TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Nukleartechnik

Development of a Multi-Physics, Multi-Scale simulation tool for LWR safety analysis

Yann Périn

Vollständiger Abdruck der von der Fakultät für Maschinenwesen
der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:	Prof. Dr. rer. nat. Rudolf Neu
Prüfer der Dissertation:	1. Prof. Rafael Macián-Juan, Ph.D.
	2. Prof. Kostadin Ivanov, Ph.D.
	North Carolina State University, Raleigh, USA

Die Dissertation wurde am 04.07.2016 bei der Technischen Universität München
eingereicht und durch die Fakultät für Maschinenwesen am 27.09.2016 angenommen.

Abstract

Most of the acceptance criteria for rigorous safety analyses are based on local values within the reactor core: thermal-hydraulic, neutron-kinetic or thermal-mechanic parameters at the fuel pin level. The main goal of performing safety analysis is to check whether these criteria are met during a transient or an accident in a Nuclear Power Plant.

The present work describes the development of a multi-physics, multi-scale simulation tool that can model the whole Nuclear Power Plant behavior, the core at the assembly level and the core hot spot at the pin level in order to evaluate local safety parameters.

For the simulation of the thermal-hydraulic processes of a Nuclear Power Plant, the thermal-hydraulics system code ATHLET is applied. In order to model the plant level as well as the pin level, ATHLET is coupled to the sub-channel code CTF using the parallel coupling method.

In this thesis, both thermal-hydraulics codes were coupled to the neutron diffusion codes QUABOX/CUBBOX and DYN3D at the assembly level using the internal coupling method.

For the pin level, several methods to compute the pin power are presented. Pin-by-pin flux reconstruction and Superhomogenization procedure for direct pin-by-pin calculation with a diffusion code are the methods currently applied.

Furthermore, two examples of internal couplings of a neutron transport code (TORT-TD and nTRACER) with a subchannel code (CTF) at the pin level are also described. Those developments are a proof of concept of so-called high fidelity couplings. High fidelity coupled systems are currently too costly for routine

applications but represent the future of neutron-physics/thermal-hydraulics simulations.

Finally, in the framework of a European project named NURESAFE, a new coupling approach using parallel processing on the Salomé platform was applied and is presented. The resulting multi-physics, multi-scale coupled system between ATHLET, CTF and DYN3D was used for advanced coupled simulations of PWR and BWR transients.

For the PWR application, a Main Steam Line Break transient at Hot Zero Power was selected. ATHLET-CTF-DYN3D models the whole system from plant level to local pin value in the hot assembly. The CTF model is a hybrid (1:1) mapping with a pin-by-pin resolution in the hot channel. The pin-by-pin power is calculated in DYN3D using the flux reconstruction method. The comparison with a coarser model shows the advantage of a pin level nodalization.

For the BWR application, a Turbine Trip Without SCRAM was selected. It was the first BWR / high void condition application of the ATHLET-CTF parallel coupling developed during this thesis. The good agreement with the validated ATHLET-DYN3D coupled solution shows the validity of this method even for challenging thermal-hydraulic simulations.

Contents

List of Figures

List of Tables

List of Acronyms

ATHLET Analysis of Thermal-hydraulics of LEaks and Transients

ATWS Anticipated Transient Without SCRAM

BMWi Bundesministerium fuer Wirtschaft und Energie

BOC Beginning Of Cycle

BWR Boiling Water Reactor

CMFD Coarse-Mesh Finite Difference

CR Control Rod

EOC End Of Cycle

FW Feedwater

GRS Gesellschaft fur Anlagen- und Reaktorsicherheit

HZDR Helmholz Zentrum Dresden Rossendorf

I/O Input/Output

IRSN Institut de radioprotection et de surete nucleaire

KIT Karlsruhe Institute of Technology

MOC Method Of Characteristics

MSIV Main Steam Isolation Valve

MSLB Main Steam Line Break

NP Neutron Physics

PWR Pressurized Water Reactor

SG Steam Generator

UPM Universidad PolitÃľcnica de Madrid

TH Thermal-Hydraulics

TSV Turbine Stop Valve

Acknowledgments

The work presented in this thesis was partly founded by the German Ministry for Economic Affairs and Energy (Bundesministerium fuer Wirtschaft und Energie (BMWi)).
Part of the work was also founded by the European Commission under the 7th EURATOM Framework Program within the NURESAFE Project contract No. 323263.

On a more personal note, I would like first to express my sincere gratitude to my adviser Prof. Rafael Macián for accepting me as an external Ph.D student and for his guidance during the long process that was this thesis.
My sincere thanks also goes to Dr. Kiril Velkov, head of the Core Behavior Department at GRS. He is more than just a boss and has always been a mentor to me.
I also wish to thank Prof. Kostadin Ivanov. First for accepting to be "Pruefer" of my thesis. But also for welcoming me several times at State College. The work achieved during those stays was very important for this work.
I thank my fellow GRS colleagues and in particular my colleagues at the Core Behavior Department. Working there with you is a real pleasure.
Also I thank my colleagues from all over Europe, met during the NURESAFE project, the work we achieved there is a big part of this thesis. In particular, I am grateful to Dr. Javier Jimenez, without his help, leading that work package would have turned into a catastrophe.

Of course, I would like to thank my family, for their support during this project. Without them regularly asking, "Is your thesis progressing?", I may have needed a few years more.
Last but not least, I want to thank my wife, Catherine. Without her, I don't think I would have made it that far.

CHAPTER 1

Introduction

Importance of Multi-Physics Multi-Scale Simulations

In a nuclear reactor core, different fields of physics interact. A prime example is the interaction between Neutron Physics (NP) and Thermal-Hydraulics (TH). The interaction between neutron physics (NP) and fuel thermo-mechanics (TM) is another one. In the early years of the nuclear industry, the available computational power only allowed the simulation of one field of physics at once and only in a limited fashion. Often, these codes included a simplified model of the other fields of physics: e.g. point kinetic model in a thermal-hydraulics code, 4-equation 2-phase model in a neutron physics code, etc. However, those simplified models sometimes led to inaccurate results and necessitated conservative assumptions (e.g. boundary conditions) to provide conservative results that could be used for safety analysis. State-of-the-art nuclear safety analysis was then to use conservative models and assumptions regarding both initial and boundary conditions.

In the nineties, thanks to the constant improvements in computational capabilities, thermal-hydraulic system codes were coupled with 3D neutronics models. Since

then, the use of coupled three-dimensional Thermal-Hydraulics / Neutron Physics (TH/NP) multi-physics codes has become state-of-the-art for performing reactivity initiated accident and several operational transient analyses.
Reference [1] is a state of the art report on thermal-hydraulics / neutron physics coupling published by the OECD/NEA in 2004. In this report, a list of transients which require a 3D neutronic modeling of the core is provided. A short description for each transient can also be found.
For PWR (and VVER), seven transient types are identified. Among them is the MSLB which is the PWR transient simulated in Chapter 7.
For BWR, eight transient types are identified. Among them is the Turbine Trip with ATWS conditions which is the BWR transient simulated in Chapter 7.
Most TH/NP coupled systems described in [1] are between system codes and neutron diffusion codes: RELAP5, TRAC, TRACE and ATHLET have been coupled with various neutron diffusion codes, such as DYN3D, QUABOX/CUBBOX, PARCS, PANTHER etc.
A few examples of couplings between subchannel and neutron diffusion codes are also given: VIPRE/ARROTA, COBRA-IIIC/QUABOX-CUBBOX and FLICA/ CRONOS.
Before the start of this work, the standard code for multi-physic applications at the GRS was the coupled code ATHLET- QUABOX/CUBBOX.

Most of the acceptance criteria for rigorous safety analyses are based on local values within the reactor core: thermal-hydraulic, neutron-kinetic or thermal-mechanic parameters at the fuel pin level. The main goal of performing safety analysis is to check whether these criteria are met during a transient or an accident in a NPP.
In order to simulate the behavior of a fuel pin during a transient, one often needs to simulate the behavior of the whole nuclear power plant. In this thesis, four scale levels are defined:

- At the plant level, the whole nuclear plant is modeled, from the core to the turbine, usually with coarse geometric nodalization. The plant controllers are also modeled. So-called thermal-hydraulics system codes are used at this level.

- At the core level, the core is described by one or few thermal-hydraulic channels, usually with a system code. In PWR (including VVER), the neutron physics is simulated with the point kinetic approximation, which means that the power profile is constant. In BWR a 1-Dimensional approximation with dynamic axial power profile is usually applied.

- At the assembly level each fuel assembly is modeled by a dedicated thermal-hydraulic channel. It is possible to use system codes, but subchannel codes are usually preferred at this level. At the assembly level, the reactor physics is usually simulated with a diffusion code wherein each assembly is modeled by (at least) one radial node.

- At the pin level, the thermal-hydraulic channels can be either fuel- or coolant-centered. Subchannel codes are the preferred option for TH modeling but recently CFD codes have started to be used. For the neutron physics, the power for each pin of an assembly can be obtained by different methods. Diffusion codes with pin power reconstruction or direct pin-by-pin diffusion calculation with Super Homogenization (SPH) correction can be applied for fast calculations. For high-fidelity simulations, transport codes are preferred.

At the GRS, for multi-scale applications up to the pin level, the following methodology is currently applied: The ATHLET plant model is always the starting point. The thermal-hydraulic core model is often developed separately from the plant model, together with the neutron physics model in order to optimize the so-called mapping scheme.
The generation of macroscopic cross-sections for the diffusion code is a crucial step where a lot of know-how is needed. During this step, the form functions used for the determination of pin-power peaking factor are also generated. The macroscopic cross-sections are used for:

- The generation of reactivity coefficients for the point-kinetics model in the system code. Those coefficients are generated with stand-alone neutron diffusion code calculations.

- In parametrized cross-section libraries for calculations with the multidimensional neutron physics code coupled with ATHLET.

The thermal-hydraulic core model is usually composed of so-called super-channels each modeling several fuel assemblies. In addition, one (or several) "hot channel", describing the assembly with the highest power is modeled. To access values at the pin level, a "hot pin" channel is also modeled. The power in this pin is computed using the hot assembly power multiplied by a hot pin factor. With this approach, the hot pin factor remains constant during the transient.

High-fidelity simulation tools refer to systems in which the thermal-hydraulics and the neutron physics are both described at a pin level. In addition, the individual models should provide a high accuracy. For the TH model, a 3D subchannel code or a CDF code is typically applied. The neutronic model solves the neutron transport equation without approximation, either with deterministic or with probabilistic Monte Carlo methods.
Several high-fidelity coupled systems are currently under development worldwide. Recently, the neutronic code DYN3D using its simplified transport SP3 solver was coupled to the commercial CDF code Ansys CFX [2]. Another example is the successful coupling of the transport code nTRACER and the subchannel code MATRA [3, 4]. Finally, the most famous of such projects in development is the American CASL initiative. Within this framework CTF was successfully coupled to the MPACT (Michigan PArallel Characteristics Transport Code) transport code [5].
High-fidelity coupled systems represent the future of neutron-physics/thermal-hydraulics simulations.

Ideally a high-fidelity simulation tool could be applied for the simulation of safety parameters at the fuel pin level. However, all the above-mentioned high-fidelity coupled systems are currently too costly for routine applications or even lack transient capabilities.
Nevertheless, at each scale level, reliable thermal-hydraulics and neutron physics codes are available. By coupling them, it is possible to develop a complete multi-physics, multi-scale simulation tool giving access to the required precision for safety

analysis at each scale level while keeping the computational costs to a reasonable level.
The work performed for toward this goal is presented with the following structure.

First, the codes and the different coupling methods that were used for this work are presented in more detail in Chapter 2.

The TH system code ATHLET is systematically applied for modeling the plant level. Its coupled version with QUABOX/CUBBOX at the assembly level is a legacy that was the basis for most of this work and is described in Chapter 3.
In a multi-scale model, the subchannel code is usually used to simulate the whole core or a subregion of it. For instance, in some licensing relevant cases a "hot channel" analysis might be sufficient. Nevertheless, it is desired for the subchannel code to have the capability of calculating a full core model at the assembly level. The development of the coupled system CTF-QUABOX/CUBBOX, described in Chapter 3, was thus a preliminary but essential step toward a fully operational multi-physics, multi-scale simulation tool.

The aim of this work is to develop a multi-scale simulation tool that can model the core at the pin level in order to evaluate local safety parameters. Several methods to simulate pin power distributions are presented in Chapter 4. Pin-by-pin flux reconstruction in a diffusion code, SPH procedure for direct pin-by-pin calculation with a diffusion code and finally neutron transport codes can be used for this purpose. Two examples of coupling of a neutron transport code with a subchannel code at the pin level were developed and are described. Those examples are a proof-of-concept of so-called high-fidelity coupling.

In order to model the plant level as well as the pin level, in addition to TH/NP multi-physics couplings, the system code and the sub-channel code also need to be coupled. These developments are described in Chapter 5

Finally, in the framework of a European project named NURESAFE, a new coupling approach on the Salomé platform has been applied and is presented in Chapter 6. The resulting tri-code (system code ATHLET, subchannel code CTF and neutronic diffusion code DYN3D) multi-physics (TH and NP) multi-scale (from

plant to pin level) coupled system was used for advanced coupled simulations of a PWR and a BWR transient. Those results are described in the last chapter of this thesis.

This new multi-physics multi-scale simulation tools achieves the goal of providing a more realistic representation of the reactor core to improve the estimation of the safety margins while keeping the computational costs to reasonable levels.

CHAPTER 2

Coupling Methods and Codes

In Chapter 1 we have introduced the concepts of multi-scale and multi-physics simulations. These rely on the coupling of codes that are each specialized in a branch of physics and applicable to a given scale. This chapter consists of two parts. In the first one the different methods for code coupling are explained. In the second one, the main codes used in this thesis are briefly introduced.

2.1 Coupling Methods

2.1.1 Offline Coupling vs. Online Coupling

We differentiate the online and offline couplings. With the offline coupling, the data transfer between codes is achieved through I/O (Input/Output) files. The consequence is that no changes are needed in the codes. The I/O data exchange is usually automatized with scripts. This is the advantage of the offline coupling strategy over the online coupling. The offline coupling strategy is usually only a step toward the online coupling, as I/O processes are time inefficient compared to direct memory transfers. Two examples of offline couplings are presented in this work. In Section 5.1, the offline coupling between ATHLET and CTF is presented. In Section 7.1, a so-called mixing matrix calculated by a CFD code

is applied at the core inlet to distribute the inlet temperature in each CTF core channel. This application can also be considered as an offline coupling between CFD and subchannel code.
With the online coupling data transfer is done directly in the memory through programmed interfaces. Online coupling can use either serial integration or parallel processing (see description thereafter). The couplings presented presented in this thesis mostly use the online coupling method.

2.1.2 Serial integration vs. Parallel processing

With the "serial integration", the coupled codes are compiled together. One code is the master program while the other(s) code(s) is a subroutine of the master program. Typically, in TH/NP coupling the TH code is the master. This approach necessitates extensive changes in both programs. The resulting executable is usually unable to run stand-alone cases of the slave programs.

With parallel processing, the data exchange is made by a third party program such as PVM, MPI or more recently Salomé. The simulations are not necessarily run in parallel but can actually be consecutive. The coupling interfaces for each code must be compatible with the coupling supervisor without depending directly on the other code. When a well-defined standard is adopted, a coupling with one code in such an environment allows a coupling to any other code compliant with the standard. One other advantage is that all codes retain their stand-alone capability.

2.1.3 Internal, Parallel and External Coupling

For the coupling of two thermal-hydraulic codes or between a neutron kinetic and a thermal-hydraulic code, several coupling methods are available: external, internal and parallel coupling.

Internal Coupling

The internal coupling is only applicable to Thermal-Hydraulics / Neutron physics coupling. In an internal coupled system (see Figure 2.1), the thermal-hydraulic

code simulates the fluid dynamics and the heat exchanges in the reactor core. The neutron physics code only simulates neutronics in the core (any existing internal TH model is switched off). TH feedback core distributions (moderator density and/or temperature, fuel temperature, boron concentration) are transferred to the neutronic code which returns core power distributions.
The internal coupling solution has been adopted for all Thermal-Hydraulic / Neutron Physics couplings presented in this thesis.

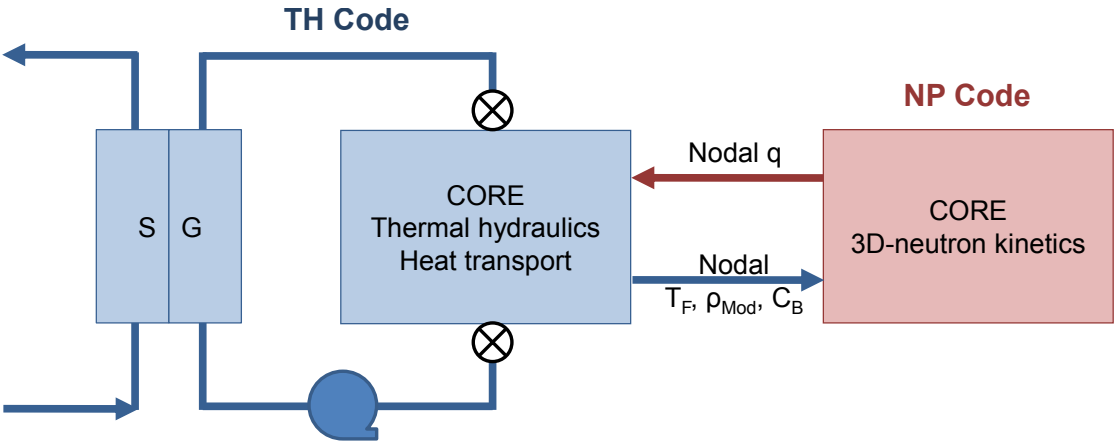

Figure 2.1: Internal Coupling

Parallel Coupling

The parallel coupling method can be used for the coupling of two thermal-hydraulic codes or between a thermal-hydraulic and a neutron kinetic code. When coupling a thermal-hydraulic code to neutron kinetic code, an internal TH model is needed in the neutron physics code. In a parallel coupled system (see Figure 2.2), the thermal-hydraulic code simulates the fluid dynamics and the heat exchanges in the reactor core. The neutron kinetic code simulates neutronics as well as the thermal-hydraulic in the core (using its internal TH model). Core thermal-hydraulic boundary conditions (inlet mass flow, inlet temperature, inlet boron concentration and outlet pressure) are transferred to the neutronic code which returns core power distributions.

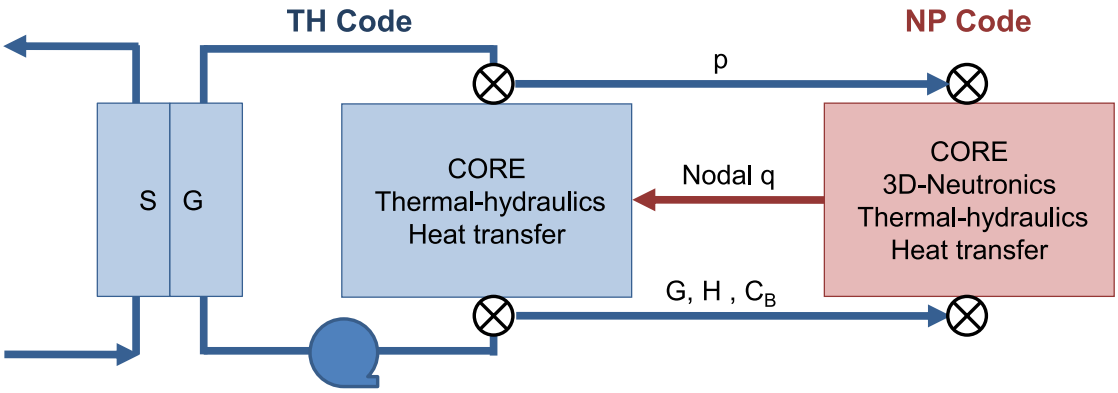

Figure 2.2: Parallel Coupling

External Coupling

The external coupling method can be used for the coupling of two thermal-hydraulic codes or between a thermal-hydraulic and a neutron physics code. When coupling a thermal-hydraulic code to neutron kinetic code, an internal TH model is needed in the neutron physics code. In an external coupled system (see Figure 2.3), the thermal-hydraulic code does not simulate the reactor core. The neutron kinetic code simulates neutronics as well as the thermal-hydraulic in the core (using its internal TH model). Core thermal-hydraulic boundary conditions are exchanged between the neutronic code and the thermal-hydraulic code: inlet/outlet mass flow, inlet/outlet enthalpy (or temperature), inlet/outlet boron concentration and inlet/outlet pressure.

2.1.4 Numerical Methods and Time Coupling

In numerical simulation of time-dependent systems, one usually differentiates explicit, implicit and sometimes semi-implicit methods.

Explicit Coupling

In explicit schemes, the value of a given parameter at time t_{n+1} does not depend upon other values at the current level but only of values at t_n. For code coupling explicit methods have the advantage of being relatively easy to implement. How-

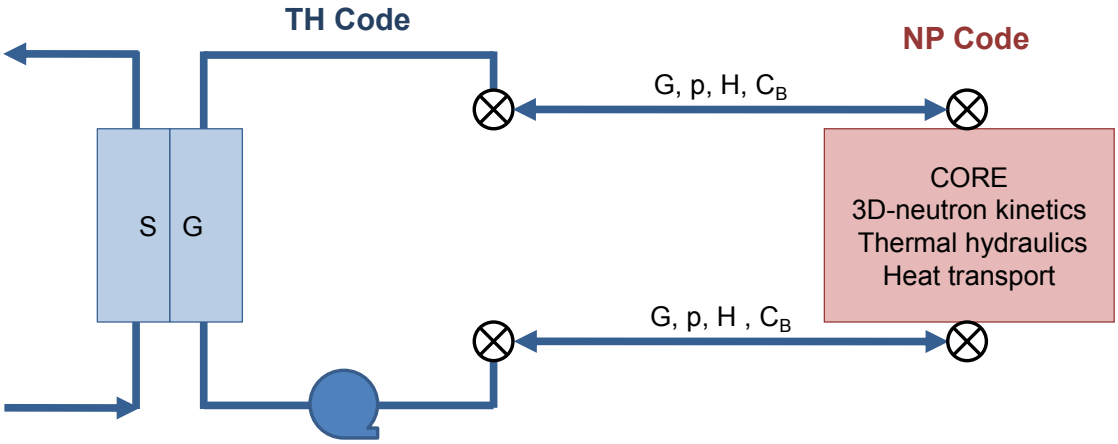

Figure 2.3: External Coupling

ever, in order to keep a good accuracy and avoid numerical instability, short time steps are usually necessary. An illustration of this problem on an actual transient simulation is provided in Section 3.1.

An example of explicit time coupling is the staggered time step synchronization scheme. This scheme is used throughout this study.
An illustration of this scheme for a TH/NP coupling is presented in Figure 2.4.

1. Power (at t_n) transferred from the NP code to the TH code
2. TH code performs a time-step (Δt)
3. TH Feedback (at t_{n+1}) transferred from TH code to the NP code
4. NP code performs the same time-step (Δt)
5. Power (at t_{n+1}) transferred from the NP code to the TH code

Semi-Implicit Coupling

Semi-implicit methods are actually closer to explicit methods than implicit ones. Indeed, with semi-explicit methods, the equation systems are still solved separately. The difference with explicit methods however is that the data exchange

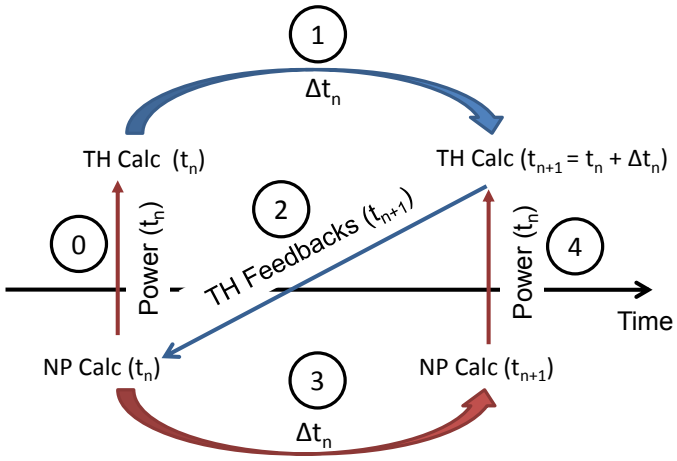

Figure 2.4: Explicit Staggered Time Step Synchronization Scheme

becomes an iterative process. Each time step is repeated in each code separately with new boundary conditions until convergence.
Like explicit methods, semi-implicit methods can be implemented without changing the equation system. They are more stable and more accurate than the latter (for long time steps) but more costly.

Semi-implicit couplings are relatively popular for TH/TH couplings. Several examples can be found in the literature. In the present work, no semi-implicit coupling was developed. It is however a possibility for future work which will be discussed in the conclusions.

Implicit Coupling

In an implicit numerical method all equations coupled by common variables must be solved simultaneously. The system can be solved either with direct matrix inversion or with an iterative technique. In the case of code coupling, this would mean rewriting the codes to obtain a single set of inter-related equations.
This approach is therefore more difficult to implement and can involve many iterations per time step. The advantage is that implicit schemes are very stable and can potentially have much larger time steps.

No implicit coupling was developed in the present work. In the litterature, examples of implicit multi-physics coupling can be found but are usually limited to simple models, e.g. 1D thermal-hydraulics with 1D neutron diffusion.

2.2 Codes Presentation

In order to cover all levels of simulation (from plant to fuel pin level), a wide range of codes were used for this work. The most important ones are presented in more detail thereafter. The section is divided according to the different branches of physics: thermal-hydraulics and neutron physics.

2.2.1 Thermal-Hydraulics Codes

ATHLET

The thermal-hydraulic computer code ATHLET (Analysis of Thermal-hydraulics of LEaks and Transients) is developed by the GRS for the analysis of anticipated and abnormal plant transients, small and intermediate leaks as well as large breaks in light water reactors [6]. ATHLET is being applied by numerous institutions in Germany and abroad. The development and validation of ATHLET is sponsored by the German Federal Ministry of Economics and Technology (BMWi).
The aim of the code development is to cover the whole spectrum of design basis and beyond design basis accidents (without core degradation) for PWRs and BWRs with only one code. Since the release of version 2.2c ATHLET also has the capability to simulate different working fluids such as: heavy water, helium, liquid sodium and lead.
For all working fluids, the system of differential equations used in ATHLET is based on the following general conservation equations for the liquid and vapor phases.
The mass conservation equation:

$$\frac{\partial(\alpha_k \rho_k)}{\partial t} + \nabla \cdot (\alpha \rho_k \vec{V}_k) = \psi_k \tag{2.1}$$

Where the subscript k can only take two values: l for the liquid phase and v for the vapor phase. On the right-hand side, ψ_k is the inter-phase mass transfer. Since only two phases are considered, $\alpha_v = 1 - \alpha_l$.

The energy conservation equation:

$$\frac{\partial(\alpha_k \rho_k h_k)}{\partial t} + \nabla \cdot (\alpha \rho_k \vec{V}_k h_k) = \\ - p\frac{\partial \alpha}{\partial t} + \vec{\tau}_{i,k}\vec{V}_k + \alpha_k \vec{\tau}_i(\vec{V}_k - \vec{V}_m) + \alpha_k \rho_k \vec{g}\vec{V}_k + q'''_{w,k} + q_i + \psi h^i_k + S_{ext,k} \quad (2.2)$$

Where the subscript k can only take two values: l for the liquid phase and v for the vapor phase.
On the right-hand side, the terms are: pressure work, shear work at the phase interface, dissipation due to interfacial shear, gravitational work, heat flow through structures, heat flow at the phase interface, energy flow due to phase change, and external source term.

The momentum conservation equation:

$$\frac{\partial(\alpha_k \rho_k)\vec{V}_k}{\partial t} + \nabla \cdot (\alpha \rho_k \vec{V}_k \vec{V}_k) = \\ - \nabla(\alpha p) + \vec{\tau}_{i,k} - \alpha_k \vec{f}_w - \psi \vec{V}_k - alpha_k \rho_k \vec{g} + M_{WL} + M_{VM} + S_{M,k} \quad (2.3)$$

Where the subscript k can only take two values: l for the liquid phase and v for the vapor phase.
On the right-hand side, the terms are: pressure work, interfacial friction, wall friction, momentum flux due to phase change, gravitation, water level force, virtual mass, and external momentum source term.

After spatial integration, the above conservation equations lead to a set of six first order differential equations. The spatial integration of the conservation equations in ATHLET is performed on the basis of a finite-volume approach.

In this work, ATHLET is applied for modeling the plant level of the PWR and BWR applications presented in Chaper 7. The coupling with neutron physics codes at the assembly level is a legacy that has been the groundwork for this study. A new coupling approach, on the Salomé platform is newly developed and is presented in Chapter 6. Finally, the coupling with CTF, described in Chapter 5 is essential to the multi-scale aspect of this work.

CTF

Coolant-Boiling in Rod Arrays | Two Fluids (COBRA-TF) is a Thermal/Hydraulic (TH) simulation code designed for Light Water Reactor (LWR) subchannel analysis [7, 8].
The code was originally developed by Pacific Northwest Laboratory in 1980 and had been used and modified by several institutions over the last few decades. COBRA-TF also found use at the North Carolina State University (NCSU) by the Reactor Dynamics and Fuel Management Group (RDFMG) and has been improved, updated, and subsequently re-branded as CTF.
CTF is a 3D thermal hydraulics code which solves the transient balance equations based on a separated fluid representation of the two-phase flow. The two-fluid formulation, generally used in thermal-hydraulic codes, separates the conservation equations of mass, energy, and momentum to vapor and liquid. CTF extends this treatment to three fields: vapor, continuous liquid and entrained liquid droplets, which results in a set of nine time-averaged conservation equations: four mass, three momentum, and two energy. Four mass conservation equations are solved, respectively, for the vapor phase, continuous liquid phase, entrained liquid phase, and non-condensable gas mixture. The non-condensable gas mixture transport equation is solved explicitly at the end of each time step. The mass conservation equations are given below.

$$\frac{\partial}{\partial t}\left(\alpha_k \rho_k\right) + \nabla \cdot \left(\alpha_k \rho_k \vec{V}_k\right) = L_k + M_e^T \tag{2.4}$$

Where the k subscript takes the following value: the continuous liquid phase l, the vapor phase v, the entrained droplet phase e or the non-condensable gas mixture g.
On the right-hand side, the L_k term represents the mass transfer into or out of phase k. Inter-phase mass transfer can occur by either evaporation/condensation or by entrainment/deentrainment. The M_e^T term is the mass transfer in the mesh cell due to turbulent mixing and void drift.

Three momentum conservation equations are solved, respectively, for the vapor phase, continuous liquid phase and entrained liquid phase. The momentum conservation equations are given below.

$$\frac{\partial}{\partial t}\left(\alpha_k \rho_k \vec{V}_k\right) + \nabla\left(\alpha_k \rho_k u_k \vec{V}_k\right) = \alpha_k \rho_k \vec{g} - \alpha_k \nabla P + \nabla \cdot \left[\alpha_k \left(\boldsymbol{\tau}_k^{ij} + \boldsymbol{T}_k^{ij}\right)\right] + \vec{M}_k^L + \vec{M}_k^d + \vec{M}_k^T \quad (2.5)$$

Where the k subscript takes the following value: the liquid film field l, the vapor field v and the entrained droplet field e.
On the right-hand side, the terms are: gravitational force, pressure force, viscous and turbulent shear stress, momentum source/sink due to phase change and entrainment, interfacial drag forces, and momentum transfer due to turbulent mixing.

Only two energy conservation equations are solved: one for the vapor phase and one for the liquid phase. The use of a single energy equation for the liquid phase both continuous and entrained droplets, implies that both fields are at the same temperature for a given computational cell. The energy conservation equations are given below.

$$\frac{\partial}{\partial t}\left(\alpha_k \rho_k h_k\right) + \nabla \cdot \left(\alpha_k \rho_k h_k \vec{V}_k\right) = -\nabla \cdot \left[\alpha_k \vec{q}_k^T\right] + \Gamma_k h_k^i + q_{wk}''' + \alpha_k \frac{\partial P}{\partial t} \quad (2.6)$$

The right-hand side terms are: k-phase turbulence heat flux, energy transfer due to phase-change, volumetric wall heat transfer, and the pressure work term.

The conservation equations for each of the three fields and for heat transfer from and within the solid structure in contact with the fluid are solved using a semi-implicit, finite-difference numerical technique on an Eulerian mesh. It allows LWR (PWR, BWR and VVER) core simulation for best-estimate evaluations of nuclear reactor safety margins at the assembly and pin levels.

In the frame of the development of a multi-scale multi-physics simulation tool, several coupling with several neutron physics codes were developed. At the assembly level, CTF is coupled with QUABOX/CUBBOX (see Section 3.2) and DYN3D (see Section 6). At the pin level, CTF is coupled with two transport codes TORT-TD (see Section 4.3.1) and nTRACER (see Section 4.3.2) but also with DYN3D. For the transition from plant level to assembly/pin level, a TH/TH coupling with ATHLET was necessary. This coupling is described in Chapters 5 and Section 6.6

2.2.2 Neutron Physics Codes

QUABOX/CUBBOX

Developed at the GRS, QUABOX/CUBBOX [9] is a coarse-mesh 3D neutronics core model. It solves the diffusion equation (2.7) in two energy groups and up to six groups of delayed neutrons. The time-dependent equation is coupled to the delayed neutron precursors equation (2.8).

$$\frac{1}{\upsilon_g}\frac{\partial\phi\left(r,t\right)}{\partial t} = D_g\nabla^2\phi_g\left(r,t\right) - \Sigma_{rem,g}\phi_g\left(r,t\right) + \left(1-\beta\right)\sum_{g=1}^{2}\nu_g\Sigma_{f,g}\phi_g\left(r,t\right) + \sum_{i=1}^{6}\lambda_i C_i\left(r,t\right) \tag{2.7}$$

$$\frac{\partial}{\partial t}C_i\left(r,t\right) = \lambda_i C_i\left(r,t\right) + \beta_i\sum_{g=1}^{2}\nu_g\Sigma_{f,g}\phi_g\left(r,t\right) \tag{2.8}$$

where g is the energy group index and i the delayed neutron group index.

These equations depend on the thermal-hydraulic state of the system through the macroscopic cross-sections: Σ_f the fission cross-section, Σ_{rem} the removal cross-section including absorption and scattering, and the diffusion constants D. The cross-sections (and diffusion constants) depend on the fuel temperature (Doppler effect), the moderator density and temperature and the boron concentration in the coolant (for PWR).
The coarse-mesh method in QUABOX/CUBBOX is based on a polynomial expansion of the neutron flux in each energy group. The flux can be approximated with either quadratic approximation (QUABOX) or cubic approximation (CUBBOX). Over the years, QUABOX-CUBBOX has been validated and extensively applied to perform Light Water Reactor design calculations and safety analyses.

Started in the nineties, the coupling with ATHLET was the first TH/NP coupling developed at the GRS and is described in more details in Section 3.1.
The coupling with CTF, presented in Section 3.2, was achieved for this thesis.
Pin level simulations are possible using either a pin power reconstruction method (see Section 4.1) or direct pin-by-pin diffusion calculation with SPH correction (see Section 4.2).
QUABOX/CUBBOX was not part of the official of the EU FP7 founded NURESAFE Project. Therefore, part of the developments performed during this thesis and presented in chapters 6 and 7 were made on the DYN3D code rather than QUABOX/CUBBOX. DYN3D is presented in the next section.

DYN3D

DYN3D is a reactor simulator developed at the Helmholz Zentrum Dresden Rossendorf (HZDR) [10, 11]. It performs calculations of steady states and transients in light water reactors (LWR) with hexagonal or square fuel assemblies. The 2-group diffusion equation (see (2.7) in the previous section) is solved using a nodal expansion method.
Pin level simulations are possible using either a pin power reconstruction method or the recently developed multi-group simplified transport (SP3) capabilities.

The neutron kinetics code is connected with an internal one dimensional two-phase thermal-hydraulic model (FLOCAL) and a fuel rod model.

A coupling between DYN3D and ATHLET was developed several years ago in a GRS/HZDR collaboration.
More recently, in the scope of this thesis and the European project NURESAFE, DYN3D was coupled to both ATHLET and CTF on the Salomé platform. This work and the resulting multi-scale, multi-physics transient analyses are presented in Chapter 6 and 7.

