

# **Multi-Scale Thermal-hydraulic Developments for the Detailed Analysis of the Flow Conditions within the Reactor Pressure Vessel of Pressurized Water Reactors**

Zur Erlangung des akademischen Grades  
**Doktor der Ingenieurwissenschaften**  
von der KIT-Fakultät für Maschinenbau  
des Karlsruher Instituts für Technologie (KIT)

genehmigte

**Dissertation**

Von

**Kanglong Zhang**

geboren in Hebei, China

Tag der mündlichen Prüfung: 17.01.2020

Referent: Prof. Dr.-Ing. Robert Stieglitz

Karlsruhe Institut für Technologie

Korreferent: Prof. Dr.-Phys. César Queral

Universidad Politécnica de Madrid



# Acknowledgment

To my soul mate, lover, and wife Shanshan Kang: I owe it all to you. My elder son Zongshi Zhang and little daughter Muyin Kang: my lovely babies, you brought me omnipotent highs and complete my life.

My dear worthy supervisor Dr. Victor Hugo Sanchez-Espinoza: I will never forget your meticulous care and continuous support for my Ph.D. study and also for my life. Your guidance always indicates the direction of my research and I marvel at all the patience, motivation and immense knowledge you have shown in the routine duties. I appreciate all of your amiable and precious edifications during my Ph.D. career, many thanks!

I would like to express my sincere gratitude to my supervisor Prof. Dr. Robert Stieglitz for his professional and valuable guide to my Ph.D. work and the thesis composing. Without that, I can't imagine my slogging through research under the unsustainable burden. I also would like to thank Prof. César Queral for his insightful and significant comments which improve the thesis quality substantially.

My sincere thanks also go to Dr. Imke Uwe, Dr. Javier Jimenez, Dr. Jose Angel Gonzalez Vargas, Dr. Yuefeng Qiu, Dr. Guangming Zhou, Dr. Han Zhang, and Xilin Zhang for their great help and support when I was struggling with burning questions. I thank my fellow workmates Yousef Ibrahim Alzaben, Diego Ernesto Ferraro, Manuel Garcia, Abdullah Abalkhail, and all the other colleges in our group and institute. It was fantastic to work together with you guys.

With a special mention to my secretaries Ms. Birgit Zagolla, Ms. Petra Klug, and Ms. Ingeborg Schwartz. I am really grateful for their outstanding and truly works which dramatically simplify the administrative affairs so that I could focus on my research effectively.

Then, special gratitude goes to the China Scholarship Council (CSC). I am grateful to the consistent financial support for the last four years. The scholar solves my living concerns and is exactly the base of everything. Also, I thank all of my friends in Karlsruhe, in Germany, in China, for their company and encourage. Finally, I have to say thank you to my parents and younger brother in China. It is cozy to have you there.

Last, but by no means least, my cordial gratitude also goes to the lives of everyone.

Thanks for all your encouragement!



# Abstract

The multi-scale thermal-hydraulic analysis and the development of multi-scale thermal-hydraulic coupled codes have been becoming a promising area in the nuclear communities in recent years. They aim to enhance the capabilities of the thermal-hydraulic simulation tools and to better describe the thermal-hydraulic phenomena in the nuclear power systems.

The multi-scale thermal-hydraulic simulation of the Pressurized Water Reactor (PWR) is the emphasis of this thesis. A generic classification of the diverse multi-scale coupling approaches is put forward and the coupled codes and methods are compared. The thesis develops two multi-scale thermal-hydraulic coupling systems: 1) the coupled codes of the U.S. Nuclear Regulatory Commission (NRC) system code TRACE with the Karlsruhe Institute of Technology (KIT) in-house sub-channel code SubChanFlow (SCF) using the Exterior Communication Interface (ECI); 2) the coupled codes of the U.S. NRC system code TRACE with the French open-source CFD code TrioCFD using the Interface for Code Coupling (ICoCo) methodology.

The TRACE/SCF-ECI is developed as a server-less, parallel, and explicit coupling system, which adopts the domain-decomposition method and uses a newly-developed toolkit to handle the field mapping issues. This coupling pair is verified and validated against an academic coolant mixing problem and a VVER-1000 coolant mixing benchmark. It shows that the coupled code is able to predict the coolant mixing in the reactor pressure vessel in a more precise manner. Moreover, the coupled codes have been optimized in order to be more efficient.

The coupling of TRACE/TrioCFD using ICoCo is developed as a server-client, parallel explicit coupling system, which adopts the domain-overlapping method and uses the SALOME MEDCoupling library to handle the field mapping and data transfer between different meshes. A novel Dynamic-Implicit-Additional-Source (DIAS) method is implemented. It uses the coolant velocity, pressure, coolant temperature, and boron concentration fields translated by MEDCoupling from TrioCFD to correlate the four corresponding fields of TRACE. The correlation takes place in the entire overlapped domain. The coupled code is verified against the VVER-1000 coolant mixing benchmark and the improved codes' capability in predicting the coolant mixing in the reactor pressure vessel compared to previously used codes is demonstrated.



# Kurzfassung

Die mehrskalige thermohydraulische Analyse und die Entwicklung von mehrskaligen thermohydraulisch gekoppelten Programmen haben sich in den letzten Jahren zu einem vielversprechenden Gebiet im Bereich der Reaktortechnik entwickelt. Sie zielen darauf ab, die Fähigkeiten der thermohydraulischen Simulationswerkzeuge zu verbessern und die thermohydraulischen Phänomene in den Kernkraftsystemen umfassender zu beschreiben.

Die mehrskalige thermohydraulische Simulation eines Druckwasserreaktors (PWR) bildet den Schwerpunkt dieser Arbeit. Eine generische Klassifizierung der verschiedenen Multi-Skalen-Kopplungsansätze wird vorgeschlagen und die gekoppelten Programme und Methoden werden verglichen. Die Dissertation entwickelt zwei mehrskalige thermohydraulische Kopplungssysteme: 1) die Verbindung des Systemcodes TRACE der US Nuclear Regulatory Commission (NRC) mit dem Unterkanalprogramm SubChanFlow (SCF) des Karlsruher Instituts für Technologie (KIT) unter Verwendung einer externen Kommunikationsschnittstelle (ECI); 2) die Verbindung des US-amerikanischen NRC-Systemcodes TRACE mit dem französischen Open-Source-CFD-Code TrioCFD unter Verwendung der ICoCo-Methode (Interface for Code Coupling).

Die Kopplung TRACE / SCF-ECI wurde als serverloses, paralleles und explizites Kopplungssystem entwickelt, das die Methode der Domänenzerlegung anwendet. Ein neu entwickeltes Toolkit löst dabei die Feldzuordnungsprobleme. Dieses System wurde anhand eines akademischen Kühlmittelmischproblems für einen VVER-1000 als Referenz verifiziert und validiert. Es zeigt sich, dass der gekoppelte Code die Vermischung des Kühlmittels im Reaktordruckbehälter genauer vorhersagen kann. Darüber hinaus wurden die gekoppelten Codes optimiert, um effizienter zu arbeiten.

Die Kopplung von TRACE / TrioCFD mithilfe von ICoCo wurde als explizites Server-Client-Kopplungssystem entwickelt, das die domänenübergreifende Methode anwendet und die SALOME MED-Kopplungs-Bibliothek verwendet, um die Feldzuordnung und den Datentransfer zwischen verschiedenen Volumenzellen zu verwalten. Eine neuartige DIAS-Methode (Dynamic-Implicit-Additional-Source) wurde implementiert. Dabei werden die von der MED-Bibliothek aus TrioCFD übersetzten Felder für die Kühlmittelgeschwindigkeit, den Druck, die Kühlmitteltemperatur und die Borkonzentration verwendet, um die vier entsprechenden Felder in TRACE abzubilden. Die Übertragung findet in der gesamten überlappenden Domäne statt. Das Ergebnis wird mit dem VVER-1000-Referenzwert für die Kühlmittelvermischung verglichen, und die verbesserte Fähigkeit der Codes, das Mischen von Kühlmittel im Reaktordruckbehälter vorherzusagen, wird aufgezeigt.



