a lag of approximately 0.2 s and a sudden drop (fig. 5.15a). At 5 seconds the coupled simulation does not reproduce the standalone results, almost missing the transient completely at interface 1 (fig. 5.16a) and not following the trend at interface 2 (fig. 5.17a).

When the inlet mass flow rate is imposed as a cosinusoidal function the coupled system is able to follow the trends better. Fig. 5.15b shows the absence of oscillations in the standalone simulation. However, the coupled results show a small oscillation just before 1 s. With cosinusoidal mass flow rate inlet conditions the transients at 5 s are followed much better by the coupled system (fig. 5.16b and fig. 5.17b).



(a) Ramp (b) Cosine

Figure 5.16: Pressure oscillations at interface 1 around 5 s - Data exchanged every 0.1 s



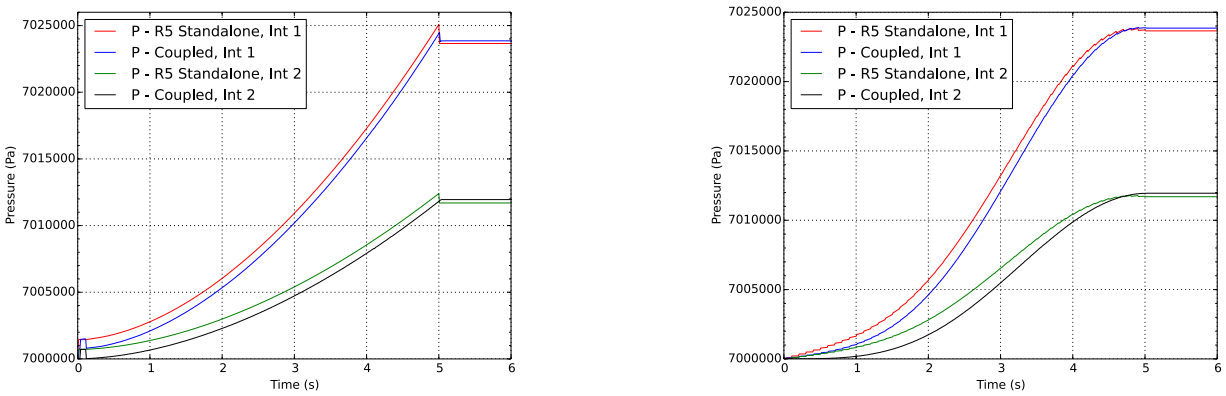
(a) Ramp (b) Cosine

Figure 5.17: Pressure oscillations at interface 2 around 5 s - Data exchanged every 0.1 s

The differences in the oscillatory transient behaviour is deemed to be caused by the use of the RELAP5-3D restart option. Further investigation is needed to validate this hypothesis and these calculations will be performed again with RELAP5-3D's release that fixes this problem.

Data exchange every 0.01 s

In this configuration the coupled simulations yield again good results. Similarly to the previous case, the results from the coupled simulations do not match the standalone simulation perfectly and oscillatory behaviour is observed in two locations: at the beginning of the simulation and at around 5 s (fig. 5.18). Details for the initial oscillations are presented in fig. 5.19 and details for the oscillations around 5 s are presented in fig. 5.20 and fig. 5.21.

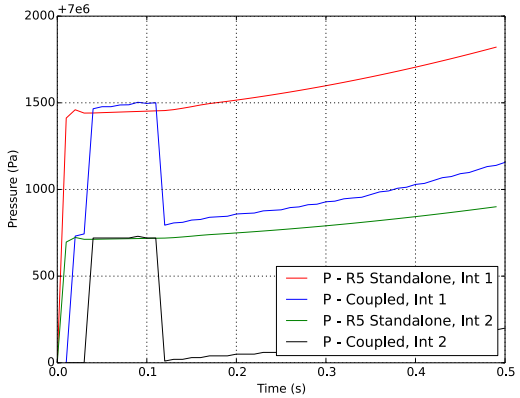


(a) Ramp (b) Cosine

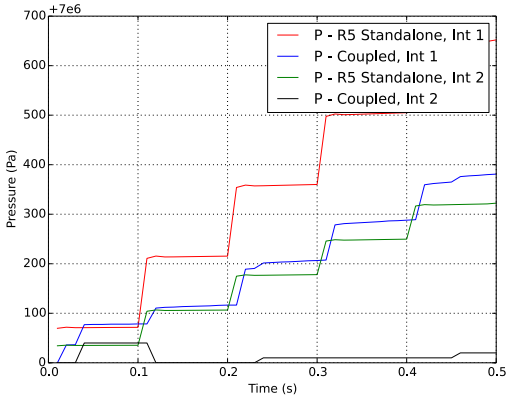
Figure 5.18: Pressure at coupling interfaces - Data exchanged every 0.01 s

When the inlet mass flow rate is ramped, the standalone RELAP5-3D simulation shows that the initial transient lasts approximately 0.01 s; in the coupled simulation the shape of the oscillation is identical but its length is approximately an order of magnitude higher than its standalone counterpart. At 5 seconds the coupled simulation does not follow the standalone results exactly, in fact it has a different oscillation amplitude and a lag in following the transient at interface 1 (fig. 5.20a) and does not reproduce the trend at interface 2 (fig. 5.21a). Differently from the previous case, the duration of the oscillation is comparable between RELAP5-3D standalone and coupled simulations.

Similarly to the case with timestep 0.1 s, when the inlet mass flow rate is imposed as a cosinusoidal function the coupled system performs better. fig. 5.19b shows the initial transient of the standalone simulation. This transient has a stairway shape, which is followed by the coupled system, even if with a lag. Around 5 s the standalone simulation presents a small oscillatory behaviour unobserved when the timestep was 0.1 s. The coupled system captures this oscillatory behaviour at interface 1 (fig. 5.20b), but not at interface 2 (fig. 5.21b). The oscillations captured by the coupled system at interface 1 are damped compared to their standalone counterparts. The cause of the oscillations is deemed to be numerical rather then physical.

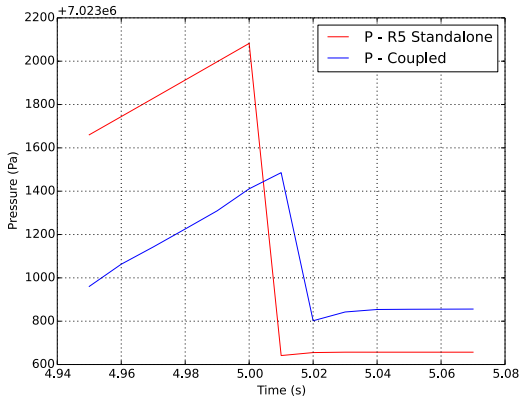


(a) Ramp

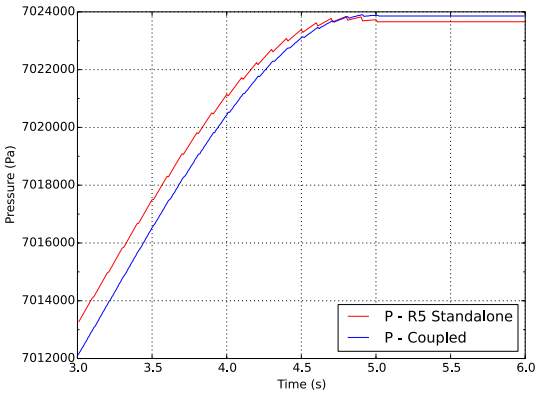


(b) Cosine

Figure 5.19: Initial pressure transient - Data exchanged every 0.01 s

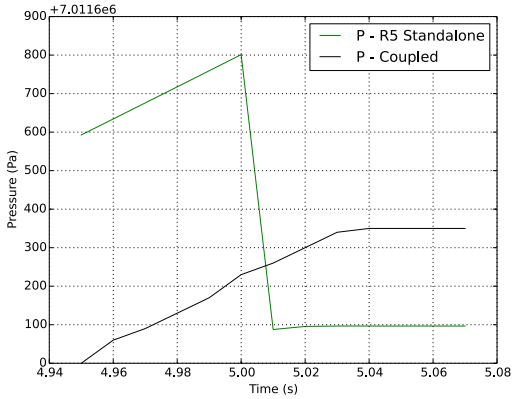


(a) Ramp

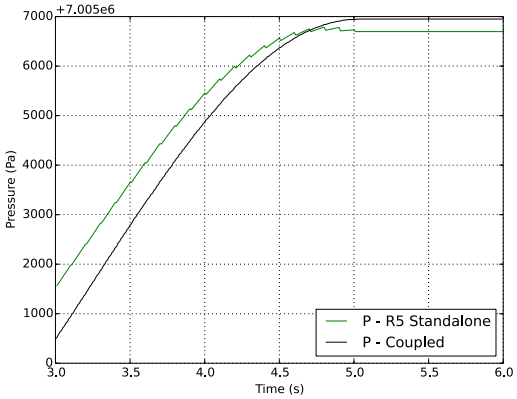


(b) Cosine

Figure 5.20: Pressure oscillations at interface 1 around 5 s - Data exchanged every 0.01 s



(a) Ramp



(b) Cosine

Figure 5.21: Pressure oscillations at interface 2 around 5 s - Data exchanged every 0.01 s

The differences in the oscillatory transient behaviour is deemed to be caused by the use of the RELAP5-3D restart option. Further investigation is needed to validate this hypothesis and these calculations will be performed again with RELAP5-3D's release that fixes this problem.

Data exchange every 0.001 s

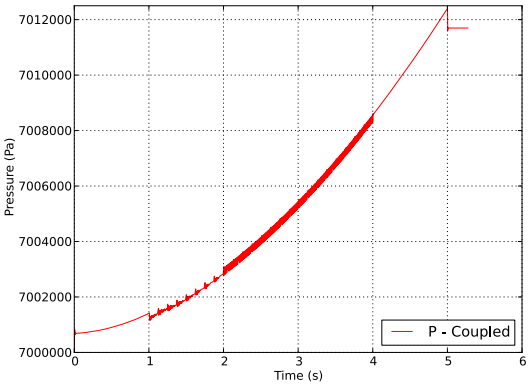
When the timestep is reduced to 0.001 s the coupled system presents a strong oscillatory pressure behaviour between 1 s and 2 s and between 2 s and 4 s (fig. 5.22a). Details of oscillations between 1 s and 2 s are shown in fig. 5.22c, whereas Fig. 5.22d and fig. 5.22e show details of oscillations between 2 s and 4 s.

Other oscillations are present also at the beginning of the simulation (fig. 5.22b) and at around 5 s (fig. 5.22f). The nature of these oscillations seems mostly numerical as their period is twice the simulation timestep. For this reason only a qualitative analysis is presented for this timestep. Furthermore, since the standalone simulation at timestep 0.01 s presented oscillations of numerical nature when the inlet mass flow rate was imposed as a cosinusoidal profile and its ramped counterpart did not, the following discussion will focus only on the case with a ramped mass flow rate inlet condition.

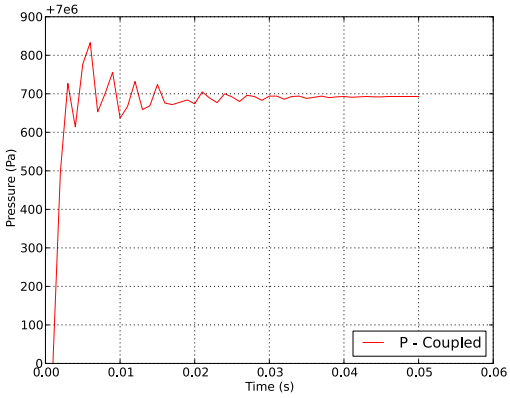
As already mentioned, using the restart option might cause RELAP5-3D to yield different results with respect to a simulation run in normal mode. The pressure oscillations shown in fig. 5.22a are not caused by STAR-CCM+, but by the downstream section of RELAP5-3D.

To ensure that the cause of these oscillations was of numerical nature a variation of this case, identified as "case 2", was run. For this simulation the length of the downstream part of the pipe was changed to be 5 m long by increasing the size of the cells to 0.075 m each. As visible in fig. 5.23a the oscillations seem to have disappeared.

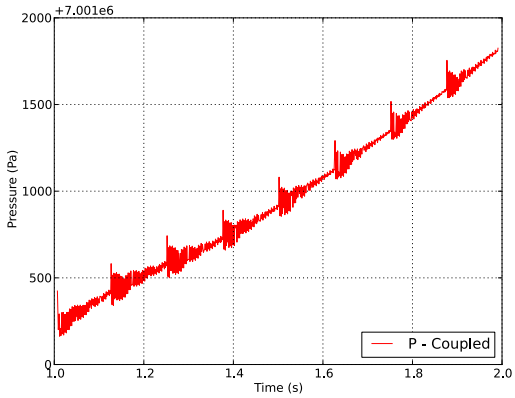
Details presented in fig. 5.23c and fig. 5.23e show that numerical oscillations are still present, although significantly damped. The oscillations at the beginning of the simulation (fig. 5.23b) and at 5 s (fig. 5.23f) present a physical behaviour and do not seem to be influenced by numerical instabilities.



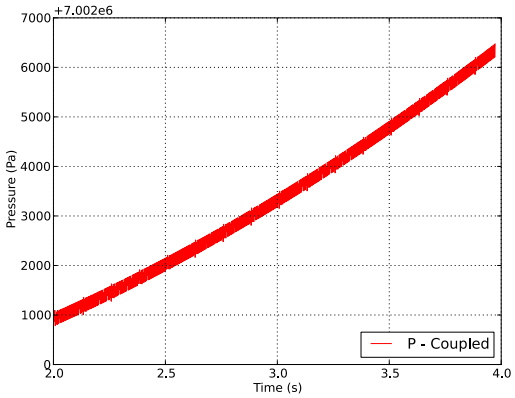
(a) Pressure at interface 2



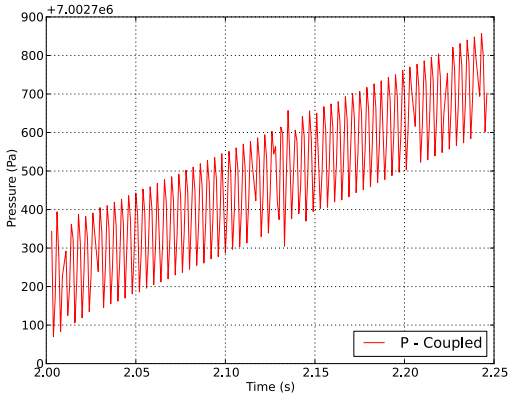
(b) Initial oscillations



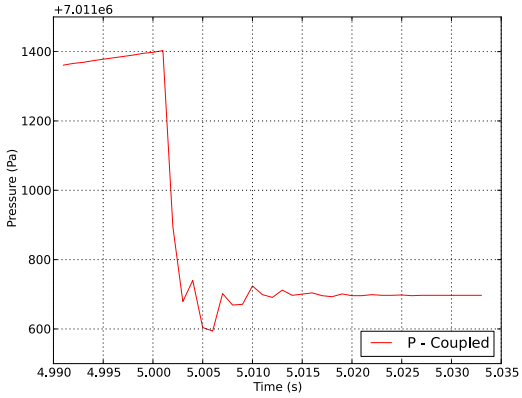
(c) Oscillations between 1 and 2 s



(d) Oscillations between 2 and 4 s

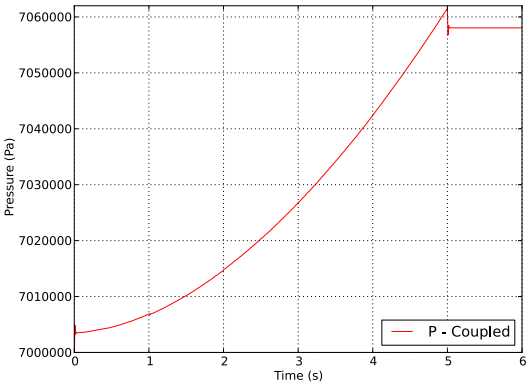


(e) Details of oscillations between 2 and 4 s

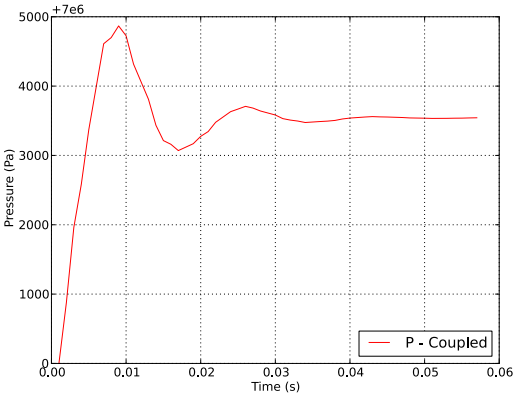


(f) Oscillations at 5 s

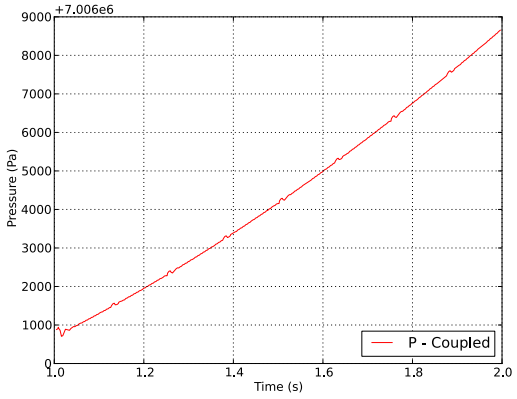
Figure 5.22: Data exchanged every 0.001 s



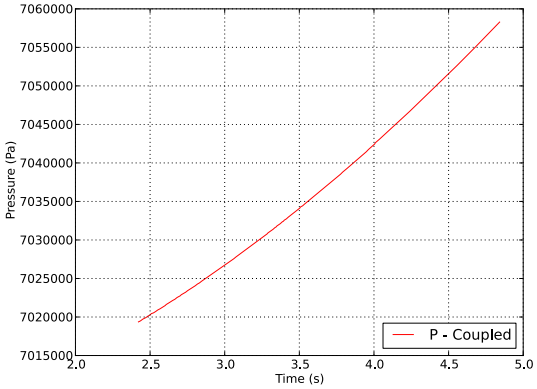
(a) Pressure at interface 2



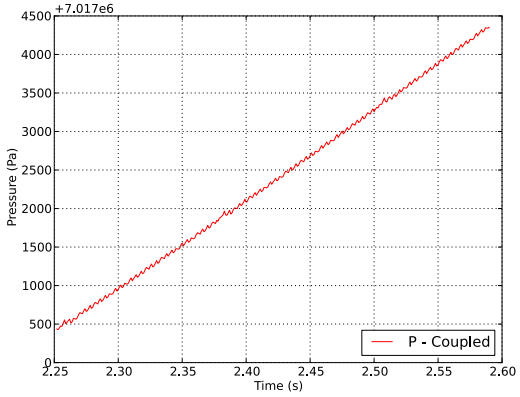
(b) Initial oscillations



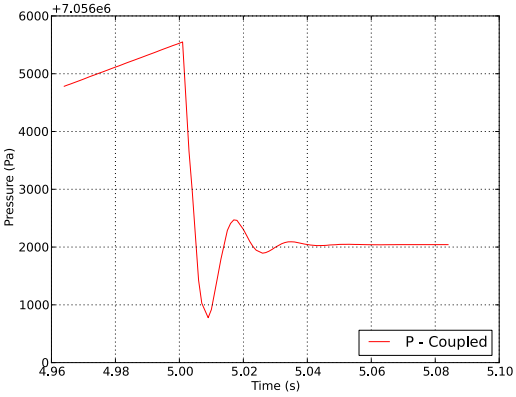
(c) Oscillations between 1 and 2 s



(d) Oscillations between 2 and 4 s



(e) Details of oscillations between 2 and 4 s



(f) Oscillations at 5 s

Figure 5.23: Data exchanged every 0.001 s - case 2

Global considerations on transient calculations

The analysis of the results of the transient cases presented in the previous sections indicates two weaknesses in the methodology presented. The first weakness concerns the data exchange paradigm. It is possible to see from fig. 5.14 and Fig. 5.18 that the results from the coupled simulations present a lag with respect to the results of the standalone simulations. This is due to the explicit data exchange paradigm used by COWS.

At the beginning of the simulation, old-step values are the specified initial values. In the standalone RELAP5-3D model, new-step values are calculated throughout the entire pipe with the new-step values boundary conditions specified at the inlet and outlet. However, in the coupled simulation the pipe is broken into three subsections and a calculation with the same global boundary conditions is performed. The old-step values do not change. The upstream RELAP5-3D part of the pipe is given a new-step boundary value at the inlet.

However, the boundary condition at the outlet cannot be defined in terms of the new-step value because the latter has not been calculated yet. This forces the model to use the old-step value. On the other hand, in the standalone model there is no artificial boundary at this point and new-step values can be calculated normally. Calculations for the downstream RELAP5-3D section are performed at the same time as for the upstream. However, inlet boundary information for the new step of the downstream section is not yet available.