TORT-TD

The Three Dimensional Discrete Ordinates Transport - Time Dependent (TORT-TD) code [12] is a dynamic 3D multi-group discrete ordinates (S_N) neutron transport code that solves the steady-state and time-dependent multi-group Boltzmann transport equation with an arbitrary number of delayed neutron precursor groups. It is based on the steady-state 3D transport code TORT from the DOORS package which has been developed at Oak Ridge National Laboratory.
The time dependent transport (Boltzmann) equation:

$$\left[\frac{1}{\nu_g}\frac{\partial}{\partial t} + \vec{\Omega}\cdot\vec{\nabla} + \sigma_g^{tot}(\vec{r})\right]\Psi_g\left(\vec{r},\vec{\Omega},t\right) = q_g^{ex}\left(\vec{r},\vec{\Omega},t\right) + \sum_{g'}\int_{4\pi} d\vec{\Omega}\sigma_{gg'}\left(\vec{r},\vec{\Omega},t\right)\Psi_{g'}\left(\vec{r},\vec{\Omega},t\right) + \chi_g(1-\beta)\sum_{g'}\upsilon\sigma_{fg'}(\vec{r})\Phi_g\left(\vec{r},t\right) + \sum_l \chi_{gl}^d\lambda_l c_l\left(\vec{r},t\right) \quad (2.9)$$

is solved in parallel with the precursor equation:

$$\frac{\partial}{\partial t}c_l\left(\vec{r},t\right) = \beta_l\sum_{g'}\upsilon\sigma_{fg'}(\vec{r})\Phi_g\left(\vec{r},t\right) - \lambda_l c_l\left(\vec{r},t\right) \quad (2.10)$$

where g is the energy group and l the delayed neutron precursor group.
It is solved using the (S_N) theory with quadrature order N.

Unconditional stability in transient calculations is provided using a fully-implicit time discretization scheme. The anisotropic scattering is treated in terms of a Legendre P_n cross section expansion where n denotes the scattering order. It is an accurate and efficient code for 3D analysis and has its advantages comparing to the Method of Characteristics (MOC) in terms of efficiency for direct 3D simulations especially for transient applications.
TORT-TD uses pin-homogenized multi-group nuclear cross sections. Nuclear cross sections are considered to spatially depend on fuel temperature, moderator density and boron concentration. Within the parametrized cross section libraries, the interpolation to the required feedback parameter values is done using cubic splines.

For the needs of this work, TORT-TD was coupled with CTF in a first attempt at developing a high-fidelity coupled simulation tool. This coupling is described in Section 4.3.1.

nTRACER

nTRACER is a direct whole core transport code developed at the Seoul National University (SNU) for light water reactors (and fast reactors) practical applications [13]. The expression *direct whole core transport code* refers to the fact that nTRACER, unlike TORT-TD for example, uses microscopic cross-sections libraries. It means that the usual cross-sections homogenization procedure is not necessary. Moreover, this allows for an explicit geometry representation of the fuel pellet, cladding, grid, shroud, etc.
By default, the multigroup cross section library used by nTRACER is a microscopic cross section library consisting of reaction cross sections, scattering matrices, subgroup resonance parameters, kinetics data, and etc. These cross-sections are dependent on the temperature of the material. Resonances are treated using the subgroup method.
nTRACER employs a so-called 2D+1D calculation method [4]: In the 2D plane, the Bolzmann equation is solved by using the Method Of Characteristics (MOC). To go from 2D to 3D a Coarse-Mesh Finite Difference (CMFD) formulation employing a SP3 axial nodal kernel is used.

The rationale behind this method is that the direct application of MOC to 3D core configuration requires tremendous amounts of memory and computing time associated with the excessive number of rays. The planar and axial problems are coupled through the transverse leakage. The use of a lower order 1D solution in the axial direction is justified by the fact that most heterogeneity in the core occurs in the radial direction rather than the axial.
The method of characteristics is an alternative to the discrete representation of the spatial variable, in which the transport equation is solved analytically along characteristic directions within a computational cell. The angular flux Ψ is solved along the s-axis, where this axis is oriented along the characteristic direction $\vec{\Omega}$. The 2D MOC solver in nTRACER is the Assembly Based Modular Ray Tracing (AMRT) technique [14].
The Boltzmann equation can be reduced to the characteristic form (energy dependence is omitted):

$$\frac{d\Psi}{ds} + \sigma_t \Psi = Q \tag{2.11}$$

where Q is a single total source term, including the scattering, fission and external neutron sources.
This equation has a solution of the following form:

$$\Psi(s) = \Psi_0 e^{-\sigma_t s} \int_0^s Q e^{-\sigma_t s'} ds' \tag{2.12}$$

where s is the distance along the characteristic direction, and Ψ_0 is the known angular flux at $s = 0$.

nTRACER has an internal 1D subchannel thermo-fluid and intra-pellet heat conduction calculation for incorporation of the thermal feedback effects and fuel temperature profiles.
Recently, in the scope of a collaboration between GRS and SNU, nTRACER was coupled with CTF to take advantage of CTF advanced subchannel capabilities. This work is still ongoing and the preliminary results of the coupling are presented in Section 4.3.2.

CHAPTER 3

Assembly Level Thermal-Hydraulics/Neutron Physics Coupling

3.1 The coupled code ATHLET-QUABOX/CUBBOX

In the nineties, several couplings between ATHLET and neutron physics codes were developed. After extensive tests [15], the serial integration strategy with an internal coupling approach was chosen. A standard interface which includes data exchange, convergence checking and time synchronization was developed. It is included in a set of subroutines which are compiled together with the two codes. Using this interface, ATHLET was successfully coupled with the neutronic core model QUABOX-CUBBOX and later with other 3D neutronic models, such as BIPR-VVER from Kurchatov Institute (Russia), DYN3D from HZDR (Germany), KIKO3D from KFKI Budapest (Hungary) and SADKO from RDIPE (Russia).

3.1.1 Spatial Coupling

ATHLET is a 1D system code. Its input contains information on the position of a given object only in the axial direction. On the other hand, neutron physics codes

are 3D codes. The challenge is thus to correctly transfer data from a 1D to a 3D model (and back).
This problem is solved by dividing the 3D problem into a 2D + 1D problem.
In the 1D axial direction, if the nodalization in the TH model coincides with the one in the neutronics model, the data transfer is straightforward. When the axial nodalizations of both models differ, the data is automatically linearly interpolated in the interface.
In the 2D radial plane the space coupling is achieved using a so-called "mapping scheme". The mapping scheme is a radial representation of the reactor core. A TH channel is assigned to each fuel assembly. A single TH channel can be assigned to several fuel assemblies. In such cases, all assemblies assigned to the TH channel receive the same TH feedbacks while the TH channel receives power from all contained assemblies. The "reflector" assemblies are usually mapped to bypass channels. The mapping of fuel rod objects, on the other hand, is not explicit. It is made internally in ATHLET where each declared rod object is connected to a TH channel.
For the coupling with QUABOX/CUBBOX, the mapping scheme is included in the ATHLET input. This is not necessarily the case. For instance, when coupled with DYN3D, the mapping scheme is part of the neutronic input.

Several examples of mapping schemes can be found in this thesis. Figure 5.6 in Chapter 5 is particularly interesting because it shows two mapping schemes of the same core. In the first one only 17 channels are used to model the active core. In the second mapping scheme, each fuel assembly has its own TH channel. In the remaining of the thesis, this type of scheme is referred to as a (1:1) mapping scheme
It is obvious that, a (1:1) mapping scheme provides a higher accuracy than a scheme in which several assemblies are grouped in a single TH channel. It is often used when the ATHLET model is a so-called "open-core" model wherein only the core is modeled and a set of boundary conditions are used at core inlet and exit.
In full nuclear power plant models however, because of computation cost limitations, lower resolution mapping schemes are often used in ATHLET.

Over the years, GRS has developed a set of rules to optimize the mapping scheme's accuracy versus its computation cost.

- Assemblies with control rods should be united in a separate TH channel
- Each control rod cluster should have its own TH channel, if it has different initial insertion depth in comparison with the other clusters, or if at different times it is expected to start rods movement
- Assemblies are grouped in separate THC if they are located around strong neutron flux disturbances such as:
  - Stuck rod / dropped rod
  - Assembly with different burnup (i.e. from different cycles)
  - Assembly with very different enrichment
  - Assembly with burnable absorbers (during their first cycle)
- Assemblies which cannot be controlled are grouped in one THC if they are located in one radial core ring
- For unsymmetrical disturbances at core inlet around the location of the disturbance each assembly should be assigned to a separate TH channel

Following these rules, the optimal mapping scheme is thus problem-dependent and cannot be universal. A universal mapping scheme automatically tends toward the (1:1) mapping scheme.

3.1.2 Temporal Coupling

For the time coupling, the staggered time synchronization scheme presented in Section 2.1.4 is used. ATHLET selects its time step size according to its internal algorithm. If the time step is too large for the neutronic code, the interface allows the neutron kinetic code to make several smaller time steps using linearly interpolated TH feedbacks as illustrated in Figure 3.1.

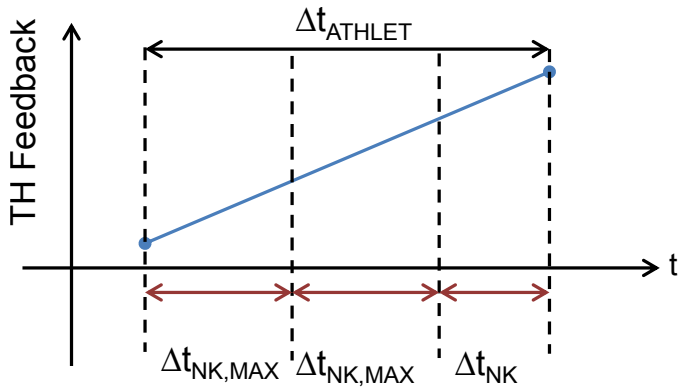

Figure 3.1: ATHLET-QUABOX/CUBBOX Time Synchronization

However some transients are driven by the neutronics such as rod movement/ejection or boron dilution. A rod movement (or a boron dilution) does not impose smaller time steps from the ATHLET algorithm. In such cases, a large time step size in ATHLET can decrease the precision of the coupled simulation.
As stated in Section 2.1.4, problems of time convergence are a known limitation of explicit time coupling schemes. Time convergence is reached when a smaller time-step does not change the result. An illustration of this phenomenon is shown on a rod ejection transient in Figure 3.2. It can be seen clearly that the first simulations with larger time step lead to a mild transient (only 50% of nominal power is reached). By decreasing the time step size, the system response changes toward a more severe transient until convergence is reached.
It is up to the user to impose an upper limit on the time step size in ATHLET which:

- On the one hand, offers sufficient accuracy for the simulated transient.
- On the other hand, keeps the computational costs to reasonable levels.

3.1.3 Control Rod Bank Coupling

When coupled with QUABOX/CUBBOX, ATHLET controls the axial position of the control rod banks. The control rod bank composition and radial position are still defined in the QUABOX/CUBBOX input. The interface checks if the

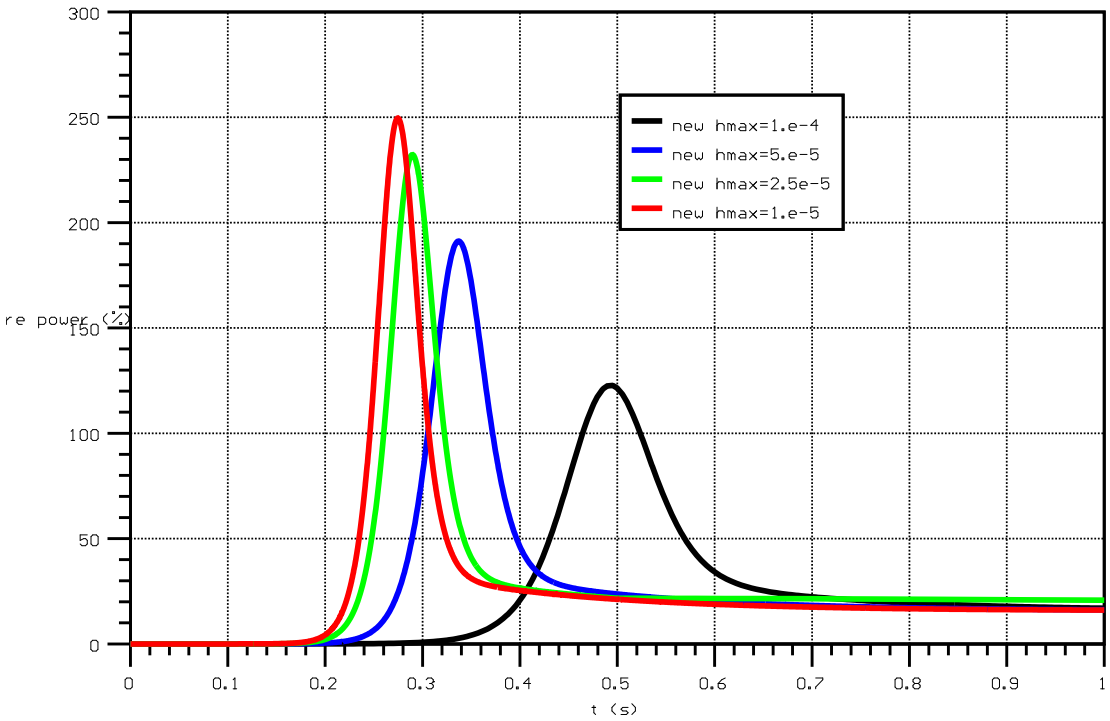

Figure 3.2: Time Convergence of a Rod Ejection Transient

number of CR group is the same in AHTLET and in QUABOX/CUBBOX and sends an error message when it is not the case. This feature is not available in most couplings with other neutron physics code. ATHLET controls the axial position of the control rod banks with GCSM signals. This option is advantageous because ATHLET is able to model very complex control logic. Moreover, GCSM signals can use TH values from the ATHLET model.

At the GRS the development of plant simulators is a common task. The power in those simulators is usually determined by ATHLET point kinetics model. But thanks to the coupled control rod control from ATHLET, 3D simulators are a possibility.

3.1.4 Application and Validation of ATHLET-QUABOX/CUBBOX

ATHLET-QUABOX/CUBBOX is the current work-horse for multi-physics simulations at the GRS.
This coupling is validated on several international benchmarks such as PWR Main Steam Line Break [16], BWR Turbine Trip [17], and PWR MOX/UO2 Core Transient Benchmark [18].
It is also validated on German power plant data but those data are proprietary and thus cannot be published.

3.2 The coupled code CTF-QUABOX/CUBBOX

In multi-scale coupling schemes, the subchannel code can either simulate the whole core or a subregion of it. For instance, in some licensing relevant cases a "hot channel" analysis might be sufficient. Nevertheless, it is desired that the subchannel code has the capability of calculating a coarse-mesh full core model. The development of the coupled system CTF-QUABOX/CUBBOX was thus a preliminary but essential step toward a fully operational multi-physics, multi-scale simulation tool. The results presented in this section were first reported in [19].

3.2.1 Description of the coupling

The experience from the ATHLET-QUABOX/CUBBOX coupling was used for the development of the CTF-QUABOX/CUBBOX coupled system. Therefore, the serial integration strategy and the internal coupling approach are also used.

Spatial coupling

For the spatial coupling, the same basic approach is used. However, the mapping scheme logic is adapted to the specific features of CTF. Two mapping schemes are used in CTF: one for the TH channels and one for the fuel rods. The main reason

is to allow easier coupling at pin level. Additional information on subchannel modeling at pin level can be found in the next chapter.

Pseudo Steady-State Convergence

Due to the absence of an actual steady-state mode in CTF, a "Pseudo Steady State" mode has been developed. This procedure is described in the following flow diagram:
The selected convergence criteria are the core average relative difference (Eq. 3.1) and the maximum nodal relative difference (Eq. 3.2) of TH parameter X. The following TH parameters are taken for X: fuel temperature, moderator density and power. Experience shows that fuel temperature is the limiting parameter.

$$mean|\frac{X_{i,n} - X_{i,n-1}}{X_{i,n}}| < \epsilon_1 \tag{3.1}$$

$$\max|\frac{X_{i,n} - X_{i,n-1}}{X_{i,n}}| < \epsilon_2 \tag{3.2}$$

where $\epsilon_1 = 0.1\%$ and $\epsilon_2 = 0.5\%$.

Time Coupling

During transient simulation, an explicit scheme is used for the time coupling during transient. The time step size is determined by CTF. Although an explicit scheme is applied, time convergence is less a problem than with ATHLET-QUABOX/CUBBOX. Indeed, in CTF the time step size selection algorithm must comply with the so-called Courant-Friedrichs-Lewy (CFL) condition in the flow direction. The one dimensional CFL condition is defined as:

$$\frac{u\Delta t}{\Delta z} \leq 1 \tag{3.3}$$

where u is the flow velocity, Δt the time step size and Δz the node size.
Since the node size in CTF is rather small (usually less than 10 cm), the time step size must be small to ensure the code stability. Usually, time step size in CTF do not exceed 10 ms.

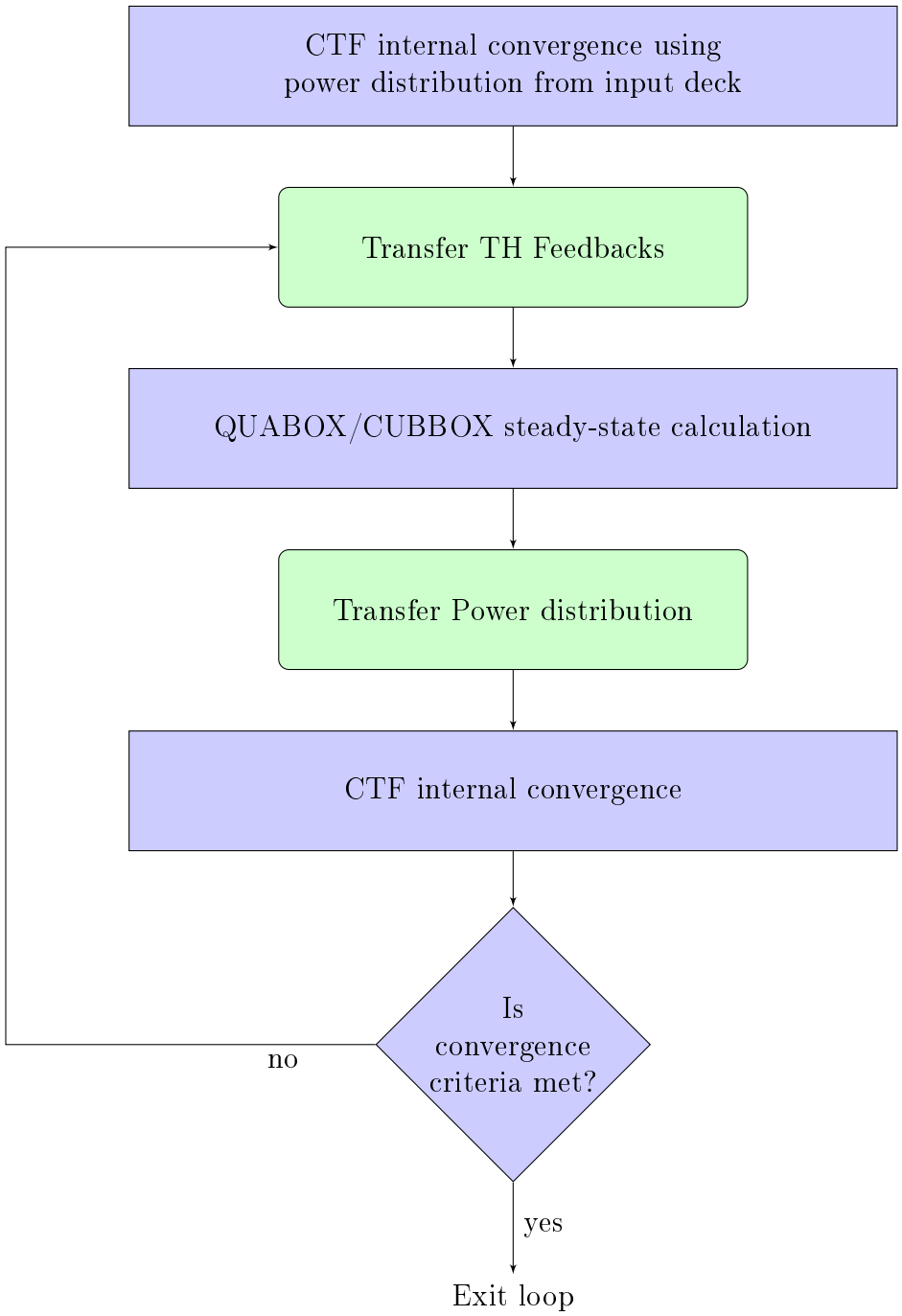

Figure 3.3: Representation of the CTF/QUABOX-CUBBOX steady-state process

3.2.2 Description of the MOX/UO2 core transient benchmark

In [19], the coupling is verified on a test case on the basis of the OECD/NEA/US-NRC MOX/UO2 core transient benchmark [20]. This benchmark is used several times as a verification tool in this thesis, therefore, it is described in more details in this section.
A specific feature of this benchmark is that the core is loaded partially with MOX fuel. Moreover, the plutonium vector of the MOX fuel has a very high-fissile fraction which is representative of weapon-grade plutonium. It is based on a four-loop Westinghouse PWR power plant and contains 193 fuel assemblies. Each fuel assembly consists of a 17x17 arrangement of square pin cells with 1.26 cm pitch. The core contains UO2 and MOX fuel assemblies with different enrichments and at several burnup levels. The enrichments are 4.2% and 4.5% for UO2 , and for the MOX fuel, the fissile Pu content is 4.0% and 4.3%. The core is at BOC and the burnup states range from 0.15 GWd/tHM to 35.0 GWd/tHM. The 2-group assembly-wise homogenized cross-sections were generated by the Benchmark organizers using the lattice and depletion code HELIOS [21]. The cross sections were calculated at three different fuel temperatures (560 K, 900 K and 1320 K), moderator densities ($661.14\,kg/m^3$, $711.87\,kg/m^3$ and $752.06\,kg/m^3$) and boron concentrations (0 ppm, 1000 ppm and 2000 ppm) in order to cover the expected range of core conditions encountered during the rod ejection transient. The moderator temperature effect is treated implicitly in the moderator density.
The libraries are provided in a parametrized tabulated format.
The core layout is represented in Figure 3.4.

3.2.3 PWR Application of CTF-QUABOX/CUBBOX

The newly developed CTF-QUABOX/CUBBOX coupled system was benchmarked against the already validated coupling between ATHLET and QUABOX/CUBBOX. For this purpose, two open-core models were compared. In both cases, a (1:1) mapping scheme with QUABOX/CUBBOX is used. The CTF model, how-

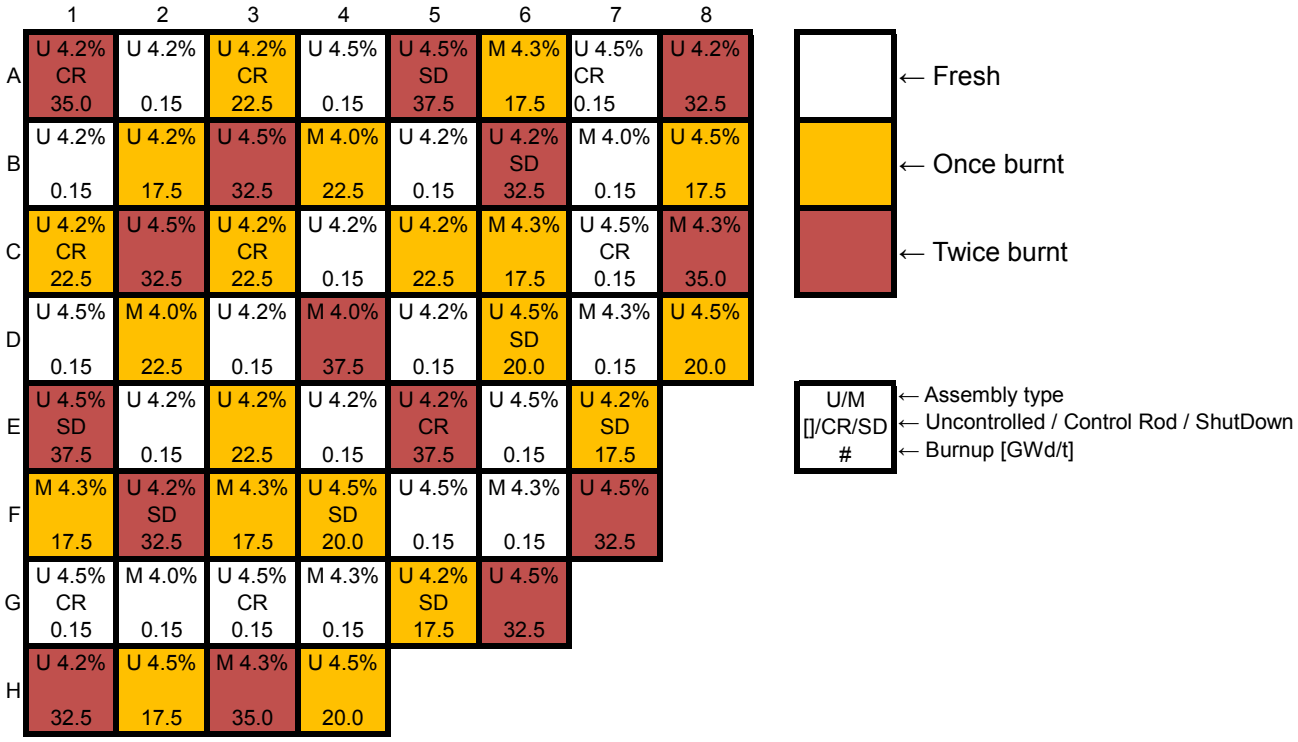

Figure 3.4: Core Layout of the "MOX/UO2 Core Transient Benchmark"

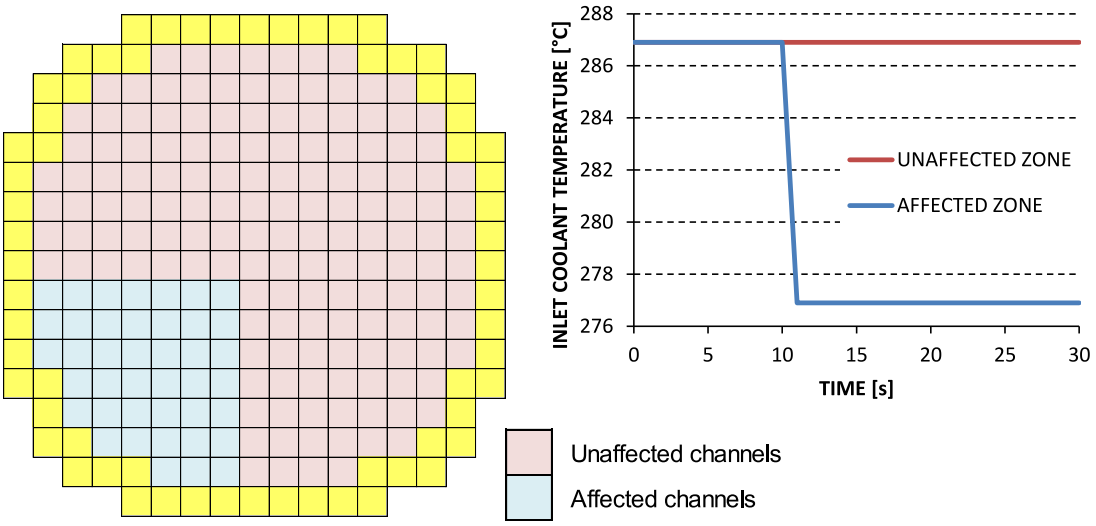

Figure 3.5: Inlet coolant temperature boundary conditions for the overcooling transient

ever, considers cross-flows between the channels. This is relevant because the transient is a non-symmetric overcooling transient. The presence of cross-flows allows the dilution of the cold water slug which can impact the power response. Overall, the steady-state results of both codes show very good agreement. For the power, the maximum radial relative difference is -0.68% in position A3. The axial relative differences are slightly bigger and range from -0.7% to 1.4%. As expected, the differences on the moderator density are almost negligible since (at equilibrium) all the generated power is transported to the fluid. For the fuel temperature, the comparison showed a systematic underestimation of the temperature in CTF. This phenomenon was investigated and is further discussed in Chapter 5.

For transient verification, a non-symmetric overcooling transient was simulated. The transient is defined as a drop of the inlet coolant temperature by 10 K within one second in one quarter of the core (41 assemblies). The inlet temperature during the transient in both the affected and the non-affected part of the core is shown in Figure 3.5. The figure also illustrates which fuel assemblies are affected by the temperature change.

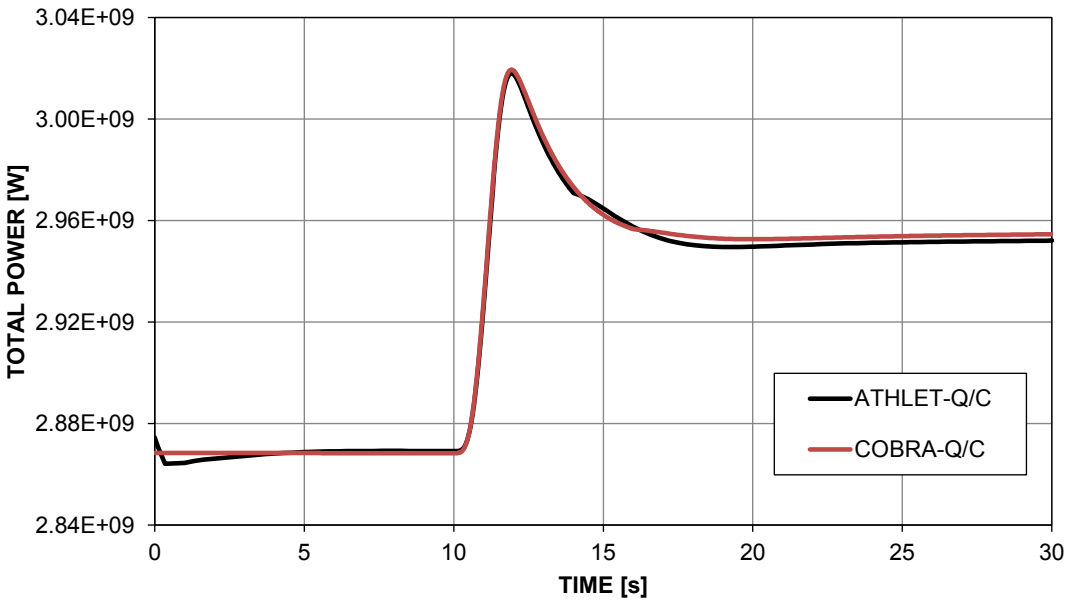

Figure 3.6: Power History During Overcooling Transient

The maximum power is reached at $t_{MAX} = 11.92s$. CTF-QUABOX/CUBBOX predicts a peak power value of 105.27% of nominal power while ATHLET-QUABOX/CUBBOX reaches 105.19%. The power at the end of the transient (t = 30s) is very close in both simulations: 103.01% of nominal power in CTF-QUABOX/CUBBOX and 102.89% in ATHLET-QUABOX/CUBBOX. The hot assembly in CTF-QUABOX/CUBBOX has a peaking factor of 1.721. In ATHLET-QUABOX/CUBBOX the hot assembly has the same location and the corresponding peaking factor is 1.714.

3.2.4 Conclusions on CTF-QUABOX/CUBBOX Coupling

The results presented in this section show that the subchannel code CTF is successfully coupled with the neutron diffusion code QUABOX/CUBBOX.
The excellent agreement found with the mature coupled system ATHLET-QUABOX/CUBBOX, for steady-state and more importantly for transient, proves the validity of the selected method.

Moreover, it shows that when equivalent models are used in ATHLET and in CTF (here a (1:1) mapping scheme), the results of the 1D system code are very close to the one of the subchannel code.

CHAPTER 4

Pin Level Thermal-Hydraulics/Neutron Physics Coupling

The assembly level coupling between ATHLET and QUABOX/CUBBOX was already mature when the work presented in this thesis started. The newly developed coupling between CTF and QUABOX/CUBBOX presented in the previous chapter already takes advantage of the advanced TH modeling offered by CTF but remains at the assembly level. The goal of this work is to develop a multi-scale simulation tool that can model the core up to a pin level in order to evaluate local safety parameters. This chapter presents several methods to simulate pin power distributions:

The first one is the reconstruction of the intranodal pin-by-pin flux distribution in diffusion codes.

The second method is the direct pin-by-pin power calculation using a diffusion code by applying the Super Homogenization (SPH) method in order to increase the accuracy.

Finally the last method is the use of neutron transport codes. Two of them were coupled to CTF in the scope of this work. Those so-called high-fidelity coupled simulation tools are presented at the end of the chapter.

4.1 Pin-by-pin flux reconstruction

The reconstruction of pin-by-pin power distribution in fuel assemblies is the state-of-the-art method for the best-estimate assessment of safety criteria at a pin level. Flux reconstruction methods are fast and usually provide enough accuracy for safety analysis.
This section describes the different approaches used in QUABOX/CUBBOX and in DYN3D.

4.1.1 In QUABOX/CUBBOX

A detailed description of QUABOX/CUBBOX flux reconstruction method can be found in [22].
The method is divided in two successive steps. The first step is the homogeneous flux interpolation. The flux interpolation method uses a smoothest analytical surface approach. The mathematics of the flux interpolation was developed in the nineties. The full mathematical demonstration of the method can be found in an internal GRS report [23]. The method takes advantage of QUABOX/CUBBOX peculiarities which is that local values of the flux are calculated in 21 points in and around a fuel assembly. In order to improve the interpolation, the mean value of the flux (nodal value) and the flux derivatives (i.e. neutron current) are used. The location of the values is represented in Figure 4.1. The 9 squares each represent a fuel assembly. The flux is reconstructed in the central one. The 21 dots are located where the local fluxes are computed in QUABOX/CUBBOX. The red ones are located where the flux derivatives are computed in QUABOX/CUBBOX. Finally, the mean value of the flux is calculated in all squares but only the one in red is used.
The interpolation method aims at finding the smoothest curve that passes through all 21 points.
Let the function (here the flux) $\phi(x, y), x \in [a, b]$ and $y \in [c, d]$ be known in N points $(x_1, y_1), ..., (x_N, y_N)$. Let the values of derivative of ϕ in the x direction be known in L points $(x'_1, y'_1), ..., (x'_L, y'_L)$. Let the values of derivative of ϕ in the y direction

be known in M points $(x''_1, y''_1), ..., (x''_M, y''_M)$. The goal is to find an interpolation function Z which is a sum of orthogonal linear independent functions R. The demonstration in [23] shows that the interpolation function has the following form:

$$Z(x,y) = \sum_{j=1}^{N} \lambda_j R(x, x_j, y, y_j) + \sum_{l=1}^{L} \lambda_{N+l} \left. \frac{\partial R(x,\zeta,y,\eta)}{\partial \zeta} \right|_{\substack{x=x_l \\ y=y_l}} + \\ + \sum_{m=1}^{M} \lambda_{N+L+m} \left. \frac{\partial R(x,\zeta,y,\eta)}{\partial \eta} \right|_{\substack{x=x_m \\ y=y_m}} + \lambda_{N+L+M+1} \int_a^b d\zeta \int_c^d d\eta R(x,\zeta,y,\eta) \quad (4.1)$$

where λ_j, $j = 1, ..., N + L + M + 1)$ are the solutions of the following set of equations:

$$Z(x_j, y_j) = \Phi(x_j, y_j) \quad j = 1, ..., N \quad (4.2)$$

$$\left. \frac{\partial Z(x,y)}{\partial x} \right|_{\substack{x=x_l \\ y=y_l}} = \phi'_x(x'_l, y'_l) \quad l = 1, ..., L \quad (4.3)$$

$$\left. \frac{\partial Z(x,y)}{\partial y} \right|_{\substack{x=x_m \\ y=y_m}} = \phi''_y(x'_m, y''_m) \quad m = 1, ..., M \quad (4.4)$$

$$\int_a^b dx \int_c^d dy Z(x,y) = \Phi \quad (4.5)$$

As in most flux reconstruction methods, the obtained pin-wise homogeneous flux can be multiplied by the so-called "form function" in order to improve the accuracy by taking into account the intra-assembly heterogeneities. The pin power P in pin r is then calculated as:

$$P(r) = \left(\kappa\Sigma_{f,1}\phi_1^{rec}(r) + \kappa\Sigma_{f,2}\phi_2^{rec}(r)\right) F(r) \quad (4.6)$$

where F is the form factor in pin r and ϕ^{rec} the reconstructed pin flux.
The form function is calculated by the spectral code during the cross-section generation process. It is defined as the relative pin power distribution as calculated by

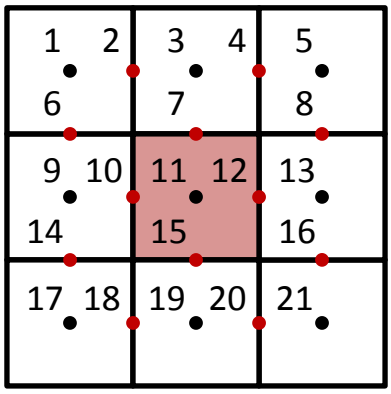

Figure 4.1: 21 point flux values in Q/C

the spectral code. It takes into account the control rod guide tubes, burnable absorbers, different enrichments, etc. The form function only depends on the average assembly burnup. No dependence on the thermal-hydraulic state is considered.