# Contents

<b>Acknowledgment .....</b>	<b>1</b>
<b>Abstract .....</b>	<b>iii</b>
<b>Kurzfassung .....</b>	<b>v</b>
<b>Contents.....</b>	<b>1</b>
<b>1 Introduction of Pressurized Water Reactor Challenges .....</b>	<b>3</b>
1.1 The principles of PWR and its simulation.....	3
1.2 The multi-scale approach to enhance PWR thermal-hydraulic simulation .....	5
1.3 The challenges for multi-scale codes coupling.....	10
1.4 The objective of the thesis .....	10
1.5 Structure of the thesis .....	11
<b>2 Selected Codes for the Multi-Scale Thermal-Hydraulic Coupling of Reactor Simulation.....</b>	<b>13</b>
2.1 The system thermal-hydraulic code TRACE.....	13
2.1.1 Governing equation set and numerical algorithm of TRACE.....	14
2.1.2 A critical review of TRACE hydraulic models .....	16
2.2 The sub-channel thermal-hydraulic code SubChanFlow (SCF).....	18
2.3 The computational fluid dynamic code TrioCFD.....	19
<b>3 Strategies for Multi-Scale Coupling of TRACE/SCF and TRACE/TrioCFD .....</b>	<b>21</b>
3.1 Principle approaches.....	21
3.1.1 The Exterior Communication Interface (ECI) .....	21
3.1.2 The Interface for Code Coupling (ICoCo) .....	24
3.2 Selection of the coupling methodologies.....	25
3.2.1 Selection criteria for multi-scale coupling .....	25
3.2.2 Selected coupling methodologies for TRACE/SubChanFlow(SCF) and TRACE/TrioCFD.....	28
<b>4 The Multi-Scale Simulator TRACE/SCF based on ECI.....</b>	<b>31</b>
4.1 Development of ECI for SCF .....	31
4.2 Description of the coupled system TRACE/SCF-ECI .....	33
4.2.1 The domain decomposition coupling approach .....	33
4.2.2 The method for field mapping between two thermal-hydraulic domains .....	36
4.2.3 The explicit temporal coupling .....	37
4.3 Verification of TRACE/SCF-ECI with an academic coolant mixing and boron dilution case .....	40
4.3.1 Description of the problems .....	40
4.3.2 Performed simulations and discussion of results .....	41
4.4 Validation of TRACE/SCF-ECI using the VVER-1000 coolant mixing benchmark .....	43
4.4.1 Description of the VVER-1000 coolant mixing experiment.....	43

4.4.2	Description of the thermal-hydraulic models of the RPV and core .....	45
4.4.3	Discussion of selected results.....	45
4.5	The optimization and options for steady-state and transient simulation of TRACE/SCF-ECI .....	50
4.5.1	Shortcomings of the current temporal coupling.....	50
4.5.2	The optimized temporal coupling approaches .....	54
4.5.3	Test of the efficiency of the optimized temporal coupling approach.....	56
<b>5</b>	<b>The Multi-Scale Simulator TRACE/TrioCFD based on ICoCo.....</b>	<b>59</b>
5.1	Modification of TRACE to allow data transfer.....	59
5.1.1	Implement ICoCo for TRACE.....	59
5.1.2	Implementation of TRACE-ICoCo into SALOME .....	66
5.2	Description of the coupled system of TRACE/TrioCFD-ICoCo .....	67
5.2.1	The domain-overlapping method .....	68
5.2.2	The data transfer between TRACE and TrioCFD thermal-hydraulic domains.....	70
5.2.3	The explicit temporal coupling of TRACE/TrioCFD .....	73
5.3	The Dynamic-Implicit-Additional-Source (DIAS) method for domain-overlapping coupling .....	75
5.3.1	Principles of the DIAS approach.....	75
5.3.2	Verification of the DIAS Method with a flow splitter case .....	83
5.3.3	Verification of the DIAS method with an academic reactor problem.....	85
5.4	Validation of TRACE/TrioCFD-ICoCo using the VVER-1000 coolant mixing benchmark	95
5.4.1	Description of the thermal-hydraulic models of TRACE and TrioCFD .....	95
5.4.2	Discussion of the selected results.....	97
<b>6</b>	<b>Summary and Perspectives .....</b>	<b>103</b>
6.1	Summary .....	103
6.2	Perspectives .....	104
	<b>References .....</b>	<b>107</b>
	<b>Appendix A. Principles of the Mapping Toolkit .....</b>	<b>115</b>
	<b>Appendix B. The peculiarity of the hydraulic coupling in TRACE/TrioCFD .....</b>	<b>123</b>
	<b>List of Acronyms .....</b>	<b>127</b>
	<b>Nomenclature.....</b>	<b>131</b>

# 1 Introduction of Pressurized Water Reactor Challenges

After the first discovery of Uranium by Martin Klaproth in 1789, the demonstration of fission reaction by Otto Hahn and Fritz Strassman in 1938, the first fission nuclear reactor built by Enrico Fermi in 1942, now in 2019, around 449 Nuclear Power Plant (NPP) (Nuclear, 2019) are in operation mainly for electricity production (Wang, et al., 2018). Meanwhile, nuclear energy's potential use for hydrogen production (El-Emam & Özcan, 2019) and water desalination (Al-Othman, et al., 2019), etc. are also investigated. It is the most important contributor to low-carbon energy generation deployment (Ye, 2016).

The nuclear reactors are classified to different types according to their coolant: light water reactor (whose coolant is light water), heavy water reactor (whose coolant is heavy water), liquid metal reactor (whose coolant is liquid metal such as sodium and lead-bismuth eutectic), gas-cooled reactor (whose coolant is gas such as carbon dioxide and helium), molten salt reactor (whose coolant is melt salt-mixture), and others. The light water reactor is the most widely built and on service commercial nuclear reactors. It mainly includes the Pressurized Water Reactor (PWR) and the Boiling Water Reactor (BWR). Especially, PWR is the most popular reactor and dominates the nuclear power market.

## 1.1 The principles of PWR and its simulation

Figure 1-1 illustrates the principal functionalities of a PWR. It consists of three loops. The primary loop is pressurized (normally over 15 MPa) to suppress the boiling. The coolant is pumped into the reactor vessel by the primary pumps. It flows from bottom to the top of the reactor core and cools the fuel bundles by taking away the nuclear heat. The heated-up coolant then flows out of the vessel and enters the steam generator. Here it flows through thousands of small tubes (mostly in U shape) and passes the heat from the primary loop side to the second loop where the water evaporated to pressurized steam. The heat transfer is accomplished without mixing the two fluids to prevent the secondary coolant from becoming radioactive. The steam expands in the turbine which drives an electrical generator connected to the electric grid for transmission. After that, the secondary coolant is cooled down and condensed in a condenser which converts the steam to liquid so that it can be pumped back into the steam generator. The coolant in the tertiary loop supplies the heat sink for the condenser. Pressure in the primary circuit is maintained by a pressurizer.

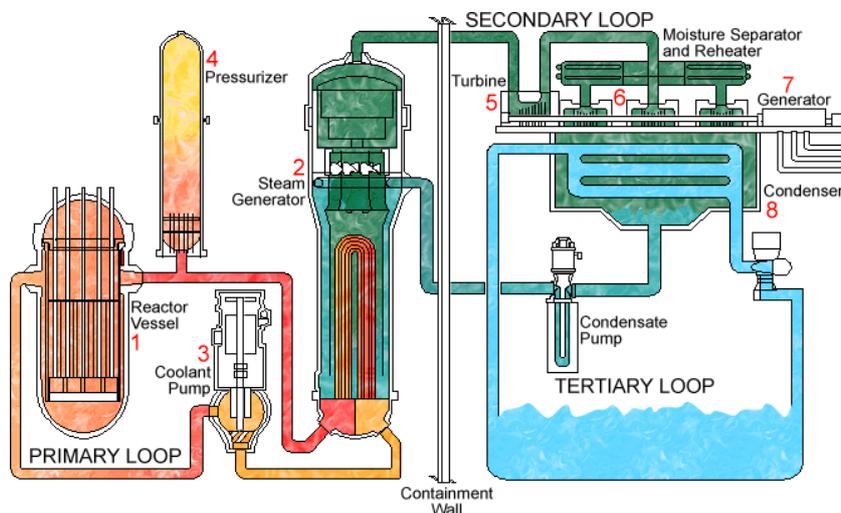


Figure 1-1 – The schematic diagram of the PWR demonstrating its basic principles

The PWR has been used for electricity generation worldwide for many years and lots of experience has been gained. The PWR allows a clean electricity generation with high safety standards. Nevertheless, continuous effort is still made by the nuclear community to improve its safety and efficiency. In order to achieve the goal, the physical phenomenon inside the reactor has to be carefully and sufficiently identified and investigated. Moreover, various other aspects have to be carefully lucubrated as well such as the economic risks and uncertainties (Kessides, 2010), the waste management and storage, and the spent fuel processing (Abu-Khader, 2009), etc. Especially after the Fukushima accident, a review of the safety status of the nuclear power plants was necessary to identify areas for improvements and the safety analysis methodologies together with the strategies for accident management are under continuous enhancement to adjust to the rapidly changing regulatory policies (Zeng, et al., 2016).

Experimental investigations are of paramount importance to understand key-safety phenomena taking place in nuclear power plants. However, experiments can hardly cover everything interested due to the expensive cost and measuring difficulties. Hence, numerical simulations complement the experimental research. Its role in accident management was already emphasized and many severe accident studies were carried out (Saghafi & Ghofrani, 2016). However, due to the limit of the computation power and the lack of mathematical models, different simulations have different emphasis according to the problem concerned. Some are interested in the dynamic of the whole system while some focus on physical phenomenon in a specified component or part of a component.