Since the outlet pressure is assumed to remain unchanged, the new-step value is the same as the old-step value. Hence, with no change in inlet and outlet values the flow in the downstream section must remain unchanged from the old step. At this point, calculations for the section modelled with STAR-CCM+ are performed. STAR-CCM+'s outlet boundary values remain unchanged as they come from the downstream RELAP5-3D section.

However, the inlet values for the STAR-CCM+ inlet have changed since new-step results have been received from the upstream RELAP5-3D pipe section. The calculation is done with a mix of new-step and old-step boundary values. Again, when this is compared with a RELAP5-3D standalone model, the latter has no such artificial breaks and no such delay in information

travelling from one end of the pipe to the other. This problem can be fixed by rewriting the coupling algorithm to allow multiple data exchanges per timestep, thus making the algorithm implicit.

The second weakness is a RELAP5-3D problem with simulations run in restart mode: they do not yield the same results of simulations run without restarts. This is deemed to be the cause of the difference of the transient behaviour between the coupled simulations and their standalone counterparts for timesteps 0.1 s and 0.01 s for both ramp and cosine inlet boundary conditions, as well as the cause for the oscillations for timestep 0.001 s.

As already mentioned, this problem was identified by INL and put on the improvement list for RELAP5-3D's next release (Mesina and Anderson, 2014). When a new version of RELAP5-3D containing a fix is released, the same tests will be performed again to validate the hypothesis that RELAP5-3D's problems with the restart mode caused the oscillatory behaviour of fig. 5.22a.

The computational times were not recorded for the smallest timestep due to the simulations not being run until the end because of the aforementioned oscillations and for the cosine transient cases. For the other two cases the computational times are presented in table 5.1.

Timestep (s)	**COWS computational time (s)**
0.1	2382
0.01	9840

Table 5.1: Computational times in seconds

It is possible to see that the simulation run with timestep 0.1 s presents a computational time roughly 25% of the simulation run with timestep 0.01 s.

The simulations run to validate COWS for transient single phase flows are very simple cases, not comparable with the complexity of a full reactor simulation. RELAP5-3D uses an adaptive time step selection algorithm, hence the "ideal" timestep will be depending on the specific problem, however it is unlikely that it will be coarser than the ones used in these verification exercises. This can be seen in the example input decks (INL, 2014c) available with the installation of RELAP5-3D suggest, which also show that the size of the computational model of an industrial

size problem will be significantly larger than the one presented in this chapter.

Furthermore, as explained in Popov et al. (2012), in order to simulate a reactor vessel in its entirety, extremely large computational facilities are needed: for a CFD model of 1.035 billion cells run on 800 cores, the computational times were of roughly 150 minutes to run 100 steady state iterations. To converge to acceptable results it took roughly 1500 iterations.

The computational times of COWS are influenced by the use of RELAP5-3D's restart option, thus introducing additional computational costs related to the loading of RELAP5-3D's restart file. A more detailed discussion on these additional computational costs is presented in section 6.2.3. Further analyses are needed to quantify the magnitude of the restart file loading operations.

It is clear, based on the results from Popov et al. (2012), that the current implementation of COWS is not suitable for the simulation of industrial size cases. This is due to the additional computational times introduced by the loading operations of RELAP5-3D's restart file. This is also discussed further in section 6.2.3.

5.4 Inlet velocity profile reconstruction

It is explained in chapter 4 how adding a separate inlet subdomain that uses periodic boundary conditions with specified mass flow rate or pressure drop to the STAR-CCM+ model allows the reconstruction of fully developed, single phase inlet velocity and temperature profile maps from average one dimensional values while minimising the additional computational costs and avoiding the need for a database for inlet profiles of different quantities.

This technique was used for a steady state, purely hydraulic, coupled simulation the results of which are shown in fig. 5.24. It is possible to see the section of the periodic domain where the variables were mapped. The mass flow rate imposed by RELAP5-3D is 0.28 kg/s and the quantites mapped in the periodic domain are axial velocity, turbulent kinetic energy and turbulent dissipation rate. The periodic domain is 0.01 m in length, whereas the main

computational domain is 0.2 m. Both are 0.01 m in diameter.

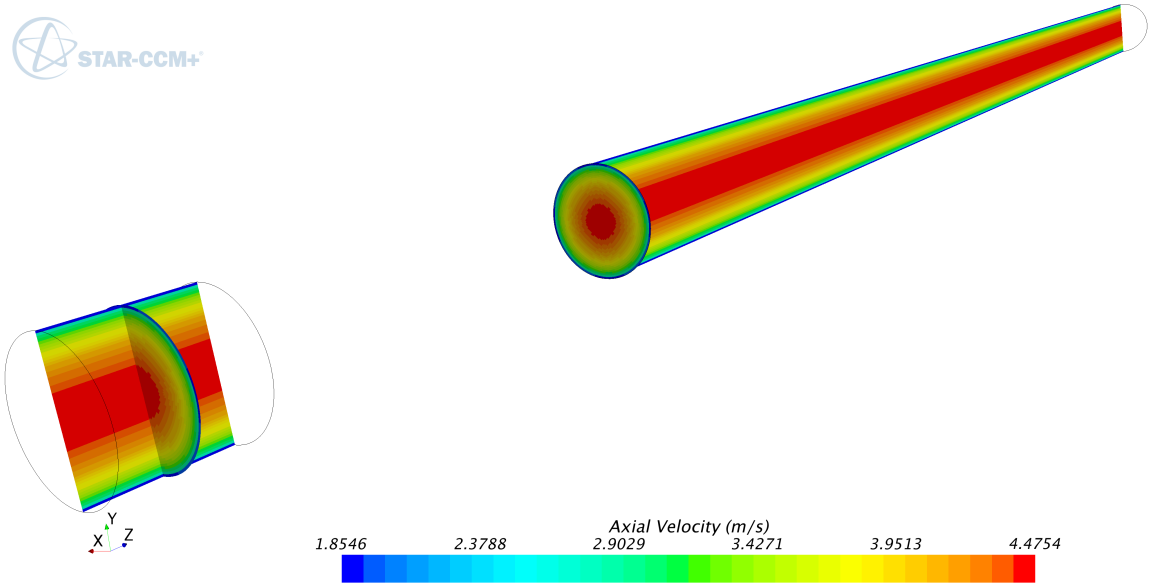


Figure 5.24: Periodic boundary inlet profile reconstruction - Axial velocity

Fig. 5.24 shows how the velocity profile does not vary throughout the whole pipe, thus confirming that the flow is fully developed. Mapping the turbulence-related quantities is critical for the reconstruction of coherent fully developed flow conditions. It would not be sufficient to map solely the axial velocity because STAR-CCM+ needs information on turbulence related quantities at the inlet. As a consequence, not mapping the turbulence related quantities would result in applying turbulence boundary conditions that are not coherent with the applied velocity profile. This translates in a flow that will not be fully developed at the inlet but at a point located downstream of the inlet.

The same configuration was used for a steady state, coupled simulation where the temperature profile was reconstructed using the periodic boundary interface method. For this case one further variable was needed to be passed from RELAP5-3D: the wall temperature. This temperature was applied to the walls of the periodic domain. The variables mapped in the periodic domain were axial velocity, turbulent kinetic energy, turbulent dissipation rate and fluid temperature. The results are shown in Fig. 5.25, where it is possible to see that the inlet temperature of the computational domain is mapped correctly from the periodic domain. For both simulations the cross section of both domains are meshed in the same way.

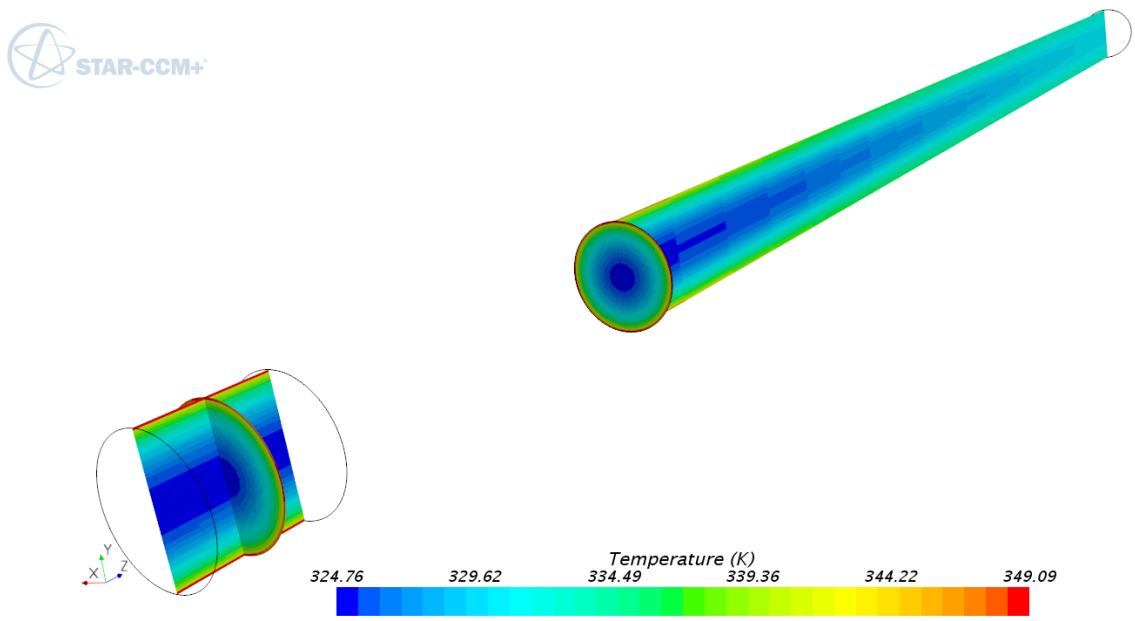


Figure 5.25: Periodic boundary inlet profile reconstruction - Temperature

The ability to reconstruct fully developed inlet profiles allows to drop, or at least reduce, the use of flat inlet profiles. The latter are a less accurate representation of the flow at inlet boundaries especially compared to a fully developed inlet reconstruction. It can be argued that a flat inlet profile is a non-physical condition hence, especially for industrial applications, choosing a coupling interface at a point where the flow is fully developed and imposing a fully developed inlet, will provide a better boundary condition.

The need for the reconstruction of a proper inlet is even more evident in multiphase flows. The presence of more than one phase introduces the problem of having not only correct velocity, turbulence and temperature profiles, but also a correct cross sectional distribution of the various phases.

In the analyses presented in the next chapter, a flat inlet profile was used in the coupled simulations because STAR-CCM+ does not allow the use of periodic boundary interfaces with multiphase flows. It can be seen in the results how this modelling choice impacted both the average results and the radial void fraction distribution.

5.5 Summary

The range of applications where STAR-CCM+'s own coupling tool can be used is limited to single phase flows where only STAR-CCM+'s inlets are coupled.

On the other hand, COWS yields good results in both transient and steady state cases albeit implementing an explicit data exchange paradigm. When transients do not present singularities or abrupt changes, the coupled system is able to follow them well, in spite of the lags due to the explicit data exchange paradigm. Periodic boundary conditions were used to reconstruct an inlet fully developed profile for velocities and temperatures from values coming from RELAP5-3D.

New tests to validate the hypothesis of RELAP5-3D's use in restart mode being the cause of numerical oscillations are needed.

Chapter 6

Multiphase coupling

Following the results presented in chapter 5, COWS capabilities were extended to allow for the simulation of multiphase flows. The extension consisted in exchanging temperature, mass flow rate and void fraction of the gas phase in addition to the existing variables for the single phase flows, namely temperature and mass flow rate of the liquid phase and pressure. Four cases from the PSBT benchmark were used to validate COWS's multiphase capabilities.

This chapter shows the performance of COWS when modelling multiphase flows. All the simulations were of steady state cases, because the data exchange algorithm is explicit. As explained in chapters 2, 4 and 5, and more in detail in Aumiller et al. (2001) and Weaver et al. (2002), an explicit data exchange algorithm would be unsuitable for transient multiphase flow simulations because the sonic Courant limit would be stringent, requiring smaller timesteps for the simulation to converge, thus resulting in larger runtimes. Furthermore, the coupled system is susceptible to instabilities when using an explicit algorithm (Aumiller et al., 2002) for transient problems. For these reasons the four cases from the PSBT benchmark were chosen among the steady state cases.

Unfortunately, the physics models chosen in STAR-CCM+ did not allow to use periodic boundary conditions, hence a study was carried on the effect of the location of the coupling interface on the void distribution in the channel.

6.1 Model description

The PSBT benchmark consists of a set of void distribution and DNB exercises at PWR rated conditions. It is divided into two phases: Phase I consists of void distribution benchmarks and phase II consists of DNB benchmarks. To assess the multiphase performance of COWS, four cases from the steady state void distribution exercises of phase I were chosen. The exercises were chosen from the steady state group to avoid issues related to the explicit nature of the data exchange algorithm, as explained in chapter 4.

The PSBT test facility is shown in fig. 6.1. It consists of a high pressure and temperature recirculation loop, a cooling loop and instrumentation and data acquisition systems. The recirculation loop comprises the test section, the circulation pump, the preheater, the steam drum and the water mixer. The design pressure is 19.2 MPa and the design temperature is 362 °C. The effective heated length is 1.555 m long and the measurement section is at 1.4 m from the inlet. The walls are heated at a constant rate.

The computational model consists of the flow area of the fully-heated channel, highlighted by the blue box in fig. 6.1. The tests chosen to validate the performance of COWS were the same as those presented in the CFD section of Kim et al. (2012). The conditions for each test are listed in table 6.1.

Test ID	**P (bar)**	**T (K)**	$\dot{\mathbf{Q}}$ **(kW)**	$\dot{\mathbf{m}}$ **(kg/s)**
1.2223	147.198	592.75	69.8	0.3246
1.2237	147.394	602.75	60.0	0.3252
1.4325	98.361	526.95	59.8	0.1496
1.4326	98.165	541.95	60.1	0.1493

Table 6.1: PSBT test cases conditions

Standalone STAR-CCM+ and RELAP5-3D simulations were run before attempting a coupled approach. When coupled, each of the PSBT test cases in table 6.1 was further subdivided into four different configurations, varying the position of the coupling interface along the subchannel axis, as shown in fig. 6.1. For every test case the domain below the interface was modelled with RELAP5-3D and the domain above the interface was modelled with STAR-CCM+. The location of each interface is shown in table 6.2.

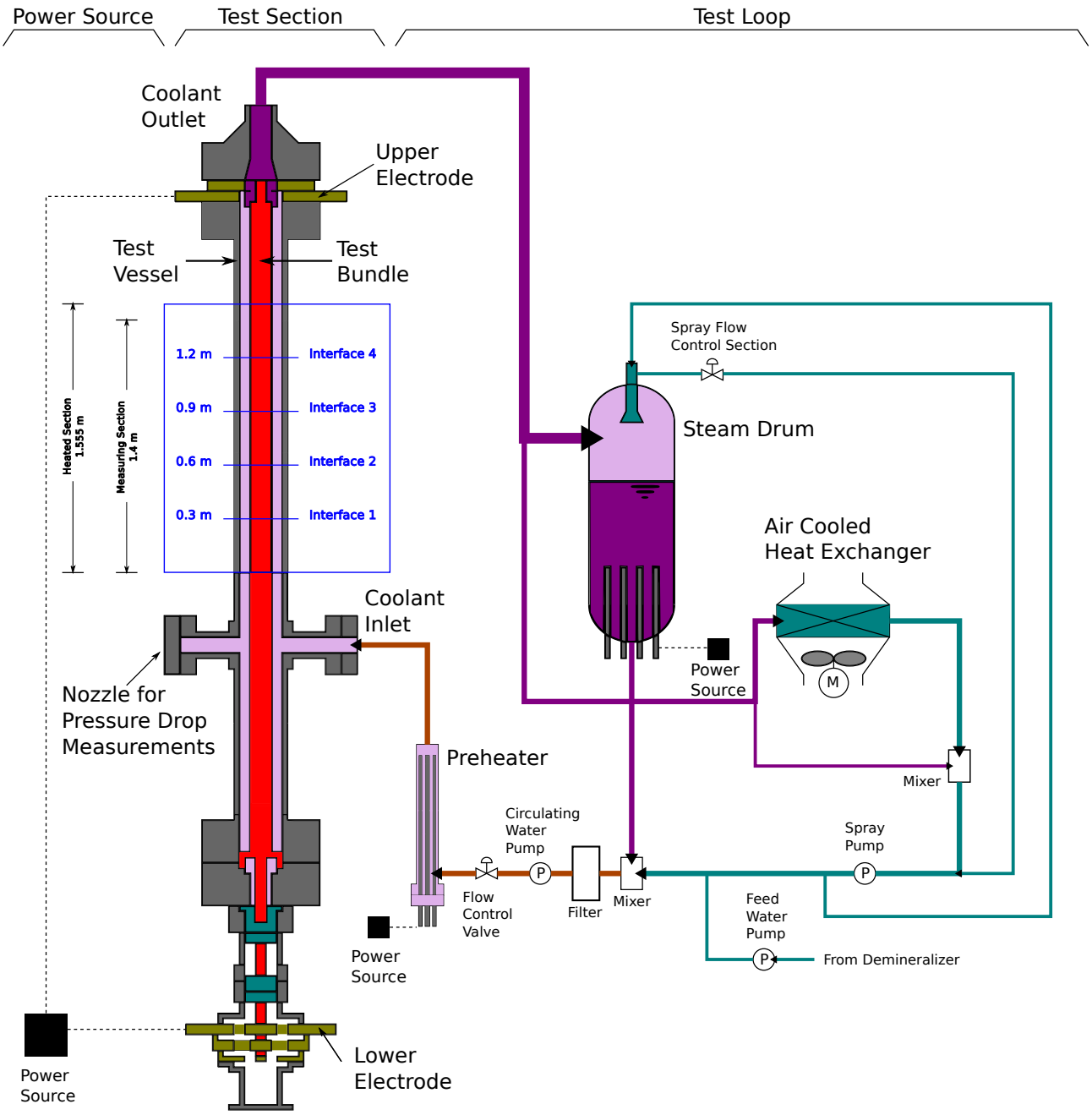


Figure 6.1: PSBT loop - Adapted from Rubin et al. (2010)

Interface	**Axial location (m)**
Interface 1	0.3
Interface 2	0.6
Interface 3	0.9
Interface 4	1.2

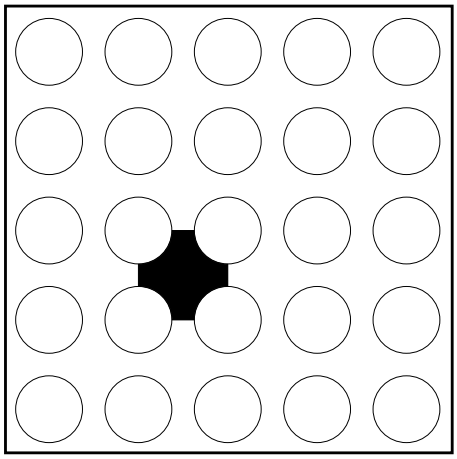
Table 6.2: Interface locations

The results from the coupled simulations were compared with the results of standalone simulations and experiments. The gamma-ray transmission mehtod was used to measure the void

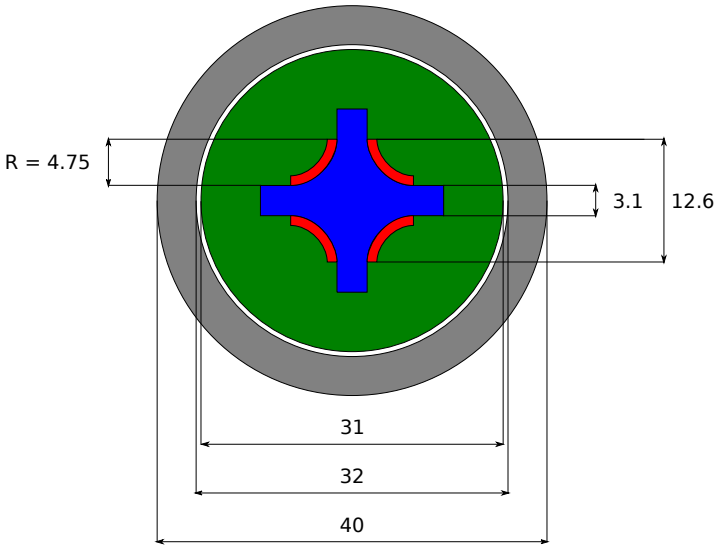
fraction in each experiment. The gamma ray source used in the experiments was ^{137}Cs. Explaining the measurement procedure would be out of scope for this work, for more details it is recommended to read NUPEC (1989) and Hori (1993).