The method was tested with code to code comparison against Monte Carlo simulations, using the MCNP code version 5 [24]. An example is provided in this section. The reference solution is a 2D MCNP model of the PWR MOX/UO2 Core Transient Benchmark [20]. The core layout is the same as in Figure 3.4. The results obtained in the hot assembly with the pin reconstruction method (Figure 4.2) are compared with the MCNP reference (Figure 4.3). The hot assembly is located in position B2 (near the center of the core). The assembly relative power is respectively 1.540 in the reference MCNP model and 1.514 in QUABOX/CUBBOX. This represents a relative difference of -1.76 percent.
Looking at the normalized pin power distribution in the assembly, one can see that the deviation is within [-2.3%; +2.7%]. After the form function correction, the resulting pin peaking factor in QUABOX/CUBBOX is 1.670 compared with the 1.734 in the MCNP reference. This difference represents a -3.83 percent relative deviation, taking into account the cumulative effect with the assembly level deviation. This level of accuracy is sufficient for standard safety analysis.
In assemblies with high flux gradients (e.g. near the reflector or an inserted control rod), the accuracy can drop radically. On the same reference case it could reach up to 20 percent relative deviation. However, those assemblies usually have a low

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0.96	0.97	0.98	1.00	1.01	1.02	1.02	1.01	1.01	1.01	1.00	1.00	0.98	0.96	0.94	0.92	0.90
2	0.97	0.98	1.00	1.02	1.04	1.06	1.04	1.04	1.05	1.03	1.03	1.04	1.01	0.98	0.95	0.92	0.91
3	0.98	1.00	1.03	1.08	1.10	0.00	1.08	1.07	0.00	1.06	1.06	0.00	1.06	1.03	0.98	0.93	0.91
4	1.00	1.02	1.08	0.00	1.12	1.10	1.07	1.06	1.07	1.05	1.05	1.07	1.07	0.00	1.01	0.95	0.92
5	1.01	1.04	1.10	1.12	1.10	1.10	1.07	1.06	1.07	1.05	1.04	1.06	1.06	1.06	1.03	0.97	0.92
6	1.02	1.06	0.00	1.10	1.10	0.00	1.08	1.07	0.00	1.06	1.06	0.00	1.05	1.04	0.00	0.98	0.93
7	1.02	1.04	1.08	1.07	1.07	1.08	1.06	1.05	1.06	1.04	1.03	1.04	1.02	1.00	1.00	0.96	0.92
8	1.01	1.04	1.07	1.06	1.06	1.07	1.05	1.04	1.06	1.03	1.02	1.03	1.01	0.99	0.99	0.95	0.92
9	1.01	1.05	0.00	1.07	1.07	0.00	1.06	1.06	0.00	1.04	1.03	0.00	1.02	1.01	0.00	0.96	0.91
10	1.01	1.03	1.06	1.05	1.05	1.06	1.04	1.03	1.04	1.02	1.01	1.02	0.99	0.98	0.98	0.94	0.91
11	1.00	1.03	1.06	1.05	1.04	1.06	1.03	1.02	1.03	1.01	1.00	1.01	0.99	0.98	0.98	0.94	0.90
12	1.00	1.04	0.00	1.07	1.06	0.00	1.04	1.03	0.00	1.02	1.01	0.00	1.01	1.00	0.00	0.95	0.90
13	0.98	1.01	1.06	1.07	1.06	1.05	1.02	1.01	1.02	0.99	0.99	1.01	1.00	1.00	0.98	0.92	0.88
14	0.96	0.98	1.03	0.00	1.06	1.04	1.00	0.99	1.01	0.98	0.98	1.00	1.00	0.00	0.95	0.89	0.87
15	0.94	0.95	0.98	1.01	1.03	0.00	1.00	0.99	0.00	0.98	0.98	0.00	0.98	0.95	0.91	0.87	0.85
16	0.92	0.92	0.93	0.95	0.97	0.98	0.96	0.95	0.96	0.94	0.94	0.95	0.92	0.89	0.87	0.85	0.84
17	0.90	0.91	0.91	0.92	0.92	0.93	0.92	0.92	0.91	0.91	0.90	0.90	0.88	0.87	0.85	0.84	0.83

Figure 4.2: QUABOX/CUBBOX Reconstructed Pin Power in Hot Assembly

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	-1.8%	-1.9%	-2.3%	-2.3%	-2.1%	-2.0%	-1.9%	-1.8%	-1.5%	-1.4%	-1.5%	-1.6%	-1.8%	-1.4%	-1.5%	-0.7%	0.0%
2	-1.9%	-1.2%	-1.0%	-1.3%	-0.9%	-1.7%	-0.8%	-0.6%	-1.4%	-0.4%	-0.7%	-1.5%	-0.8%	-0.4%	-0.2%	0.2%	1.1%
3	-2.3%	-1.0%	-0.6%	-1.4%	-1.2%		-1.1%	-1.2%		-0.9%	-0.8%		-0.7%	-0.4%	0.7%	1.1%	1.6%
4	-2.3%	-1.3%	-1.4%		-0.9%	-1.1%	0.3%	0.1%	-0.7%	0.3%	0.4%	-1.0%	-0.2%		0.0%	0.9%	1.6%
5	-2.1%	-0.9%	-1.2%	-0.9%	0.4%	-0.7%	0.2%	0.3%	-0.6%	0.7%	0.6%	-0.3%	1.2%	0.1%	0.2%	1.3%	1.7%
6	-2.0%	-1.7%		-1.1%	-0.7%		-0.8%	-0.7%		-0.7%	-0.6%		-0.5%	-0.2%		0.6%	1.5%
7	-1.9%	-0.8%	-1.1%	0.3%	0.2%	-0.8%	0.2%	-0.1%	-0.7%	0.2%	0.8%	-0.5%	0.7%	0.6%	0.0%	1.2%	1.7%
8	-1.8%	-0.6%	-1.2%	0.1%	0.3%	-0.7%	-0.1%	0.2%	-0.9%	0.5%	0.6%	-0.7%	0.6%	0.5%	0.2%	1.4%	1.8%
9	-1.5%	-1.4%		-0.7%	-0.6%		-0.7%	-0.9%		-0.4%	-0.5%		-0.3%	-0.3%		0.3%	1.7%
10	-1.4%	-0.4%	-0.9%	0.3%	0.7%	-0.7%	0.2%	0.5%	-0.4%	0.3%	0.3%	-0.7%	0.6%	0.7%	-0.1%	1.3%	2.0%
11	-1.5%	-0.7%	-0.8%	0.4%	0.6%	-0.6%	0.8%	0.6%	-0.5%	0.3%	0.8%	-0.6%	0.4%	0.7%	0.2%	1.3%	2.1%
12	-1.6%	-1.5%		-1.0%	-0.3%		-0.5%	-0.7%		-0.7%	-0.6%		-0.1%	-0.2%		0.8%	2.1%
13	-1.8%	-0.8%	-0.7%	-0.2%	1.2%	-0.5%	0.7%	0.6%	-0.3%	0.6%	0.4%	-0.1%	1.3%	0.3%	0.5%	1.9%	2.2%
14	-1.4%	-0.4%	-0.4%		0.1%	-0.2%	0.6%	0.5%	-0.3%	0.7%	0.7%	-0.2%	0.3%		0.3%	1.4%	2.0%
15	-1.5%	-0.2%	0.7%	0.0%	0.2%		0.0%	0.2%		-0.1%	0.2%		0.5%	0.3%	1.7%	1.8%	2.4%
16	-0.7%	0.2%	1.1%	0.9%	1.3%	0.6%	1.2%	1.4%	0.3%	1.3%	1.3%	0.8%	1.9%	1.4%	1.8%	2.1%	2.7%
17	-0.1%	1.1%	1.6%	1.6%	1.7%	1.5%	1.7%	1.8%	1.7%	2.0%	2.1%	2.1%	2.2%	2.0%	2.4%	2.7%	2.3%

Figure 4.3: QUABOX/CUBBOX vs. Monte Carlo Difference of Relative Pin Power

power level, therefor the relative deviation is artificially increased. Furthermore, because of the low power level, the accuracy is not critical from a safety point of view.

4.1.2 In DYN3D

Similar to the one used in QUABOX/CUBBOX, the DYN3D pin power reconstruction method features two steps: a homogeneous flux reconstruction step and a heterogeneous correction by means of a form function. The so-called method of successive smoothing is applied for the reconstruction of the neutron flux in chosen assemblies [25]. The neutron flux is approximated by an analytical solution of the two-dimensional diffusion equation in each axial layer of the selected assembly. The method was first developed for Cartesian geometries and later expanded to hexagonal geometries by Hàdek in [26]. The main difference with the method used in QUABOX/CUBBOX is that no point value of the flux is generated by DYN3D. Therefore, the nodal average values of four assemblies are used to construct corner values by linear extrapolation and a subsequent smoothing step. The resulting interpolated flux functions fulfill the two-dimensional diffusion equation in the interior of the node.

4.1.3 Conclusions on the Pin-by-pin flux reconstruction methods

Both pin-by-pin flux reconstruction methods have been tested internally (at GRS for QUABOX/CUBBOX and at HZDR for DYN3D) but have not been benchmarked against each other yet.
We have shown that pin-by-pin reconstruction methods can provide a satisfactory accuracy for transient analysis of PWRs. However, those methods also have known limitations. Indeed, the accuracy is much lower near strong flux gradients (e.g. inserted control rod, reflector, etc.). For these reasons, the pin-by-pin reconstruction methods are only applicable for transients where the hot spot is not located directly next to a strong flux gradient (e.g. Applicable to Main Steam Line Break).

Moreover, in the case of coupling with a TH code, the thermal-hydraulic feedbacks are only taken into account at the assembly level.
This explains why the more expensive but more accurate methods presented in the next sections are under development.

4.2 Superhomogenization procedure for direct pin-by-pin calculation with a diffusion code

4.2.1 Description of the method

The Superhomogenization (SPH) procedure is an equivalence procedure which was developed by Kavenoky and Hébert in the eighties. The aim of this procedure is to preserve the reaction rates at the pin-cell level in a single assembly calculation. As developed in [27], this method can provide a set of SPH factors without a normalization condition. A flux-volume normalization was first described in [28] and [27], and an application of so called "generalized Selengut" conditions was developed in [29].
The SPH method is applied using both a lattice code providing the heterogeneous solution and a diffusion (or transport) code to provide the homogeneous solution. Figure 4.4 illustrates the heterogeneous and homogeneous problems at the pin cell level. An advantage of the SPH method is that no modification of the lattice or the diffusion code is needed to compute the SPH factors: a new set of SPH corrected cross sections can be processed using the output data given by both codes. Moreover, the heterogeneous only needs to be computed once, thus keeping the computational costs to reasonable levels.

The SPH method is an iterative method. The iterative process is shown in Figure 4.5.
The SPH factors μ are defined in each pin cell i and each energy group G as:

$$\mu_i^G = \frac{\phi_i^G}{\tilde{\phi}_i^G} \tag{4.7}$$

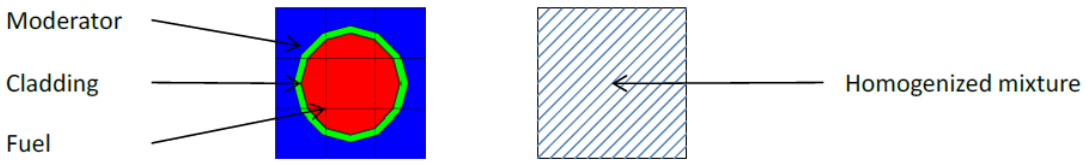

Figure 4.4: Representation of the heterogeneous and homogeneous problems at the pin cell level

where ϕ is the heterogeneous pin integrated flux and $\tilde{\phi}$ the homogeneous pin flux. The cross sections and the diffusion coefficients are multiplied with the SPH factors.

$$\sigma_{i,x}^{G,n+1} = \mu_i^G \sigma_{i,x}^{G,n} \tag{4.8}$$

$$D_{i,x}^{G,n+1} = \mu_i^G D_{i,x}^{G,n} \tag{4.9}$$

Therefore, new SPH factors are computed and the procedure is repeated until the SPH factors are converged, i.e.

$$max_{i,G} \left| \frac{\mu_i^{G,n} - \mu_i^{G,n-1}}{\mu_i^{G,n}} \right| < \epsilon \tag{4.10}$$

where n is the number of the iteration and $\epsilon = 10^{-2}$

4.2.2 Results of the method

In GRS, the SPH method was first implemented by Klein in 2010 [30]. The homogeneous solution is provided by the diffusion code QUABOX/CUBBOX. The heterogeneous solution is calculated with the spectral code NEWT from the SCALE package.

The applicability of the SPH method for QUABOX/CUBBOX pin-by-pin simulation was first tested on the C5G7 Benchmark. This benchmark is challenging because of its high heterogeneity (UOX and MOX). It is also particularly suitable for pin-by-pin calculations since the system is small (2x2 arrangement with and without reflector) and no burnup is considered. Later the method was tested on

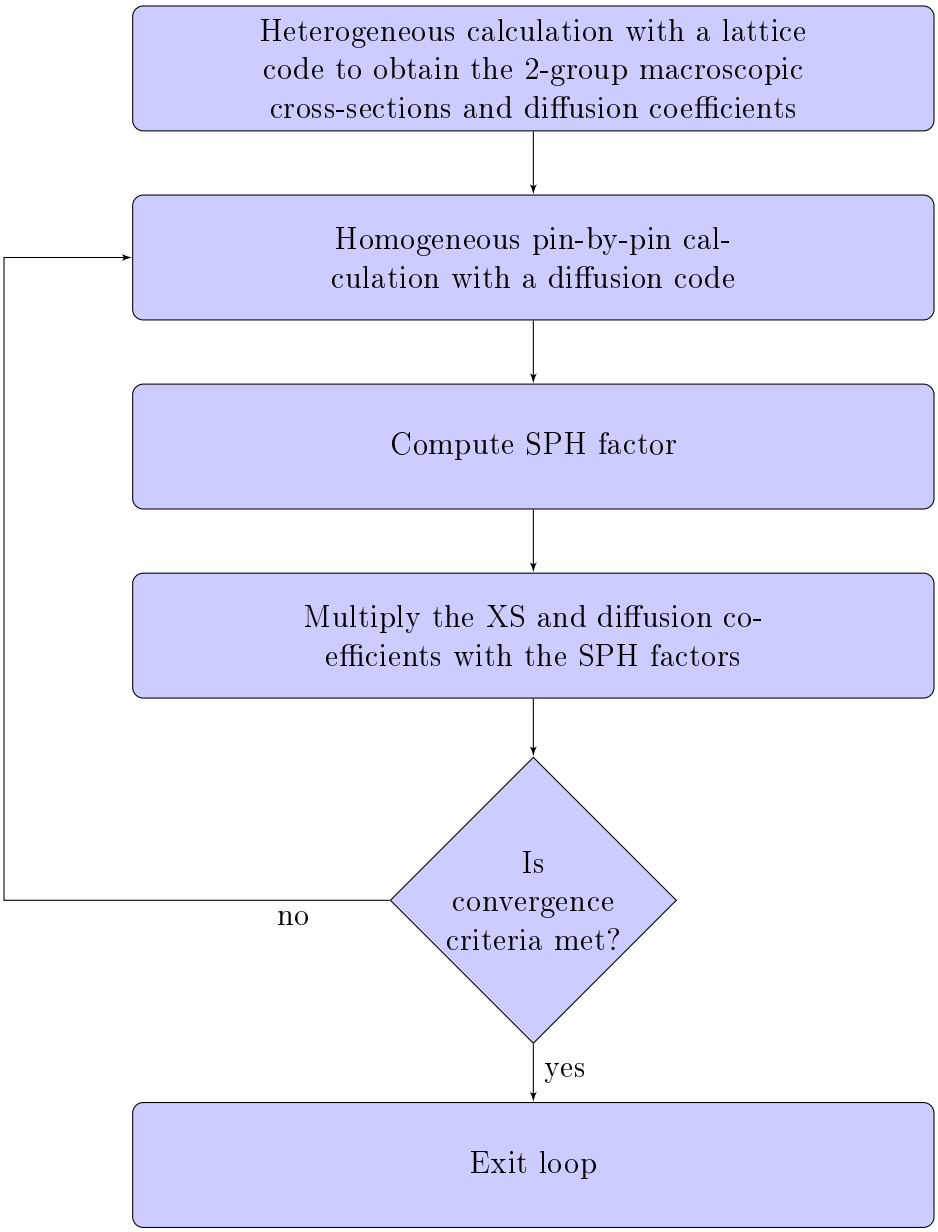

Figure 4.5: Representation of the SPH iterative process

the same 2D version of the PWR MOX/UO2 Core Transient Benchmark used in Section 4.1.1.
The results obtained in position B2 are represented in Figure 4.6. The maximum pin power deviation in the assembly is 3.2 percents in the upper left corner. The deviation in the hot pin is only -1.6 percents which is an improvement compared to the -3.83 percents obtained with the flux reconstruction method.

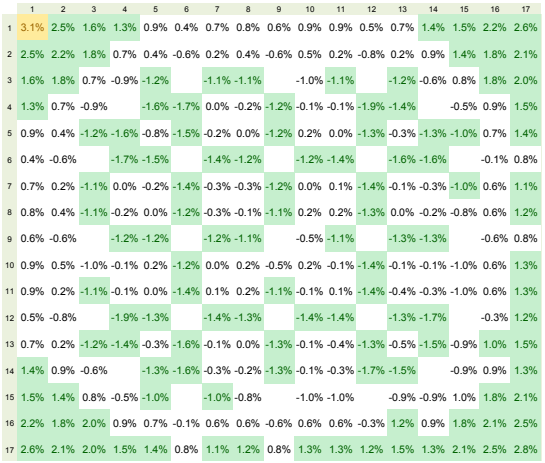

Figure 4.6: QUABOX/CUBBOX SPH vs. Monte Carlo Difference of Pin Power

Overall excellent accuracy was achieved for 2D models using NEWT for the heterogeneous solution. Indeed, the RMS for the whole core in under 2 percent the maximum deviation is below 6 percent.

More recently the implementation of the SPH method was extended to the Monte Carlo code SERPENT for the generation of cross-section and SPH factors.
The results obtained on 2D models confirm the applicability of the method.
With SERPENT, the SPH method was also applied on 3D models. For this purpose the C5G7 Benchmark was again used.
The results show that similar accuracy than for 2D can be achieved except for the axial layers next to the reflectors where the relative deviation reaches 10%. This can be explained by the generation of the SPH coefficients which is made on a 2D model, so the reflector is only considered in the radial direction.

4.2.3 Conclusions on the SPH method

The SPH procedure for direct pin-by-pin diffusion calculation is promising and provides a middle ground between pin power reconstruction (fast) methods and transport calculation (high accuracy). Further studies are needed in order to assess the dependence of the SPH factors on the TH state. If this effect is negligible then only one set of SPH factors for the nominal TH state will be needed. However, if the TH state is relevant, this means that SPH factors need to be calculated for each TH state of a cross-section library. Future developments of the method include:

- Generation of burnup and TH parametrized cross-section libraries, including parametrized SPH factors.
- Coupling with CTF at the pin level (steady-state and transient simulations)
- Finally full core coupled calculation

4.3 High-fidelity simulation tools

High-fidelity simulation tools refer to systems in which the thermal-hydraulics and the neutron physics are both described at a pin level. In addition, the individual models should provide a high accuracy. For the TH model, a 3D subchannel code or a CDF code is typically applied. The neutronic model solves the neutron transport equation without approximation, either with deterministic (S_N and MOC) or with probabilistic (Monte Carlo) methods.
In the scope of this thesis, couplings between CTF and two transport codes were developed.

- First, a coupling with the in-house discrete ordinates S_N code TORT-TD.
- More recently, a coupling with the "Method Of Characteristics" (MOC) code nTRACER.

4.3.1 The coupled code CTF/TORT-TD

Description of the coupling

TORT-TD, described in Section 2.2.2 is a discrete ordinates (S_N) neutron transport code. The CTF-QUABOX/CUBBOX coupling interface was used as a basis for the development of the CTF/TORT-TD coupled system. The same TH feedback parameters as for QUABOX/CUBBOX are transferred to TORT-TD, namely: fuel temperature, moderator density and boron concentration. The internal coupling approach with an explicit time coupling is used. The same "Pseudo Steady State" convergence procedure developed for CTF-QUABOX/CUBBOX is applied.

The spatial mapping schemes for TORT-TD and CTF are performed in both axial direction and radial plane. In the axial direction an automated capability for interpolation between the meshing of TORT-TD and CTF was implemented and it provides a framework for the exchange of parameters between the two codes. In the radial plane, two options are available for coupling TORT-TD with CTF. TORT-TD always performs pin-by-pin calculations while CTF can perform either assembly-wise or pin-wise calculations.

When using the assembly-wise model, all pins in a same assembly receive the same TH feedback in TORT-TD, while CTF uses the sum of all pin power within this assembly.

When the CTF model is made at a pin level, the question of the TH modeling arises. Indeed, the natural modeling for sub-channels at the pin level is the so-called "coolant-centered" representation. However, the neutron physics codes all use a "fuel-centered" approach. For a better understanding, these two approaches are illustrated in Figure 4.7

Therefore, those two options for TH modeling are available. Both have advantages and drawbacks:

- T/H "rod-centered" approach. In that case, the TH model is not optimal but the data exchange is facilitated since the same mesh as the neutron physics one is applied

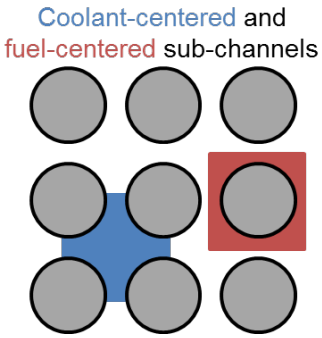

Figure 4.7: Coolant Centered vs. Pin centered Modeling

- T/H "coolant centered" approach. In that case the TH description is more accurate but approximations are needed to correctly transfer data to the neutron physics code (e.g. averaging). The use of approximations somehow neutralizes the advantage of using the more accurate T/H model.

For the coupling with TORT-TD, only the fuel-centered option is applied. It is important to note that since the cross-section in TORT-TD are homogenized, no azimuthal dependency is considered.

The coupled code CTF/TORT-TD has been verified for PWR [31, 32] and BWR [33] applications. The findings are summarized in the next two sections.

PWR Application

This section presents the results obtained on high fidelity PWR simulations with the coupled code CTF/TORT-TD.
Both steady-state and transient capabilities were tested.
The PWR test cases are based on the MOX/UO2 Core Transient Benchmark [20]. The 8-group pin-wise homogenized cross-section libraries were generated with fuel assembly infinite lattice burnup calculations using the cell and depletion code HELIOS [21]. The cross sections have been calculated in P_1 Legendre expansion order. The libraries are parametrized using the same supporting points for fuel temperature, moderator density and boron concentration as defined in the benchmark specifications.

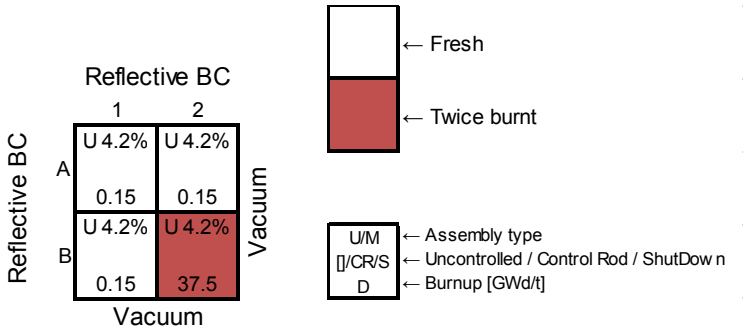

Figure 4.8: Description of the Minicore

Steady state simulations of a quarter core model were performed to check the capability of the system on a full scale simulation.
The transient capabilities were tested on a 2x2 fuel assembly mini-core which is represented in Figure 4.8.
The selected transient is a control rod ejection. The initial power is 22.9 MW. The control rod in position B2 is ejected within 0.1 s. The rod weighs 1.64$. Constant time steps of 2ms are used for the simulation.
Two CTF models are compared: an assembly-wise model (4 channels) and a pin-wise model (1156 fuel-centered channels). In both cases, the same axial nodalization is used in both codes: 50 equidistant meshes of 7.82 cm.
The power history during the transient is represented in Figure 4.9.
During the rod ejection transient, with the assembly-wise TH model the maximum power is 4081 MW at $t = 1.086s$. With the pin-wise TH model the maximum power is 3776 MW at $t = 1.082s$, which is 8 % lower. The assembly-wise TH feedbacks model is therefore conservative, which is the expected behavior. This behavior is explained by the Doppler temperature feedback. It is stronger in the more detailed model because the pins, especially in the assembly next to the ejected control rod, have a higher temperature increase than in the coarse model (+30K).

BWR Application

This section presents the results obtained on high fidelity BWR simulations with the coupled code CTF/TORT-TD. For the needs of BWR simulations, the stan-

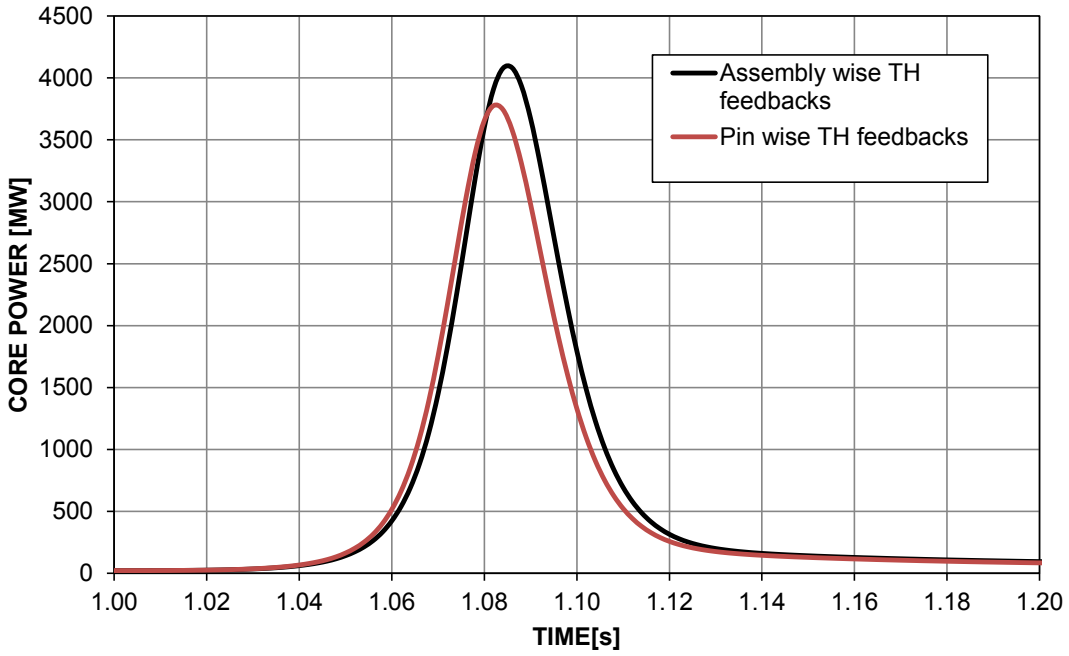

Figure 4.9: Minicore - Rod Ejection - Power History

dard "Pseudo Steady State" convergence procedure had to be adapted. Indeed, with the original version, oscillations of the system between two states could be observed (see Figure 4.10).
In order to force the convergence, a weighted value between the last two TORT-TD iterations is used in CTF (Eq. 4.11), instead of using the latest power distribution from TORT-TD. A weight repartition of 0.5/0.5 proved to be working reliably.

$$Pow_{n+1,CTF}(k) = 0.5Pow_{n,TORT-TD}(k) + 0.5Pow_{n+1,TORT-TD}(k) \qquad (4.11)$$

where n is the steady-state iteration number and k the node number.
This technique was successfully implemented and the coupled system is now able to converge toward a single, physically correct, power profile (see Figure 4.12).

In order to test the coupled system for two-phase flow, a single ATRIUM-10 assembly from the "Physics of Plutonium Fuels BWR MOX Benchmark" [34] was selected. The ATRIUM-10 bundle is a modern BWR assembly. It contains 91

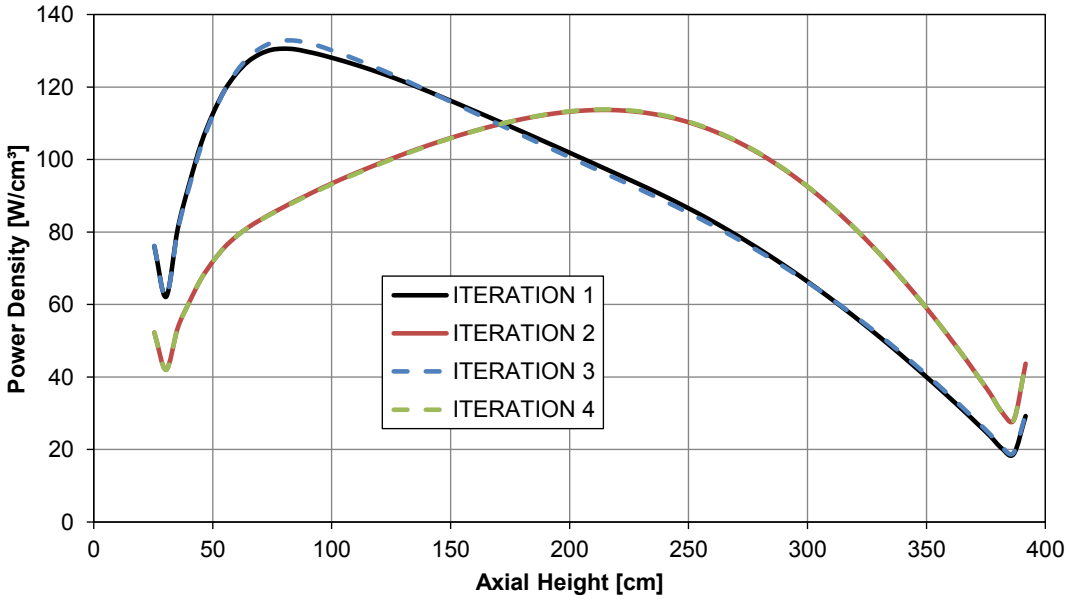

Figure 4.10: Oscillation of Axial Power Distribution During TORT-TD/CTF Steady-State Convergence

fuel rods arranged in a 10-by-10 rod array, with eight partial-length rods and nine array positions occupied by the inner water column.

The pin-cell homogenized 8 energy group cross-sections were once more generated using the lattice code HELIOS. The chosen fuel temperature and moderator density branching points cover a broad spectrum conditions allowing calculations from cold state (13 values from 20 °C to 2600 °C) to severe accident conditions (13 values from 1000 kg/m^3 to 10 kg/m^3). These extreme conditions were not used in the present study but kept for future work. It is important to note that the moderator temperature and the void fraction are implicitly included in the moderator density feedback by considering a constant pressure.

Figure 4.11 shows the ATRIUM model in CTF (left hand side) and in TORT-TD (right hand side). In the CTF model, five types of rod-centered subchannels are defined: normal, box side, box corner, water-rod side, and water-rod corner, as well as one channel each representing the bypass and water rod interior. In the TORT-TD model, each color represents a material with its own cross-sections. All TORT-TD meshes within thick black lines use the same TH feedback values.

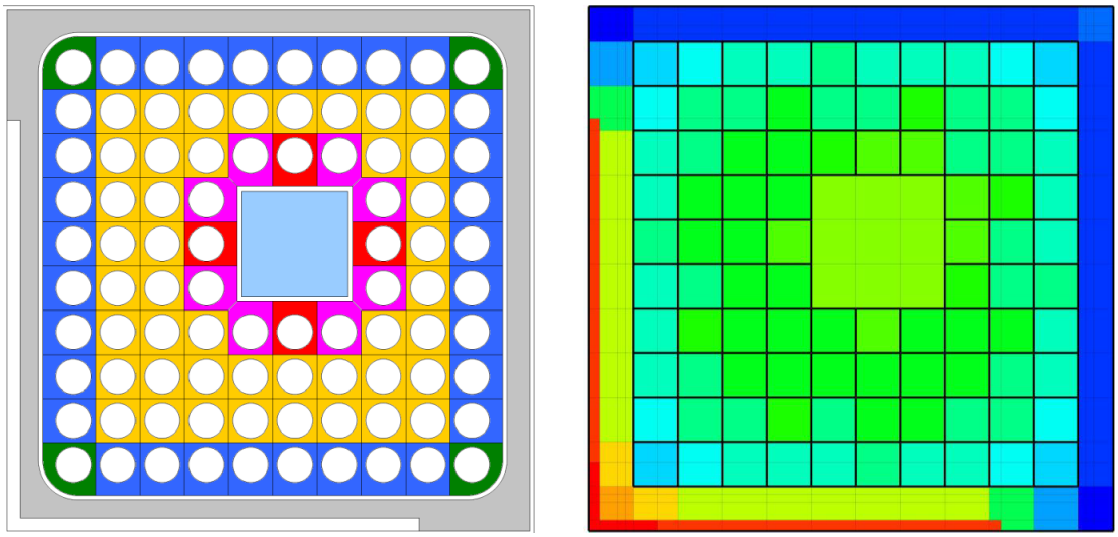

Figure 4.11: BWR ATRIUM Model - Left: CTF Model - Right:TORT-TD Model

In the axial direction, the active height of the fuel assembly (371.2 cm) is discretized using 75 equidistant nodes in both codes.

Several coupled calculations were performed using the previously described BWR assembly model. Two steady-state cases using a 5 MW assembly power were selected:

- One without control blade insertion
- One with the control blades inserted 100 cm in the active part of the core (122 cm including the reflector).

The converged axial power profile for both cases is represented in Figure 4.12. The effect of the control blades can clearly be observed on the axial power distribution. All of the obtained results show the expected tendencies, so confirming the BWR steady-state modeling capability of CTF/TORT-TD.

The system transient capabilities for BWR were tested on the same model. The selected transient was an inlet flow reduction. Starting from the uncontrolled case, after 1s of null-transient, the inlet mass-flow is reduced by approximately 10% within one second (see Figure 4.13). The resulting decrease of the assembly

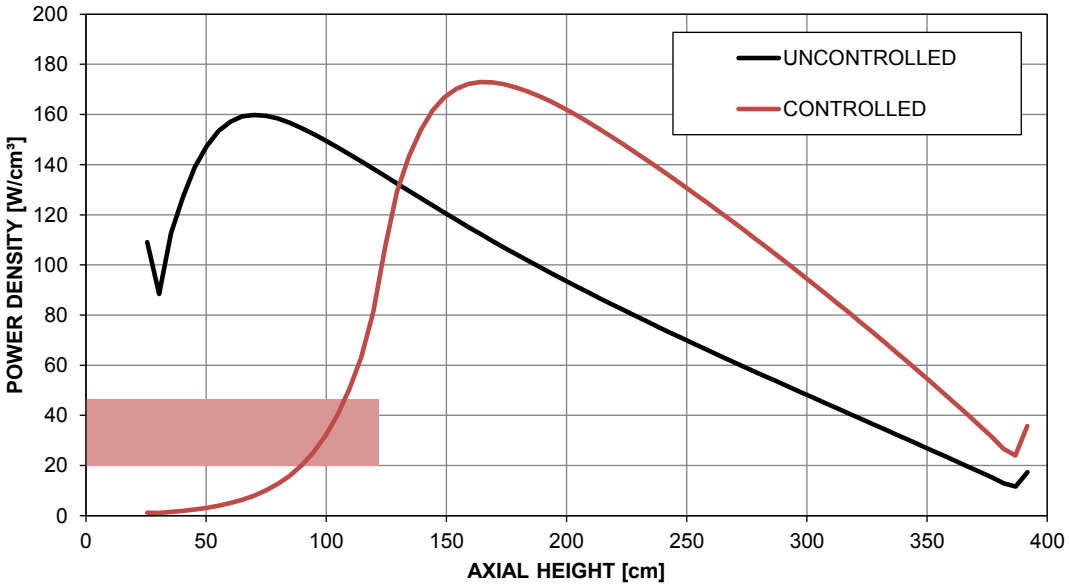

Figure 4.12: BWR Axial Power Distribution - Comparison Between the Case without Control Blade and the one with Partially Inserted Blade (100 cm)

power can be observed in Figure 4.13. It is consistent with the expected physical behavior and shows the applicability of the CTF/TORT-TD coupled code for BWR transient application.

Conclusions

The coupled code CTF/TORT-TD is a high-fidelity simulation tool which is able to predict local core parameters at a pin level. However, the computational costs of several days/weeks for a few seconds of transient simulation are very high compared to the few hours/days for several hundreds of seconds of transient with CTF-QUABOX/CUBBOX. Those computational costs cannot be easily reduced due to the impossibility to parallelize TORT-TD. This prevents its use for day to day applications and limit its applicability to reference calculations of short transients.