In a PWR, different thermal-hydraulic phenomena take place at diverse spatial scales whose characteristic lengths vary from meters down to nanometers. Hence, the simulations are typically classified into three main scales (Guelfi, et al., 2007) (D'Auria & Galassi, 2010):

- 1) System scale: the phenomenon which is tightly related to the response and dynamics of the overall nuclear power plant or facilities are concerned at this scale e.g. the large break loss of coolant accident in the PWR, where the primary cooling system of the reactor fails and loss the coolant (Petruzzi & D'Auria, 2007).
- 2) Component scale: the phenomenon occurring in special reactor components such as the reactor core and the heat exchanger is concerned at this scale e.g. the departure from nucleate boiling which is very important concerning about the safety of a PWR.
- 3) Average scale: the fine description of the reactor thermal-hydraulics is concerned at this scale e.g. the unsymmetrical coolant mixing in the downcomer and lower plenum (coolant of different temperatures entering the reactor vessel from different inlet nozzles will mix with each other in the vessel) of the Reactor Pressure Vessel (RPV) (Höhne, et al., 2009).

In the European NURESIM and NURISP projects, a similar classification was applied but the average scale was called mesoscale. In addition, another scale which focuses on more flow details named microscale was considered (Bestion, et al., 2012). Besides, in (Niceno, et al., 2010) the thermal-hydraulic simulation was expanded to nanoscale referring to e.g. molecular dynamic modeling techniques. In order to describe the thermal-hydraulic processes undergoing at the various spatial scales, many thermal-hydraulic simulation codes were developed and applied worldwide e.g. 1D or 3D system thermal-hydraulic codes, porous-media codes (CFD-porous media codes, sub-channel codes, and porous-media 3D codes), open-medium-CFD codes, large eddy simulation and detached eddy simulation codes, and direct numerical simulation codes (Bestion, 2010).

The system codes were designed to simulate almost all normal and accident scenarios of the whole nuclear power plant. It handles general tasks in the power plant e.g. the qualification of the safety margins (the threshold under which the system is assumed operating safely), the definition, and verification of emergency operation procedure, etc. (Bestion, 2018). The most wide-spread system thermal-hydraulic codes developed since many decades is RELAP (RELAP5-3D, 2019), TRACE (TRACE, 2016), ATHLET (Lerchl, et al., 2012), CATHARE (Emonot, et al., 2011), etc. The one-dimensional and three-dimensional coarse meshes favor a fast execution and robust numeric for the whole spectrum of design basis accidents. On the other hand, system codes are not intended to

describe phenomena taking place at the meso- or micro-scale e.g. turbulences due to the coarse mesh applied.

Thermal-hydraulic sub-channel codes dedicate the description of thermal-hydraulics in the core region by dividing the core flow paths to groups of channels. Codes like COBRA-TF (Salko & Avramova, 2015), COBRA-FLX (COBRA-FLX-Team, 2010), VIPRE (Sung, et al., 1999), FLICA (Toumi, et al., 2000), SubChanFlow (SCF) (Sanchez, et al., 2010), (Imke, et al., 2010) were developed for an improved description of the core thermal-hydraulics using quasi-3d models at component scale (decimeter- to centimeter-scale ) for the prediction of local safety parameters such as Critical Heat Flux (CHF) ratio, Critical Power Ratio (CPR), fuel centerline temperature, and fuel surface temperature. A very detailed review of sub-channel analysis methods and codes can be found in (Moorthi, et al., 2018) and (Cheng & Rao, 2015). In order to simulate the core thermal-hydraulics of full cores using sub-channel codes and to take profit of the increasing computer power, some sub-channel codes were parallelized e.g. SCF and CORBA-TF (Kucukboyaci & Sung, 2015). In addition, the codes using porous models to simulate the dedicated three-dimensional porous media two-phase flow with a Cartesian or unstructured grid are being developed to overcome the usually one-dimensional limitations of the system thermal-hydraulic codes. Instances are CUPID (Jeong, et al., 2010), PORFLOW (Runchal & Sagar, 1993), TWOPORFLOW (Chavez, et al., 2018), and PORFLO (Ilvonen, et al., 2010).

Furthermore, to improve the simulation of three-dimensional-effect within nuclear power plant circuits, CFD codes such as ANSYS-CFX, STAR-CD, OpenFOAM, TrioCFD are increasingly applied to nuclear system analysis. Main drivers for it are the increasing and cheap computation resources, the parallel capability of the codes, and the improved physical models and numerics (Calvin & Nowak, 2010). General guidelines of how to use CFD in reactor safety complementary to the system codes are given in (Mahaffy, et al., 2007). Recent investigations have shown the potentials of CFD-codes for the simulation of complete nuclear power plants or part of it (Böttcher, 2008) (Böttcher & Krüßmann, 2010) (Cheng, et al., 2016) (Cheng, et al., 2015) using meshing at different spatial scales ranging from millimeter to some centimeters. CFD-codes work well for single-phase problems that may be encounter in the primary and secondary circuits as well as in the containment of NPPs. Their application to two-phase flow is not yet comparable to the ones for single-phase flow due to some essential difficulties summarized in (Bestion, 2014). Nevertheless, new methods known as the Computational Multi-Fluid Dynamics (CMFD) which dedicates to simulate the two-phase flow on mesoscale or microscale are already one of the most promising trends for reactor applications (Yadigaroglu, 2005). Recent research devoted to enhance and validate the CFD-codes for two-phase flow problems of reactor safety has advanced (Mañes, et al., 2014) (Krepper, et al., 2008) (Antal, et al., 2000).

Considering the trends and the current capabilities of the different thermal-hydraulic codes for design and safety evaluations of PWR, the development of multi-scale thermal-hydraulic simulations has been initiated in different teams worldwide, in order to enhance the PWR thermal-hydraulic simulation.

## **1.2 The multi-scale approach to enhance PWR thermal-hydraulic simulation**

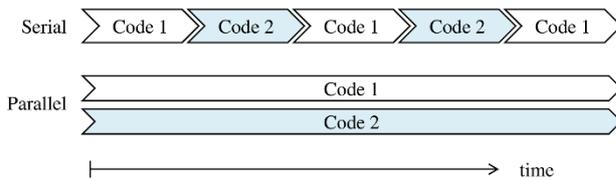
Generally speaking, phenomena occurring in system-scale, component-scale and average-scale are simulated by system code, sub-channel code, and CFD code, respectively. In the frame of different European projects e. g. NURESIM, NURISP and NURESAFE, multi-scale coupling approaches were systematically elaborated and various safety-relevant phenomena (Bestion, et al., 2012) were investigated in detail from a multi-scale perspective. Different publications (IAEA, 2007) (D'Auria, et al., 2004) (Bousbia-Salah & D'Auria, 2007) provide guidelines for potential coupling approaches. Moreover, multi-scale simulations could be extended to nano-scale or molecular-scale if molecular

dynamics (a computer simulation method for studying the physical movements of atoms and molecules) modeling techniques are considered (Niceno, et al., 2010).

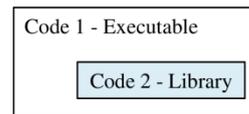
During the last two decades, various multi-scale thermal-hydraulic analysis has been carried out and different coupled codes have been developed by adopting diverse approaches. This section attempts to sort out the approaches and gives an overview of the multi-scale coupling schemes based on a literature review presented in Appendix A.

From the informatics point of view, the coupling of multi-scale thermal-hydraulic codes can be either parallel or serial, as illustrated in Figure 1-2. In this context, parallel mode especially refers to the inter-process parallelization e.g. Message Passing Interface (MPI) -based models. On the contrary, the architecture-based classification differentiates between the internal and external coupling. In an internal coupled system, one code is compiled as a static or dynamic library fully integrated inside the other code resulting in one single executable, see Figure 1-3. While, the external coupling option offers three possibilities, see Figure 1-4:

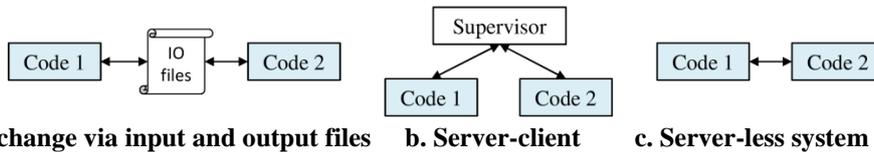
- 1) Data exchange via Input and Output (IO) files.
- 2) Use a supervisor to coordinate the coupling procedures.
- 3) The two codes directly communicate with each other forming a sever-less system.