The only experimental data available was the average of the void fraction at the measuring point. Information on the pressure, phasic velocities and axial development of the void fraction was not available. The radial void dirstribution was available only in graphical format; for this reason it was not possible to perform a quantitative comparison.

The geometry of the subchannel cross section is shown in fig. 6.2. The blue zone represents the flow area, the red zone represents the heaters (Inconel 600), the green zone represents the insulator (alumina) and the grey zone represents the vessel (titanium).



(a) Identification of the subchannel in a PWR-like assembly



(b) Subchannel geometric parameters in mm

Figure 6.2: PSBT test section cross section geometry - Adapted from Rubin et al. (2010)

As explained in Kim et al. (2012), to minimise the need for computational resources the symmetry of the subchannel was exploited. As a result, it was possible to reduce the size of the model by a factor of 8. The resulting geometry and its related boundary types are shown in fig. 6.3. The boundary types are set as follows: the green lines are symmetry boundaries, the black lines are adiabatic walls and the red line is the heated wall. A mesh sensitivity study was performed on all simulations.

In the CFD model an Eulerian-Eulerian description with the wall boiling heat transfer model

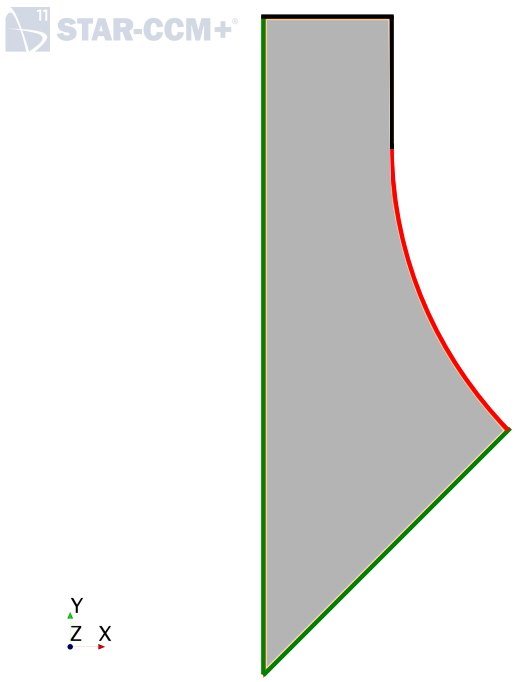


Figure 6.3: Cross section of the CFD model

(Kurul and Podowski, 1990) was used. The forces acting on the bubbles that were considered in the simulation were the turbulent dispersion force and the drag force. The lift force in the transversal direction was neglected because the simulations would not converge otherwise. All the calculations were performed keeping the physical properties constant at the saturation state corresponding to the operating pressure. The realisable $k - \varepsilon$ turbulence model with STAR-CCM+'s default wall treatment options was used.

In the RELAP5-3D model only the compulsory base thermal hydraulic modules for each of the hydraulic components were used.

6.2 PSBT benchmark results

The quality of the results of coupling tools is clearly bounded by the quality of the results of their components. For this reason, the purpose of running standalone prior to coupled simulations was to gather information on the results that can be expected from the coupled simulations.

6.2.1 Standalone simulations

Fig. 6.4 through 6.7 compare the void distribution between experimental and CFD results at the measuring point. It is not possible to perform quantitative comparisons since only graphical data was available for the radial void fraction distribution. The void distribution is always peaked towards the heated wall, even when the experimental results show that the peak is in the centre (fig. 6.7). The reasons for this is attributed to the omission of the lift force, which pushes the bubbles away from the heated wall and towards the centre of the channel.

Test ID	PSBT	RELAP5-3D	STAR-CCM+
1.2223	31.1 ± 4.0 %	28.40 %	34.41 %
1.2237	44.0 ± 4.0 %	35.92 %	38.24 %
1.4325	33.5 ± 4.0 %	40.20 %	47.84 %
1.4326	53.1 ± 4.0 %	52.76 %	57.95 %

Table 6.3: Average void fraction at measuring point - Standalone simulations

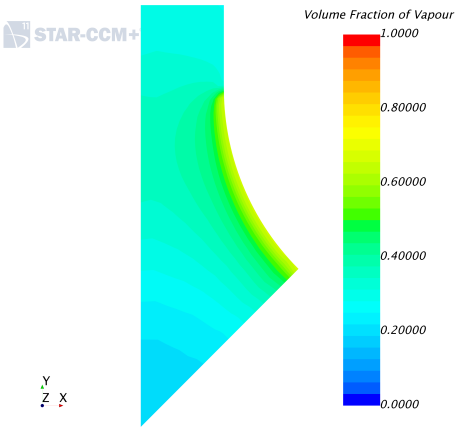
Table 6.3 shows the comparison between the standalone simulations and the experimental data. Overall, results are satisfactory. The void distribution are reproduced reasonably well in all cases.

It is necessary to remark that standalone analyses were performed to understand the level of performance to be expected by the codes chosen for the coupling, since the performance of COWS is limited by the performance of the models of the single codes. The focus of this work does not seek to improve the physics models implemented in either STAR-CCM+ or RELAP5-3D; instead it aims to give a better understanding of the interaction of these two programs when modelling complex scenarios such as multiphase flows in specific conditions.

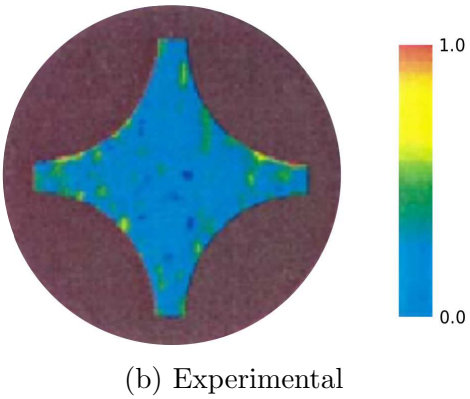
The discrepancies between the experimental results and the simulations were also observed in some of the results presented in Rubin and Avramova (2011). The cause of this is attributed to a combination of model choice and model performance.

Compared to STAR-CCM+, RELAP5-3D yields lower average void fraction predictions, which can also be seen in the axial void fraction profiles of fig. 6.8.

The reason for this can be attributed to the different treatment of boiling in the two codes.

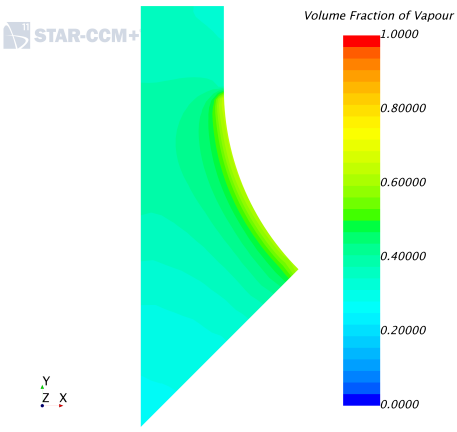


(a) STAR-CCM+

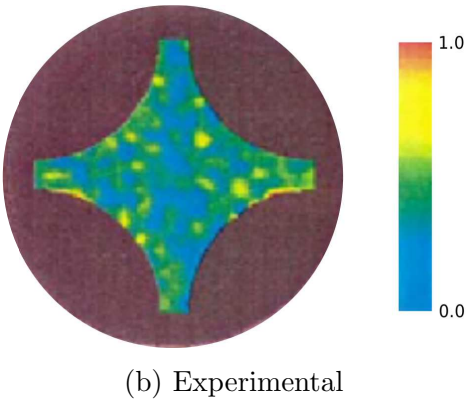


(b) Experimental

Figure 6.4: Void fraction distribution at measuring point - 1.2223

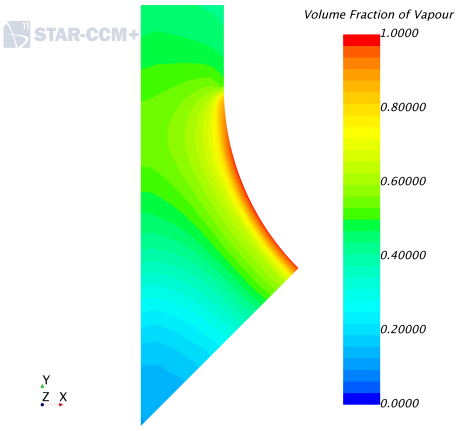


(a) STAR-CCM+

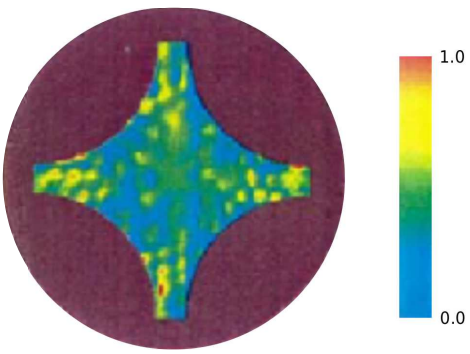


(b) Experimental

Figure 6.5: Void fraction distribution at measuring point - 1.2237

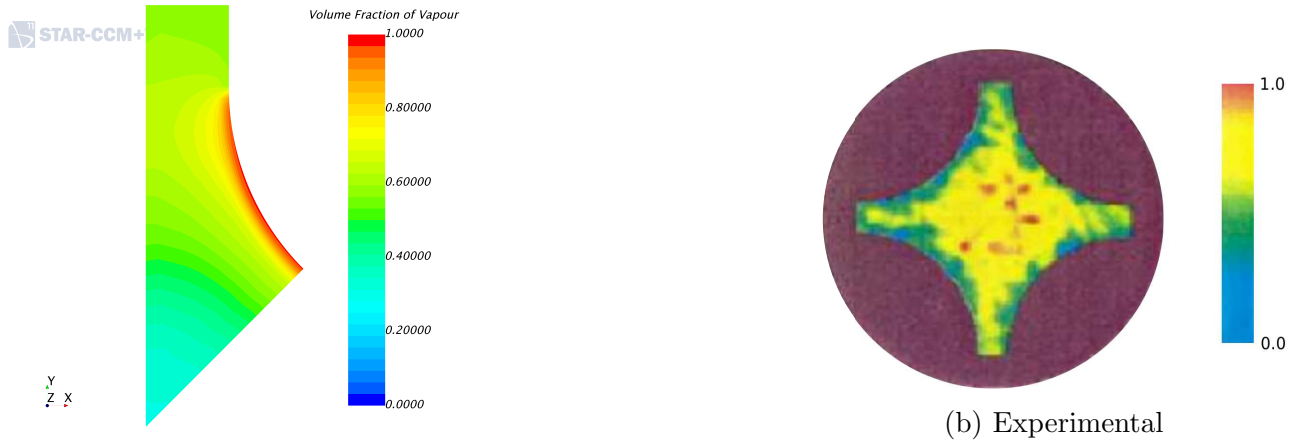


(a) STAR-CCM+



(b) Experimental

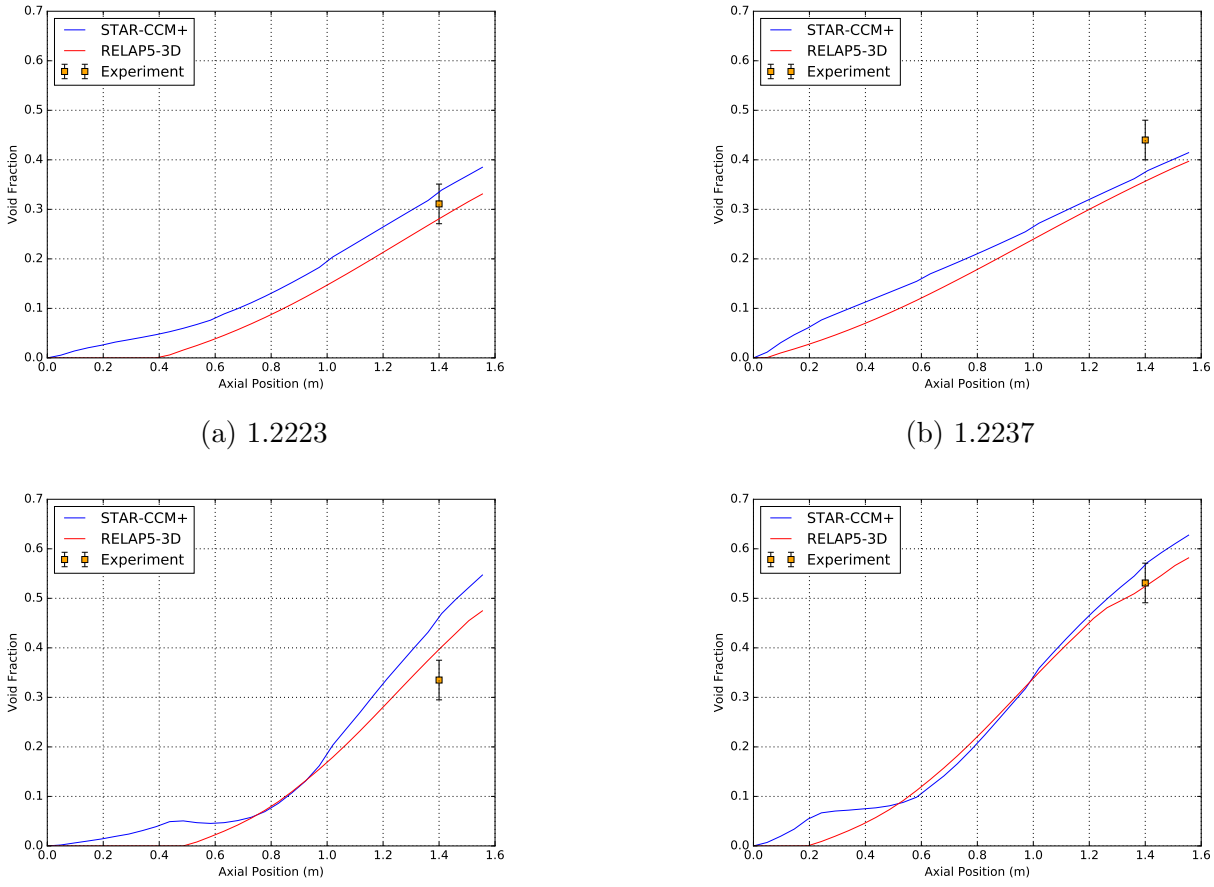
Figure 6.6: Void fraction distribution at measuring point - 1.4325



(a) STAR-CCM+

(b) Experimental

Figure 6.7: Void fraction distribution at measuring point - 1.4326



(a) 1.2223

(b) 1.2237

(c) 1.4325

(d) 1.4326

Figure 6.8: Axial void fraction profile for standalone simulations

Both formulations rely on identifying a bulk boiling region and a wall boiling region. While boiling in the bulk region is treated by the two codes in a similar way, boiling in the wall region is not. The differences in modelling boiling in the wall region will be briefly outlined in the

following paragraphs. For an exhaustive description of the models, it is recommended to read CD-adapco (2016) and INL (2014a).

The boiling model in STAR-CCM+ uses a mechanistic approach, modelling the wall boiling region with the heat flux partitioning approach proposed by Kurul and Podowski (1990).

RELAP5-3D relies on a different approach. During nucleate boiling the wall interphase heat transfer for the gas phase is zero and the wall interphase heat transfer for the liquid phase is divided into two parts: A part treated as a convective heat flux and a part resulting in the saturated pool boiling from the liquid phase. When boiling exists, a fraction of the energy is accumulated in the variable representing the wall interphase mass transfer rate due to the wall heat transfer.

The absence of information about transversal gradients led to the implementation of a different model to calculate the wall interphase mass transfer associated with the wall heat transfer, especially for subcooled boiling. During subcooled boiling the fluid in the bulk region can be subcooled while fluid near the wall can be changing phase, thus yielding a net vapour generation.

The method of Saha and Zuber (1974) for the prediction of the necessary conditions for net vapour generation is used to determine the part of the wall interphase heat transfer associated with the wall interphase mass transfer. The method consists of relying on the Peclet number of the flow to relate the total heat flux to either the Nusselt number or the Stanton number of the flow. The threshold value for the Peclet number is 70000. Below this value the total heat transfer is associated to the Stanton number, above this value the total heat transfer is associated to the Nusselt number.

The specific enthalpy of the liquid phase is then compared to a critical enthalpy, the value of which is a function of either the Nusselt or Stanton number. If the specific enthalpy of the liquid phase is greater than the critical enthalpy RELAP5-3D will calculate the wall interphase mass transfer using the model presented by Lahey (1978).

Estimation of the lift force

The lift force was neglected because the simulations would not converge if the lift force model was activated. An estimation for the order of magnitude of the lift force will be carried out to assess whether its impact is negligible. For continuous-dispersed phase interaction, STAR-CCM+ computes the drag and lift forces acting on the dispersed phase as follows:

$$\mathbf{F}_D = A_D \mathbf{u}_r \tag{6.1}$$

$$\mathbf{F}_L = f_l C_L \alpha_f \alpha_g \rho_f [\mathbf{u}_r \times (\nabla \times \mathbf{u}_f)] \tag{6.2}$$

Where:

- $\mathbf{F}_D$: Drag force
- $\mathbf{F}_L$: Lift force
- C_L: Lift coefficient
- $\mathbf{u}_r$ Relative velocity
- A_D: Linearised drag coefficient
- f_l: Lift correction factor

The relative velocity is defined as:

$$\mathbf{u}_r = \mathbf{u}_f - \mathbf{u}_g \tag{6.3}$$

And the linearised drag coefficient is defined as:

$$A_D = \frac{1}{2} C_D \rho_f |\mathbf{u}_r| \frac{a_{fg}}{4} \tag{6.4}$$

Where C_D is the drag coefficient and a_{fg} is the interfacial area density, measured in m^2/m^3. The factor $a_{fg}/4$ represents the projected area of the equivalent spherical particle. The linearised drag coefficient is measured in $kg/(m^3 s)$.

The drag force model used in the simulations relies on the Tomiyama contaminated drag coefficient correlations (CD-adapco, 2016) and can be extracted from the simulations. It is recommended (CD-adapco, 2016) to use the Tomiyama lift coefficient correlation when using the Tomiyama drag coefficient correlation.

The Tomiyama lift coefficient correlation is as follows:

$$C_L = \begin{cases} 0.288 \cdot tanh(0.121 \cdot max[Re, 7.374]) & Eo_d < 4 \\ 0.00105 Eo_d^3 - 0.0159 Eo_d^2 - 0.0204 Eo_d + 0.474 & 4 \leq Eo_d \leq 10 \\ -0.27 & Eo_d > 10 \end{cases} \tag{6.5}$$

Where Re is the Reynolds number and Eo_d is the modified Eotvos number:

$$Eo_d = Eo \cdot E^{-2/3} \tag{6.6}$$

E is an empirical correlation for the bubble aspect ratio and is defined as:

$$E = \frac{1}{1 + 0.163 Eo^{0.757}} \tag{6.7}$$

Eotvos number is defined as:

$$Eo = \frac{|\rho_c - \rho_d| g l^2}{\sigma} \tag{6.8}$$

Where:

- ρ_c: Density of continuous phase
- ρ_d: Density of dispersed phase

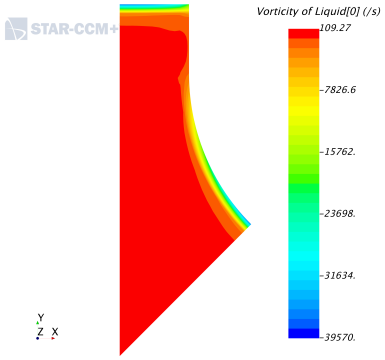
- g: Gravity acceleration
- l: Interaction length scale
- σ: Surface tension

As previously stated, if the lift force model was activated the simulations would not converge. However, thanks to STAR-CCM+ post-processing capabilities it was possible to use equations 6.1 and 6.2 to display the field distribution of drag and lift force densities (N/m^3) at the measuring point, as visible from figures 6.9, 6.10, 6.11 and 6.12. For completeness, relative velocity and $\nabla \times \mathbf{u}_f$ (i.e. vorticity of the liquid phase) distributions are also shown. The lift correction factor was set to 1.

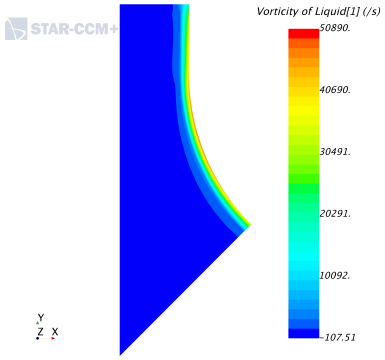
It is important to remark that the calculation of the lift force field was performed on the solution obtained without the lift model. This field differs from the one resulting from a coupled computation with the lift model activated; nonetheless it is a good indicator for the relative importance of the lift force with respect to the drag force.