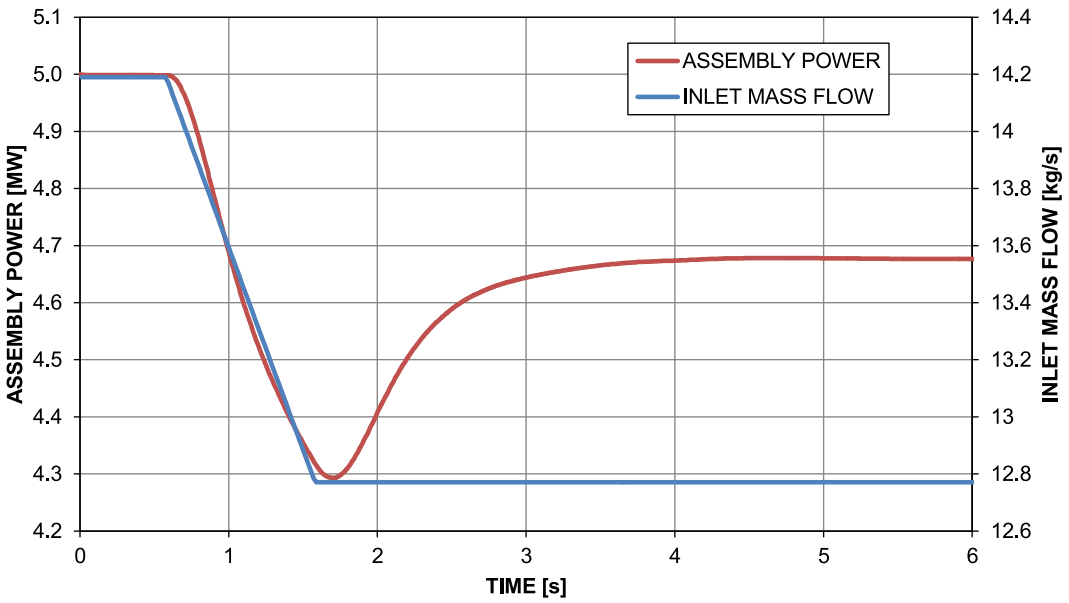

Figure 4.13: CTF/TORT-TD Results of an Inlet Flow Reduction Transient

4.3.2 The coupled code CTF/nTRACER

This high-fidelity coupled system was developed in collaboration with the Seoul National University. The work presented in this section was published in a joint-paper [3] where it is described in more details.

Description of the coupling

nTRACER is a *direct whole core transport code* which solves the transport equation using the Method Of Characteristics (see Section 2.2.2).
The objective of the coupling with CTF is to extend the TH simulation capabilities of nTRACER to boiling conditions as well as add some features described below. In the existing coupling between nTRACER and the subchannel code MATRA, the fuel temperature is calculated with a simplified model within nTRACER which uses the fluid data from MATRA. In the coupling with CTF, the fuel temperature data is taken from CTF instead.

In its original version, nTRACER was only able to use a homogeneous boron concentration in the whole system. New arrays are now added to allow the modeling of non-homogeneous boron concentration and thus taking advantage of CTF's boron tracking model. This feature is needed to simulate boron dilution (or boration) transients.

The employed coupling strategy is the serial integration in which CTF becomes a subroutine of nTRACER and is called instead of the native internal TH model. Since CTF handles all TH simulations, including the fuel temperature, this is an internal coupling. It is interesting to note that this is the only example (in this thesis) of an TH/NK coupling where the neutronic code is the master program. Within the CASL project, a new coupling interface was developed for CTF at the Oak Ridge National Laboratory and the NCSU. This interface allows to automatically transforming the "coolant-centered" values in "fuel-centered" ones.

Typically with a serial coupling approach, a mapping scheme is explicitly written either in one of the code inputs or in a separate "coupling input". Several examples of such mapping schemes are described in Chapter 6. In the case of the CTF/nTRACER coupling, the mapping scheme is written neither in nTRACER nor in CTF input. Here, the CTF input deck is generated directly by nTRACER by a dedicated subroutine which calls the CTF preprocessor. This ensures that the nTRACER and the CTF models are equivalent and that the in-memory array organization is compatible for a proper data exchange.

For steady-state calculations, the following algorithm is performed (Figure 4.14). Steps 1 to 3 are only performed once and step 8 is only once after the last call of the TH solver. The convergence criterion is based on the Doppler broadening temperature change from one iteration to the next. The aim value is $5.0 \cdot 10^{-}4$. This convergence criterion is the same as in the coupling with MATRA since it has already been proven effective [4].

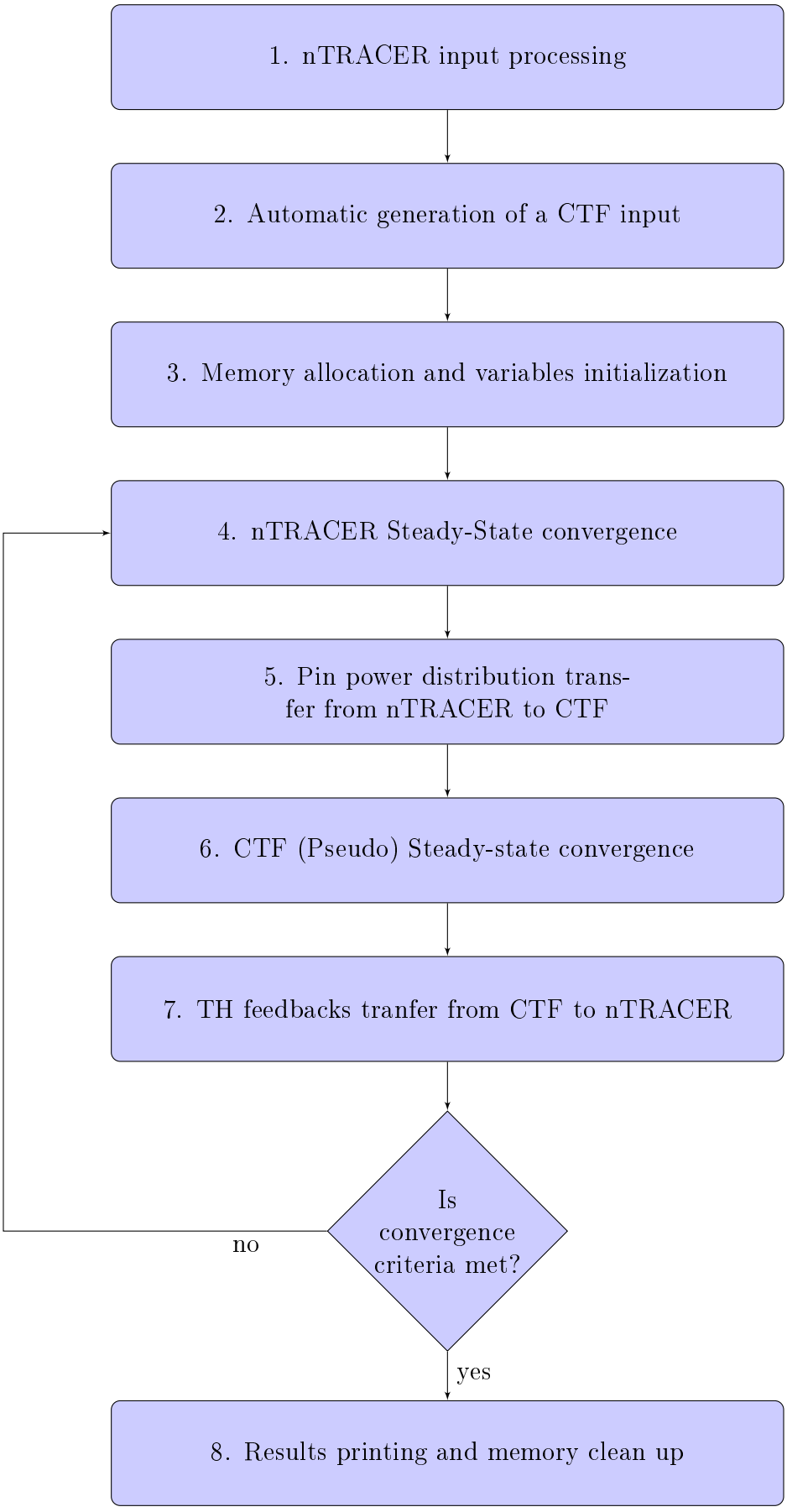

Figure 4.14: nTRACER/CTF Steady-state calculation process

	nTRACER	MATRA	CTF
k-eff	1.13849 (+29)	1.13827 (+7)	1.13820
Peaking factor (pin)	1.320	1.320	1.318
Rel. power difference	-0.61%	-0.17%	Ref.
Avg. outlet temp	328.0 °C	327.6°C	327.7°C
Max. outlet temp.	336.9°C	334.7°C	335.7°C
Min. outlet temp	321.0°C	319.2°C	319.0°C
Max. fuel temp.	1601.7°C	1588.5°C	1559.6°C

Table 4.1: Comparison of Some Relevant Parameters for nTRACER Internal TH Model, MATRA/nTRACER and CTF/nTRACER

$$RMS_{xy}\left(max_z\left|abs\left|1-\sqrt{\frac{TF_{old,xyz}}{TF_{new,xyz}}}\right|\right|\right) \tag{4.12}$$

where TF is the radially averaged fuel temperature at position xyz (at the current and the previous step).

First PWR applications

This section presents the results obtained on PWR high-fidelity simulations with the coupled code CTF/nTRACER.

The CTF/nTRACER coupled calculation capability was verified on a suite of 2x2 assembly colorset problems. nTRACER used pre-collapsed 47-group microscopic libraries. One of them, a nominal steady-state, is described here. Figure 4.15 shows that more realistic flow fields are obtained with the new coupled code than with the internal TH model. Using the internal TH model, unphysical discontinuities in the coolant temperature distribution are observed especially at the assembly interfaces. With the CTF model, which includes channel cross-connection, the coolant temperature distribution is smoothed. Table 4.1 shows a comparison of some relevant parameters for nTRACER internal TH model, MATRA/nTRACER and CTF/nTRACER. The more realistic TH modeling in MATRA and in CTF results in very slight differences in power distributions, k-eff, and other relevant output parameters compared to the nTRACER internal TH model. The agreement between the MATRA and the CTF coupled systems is excellent.

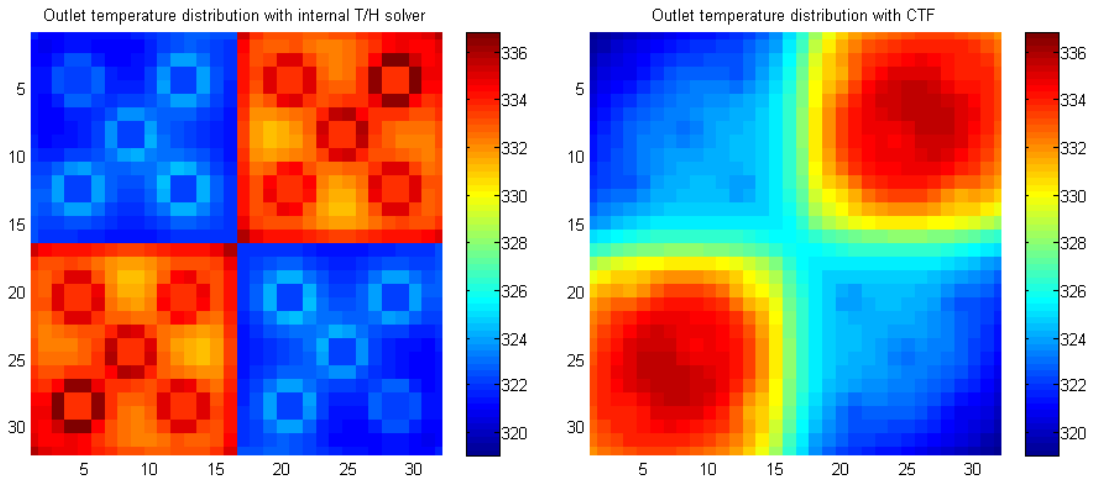

Figure 4.15: Comparison of Core Exit Temperature - Left: nTRACER Internal TH Model - Right: CTF/nTRACER

Conclusions and future work

The coupling of nTRACER with CTF was successfully achieved for PWR systems. One objective for this coupling is to take advantage of CTF 2-phase flow capabilities to allow for more accurate simulation of BWR systems. For this purpose, the CTF preprocessor will need to be extended to BWR bundles in the future. This option is currently under development at the Oak Ridge National Laboratory.

The use of CTF as TH solver increases computation costs compared to the internal TH module of nTRACER. In Table 4.2 the computation time for 16 by 16 pins single assembly problem with the native TH module and with CTF are compared. In the first case, the TH part of the problem is a negligible fraction of the total simulation time. When using CTF, it becomes significant (18.21% of the total) although the system considered is very small. The objective is to use this high-fidelity tool for the simulation of full core problems. Therefore, the evaluation and improvement of the parallel computing capability of the nTRACER/CTF is a priority. It will be necessary for the simulation of full core problems. These improvements are currently studied at the Seoul National University.

Finally, the aim of this joint project is to develop a tool able to simulate transients for safety analysis. However, nTRACER does not currently have transient

	nTRACER	CTF
Neutronics	1055 s (5 iterations)	1495 s (8 iterations)
TH	0.28 s (0.07% of total)	328 s (18% of total)
Total	1055 s	1823 s

Table 4.2: Comparison of computation costs for nTRACER Internal TH Model and CTF/nTRACER

capabilities. This feature is currently under development at the Seoul National University.

4.3.3 Conclusions on High-Fidelity Simulation Tools

High-Fidelity systems are currently under development worldwide, the most famous of such projects is the American CASL initiative. Within this framework CTF was successfully coupled to the MPACT transport code (Michigan PArallel Characteristics Transport Code) [5]. However, the computational resources available for the CASL project are much higher than what is typically accessible for daily safety analysis applications. Moreover, this coupling is currently limited to steady-state applications.
Similarly, the two coupled systems developed in the scope of this work require very high computation costs. Therefore their applicability is currently limited to reference steady-state and short transient (only for CTF/TORT-TD) simulations.

CHAPTER 5

Thermal-Hydraulics/Thermal-Hydraulics Coupling

In multi-scale coupling schemes, in addition to TH/NP coupling, a system code and a sub-channel code usually need to be coupled. The GRS multi-physics, multi-scale project includes the TH/TH coupling of the system code ATHLET with the subchannel code CTF.

5.1 Offline Coupling of ATHLET with CTF

A first attempt of a coupling between ATHLET with CTF was realized shortly after the implementation of CTF at GRS.
A pin-by-pin hot channel analysis was performed for a PWR loss of feedwater under ATWS conditions. The transient is driven by the secondary side, thus the entire NPP must be modeled. In the core, the most limiting safety criterion is the DNBR which should be computed as a pin-wise level. For this simulation, the NPP was modeled with ATHLET while the core was modeled by ATHLET/QUABOX-CUBBOX coupled system. The ATHLET core model features a 'hot pin' channel for the minimum DNBR simulation. The power in this pin is computed using the hot assembly power multiplied by a hot pin factor.

Figure 5.1: Fuel centered hot pin channel

The objective of the study was to check the applicability of the offline coupling between ATHLET and CTF for hot pin analysis, as well as to compare ATHLET and CTF on a simple model. In CTF the 'hot pin' model is reproduced with a fuel centered representation (see Figure 5.1). The following time-dependent output parameters were extracted from the ATHLET/QUABOX-CUBBOX simulation: inlet mass flow and temperature, core exit pressure, and axial power distribution. The information is used in the CTF model as boundary conditions.
Figure 5.2 shows the channel inlet pressure as calculated in CTF and compared to the ATHLET value. A very good agreement is found between the two codes. This shows that the boundary conditions were correctly extracted from ATHLET/ QUABOX-CUBBOX and implemented in the CTF model.
Figure 5.3 shows the resulting DNBR simulation. Both codes use the W3 Critical Heat Flux (CHF) correlation. The resulting minimum DNBR is in good agreement in both codes.

The results showed that a coupling between ATHLET and CTF is possible. Moreover, the results of ATHLET and CTF are very close when using a simple (1D) model.
The solution presented in this section is primitive and was only a first step toward the fully coupled system ATHLET/CTF/QUABOX-CUBBOX.

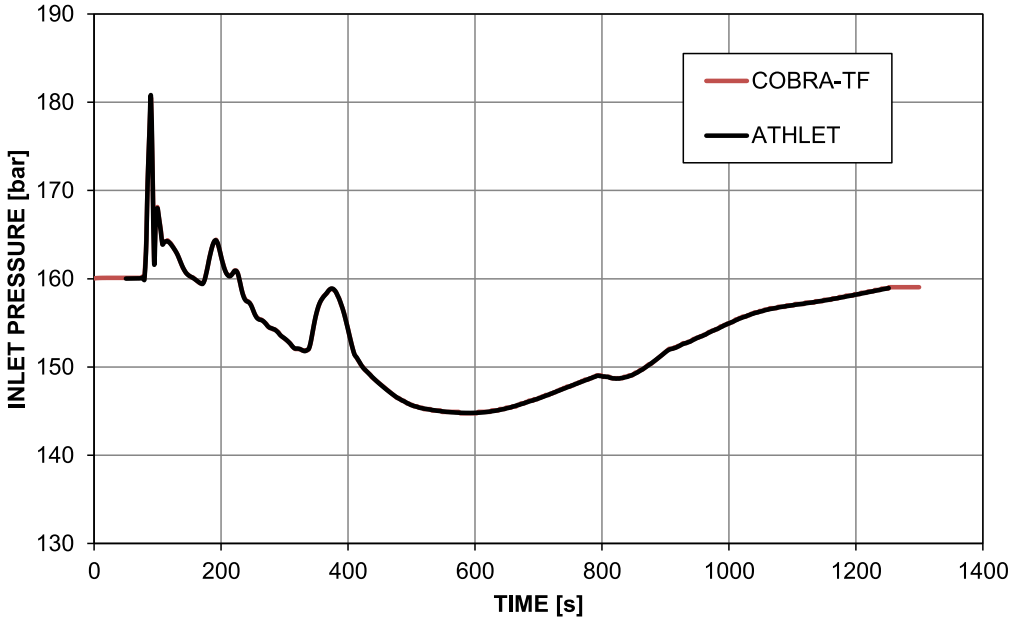

Figure 5.2: Hot pin channel inlet pressure during an ATWS transient - Comparison of ATHLET and CTF

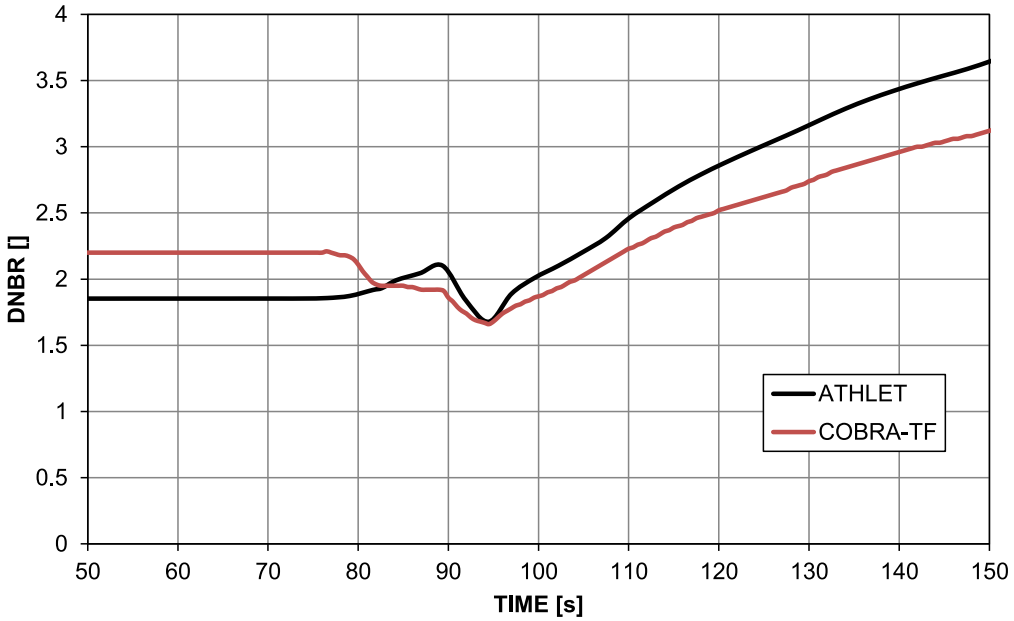

Figure 5.3: Hot pin DNBR analysis during an ATWS transient - Comparison of ATHLET and CTF

5.2 Parallel Coupling of ATHLET with CTF

For the online coupling of ATHLET with COBRA-TF, the parallel coupling method described in Section 2.1.3 is used.
The most standard way to implement a TH/TH coupling uses the external coupling method. Several examples can be found in the literature, such as COBRA/TRAC-A [35] or more recently COBRA-TF/RELAP5-3D [36]. At the GRS, the external coupling approach was applied for the coupling of ATHLET with CFD codes (Ansys CFX [37] and OpenFOAM [38]).
However, for transient analysis, ATHLET and its coupled version with QUABOX/CUBBOX are still the standard codes at the GRS. This means that for each new system to be modeled in the foreseeable future, the development of an ATHLET core model will always be a priority. Therefore, it is important that the TH coupling with CTF, which is made at the core level, does not change the ATHLET input.

With the parallel coupling approach, both thermal-hydraulics codes model the core region usually with different level of precision. This means that the ATHLET model retains its core nodalization. The system code provides core inlet (mass flow rates, fluid enthalpy, and boron concentration) and core exit boundary conditions (pressures) to the subchannel code.
Figure 5.4 shows an example of a CTF coupled input with ATHLET. Two new types of boundary conditions were defined: coupled core inlet (ISPC = 6) and coupled core exit (ISPC = 7). If these boundary condition types are used, additional inputs are needed:

- The name of the ATHLET channel
- The location of the coupling interface: I = channel inlet, O = channel exit, B = Both I and O
- The number 'nc' of CTF channels connected to this ATHLET channel
- The list of 'nc' CTF channel IDs

```
* Inlet b.c. ------------------------------------------------------------
*IBD1 IBD2 ISPC N1FN N2FN N3FN   BCVALUE1   BCVALUE2   BCVALUE3 INITGAS
    1    1    6    1    0    0        0.0   1314.066        0.0        1
    2    1    6    1    0    0        0.0   1314.066        0.0        1
    3    1    6    1    0    0        0.0   1314.066        0.0        1
    4    1    6    1    0    0        0.0   1314.066        0.0        1
* Outlet b.c. -----------------------------------------------------------
*IBD1 IBD2 ISPC N1FN N2FN N3FN   BCVALUE1   BCVALUE2   BCVALUE3 INITGAS
    1   24    7    0    0    0        0.0   1314.066     157.44        1
    2   24    7    0    0    0        0.0   1314.066     157.44        1
    3   24    7    0    0    0        0.0   1314.066     157.44        1
    4   24    7    0    0    0        0.0   1314.066     157.44        1
* Coupled BCs
* ATHLET Channel  BC Type  Number of CTF channels CTF channel IDs
           CORE1        B                       1              1
           CORE2        B                       1              2   3
           CORE3        I                       1              4
           CORE4        O                       1              4
```

Figure 5.4: Example of a CTF coupled input

When the TH model in CTF is more detailed than the one in ATHLET, which is usually the case, the inlet mass flow $\dot{m}$ in a CTF channel is proportional to the ratio between the CTF and the ATHLET channel flow area A.

$$\dot{m}_{CTF} = \dot{m}_{ATHLET} \times \frac{A_{CTF}}{A_{ATHLET}} \tag{5.1}$$

For the pressure and the fluid enthalpy (/temperature) a uniform distribution is passed from the ATHLET channel to all connected CTF channels.

In the current state, the coupling is only applicable to single-phase flow (at core inlet). In order to work for two-phase flow, the velocity of each phase should be transferred to CTF.

This is a one-way coupling, since no information from the sub-channel code is transferred to the system code.

The standard ATHLET-QUABOX/CUBBOX coupled version, described in Section 3.1, is used as a basis. Therefore, CTF is not only coupled to ATHLET but to QUABOX/CUBBOX as well. The existing interface between CTF and QUABOX/ CUBBOX is applied. However, for the rest of the work described in this chapter, the transfer of TH feedbacks from CTF to QUABOX/CUBBOX is

switched off. CTF still receives the power from QUABOX/CUBBOX. The reason for this is to facilitate the comparison of ATHLET and CTF in order to assess the TH/TH coupling.

During the steady-state phase of the calculation, the TH boundary conditions and the power are transferred to CTF only after ATHLET- QUABOX/CUBBOX is converged. CTF then performs a zero-transient calculation until convergence at nodal level of fuel temperature and moderator density is reached (as described in Section 3.2.1).
During transients, the system code is leading: it first performs its time-step and then submits the new set of boundary conditions to the sub-channel code.
For the time step size, the minimum of CTF and ATHLET proposed time step size is used.
Initially, large deviations were observed for the center-line fuel temperature with a systematic overestimation on the ATHET side (>100K). These deviations were investigated and the main reason was identified. The first reason is the different between the built-in correlations and the fuel property tables in CTF and ATHLET. This holds especially for the UO2 heat conductivity correlation. Therefore, for the sake of comparison, the ATHLET uranium oxide thermal capacity and thermal conductivity correlations were implemented in CTF. Using these correlations, a good agreement can be observed for the fuel temperature results of both codes in Figure 5.5 where the deviation drops 101K from to 7K.

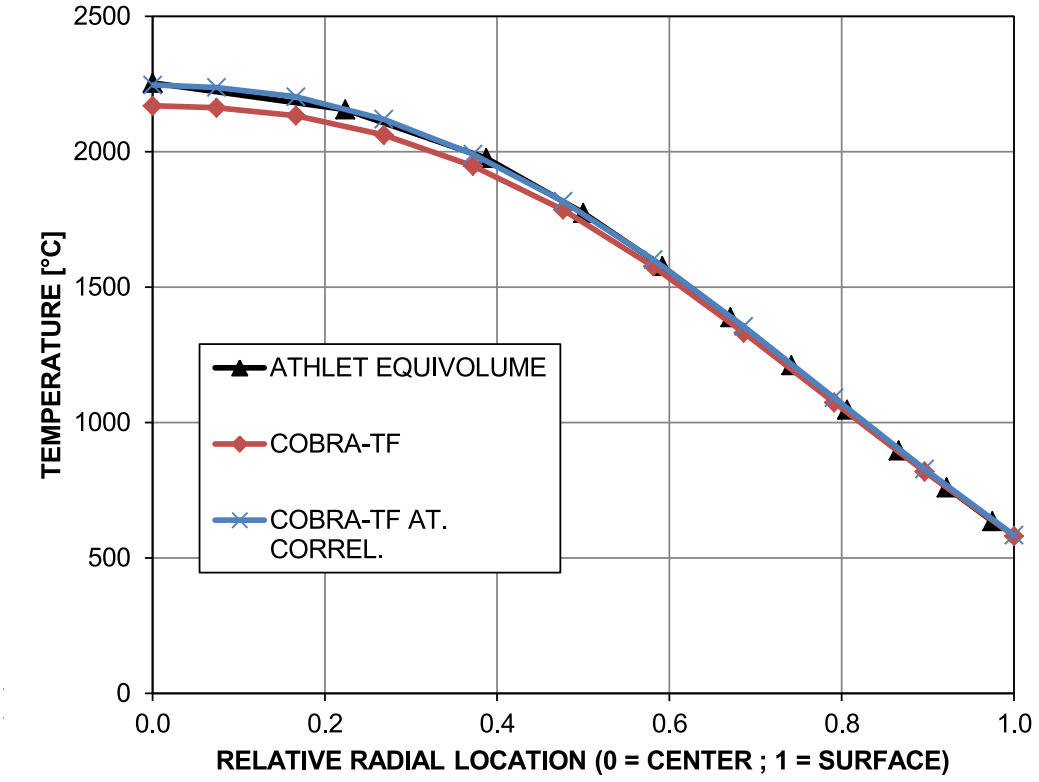

Figure 5.5: Comparison of the radial temperature distribution in the fuel pellet in ATHLET and in CTF

5.3 Application of the ATHLET/CTF Coupling

5.3.1 Model Description

The results of the triple coupling which are presented in the this section were published in [39].
For the first validation test of the coupled system, a generic PWR NPP model was used. It is based on a German four-loop reactor which is fully modeled (primary and secondary control systems included) in the system code ATHLET. The core neutronics is modeled using QUABOX-CUBBOX. This model was preferred to an ATHLET/QUABOX-CUBBOX open-core model in order to have an actual multi-scale simulation: from plant level to assembly level.
In this study, two CTF models are developed and compared. Both are coupled to the same ATHLET/QUABOX-CUBBOX generic PWR NPP model. As mentioned in the precedent section, the TH feedbacks are taken from the coarse ATHLET model so that the observed differences only come from the TH models.
The first core model is the model originally developed for ATHLET. It is a coarse representation of the core. Nevertheless, it provides a good compromise between accuracy and computational costs. In this model, each core TH-channel represents several fuel assemblies. The 193 fuel assemblies are grouped under the following principles:

- four azimuthal zones (one per primary loop) in case of asymmetric transients (e.g. MSLB)
- and three rings in order to follow the power profile.

Figure 5.6 shows the resulting model which contains 17 TH channels, each representing 9 to 13 fuel assemblies.
Adjacent channels are connected with cross-connections allowing mass transfer between them. This model has been reproduced rigorously in CTF. Although this coarse model doesn't take advantage of the sub-channel capabilities of CTF, it is useful to compare ATHLET and CTF results with equivalent models.

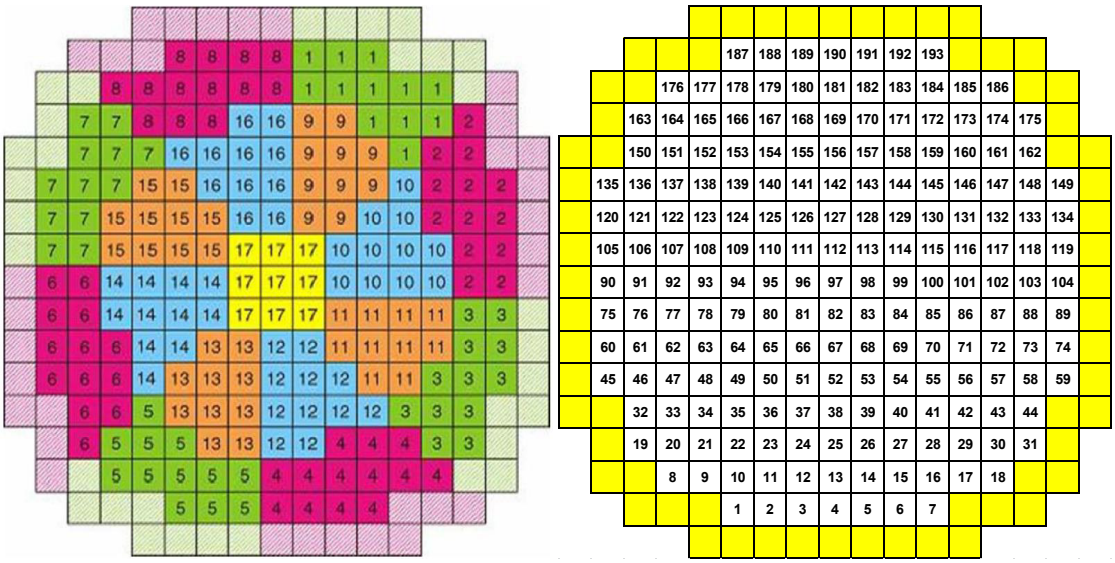

Figure 5.6: 17-Channel TH Core Model (left-hand side) - 193-Channel TH Core Model (right-hand side)

A second TH core model has been developed for CTF. It features a 1:1 assembly-wise representation of the core (each fuel assembly is assigned to one channel). As in the first model, adjacent channels are connected with cross-connections. This model is used for subchannel transient analysis.

5.3.2 Steady-State Results

The steady-state results in channel #1, where the largest deviations between the two codes can be found, are shown in Figure 5.7. Nevertheless, the results are in very good agreement.
The simulation was repeated using the second CTF model (with 193 channels). No data exchange or convergence problems were encountered when using different core models in ATHLET and in CTF.

5.3.3 Transient results

As a first test to prove the transient capabilities of the coupled system, an asymmetric uncontrolled rod removal has been modeled. Initially, all control rods are

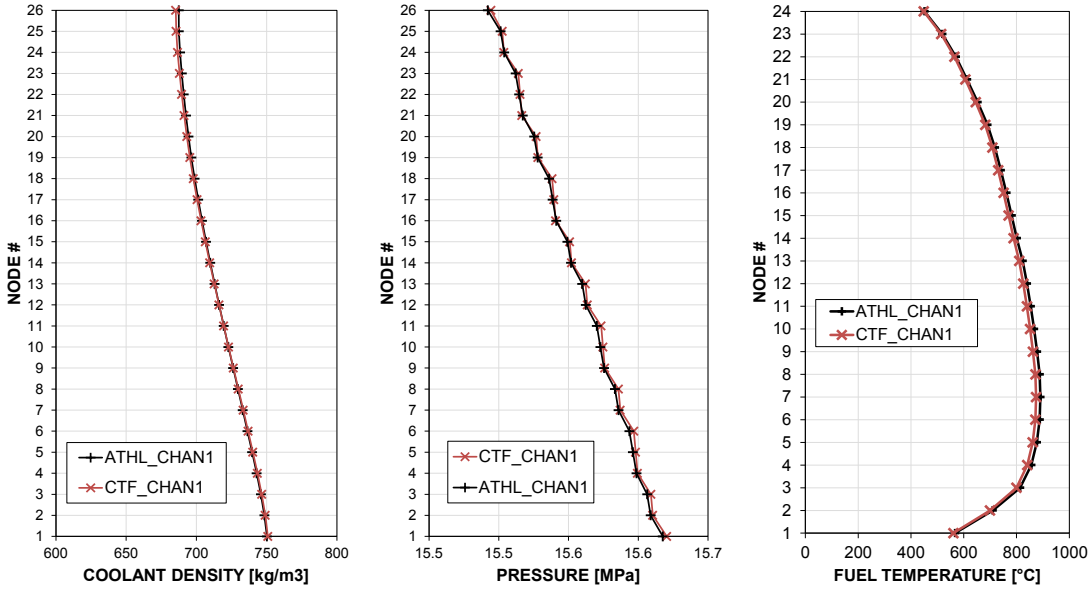

Figure 5.7: Steady-State Comparison of the 17-Channel Core Models in ATHLET and CTF

fully withdrawn except for the four control rods in Cluster C1 which are inserted 56 cm inside the active core. One of them, located in assembly#183 in the 193-channel TH core model, is slowly extracted from 56 cm at $t = 0s$ to 0 cm at $t = 22.4s$ (rod velocity = 2.5 cm/s).

This transient has a global impact on the core power as well as local effects on the radial and axial power distributions. However, these effects are limited and much attenuated in case of a coarse modelisation of the core.

The results presented in this paragraph were generated using the 17-channel model. As expected, the effect of the rod withdrawal on the core power calculated by QUABOX/CUBBOX is very limited (only 0.5%). The effect on the axial power profile in TH channel#1 (in which the control rod is extracted) is limited to the top of the core as seen in Figure 5.8. The comparison of ATHLET and CTF shows that the slight deviations already present after the steady state remain constant during the whole transient.

The transient simulation was repeated using the second CTF model (with 193 channels). The results in channel#183, where the control rod is extracted, are com-

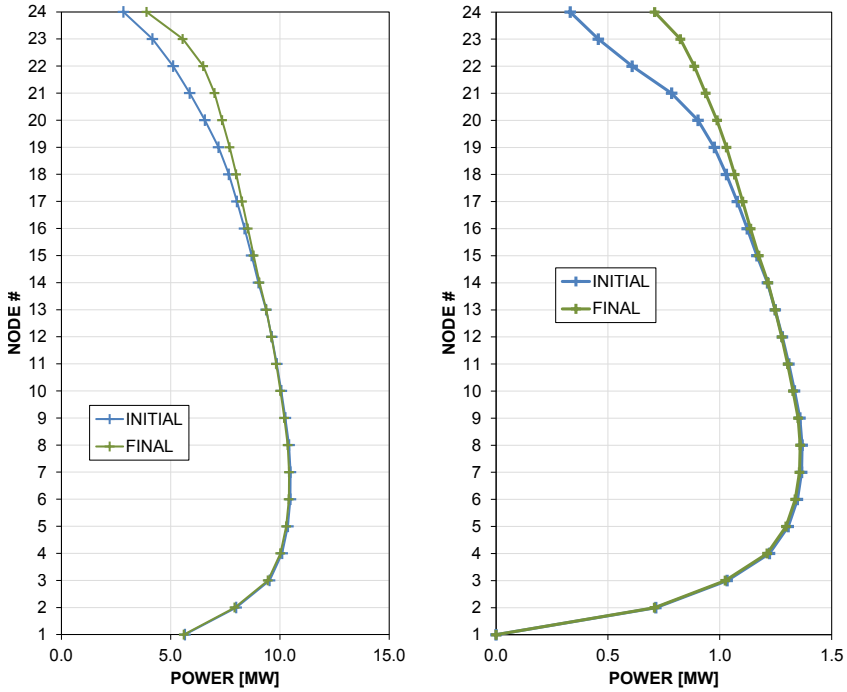

Figure 5.8: Axial power profiles at beginning and end of the transient - Left in coarse model channel#1 - Right in channel#183

pared with the results previously obtained in channel#1 (with the coarse model). As expected, the effect of the rod movement on the power profile is larger than with the coarse TH core model. (see Figure 5.8) A larger decrease of outlet moderator density (-8 kg/m^3 vs. -2 kg/m^3) and a larger fuel temperature rise (+280K vs. +75K) are thus induced.