**Figure 1-2 – Operational logics of the serial and parallel coupling.**



**Figure 1-3 – Arrangement of codes for internal code coupling.**

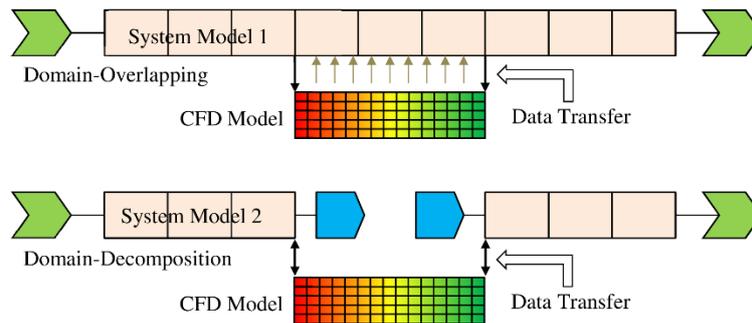


**a. Data exchange via input and output files    b. Server-client    c. Server-less system**

**Figure 1-4 – Schematics of data flows for external code coupling.**

The internal coupling is the most applied methods due to their relatively simple implementation. The server-client and server-less external coupling can achieve high performance and pose better expandability and maintainability compared to the internal coupling. However, intensive programming effort is necessary as well as in-deep knowledge of the coding structure of the involved solvers is required. Moreover, the condition could become more complex when the parallel mode is expected.

In a multi-scale coupling system, the system code, the sub-channel code, and the CFD code are coupled to achieve a more realistic simulation of a nuclear power plant using a code-specific spatial resolution of the computational domain. For instance, the coarse mesh of system code for the whole nuclear power plant, the relatively refined mesh of sub-channel code for the core region, and the fine mesh for several specific parts e.g. the downcomer. So, one of the critical tasks for the multi-scale coupling is how to specify which code is in charge of which computational domains within the entire problem. In this context, the way of coupling the different computational domains of the involved thermal-hydraulic solvers (system, sub-channel, and CFD) is of great importance. Therefore, two methods are known: domain decomposition and domain overlapping. Figure 1-5 demonstrates the two options with the simulation of a pipe using a system code and a CFD code.



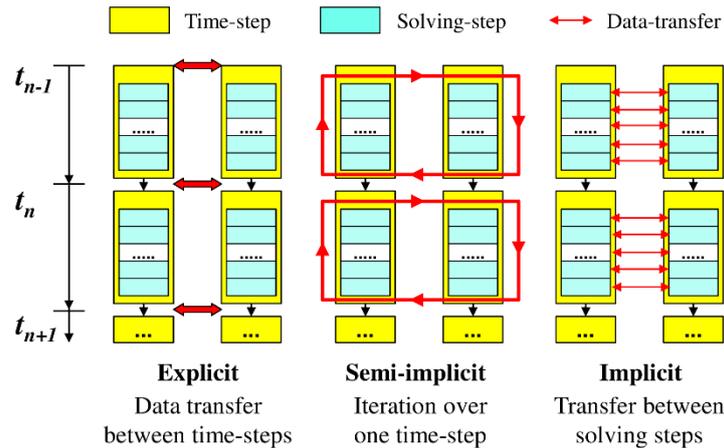
**Figure 1-5 – Demonstration of the domain-overlapping (top) and domain-decomposition (bottom) coupling approaches.**

For the domain-overlapping case shown in Figure 1-5, the system code intends to model the entire pipe while the CFD code only simulates part of the pipe. The lower part of the figure illustrates the domain-decomposition method where the system code describes the pipe without the three central nodes, which are simulated by the CFD-code. The inter-code data transfer occurs at the two boundaries between the CFD model and system model represented by the two double-sided arrows. The upper part of Figure 1-5 illustrates the domain-overlapping method. There, the system code simulates the whole pipe and the CFD-code simulates the three central cells only. These three cells represent the overlapping domain. The fields at the face where the overlapping domain starts and ends are extracted and transferred to the CFD-code as its updated boundary conditions. This data transfer is represented by the two one-way arrows. On the contrary, the data from CFD to the system code are body fields derived from the whole CFD computation domain. Those body data are used to correlate the system solution. A bunch of grey one-way arrows represents them.

In general, the meshes of system code, sub-channel code, and CFD code have a different spatial resolution, see Figure 1-5. There the CFD mesh is much finer than the one of the system code. Consequently, the fields have to be translated from one mesh to the other one in a proper way before the data transfer. For it, three options are available:

- 1) Manual definition of mesh spatial matching relations. It is a direct method free of programming effort but calls for massive work at any time a new coupled code is developed.
- 2) User-developed toolkit. In this method, the mesh matching is done in an automatic way and it is relatively efficient. It can focus on specific mapping problems for a special coupling scheme but it is not universal and powerful.
- 3) Use of a third-party mesh-processing toolkit e.g. the MEDCoupling library (MED, 2019) of the SALOME platform. This toolkit is general and powerful and could properly handle different mesh-mapping scenarios in an automatic manner. However, those toolkits mostly require mesh definitions while most of the system and sub-channel codes do not have meshes.

The mesh interpolation and physical field mapping methods together with the domain coupling are known as the spatial coupling since they deal with the geometric objects. Beyond the spatial coupling, the other element for multi-scale coupling in terms of mathematics and physics is the temporal coupling approach that is in charge of the synchronization of the data transfer among the involved solvers. The spatial coupling solves the problem of how to transfer the data properly, while the temporal coupling solves the problem of how to use these data in order to establish a synchronous multi-scale coupling code. The following three different temporal coupling approaches are implemented in different multi-scale codes: 1) explicit coupling, 2) semi-implicit coupling, 3) implicit coupling. Figure 1-6 illustrates the basic principles and the differences among these approaches.



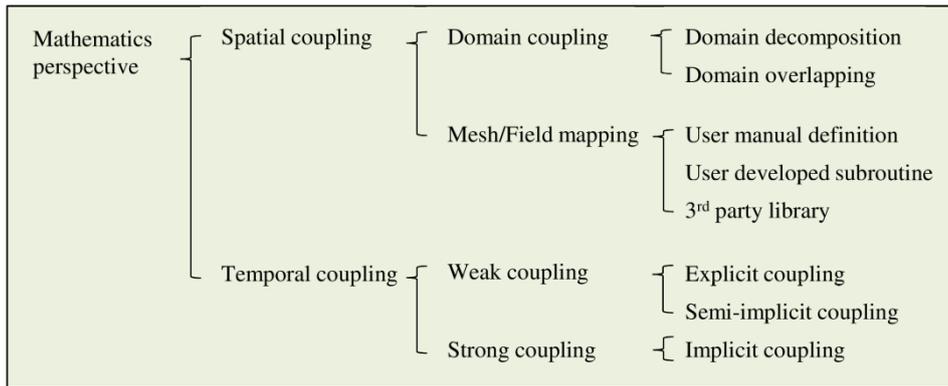
**Figure 1-6 – Working principles and differences of the explicit, semi-implicit, and implicit temporal coupling.**

The explicit coupling is the most straightforward temporal coupling method where the data transfer is carried out only once at the beginning or end of each time step. No check of the consistency of the codes' fields within a time step is performed. Hence, the codes' results may be inconsistent with each other, which might result in a consequently unsteady situation during the solution. Additionally, the solution accuracy is usually in first order thus slows down the overall system converging speed.

In order to ensure the data consistency within one time step of the coupled codes, the semi-implicit coupling must be implemented, where the data transfer is done many times until all the codes' result convergence within one time step. Hence, it is more robust than the explicit coupling; while on the other hand, it is more complicated to be implemented. In a semi-implicit coupling approach, the data is consistent for a single time step. However, the performance of the coupled codes can be further improved by spreading the data transfer down to the solution procedures of the set of equations, which is the implicit coupling. This is the most efficient, robust coupling approach but needs significant modification of the source code, and hence it is challenging.