Figures from 6.9 to 6.12 show that a strong lift force acts on the vapour phase, in the negative direction of the axes perpendicular to the flow direction. As expected, the lift force is strongest at the heated wall, where the void fraction has its highest value. The order of magnitude of these components of the lift force is at least an order of magnitude higher than the drag force components, as shown by the figures.

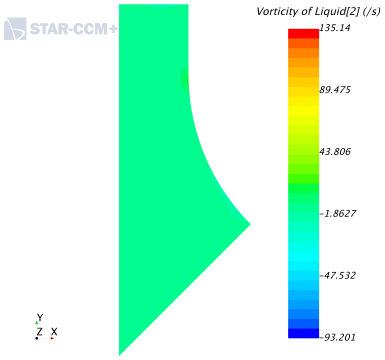
From this analysis it follows that the magnitude of the lift force is not negligible. Its absence in the simulations is deemed to be the cause of the discrepancy in the calculated void fraction radial distribution and the experimental data (Fig. 6.4 through 6.7).



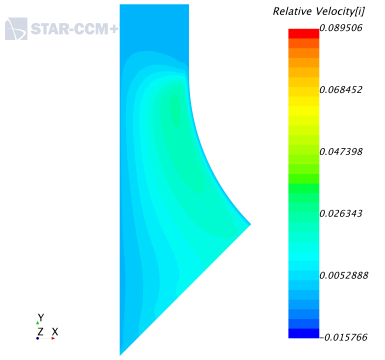
(a) $\nabla \times \mathbf{u}_f$ - X component



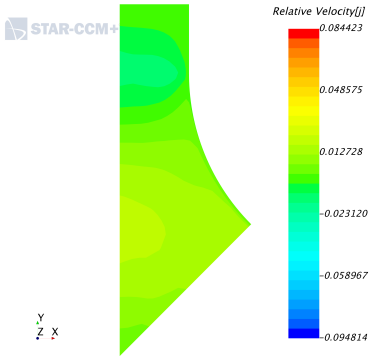
(b) $\nabla \times \mathbf{u}_f$ - Y component



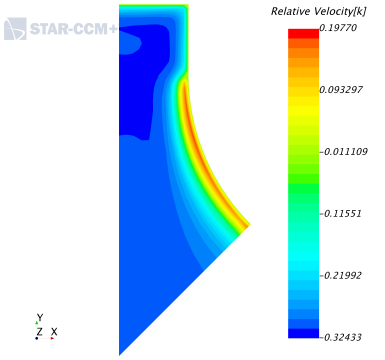
(c) $\nabla \times \mathbf{u}_f$ - Z component



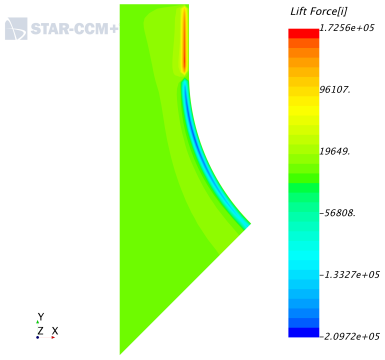
(d) $\mathbf{u}_r$ - X component



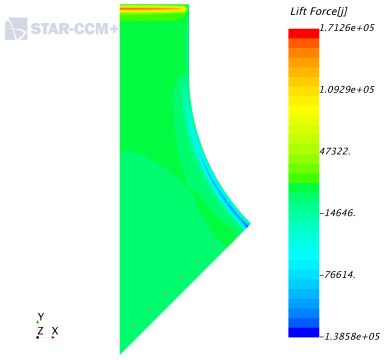
(e) $\mathbf{u}_r$ - Y component



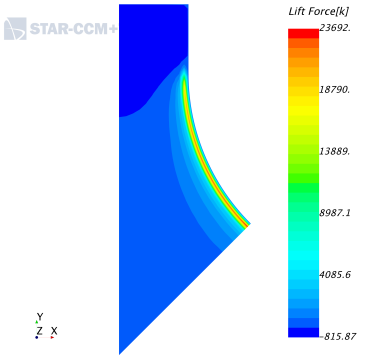
(f) $\mathbf{u}_r$ - Z component



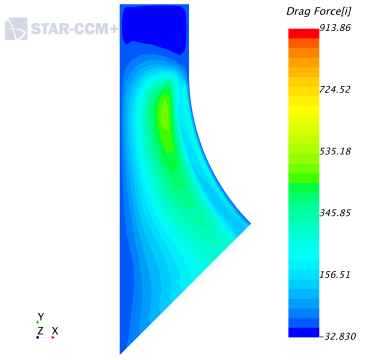
(g) $\mathbf{F}_L$ - X component



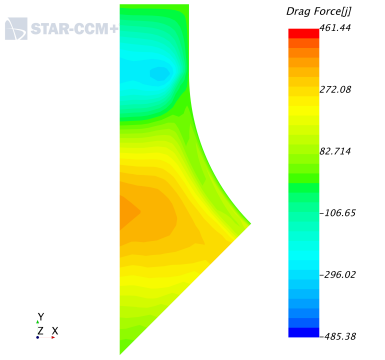
(h) $\mathbf{F}_L$ - Y component



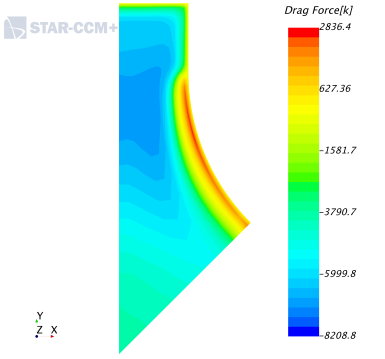
(i) $\mathbf{F}_L$ - Z component



(j) $\mathbf{F}_D$ - X component

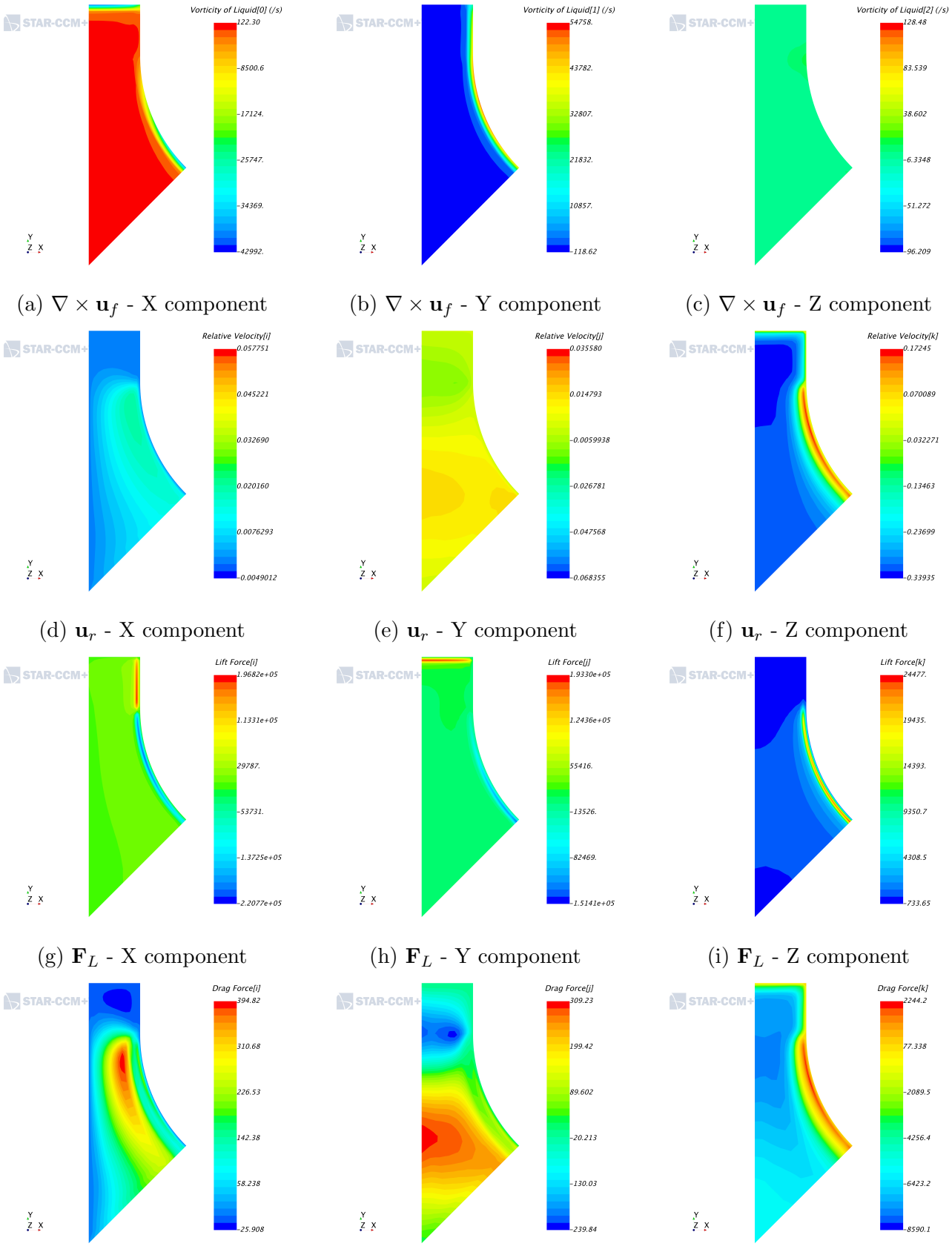


(k) $\mathbf{F}_D$ - Y component



(l) $\mathbf{F}_D$ - Z component

Figure 6.9: Lift estimation - 1.2223



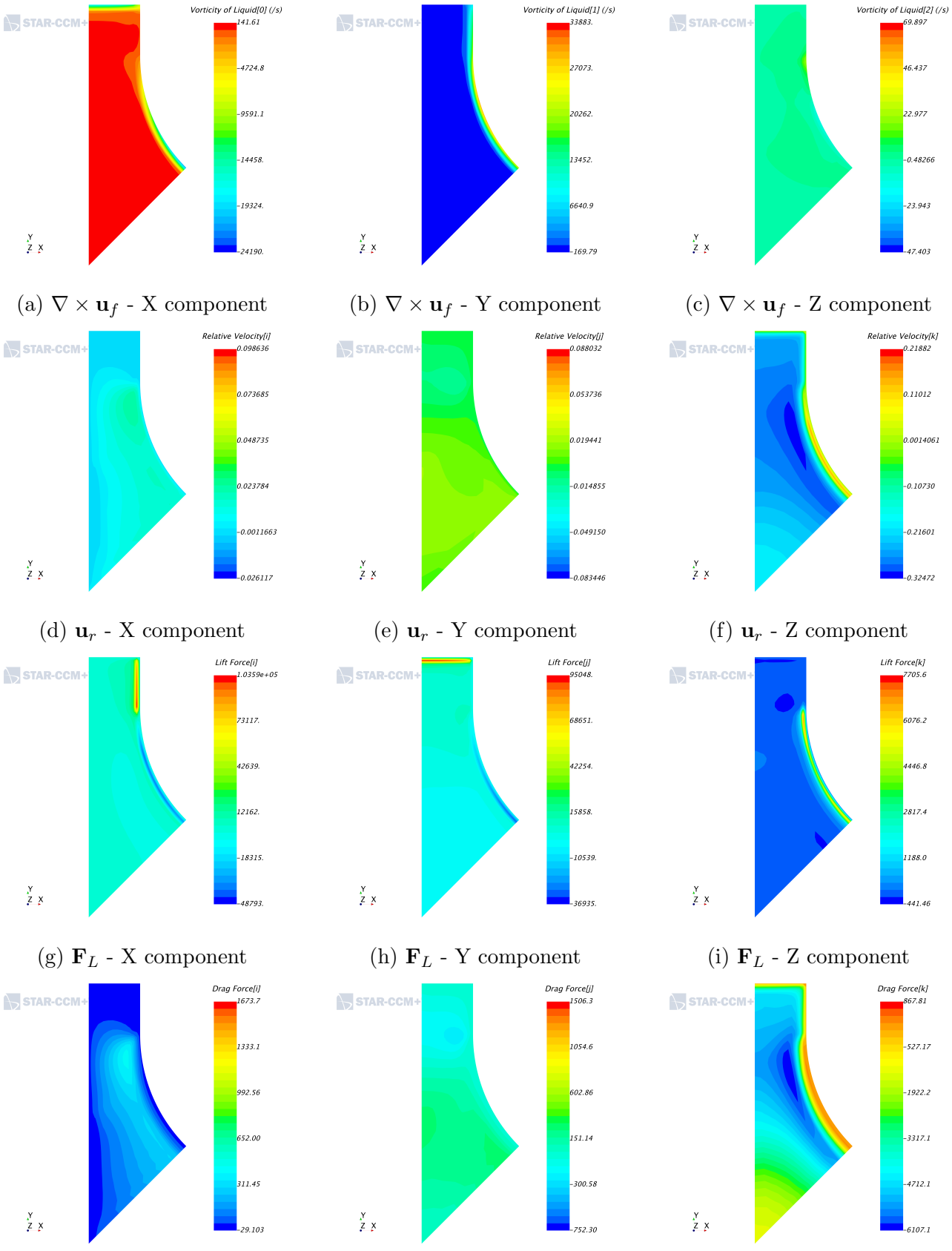
(a) $\nabla \times \mathbf{u}_f$ - X component (b) $\nabla \times \mathbf{u}_f$ - Y component (c) $\nabla \times \mathbf{u}_f$ - Z component

(d) $\mathbf{u}_r$ - X component (e) $\mathbf{u}_r$ - Y component (f) $\mathbf{u}_r$ - Z component

(g) $\mathbf{F}_L$ - X component (h) $\mathbf{F}_L$ - Y component (i) $\mathbf{F}_L$ - Z component

(j) $\mathbf{F}_D$ - X component (k) $\mathbf{F}_D$ - Y component (l) $\mathbf{F}_D$ - Z component

Figure 6.10: Lift estimation - 1.2237



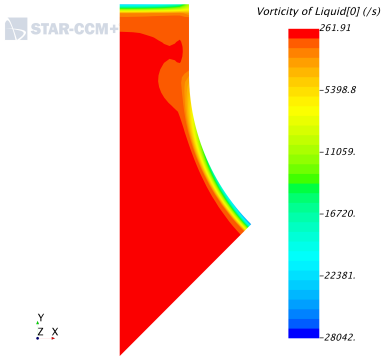
(a) $\nabla \times \mathbf{u}_f$ - X component (b) $\nabla \times \mathbf{u}_f$ - Y component (c) $\nabla \times \mathbf{u}_f$ - Z component

(d) $\mathbf{u}_r$ - X component (e) $\mathbf{u}_r$ - Y component (f) $\mathbf{u}_r$ - Z component

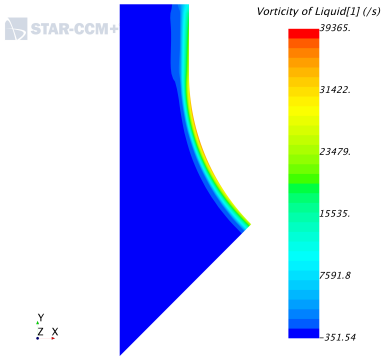
(g) $\mathbf{F}_L$ - X component (h) $\mathbf{F}_L$ - Y component (i) $\mathbf{F}_L$ - Z component

(j) $\mathbf{F}_D$ - X component (k) $\mathbf{F}_D$ - Y component (l) $\mathbf{F}_D$ - Z component

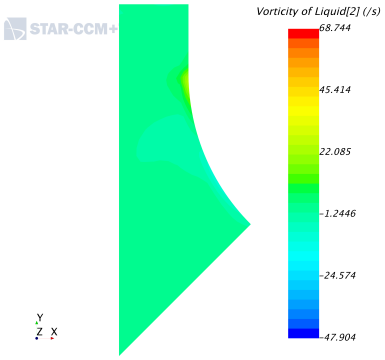
Figure 6.11: Lift estimation - 1.4325



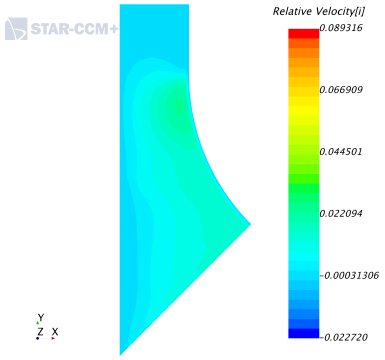
(a) $\nabla \times \mathbf{u}_f$ - X component



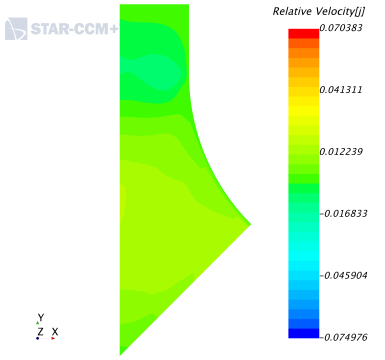
(b) $\nabla \times \mathbf{u}_f$ - Y component



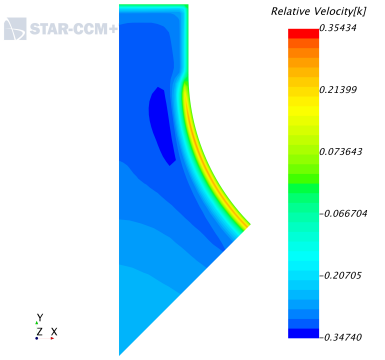
(c) $\nabla \times \mathbf{u}_f$ - Z component



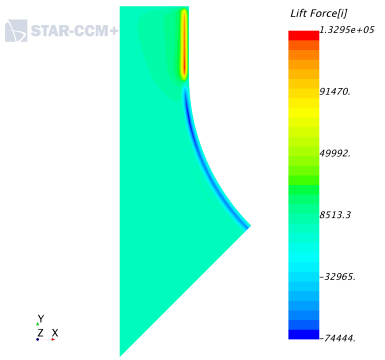
(d) $\mathbf{u}_r$ - X component



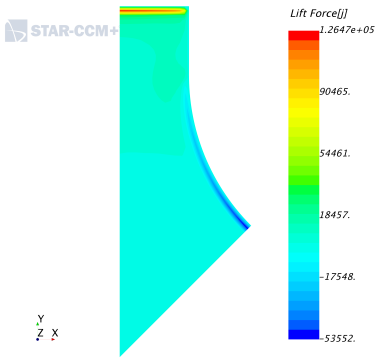
(e) $\mathbf{u}_r$ - Y component



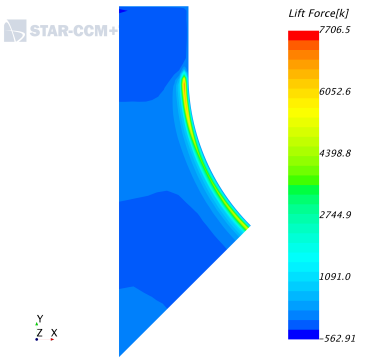
(f) $\mathbf{u}_r$ - Z component



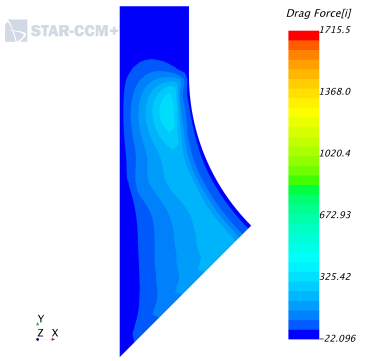
(g) $\mathbf{F}_L$ - X component



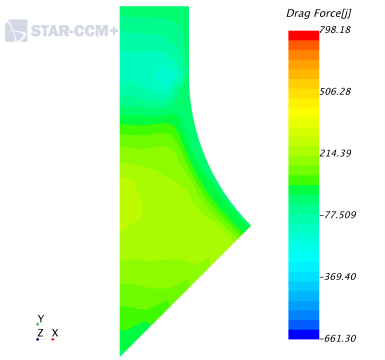
(h) $\mathbf{F}_L$ - Y component



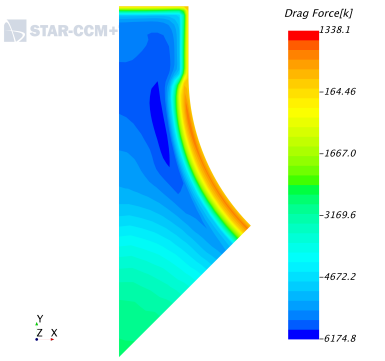
(i) $\mathbf{F}_L$ - Z component



(j) $\mathbf{F}_D$ - X component



(k) $\mathbf{F}_D$ - Y component



(l) $\mathbf{F}_D$ - Z component

Figure 6.12: Lift estimation - 1.4326

6.2.2 Coupled simulations

All the considerations on the standalone simulations are still valid for the coupled simulations.