5.4 Conclusions on the ATHLET/CTF Coupling

The subchannel code CTF was successfully coupled to the system code ATHLET using a one-way parallel coupling approach.
For the time coupling, an explicit staggered scheme is used wherein ATHLET is leading but the time step size is the minimum from ATHLET and CTF.
The multi-physics tri-code coupling with the neutron physics code QUABOX/CUBBOX is also operational. However, in the presented example, the TH feedbacks were generated by ATHLET. This ways, the observed differences in the results only come from the difference in TH models in CTF and ATHLET. In order to take advantage of the subchannel modeling from CTF, in the future the TH feedbacks should be transferred from CTF.

CHAPTER 6

Coupling on the Salomé Platform

In the framework of the NURESAFE project [40], a fully functional multi-physics multi-scale simulation tool was developed. The TH system code ATHLET, the sub-channel code CTF and the neutronics code DYN3D were integrated and coupled on the Salomé platform. This European platform, product of three consecutive European projects, is described in the next section. The developments achieved during this three-year project include assembly level and pin-by-pin TH/NP couplings, as well as a TH/TH coupling between ATHLET and CTF. All those developments are described in this chapter.

6.1 Presentation of the Salomé Platform

The Salomé platform is an open-source software co-developed by EDF, CEA and OpenCascade. Originally created for CAD applications, it has since evolved into a platform for code coupling in the framework of a series of three consecutive European Commission founded projects: NURESIM (2006-2008), NURISP (2009-2011) and NURESAFE (2013-2015).
When coupling codes on the Salomé platform, the codes do not directly communicate with each other but rather with the platform. It is a parallel processing type

of coupling as described in Section 2.1.2. The codes are integrated on the platform as "components". The code integration can either be soft or strong. In the first case, the component is a black box and only the code's executable is accessible from the platform. In the case of a strong integration, single code functions can be called from the platform (e.g. initialize code, read input, perform steady-state, etc). To allow the coupling with other codes through the platform, a strong code integration is necessary.

The Salomé platform is coded in C++. Therefore, the component's interface should preferably be written in C++. However, most codes in the nuclear industry are written in Fortran. This is the case for ATHLET, CTF and DYN3D. Therefore the codes' interface should be able to interoperate with C++ and Fortran libraries.

In addition, Salomé contains a component named YACS which allows the visualization of code and data flow. This option is useful for introducing code coupling to beginners since it allows building an application by simple drag-and-drop of the different code functions and to exchange data by connecting the boxes. However the graph rapidly becomes cumbersome. Figure 6.1 shows an example of a YACS graph which initializes the following components: ATHLET, DYN3D and the interpotation tool INTERP_2_5D (presented in the next section). One can easily imagine how fast the graph would grow if a third component or a transient calculation was added. In addition the use of a YACS graph also represents a computational overhead. Therefore, another solution, more adapted for advanced users, is usually preferred.

The Salomé platform also feature an internal Python console, in which all loaded components' functions can be called. Furthermore, the whole platform environment, including the components, can be loaded into an external Python console and executed there. The use of Python scripts is highly recommended for production applications, as it is easily adaptable and prevents computational overheads from the graphical interface. An example of such script, applicable for both applications presented in the next chapter, is given in Appendix A.

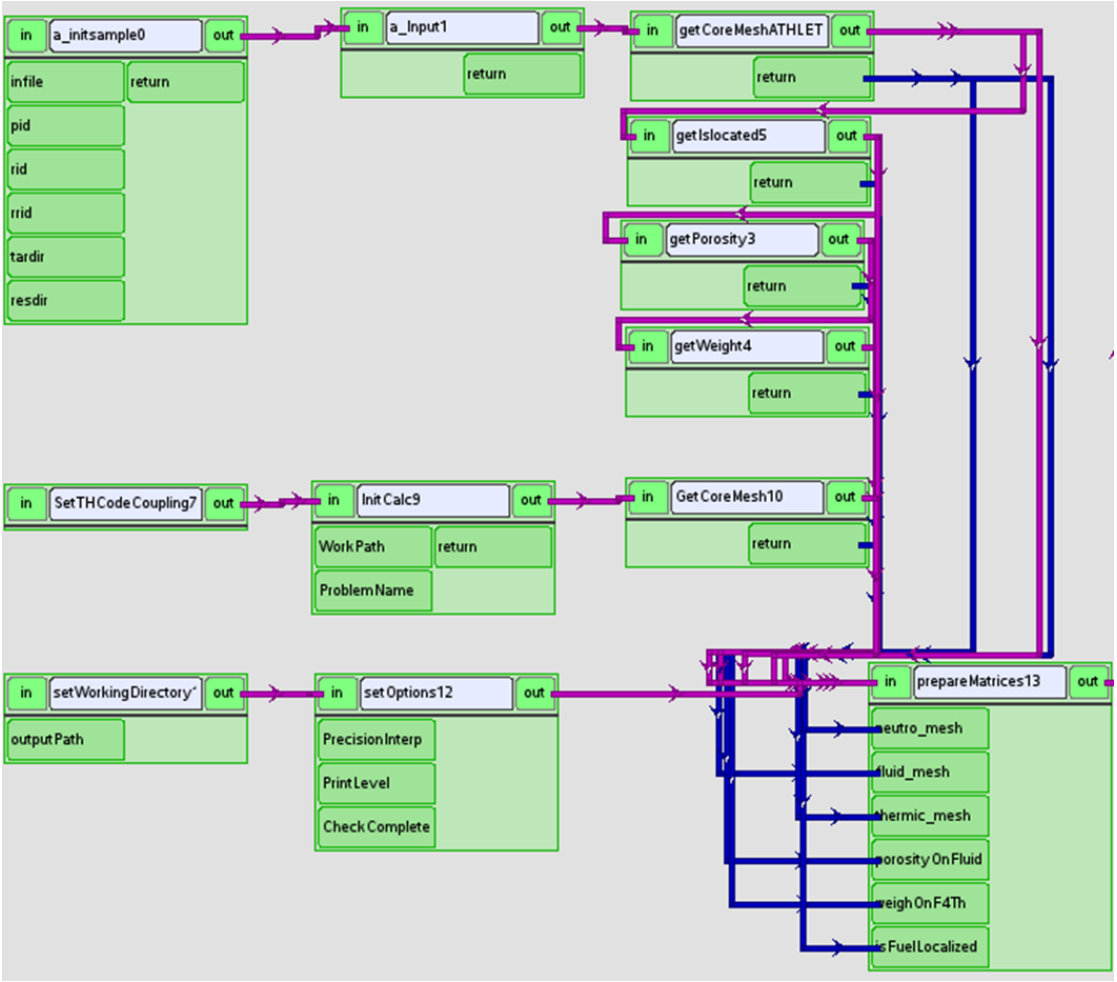

Figure 6.1: YACS Graph for ATHLET-DYN3D-INTERP_2_5D Initialization

6.2 Data Exchange on the Platform

The data exchange is performed through memory using a dedicated data structure: MEDCoupling. More information on the MEDCoupling format and the interpolation tools is available in the Salomé platform documentation (http://docs.salome-platform.org/).
The MED format (and later MEDCoupling), was developed by EDF and CEA to answer the challenges of data exchange for multi-physics simulations. The goal was to design a standardized approach that could be used to exchange data between codes while being code-independent.
The MED data model has two components:

- Mesh: The mesh contains the geometry of a domain which is represented by a set of cells and nodes. In this study, 3D surface mesh (3D space, 2D cells) and 3D mesh (3D space, 3D cells) are used.
- Fields: The fields are the information that the codes actually exchange. They can be set on the mesh cells (so-called P0 field) or nodes (so-called P1 field).

Fields can be either intensive or extensive. Intensive data do not depend on the volume of the physical system represented. Examples of intensive data are: moderator density, power density, temperature or pressure. Extensive data is proportional to the volume of the physical system represented. Examples of extensive data are: mass and power.
There are two ways to access to the MED data:

- Files: MED data can be stored in files that follow the HDF5 library.
- Memory: The MEDCoupling library allows to access and to process the MED data using C++ or Python.

In order to couple a code on the platform, one first needs to create a MED formatted mesh, then save the data to be exchanged as a field on the generated mesh.

A set of interpolation tools for the MED format is available on the platform.
The interpolation algorithm depends on whether the field type is P0 or P1. For all codes in this study, the fields are cell-based P0. A short description of the P0 field interpolation methods is given thereafter.
The aim of the interpolation is to take a source field Φ_s on a source mesh and transform it in a target field Φ_t. The relation between source field and target field is expressed as a matrix-vector product:

$$\Phi_t = W.\Phi_s \tag{6.1}$$

where W is called the interpolation matrix.
The interpolation of P0 fields follows two rules:
The first is the conservation principle which means that the integral of the field must be conserved. It is expressed by the following *general interpolation formula*:

$$\int_{T_i} \Phi_t = \sum_{S_j \cap T_i \neq 0} \int_{T_i \cap S_j} \Phi_s \tag{6.2}$$

where S_j is a source cell and T_i a target cell.
The second rule is the maximum principle: the target field values must remain between the upper and lower bounds of the source field.
It is only possible to satisfy both rules if the source and the target mesh *fully overlap*. This is expressed by:

$$\sum_{S_j} Vol(T_i \cap S_j) = Vol(T_i) \; and \; \sum_{T_i} Vol(S_j \cap T_i) = Vol(S_j) \tag{6.3}$$

If the two meshes do not fully overlap then it is only possible to respect one of the principles.

In most cases presented in this study, the meshes are fully overlapping. It is not the case of the pin level rod meshes generated by CTF (see Section 6.5). In that case the following rules apply: For extensive fields like the power, the conservation

principle is the most important. For intensive fields like fuel temperature, the maximum principle is the most important.
For the coupling of Thermal-Hydraulics with Neutron Physics models on the Salomé platform, the INTERP_2_5D component is usually applied. This component was originally developed by the CEA for the coupling between FLICA4 (TH subchannel code) and CRONOS (neutron diffusion code). It takes one mesh from the neutronic side and two meshes from the TH one: one for the fluid mesh and one for the thermal mesh. At the assembly level the same mesh is used both for the fluid and the thermal mesh. At the pin level, the fluid mesh and the thermal mesh usually differ. The INTERP_2_5D component computes the interpolation matrix once for a given Neutronic mesh/Fluid mesh/Thermal Mesh system. Afterward, three component functions are used:

- projectPower which interpolates the power from the neutronic mesh to the thermal mesh and to the fluid mesh.
- projectThField which interpolates a field from the thermal mesh to the neutronic mesh
- projectFluidField which interpolates a field from the fluid mesh to the neutronic mesh

6.3 DYN3D Integration and Coupling on the Platform

The integration of DYN3D on the platform was performed at the HZDR during the NURESIM and NURISP project. The coupling with the thermal-hydraulic subchannel code FLICA4 was achieved at that time. This coupling uses the INTERP_2_5D component described in Section 6.2. The DYN3D component can be used either as a stand-alone code or coupled with a TH code. The mesh generation function supports quadratic and hexagonal fuel geometries. When the pin flux reconstruction option is activated, a mesh refinement is automatically performed.

During the NURESAFE project, the interface with the platform was extended at the HZDR to allow coupling with the CFD code TRIO-U.
For this thesis, in the framework of the NURESAFE project, the coupling to ATHLET and CTF was developed. It is described in more details in the next two sections.

6.4 ATHLET Integration and Coupling on the Platform

A first (soft) integration of ATHLET was achieved during the NURISP project. ATHLET could be launched from the platform but only as an executable. For this thesis, in the framework of the NURESAFE Project, ATHLET was fully integrated on the platform. All important code functions can be called from the platform and allow to run stand-alone as well as coupled applications.
The methods in the API (Application Program Interface) can be divided into two groups: code control and data exchange.
The code control methods developed for the ATHLET API allow to:

- Initialize the code (including input processing),
- Perform a steady-state convergence,
- Initialize the transient calculation
- Control the transient calculation
- Finalize the simulation

The transient calculations can either be run at once (preferable for stand-alone simulations) or controlled very finely. At each time step, it is possible to check the ATHLET proposed time step size, to set manually the time step size and to compute one or several time steps. However, for the time being, it is not possible to rewind and repeat a time step. Explicit time coupling is thus the only possible approach on the platform.

The methods developed for data exchange with a neutron physics code are compatible with the interpolation component INTERP_2_5D. ATHLET can also be coupled with CTF on the platform but that particular coupling is the object of Section 6.6. The methods generating the 3D MEDCoupling mesh are called automatically during the initialization phase if a mapping scheme is provided in the input. The generation of the mesh is described in the next paragraph. After each steady-state convergence or time step, it is possible to extract the following fields on the generated mesh:

- Fuel Doppler temperature
- Moderator density
- Moderator temperature
- Boron concentration
- Power (for the coupling with CTF)

For coupled simulation, a 3D mesh generation function was developed. However, ATHLET is a 1D system code. Its input contains information on the position of a given object only in the axial (z-)direction. The challenge of generating a 3D mesh out of a 1D model is solved by using the already existing coupled version of the code.
For the purpose of coupling with NP codes, the ATHLET input contains an additional part wherein the so-called mapping scheme is given. The mapping scheme is a 2D representation of the core which shows where each channel is located in the radial plane. Figure 6.3 shows two examples of mapping schemes in the ATHLET input:

- On the left hand side, the mapping scheme for a quadratic 5x5 minicore
- On the right hand side, the mapping scheme for a hexagonal 7 assembly minicore.

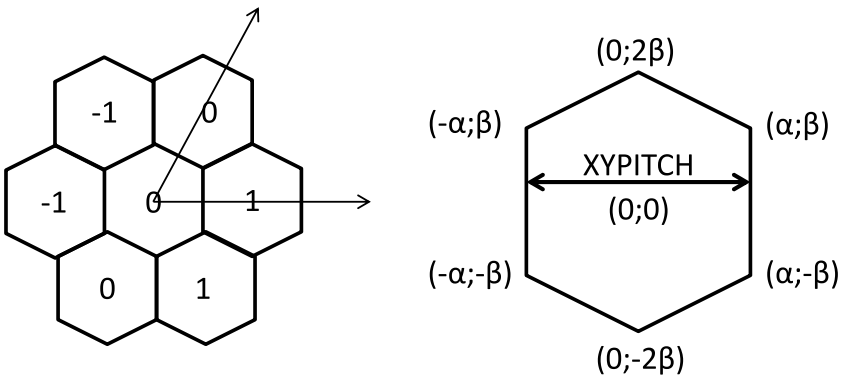

Figure 6.2: Hexagonal referential and coordinates system

A geometry type identifier (GEOMTYP in Figure 6.3) was added to the input in order to select automatically the correct algorithm for the mesh generation. For quadratic geometries (GEOMTYP = 0), the number of lines in the mapping scheme is given with the parameter NSTFM. Then, the number of channel NNOSST is defined for each of the NSTFM lines. For hexagonal geometries (GEOMTYP = 1), NSTFM is still the number of lines in the mapping scheme. Then, the [Xmin,Xmax] coordinates in the hexagonal referential.
In order to generated a 3D mesh from the mapping scheme, the only missing piece of information is the assembly pitch in the case of quadratic assemblies and the face to face width in the case of hexagonal assemblies (XYPITCH in Figure 6.3). Two mesh generators were developed, one for each type of geometry.
Figure 6.2 shows how the coordinates of the hexagons are calculated in the mesh generator. On the left hand side, the X coordinate in the hexagonal referential is shown. The origin is placed at the center of the core. On the right hand side, the local coordinates $(\alpha;\beta)$ of the hexagon's 6 corners are presented.

$$\alpha = \frac{XYPITCH}{2} \tag{6.4}$$

$$\beta = \frac{XYPITCH}{2\sqrt{3}} \tag{6.5}$$

As illustration, 3D meshes for a quadratic and a hexagonal fuel geometry are shown respectively in Figure 6.4 and Figure 6.5.

```
----- FEEDBMAP
@
@ GEOMTYP   XYPITCH
  0         0.2142
@
@  NSTFM
       5
@
@  NNOST (1,..NSTFM)
       5     5     5     5     5
@
@  FEEDBACK MAP: CHANNEL FOR EACH N3-NODE IN (X-Y) PLANE
@
    'BY'  'BY'  'BY'  'BY'  'BY'
    'BY'  'A7'  'A8'  'A9'  'BY'
    'BY'  'A4'  'A5'  'A6'  'BY'
    'BY'  'A1'  'A2'  'A3'  'BY'
    'BY'  'BY'  'BY'  'BY'  'BY'
```

```
----- FEEDBMAP
@
@ GEOMTYP   XYPITCH
  1         0.237178
@
@  NSTFM
       3
@
@  IXMIN (1,..NSTFM)
       0    -1     -1
@  IXMAX (1,..NSTFM)
       1     1      0
@
@  FEEDBACK MAP
@
    'A6'  'A7'
 'A3'  'A4'  'A5'
    'A1'  'A2'
```

Figure 6.3: Examples of mapping schemes in the ATHLET input - Left: Quadratic geometry - Right: Hexagonal geometry

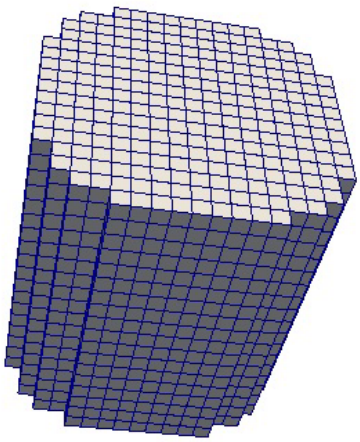

Figure 6.4: ATHLET 3D Mesh of a Core With Quadratic Assemblies

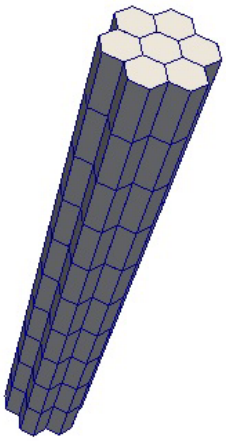

Figure 6.5: ATHLET 3D Mesh of a Core With Hexagonal Assemblies

Once a code is integrated on the platform, coupled simulations can be run in the Salomé environment with the YACS graphical interface. For ease of use reasons, Python scripts are usually preferred. The platform components are dynamic libraries that can be imported in Python. For the coupling of ATHLET and DYN3D, both component must be imported as well as the INTERP_2_5D component. After the codes are initialized and the meshes generated, the interpolation matrix is computed. Steady-state calculations are usually programmed with a "while loop" which is parametrized with the convergence criteria. For transient simulations, the explicit staggered time step synchronization scheme is used wherein ATHLET is leading.

The coupling on the platform was first tested on mini-core arrangements with open core TH models. Later, full scale simulations of safety relevant transients were computed. The results are presented in the next chapter.

6.5 CTF Integration and Coupling on the Platform

CTF was not part of the NURESIM and NURISP projects. The code integration work was performed integrally during the NURESAFE project.
All important code functions can be called from the platform and allow to run stand-alone as well as coupled applications.
The methods in the API (Application Program Interface) can be divided into two groups: code control and data exchange.
The code control methods developed for the CTF API allow to:

- Initialize the code (including input processing),
- Check for steady-state convergence,
- Initialize the transient calculation
- Control the transient calculation
- Finalize the simulation

A commented version of the C++ header file from the CTF API can be found in Appendix B. All functions for code control but also for mesh generation and field exchange are listed.

As it was already mentioned, CTF does not have a steady-state mode. Pseudo steady-state simulations are used instead, during which no perturbation in the model occurs and the time step size for the fuel heat conduction can be artificially increased in order to accelerate the convergence. A function that checks the convergence is available. Before starting the actual transient simulation, the heat conduction time step multiplication factor is reset to 1.0.
The control of the transient calculations is even finer than for the ATHLET API. At each time step, it is possible to check the CTF proposed time step size. The time step size can also be set manually. After a time step is solved, it can either

be validated or repeated (with a different size). It is thus theoretically possible to implement a semi-implicit time coupling on the platform. In this work, only the explicit coupling approach has been implemented and tested.

The methods developed for data exchange with a neutron physics code are compatible with the interpolation component INTERP_2_5D. ATHLET can also be coupled with CTF on the platform but that particular coupling is the object of Section 6.6.
In its latest version, the CTF input contains the position of the center (and the width) of each channel (and thus fuel rod) in the radial plane. This information is enough to generate a 3D model in the case of regular quadratic geometries.
In CTF, the assembly level and the pin level are treated differently. For the assembly scale, the same mesh is used for both fluid and thermal mesh. For the pin scale, the thermal mesh is disjointed (since the pins don't touch each other) and can even contain holes (e.g. where control rod guiding tubes are located). The algorithm can automatically select the correct model by checking the multiplication factor of the fuel pin object. If it is more than one, the assembly scale is assumed. At the pin scale, it is possible to model the thermal-hydraulics with rod-centered or the coolant-centered models. The developed algorithm automatically selects the correct model by checking the channel and the fuel rod maps contained in the input. If their sizes are the same, the rod-centered model is assumed, otherwise, the coolant-centered is applied.
It is possible to mix assembly scale and pin scale, i.e. use a refined mesh for a hot spot analysis. However, in that case, only the rod-centered model is allowed for the pin scale. An example of fuel temperature field for such an hybrid assembly/pin level thermal mesh is represented in Figure 6.6.

For hexagonal geometry, at the assembly level, the algorithm can easily be adapted. The geometric data in the CTF input is interpreted differently than in the standard case. The channel center coordinates stay the same but the X and Y direction width are replaced by the face to face width (equivalent to XYPITCH in the ATHLET input). The channel and fuel rod maps are also adapted.

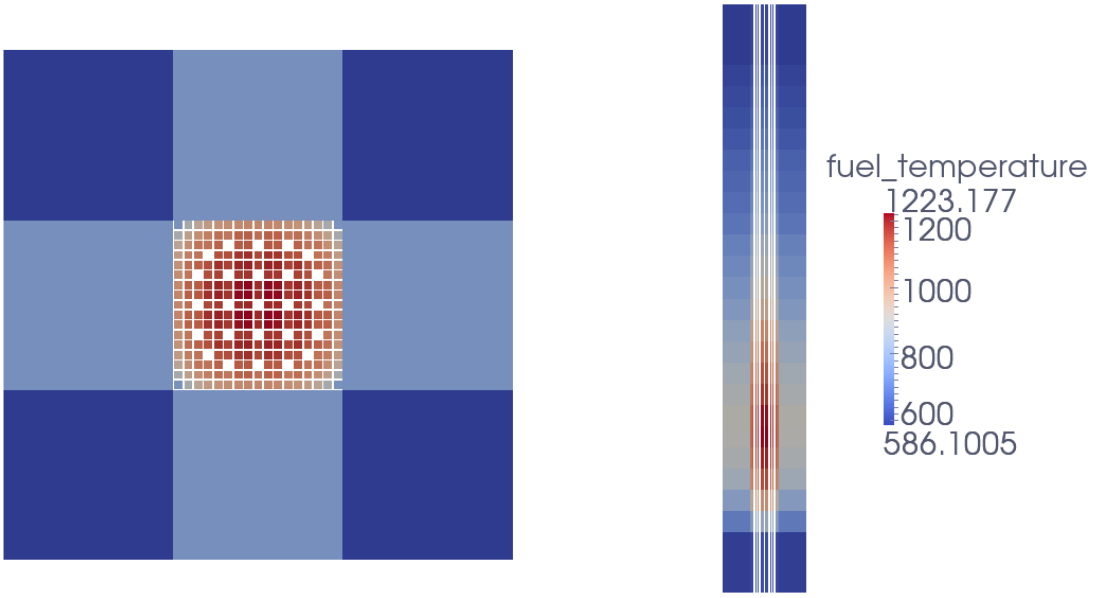

Figure 6.6: Fuel Temperature Field on a CTF Hybrid Assembly/Pin Level Thermal Mesh - Left: 2D Radial cut - Right: 2D Axial cut

Figure 6.7 shows two examples of mapping schemes in the CTF input:

- On the left hand side, the mapping scheme for a quadratic 5x5 minicore
- On the right hand side, the mapping scheme for a hexagonal 7 assembly minicore.

At the pin level however, the generation of a 3D mesh is more complex because of the inter-assembly water gaps which prevent the use of regular hexagons (for rod-centered models) or equilateral triangles (for coolant-centered models). In the scope of the NURESAFE project, an algorithm was developed for each case: in collaboration with the INRNE (Bulgaria) for the rod-centered model and with KIT (Germany) for the coolant-centered model. The resulting fluid and thermal mesh generated with these algorithms are represented in Figure 6.8. The coolant-centered algorithm also serves as preprocessor and can generate fully functional CTF inputs. These models have only been tested as stand-alone CTF applications. Further development is needed for pin-by-pin coupled simulation of hexagonal geometry on the platform.

```
*Card 17.2 - Rod Map Dimensions
**TOTRODSROW TOTRODSCOL
           5           5
*Card 17.3 - Channel Map Dimensions
**TOTCHANSROW TOTCHANSCOL
           5           5
*Card 17.4 - Rod Map
    0    0    0    0    0
    0    1    1    1    0
    0    1    1    1    0
    0    1    1    1    0
    0    0    0    0    0
*Card 17.4 - Channel Map
    1    1    1    1    1
    1    1    1    1    1
    1    1    1    1    1
    1    1    1    1    1
    1    1    1    1    1
```

```
*Card 17.2 - Rod Map Dimensions
**TOTRODSROW TOTRODSCOL
           3           3
*Card 17.3 - Channel Map Dimensions
**TOTCHANSROW TOTCHANSCOL
           3           3
*Card 17.4 - Rod Map
    0    1    1
    1    1    1
    1    1    0
*Card 17.4 - Channel Map
    0    1    1
    1    1    1
    1    1    0
```

Figure 6.7: Examples of mapping schemes in the CTF input - Left: Quadratic geometry - Right: Hexagonal geometry

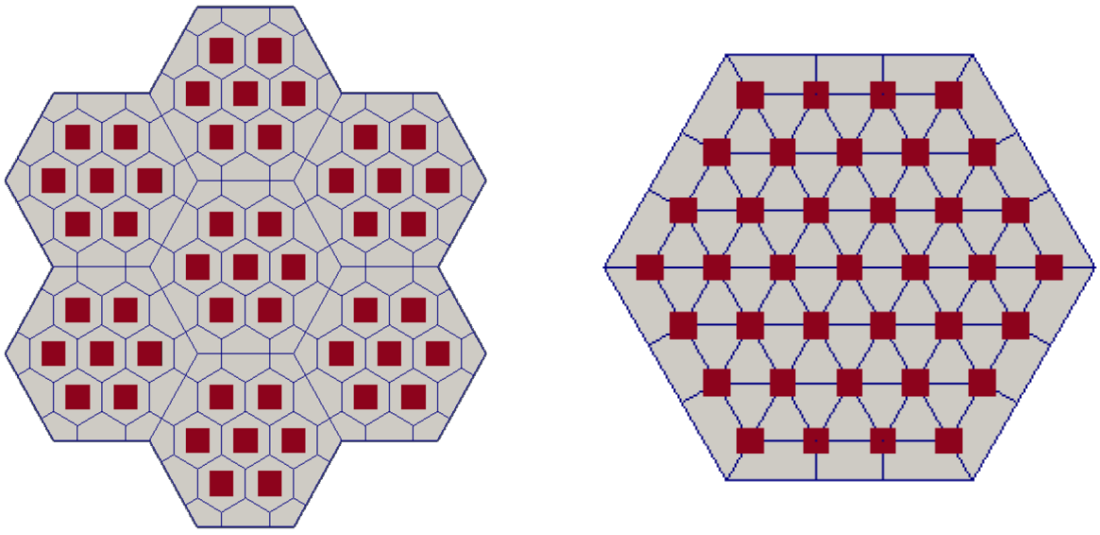

Figure 6.8: Hexagonal Assembly at the Pin Level: Fuel-Centered and Coolant-Centered 3D Meshes

After a time step is validated, it is possible to extract the following fields on the fluid mesh: moderator density, moderator temperature and boron concentration; and fuel temperature on the thermal mesh.

Once a code is integrated on the platform, coupled simulations can be run in the Salomé environment with the YACS graphical interface. For ease of use reasons, Python scripts are usually preferred. The platform components are dynamic libraries that can be imported in Python. For the coupling of CTF and DYN3D, both component must be imported as well as the INTERP_2_5D component.
After the codes are initialized and the meshes generated, the interpolation matrix is computed. Steady-state calculations are programmed with two embedded "while loops", one for the internal convergence of CTF and one for the convergence of the coupling. For transient simulations, the explicit staggered time step synchronization scheme is used wherein CTF is leading.

The coupling on the platform was first tested on mini-core arrangements at the assembly level and later with hybrid assembly/pin models. Finally, full core simulations of safety relevant transients were computed. The results are presented in the next chapter.

6.6 Coupling of ATHLET and CTF on the Platform

The TH/TH coupling of ATHLET and CTF on the Salomé platform was performed after the TH/NP coupling with DYN3D was already achieved. Therefore, the tools to generate 3D mesh of the core were available for both codes. From the 3D mesh, it becomes trivial to extract 2D mesh of core inlet and exit.
As in the previous chapter, the parallel coupling method is applied. Again the data is transferred only from ATHLET to CTF. The exchanged fields are:

- core inlet mass flow,
- core inlet temperature,

- core inlet boron concentration,
- core exit pressure.

For field interpolation, another component, REMAPPER, is used instead of INTERP_2_5D which is not adapted for 2D applications. Unlike for the coupling with a neutron physics code, the REMAPPER component is called directly in the method of the CTF C++ API. This means that the REMAPPER component does not need to be imported in the coupling Python script.

In order to generate the ATHLET mesh, the same data as for the neutronics coupling is necessary. This means that the mapping scheme part of the input is needed in ATHLET even when the coupling with a neutronic code is not actually used.
Since the data is available, the coupling interface was expanded to allow ATHLET to export its power field on the 3D core mesh. In order to use this option, the INTERP_2_5D is needed, wherein ATHLET plays the role of the neutron physics code. This option has been tested at KIT in the scope of NURESAFE Work Package 3.3 and reported in a deliverable [41].

Coupled calculations are usually run through Python scripts. The logic is the same as in the two previous sections.
For transient simulations, the explicit staggered time step synchronization scheme is used. The time step size is the smallest of CTF and ATHLET proposed time step size. This solution increases the computation time but improves the stability. No stability problems have been encountered in any of the coupled simulations.
It is also possible to couple ATHLET and CTF together with DYN3D. In that case two instances of INTERP_2_5D are needed, one for the ATHLET/DYN3D and one for the CTF/DYN3D interpolation matrix. The thermal-hydraulic feedbacks for DYN3D can be provided either from ATHLET or from CTF. The preferred option is CTF since the model is more detailed. In the case of hot channel analysis, where the CTF model only covers one assembly, the feedbacks would be taken from the ATHLET model. This three code coupling on the Salomé platform is illustrated in a flow chart in Figure 6.9.

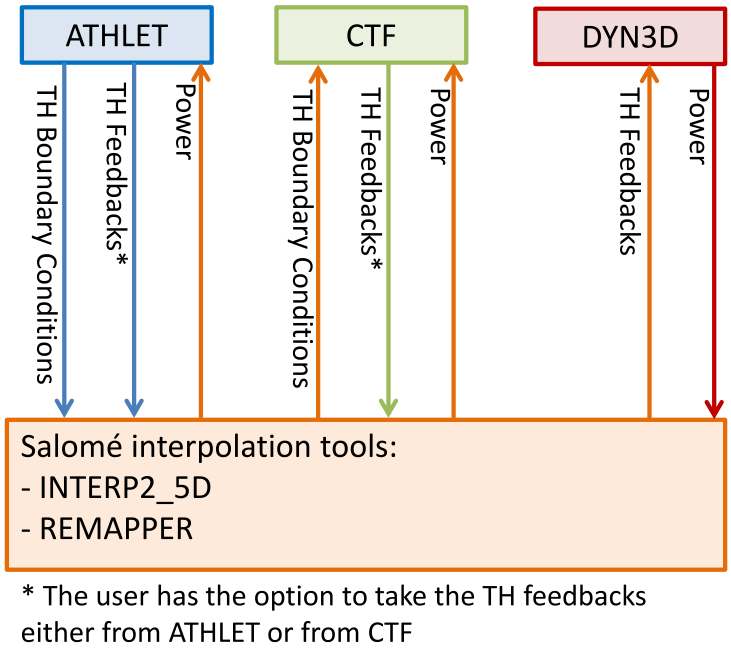

Figure 6.9: Flow chart of the ATHLET/CTF/DYN3D coupling on the Salome platform

The tri-code coupled system ATHLET, CTF and DYN3D covers the whole range, from plant level to the pin level. It has been tested on representative PWR and BWR transient analyses, the results of which are presented in the next chapter.

CHAPTER 7

Examples of Multi-Physics, Multi-Scale Applications

In the previous chapter, the development of a Multi-Physics, Multi-Scale simulation tool was described. The thermal-hydraulic system code ATHLET, the subchannel code CTF and the neutron physics code DYN3D were integrated and coupled on the Salomé platform. In the scope of this project, this tri-code coupled system has been applied for the simulation of a PWR Main Steam Line Break Transient and a BWR Turbine Trip under ATWS conditions. In this chapter, the results on those two transients are presented. Comparisons with the ATHLET-DYN3D coupled system are provided, thus showing the impact of the more realistic model.

7.1 PWR Application: Main Steam Line Break Transient

In this section, models and results from simulations performed in the scope of the NURESAFE Work Package 1.2 (WP1.2 Higher-order PWR MSLB simulation) are used. According to the project description [40], the objective of WP1.2 is

the *development and execution of a set of simulation schemes towards higher-fidelity simulation of a PWR MSLB transient for improved predictions the key safety parameters*. For the MSLB analysis, the Zion PWR was selected by the Work Package participants. The two units of the Zion power plant are 4-loop PWR of Westinghouse design which are permanently shut down since 1998. The core loading is once again taken from the UOX/MOX Core Transient Benchmark [20].

7.1.1 Transient description

The initiating event is a double-ended main steam line break in one of the secondary loop (here loop #1). The break causes a drop of the pressure on the Steam Generator (SG) secondary side. The magnitude of this pressure drop is affected by the break mass flow and the feedwater (FW) supply. The pressure on the secondary side determines the saturation temperature of the coolant in the SG and consequently the heat removal from the primary side. A (high) heat transfer between primary and secondary side exists as long as the SG contains enough liquid water. The transient is thus terminated when the SG falls dry. The temperature in the cold leg depends directly on the decrease of the saturation temperature (i.e. the pressure) on the secondary side. The coolant temperature in the reactor core, which determines the criticality conditions, depends mostly on the mixing phenomena between the coolant flows from the affected and intact primary loop in the downcomer and lower plenum. The mixing in the core region plays a role as well.
The following list of assumptions is made:

- The plant's initial state is Hot Zero Power (HZP) with All Rod In (ARI).
- The Control Rod (CR) with the highest worth is stuck out in the core affected area.
- The closure of the Main Steam Isolation Valves (MSIVs) starts when the Main Steam Head pressure drops below 5.5 MPa. The duration until the

MSIVs full closure is 10 s. The cross flow area of closing MSIVs is reduced linearly in time.

- All main primary coolant pumps remain in operation during the whole transient.
- Pressurizer heaters are switched off during the whole transient.
- All SG FW pumps are switched on at 0.0 s of the transient. The SG-FW mass flow rates are: 83.3 kg/s in SG1 and 5.6 kg/s in SGs 2-4. The FW pumps are switched off by the signal of SIS actuation. The feedwater temperature is 10 °C.
- The signal for SIS actuation is generated when the pressure in the reactor upper plenum drops below 12.05 MPa, but SIS is not actuated (i.e. boron is not injected to the primary system).
- There is no coolant injection from accumulators.