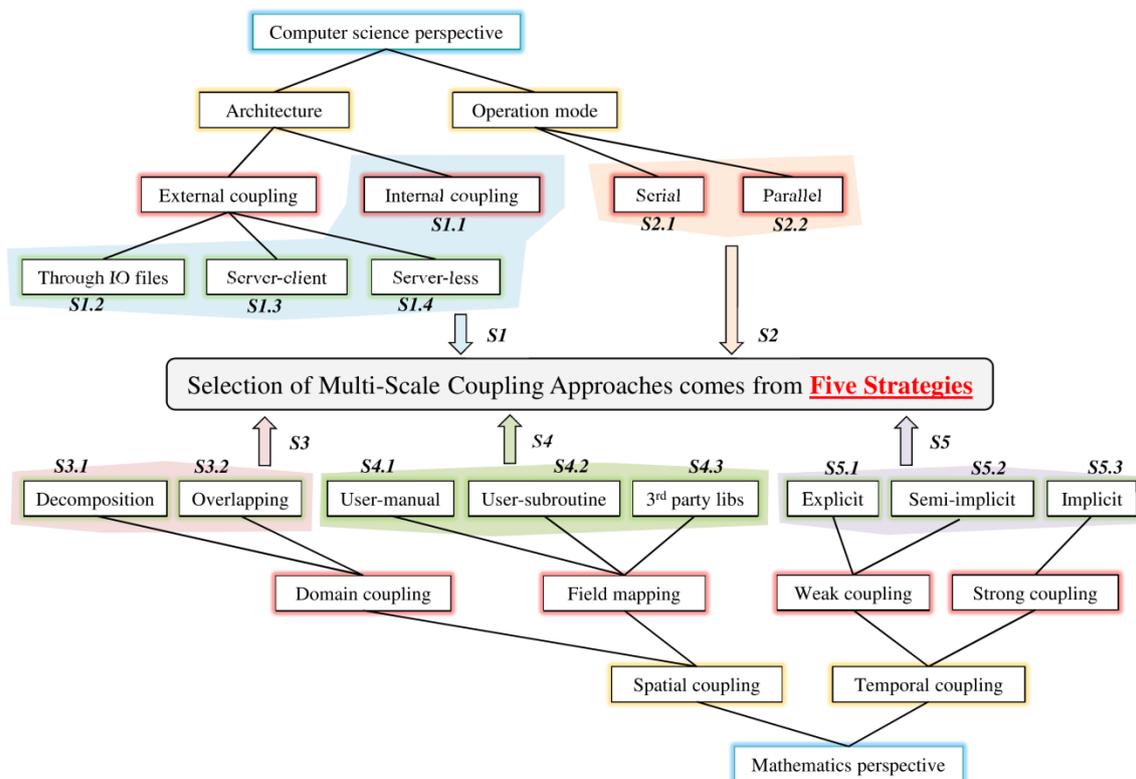
It is worth to note that Figure 1-6 only presents the general form of the three temporal coupling methods. The typical explicit coupling method is the Operator Splitting (OS) method, which is demonstrated in (Chorin, 1968) and now widely used in various coupling works. It is sometimes also called Operator Splitting Semi Implicit (OSSSI) method though it is actually an explicit method (Zhang, et al., 2018). There are also diverse varieties for the semi-implicit e.g. Picard method (Hamilton, et al., 2016), (Zhang, et al., 2018) and implicit e.g. the Jacobian-Free Newton-Krylov (Knoll & Keyes, 2004) coupling methods. Moreover, some other acceleration techniques are also passively developed and implemented to the codes temporal coupling e.g. staggered-time method (Aragonés, et al., 2004), the Anderson-acceleration method (Walker & Ni, 2011), the Approximate Block Newton method (Yeckel, et al., 2009), the NLeM method (Zhang, et al., 2018). They are also developed aimed at the optimization of the inter-codes synchronization. In (Zerkak, et al., 2015), a review of the temporal coupling methods is done in detail. It is worth to note that those methods are not limited to multi-scale coupling but they are universal methods for various coupling issues e.g. multi-physics coupling and for non-nuclear applications as well.

In view of the development of multi-scale coupling methods for thermal-hydraulic codes to solve reactor safety problems, it is important to look at the solution procedure applied to solve the system of equations in the thermal-hydraulic codes. If the set of equations are solved separately in the thermal-hydraulic codes, it is a weakly coupled code. Otherwise, it is a strong coupling if the equation sets are solved simultaneously. Figure 1-7 presents a classification of the multi-scale coupling approaches from the mathematical perspective.



**Figure 1-7 – Classification cladogram of the multi-Scale coupling approaches from the mathematical perspective.**

Figure 1-8 summarizes the all possible coupling approaches for multi-scale coupling of thermal-hydraulic codes (a summary from Figure 1-2 to Figure 1-7). From the figure, it is apparent that there are four possibilities - S1.1, S1.2, S1.3, and S1.4 for Strategy 1 (S1 - coupling architecture), two options – S2.1 and S2.2 for Strategy 2 (S2 – operation mode), two choices – S3.1 and S3.2 for Strategy 3 (S3 – domain coupling), three methods – S4.1, S4.2, and S4.3 for Strategy 4 (S4 – physical field mapping), and three patterns – S5.1, S5.2, and S5.3 for Strategy 5 (S5 – temporal coupling). Each of the five strategies has to be determined for a coupling strategy. In other words, a multi-scale coupling approach is composed of the five pieces below.



**Figure 1-8 – Overall classification of the multi-scale coupling approaches and there are five strategies to identify a multi-scale coupling system.**

## 1.3 The challenges for multi-scale codes coupling

The former section makes a review of the current multi-scale coupling approaches and puts forward a universal standard to classify them. According to the descriptions, there are mainly five challenges to implement a multi-scale thermal-hydraulic coupling system:

- 1) Selecting coupling architecture (S1 in Figure 1-8). Developers have to balance the complexity and efficiency according to their demands. From S1.1 to S1.4, the coupling architecture is getting more complex but more efficient and flexible.
- 2) Selecting operating mode (S2 in Figure 1-8). Normally parallel is better than serial because it is more efficient and has good expandability. But the drawback is complexity. Developers should think over the options because they differ significantly.
- 3) Managing the geometry adaption (S3 in Figure 1-8). Developers must specify the relations between the different codes' thermal-hydraulic domains. Different strategies mean distinct developing work. Nevertheless, for both of them, careful attention and heavy work are necessary to properly define the inter-domain communications.
- 4) Handling the field mapping (S4 in Figure 1-8). The data and field translation between different codes' meshes is always a crucial problem for codes coupling. Three options are available. However, all of them need significant work.
- 5) Assuring the data synchronization (S5 in Figure 1-8). This is the base of a correct coupled simulation. The challenge is to solve the problem of how to transfer the right data in the right place at the right time. This is also known as the temporal coupling between coupled codes.

If two or more codes are to be coupled, the above five challenges have to be solved and proper approaches have to be carefully selected.

## 1.4 The objective of the thesis

At the Karlsruhe Institute of Technology (KIT), the multi-scale investigations devoted to couple system codes with sub-channel and CFD-codes has been started. This thesis has three general objectives which consists of several sub-objectives. They are:

- 1) Classify and evaluate the various multi-scale coupling approaches.
  - a. Propose a general standard to classify the approaches, which could be referenced as a guideline to organize a new coupling work;
  - b. Perform a quantized evaluation of the classified approaches, which helps to select appropriate approaches for the coupled codes.
- 2) Implement the coupling system of the system code TRACE (TRACE, 2016) and the sub-channel code SubChanFlow (SCF) (Sanchez, et al., 2010) through a newly developed SOCKET-based Exterior Communication Interface (ECI) (Mahaffy & C., 2003).
  - a. Develop the ECI component for SCF;
  - b. Implement TRACE/SCF-ECI as a server-less, parallel system. Adopting the domain-decomposition and explicit method.
  - c. Develop a new toolkit to handle the inter-domain field translation;
  - d. Verify and validate the coupled code against academic and benchmark cases.
- 3) Implement the coupling system of TRACE and the open-source CFD code TrioCFD (Angeli, et al., 2015) through an Interface for Code Coupling (ICoCo) (Deville & Perdu, 2012).
  - a. Develop the ICoCo module for TRACE: split TRACE to functional components, equip TRACE with mesh-generating functions;
  - b. Implement TRACE/TrioCFD-ICoCo as a server-client, parallel system. Adopting the domain-overlapping and explicit method. Use the 3<sup>rd</sup> party library - MEDCoupling to handle the inter-domain field translation;

- c. Develop practical methods to realize the domain-overlapping coupling;
- d. Verify and validate the coupled code against academic and benchmark cases.

In summary, this thesis should develop two multi-scale thermal-hydraulic coupled codes – TRACE/SCF-ECI and TRACE/TrioCFD-ICoCo. Their functions should be verified and validated.

## 1.5 Structure of the thesis

The thesis consists of six chapters.

The 1<sup>st</sup> chapter. It is the current chapter where the background of PWR along with its simulation is firstly discussed. Then, the multi-scale thermal-hydraulic method for PWR simulation is proposed and a classification of the multi-scale thermal-hydraulic approaches is put forward. Additionally, the challenges to implement a multi-scale coupling system are clarified. At the end of this chapter, the thesis's objectives and its structure is introduced.

The 2<sup>nd</sup> chapter. The three to-be-coupled codes – TRACE, SCF, and TrioCFD are described in this chapter. Especially, the details of TRACE governing equations and its models are carefully depicted and investigated.

The 3<sup>rd</sup> chapter. This chapter denotes to select proper coupling approaches for the two to-be-implemented pair – TRACE/SCF-ECI and TRACE/TrioCFD-ICoCo. First, details of ECI and ICoCo are explained. Then the coupling approaches proposed in chapter 1 are evaluated. Based on the particularities of the two interfaces and the evaluations of the coupling approaches, appropriate methods are selected for the two coupled codes.

The 4<sup>th</sup> chapter. This chapter explains the coupling system of TRACE/SCF-ECI in detail. It begins with the development of ECI for SCF. Based on that, each piece of the coupling work is explained including the domain-decomposition method, the field mapping, and the explicit temporal coupling. The verification and validation are also presented, where an academic case and a VVER-1000 coolant mixing benchmark are used. Moreover, this chapter also proposes an optimized scheme for the coupled codes in order to enhance its efficiency.