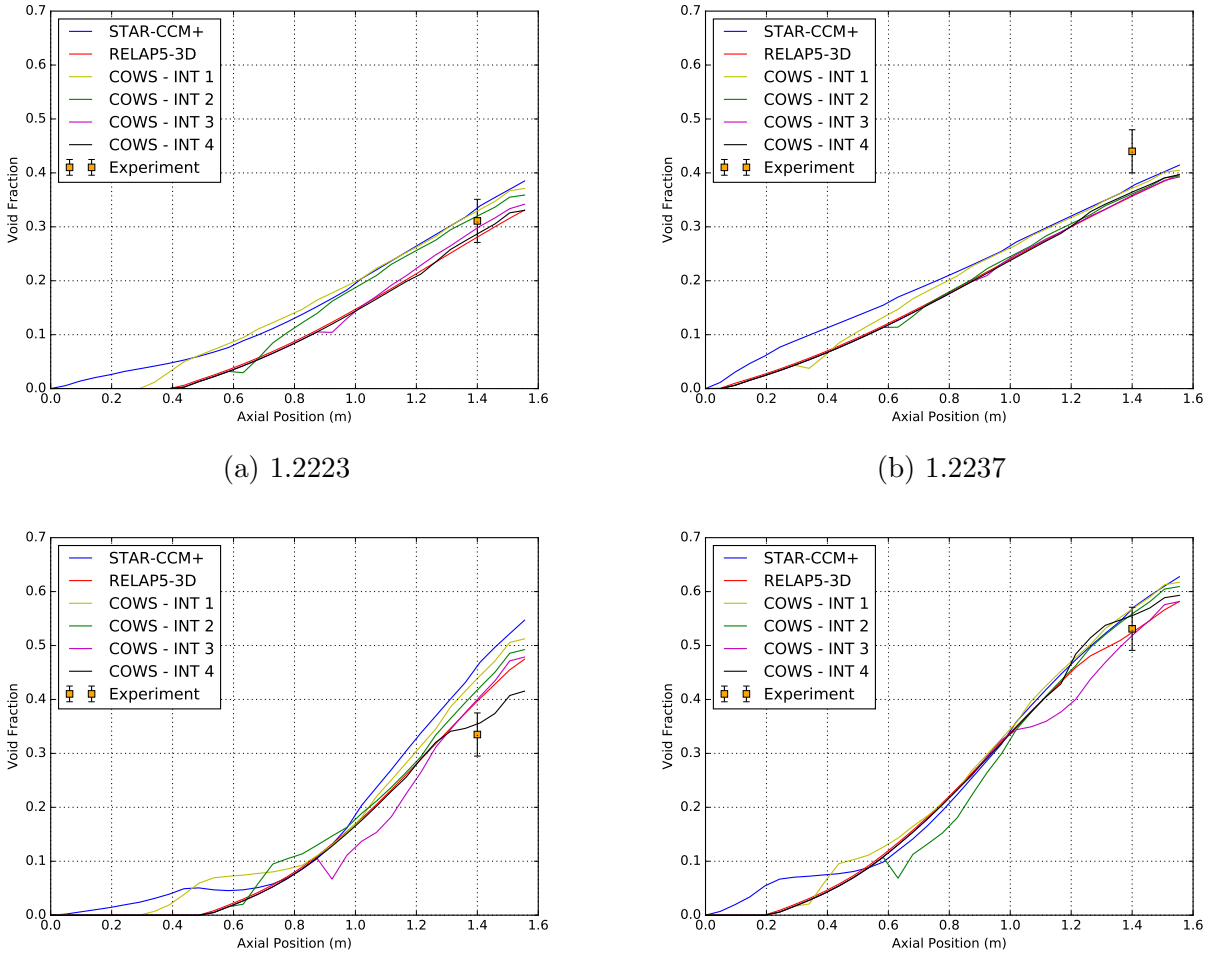
Varying the position of the coupling interface causes a change in the axial void fraction trends, as shown in fig. 6.13. The axial void fraction distribution of the various coupled simulation and the RELAP5-3D standalone case show the same behaviour until the point where the interface is located. This behaviour is both expected and desired, as the code that is modelling the channel until the coupling interface is RELAP5-3D.

The axial void fraction distribution of the various coupled simulations tends to the STAR-CCM+ behaviour when the CFD computational domain is axially long enough to allow for the dissiaption of boundary effects. This behaviour is less visible the more the interface is moved towards the outlet of the channel. Similarly, this behaviour is both expected and desired as the code that is modelling the channel after the coupling interface is STAR-CCM+.

For the coupled simulations, it is possible to notice a change in the slope of the axial void fraction distribution in the vicinity of the coupling interfaces. This is due to the change in the code, and hence the models used, that is evaluating the void fraction. The position of the coupling interfaces is indicated in table 6.2. In addition to this, not imposing the proper radial void fraction distribution is deemed to have an influence on the slope of the axial distribution at the coupling interfaces.

More tests will be needed to assess the entity of the influence of the correct radial void fraction distribution on the axial void fraction distribution when an accurate method of reconstructing the radial void fraction distribution will be available.

In addition to this behaviour it is possible to see that varying the position of the coupling interface causes a change in the value of the average void fraction (table 6.4 through 6.7).



(a) 1.2223

(b) 1.2237

(c) 1.4325

(d) 1.4326

Figure 6.13: Axial void profile comparison between coupled and standalone simulations

Fig. 6.14 through 6.17 show the radial void fraction distribution of the four coupled simulations.

Comparing the void distributions of coupled and standalone distributions, it is possible to notice a distortion of the coupled distributions and a shift of the void peak towards the heated wall. The length of the STAR-CCM+ model affects the position of the void peak. Shorter axial lengths result in larger shifts of the void peak towards the heated wall region.

The cause of this is the transition between the one-dimensional computational model provided by RELAP5-3D and the three-dimensional computational model provided by STAR-CCM+. Appropriate boundary conditions are vital to the correct modelling of an engineering problem with coupled codes. The main shortcoming of this approach is the inability of using periodic boundary conditions, or an equivalent tool, to allow for the reconstruction of the radial profile

of quantities needed to provide for coherent boundary conditions.

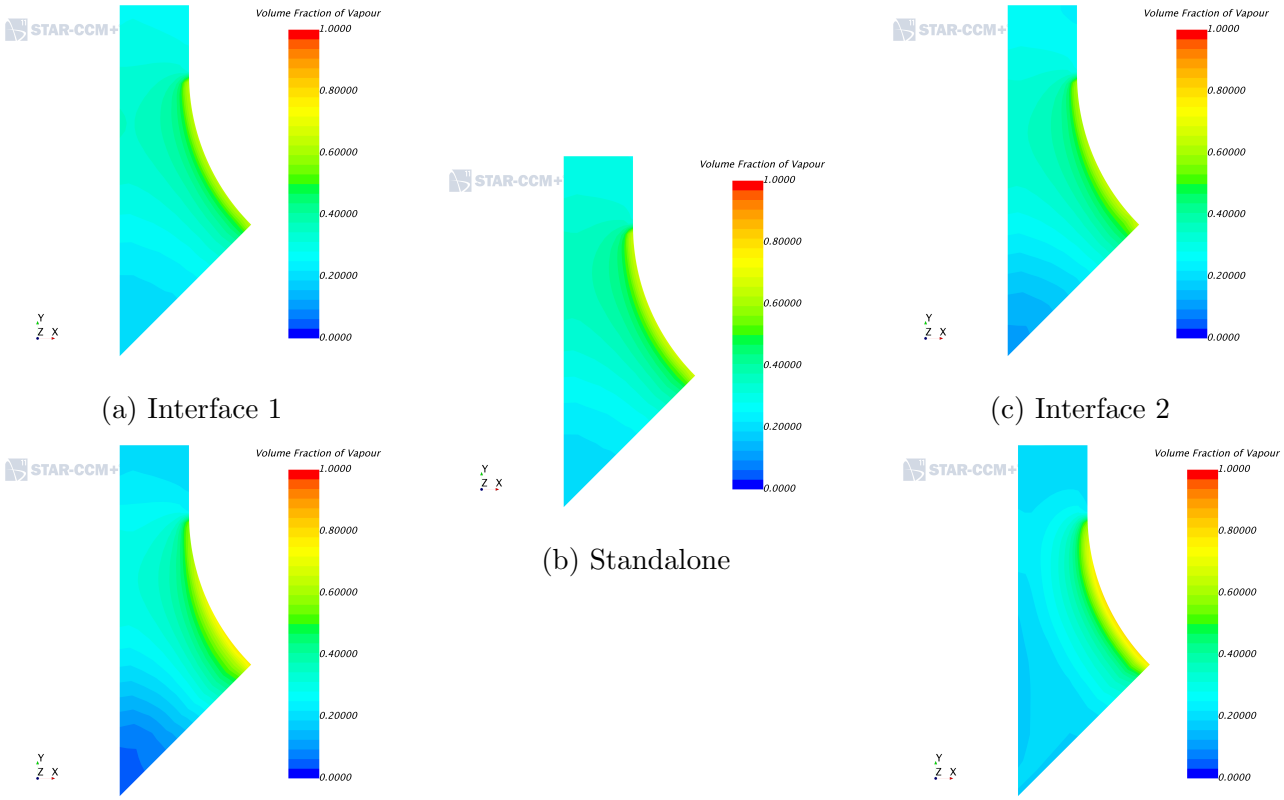
In this case, the problem that is being modelled is a turbulent two-phase flow of water and steam. Since the two phases are modelled using a two-fluid approach, the ideal boundary conditions for each phase on the CFD inlets would be as follows:

- Mass flow rate
- Temperature profile
- Void fraction distribution
- Velocity profile
- Turbulent dissipation rate profile [1]
- Turbulent kinetic energy profile [1]

One-dimensional codes like RELAP5-3D handle turbulence through the many empirical correlations they implement, hence do not have any formulation in terms of turbulent quantities normally used by CFD codes. This is a major shortcoming, since two of the six quantities needed to accurately represent a CFD inlet boundary conditions are unavailable to one dimensional codes. It can be deduced, from what has been already considered for fully developed flows in section 5.4, that not taking into consideration turbulent related quantities will lead to the imposition of wrong values for turbulent quantities in the boundary conditions, thus resulting in the flow needing an inlet length to reach its actual turbulent state.

Ideally, radial profiles for temperature, velocity and void fraction should be imposed when transitioning to the CFD domain. However, the one dimensionality of system codes only allows for averaged values of temperature, mass flow rate and void fraction. These values, although useful, are obviously a subset of an already reduced set, thus leading to the imposition of correct but incomplete boundary conditions.

[1] This quantity depends on the turbulence models chosen for the problem and on the boundary conditions for turbulence related quantities used at inlets by the CFD code



(a) Interface 1

(b) Standalone

(c) Interface 2

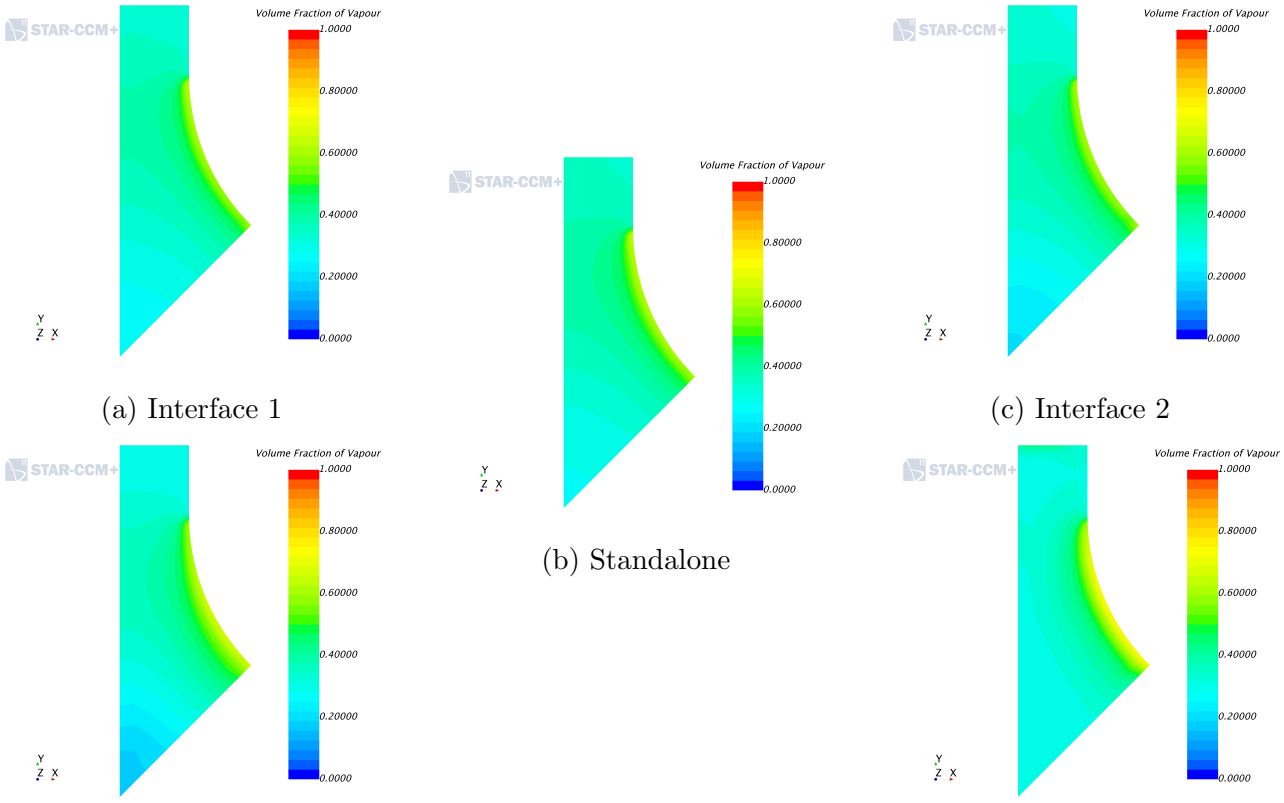
(d) Interface 3

(e) Interface 4

Figure 6.14: Void fraction distribution at measuring point, coupled simulations - 1.2223

PSBT	RELAP5-3D	STAR-CCM+	COWS
31.1 ± 4.0 %	28.40 %	34.41 %	32.23 % - INT 1
			31.33 % - INT 2
			29.22 % - INT 3
			27.94 % - INT 4

Table 6.4: Average void fraction at measuring point - 1.2223



(a) Interface 1

(b) Standalone

(c) Interface 2

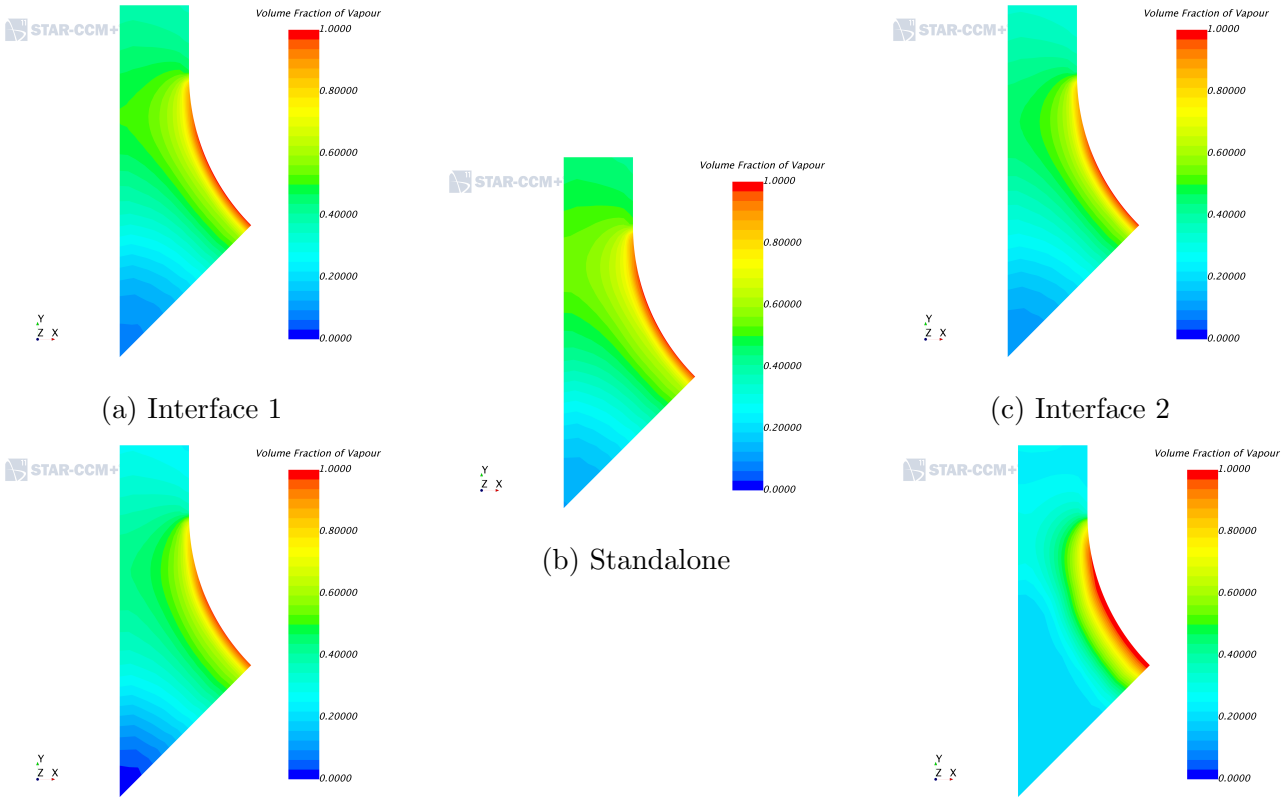
(d) Interface 3

(e) Interface 4

Figure 6.15: Void fraction distribution at measuring point, coupled simulations - 1.2237

PSBT	RELAP5-3D	STAR-CCM+	COWS
44.0 ± 4.0 %	35.92 %	38.24 %	36.57 % - INT 1
			35.53 % - INT 2
			35.39 % - INT 3
			36.04 % - INT 4

Table 6.5: Average void fraction at measuring point - 1.2237



(a) Interface 1

(b) Standalone

(c) Interface 2

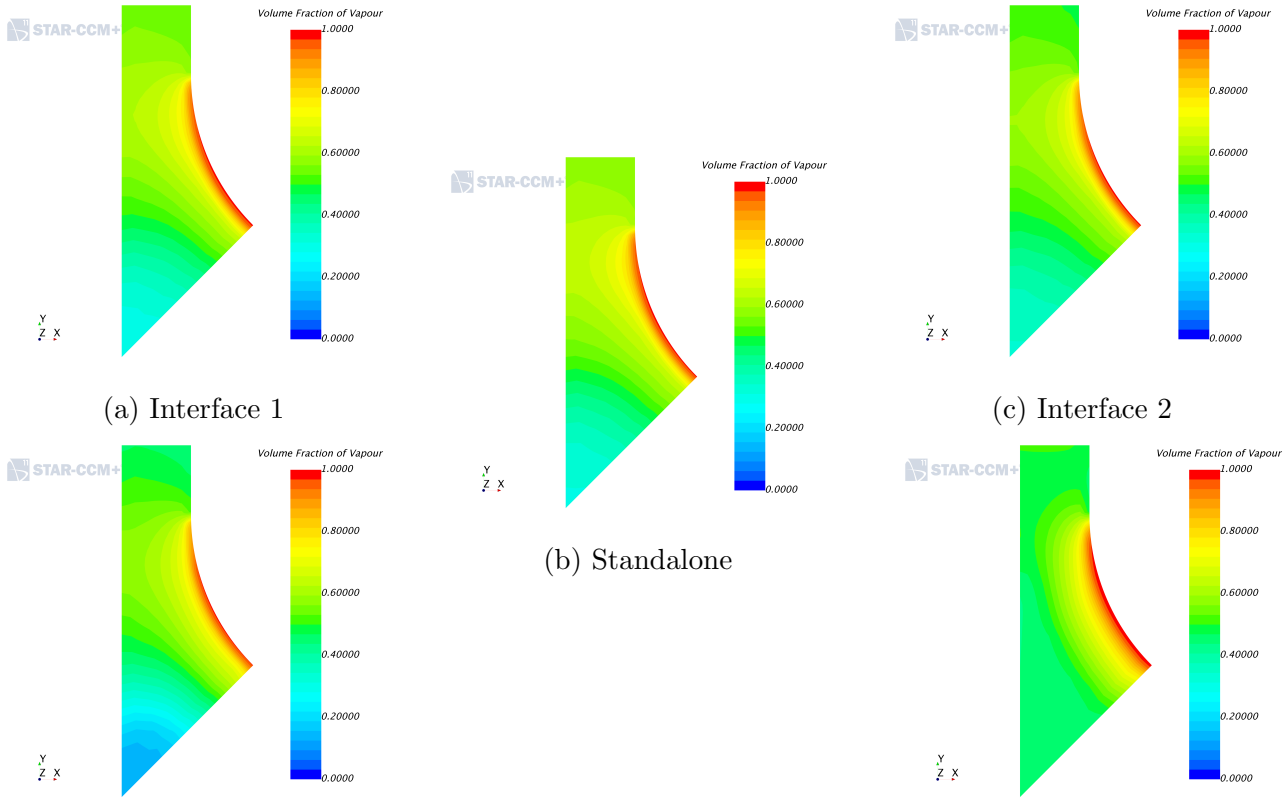
(d) Interface 3

(e) Interface 4

Figure 6.16: Void fraction distribution at measuring point, coupled simulations - 1.4325

PSBT	RELAP5-3D	STAR-CCM+	COWS
33.5 ± 4.0 %	40.20 %	47.84 %	42.53 % - INT 1
			41.36 % - INT 2
			39.59 % - INT 3
			34.90 % - INT 4

Table 6.6: Average void fraction at measuring point - 1.4325



(a) Interface 1

(b) Standalone

(c) Interface 2

(d) Interface 3

(e) Interface 4

Figure 6.17: Void fraction distribution at measuring point, coupled simulations - 1.4326

PSBT	RELAP5-3D	STAR-CCM+	COWS
53.1 ± 4.0 %	52.76 %	57.95 %	55.86 % - INT 1
			55.58 % - INT 2
			51.48 % - INT 3
			55.04 % - INT 4

Table 6.7: Average void fraction at measuring point - 1.4326

Fig. 6.14 through 6.17 show how using a reduced set of boundary condition yields distorted radial void fraction distributions when compared with the standalone results. It is possible to notice that the coupled void fraction radial distribution is closer to the standalone CFD results when the axial length of the CFD domain is larger.