7.1.2 Models description

ATHLET Model

The ATHLET input deck for the Zion PWR was initially developed in GRS to perform LBLOCA in the framework of the OECD BEMUSE benchmark [42]. Therefore, it was necessary to modify the input deck to take into account the specific features of a MSLB transient. Those modifications were carried out at HZDR. The input was further adapted by GRS for coupled calculations on the Salomé platform. A graphical representation of the ATHLET plant model is shown in Figure 7.1.
An 11 channels core model was developed. The mapping scheme is represented in Figure 7.2. This mapping scheme is specific to this particular transient. A superchannel models the 3/4 of the core which are connected to the unaffected loops. A second superchannel models the remaining quarter of the core connected to the

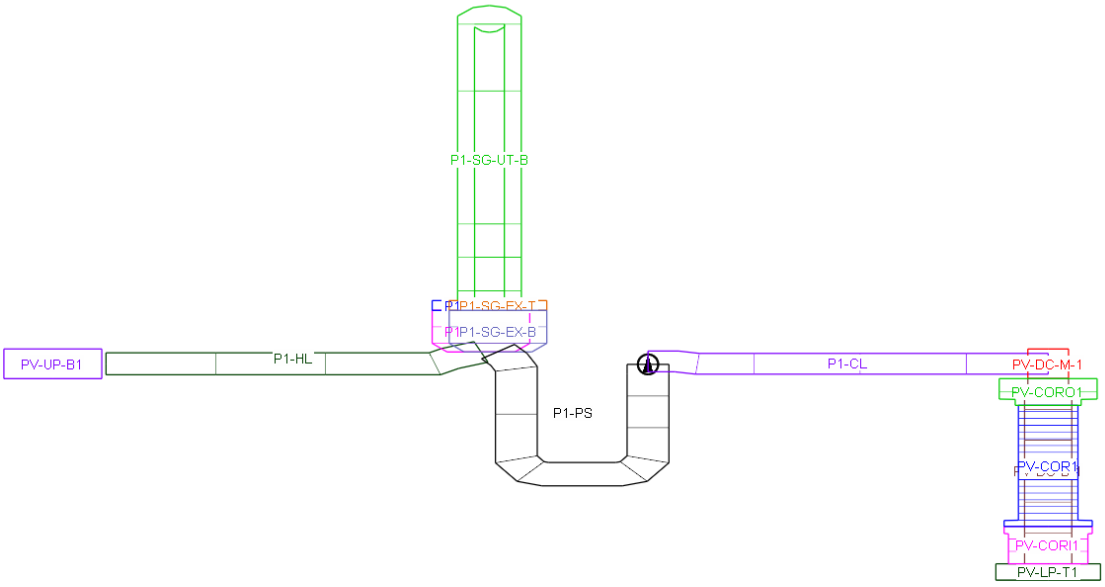

Figure 7.1: Zion ATHLET Plant Model

affected loop. The assembly with the stuck-out control rod and the surrounding 8 assemblies are modeled separately.
The mixing between the cold / hot legs is modeled as follow: All core channels are connected to a single upper plenum, this means that a full mixing is assumed for the hot legs. In the cold legs, the mixing is modeled in the four downcomer channels and four lower plenum branches. Cross-flows are also modeled between the core channels.

CTF Model

The CTF input deck for PWR Zion was developed by GRS in the framework of WP1.2. In the CTF model each assembly is modeled by a dedicated channel. The assembly with the stuck out control rod is modeled at the pin level with a rod-centered model. This makes for a total of 192 + 264 channels. The model features cross-connections between the channels. The bypass/reflector is not modeled in CTF because the reflector cross-sections are constant values. The resulting 3D mesh created for the data exchange in Salomé can be found in Figure 7.3.

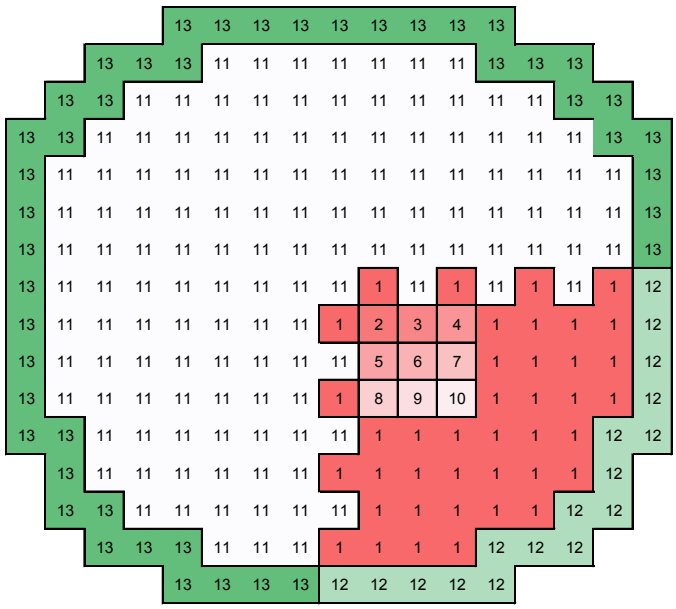

Figure 7.2: 2D Representation of the ATHLET Core Model and Mapping Scheme with DYN3D

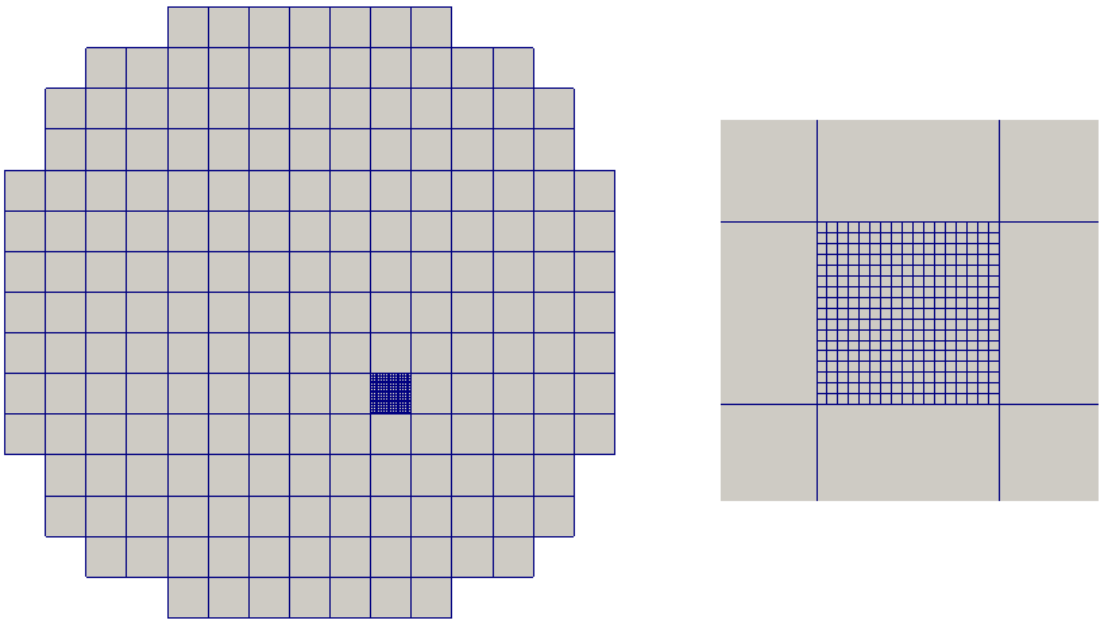

Figure 7.3: CTF 3D Hybrid Mesh - Assembly Level + Zoom Pin Level

DYN3D Model

The DYN3D input deck for PWR ZION was developed at the HZDR. The core loading is taken from the UOX/MOX Core Transient Benchmark. However, the core in the benchmark is at BOC, which is not limiting for MSLB analysis. The cycle calculation needed to generate cross-sections at EOC was performed at the IRSN with the help of the UPM using the APOLLO2 lattice and depletion code. The cross-section generation procedure is the subject of a NURESAFE deliverable [43]. The reflector cross-sections do not depend on the TH feedbacks (i.e. constant value). The pin-by-pin flux reconstruction capability is used in the assembly with the stuck control rod. This means that the power is calculated in each pin. However, the TH feedbacks are still taken at the assembly level.

7.1.3 Coupling Schemes

Three coupling schemes are applied. Their results are compared in the next sections.

In the first coupling scheme, the ATHLET model covers the whole PWR system with boundary conditions for the steam line pressure and the feedwater mass flow/temperature. ATHLET is coupled to DYN3D inside the core using the so-called internal coupling approach.

In the second coupling scheme, the ATHLET model is the same as in the previous section, including the core model. The CTF model covers only the core and is coupled (one-way coupling) with ATHLET at core inlet/exit. The CTF model features a hybrid (1:1) mapping scheme with pin by pin resolution in the hot channel (see Figure 7.3). The power calculated by the DYN3D model is transferred to both ATHLET and CTF models but only CTF sends TH feedbacks to DYN3D. The pin by pin power in the hot assembly is calculated by DYN3D using the flux reconstruction method.

Two versions of the second coupling scheme are used. In the first one, CTF receives the inlet temperature directly from the ATHLET core model. This means that only two different temperatures are used at core inlet. In the second version, a

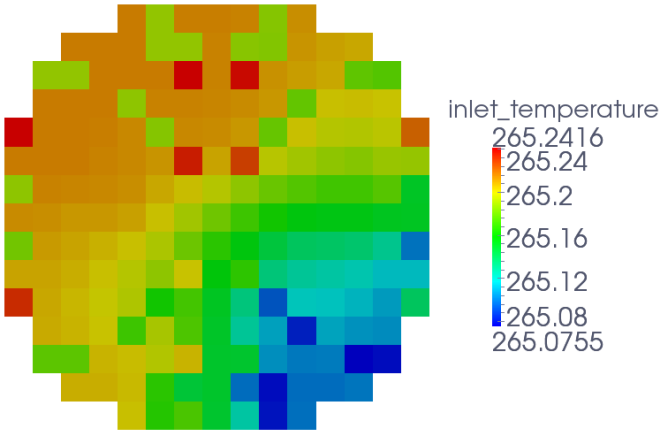

Figure 7.4: CFD Simulated Mixing Matrix

mixing matrix calculated by a CFD code on the base of the ROCOM experiment is used. The CFD simulations were conducted at UJV Rez using the commercial code Ansys Fluent [44]. The mixing matrix is based on the temperature in the four cold loops. Using this matrix, a temperature for each core channel in CTF is computed as shown in Figure 7.4. This is an example of an offline coupling.

7.1.4 Transient Analysis of Integral Parameters

In this section the transient response of some relevant integral parameters is presented. The three coupling schemes presented in the previous section are compared.

Figure 7.5 shows the power history during the transient as calculated by ATHLET-DYN3D and ATHLET-CTF-DYN3D (without mixing matrix).
In the ATHLET-DYN3D simulation, a first power peak occurs after 44.39 s and reaches 209.4 MW. A second power peak occurs after 109.76 s and reaches 499.5 MW. At the end of the transient, the power is stable at 224.5 MW which represents the new equilibrium state between primary and secondary side for the given feedwater mass flow.
In the ATHLET-CTF-DYN3D simulation, the first power peak occurs after 44.52 s and reaches 150.8 MW. The second power peak occurs after 129.56 s and reaches

423.1 MW. At the end of the transient, the power is stable and at the same level as for the ATHLET-DYN3D case.

Figure 7.9 shows the primary pressure history during the transient as calculated by ATHLET-DYN3D and ATHLET-CTF-DYN3D (without mixing matrix).
In both simulation, there is no SIS actuation because the pressure remains above 12.05 MPa. As a consequence, the secondary feedwater pumps remain active during the whole transient.
Since the FW mass flow is never stopped, the power removal is SG1 stays high and reaches a stable value after approximately 250s.
As stated in 7.1.1, the overcooling is determined by the pressure difference between the primary and the secondary side. The initial difference in the power response causes the differences observed in the primary pressure. The difference in primary pressure then causes the differences in the primary coolant temperature described next.

Figure 7.6 to 7.8 shows the primary loops' temperature in the cold and hot legs during the transient as calculated by ATHLET-DYN3D and ATHLET-CTF-DYN3D (without mixing matrix).
At first, both coupled simulations give similar results. Once the differences in the power history appear, they result in different pressure and thus coolant temperatures. Since a full mixing is used at core outlet, all hot loops have the same behavior.

Figure 7.10 and 7.11 show the power in the affected and unaffected steam generator respectively. Once the unaffected SGs are isolated, at first, the heat exchange is reversed (primary cooling the secondary side) and then goes to 0 (i.e. hot leg = cold leg temperature). At the end of the transient, the power extracted in the affected steam generator is the same in both simulations and as the power generated in the core. This confirms that a new stationary state is reached.

The observed differences in the system response can all be explained by the more accurate description of the core in the CTF model as well as differences in the

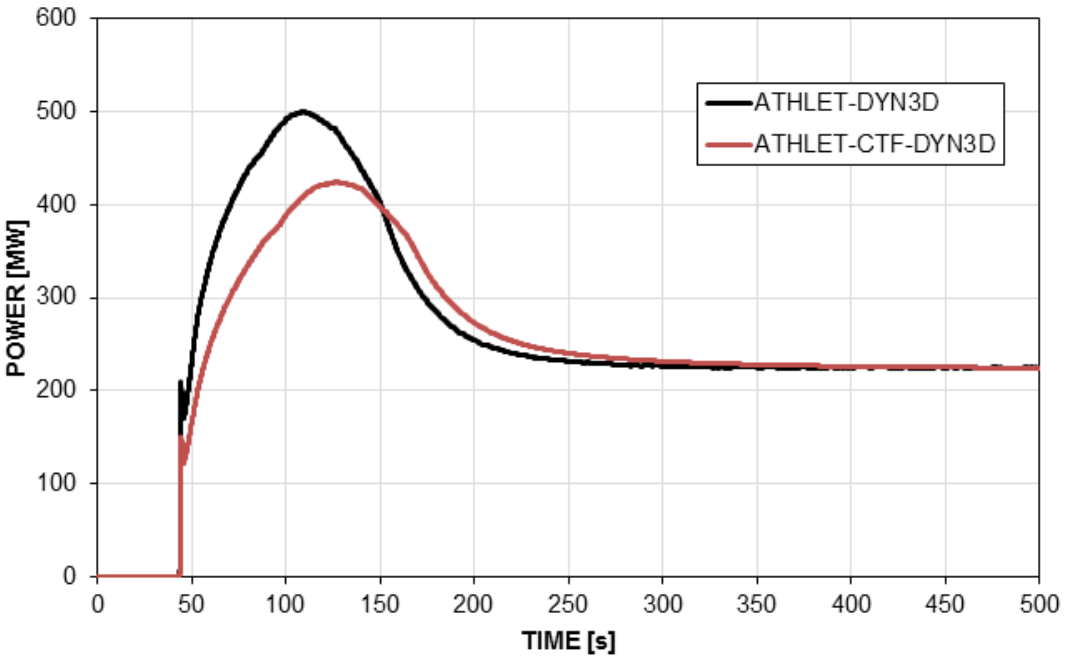

Figure 7.5: Fission Power History

fuel models and properties in ATHLET and CTF for the Doppler temperature calculation.

The results of the ATHLET-CTF-DYN3D simulation with the mixing matrix are displayed in Figure 7.12 and 7.13. When compared with the other two simulation, two main differences can be observed:
First, unlike in the other two simulations, there is no first power peak. Investigations were conducted to explain this behavior. The following Figure 7.14 shows the core inlet coolant energy in CTF with and without mixing matrix. One can see that the overcooling is actually stronger with the mixing matrix than in the other case. However, as shown in the mixing matrix, the overcooling is not homogeneous in the affected quarter and the stronger overcooling does not occur near the stuck control rod but closer to the reflector.
These results were confirmed by other NURESAFE participants using a CTF open core model with boundary conditions and the neutron physics code COBAYA3.

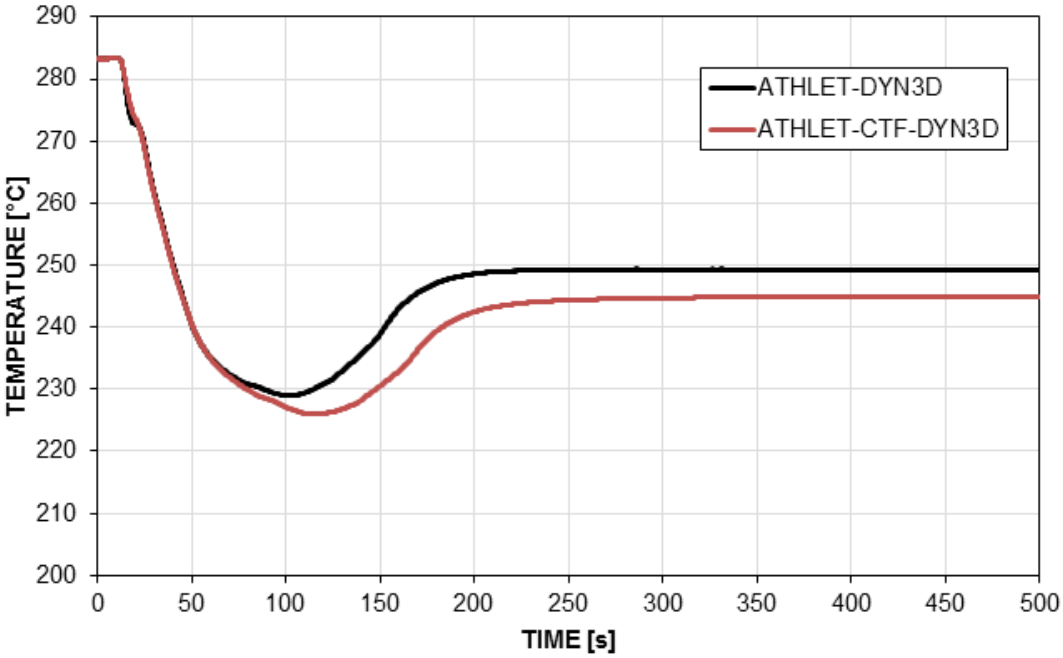

Figure 7.6: Cold Leg Temperature History in the Affected Loop

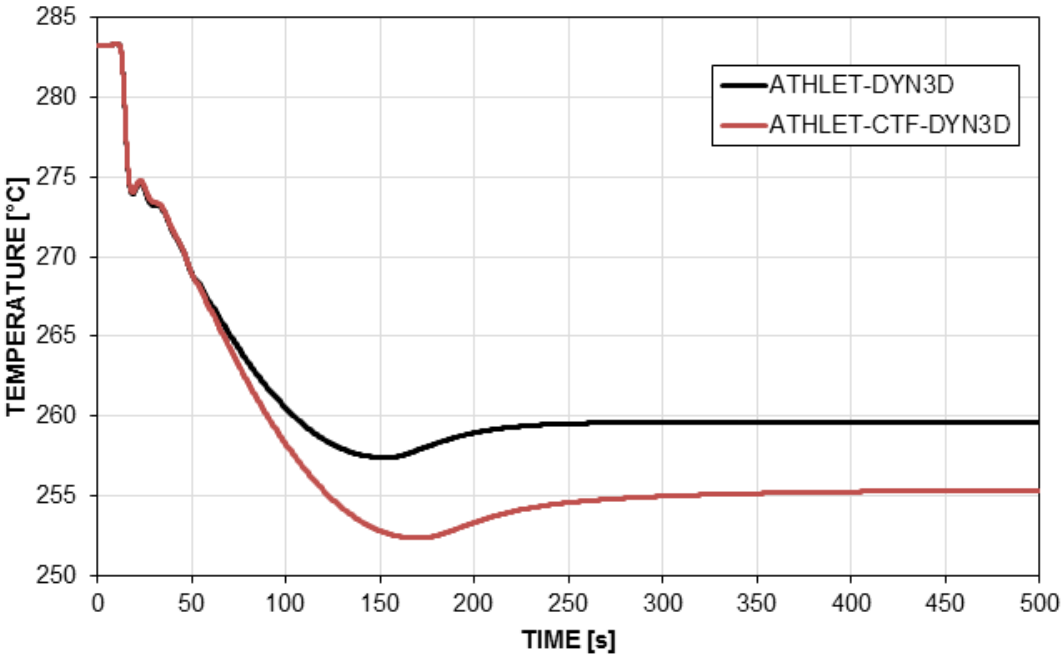

Figure 7.7: Cold Leg Temperature History in the Unaffected Loop

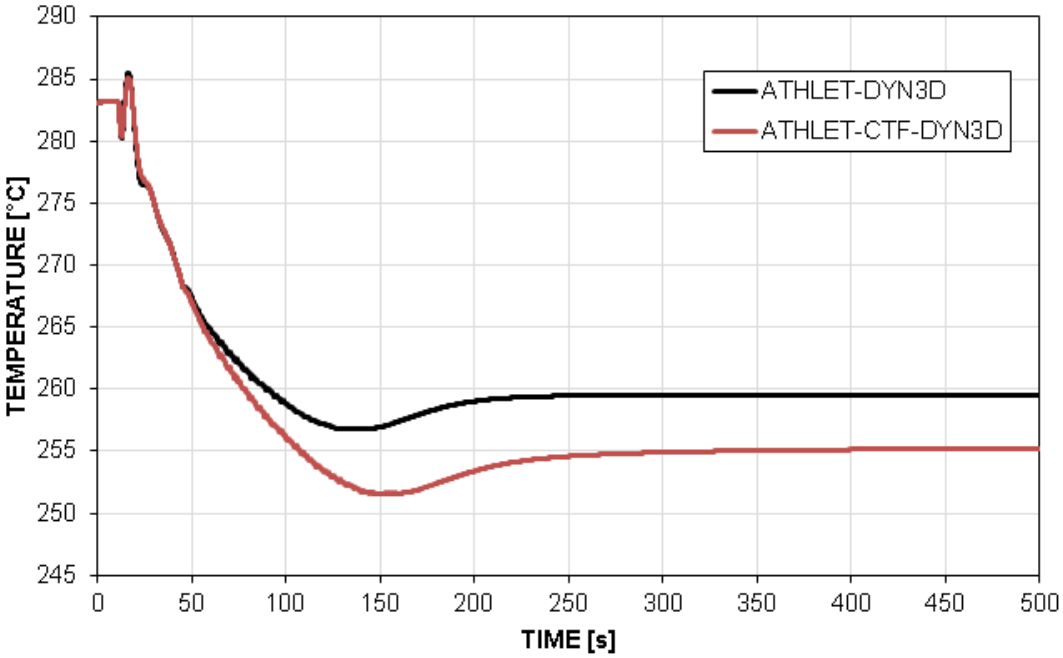

Figure 7.8: Hot Leg Temperature History

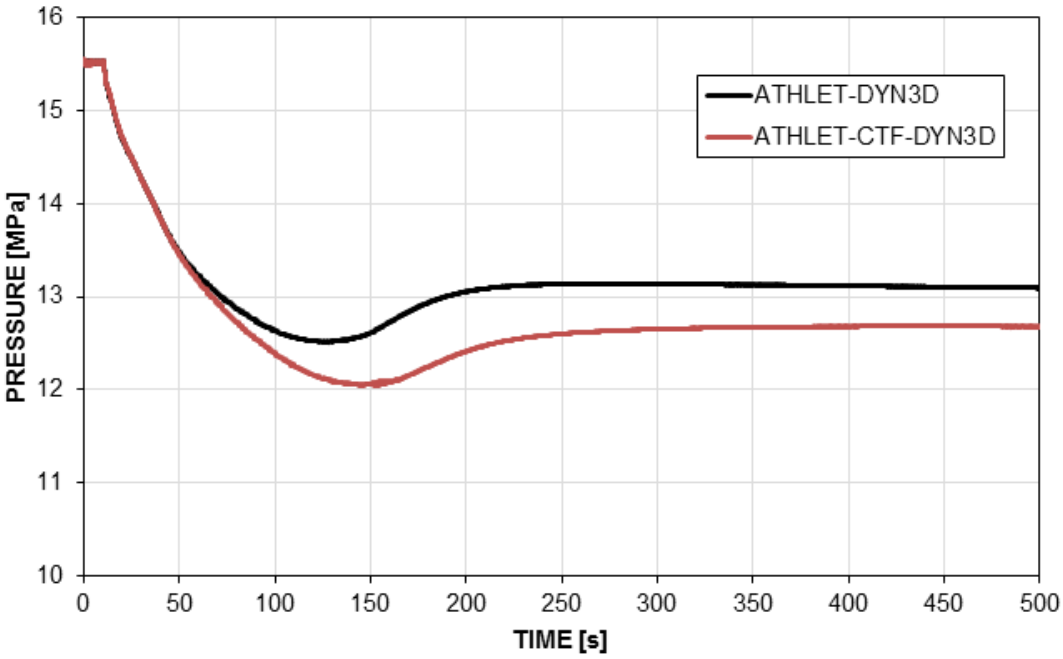

Figure 7.9: Primary Pressure History

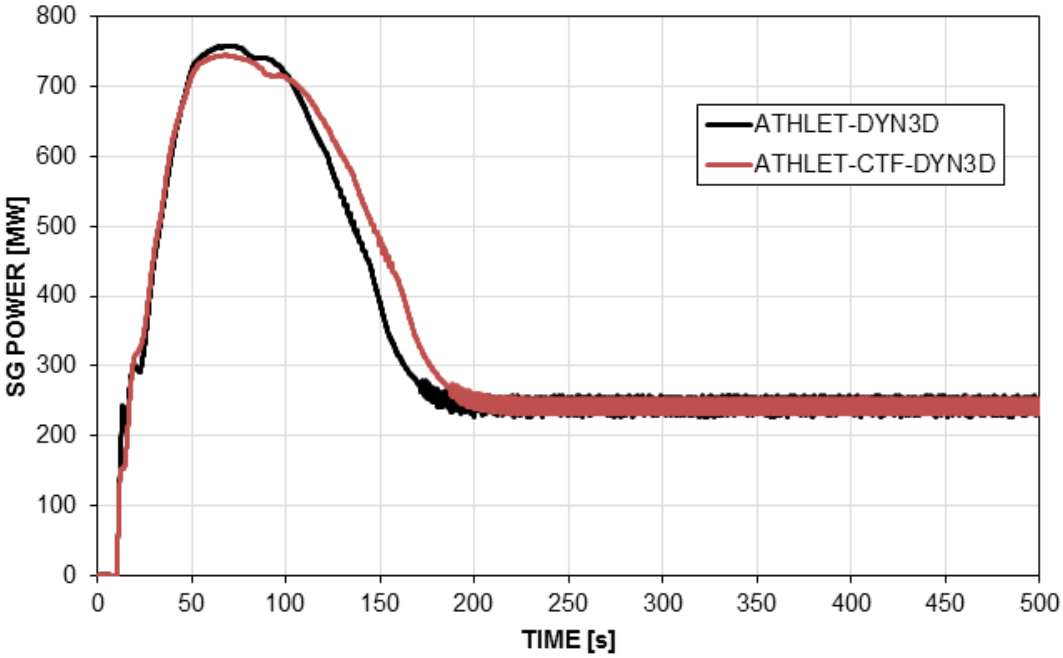

Figure 7.10: Affected Steam Generator Power History

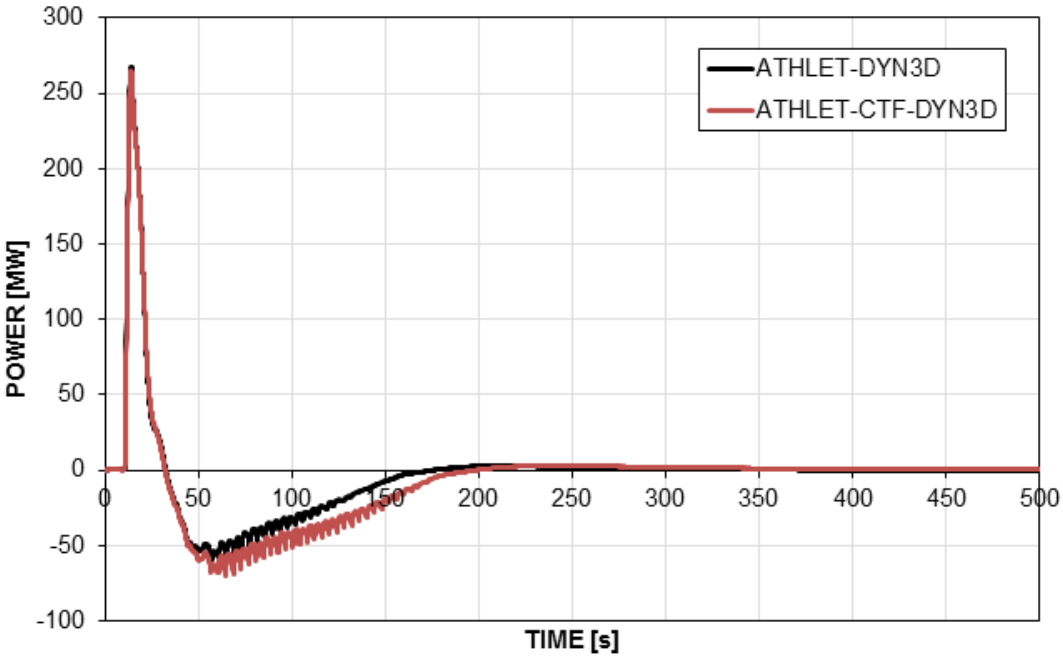

Figure 7.11: Unaffected Steam Generator Power History

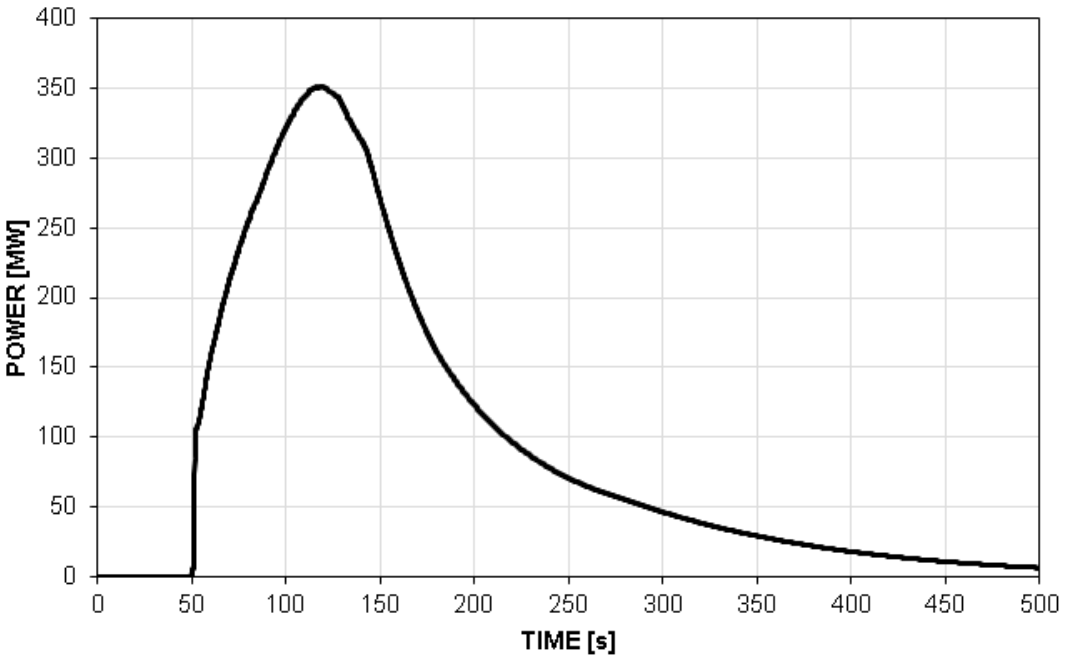

Figure 7.12: Simulation With Mixing Matrix - Fission Power History

This results puts in question the placement of the stuck control rod (i.e. hot channel) which could have been more conservative closer to the core edge.
The second relevant difference is the last phase of the transient. Indeed, at the end of the transient, the power level goes down to 6.1 MW. This is explained by the pressure response on the primary side. The SIS actuation value (12.05 MPa) is reached after 109 s. As a result, the FW pumps are stopped and the SG1 falls dry shortly after. The "second" power peak reaches 351.7 MW at 118 s.
Until the actuation of the SIS signal which stops the feedwater injection, the results agree reasonably well with the two previous models for both primary and secondary side parameters. After this point however, very large differences can be observed as the heat exchanges in all SGs go to 0 and therefore the power in the core also decreases.

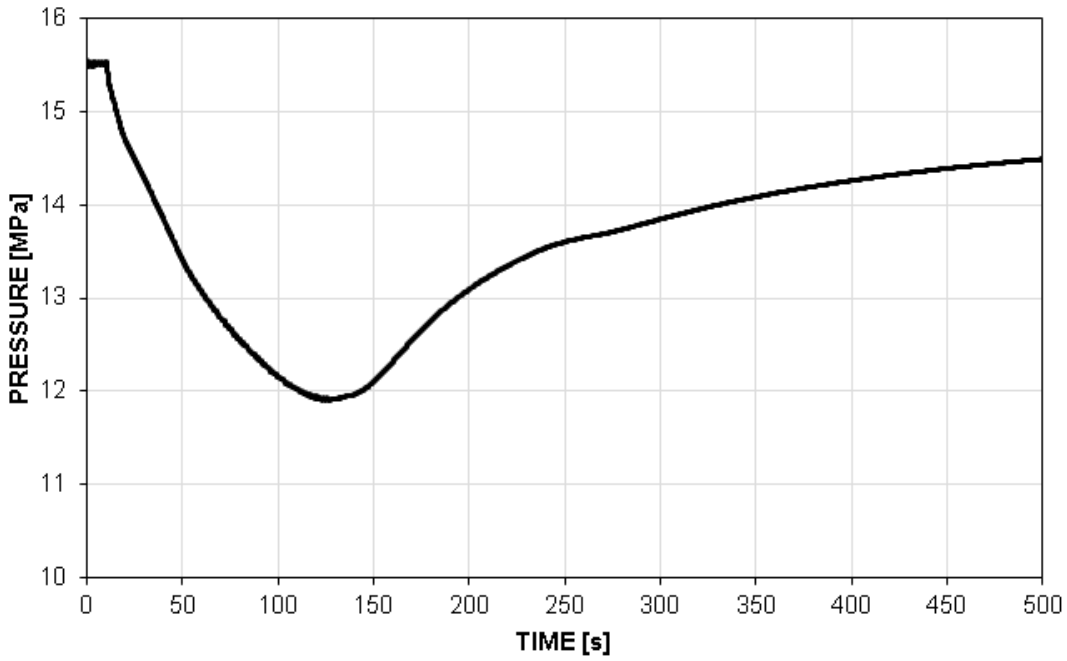

Figure 7.13: Simulation With Mixing Matrix - Primary Pressure History

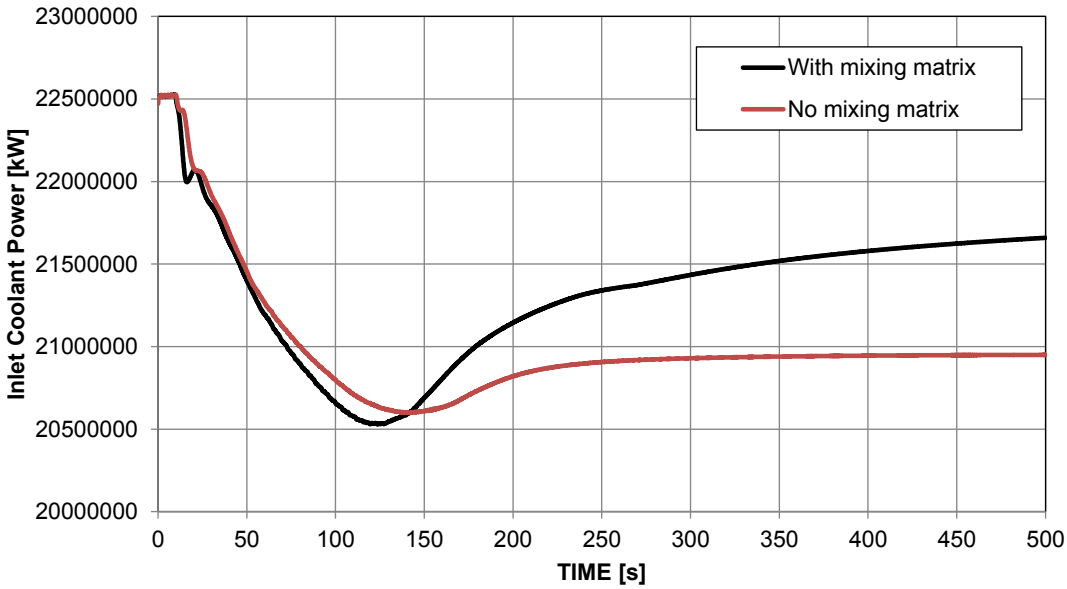

Figure 7.14: Comparison of the Core Inlet Coolant Energy with/without Mixing Matrix

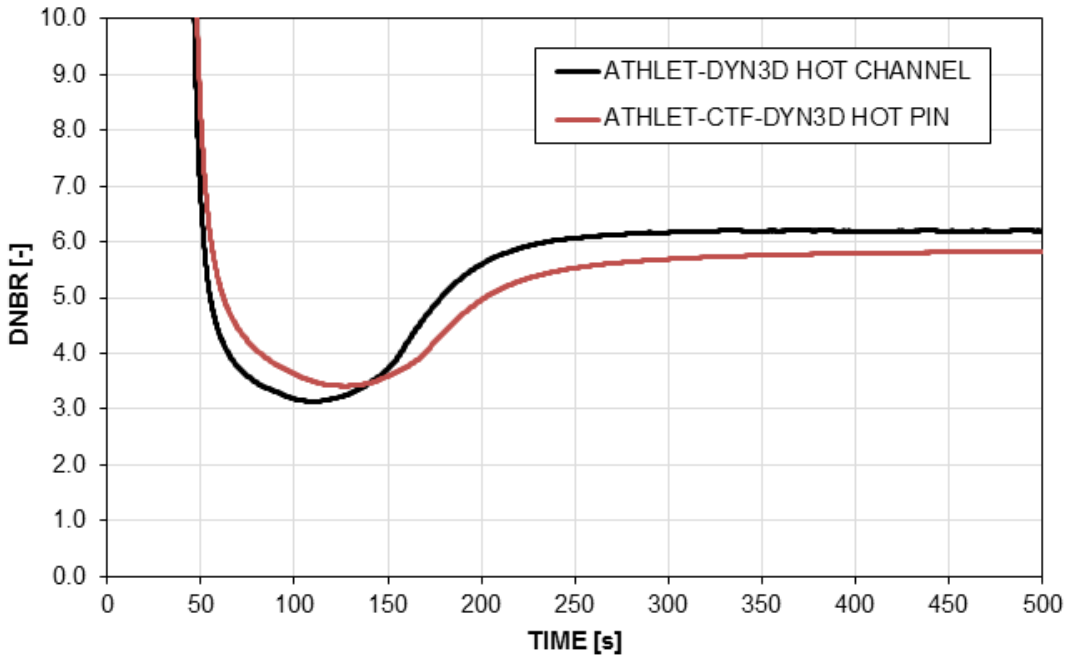

Figure 7.15: Minimum DNB History

7.1.5 Transient Analysis of Local Parameters

In this section the local values of safety parameters are compared. The version with mixing matrix is excluded from the local study because the power level stay quite low during the transient. In the ATHLET-DYN3D model, the hot channel is at the assembly level. The hot channel is channel #6 where the control rod is stuck out. In the ATHLET-CTF-DYN3D (without mixing matrix), the hot channel is at the pin level.