The 5<sup>th</sup> chapter. This chapter explains the coupling system of TRACE/TrioCFD-ICoCo in detail. It begins with the development of ICoCo for TRACE. Based on that, each piece of the coupling work is explained including the domain-overlapping method, the field mapping, and the explicit temporal coupling. The validation are also presented, where a VVER-1000 coolant mixing benchmark are used. Moreover, this chapter introduces a newly-developed Dynamic-Implicit-Additional-Coupling (DIAS) method for the domain-overlapping approach.

The 6<sup>th</sup> chapter. It makes a brief summary of the thesis and poses several perspectives in order to optimize and improve this doctoral work.



## 2 Selected Codes for the Multi-Scale Thermal-Hydraulic Coupling of Reactor Simulation

In the frame of this dissertation, the coupling of a system thermal-hydraulic code with a sub-channel and a CFD code is implemented. For this purpose, the reference thermal-hydraulic code of the U.S. Nuclear Regulatory Commission (NRC) for Light Water Reactor (LWR) transient analysis named TRACE, the Karlsruhe Institute of Technology (KIT) sub-channel code SubChanFlow (SCF) and the French open-source TrioCFD code developed at CEA are selected. The system code TRACE within the coupling system could catch the behavior and dynamic of the whole plant (phenomenon occurring at macro-scale) taking account into the impact from the core (from SCF) or other parts (from TrioCFD) with fine resolution. On the other way round, the simulation of the core or other parts by SCF or TrioCFD could be improved by receiving the dynamic boundary conditions from TRACE which reflecting the behavior of the whole plant. This is a bidirectional promoting system.

### 2.1 The system thermal-hydraulic code TRACE

TRACE (TRAC/RELAP Advanced Computational Engine) is a best-estimate reactor system analysis code developed by the U.S. NRC for the analysis of design basis accident of light water reactors.

The TRACE code is able to represent a complete nuclear power plant or any experimental facility on a reactor component-based approach. It describes the key phenomena inside pipes, pumps, valves, reactor pressure vessel, core, pressurizer, separators, etc. Each physical piece of equipment in a flow loop can be represented as some type of component, and each component can be further divided into some number of physical volumes (also called cells) over which the governing equations are averaged. The VESSEL component is the only one with the capability to describe the flow and heat transfer in a 3D coarse mesh. It is used mainly to represent the Reactor Pressure Vessel (RPV) and it may be also used to represent a steam generator or a pressurizer. The VESSEL component in TRACE can use 3D  $(x, y, z)$  Cartesian (Figure 2-1c) and/or 3D  $(r, \theta, z)$  cylindrical (Figure 2-1b) for example to represent the flow in the core (TRACE, 2016). The Cartesian and Cylindrical coordinates are also given in Figure 2-1a.

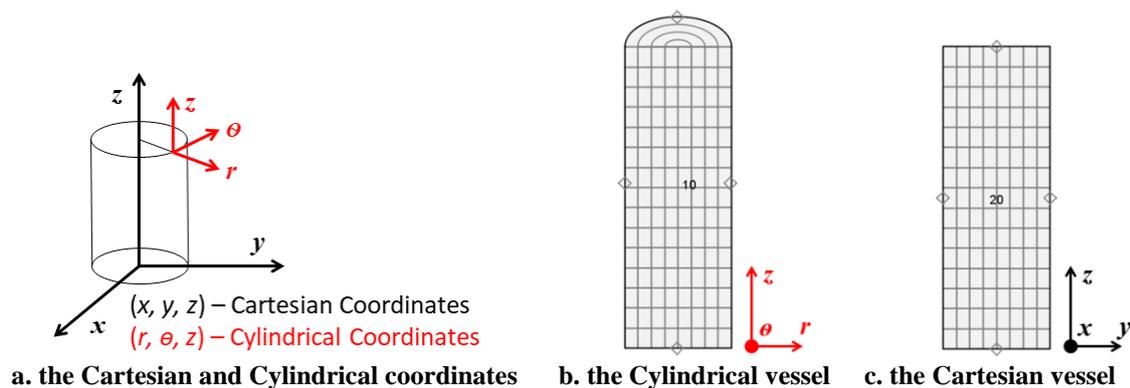


Figure 2-1 – The Cylindrical and Cartesian VESSEL component in TRACE and the two coordinates.

TRACE can solve both steady-state and transient problems. The steady-state solution behaves quite similar to the transient one except that it has a special criterion check after every predetermined time step and applies a slight modification of time term of the transient equations. Since TRACE plays a key role in this dissertation and its numerics has tight causal connections with the coupling

methodologies, it is necessary to carefully inspect TRACE solution procedures and some related issues. This is the base of everything discussed in this dissertation.

## 2.1.1 Governing equation set and numerical algorithm of TRACE

TRACE solves a system of six conservation equations in the two-fluid formulation for 1D and/or 3D single and/or two-phase flow in a nuclear power plant. For the solution of the equation set, two methods are available in TRACE: the semi-implicit method and the Stability-Enhancing Two-Step (SETS) method. Compared to the explicit method, the semi-implicit method eliminates the sound speed from the Courant stability by implicit calculation of the pressure in momentum equations, leaving what is commonly referred to as the “material Courant” limit. The material Courant limit is eliminated to some extent by the SETS method through pre-solving a so-called “stabilizer momentum equation” before the semi-implicit step and post-solving two so-called “stabilizer continuous and energy equations”. The solution of the “stabilizer continuous equation” is an attempt to solve the density equation implicitly. However, it is actually not a complete implicit way, which means even though the material Courant limit may be loosed sufficiently; it is still not an unconditionally stable method anyway.

SETS is the recommended numerical method in TRACE. Specifically followed the previous paragraph, it solves a stabilizer motion equation – Equation 2-1, a semi-implicit equation set (a typical thermal-hydraulic governing equation set including motion - Equation 2-2, continuity - Equation 2-3 and energy - Equation 2-4 equations) and a stabilizer equation set (including continuity - Equation 2-5 and energy - Equation 2-6 equations) in sequence. Here in the equation set,  $V$  is the coolant velocity,  $\rho$  is the coolant density,  $P$  is the pressure,  $K$  is the friction coefficient,  $e$  is the internal energy,  $q$  is the heat source or sink,  $\Delta t$  is the time step,  $x$  stands for the mesh size. The superscript “ $n$ ,  $n+1$ ” indicate the problem time level. The subscript “ $j$ ,  $j+1/2$ ,  $j-1/2$ ” stand for the cell numbering along the flow direction. The variable wrapped with “ $\langle \rangle$ ” denotes the volume averaged value in the cell center. The “ $\sim$ ” assign a process variable. Those equations are all in 1D and single-phase form for the sake of clear and simple explanation.

### 1) Stabilizer Motion Equation

$$\frac{\tilde{V}_{j+\frac{1}{2}}^{n+1} - V_{j+\frac{1}{2}}^n}{\Delta t} + V_{j+\frac{1}{2}}^n \cdot \frac{\partial \tilde{V}^{n+1}}{\partial x} \Big|_{j+\frac{1}{2}} + \frac{1}{\langle \rho \rangle_{j+\frac{1}{2}}^n} \cdot \frac{P_{j+1}^n - P_j^n}{\Delta x} + K_{j+\frac{1}{2}}^n \cdot \left( 2\tilde{V}_{j+\frac{1}{2}}^{n+1} - V_{j+\frac{1}{2}}^n \right) \cdot \left| V_{j+\frac{1}{2}}^n \right| = 0. \quad \text{Equation 2-1}$$

### 2) Semi-Implicit Equation Set

$$\frac{V_{j+\frac{1}{2}}^{n+1} - V_{j+\frac{1}{2}}^n}{\Delta t} + V_{j+\frac{1}{2}}^n \cdot \frac{\partial \tilde{V}^{n+1}}{\partial x} \Big|_{j+\frac{1}{2}} + \frac{1}{\langle \rho \rangle_{j+\frac{1}{2}}^n} \cdot \frac{\tilde{P}_{j+1}^{n+1} - \tilde{P}_j^{n+1}}{\Delta x} + K_{j+\frac{1}{2}}^n \cdot \left( 2V_{j+\frac{1}{2}}^{n+1} - V_{j+\frac{1}{2}}^n \right) \cdot \left| V_{j+\frac{1}{2}}^n \right| = 0. \quad \text{Equation 2-2}$$

$$\frac{\tilde{\rho}_j^{n+1} - \rho_j^n}{\Delta t} + \frac{\partial}{\partial x_j} \rho^n V^{n+1} = 0. \quad \text{Equation 2-3}$$

$$\frac{\tilde{\rho}_j^{n+1} e_j^{n+1} - (\rho e)_j^n}{\Delta t} + \frac{\partial}{\partial x_j} \rho^n e^n V^{n+1} + \tilde{P}_j^{n+1} \cdot \frac{V_{j+\frac{1}{2}}^{n+1} - V_{j-\frac{1}{2}}^{n+1}}{\Delta x} = \tilde{q}_{heat}^{n+1}. \quad \text{Equation 2-4}$$

### 3) Stabilizer Continuity and Energy Equations

$$\frac{\rho_j^{n+1} - \rho_j^n}{\Delta t} + \frac{\partial}{\partial x_j} (\rho^{n+1} V^{n+1}) = 0. \quad \text{Equation 2-5}$$

$$\frac{(\rho e)_j^{n+1} - (\rho e)_j^n}{\Delta t} + \frac{\partial}{\partial x_j} [(\rho e)^{n+1} V^{n+1}] + \tilde{P}_j^{n+1} \cdot \frac{V_{j+\frac{1}{2}}^{n+1} - V_{j-\frac{1}{2}}^{n+1}}{\Delta x} = \tilde{q}_{heat}^{n+1}. \quad \text{Equation 2-6}$$

The equation set above can be solved in a quasi-full-implicit way since the Courant stability limit is completely eliminated by the implicitness of the pressure in the semi-implicit motion equation, and the material courant stability limit is sufficiently relaxed by the implicitness of the density in the stabilizer continuity equation. The information on sound waves and continuity waves now can be propagated to the whole computation domain in a quite fast manner with large time steps. This semi-implicit numerical method is more like a mathematical trick to accelerate the simulation and does not physically reflect the practical procedures as that by the explicit methods.