Overall, the results calculated with COWS are satisfactory, but an improvement on the boundary conditions imposed at the inlet of the CFD model is required to prevent the distortion of both the and radial void distributions. A possible candidate is the technique presented in Palazzi et al. (2016), which allows the reconstruction of fully-developed flows thanks to the use of periodic boundary conditions.

The experimental measurements could be improved as well: a higher number of measuring point covering the whole channel would provide information on the axial void fraction distribution. In addition to that, information on the pressure distribution and the velocity of the phases would allow for a better validation of computational results, as pressure and velocity data was not available in the PSBT data set.

6.2.3 Computational time comparison

Simulation times for both standalone and coupled simulations are reported in table 6.8.

As expected, the times of the coupled simulations decrease with the size of the CFD model. Unexpectedly, the coupled simulation times were higher than the standalone CFD times. The sum of the standalone simulations is significantly lower than the coupled simulation. The cause is attributed to how COWS operates.

As discussed in chapter 4, RELAP5-3D's restart option is used for the coupling. By doing this, a file containing the state of the system to be restored, called restore file, has to be loaded by RELAP5-3D. INL (2014a) explains that RELAP5-3D appends information to the existing restore file at the end of the run. This implies that by using the restart option, the state of the last iteration is added to the restart file at the end of each RELAP5-3D run, thus increasing the number of information to be loaded by RELAP5-3D at every next iteration by exactly the size

PSBT case	RELAP5-3D	STAR-CCM+	COWS
1.2223	4.67	8250	15818 - INT 1
			13179 - INT 2
			10875 - INT 3
			8120 - INT 4
1.2237	5.41	8227	15815 - INT 1
			13189 - INT 2
			10873 - INT 3
			8110 - INT 4
1.4325	6.33	8269	15819 - INT 1
			13185 - INT 2
			10873 - INT 3
			8117 - INT 4
1.4326	4.13	8253	15819 - INT 1
			13185 - INT 2
			10873 - INT 3
			8120 - INT 4

Table 6.8: Computational times in seconds

of the system. At a generic iteration N, a total of $N-1$ sets of information will be present on the restart file, directly impacting its size and loading time. This relationship can be expressed with the following sum:

$$t = C_R N_R \sum_{i=1}^{N} \tau_i \rightarrow O(N^2) \tag{6.9}$$

Where:

- t: Restart file loading time (s)
- C_R: Restart file loading coefficient ($s/[cell \cdot iteration]$)
- N_R: Size of the RELAP5-3D model (cells)
- τ_i: i-th iteration
- N: Total number of iterations

This analysis showed that COWS cannot be used in its current implementation for large system simulations such as nuclear reactors, due to the large times taken by the restart file loading

operations. Further refinement of COWS is needed, with a potential candidate being RELAP5-3D's PVMEXEC, an API based on Oak Ridge National Laboratory's parallel virtual machine, that allows the user to extract data from RELAP5-3D without the need to rely on the restart option. Further analyses are needed to verify that the loading operations effectively follow equation 6.9.

6.3 Summary

Extending COWS allowed to include multiphase flows into the spectrum of problem that is possible to model with a coupled system.

Only steady state analyses were performed because the coupling algorithm implemented by COWS is explicit. As explained in Aumiller et al. (2002), explicit coupling algorithms show signs of instability when used to model transient cases.

Prior to coupled simulations, standalone analyses were performed to understand the level of performance to be expected by STAR-CCM+ and RELAP5-3D, since the performance of COWS is limited by the performance of the models of the single codes. It is important to remark that the focus of this work does not seek to improve the physics models implemented in either STAR-CCM+ or RELAP5-3D; instead it aims to give a better understanding of the interaction of these two programs when modelling complex scenarios such as multiphase flows. A discrepancy between the computational results and the experimental data of the radial void fraction distribution was identified and attributed to the lift force being neglected in the simulations.

COWS was tested with four steady state cases from the PSBT benchmark. The results from the coupled simulations indicate that a reasonable agreement with the experiments and the standalone simulations is obtained. However, both the radial and axial void distributions and the average value of the void fraction are influenced by the location of the coupling interface. An improvement in the boundary conditions for the inlet of the CFD model is needed to prevent distortions in both the axial and radial void distribution.

This analysis highlighted a major bottleneck in the current implementation of COWS. Using RELAP5-3D's restart option means that RELAP5-3D will load a restart file at each iteration, which grows at each iteration by the size of the system modelled in RELAP5-3D. The resulting added time is proportional to the square of the number of iterations. As a consequence, COWS cannot be used in its current implementation for large system simulations such as nuclear reactors, due to the large times taken by the restart file loading operations.

It was not possible to use periodic boundary conditions to reconstruct inlet fully developed profiles for velocities, temperatures and void fraction from values coming from RELAP5-3D because STAR-CCM+ does not yet allow the use of periodic boundary conditions when modelling multiphase flows.

Chapter 7

Applications

Following the validations presented in chapters 5 and 6, COWS was used to simulate a possible industrial scenario. For this, one case from the bundle section of the PSBT benchmark was used, together with a fictitious loop modelled in RELAP5-3D. The size of the model was kept small so that it could run on a desktop workstation in a reasonable time.

This model can be considered as a simplification of a PWR loop and shows the reliability of COWS in a complex case.

7.1 Model description

The PSBT benchmark was used to test the capabilities of COWS to model closed loop systems. One case from the steady state void distribution exercises for rod bundles of phase I were chosen. The exercises were chosen from the steady state group to avoid issues related to the explicit nature of the data exchange algorithm, as explained in chapters 5 and 6.

Differently from the cases presented in chapter 6, the effective heated length is 3.658 m long and the measurement sections are at at 2.216 m, 2.669 m and 3.177 m from the inlet. The geometric parameters of the bundle are listed in tab. 7.1.

Item	Data
Rods array	5x5
Rod outer diameter (mm)	9.50
Rod pitch (mm)	12.60
Flow channel inner width (mm)	64.9
Axial heated length (mm)	3658
Axial power shape	Uniform

Table 7.1: Geometric parameters for PSBT rod bundle

The test chosen was the same as the one presented in Lo and Osman (2012). The conditions for the test are listed in table 7.2.

P (bar)	T (K)	$\dot{Q}$ (kW)	$\dot{m}$ (kg/s)
164.15	590.05	2990	10.14

Table 7.2: PSBT test case 5.1121 conditions

The bundle geometry and its radial power distribution are shown in fig. 7.1. All powers shown are relative powers.

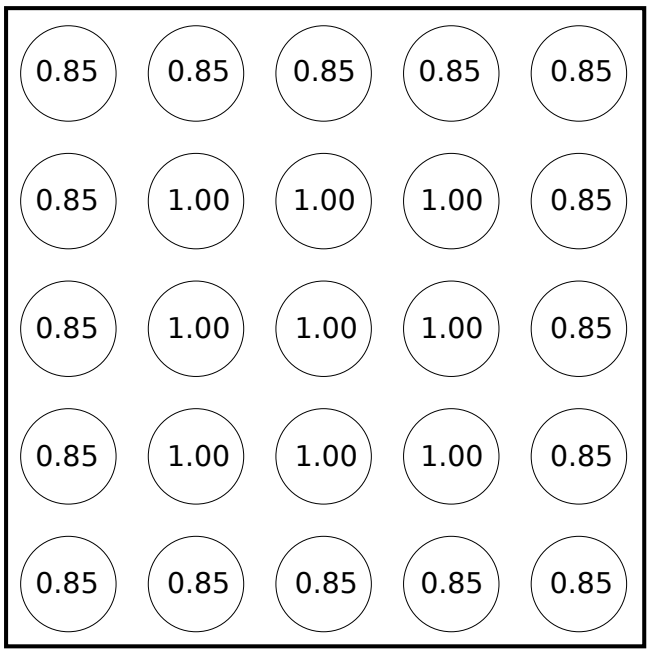
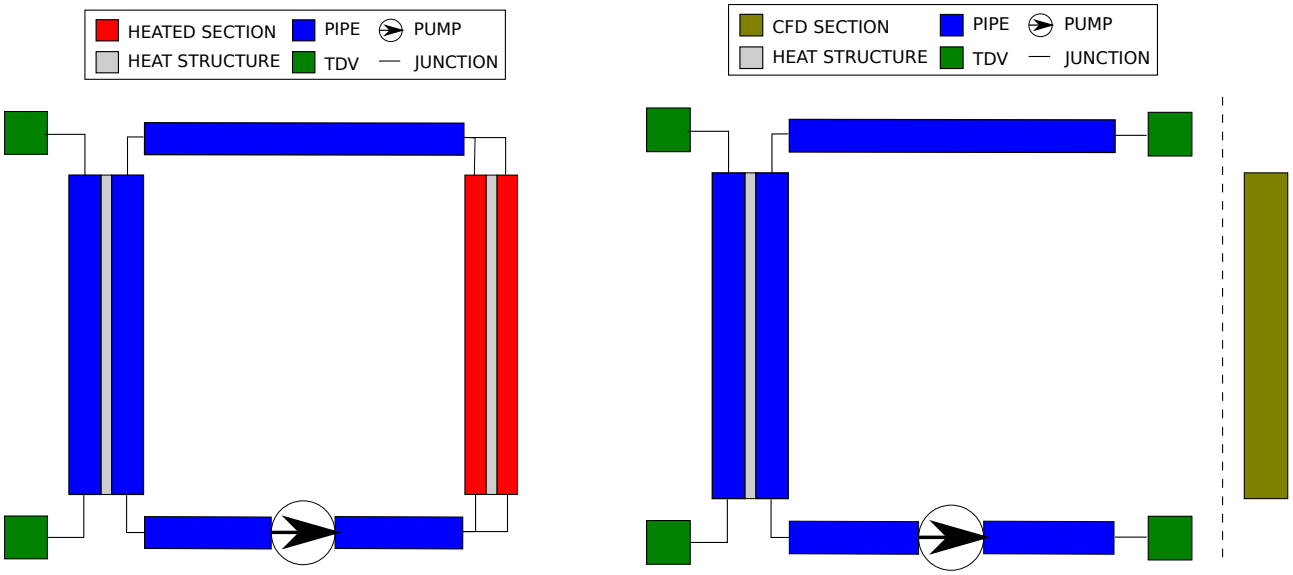


Figure 7.1: Bundle geometry and radial power distribution

Standalone STAR-CCM+ and RELAP5-3D simulations were run before attempting a coupled approach. The standalone STAR-CCM+ simulation modelled the full bundle whereas the RELAP5-3D standalone simulation modelled the full bundle and the rest of the circuit. In the coupled simulations, the rod bundle was modelled with STAR-CCM+ and the rest of the

circuit was modelled with RELAP5-3D. Fig 7.2 shows the computational model of the circuit.



(a) RELAP5-3D standalone

(b) Coupled

Figure 7.2: Computational model

The circuit modelled in 7.2 is not the actual rig used in the experiments, because the data was not available. To model a closed loop scenario, which is of relevant importance in industrial applications, a simple three-component loop has been implemented in RELAP5-3D. The three components are a hot leg (HL), a downcomer/heat exchanger (DHX), which was modelled as two concentric pipes and a cold leg (CL), where the pump is placed. The flow in the two pipes of the DHX is countercurrent.

The HL is connected to the outlet of the CFD model, thus providing a pressure outlet boundary condition. The CL is connected to the inlet of the CFD model, thus providing a mass flow inlet boundary condition. The length of HL and CL is 3.45 m each, whereas the length of DHX is 3.658 m, same as the CFD model. CL, HL and the inner pipe of the DHX all have a cross section of 0.00244 m^2. The outer pipe of the DHX has the same length as the inner pipe and a cross section of 60 m^2, where a mass flow rate of 30 ton/s was injected. The boundary conditions this circuit imposes to the CFD model are the ones of table 7.2.

In the standalone RELAP5-3D the bundle was modelled with two vertical pipes connected radially with a multiple junction component. The two pipes represent the inner and outer section of the bundle, respectively highlighted in blue and green in fig. 7.3. Three heat structures were

used: One for the pins at full power facing the inner channel (red), one for the pins at full power facing the outer channel (yellow) and a third for the fuel pins at reduced power (orange).

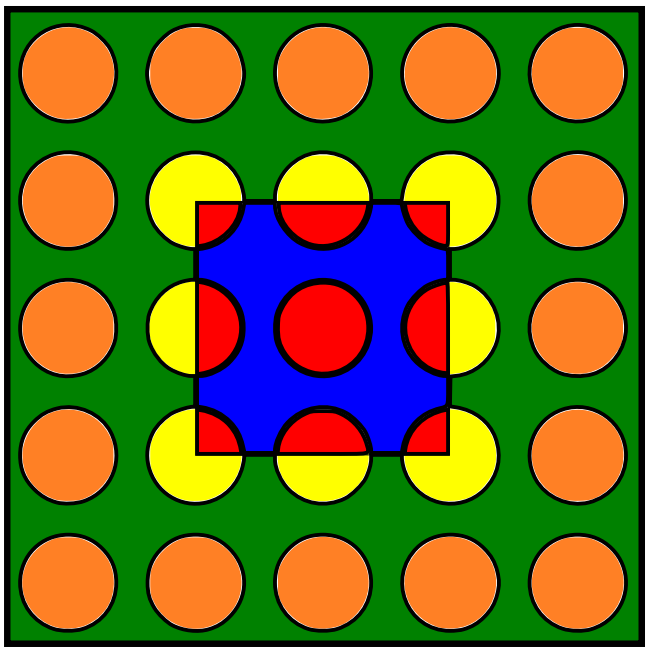


Figure 7.3: Sections of the RELAP5-3D model

Void fraction measurements in the bundle cases of the PSBT benchmark were performed by averaging the void fraction of the four central subchannels. By modelling the bundle as shown in fig. 7.3 it was possible to "isolate" the flow area of interest for the evaluation of the results.

As explained in Kim et al. (2012), to minimise the need for computational resources the symmetry of the subchannel was exploited. The size of the STAR-CCM+ model was reduced by a factor of 4 in both standalone and coupled simulations. The spacer grids were not modelled, as opposed to the work presented by Lo and Osman (2012), so that the simulation could be easily run on a 12-core workstation in reasonable time.

The resulting geometry and its related boundary types are shown in fig. 7.4. The boundary types are set as follows: the green lines are symmetry boundaries, the black lines are adiabatic walls and the red line is the heated wall. A mesh sensitivity study was performed on all simulations. The results from the coupled simulations were compared with the results of standalone simulations and experiments.

In the CFD model an Eulerian-Eulerian description with the wall boiling heat transfer model

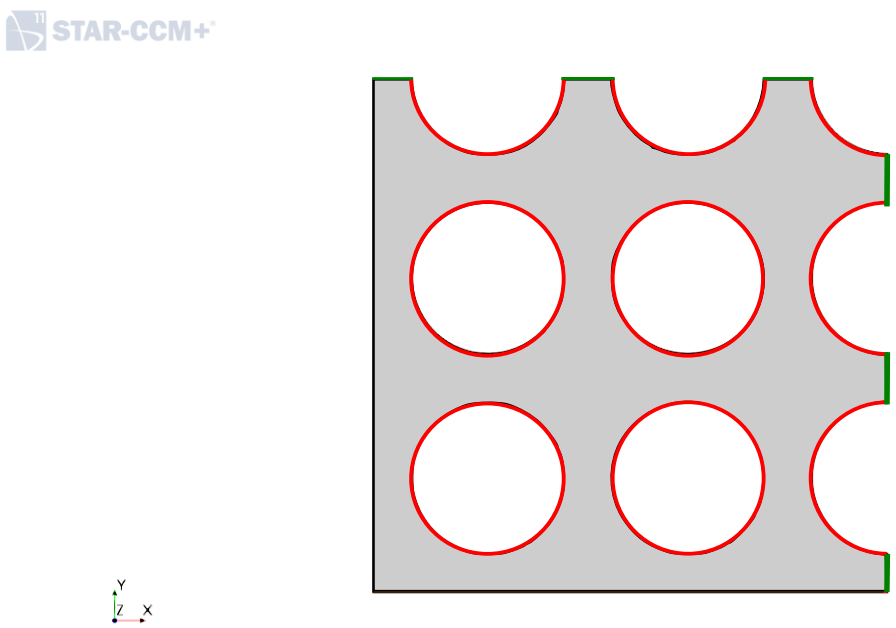


Figure 7.4: Cross section of the CFD model

(Kurul and Podowski, 1990) was used. The forces acting on the bubbles that were considered in the simulation were the turbulent dispersion force and the drag force. The lift force in the transversal direction was neglected because the simulations would not converge otherwise, as explained in section 6.2.1. All the calculations were performed keeping the physical properties constant at the saturation state corresponding to the operating pressure. The realisable $k - \varepsilon$ turbulence model with STAR-CCM+'s default wall treatment options was used.

In the RELAP5-3D model only the compulsory base thermal hydraulic modules were activated.

7.2 Results

The quality of the results of coupling tools is clearly bounded by the quality of the results of their components. For this reason, the purpose of running standalone prior to coupled simulations was to gather information on the results that can be expected from the coupled simulations.

7.2.1 Standalone simulations

Fig. 7.5a through 7.5c compare the void distribution between experimental and CFD results at the measuring point. It is not possible to perform quantitative comparisons since only graphical data was available for the radial void fraction distribution. The flow distributions in fig. 7.5 are symmetrical, as opposed to the ones presented by Lo and Osman (2012). This is due to the absence of spacer grids in the CFD model and justifies some discrepancies with the CFD results presented here.

Table 7.3 shows the comparison between the standalone simulations and the experimental data. Overall, results are satisfactory. The void distributions are reproduced reasonably well in all cases. The discrepancies between some experimental results and the CFD simulations were also observed in the results presented in Lo and Osman (2012). In chapter 6 it is explained how these differences can be attributed to a combination of model choice and model performance.

Test 5.1121	**Lower**	**Middle**	**Upper**
Experiment	0.00 ± 4.0 %	0.00 ± 4.0 %	17.91 ± 4.0 %
Lo and Osman	5.04 %	9.07 %	15.76 %
STAR-CCM+	2.11 %	6.23 %	13.50 %
RELAP5-3D	0.23 %	6.83 %	18.03 %

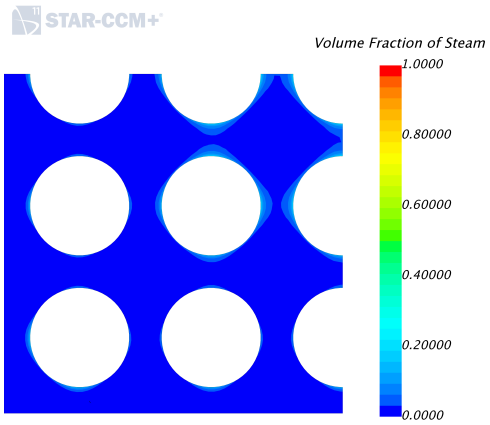
Table 7.3: Central four channels average void fraction at measuring points - Standalone simulations

It is important to remark that the experimental data were taken by averaging the void fraction over the four central cross sections of the 5x5 bundle, not the whole channel. Hence, discrepancies in the values do not necessarily reflect inaccuracies in the computational models.