Although the maximum core power is higher in the ATHLET-DYN3D model, the local maximum temperature as well as the minimum DNBR are rather close. This is explained by the refined modeling in CTF with a pin resolution: The relative power in the pin is higher than in the assembly. Nevertheless, the ATHLET-DYN3D model stays more conservative for the local safety parameters.

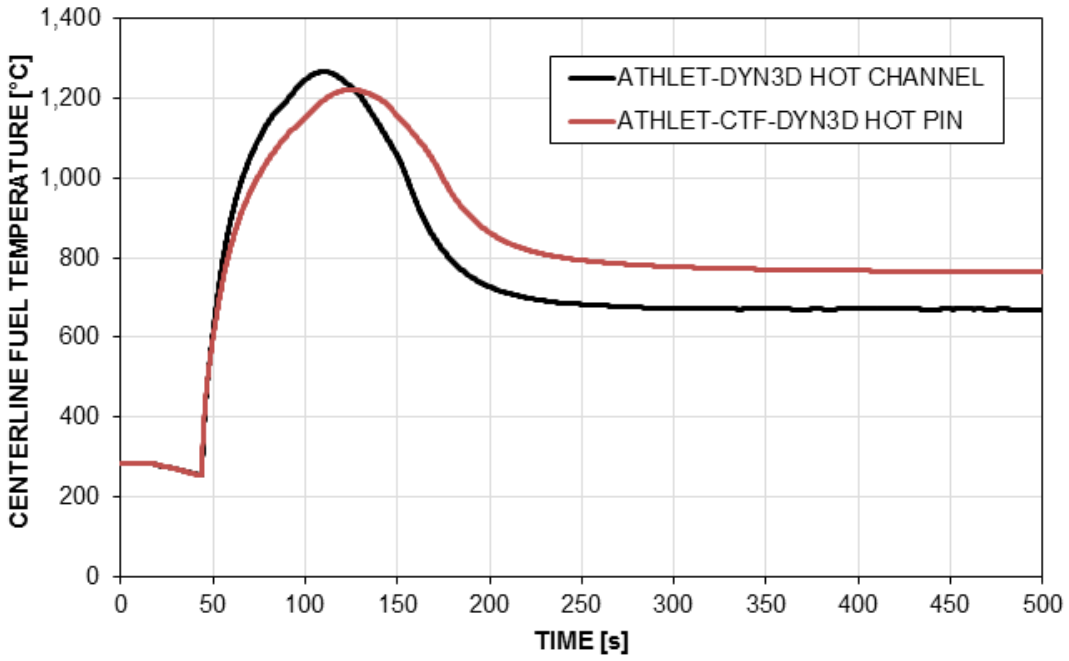

Figure 7.16: Maximum Centerline Fuel Temperature History

7.1.6 Conclusions on the PWR MSLB Transient

The PWR Main Steam Line Break Transient was successfully simulated on the Salomé platform with three coupled code systems. The ATHLET-DYN3D solution is a legacy model using a coarse mapping scheme with superchannels. The first ATHLET-CTF-DYN3D uses the mixing simulated from ATHLET in the downcomer and the lower plenum. The CTF model uses a hybrid (1:1) mapping with a pin by pin resolution in the hot channel. The pin by pin power is calculated in DYN3D using the flux reconstruction method. Finally the third simulation is the same ATHLET-CTF-DYN3D model with the addition of an offline coupling with a CFD code which was used to compute mixing matrix at core inlet.

The ATHLET-DYN3D model is (as expected) the more conservative. However, the difference of the local parameters are much closer, thus showing the influence of a pin level nodalization.

The model using a mixing matrix shows a threshold effect. Indeed, the primary pressure reaches the SIS actuation value which is not the case of the two other

simulations. The SIS actuation causes the feedwater pump trip which changes completely the second part of the transient. This confirms the statement from section 7.1.1 that the mixing phenomena between the affected/unaffected loops are primordial for the simulation of the overcooling transient.

7.2 BWR Application: Turbine Trip Transient under ATWS Conditions

In this section, models and results from simulations performed in the scope of the NURESAFE Work Package 1.3 (WP1.3 BWR ATWS with Uncertainty Quantification) are used. According to the project description [40], *the objective of WP1.3 is to develop and execute simulation schemes using the NURESAFE codes to analyze a BWR ATWS transient coupled with Uncertainty Quantification (UQ).* The Peach Bottom 2 reactor from the OECD/NEA Turbine Trip benchmark was selected for two main reasons. First, an ATWS case was part of the benchmark (Extreme Scenario #2). Moreover, cross-section libraries, as well as core neutronic (DYN3D) and system TH (ATHLET) models were already available to the Work Package participants.

7.2.1 Transient description

The transient is a Turbine Trip (TT) without SCRAM which begins with a sudden Turbine Stop Valve (TSV) closure.
During the first phase of the transient, the nuclear power is driven by the increase in pressure, which causes a void collapse in the core. The positive power response is almost immediate. The transient is slowed down by the feedback from the increased direct and conducted heat flux to the coolant, which produces void and give a negative reactivity feedback. The fuel temperature rise is moderate and the Doppler effect only plays a limited role, compared to the void effect.

While the TSVs are closing, the bypass system valves open, allowing a steam release (i.e. pressure relief). If necessary, SRVs may open at pre-established set points, giving additional pressure relief.
In the case of a normal Turbine Trip transient, the SCRAM is triggered at a defined power level and stops the transient. It is interesting to note that the first power peak is not influenced by the SCRAM actuation, since the power has already begun to decrease when the control rods insertion starts.
In the case of an ATWS however, the reactor SCRAM does not take place. As a result, the reactor power shows an oscillatory behavior driven by the interactions between power and feedback mechanisms. This behavior is interrupted by the opening of the Release and Safety Valves which stabilized the pressure and the power.

7.2.2 Models description

ATHLET Model

The ATHLET input was generated at the GRS as a participant of the OECD/NEA benchmark. The Peach Bottom Unit 2 BWR model is based on the given plant specifications. The whole system is composed of a 33-channel core model including bypass, an upper and a lower plenum, a recirculation loop with pumps, a downcomer, an ideal separator and the steam domes as well as a steam line. A graphical representation of the ATHLET plant model is shown in Figure 7.17. A General Control System (GCSM) is used for the implementation of a steady-state (and transient) control system. The 764 fuel bundles in the reactor core are merged into 33 parallel super-channels according to the bundle types and the radial position (see Figure 7.18). This mapping scheme is the same one developed for the official benchmark results.
Each core channel is nodalized with 26 cells. The first and last cells are the reflector cells. The active part of the core is equally divided into 24 cells and contains Heat Conduction Objects which transfer the heat from the fuel rods into the moderator.

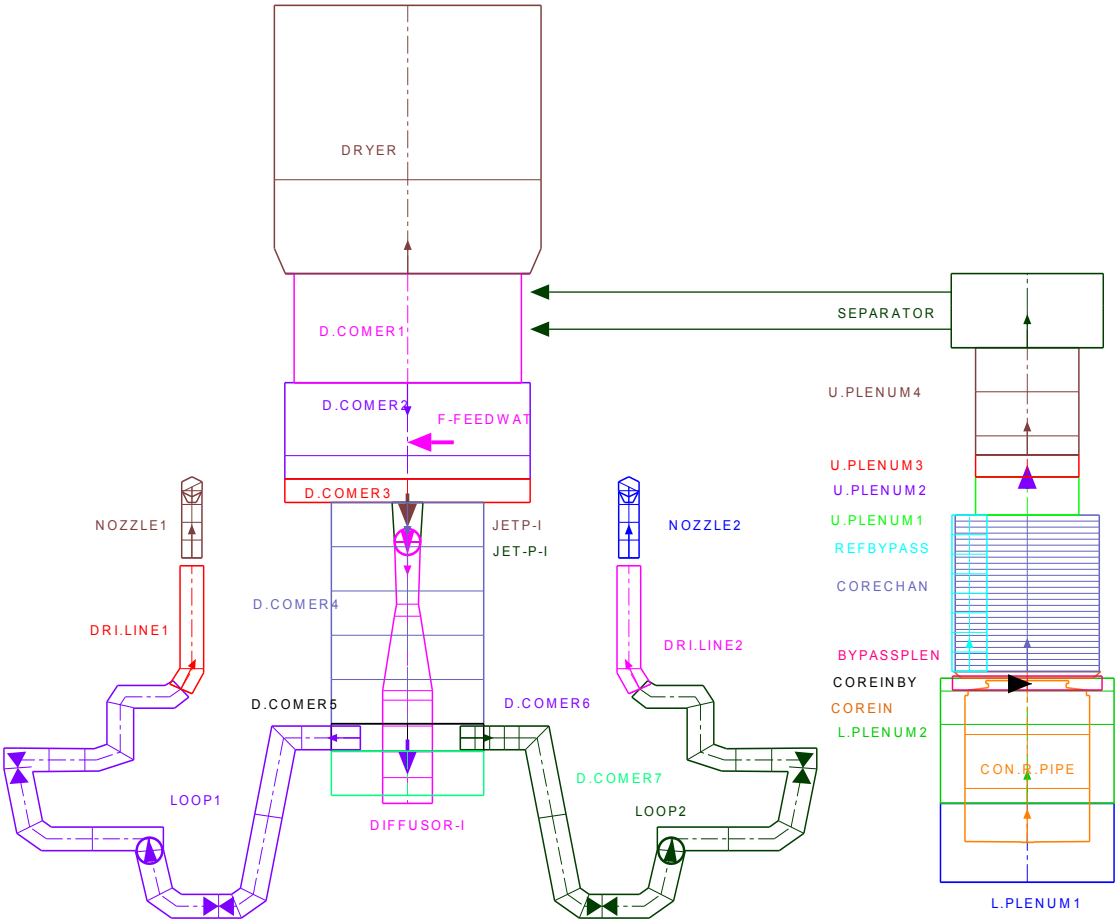

Figure 7.17: Peach Bottom ATHLET Plant Model

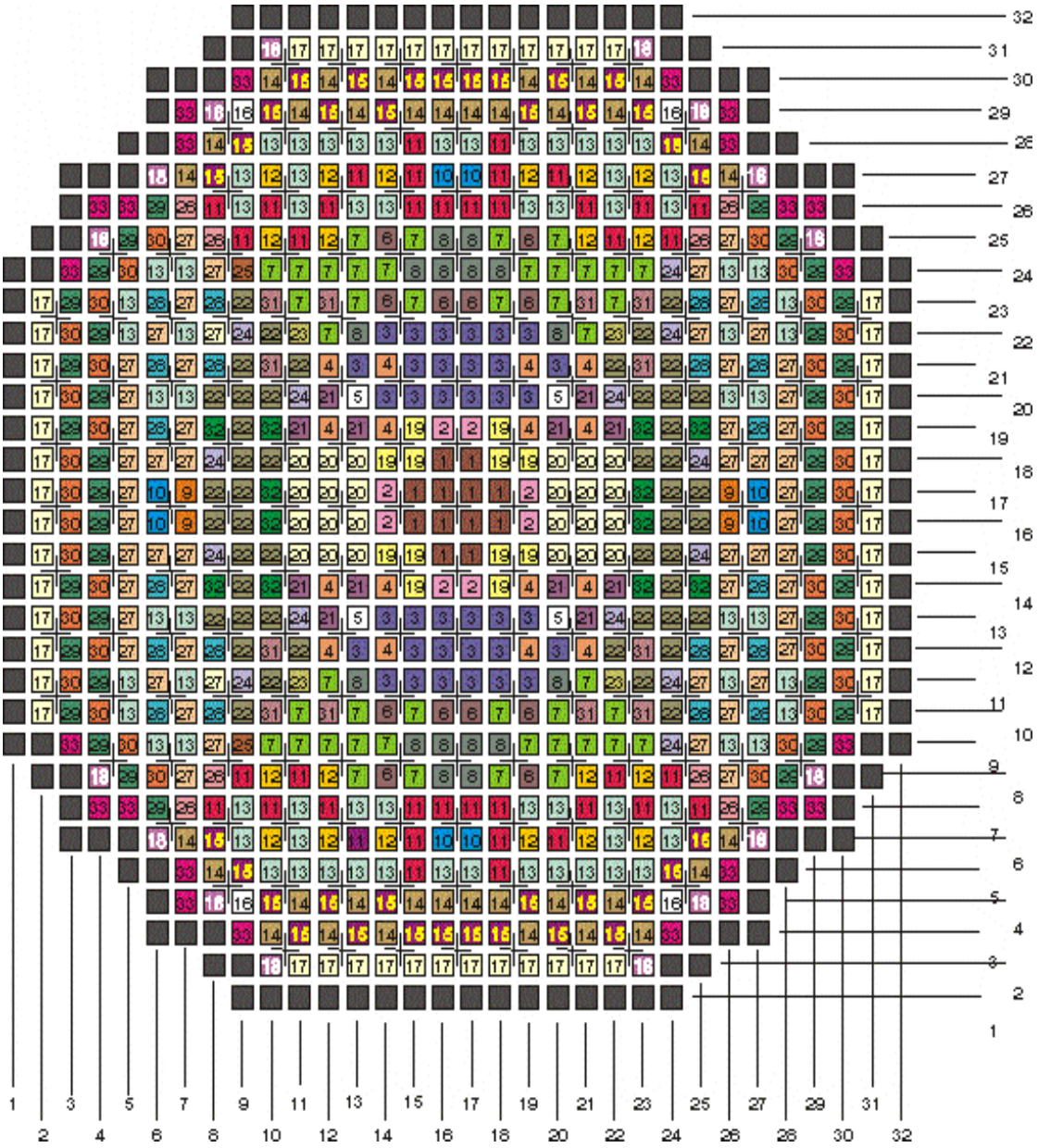

Figure 7.18: 33-Channel ATHLET DYN3D Mapping Scheme

CTF Model

The CTF input was generated at the GRS for the need of the NURESAFE project. The Peach Bottom 2 reactor core contains 764 fuel bundles. The fuel bundles are of three different TH types. In addition, the core is divided into three radial zones with different inlet orifices. In CTF each of the 764 fuel assemblies is modelled explicitely. The flow area, wet perimeter and pressure loss coefficients are all taken from the benchmark specifications. The CTF model is coupled to the ATHLET power plant model and receives the inlet mass flows, inlet temperature as well as core outlet pressure, from the 33 channel model of ATHLET described above.

DYN3D Model

The DYN3D model was built at the HZDR as a participant of the OECD/NEA benchmark. The neutron physics modeling is dictated by the cross-section libraries which were delivered to the benchmark participants. In these libraries, the burnup and the historical parameters are implicitly included in the macroscopic cross-section. Nineteen assembly types are defined where assembly type #19 is the radial reflector. Twenty-four equidistant axial nodes are used in the active core. Two additional nodes are needed for the axial reflector.

7.2.3 Coupling Schemes

Two coupling schemes are are applied. Their results are compared in the next sections.
In the first coupling scheme, the ATHLET model covers the whole BWR system with boundary conditions for the steam line pressure and the feedwater mass flow/temperature. In the core, the 33-channel model from ATHLET is coupled to DYN3D using the mapping scheme shown in Figure 7.18.
In the second coupling scheme, the ATHLET model is the same as the previous one, including the core model. The CTF model covers only the core and is coupled (one-way coupling) with ATHLET at core inlet/exit. The DYN3D model is coupled

to both ATHLET and CTF models but receives the TH feedbacks only from the CTF model (1:1 mapping scheme).
Unlike for the PWR MSLB transient simulation, no pin level coupling is provided. The reason is that no form functions are available for DYN3D flux reconstruction method.
The simulations are performed as described thereafter: First, a 30s seconds of null-transient is computed, during which DYN3D is still in steady-state mode and called every second of simulation. Then during 2s all codes are operating in transient mode but only the regulation systems are activated. Finally, at t=32s, the Turbine Trip Transient starts. In the next sections, t_0 is set to 30s. The turbine trip is thus initiated at t=2s.

7.2.4 Transient Analysis of Integral Parameters

In this section the transient response of some relevant integral parameters is presented. The time dependent behavior of the core power, core averaged fuel temperature, core averaged void content and core exit pressure are represented in Figure 7.19 to 7.21.
In the ATHLET-DYN3D simulation, the maximum power, 9.2 GW, is reached during the first peak 0.7s after the beginning of the transient. In the ATHLET-CTF-DYN3D, the maximum power, 8.4 GW, is reached during the first peak 0.8s after the beginning of the transient. Toward the end of the transient (after 8s), both systems converge toward the same power level which is determined by the pressure in the steam line.
The core void history (inversely) matches the power. This confirms that the void (i.e. moderator density) feedback is the driving force on the neutron physics side. Initially the void is the same in both systems (30%). During the transient an almost constant 1% void difference appears (where ATHLET-DYN3D predicts the higher value).
The fuel temperature response follows the power with a delay, therefore, the first peak is not as pronounced. The amplitude of the changes stays relatively low: less than 100K between the minimum and the maximum. This confirms the statement

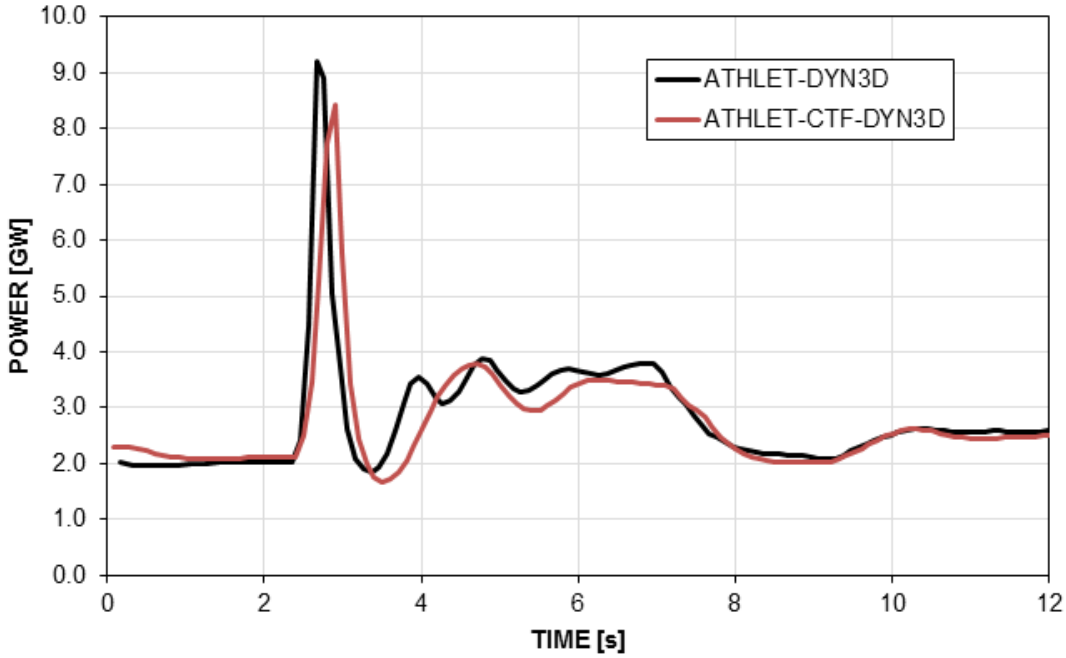

Figure 7.19: Core Power History

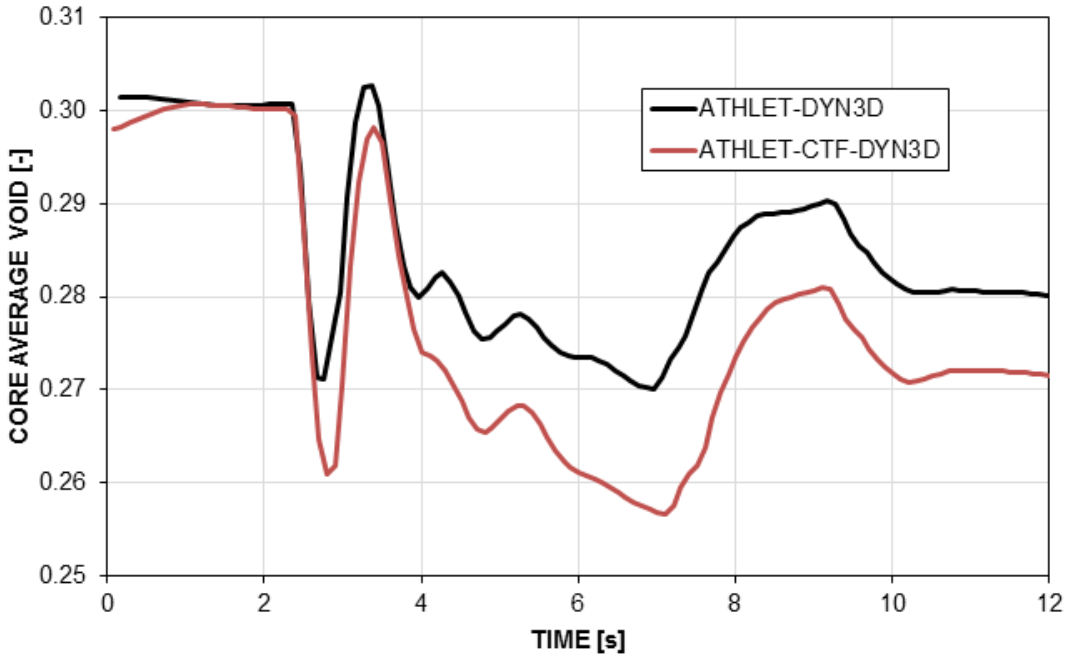

Figure 7.20: Core Average Void History

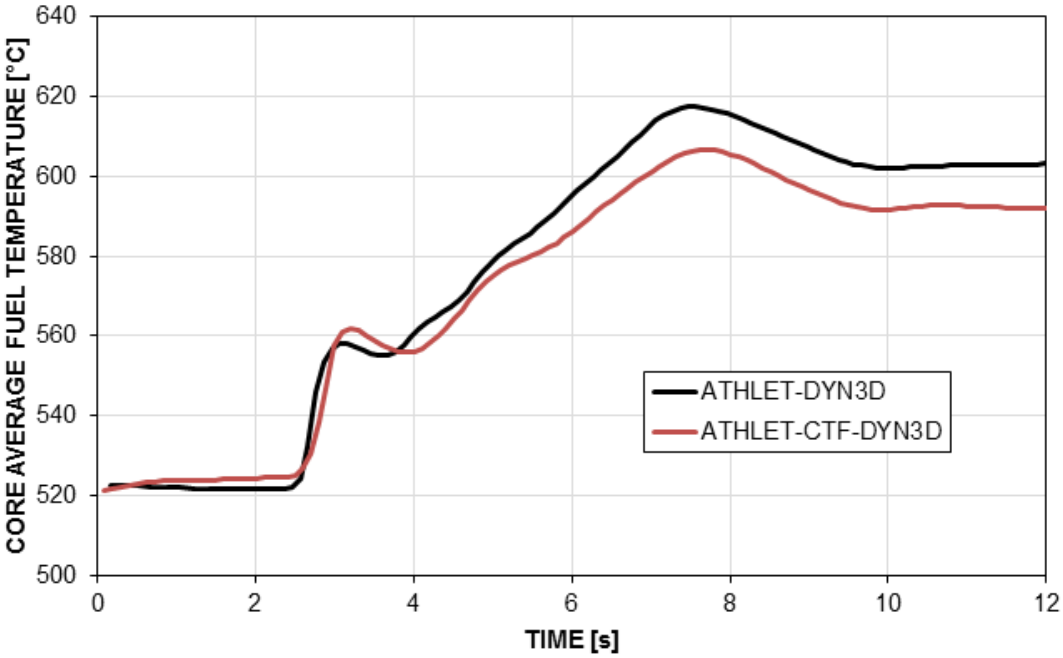

Figure 7.21: Core Average Fuel Temperature History

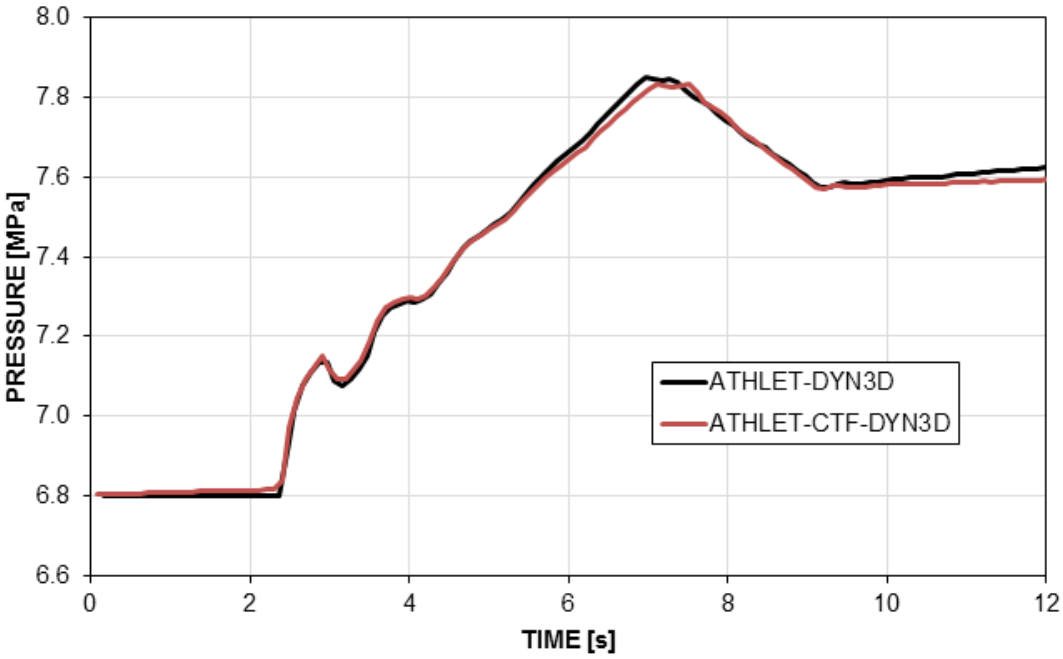

Figure 7.22: Reactor Pressure History

	RV-1		RV-2		RV-3		SRV	
	open [s]	close [s]	open [s]	close [s]	open [s]	close [s]	open [s]	close [s]
ATHLET-DYN3D	4.8	-	5.2	7.2	-	-	-	-
ATHLET-CTF-DYN3D	4.9	-	5.4	7.3	-	-	-	-

Table 7.1: Opening and closing time of the Safety and Release Valves during the ATWS transient

that the Doppler effect only plays a limited role in the first phase of the transient, compared to the void effect. Initially, both coupled systems agree well (around 520°C). During the transient an almost constant 10K difference appears (where ATHLET-DYN3D predicts the higher value).
The opening and closing time of the Safety and Release Valves are shown in Table 7.1. They agree rather well and are consistent with the pressure history found in Figure 7.22.

7.2.5 Transient Analysis of Local Parameters

Unlike for the PWR analysis, no form function was available for the flux reconstruction methods in DYN3D. Therefore, only the assembly level is modeled in CTF.
The maximum power density is reached in the center of the core in channel#1 of the ATHLET model and channel#395 in the CTF model. Channel#1 in ATHLET represents eight fuel assemblies (see Figure 7.18).
Figure 7.23 shows the fuel centerline temperature history during the transient. Although initially at the same power level, the temperature is higher in CTF than in ATHLET (approx. 20K). Unlike for the core averaged temperature, the effect of the first power peak can clearly be seen in both cases. The peak temperature is higher in CTF (1320°C) than in ATHLET (1291°C) although the core peak power is higher in the ATHLET-DYN3D simulation. Although this difference is small (2.2%), this result shows the advantage of a more detailed model for the simulation of local safety parameters.

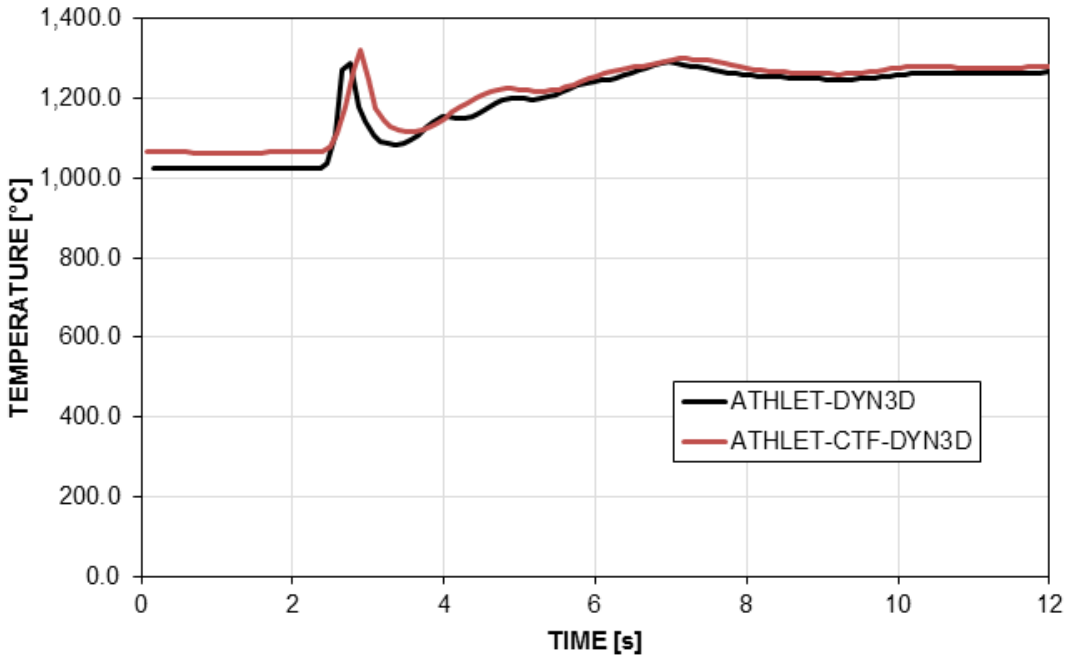

Figure 7.23: Maximum Fuel Temperature History

7.2.6 Conclusions on the BWR Turbine Trip under ATWS Conditions

The Turbine Trip under ATWS condition is the first BWR application of ATHLET-CTF-DYN3D. It is even the first BWR / high void condition application of the ATHLET-CTF parallel coupling. The results agree relatively well for all studied integral parameters.

The maximum power which is the limiting parameter, is 10% higher in the ATHLET-DYN3D case and a time shift of 0.1s is observed. Such deviations are expected and the uncertainty analysis performed on the ATHLET-DYN3D model in the scope of the NURESAFE Project Work Package 1.3 showed that the uncertainty bands cover the ATHLET-CTF-DYN3D results [45].

The good agreement with the ATHLET-DYN3D coupled solution shows the validity of this method even for challenging thermal-hydraulic simulations.

Finally, the analysis of a local safety parameter in the hot channel of both systems, the fuel centerline temperature, shows that the more detailed model is more conservative than the coarser one. This result shows the advantage of a more detailed model for the simulation of local safety parameters.

CHAPTER 8

Conclusions and Suggestions for Future Work

8.1 Conclusions

Initially, when this work started, the multi-physics coupled system ATHLET-QUABOX/CUBBOX was already a mature tool for transient analysis. However, this coupling is made at the fuel assembly level. In order to access safety parameters at a fuel pin level, the use of "hot rod models" with a conservative multiplication factor was necessary. The objective of this work was the development of a multi-physics, multi-scale simulation tool for LWR transient analysis. This tool should provide a better accuracy at the pin level thus reducing the conservatism while keeping the computational costs to reasonable levels for routine transient analysis.

In order to achieve this goal, the subchannel code CTF was implemented in the GRS code suites. In multi-scale coupling schemes, the subchannel code can either simulate the whole core or a subregion of it. For instance, in some licensing relevant cases a "hot channel" analysis might be sufficient. Nevertheless, it is desired that the subchannel code has the capability of calculating a full core model at the assembly level. The development of the coupled system CTF-QUABOX/CUBBOX was thus a preliminary but essential step toward a fully operational multi-physics,

multi-scale simulation tool. This step was successfully achieved and reported in Chapter 3.
In order to model the power distribution at the pin level, several approaches are available. Pin-by-pin reconstruction methods within a neutron diffusion core usually provide a satisfactory accuracy for safety analysis while being fast. This approach has been preferred to more expensive methods which have been investigated in Chapter 4.
In order to model the plant level as well as the pin level, in addition to TH/NP multi-physics couplings, the system code and the sub-channel code also need to be coupled. A one-way parallel coupling approach was selected in order to retain the ATHLET core model and limit the changes in the codes. The results obtained for steady-state and simple transients presented in Chapter 5 were satisfactory. For more challenging transients, the explicit time coupling imposes the use of small time steps, which are already a requirement of the Courant limit criterion found in CTF.

In the framework of the NURESAFE project [40], a fully functional multi-physics multi-scale simulation tool was developed. The TH system code ATHLET, the subchannel code CTF and the neutronics code DYN3D were integrated and coupled on the Salomé platform. This European platform is the product of three consecutive European projects. It can be used for code coupling and 3D visualisation of the results. The developments achieved during this three-year project include assembly level and pin-by-pin TH/NP couplings, as well as a TH/TH coupling between ATHLET and CTF. These couplings, unlike all the other presented in this work, follow the parallel processing approach.
The resulting tri-code coupled system is a multi-physics, multi-scale simulation tool for LWR transient analysis. As proof of principle, it has been tested on two actual safety relevant transients: a PWR Main Steam Line Break and a BWR Turbine Trip under ATWS conditions. The results agree well with the legacy TH system code / diffusion code coupling while providing access to local values at the pin level.

Parallel to the current trend toward more realistic simulation tools (such as multi-scale multi-physics coupled systems or high-fidelity systems), the quantification of the uncertainties is a hot topic. This is the Best Estimate Plus Uncertainty (BEPU) approach. A BEPU analysis was the objective of the NURESAFE WP13 described in the last chapter.
At the GRS, the uncertainty analysis tool SUSA is developed. It has been applied on thermal-hydraulic (for both ATHLET and CTF) as well as coupled analyses (ATHLET- QUABOX/ CUBBOX).
In the NURESAFE project, the URANIE platform was used. This shows that the coupled systems developed during this thesis are compatible with the state-of-art BEPU approach.

8.2 Suggestions for Future Work

The ATHLET-CTF-DYN3D coupled system running on the Salomé developed for this work is now available. The results obtained on actual safety relevant transient for both PWR and BWR reactor types show that it can already be applied for safety analysis.
Several points can be the subject of further developments in order to improve the final product.

In Chapter 6, the development of hexagonal geometry capabilities is described. They are needed for the modeling of VVER reactor type but are also a necessary step for the modeling of Generation IV reactors. For the time being, assembly level simulations are possible for VVER but further development is needed for the pin level.

Recently, CTF has been parallelized using the MPI approach. The Salomé platform supports MPI but at the moment, this capability is not used for CTF. Although the computation overhead from CTF is not prohibitive it is a development which would profit the usability of the system.

In Chapter 4, two other methods for the coupling at the pin level were presented.

The SPH method is a promising method, using diffusion codes with corrected cross-sections to compute directly the pin power distribution. Unlike with the flux reconstruction method, when coupled with a TH code, the feedback can be used at the pin level.

Couplings between subchannel codes (or even CFD codes) with a transport code in a so-called high fidelity simulation tool is a current trend worldwide. The experience with the two couplings of CTF with an S_N transport code and with an MOC transport code has shown a great potential for the future. The prohibitive computation costs currently prevent their use for daily applications. But with the use of parallelization and the continuous development of computational capacities, high-fidelity coupled transient calculations may become the standard in a few years.

Bibliography

[1] F. D'auria et. al. *Neutronics/Thermal-hydraulics Coupling in LWR Technology, Vol. 2.* OECD/NEA, 2004.

[2] A. Grahn, S. Kliem, and U. Rohde. Coupling of the 3D neutron kinetic core model DYN3D with the CFD software ANSYS CFX. *Annals of Nuclear Energy*, 84:193–203, 2015.