The stabilizer motion equation is a linear partial differential equation with only one unknown -  $\tilde{V}^{n+1}$  is firstly solved to get an initial guess of the coolant velocity for the current new time step. With the stabilizer velocity -  $\tilde{V}^{n+1}$  solved from the stabilizer motion equation, the semi-implicit motion equation can be transformed and rewritten to Equation 2-7, where A is all the known remaining physical quantities from Equation 2-2. Now the coupled relationship between the pressure and velocity fields is more obvious.

$$V_{j+\frac{1}{2}}^{n+1} = \left. \frac{dV}{dP} \right|_{j+\frac{1}{2}} \cdot (\tilde{P}_{j+1}^{n+1} - \tilde{P}_j^{n+1}) + A. \quad \text{Equation 2-7}$$

The pressure  $\tilde{P}^{n+1}$  and velocity  $V^{n+1}$  in Equation 2-7 is all new-time value which can't be solved directly. Anyway, it is known that the velocity is a function of the pressure. Moreover, since the density  $\rho$  and energy  $e$  both depend on the pressure  $P$  and temperature  $T$ , given by the equation of state, the flowing two state equations could be used to close the TRACE governing equation set.

$$\tilde{\rho}^{n+1} = f(\tilde{P}^{n+1}, \tilde{T}^{n+1}), \quad \text{Equation 2-8}$$

$$\tilde{e}^{n+1} = f(\tilde{P}^{n+1}, \tilde{T}^{n+1}). \quad \text{Equation 2-9}$$

Substitute Equation 2-7, Equation 2-8 and Equation 2-9 into Equation 2-3 and Equation 2-4, the transformed new continuity and energy equations now only contain two unknowns – the new time pressure  $\tilde{P}^{n+1}$  and temperature  $\tilde{T}^{n+1}$ . Due to the  $\rho V$ ,  $\rho e$  and  $\rho e V$  lied in Equation 2-3 and Equation 2-4, the new continuity and energy equations are both nonlinear partial differential equations, which are to be solved with linearization and iteration techniques. Here a standard Newton iteration is applied, in which the difference of variables at two linearization iteration steps ( $i^{th}$  and  $i^{th}+1$ ) are represented with tiny changes, Equation 2-10 and Equation 2-11. Substitute the two equations to the new continuity and energy equations derived from Equation 2-3 and Equation 2-4 which only contains pressure  $\tilde{P}^{n+1}$  and temperature  $\tilde{T}^{n+1}$ , leads to a solution matrix shown in Equation 2-12. The “ $\delta$ ” means this is a tiny perturbation between two iteration steps. The “ $a, b, c$ ” denote the known coefficients in the equations

$$\bar{p}_j^{n+1,i+1} = \bar{p}_j^{n+1,i} + \delta P_j^i, \quad \text{Equation 2-10}$$

$$\tilde{T}_j^{n+1,i+1} = \tilde{T}_j^{n+1,i} + \delta T_j^i, \quad \text{Equation 2-11}$$

$$\begin{bmatrix} \delta P_j^i \\ \delta T_j^i \end{bmatrix} = \begin{bmatrix} a_1^i \\ a_2^i \end{bmatrix} - \begin{bmatrix} b_1^i \\ b_2^i \end{bmatrix} (\delta P_j^i - \delta P_{j-1}^i) + \begin{bmatrix} c_1^i \\ c_2^i \end{bmatrix} (\delta P_{j+1}^i - \delta P_j^i). \quad \text{Equation 2-12}$$

The first line in Equation 2-12 corresponds for the semi-implicit continuity equation while the second line is for the semi-implicit energy equation. For the continuity equation (first line) of Equation 2-12, the only unknown is the pressure difference which is linear and could be conveniently worked out. With the new  $\delta P^i$  substituted to the energy equation (second line of Equation 2-12),  $\delta T^i$  is obtained. Then  $\bar{p}^{n+1,i+1}$  and  $\tilde{T}^{n+1,i+1}$  can be solved by substituting the two  $\delta P^i$  and  $\delta T^i$  to Equation 2-10 and Equation 2-11. Consequently, the velocity on the current iteration step can be derived in sequence. If variations in pressure and temperature fall below the pre-defined criteria, the iteration is terminated and the values of pressure, temperature, and velocity for the current semi-implicit time step are determined. Afterwards, the stabilizer continuity and energy equations are solved to obtain the final density and energy for the current time step.

As supplementary to the six elementary equation set, two additional equations are solved to describe the transport of non-condensable gas and the boron concentration in the liquid phase. Hence, a total of eight equations are solved in TRACE. Additionally, a large number of models are needed to mathematically close the system of equations. These relations are derived mainly from experiments for the unknown terms of the mass, momentum and energy equations. They describe among others the wall and phase-interface friction, the evaporation (from water to steam) and condensation (from steam to water) rate (interphase mass and heat transfer) as well as the wall to liquid heat transfer for both vertical and horizontal flow regimes covering the whole boiling curve. One major defect of the models is their empirical or semi-empirical inherent attribute. This is exactly why the fine solutions from SCF or TrioCFD are desired instead of that by TRACE in some specific components or parts. Exploration of this defect is necessary because it is the cornerstone of the original motivation of the code couplings.

## 2.1.2 A critical review of TRACE hydraulic models

Here only the 1D and single-phase flow situations are presented for a clear and simple explanation. A flow-abrupt-expansion pipe is selected to illustrate the TRACE calculation of the convective and friction pressure drops, as shown in Equation 2-15.

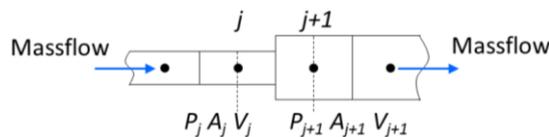


Figure 2-2 – Sketch of treatment of a discontinuous flow area expansion.

Construct the steady-state incompressible motion conservation equation for the control volume between edge  $j$  and  $j+1$ , as shown in Equation 2-13, where  $P$  is the static pressure,  $A$  is the edge cross-sectional area,  $V$  is the velocity and  $\Delta t$  the time interval. By eliminating the time interval and performing simple operations, Equation 2-13 is transformed to Equation 2-14. Actually, the  $\Delta P$  in Equation 2-14 is exactly the convective pressure drop for the abrupt expansion geometry. This is a first-order upwind numerical method inherited from TRACE solution.

$$\begin{aligned} & [P_j \cdot A_j \cdot \Delta t + \rho \cdot V_j \cdot (V_j \cdot A_j \cdot \Delta t)] - [P_{j+1} \cdot A_{j+1} \cdot \Delta t + \rho \cdot V_{j+1} \cdot (V_{j+1} \cdot A_{j+1} \cdot \Delta t)] \\ & + (A_{j+1} - A_j) \cdot P_j \cdot \Delta t = 0. \end{aligned} \quad \text{Equation 2-13}$$

$$\Delta P \gg P_j - P_{j+1} = -\rho \cdot V_{j+1} \cdot (V_j - V_{j+1}). \quad \text{Equation 2-14}$$

On the other direction, build the incompressible Bernoulli equation for this 1D pipe, shown as Equation 2-15.  $g$  is the gravitational acceleration constant.  $\Delta H$  is the hydraulic head loss due to the abrupt expansion. Substitute Equation 2-14 to Equation 2-15 to remove the pressure terms, the head loss could be written to Equation 2-16 which is exactly the Borda-Carnot equation.

$$\frac{P_j}{\rho g} + \frac{V_j^2}{2g} - \Delta H = \frac{P_{j+1}}{\rho g} + \frac{V_{j+1}^2}{2g}. \quad \text{Equation 2-15}$$

$$\Delta H = \frac{1}{2g} \cdot \left(1 - \frac{A_j}{A_{j+1}}\right)^2 \cdot V_{j+1}^2. \quad \text{Equation 2-16}$$

Combining Equation 2-14, Equation 2-15 and Equation 2-16 together, Equation 2-17 is derived which indicates that the convective pressure drop in TRACE not only includes the recoverable dynamic pressure drop but also implicitly includes an irrecoverable form pressure drop.