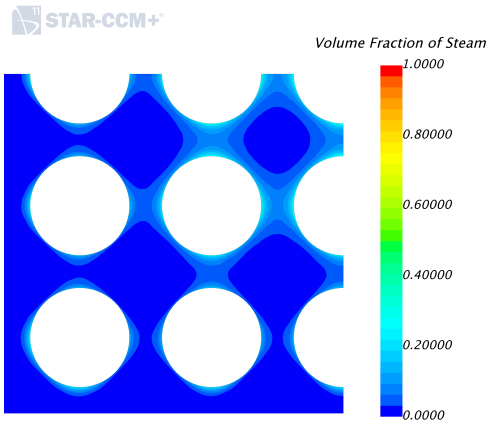
7.2.2 Coupled simulations

All the considerations on the standalone simulations are still valid for the coupled simulations. Fig. 7.6 through 7.8 show the radial void fraction distribution at the three measuring point for both standalone and coupled simulations.

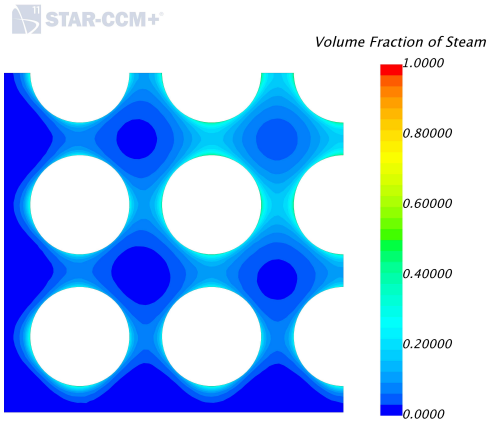
Comparing the void distributions of coupled and standalone CFD distributions, it is possible



(a) Lower elevation



(b) Middle elevation



(c) Upper elevation

Figure 7.5: Void fraction distribution at the measuring locations

to notice that they are similar in shape, with the coupled simulations predicting a higher void fraction at all the measuring points. Table 7.4 shows that the computational results match reasonably well with the experiments.

The main discrepancies are observed in the middle measuring point. All codes yield results outside of the experimental uncertainty. The length of the channel between the first and the third measuring point is $0.961m$, which is approximately 25% of the full length of the bundle. According to the experimental data the phase change starts after the second measuring point, which means that the experimental apparatus covers a portion of the bundle that is only about 13% of the full length.

Ideally, a higher number of measuring point covering the whole bundle would provide a better insight on the axial void fraction distribution. Furthermore, no information on the pressure distribution and the velocity of the phases was available. Three measuring points and the corresponding values of the average void fraction are not sufficient to thoroughly assess the quality of the computational results for phenomena as complex as multiphase flows.

At a first glance at table 7.4, RELAP5-3D seems to have the best performance. However, this analysis focused on average values, because there were no data available to compare the performance of the CFD codes to calculate the void fraction radial distribution.

RELAP5-3D yields a better average value when compared to the other codes, but is not able to provide information about the radial void fraction distribution. As a consequence, in a scenario where information on the location of a possible hot patch is needed (e.g. designing a new type of fuel assembly) a code like RELAP5-3D will not be as useful as a CFD code.

In both standalone and coupled simulation the spacer grids were not modelled. The presence of spacer grids in the experimental setup caused the void fraction distribution to be asymmetric, as mentioned in Rubin et al. (2010) and Lo and Osman (2012). Accurate reconstruction of the void distribution inside the rod bundle is certainly desired, however the experimental data available consisted only on values averaged over the central four subchannel.

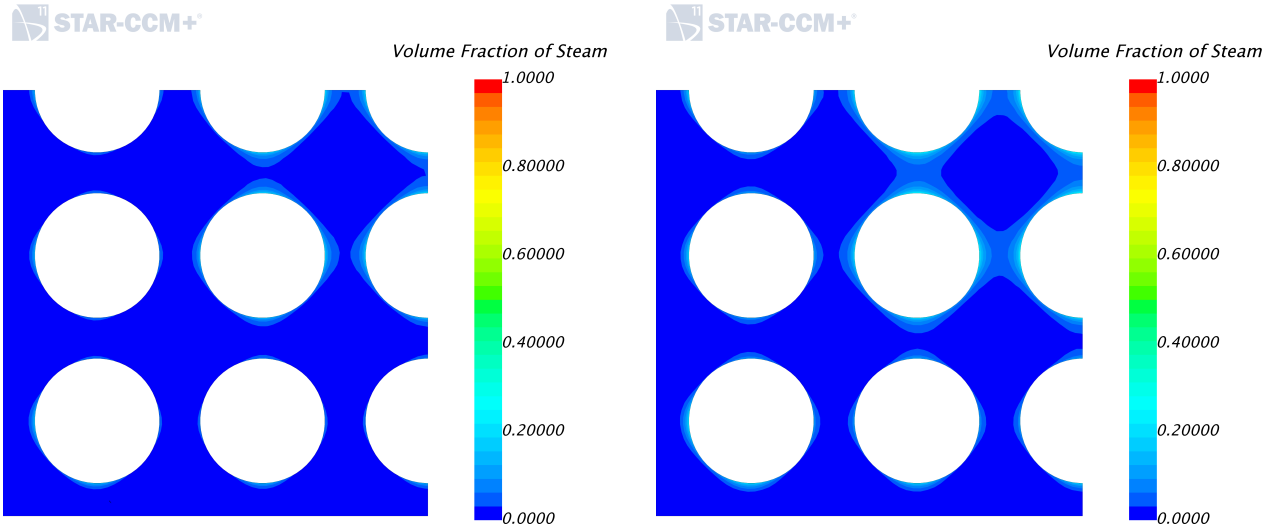
Since no other experimental data was available apart from the subchannel average void fraction, the discrepancies in the results cannot be attributed solely to deficiencies in the computational models. Spacer grids influence the bundle void fraction distribution. In addition, the total cross-flow of steam within the averaging area has to be zero if the four central subchannels are used in the averaging of the void fraction. Only under these hypotheses the average of the void fraction over the four central subchannel would be reliable.

For this reason an average of the void fraction over the full bundle, instead of the four central subchannels, would be a more suitable choice to assess code performance accurately.

Based on the previous considerations, the results are deemed acceptable.

Test 5.1121	Lower	Middle	Upper
Experiment	0.00 ± 4.0 %	0.00 ± 4.0 %	17.91 ± 4.0 %
Lo and Osman	5.04 %	9.07 %	15.76 %
STAR-CCM+	2.11 %	6.23 %	13.50 %
RELAP5-3D	0.23 %	6.83 %	18.03 %
COWS - Coupled simulations	3.84 %	9.00 %	17.05 %

Table 7.4: Central four channels average void fraction at measuring points



(a) Standalone (b) Coupled

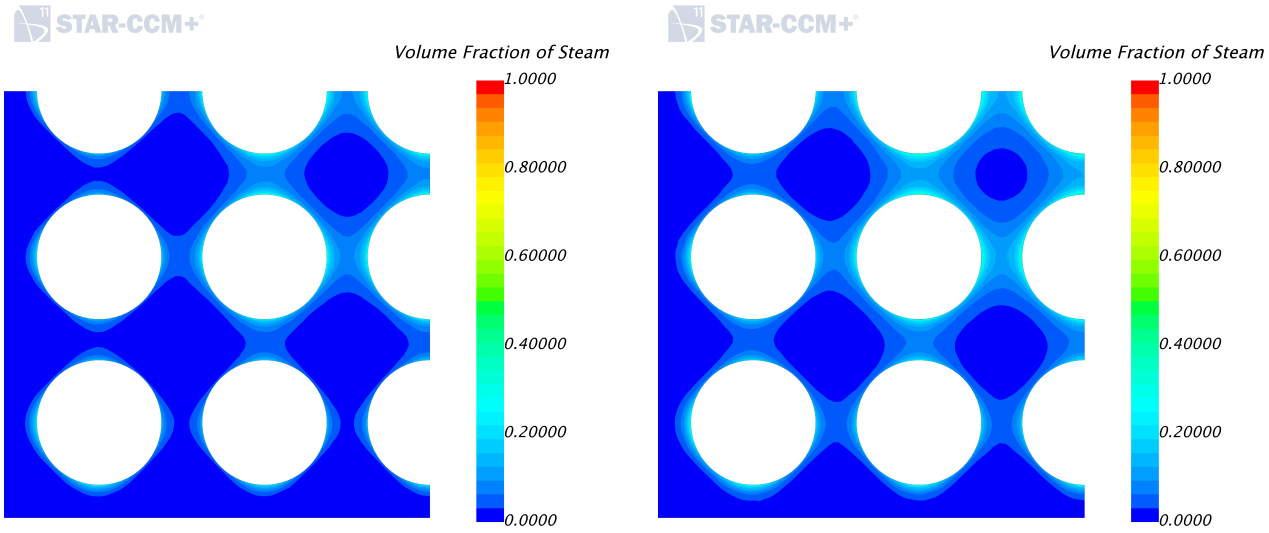
Figure 7.6: Void fraction distribution at lower measuring point

7.3 Summary

COWS was applied to one case from the rod bundle problem set of the PSBT benchmark.

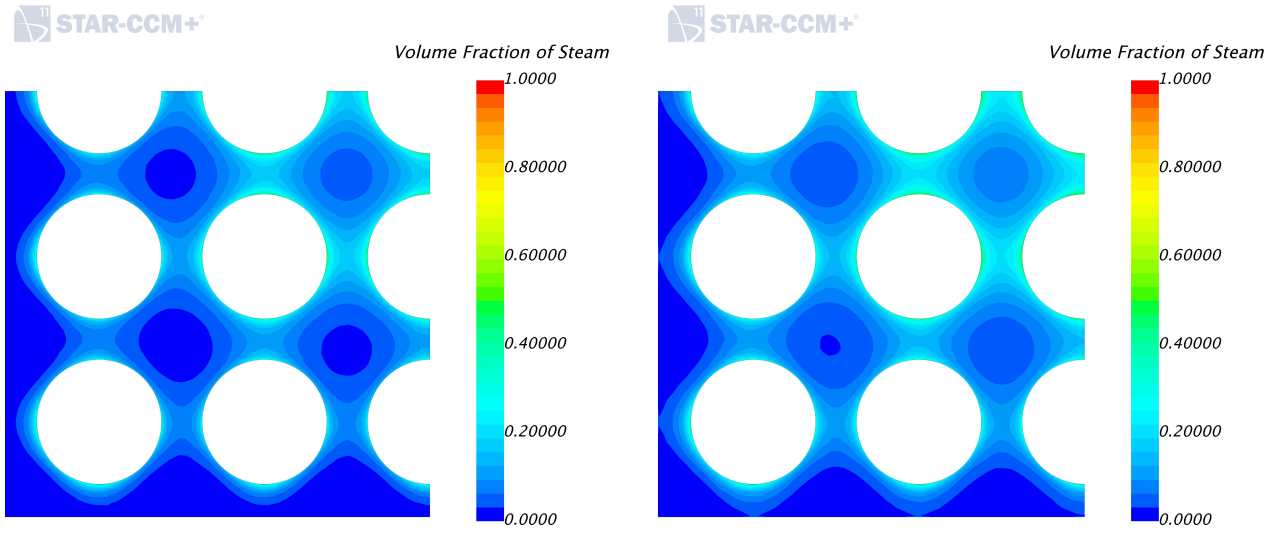
For this case as well, standalone analyses were performed to understand the level of performance to be expected by STAR-CCM+ and RELAP5-3D. Again, it is important to remark that the focus of this work does not seek to improve the physics models implemented in either STAR-CCM+ or RELAP5-3D; instead it aims to give a better understanding of the interaction of these two programs when modelling complex scenarios such as multiphase flows.

The results from the coupled simulations indicate that a reasonable agreement with both the experiments and the standalone simulations is obtained. An improvement in the coupling algorithm implementation is needed to allow the code to run transient cases accurately.



(a) Standalone (b) Coupled

Figure 7.7: Void fraction distribution at middle measuring point



(a) Standalone (b) Coupled

Figure 7.8: Void fraction distribution at upper measuring point

It was not possible to use periodic boundary conditions to reconstruct inlet fully developed profiles for velocities, temperatures and void fraction from values coming from RELAP5-3D because STAR-CCM+ does not yet allow the use of periodic boundary conditions when modelling multiphase flows.

Chapter 8

Conclusions

8.1 Summary

The thesis documents an object oriented coupling scheme (COWS) for RELAP5-3D and STAR-CCM+ relying on the Co-Simulation API implemented in STAR-CCM+. COWS yields good results in transient simulations of single phase flows and in steady simulations of both single and multiphase flows. COWS was tested against standalone RELAP5-3D simulations for transient single phase flows, Moody's chart for steady single phase flows and the PSBT benchmark for steady multiphase flows. A solution to the problem of reconstructing two-dimensional inlet flow profiles from the one-dimensional values extracted from the system code was proposed. COWS yields good results in all the cases but it has reached the upper bounds of performance given its implementation of an explicit data exchange algorithm.

8.2 Summary of thesis achievements

COWS is a generic coupling interface implementing an explicit data exchange paradigm between STAR-CCM+ and system codes. COWS was developed in C++ following an OO paradigm because the code resulting from this approach is easier to maintain, extend and modify.

STAR-CCM+'s existing coupling tool for RELAP5-3D is able to perform only a limited set of simulations, namely those cases where only the inlet of the CFD model is coupled to RELAP5-3D. The initial development of COWS focused on eliminating these shortcomings by creating a new program, thus allowing coupled simulations of pipe sequences or loops for single phase flows. Both steady and transient calculations were performed as validation.

COWS yields good results in both transient and steady state cases albeit implementing an explicit data exchange paradigm. When transients do not present singularities or abrupt changes, the coupled system is able to follow them well, in spite of the lags due to the explicit data exchange paradigm.

Periodic boundary conditions were used to reconstruct single-phase two-dimensional fully developed profiles from the one-dimensional quantities sent from RELAP5-3D.

COWS was then extended to allow for the simulation of multiphase flows. The extension consisted in exchanging temperature, mass flow rate and void fraction of the gas phase in addition to the existing variables for the single phase flows, namely temperature and mass flow rate of the liquid phase and pressure. It was explained how an explicit data exchange algorithm would be unsuitable for transient multiphase flow simulations thus leading to the choice of four cases from the steady state section of the PSBT benchmark. STAR-CCM+ does not allow the use of periodic boundary conditions for multiphase simulations, hence a study on the effect of the location of the coupling interface on the void distribution in the channel was performed.

Finally, COWS was used to simulate a possible industrial scenario. One case from the rod bundle section of the PSBT benchmark was used, together with a fictitious loop modelled in RELAP5-3D. This model can be considered as a simplification of a PWR loop and shows the reliability of COWS in a complex case.

For the multiphase flow simulations, standalone analyses were performed to understand the level of performance to be expected by STAR-CCM+ and RELAP5-3D, as the performance of COWS is bound by the performance of its components. This work does not seek to improve the physics models implemented in either STAR-CCM+ or RELAP5-3D; instead it aims to give a

better understanding of the interaction of these two programs in complex scenarios.

The results from the coupled multiphase simulations show a reasonable agreement with the experiments and the standalone simulations. However, both the radial and axial void distributions and the average value of the void fraction are influenced by the location of the coupling interface. An improvement in the boundary conditions for the inlet of the CFD model is needed to prevent distortions in both the axial and radial void distribution.

8.3 Future Work

COWS is pushed to its limits when used to model steady multiphase flows in forced convection:

- Problems involving rapid transients are more demanding than steady or slow transient scenarios and impractical to solve with an explicit algorithm (Aumiller et al., 2001)
- When the RELAP5-3D model is sufficiently large or the timestep is sufficiently small, the files produced by each RELAP5-3D run are significantly large in size, thus making the load operations of those files the actual bottleneck in the coupling process
- Use of RELAP5-3D in restart mode might cause numerical oscillations (Mesina and Anderson, 2014)
- Location of the coupling interface influences radial void fraction distribution
- Only thermal hydraulic coupling is available
- Reconstruction of flow profiles at inlet boundaries only available for single-phase flows

CD-adapco is working on extending the Co-Simulation API so that it would allow multiple data exchanges per timestep. This, together with the flexibility provided by RELAP5-3D's restart option, would allow to implement a semi-implicit coupling algorithm in COWS, thus expanding the range of problems that would be possible to solve with a coupled approach.

A feature that is not currently supported in STAR-CCM+ is the use of periodic boundary interfaces for multiphase flows. At this stage, the most efficient approach would consist of running different cases in order to generate a database of inlet profiles to be used in future simulations. The benefits of this approach are the same discussed in sections 5.4 and 4.3.1.

File I/O operations are time-intensive. Using the RELAP5-3D PVMEXEC API instead of files would allow to use TCP socket connections, thus removing the need to update files every timestep. Furthermore, RELAP5-3D would not be run in restart mode, which requires files containing data for the whole system to be loaded at the beginning of each timestep. Assessment of the performance gained by switching from file-based data exchange to the use of the PVMEXEC API would need an evaluation of the time elapsed by the loading operations. During the course of this research project, which involved models that were not as complex as a full nuclear reactor, data files of the order of gigabytes were not uncommon, hence for large simulation cases this will be a major bottleneck.

Further features could be added to extend the range of the problems where COWS can be applied.

At this stage COWS supports only thermal hydraulic coupling. However, the physics of a nuclear reactor is rather complicated and it is not possible to describe it accurately by modelling solely the thermal hydraulics of the reactor. Codes like Serpent are used to model the neutronics of a nuclear reactor and some work on coupling with CFD codes has been performed thanks to Serpent's multi-physics interface (Tuominen et al., 2016).

COWS could be used to perform three-way coupling between RELAP5-3D, STAR-CCM+ and Serpent, or, more generally, multi-way coupling between any code, thus combining the individual strengths of the specialised programs into a very powerful tool that could be used to enable to simulate a full reactor with the maximum level of detail possible and optimise the reactor not only at a component scale, but also at a system scale. This is the idea behind the "*virtual reactor*" concept presented by (Baglietto, 2012): "a new generation of analysis capability that can provide better insight into how to improve designs" which is CASL's main purpose. Since COWS is built on STAR-CCM+'s Co-Simulation API and STAR-CCM+ is CASL's CFD code

of choice, COWS could be a blueprint for a tool with a potential beyond simple university research. Among other multiphysics frameworks dedicated to the simulation of nuclear reactors where COWS could be used there is also INL's MOOSE.

Bibliography

Anderson, N., Hassan, Y., Schultz, R., 2008. Analysis of the hot gas flow in the outlet plenum of the very high temperature reactor using coupled RELAP5-3D system code and a CFD code. Nuclear Engineering ad Design 238, 274 – 279.

Anglart, H., 2011. Applied reactor technology.

Aumiller, D., Tomlinson, E., Bauer, R., 2001. A coupled RELAP5-3D/CFD methodology with a proof-of-principle calculation. Nuclear Engineering and Design 205, 83 – 90.

Aumiller, D., Tomlinson, E., Weaver, W., 2002. An integrated RELAP5-3D and multiphase CFD code system utilizing a semi-implicit coupling technique. Nuclear Engineering and Design 216, 77 – 87.

Baglietto, E., 2012. New perspectives for nuclear reactor design. ATW - International Journal for Nuclear Power 57.

Baviere, R., Tauveron, N., Perdu, F., Garre, E., 2013. SYSTEM-CFD coupled simulation of the Phénix reactor natural circulation test, in: Proceedings of NURETH 15, American Nuclear Society.

Bertolotto, D., 2011. Coupling a system code with computational fluid dynamics for the simulation of complex coolant reactivity effects. Ph.D. thesis. EPF Lausanne.

Bertolotto, D., Manera, A., Frey, S., Prasser, H., Chawla, R., 2009. Single phase mixing studies by means of directly coupledCFD/system-code tool. Annals of Nuclear Energy 36, 310–316.

CD-adapco, 2015. STAR-CCM+ 10.04 User Guide. CD-adapco.

CD-adapco, 2016. STAR-CCM+ 11.02 User Guide. CD-adapco.

Cebeci, T., Bradshaw, P., 1977. Momentum transfer in boundary layers. McGraw-Hill.

Gamma, E., Helm, R., Johnson, R., Vlissides, J., 2014. Design patterns: elements of reusable object-oriented software. Addison-Wesley.

Genic, S., Kolendic, P., Jaric, M., Budimir, N., 2011. A review of explicit approximations of Colebrook's equation. FME Transactions 39, 67 – 71.

Hori, K., 1993. In bundle void fraction measurement of PWR fuel assembly, in: Proceedings of ICONE 2, San Francisco, CA.

Hyams, D., 2012. Sample to contribute to mpl gallery - `http://comments.gmane.org/gmane.comp.python.matplotlib.general/30254`. Python source code accessible online on SourceForge.

Idel'chik, I., 1966. Handbook of hydraulic resistence. Israel program for scientific translations.