[3] J. Lee, H.G. Joo, Y. Perin, and K. Velkov. nTRACER/COBRA-TF Coupling and Initial Assessment. In *Proceedings of Transactions of the Korean Nuclear Society Spring Meeting 2015, Jeju, Korea*, 2015.

[4] Y. S. Jung. *Development of Practical Numerical Nuclear Reactor for High Fidelity Core Analysis.* PhD thesis, Seoul National University, 2013.

[5] B. Kochunas, D. Jabaay, B. Collins, and T. Downar. Coupled thermal-hydraulics and neutronics calculations with COBRA-TF and MPACT. In *Proceedings of Embedded Topical Meeting on Advances in Thermal Hydraulics, ATH 2014, Held at the American Nuclear Society 2014 Annual Meeting*, 2014.

[6] G. Lerchl, H. Austregesilo, P. Schöffel, D. von der Cron, and F. Weyermann. ATHLET Mod 3.1 Cycle A - Models and Methods, 2016.

[7] M. Avramova. COBRA-TF Development, Qualification, and Application to Light Water Reactor Analysis. Master's thesis, The Pennsylvania State University, 2003.

[8] R. Salko and M. Avramova. CTF Theory Manual. Technical report, NCSU, 2016.

[9] S. Langenbuch, W. Maurer, and W. Werner. Coarse mesh flux expansion method for the analysis of space-time effects in large LWR cores. *Nuclear Science and Engineering*, 63:437–456, 1977.

[10] U. Grundmann, U. Rohde, S. Mittag, and S. Kliem. DYN3D Version 3.2 - Code for Calculation of Transients in Light Water Reactors (LWR) with Hexagonal or Quadratic Fuel Elements - Description of Models and Methods, 2005.

[11] U. Rohde, S. Kliem, U. Grundmann, S. Baier, Y. Bilodid, S. Duerigen, E. Fridman, A. Gommlich, A. Grahn, L. Holt, Y. Kozmenkov, and S. Mittag. The reactor dynamics code DYN3D - models, validation and applications. *Progress in Nuclear Energy*, 89, 2016.

[12] A. Seubert, K. Velkov, and S. Langenbuch. The time-dependent 3-d discrete ordinates code TORT-TD with thermal-hydraulic feedback by ATHLET models. In *Proceedings of Physor 2008, Interlaken, Switzerland, September 14-19*, 2008.

[13] R. Min, S.J. Yeon, H.C. Hyun, and H.G. Joo. Solution of the BEAVRS benchmark using the nTRACER direct whole core calculation code. *Journal of Nuclear Science and Technology*, 52, 2015.

[14] J.W. Thomas and H. G. Joo. Assembly Based Modular Ray Tracing and CMFD Acceleration for BWR Cores with Different Fuel Lattices. In *Proceedings of ICAPP 2006, June 4-8, 2006, Reno, Nevada, USA*, 2006.

[15] S. Langenbuch and K. Velkov. Overview on the Development and Application of the Coupled Code System ATHLET-QUABOX/CUBBOX. In *Proceedings of Mathematics and Computation, Supercomputing, Reactor Physics and Nuclear and Biological Applications, Avignon, France*, 2005.

[16] Pressured Water Reactor Main Steam Line Break (MSLB) Benchmark, Volume IV: Results of Phase III on Coupled Core-plant Transient Modelling, 2003.

[17] Boiling Water Reactor Turbine Trip (TT) Benchmark, Volume IV: Summary Results of Exercise 3, 2010.

[18] I. Pasichnyk, Y. Périn, and K. Velkov. Quantitative Uncertainty Analysis of a PWR Control Rod Ejection Accident. In *Proceedings of M&C 2013 - Sun Valley, Idaho, USA - May 5-9*, 2013.

[19] Y. Périn, K. Velkov, and A. Pautz. COBRA-TF / QUABOX-CUBBOX: Code System For Coupled Core and Subchannel Analysis. In *Proceedings of PHYSOR 2010, Pittsburgh, Pennsylvania, USA*, 2010.

[20] T. Kozlowski and T. Downar. OECD/NEA and U.S. NRC PWR MOX/UO2 Core Transient Benchmark, Final Specification, 12 2003.

[21] C.A. Wemplea, H-N.M. Gheorghiu, R.J.J. Stammler, and E.A. Villarino. Recent Advances in the HELIOS-2 Lattice Physics Code. In *Proceedings of International Conference on the Physics of Reactors "Nuclear Power: A Sustainable Resource", Interlaken, Switzerland*, 2008.

[22] Y. Périn, M. Klein, I. Pasichnyk, A. Seubert, K. Velkov, and A. Pautz. Multiscale coupled code systems: from coarse mesh to high-fidelity LWR core calculations. *Kerntechnik*, 76, 2011.

[23] K. Velkov, S. Langenbuch, and T. Kupenova. HYCA-PIN Code - Reconstruction of homogeneous intranodal power distribution. Technical report, GRS, 1993.

[24] X-5 Monte Carlo Team. MCNP Ű A General N-Particle Transport Code, Version 5, 2003.

[25] H. Finnemann and R. Böer and R. Böhm and R. Müller. Evaluation of Safety Parameters in Nodal Space-Time Nuclear reactor Analysis. In *Special-*

ists Meeting on Advanced Methods for Power Reactors, Cadarache, France, September, 10-14, 1990.

[26] J. Hàdek and U. Grundmann. Neutron Flux Reconstruction in a Hexagonal Cassette - Theory and Implementation into the Code DYN3D/H1.1. *Nucleon*, 3, 1997.

[27] A .Hébert and G. Mathonnière. Development of a Third-Generation Superhomogeneisation Method for the Homogenization of a Pressurized Water Reactor Assembly. *Nuclear Science and Engineering*, 115:129–141, 1993.

[28] A. Hébert. A Consistent Technique for the Pin-by-Pin Homogenization of a Pressurized Water Reactor Assembly. *Nuclear Science and Engineering*, 113:227–238, 1993.

[29] T. Courau, M. Cometto, E. Girardi, D. Couyras, and N. Schwartz. Elements of Validation of Pin-by-pin Calculations with the Future EDF Calculation Scheme Based on APOLLO2 and COCAGNE Codes. In *Proceedings of ICAPP 08 Anaheim, California, June 8-12*, 2008.

[30] M. Klein, I. Pasichnyk, A. Pautz, K. Velkov, and W. Zwermann. Accuracy Enhancements of the Coarse-Mesh Diffusion Core Model QUABOX/CUBBOX for Highly Heterogeneous Core Configurations. In *Proceedings of PHYSOR 2010, Pittsburgh, Pennsylvania, USA*, 2010.

[31] M. Christienne, M. Avramova, Y. Périn, and A. Seubert. Coupled TORT-TD/CTF Capability for High-Fidelity LWR Core Calculations. In *Proceedings of PHYSOR 2010, Pittsburgh, Pennsylvania, USA*, 2010.

[32] Y. Périn, A. Seubert, K. Velkov, and A. Pautz. Multi-scale coupled code systems: From coarse-mesh to high-fidelity LWR core calculations. In *Proceedings of the Jahrestagung Kerntechnik, Berlin, 2010*, 2010.

[33] J. Magedanz, Y. Périn, M. Avramova, A. Pautz, F. Puente-Espel, A. Seubert, A. Sureda, K. Velkov, and W. Zwermann. High-Fidelity Multiphysics Simu-

lation of BWR Assembly With Coupled TORT-TD/CTF. In *Proceedings of PHYSOR 2012, Knoxville, Tennessee, USA*, 2012.

[34] Working party on physics of plutonium fuel NEA/NSC and innovative fuel cycles. Physics of Plutonium Fuels BWR MOX Benchmark, 2003.

[35] M.J. Thurgood and et al. COBRA/TRAC-A Thermal Hydraulics Code for Transient Analysis of Nuclear Reactor Vessels and Primary Coolant Systems, 1983.

[36] M. D. Soler-Martinez. Semi-implicit thermal-hydraulic coupling of advanced subchannel and system codes for pressurized water reactor transient applications. Master's thesis, The Pennsylvania State University, 2011.

[37] A. Papukchiev and G. Lerchl. Development of a coupling methodology for the system code ATHLET and the advanced 3D CFD tool ANSYS CFX. In *Proceedings of the Jahrestagung Kerntechnik, Berlin, 2010*, 2010.

[38] J. Herb. Coupling of OpenFOAM with Thermo-Hydraulic Simulation Code ATHLET. In *Proceedings of 9th OpenFOAM Workshop, Zagreb, Croatia*, 2014.

[39] Y. Périn, A. Soler, and K. Velkov. Coupling of the system code ATHLET-QUABOX/CUBBOX with the sub-channel code COBRA-TF. In *Proceedings of The 9th International Topical Meeting on Nuclear Thermal-Hydraulics, Operation and Safety (NUTHOS-9), Kaohsiung, Taiwan*, 2012.

[40] 7th Framework Programme - Simulation Platform for Nuclear Reactor Safety - Grant agreement for: Collaborative project - Annex I - "Description of Work", 2012.

[41] J. Jimenez and V. Sanchez. D33.11.5 - Final report on BWR thermal-hydraulics at system scale and subchannel scale. Technical report, KIT, 2015.

[42] BEMUSE Phase IV Report: Simulation of a LB-LOCA in ZION Nuclear Power Plant, 2008.

[43] F. Bernard, S. Sanchez-Cervera, and N. Garcia-Herranz. D12.23 - Nodal-level XS library parametrized for MLSB. Technical report, IRSN, 2014.

[44] L. Vyskocil. D12.27 - Unsteady Core Inlet Flow Mixing Matrix for PWR MSLB. Technical report, UJV, 2015.

[45] A. Kubarev, S. Roshan, Périn, and J. Jimenez. D13.51 - Report on Best Estimate Plus Uncertainty Analysis of BWR ATWS. Technical report, KTH, 2015.

APPENDIX A

Sample Python Script for ATHLET-CTF-DYN3D

```
#############################################################################
#                Run a ATHLET-CTF-DYN3D coupled calculation.
#                Developed by Y. Perin (January-2015)
#############################################################################
#
# Please setup the NURESIM environment before running
#
#############################################################################
#
#  ------------- USER SECTION -------------
#
# Case dependent variables for the steady state scenario

# Name prefix of the input decks
casename="PBTT_hfp"
#casename="MSLB_hzp"

# Type of the run, steady state or transient
#typecase="steady"
typecase="transient"
```

```
# Choose from where DYN3D is getting the TH feedback
#fb_option=1  # Take it from ATHLET
fb_option=2  # Take it from COBRA-TF

# Enable the mesh printing for COBRATF
do_printMeshes=1
do_damping=0

###############################################################################
#
#-----------------------------------------------
# 1-  Define the libraries and the environment
#-----------------------------------------------

from os import getenv,system,remove,getcwd,path,makedirs
import math
import shutil
from time import *
from sys import exit
output=open('./python_debug_ATHLET_DYN3D_COBRATF.txt', 'w+')

if not getenv("SALOMEPATH"):
        print "Please run inside of Salome!"
        exit(1)
#--------------------
#
import CTF_ORB
#
import ATHLET_ORB
#
import DYN3D_2G_ORB
#
import INTERP_2_5D_ORB
#
import salome
salome.salome_init()
#
import salome_version
```

```
salomeversion = salome_version.getVersions("GUI")[0]
if not salomeversion: salomeversion = salome_version.getVersions("
    KERNEL")[0]
msg = "Salome version "+str(salomeversion)+" found"
print msg
#
from MEDCouplingClient import *
from MEDLoader import MEDLoader
#
import steady_post # visu post functions
#
# Shutdown containers if present
from orbmodule import client
from string import split

print "-----------------------------------------"
print "Initialization of script done :-) "
print "-----------------------------------------"
###############################################################################
#-----------------------------------------------------------------------------
# 2- Initialize paths
#-----------------------------------------------------------------------------

def main():

    c=client()
    machine=split(getenv("HOSTNAME"),'.')[0]
    for pfix in ["ATHLET","CTF","I25"]:
        #
        cont=c.Resolve("/Containers/"+machine+"/FactoryServer"+pfix)
        if(cont!=None): cont.Shutdown()
    #
    a30=salome.lcc.FindOrLoadComponent("FactoryServerATHLET","ATHLET"
        )
    ctf=salome.lcc.FindOrLoadComponent("FactoryServerCTF","CTF")
    d3d=salome.lcc.FindOrLoadComponent("FactoryServerDYN3D_2G","
        DYN3D_2G")
```

```
i25d_a=salome.lcc.FindOrLoadComponent("FactoryServerI25_ATHLET","
    INTERP_2_5D")
i25d_c=salome.lcc.FindOrLoadComponent("FactoryServerI25_CTF","
    INTERP_2_5D")

# Clean Environment
system("rm -rf *.med")

# Name of the input deck for COBRATF and ATHLET
if casename=="PBTT_hfp":
    geom=0
    boron=False
    d3d_xorig=0.0 ; d3d_yorig=0.0 ; d3d_zorig=0.0
    ctf_xorig=0.1524 ; ctf_yorig=0.1524 ; ctf_zorig=0.0
    # Minimum number of CTF iteration for steady state convergence
    minitr=300
    if typecase=="steady":
        CTF_file="deck.inp"
        ATHLET_file="PBTT_hfp.inp"
    elif typecase=="transient":
        CTF_file="deck-TRAN.inp"
        ATHLET_file="PBTT_hfp.inp"
    else:
        exit("Case not allowed, check input parameters: casename & 
            typecase")
elif casename=="MSLB_hzp":
    geom=0
    boron=False
    d3d_xorig=0.0 ; d3d_yorig=0.0 ; d3d_zorig=0.0
    ctf_xorig=0.2142 ; ctf_yorig=0.2142 ; ctf_zorig=0.0
    # Minimum number of CTF iteration for steady state convergence
    minitr=100
    if typecase=="steady":
        CTF_file="deck.inp"
        ATHLET_file="MSLB_hzp.inp"
    elif typecase=="transient":
        CTF_file="deck.inp"
        ATHLET_file="MSLB_hzp.inp"
```

```
    else:
        exit("Case not allowed, check input parameters: casename & typecase")
else:
    exit("Case not allowed, check input parameters: casename & typecase")

if do_damping :
    damp=0.3  # Damping for the SS iterations between new and old values
    import copy

############################################################
bar = "
    ------------------------------------------------------------------------------
    "
msg="     Starting coupled ATHLET-DYN3D-COBRATF "+ typecase +" calculation of "+casename
print bar
print
print msg
print
print bar

# Ressources of test base
ressourcedir=getenv("NURESAFE_TEST_DATA")

## DYN3D PARAMETERS
# Directory where the DYN3D input files are located and the calculation will performed inside
DYN3D_init_dir = ressourcedir + "/data/dyn3d/" + casename + "/" + typecase
DYN3D_work_dir = DYN3D_init_dir + "/results"
DYN3D_med_file = DYN3D_work_dir + "/DYN3Dresult.med"
# Intialize input files for DYN3D
system("mkdir -p "+ DYN3D_work_dir)
system("rm -rf " + DYN3D_work_dir + "/* ")
system("cp -rp " + DYN3D_init_dir+"/*.dat " + DYN3D_work_dir)
```

```
if casename=="recmini25_hfp":
  system("ln -s " + DYN3D_init_dir+"/XSLIB_110103 " +
      DYN3D_work_dir)
if casename=="MSLB_hzp":
  system("ln -s " + DYN3D_init_dir+"/XSLIB_27 " + DYN3D_work_dir)
if path.exists(DYN3D_med_file): remove(DYN3D_med_file)

## COBRATF PARAMETERS
CTF_in    = ressourcedir + "/data/cobratf/" + casename + "/" +
    typecase
CTF_out   = getenv("PWD") # CTF_in + "/results"
CTF_mesh  = getenv("PWD") + "/COBRATFMESH.med"
CTF_mesh2= getenv("PWD") + "/COBRATFSTRUCTURE.med"
system("ln -sf " + CTF_in + "/" + CTF_file + " deck.inp")
# Create the output folder if not existing
if path.exists(CTF_mesh): remove(CTF_mesh)
if path.exists(CTF_mesh2): remove(CTF_mesh2)

## ATHLET PARAMETERS
ATHLET_in    = ressourcedir + "/data/athlet/" + casename + "/" +
    typecase
ATHLET_out   = ATHLET_in + "/results"
ATHLET_mesh  = getenv("PWD") + "/ATHLETMESH.med"
ATHLET_mesh2= getenv("PWD") + "/ATHLETSTRUCTURE.med"
# Create the output folder if not existing
shutil.rmtree(ATHLET_out,True)
makedirs(ATHLET_out)
if path.exists(ATHLET_mesh): remove(ATHLET_mesh)
if path.exists(ATHLET_mesh2): remove(ATHLET_mesh2)

t_start=time()
####################
# Init the engines
####################

####################
# Init DYN3D
```

```
# Setup for coupling with other TH code
if fb_option ==1: d3d.SetUseATHLETCoupling()
if fb_option ==2: d3d.SetUseFLICACoupling()

# Result MED file writting
d3d_res_med=DYN3D_med_file # Path to the result file
d3d.WriteResultsInFile(d3d_res_med)
# Specify the parameter in MED file
d3d.SetResultParameter("POWERW")
d3d.SetResultParameter("TF")
d3d.SetResultParameter("TM")
d3d.SetResultParameter("DM")
d3d.SetCoreMeshRotation(0.0)
d3d.SetCoreMeshTranslation(d3d_xorig,d3d_yorig,d3d_zorig)
if boron: d3d.SetResultParameter("CB")

# Init the engine
workdir=DYN3D_work_dir # directory where the DYN3D input files
    are located and the calculation will performed inside
print "DYN3D work dir  :",DYN3D_work_dir
print "DYN3D input case:",casename
d3d.InitCalc(DYN3D_work_dir,casename)

# Get the core mesh for the interpolator setup
d3d_mesh_name="DYN3DMESH"
# Write the core mesh in med file for debugging
d3d_mesh_med=DYN3D_med_file
d3d.WriteCoreMeshInFile(d3d_mesh_med)
print "-----------------------------------"
print "Initialization of DYN3D DONE"
print "-----------------------------------"
####################

####################
# Init ctf
print "COBRATF input dir: ",CTF_in
print "COBRATF Output dir:",CTF_out
print "COBRATF Input file:",CTF_file
```

```
ctf.SetCoreMeshTranslation(ctf_xorig,ctf_yorig,ctf_zorig)
s1=ctf.initialize()
ctf.genMeshCTF(geom,0)
print "Initializing CTF",s1

print "------------------------------------------"
print "Initialization of COBRATF DONE"
print "------------------------------------------"

###################
# Init ATHLET
print "ATHLET input dir: ",ATHLET_in
print "ATHLET Output dir:",ATHLET_out
print "ATHLET Input file:",ATHLET_file
print ATHLET_in+"/"+ATHLET_file
a30.SetTHCoupling()
s1=a30.a_initsample(ATHLET_in+"/"+ATHLET_file,casename,"run","",
    ATHLET_out,"")
s1=a30.a_Input()
print "Initializing ATHLET",s1

print "------------------------------------------"
print "Initialization of ATHLET DONE"
print "------------------------------------------"
###################


###################
# Get the meshes
d3d_mesh_a=d3d.GetCoreMesh()
d3d_mesh_c=d3d.GetCoreMesh()
a30_mesh=a30.getCoreMeshATHLET()
a30_porosity=a30.getPorosity()
a30_weight=a30.getWeight()
a30_islocated=a30.getIslocated()
ctf_fmesh=ctf.getFluidMeshCTF()
ctf_rmesh=ctf.getRodMeshCTF()
ctf_porosity=ctf.getOutputMEDField("porosity")
```

```
ctf_weight=ctf.getOutputMEDField("weight")
ctf_islocated=ctf.getOutputMEDField("islocated")

#######################################################################
# Verify that the meshes are in the right position, dump to MED
    file
if do_printMeshes:
  d3dm_c = MEDCouplingUMeshClient.New(d3d_mesh_c)
  athletm_c = MEDCouplingUMeshClient.New(a30_mesh)
  ctf_fm_c = MEDCouplingUMeshClient.New(ctf_fmesh)
  ctf_rm_c = MEDCouplingUMeshClient.New(ctf_rmesh)
  MEDLoader.WriteUMesh("COUPLING_MESHES.med",d3dm_c, True)
  MEDLoader.WriteUMesh("COUPLING_MESHES.med",athletm_c, False)
  MEDLoader.WriteUMesh("COUPLING_MESHES.med",ctf_fm_c, False)
  MEDLoader.WriteUMesh("COUPLING_MESHES.med",ctf_rm_c, False)
  print "
     ------------------------------------------------------------
     "
  print "Open the COUPLING_MESHES.med in POST-PRO/PARAVIS for
     debugging"
  print "
     ------------------------------------------------------------
     "
#######################################################################

##################
# Init interpolator
i25d_a.setWorkingDirectory(ATHLET_in+"/INTERP_2_5D");
i25d_a.setOptions(1.0E-8,4,1)
i25d_a.prepareMatrices(d3d_mesh_a,a30_mesh,a30_mesh,a30_porosity,
    a30_weight,a30_islocated)
i25d_c.setWorkingDirectory(CTF_in+"/INTERP_2_5D");
i25d_c.setOptions(1.0E-8,4,1)
i25d_c.prepareMatrices(d3d_mesh_c,ctf_fmesh,ctf_rmesh,
    ctf_porosity,ctf_weight,ctf_islocated)
##################
print "-----------------------------"
print "Initialization of INTERP DONE"
```

```
print "-----------------------------------"

##########################
# Do the steady-state
##########################
ss_iter=0
status=0
while(status==0):
  # Do DYN3D iteration
  status=d3d.ComputeSteadyStateStep()
  if status>0:
      #
      print "Error in DYN3D steady-state"
      break

  # Get power distribution from DYN3D
  power_a=d3d.GetCorePowerField(0)
  power_c=d3d.GetCorePowerField(0)
  ptot=d3d.GetTotalPower()
  print "Power:",ptot
  print "Keff:",str(d3d.GetKeff())

  # Interpolate Power field to both TH codes
  (ffluid_a,ffuel_a)=i25d_a.projectPower(power_a)
  a30.setInputMEDField("pow",ffluid_a)
  if (do_damping):
     if ss_iter >=2 :
        # Apply damping
        power_c=i25d_c.applyDamp(damp,power_c,power_cpre)
     # Backup previous iteration step
     power_cpre=copy.deepcopy(power_c)
  (ffluid_c,ffuel_c)=i25d_c.projectPower(power_c)
  ctf.setInputMEDField("power",ffuel_c)

  if salome:
     ffluid_a.UnRegister()
     ffuel_a.UnRegister()
     ffluid_c.UnRegister()
```

```
    ffuel_c.UnRegister()

# Do ATHLET iteration
a30.a_Steady()

###################
# Give BC to CTF
massflow=a30.getOutputMEDField("inlet_massflow")
pressure=a30.getOutputMEDField("outlet_pressure")
enthalpy=a30.getOutputMEDField("inlet_enthalpy")

ctf.setInputMEDField("inlet_massflow",massflow)
ctf.setInputMEDField("inlet_enthalpy",enthalpy)
ctf.setInputMEDField("outlet_pressure",pressure)

# Print the fields that are being exchanged.
#############################################################
if do_printMeshes:
  inlet_mesh= enthalpy.getMesh()
  outlet_mesh= pressure.getMesh()
  inlet_mesh_c=MEDCouplingUMeshClient.New(inlet_mesh)
  outlet_mesh_c=MEDCouplingUMeshClient.New(outlet_mesh)
  massflow_c=MEDCouplingFieldDoubleClient.New(massflow)
  massflow_c.setIteration(-1)
  massflow_c.setOrder(-1)
  pressure_c=MEDCouplingFieldDoubleClient.New(pressure)
  pressure_c.setIteration(-1)
  pressure_c.setOrder(-1)
  enthalpy_c=MEDCouplingFieldDoubleClient.New(enthalpy)
  enthalpy_c.setIteration(-1)
  enthalpy_c.setOrder(-1)
  MEDLoader.WriteUMesh("CTF_ATHLET_IN_OUT_MESH.med",
      inlet_mesh_c, True)
  MEDLoader.WriteFieldUsingAlreadyWrittenMesh("
      CTF_ATHLET_IN_OUT_MESH.med",massflow_c)
  MEDLoader.WriteFieldUsingAlreadyWrittenMesh("
      CTF_ATHLET_IN_OUT_MESH.med",enthalpy_c)
```

```
    MEDLoader.WriteUMesh("CTF_ATHLET_IN_OUT_MESH.med",
        outlet_mesh_c, False)
    MEDLoader.WriteFieldUsingAlreadyWrittenMesh("
        CTF_ATHLET_IN_OUT_MESH.med",pressure_c)
    print "
        ------------------------------------------------------------------------
        "
    print "Open the CTF_ATHLET_IN_OUT_MESH.med in POST-PRO/
        PARAVIS for debugging"
    print "
        ------------------------------------------------------------------------
        "
#
    ##############################################################################


# Do CTF iteration
conv = False
itctf = 0
while conv==False:
    itctf += 1
    dt=ctf.computeTimeStep()
    ctf.initTimeStep(dt)
    success=ctf.solveTimeStep()
    if success:
      ctf.validateTimeStep()
      #print "Iteration number ", itctf, " Time step used was: ",
          dt
    else:
      ctf.abortTimeStep()
    conv=ctf.isStationary()
    if itctf <= minitr:
      conv = False

# Transfer the feedback data
if fb_option==1:
    ft=a30.getOutputMEDField("fuel_temperature")
    mt=a30.getOutputMEDField("moderator_temperature")
```

```
      md=a30.getOutputMEDField("moderator_density")
      if boron: cb=a30.getOutputMEDField("boron_concentration")
      ift=i25d_a.projectThField(ft)
      imt=i25d_a.projectFluidField(mt)
      imd=i25d_a.projectFluidField(md)
      if boron: icb=i25d_a.projectFluidField(cb)
    else:
      ft=ctf.getOutputMEDField("fuel_temperature")
      mt=ctf.getOutputMEDField("moderator_temperature")
      md=ctf.getOutputMEDField("moderator_density")
      if boron: cb=ctf.getOutputMEDField("boron") # TODO ENABLE
          BORON TRANSPORT IN COBRA-TF
      ift=i25d_c.projectThField(ft)
      imt=i25d_c.projectFluidField(mt)
      imd=i25d_c.projectFluidField(md)
      if boron: icb=i25d_c.projectFluidField(cb)

    #Set feedback in DYN3D
    d3d.SetFuelTemperatureField(ift)
    d3d.SetModeratorTemperatureField(imt)
    d3d.SetModeratorDensityField(imd)
    if boron: d3d.SetBoronConcentrationField(icb)

    if salome:
        ift.UnRegister()
        imt.UnRegister()
        imd.UnRegister()
        if boron: icb.UnRegister()

    #Counter over iterations
    ss_iter+=1
    print str(ss_iter), "CTF internal iterations= ", str(itctf)

####################
# Do the transient
####################
if typecase=="transient":
```

```
###################
# Calculation
status=0
ttime=0.0  #Current time
Tend=52.0  #Ending time
zerotrans=1.0
tran_iter=1

while(ttime<=Tend):

  # Initialize transient mode
  if tran_iter==1:
    a30.a_Trans01()
    ctf.transientMode()

  # Chose minimum time step size between ATHLET and CTF
  dta=a30.a_GetTimeStep()
  dtc=ctf.computeTimeStep()
  if dta > dtc:
    dt=dtc
    #print "dt was changed in ATHLET"
  if dtc > dta:
    dt=dta
    print "dt was changed in CTF"
  a30.a_SetTimeStep(dt)
  ctf.initTimeStep(dt)
  ttime+=dt

  # Perform ATHLET time step
  a30.a_Trans_Loop(0,ttime)

  # Give BC to CTF
  massflow=a30.getOutputMEDField("inlet_massflow")
  pressure=a30.getOutputMEDField("outlet_pressure")
  enthalpy=a30.getOutputMEDField("inlet_enthalpy")
  ctf.setInputMEDField("inlet_massflow",massflow)
  ctf.setInputMEDField("inlet_enthalpy",enthalpy)
  ctf.setInputMEDField("outlet_pressure",pressure)
```

```
# Do CTF time step
success=ctf.solveTimeStep()
if (success == False):
   #
    print "Error in CTF transient"
    break
if success:
    ctf.validateTimeStep()

# Transfer the feedback data
if fb_option==1:
  ft=a30.getOutputMEDField("fuel_temperature")
  mt=a30.getOutputMEDField("moderator_temperature")
  md=a30.getOutputMEDField("moderator_density")
   if boron: cb=a30.getOutputMEDField("boron_concentration")
   ift=i25d_a.projectThField(ft)
   imt=i25d_a.projectFluidField(mt)
   imd=i25d_a.projectFluidField(md)
   if boron: icb=i25d_a.projectFluidField(cb)
else:
   ft=ctf.getOutputMEDField("fuel_temperature")
   mt=ctf.getOutputMEDField("moderator_temperature")
   md=ctf.getOutputMEDField("moderator_density")
   if boron: cb=ctf.getOutputMEDField("boron")
   ift=i25d_c.projectThField(ft)
   imt=i25d_c.projectFluidField(mt)
   imd=i25d_c.projectFluidField(md)
   if boron: icb=i25d_c.projectFluidField(cb)

#Set feedback in DYN3D
d3d.SetFuelTemperatureField(ift)
d3d.SetModeratorTemperatureField(imt)
d3d.SetModeratorDensityField(imd)
if boron: d3d.SetBoronConcentrationField(icb)

# Do DYN3D iteration
if (ttime >= zerotrans):
```

```
            status=d3d.ComputeSteadyStateStep()
            keff = d3d.GetKeff()
            print "Keff=",keff
            status=0
            zerotrans=zerotrans+1.0
        if (zerotrans >= 31.0):
            zerotrans=10000.0
            status=d3d.ComputeTransientStep(dt)
        if status>0:
            #
            print "Error in DYN3D transient"
            break
        # Get power distribution from DYN3D
        power_a=d3d.GetCorePowerField(0)
        power_c=d3d.GetCorePowerField(0)
        ptot=d3d.GetTotalPower()
        print "%-20s%-20f%-20f" %("Time:",ttime,ptot)
        print >> output, ("%-20s%-20f%-20f" %("Time:",ttime,ptot))

        # Interpolate Power field to both TH codes
        (ffluid_a,ffuel_a)=i25d_a.projectPower(power_a)
        a30.setInputMEDField("pow",ffluid_a)
        #a30.setInputMEDField("pow",ffuel_a)
        (ffluid_c,ffuel_c)=i25d_c.projectPower(power_c)
        ctf.setInputMEDField("power",ffuel_c)

        # Advance time step iteration counter
        tran_iter+=1
        #
        pass

##################################################


###################
# Additional printing
msg0="Coupled ATHLET-DYN3D-COBRATF " + typecase + " calculation
    of "+casename+" finished after "+str(time()-t_start)+" sec."
```

```
    msg1="The output files can be found in: "+ATHLET_out+" folder"
    msg2="                                  and "+CTF_out+" folder"
    msg2="                                  and "+DYN3D_work_dir+" folder
        "

    print bar
    print
    print msg0
    print msg1
    print msg2
    print bar

    output.close()
    ###################
    #Finalize the calculation
    ctf.terminate()
    d3d.Finalize()
    a30.a_Terminate()

    # Delete Interpolator files
    #files=["matrix.txt","matrix1D.txt","matrix2D.txt","src1D.txt","
        trg1D.txt"]
    #for file in files:
    #    if os.path.exists(file): remove(file)
    #
    print
    print "----------- Fin COBRATF ! -----------"
    print
    print "----------- Fin DYN3D ! -----------"
    print
    print "----------- Fin ATHLET ! -----------"

if __name__ == '__main__':
    main()
```

APPENDIX B

Header of the CTF C++ SALOME API

```
#ifndef _CTF_HXX_
#define _CTF_HXX_

#include <vector>
#include "MEDCouplingFieldDouble.hxx"
#include "MEDCouplingUMesh.hxx"
#include <math.h>

using namespace std;

#define FIRST_GEOMETRY 0
#define RECTANGULAR_GEOMETRY 0
#define HEXAGONAL_GEOMETRY 1
#define HEX_COOLCENT_FINEMESH_GEOMETRY 2
#define HEX_RODCENT_FINEMESH_GEOMETRY 3
#define LAST_GEOMETRY 3

class CTF {

public :
```

```
CTF();
~CTF();

//Set input path
void setInput(const char* name);
// Initializes the code variables and Reads input
bool initialize();
// Returns present time in CTF simulation
double presentTime() const;
// Returns the next proposed time step size in CTF
double computeTimeStep() const;
// Sets next time step size in CTF
bool initTimeStep(double dt);
// Performs next time step in CTF
bool solveTimeStep();
// Finalizes time step calculation in CTF
void validateTimeStep();
// Checks if steady-state is reached in CTF
bool isStationary() const;
// Resets variables to previous time step value
void abortTimeStep();
// Initializes transient by changing RTWP parameter to 1.0 and
   resetting forcing tables
void transientMode();
// Finalizes code run
void terminate();

// Mesh and Field Methods
// Rotates the CTF mesh (in degrees from 0 - 360)
void SetCoreMeshRotation (double rot);
// Translate the CTF mesh along X, Y and Z (in meters)
void SetCoreMeshTranslation (double xtrans, double ytrans, double
   ztrans);
// Generate the meshing, calls the privates methods
void genMeshCTF(int geom, int count_of_assemblies);
// Returns the 3D core fluid mesh
ParaMEDMEM::MEDCouplingUMesh* getFluidMeshCTF() const;
// Returns the 3D core rod mesh
```

```
  ParaMEDMEM::MEDCouplingUMesh* getRodMeshCTF() const;
  // Returns the 2D core inlet mesh
  ParaMEDMEM::MEDCouplingUMesh* getInletMeshCTF() const;
  // Returns the 2D core exit mesh
  ParaMEDMEM::MEDCouplingUMesh* getOutletMeshCTF() const;
  // Gives the list of accepted names for getOutputMEDField
  // moderator_density, moderator_temperature, fuel_temperature,
      porosity, weight, islocated
  std::vector<std::string> getOutputFieldsNames() const;
  // Returns a given MEDCoupling field in CTF
  ParaMEDMEM::MEDCouplingFieldDouble* getOutputMEDField(const std::
      string& name) const;
  // Sets the value of a given MEDCoupling field in CTF
  // Accepted names: power, inlet_massflow, inlet_enthalpy,
      outlet_pressure
  void setInputMEDField(const std::string& name, const ParaMEDMEM::
      MEDCouplingFieldDouble*);
  // Write a given MEDCoupling field in a VTK file (.vtu)
  void writeFieldinVTK(const char* name, const ParaMEDMEM::
      MEDCouplingFieldDouble* afield);
  // Gives the average value of a field object
  double AverageValue(const ParaMEDMEM::MEDCouplingFieldDouble*
      afield) const;

private:
  // Mesh and Field Methods
  // Find if a point is already defined in the mesh
  int _GetNearOrAddPointXYZ(vector<double>& coords,const double point
      [3],const int start_id=0);

  // Generate CTF Meshings
  // Generate fluid mesh for RECTANGULAR_GEOMETRY
  ParaMEDMEM::MEDCouplingUMesh *genFluidRectangular();
  // Generate fluid mesh for HEXAGONAL_GEOMETRY
  ParaMEDMEM::MEDCouplingUMesh *genFluidHexagonal();
  // Generate rod mesh for RECTANGULAR_GEOMETRY
  ParaMEDMEM::MEDCouplingUMesh *genRodRectangular();
  // Generate rod mesh for HEXAGONAL_GEOMETRY
```

```
  ParaMEDMEM::MEDCouplingUMesh *genRodHexagonal();
  // Generate fluid mesh for HEX_COOLCENT_FINEMESH_GEOMETRY
  ParaMEDMEM::MEDCouplingUMesh *genHexFineCoolCent();
  // Generate fluid mesh for HEX_RODCENT_FINEMESH_GEOMETRY
  ParaMEDMEM::MEDCouplingUMesh *genHexFineRodCent(int
      count_of_assemblies);
  // Generate rod mesh for HEX_RODCENT_FINEMESH_GEOMETRY
  ParaMEDMEM::MEDCouplingUMesh *genRodHexFineRodCent(int
      count_of_assemblies);
  // Generate rod mesh for HEX_COOLCENT_FINEMESH_GEOMETRY
  ParaMEDMEM::MEDCouplingUMesh *genRodHexFine();

  // CTF Meshing objects
  // CTF 3D Fluid Meshing
  ParaMEDMEM::MEDCouplingUMesh* _FluidMeshCTF;
  // CTF 3D Rod Structure Meshing
  ParaMEDMEM::MEDCouplingUMesh* _RodMeshCTF;
  // CTF Inlet Plane for boundary conditions
  ParaMEDMEM::MEDCouplingUMesh* _InletMeshCTF;
  // CTF Outlet Plane for boundary conditions
  ParaMEDMEM::MEDCouplingUMesh* _OutletMeshCTF;

  // Mesh translation [m] and rotation [grad]
  double _XTRANS;   // IN X direction
  double _YTRANS;   // In Y direction
  double _ZTRANS;   // In Z direction
  double _ROT;      // In grad

};
#endif
```