Iishi, M., Hibiki, T., 2011. Thermo-fluid dynamics of two-phase flow. Springer.

INL, 2014a. RELAP5-3D User Manual, Vol. 1. Idaho National Laboratory.

INL, 2014b. RELAP5-3D User Manual, Vol. 2 appendix A. Idaho National Laboratory.

INL, 2014c. RELAP5-3D User Manual, Vol. 5. Idaho National Laboratory.

Jeltsov, M., Koop, K., Kudinov, P., Villanueva, W., 2013. Development of a domain overlapping coupling methodology for STH/CFD analysis of heavy metal liquid thermal hydraulics, in: Proceedings of NURETH 15, American Nuclear Society.

Jimack, P., 2007. Application of multigrid techniques in CFD. International journal for numerical methods in fluids 2007, 1–12.

Kim, T., Petrov, V., Manera, A., Lo, S., 2012. Analysis of void fraction distribution and departure from nucleate boiling in single subchannel and bundle geometries using subchannel, system and CFD codes. Science and Technology of Nuclear Installation 2012.

Kurul, N., Podowski, M., 1990. Multi-dimensional effects in forced convection subcooled boiling, in: Proceedings of the 9th heat transfer conference, Jerusalem, Israel.

Lahey, R., 1978. A mechanistic subcooled boiling model, in: Proceedings of the 6th heat transfer conference, Toronto, Canada.

Lee, H., Mousseau, K., 2010. Nuclear energy CFD application management system, in: CFD for nuclear reactor safety applications, Workshop proceedings. CFD4NRS-3.

Liles, D.R., Reed, W.H., 1978. A semi-implicit method for two-phase flow dynamics. Journal of Computational Physics 26, 390–407.

Lo, S., Osman, J., 2012. CFD modeling of boiling flow in PSBT 5x5 bundle. Science and Technology of Nuclear Installations 2012.

Mahaffy, J.H., 1982. A stability-enhancing two-step method for fluid flow calculations. Journal of Computational Physics 46, 329–341.

Mesina, G., Anderson, N., 2014. Restart and backup improvements.

Meyers, S., 2014. Effective C++ - 55 specific ways to improve your programs and designs. Addison-Wesley.

NEA, 2007. Best practice guidelines for the use of CFD in nuclear reactor safety applications.

Neykov, B., Aydogan, F., Hochreiter, L., Ivanov, K., Utsuno, H., Kasahara, F., Sartori, E., Martin, M., 2006. NUPEC BWR full-size fine-mesh bundle test (BFBT) benchmark, volume 1: specifications. OECD/NEA.

NUPEC, 1989. Proving test on the reliability for nuclear fuel assemblies.

Nystrom, R., 2014. Game programming patterns. Genever Benning.

Palazzi, A., Bluck, M., Lo, S., 2014. A coupled RELAP5-3D/STAR-CCM+ simulation for the calculation of friction factor in pipes, in: Proceedings of ICAPP 2014, American Nuclear Society, Charlotte, NC.

Palazzi, A., Bluck, M., Lo, S., Slijepčević, S., 2016. Coupling RELAP5-3D and STAR-CCM+ for simulations of steady and transient single phase flow, in: Proceedings of ICAPP 2016, American Nuclear Society, San Francisco, CA.

Patankar, S., 1980. Numerical heat transfer and fluid flow. Hemisphere publishing corporation, T& F group.

Pope, S., 2010. Turbulent flows. Cambridge University Press.

Popov, E., Yan, J., Karoutas, Z., Gehin, J., Brewster, R., Baglietto, E., 2012. PWR internal flow modeling with fuel assemblies, in: Proceedings of ICAPP 2012, American Nuclear Society, Chicago, IL.

Prosperetti, A., Tryggvason, G., 2009. Computational methods for multiphase flow. Cambridge University Press.

Rodriguez, O., 2012. RELAP5-3D thermal hydraulics computer program analysis coupled with DAKOTA and STAR-CCM+ codes. Master's thesis. Texas A & M University.

Rubin, A., Avramova, M., 2011. Comparative analysis of submitted final results for excercise I-1, in: Proceedings of the second PSBT workshop, Stockholm, Sweden.

Rubin, A., Schoedel, A., Avramova, M., Utsuno, H., Bajorek, S., Velasquez-Lozada, A., 2010. OECD/NRC benchmark based on NUPEC PWR subchannel and bundle test (PSBT), volume 1: experimental database and final problem specifications. NEA/NSC/DOC.

Saha, P., Zuber, N., 1974. Point of net vapor generation and vapor void fraction in subcooled boiling, in: Proceedings of the 5th heat transfer conference, Tokyo, Japan.

Stroustrup, B., 2013. The C++ programming language. 4 ed., Pearson education.

Tong, L.S., Tang, Y.S., 1997. Boiling heat transfer and two-phase flow. Taylor & Francis.

Tuominen, R., Valtavirta, V., Peltola, J., Leppänen, J., 2016. Coupling Serpent and OpenFOAM for neutronics-CFD multi-physics calculations, in: Proceedints of PHYSOR, Sun Valley, ID.

USNRC, 2007. Standard review plan for the review of safety analysis reports for nuclear power plants, LWR edition. NUREG-0800.

USNRC, 2012a. Standard technical specifications - Babcock and Wilcox Plants. NUREG-1430, Rev. 4, Vols 1 and 2.

USNRC, 2012b. Standard technical specifications - Combustion Engineering Plants. NUREG-1432, Rev.4, Vols 1 and 2.

USNRC, 2012c. Standard technical specifications - General Electric BWR/4 Plants. NUREG-1433, Rev. 4, Vols 1 and 2.

USNRC, 2012d. Standard technical specifications - General Electric BWR/6 Plants. NUREG-1434, Rev. 4, Vols 1 and 2.

USNRC, 2012e. Standard technical specifications - Westinghouse Advanced Passive 1000 (AP1000) Plants. NUREG-2194, Rev. 4, Vols 1 and 2.

USNRC, 2012f. Standard technical specifications - Westinghouse Plants. NUREG-1431, Rev. 4, Vols 1 and 2.

Versteeg, H., Malalasekera, W., 2007. An introduction to computational fluid dynamics - The finite volume method. Pearson education.

Weaver, W., Tomlinson, E., Aumiller, D., 2002. A generic semi-implicit coupling methodology for use in RELAP5-3D. Nuclear Engineering and Design 211, 13 – 26.

Xing, L., Yeung, H., Lo, S., 2011. Investigation of slug flow induced forces on pipe bends applying STAR-OLGA coupling, in: 15 International Conference on Multiphase Production Technology.

Yan, Y., 2011. Development of a coupled CFD - system code capability (with a modified porous media model) and its applications to simulate current and next generation reactors. Ph.D. thesis. University of Illinois at Urbana-Chamapaign.

Appendix A

Object-oriented design principles

The content of this appendix has been adapted or reproduced from Stroustrup (2013) and Meyers (2014).

A.1 Encapsulation

Encapsulation is the enforcement of abstraction by preventing the access to the implementation details of an object or a group of objects except through a well defined interface (Stroustrup, 2013). Access to class members can be controlled by declaring them public, protected or private. If a member is protected or private it can be accessed only from inside the class hence, to access it from outside, the class must have methods that allow the interaction with the member. Typically, this is done with `set` and `get` methods.

For more information it is advised to read Stroustrup (2013) and Meyers (2014).

A.2 Inheritance

Inheritance is when a class is based on another class. This helps code reuse and allows independent extension of the original code. In C++ inheritance establishes an "*is-a*" relationship between the derived class **D** and the base class **B** such that **D** *is-a* **B** but not vice versa. Inheritance can be of two types:

- Interface inheritance: An object of a derived class can be used whenever an object of a base class is required (The base class acts as an interface for the derived class). This feature is very useful to replace conditional statements and type casting. It is also known as run-time polymorphism and it is achieved with virtual functions.
- Implementation inheritance: A base class provides functions or data that are common to the derived classes, thus simplifying the implementation of the latter.

For more information it is advised to read Stroustrup (2013) and Meyers (2014).

A.3 Polymorphism

Polymorphism consists of providing a single interface to entities of different types (Stroustrup, 2013). Polymorphism can be subdivided into two main categories: run-time and compile-time polymorphism. In C++ the operations that allow the two types of polymorphism are the following:

- Run-time polymorphism - Virtual functions, interface inheritance
- Compile-time polymorphism - Overloading, templates

Run-time polymorphism is described in appendix A.2. Compile-time, or parametric, polymorphism is the uniform use of classes not related by inheritance provided by templates or

overloading. In other words, compile-time polymorphism allows functions/classes to operate with different data types without the need for the function/class to be rewritten for each type.

The main difference between run-time and compile-time polymorphism is that in run-time polymorphism if a class **A** implements some virtual functions, the specific functions of **A** to be called will be determined at run-time based on **A**'s dynamic type, whereas in compile-time polymorphism it is possible to have calls to functions involving **A** that might involve instantiating templates to make these calls succeed. These instantiations occur during compilation and they lead to different functions being called, hence the name compile-time polymorphism.

For more information it is advised to read Stroustrup (2013) and Meyers (2014).

A.4 RAII

Resource acquisition is initialisation (RAII) is a programming technique that relies on the properties of constructors and destructors and their interaction with exception handling to manage resources by using local objects (Stroustrup, 2013).

When a part of code acquires a resource - i.e. allocates memory, opens files etc. - it is essential for the resource to be released properly so that memory leaks or any unexpected behaviour of the system are prevented. Careful programming can prevent those issues but the code might change over time due to refactoring, maintenance or other reasons that require modification of the original code. A person that did not design the part of software that is acquiring the resource might not fully grasp the resource management strategy implemented and might break the code inadvertently.

RAII is a good candidate to prevent this from happening. RAII consists of turning over the acquired resources to resource-managing objects. By doing this the resource acquired by the managing object will certainly be released when the managing object's destructors will be called.

For more information it is advised to read Stroustrup (2013) and Meyers (2014).

Appendix B

Design patterns

The content of this appendix has been adapted or reproduced from Gamma et al. (2014) and Nystrom (2014).

B.1 Singleton

Ensure a class only has one instance, and provide a global point of access to it.

B.1.1 Motivation

It is important for some classes to have exactly one instance. A global variable makes an object accessible but it does not prevent multiple instantiations. Making the class itself responsible for keeping track of its sole instance enables the class to ensure that no other instance will be created. The class also provides a way of accessing the instance.

- It does not create the instance if it is not being used
- It is initialized at runtime

- It is possible to subclass

B.1.2 Applicability

Use the Singleton pattern when:

- There must be exactly one instance of a class and it must be accessible from a known access point

B.1.3 Implementation

Two implementations relying lazy initialisation are presented here: one returns a pointer and the other returns a reference.

```
class Singleton {
public:
  static Singleton* getInstance()
  {
    if (_instance == nullptr)
    {
      _instance = new Singleton();
    }
    return _instance;
  }
protected:
  Singleton()
  {
    // Singleton ctor implementation here
  }
private:
  static Singleton* _instance;
```

Listing B.1: Pointer implementation

```
class Singleton {
public:
  static Singleton & getInstance()
  {
    static Singleton instance;
    return instance;
  }
protected:
  Singleton()
  {
    // Singleton ctor implementation here
  }
}
```

Listing B.2: Reference implementation

B.2 State

Allow an object to alter its behaviour when its internal state changes. The object will appear to change its class.

B.2.1 Motivation

An object can be in one of several different states. When an object receives requests from other objects, it responds differently depending on its current state. Introducing an abstract class that represents the possible states of the object means declaring an interface common to all classes that represent different operational states. Every State subclass is also a Singleton.

- It provides a common interface for all the classes representing operational states
- It localises state-specific behaviour
- It makes state transitions explicit

B.2.2 Applicability

Use the State pattern when:

- An object's behaviour depends on its state and must change its behaviour at run-time depending on its state
- Operations have large multi-part conditional statements depending on the object's state. The state pattern puts each branch of the conditional in a separate class, allowing the object's state to be treated as an object in its own right that can vary independently from other objects.

B.2.3 Implementation

In this example, every time the method `action()` from class `GenericObject` is instantiated the state is updated to StateTwo if the current state is StateOne and vice versa.

```
class GenericObject {
public:
  GenericObject()
  {
    // GenericObject ctor implementation here

    _state = StateOne::getInstance();
  }
  void action()
  {
    // action() method implementation here

    _state->changeState(this);
  }
  // Other methods may be present

private:
  void changeState(State * newState)
  {
    _state = newState;
  }
  State * _state;
}
```

Listing B.3: Generic object class

```
class State {
public:
  State()
  {
    // State ctor implementation here
  }
  ~State() {}
  virtual void changeState(GenericObject * object) = 0;

  // Other methods may be present
}
```

Listing B.4: State base class

```
class StateOne : public State {
public:
  static StateOne & getInstance()
  {
    static StateOne instance;
    return instance;
  }
  void changeState(GenericObject * object)
  {
    object->changeState(StateTwo::getInstance());
  }

  // Other methods may be present
}
```

Listing B.5: StateOne derived class

```
class StateTwo : public State {
public:
  static StateTwo & getInstance()
  {
    static StateTwo instance;
    return instance;
  }
  void changeState(GenericObject * object)
  {
    object->changeState(StateOne::getInstance());
  }

  // Other methods may be present
}
```

Listing B.6: StateTwo derived class

B.3 Factory method

Define an interface for creating an object, but let subclasses decide which class to instantiate.

B.3.1 Motivation

Frameworks use abstract classes to define and maintain relationships between objects. Often it is the framework's responsibility to create objects as well. Clients have to subclass the aforementioned abstract classes in order to realise their framework-specific implementation. Because the particular subclass to instantiate is framework-specific, the framework's abstract classes are not able to predict which subclasses to instantiate. The factory method solves the problem by encapsulating the knowledge of which subclass to create and moves this knowledge out of the framework.

- It provides hooks for subclasses
- Connects parallel classes hierarchies

B.3.2 Applicability

Use the Factory Method pattern when:

- A class is not able to anticipate the class of object it must create
- A class wants its subclasses to specify the objects it creates
- Classes delegate responsibility to a helper subclass and the knowledge of which subclass is delegated is needed

B.3.3 Implementation

Several variations of the factory method pattern are available. The one presented hereafter is and adaptation of the parametrised factory method presented in Gamma et al. (2014).

```
class GenericObject
{
public:
  GenericObject()
  {
    // GenericObject ctor implementation here
  }
  virtual ~GenericObject() {}

  // Other methods may be present
}
```

Listing B.7: GenericObject base class

```
class ObjectOne : public GenericObject
{
public:
  ObjectOne()
  {
    // ObjectOne ctor implementation here
  }

  // Other methods may be present
}
```

Listing B.8: ObjectOne derived class

```
class ObjectTwo : public GenericObject
{
public:
  ObjectTwo()
  {
    // ObjectTwo ctor implementation here
  }

  // Other methods may be present
}
```

Listing B.9: ObjectTwo derived class

```
class Creator
{
public:
  Creator()
  {
    // ObjectTwo ctor implementation here
  }
  ~Creator() {}
  virtual GenericObject * create(int id)
  {
    if (id == 1)
    {
      return new ObjectOne();
    }
    if (id == 2)
    {
      return new ObjectTwo();
    }
    // Repeat for other objects

    throw std::invalid_argument("Unable to create object");
  }
  // Other methods may be present
}
```

Listing B.10: Creator class

Appendix C

RELAP5-3D input deck

C.1 Input deck structure

A RELAP5-3D input deck consists of at least one title card, optional comment cards, data cards and a terminator card. The order of the title, data and comment cards is not critical, however if multiple title cards or data cards with the same id number are inserted, only the last title and/or data card is used. The input deck must have a termination card at the end.

A limit of 80 columns must not be exceeded for all lines unless commented out. In case it is not possible to fit a line within 80 columns it is possible to use a continuation card. The name of the variables must not exceed a maximum limit of eight characters in length.

General practice consists of an input deck starting with the title card, followed by data cards arranged in number order, however data cards can be inserted in any order.

Several types of input are specified in sequential expansion format, which consists of sets of data, each of which contains one or more data items followed by an integer number. This allows the code to recognise which parameters are to be expanded (i.e. the data set) and expansion stopping points (i.e. the integer number).

The stopping points are generally junction, volume or radial mesh point numbers, always

forming an increasing sequence.

Expansions begin at the point following the termination point of the previous set and continue to the termination point of the current set. For the first set the expansion starts at the first point.

RELAP5-3D general cards are listed hereafter:

- Title card (=)
- Comment card (*)
- Continuation card (+)
- Termination card (.)

Data cards are divided by id as follows:

- Miscellaneous (Sparsely from 1 to 147)
- Time control (201 to 299)
- Minor edits (301 to 399)
- Trips (400 to 799)
- Hydraulic components (CCCxxxx)
- Heat structures (1CCCGxxxx)
- Material tables (201MMMxx)
- General tables (202TTTxx)
- Control variables (205CCCxx or 205CCCCx)
- Radiation modules (6SSNN001 to 6SSNN199)
- Reactor kinetics (30000000 to 30002999)

The letters C, G, M, N, S and T represent values chosen by the user to identify the different components or models used, whereas the letter x identifies the specific cards of each component. For more details the reader is advised to read INL (2014b).

C.1.1 List of hydrodynamic components

The hydrodynamic components, with their respective alphanumeric identifier for the input deck are listed hereafter, divided into general and specialised components.

Specialised components are implemented with more complex models to simulate behaviours that would require a sensible modelling effort if only the general components were to be used.

Component	Id	Component	Id
Time dependent volume	`tmdpvol`	Time dependent junction	`tmdpjun`
Single junction	`sngljun`	Multiple junction	`mtpljun`
Pipe	`pipe`	Annulus	`annulus`
Branch	`branch`		

Table C.1: General hydrodynamic components

Component	Id	Component	Id
Separator	`separatr`	Jet mixer	`jetmix`
Pump	`pump`	Turbine	`turbine`
Valve	`valve`	Accumulator	`accu`
ECC mixer	`eccmix`	Pressuriser	`prizer`
CANDU channel	`canchan`	Feedwater heater	`fwhrt`

Table C.2: Specialised hydrodynamic components

C.2 Sample input deck

Input decks can be constructed by manually creating or editing a text file, which can be tedious but it ensures accuracy. Alternatively, the USNRC software SNAP can be used to generate the input deck automatically. However, an input deck generated in such way might require some manual adjustments. SNAP is a Java based GUI with different plugins for different system codes, so that the correct syntax can be used while creating the input deck.

The following listing is the input deck for a simple pipe flow problem. The parameters of the problem are the following:

- L = 3 m
- D = 0.01 m
- $\Delta x = 0.04$
- P = 101325 Pa
- T = 300 K

The inlet mass flow rate is increased linearly from 0 kg/s to 0.366 kg/s over a time of 5 s.

```
=SIMULATION NAME

**************************
* PROBLEM SPECIFICATIONS *
**************************
*
* PROBLEM TYPE
*
* NEW/RESTART
* STEADY/TRANSIENT
*
100  new    transnt

*
* INPUT/OUTPUT UNITS SELECTION
*
* SI
* IMPERIAL
*
102  si     si

*
* TIME CONTROL
*
* END TIME
* MINIMUM TIMESTEP
* MAXIMUM TIMESTEP
* COMPUTATIONAL OPTIONS
* FILE DUMP OPTIONS
```

```
*
201   20.    1.e-10    1.e-2    7    50    50    50

********************
* HYDRAULIC SYSTEM *
********************
*
* 10x components: PIPES
* 20x components: JUNCTIONS following PIPES 10x
* 50x components: TDVs of the secondary loop
* 60x components: JUNCTIONS of the secondary loop
*

1300000 source tmdpvol
1300101 7.854e-5 0. 1000.
1300102 0. 0. 0.
1300103 0. 0. 0
1300200 3
1300201 0. 101325. 300.

1400000 inlet tmdpjun
1400101 130010000 150000000 7.854e-5 0
1400200 1
1400201 0. 0. 0. 0.
1400202 5. 0.366 0. 0.

1500000 duct pipe
1500001 75
1500101 7.854e-5 75
1500301 0.04 75
1500401 0. 75
1500501 0. 75
1500601 90. 75
1500801 0. 0. 75
1501001 0 75
1501101 0 74
1501201 003 101325. 300. 0. 0. 0. 75
1501300 1
1501301 0. 0.366 0. 74

1600000 outlet sngljun
1600101 150010000 170000000 7.854e-5
1600102 0. 0. 0
1600201 1 0.366 0. 0.

1700000 sink tmdpvol
1700101 7.854e-5 0. 1000.
1700102 0. 0. 0.
```

```
1700103 0. 0. 0
1700200 3
1700201 0. 101325. 300.

*
* TERMINATION CARD

.
```

Listing C.1: RELAP5-3D sample input deck