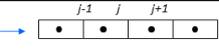
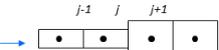
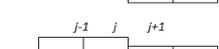
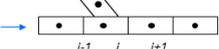
$$\Delta P = \rho \cdot V_{j+1} \cdot (V_j - V_{j+1}) = P_{j+1} - P_j = \left(\frac{\rho \cdot V_j^2}{2} - \frac{\rho \cdot V_{j+1}^2}{2}\right) - \frac{\rho}{2} \cdot \left(1 - \frac{A_j}{A_{j+1}}\right)^2 \cdot V_{j+1}^2. \quad \text{Equation 2-17}$$

**Dynamic Pressure Drop**   **Form Pressure Drop**  
**Recoverable**                      **Irrecoverable**

The implicit irrecoverable form pressure drop is exactly the source of inaccuracy introduced by TRACE. Five typical TRACE 1D hydraulic structure along with brief evaluations listed in Table 2-1 have been selected to analyze this inaccuracy.

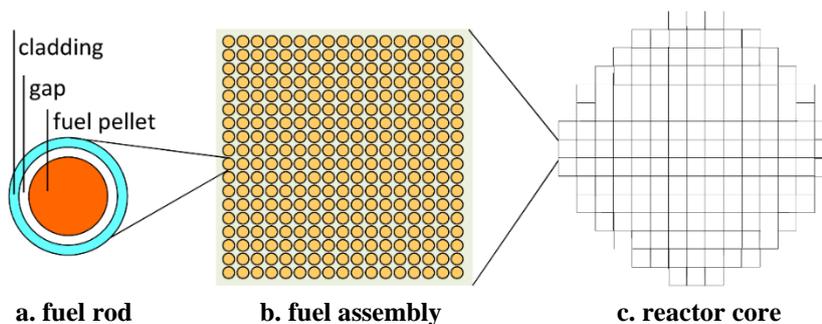
From this table, it becomes obvious that all flow phenomena associated with form pressure losses, TRACE applies either semi-empirically or full-empirically correlations to approximate the irrecoverable pressure drop. Moreover, in order to ensure the solution stability for the side junction flow out the problem, the should-be-there momentum sink is erased, which results in distorted fields. Moreover, TRACE uses built-in models to calculate the friction factors. However, they are all semi-empirical correlations and the users can even define those factors in the input files, which make the simulation more unreliable.

**Table 2-1 – Analyzed 1D hydraulic geometries in TRACE**

Geometries	Description	Model Evaluation	Accuracy
	Normal 1D PIPE	No irrecoverable.	Accurate.
	Abrupt Expansion	The irrecoverable is calculated semi empirically.	Comparatively accurate.
	Abrupt Contraction	The irrecoverable is calculated full-empirically.	Not accurate.
	Side Junction Flow In	A momentum source is added but calculated empirically.	Not accurate.
	Side Junction Flow Out	In order to ensure stability, the desired momentum sink is dropped leading to an additional artificial abrupt expansion.	Wrong.

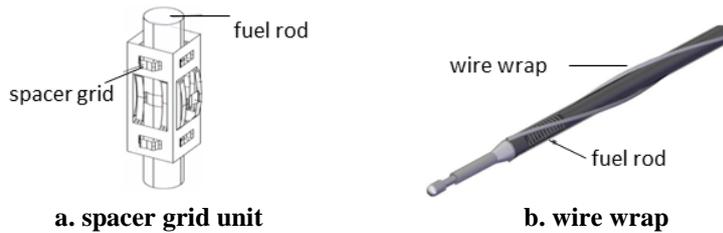
## 2.2 The sub-channel thermal-hydraulic code SubChanFlow (SCF)

SCF is a thermal-hydraulic sub-channel code developed at KIT for the simulation of fuel assemblies (Figure 2-3b) and cores (Figure 2-3c) of light water and innovative reactor systems. It is programmed in Fortran-2003 with dynamic memory allocation and equipping with external Graphical User Interface (GUI) platform for pre- and post-processing. SCF solves three mixed equations for mass, momentum, and enthalpy conservation for a quasi-3D single and two-phase flow and one additional equation for lateral transport. In order to close the system of equations, dedicated models are implemented for the prediction of wall drag and friction, boiling rate, and wall-to-liquid heat transfer correlations. A 2D (r-z) (refer to the Cylindrical coordinates in Figure 2-1a, here only  $r$  and  $z$  directions are considered) fuel pin heat conduction model is applied to describe the heat transfer from the fuel pellet over the gap and cladding (Figure 2-3a) to the coolant.



**Figure 2-3 – The cross-section diagram of PWR fuel rod (left), fuel assembly (middle), and reactor core (right).**

Pressure drop models are included for spacer grids (Figure 2-4a) of PWR and wire-wraps (Figure 2-4b) of fast reactors. In addition, simplified correlation is implemented to describe the fuel pin behavior (cracking, swelling, fission gas release) and the heat transfer over the fuel-cladding gap. SCF is also able to describe the interplay of the thermal-hydraulics with the neutronics.



**Figure 2-4 – The spacer grid (left) and wire wrap (right) for fuel rod (Yetisir, et al., 2018) (Song, et al., 2007).**

SCF performs a sub-channel level and a fuel assembly-level simulation of both rectangular and hexagonal fuel assemblies (Figure 2-3b displays the rectangular assembly) loaded in a core. The set of conservation equations can be solved for steady-state or transient problems.

SCF is able to simulate various coolant fluids, including water, lead, lead-bismuth, sodium, helium, and air. In SCF, the pressure, mass, and enthalpy are solved fully implicit for pure upward flow (because each external iteration step proceeds axially level by level starting from the bottom) while for recirculation and natural convection dominated flow, a semi-implicit solution approach is adopted. The convection term in the momentum and enthalpy equations is solved by a first-order upwind method. The boron concentration and the neutronic models are solved explicitly. The critical heat flux is determined by an iterative procedure.

For light water reactor applications, SCF is validated using appropriate experimental data representative for Pressurized Water Reactor (PWR) and Boiled Water Reactor (BWR) e.g. the NUPEC BFBT and PSBT tests (Imke & Sanchez-Espinoza, 2012) (Berkhan, et al., 2011). The data of the Karlsruhe compact sodium boiling loop (KNS) was used for the validation of the liquid metal heat transfer modules of SCF (Sanchez, et al., 2010). Moreover, SCF was already coupled with both stochastic such as SERPENT (Daeubler, et al., 2015) and MCNP (Ivanov, et al., 2011) and deterministic neutronic solvers e.g. PARCS (Basualdo & Sanchez, 2017), DYN3D-SP3 (Gonzalez-Vargas, et al., 2018) and it is part of the European simulation platform named NURESIM which is built on the open-source SALOME-platform (Calleja, et al., 2014).

## 2.3 The computational fluid dynamic code TrioCFD

TrioCFD is an open-source CFD code based on the TRUST-platform (TRio\_U Software for Thermo-hydraulics) being developed by CEA for scientific and industrial applications related to the nuclear industry (Bieder & Graffard, 2007).

TrioCFD includes many physical models and it applies advanced numerical methods to solve various problems varying from local two-phase flows to turbulent flows on industrial facilities such as a nuclear reactor or part of it. In TrioCFD, the Reynolds-Averaged Navier-Stokes (RANS), the Large Eddy Simulation (LES), and the Direct Numerical Simulation (DNS) are available for the solution of fluid dynamic problems. The governing equations are solved with a staggered finite-volume approach. TrioCFD is able to generate robust meshes or import meshes from other software. The code supports full parallelepiped and tetrahedral structured or unstructured meshes. The spatial discretization of the parallelepipeds is based on Finite Volume Differences (V.D.F.) and of the tetrahedron is based on Finite Volume Elements (V.E.F.).

TrioCFD is an object-oriented code written in C++ and it is composed of over 1500 classes. The data structure and functions of TrioCFD is appropriate for massively parallel computing where the data transfer and communication between the cores use the MPI protocol (Message Passing Interface).

Extensive verification and validation of TrioCFD are underway using different test data (Angeli, et al., 2015). Typical applications of TrioCFD are e.g. boron mixing, main steam-line break, and induced break severe accident (Bieder & Graffard, 2007).

In the last years, TrioCFD was coupled with other codes and for this purpose powerful mesh-manipulation library such the MEDCoupling and the coupling interface ICoCo (Interface for Code Coupling) were developed. It is worth to note that compulsive modifications and re-compilation of the source code are required in order to adapt the stand-alone TrioCFD code version for coupling with e.g. system thermal-hydraulic code CATHARE (Bavière, et al., 2014).

Since TrioCFD is open-source and already applied for nuclear engineering equipped with powerful in-built libraries such as the MEDCoupling and the generic built-in coupling-oriented and object-oriented interface named ICoCo, it has been selected for the multi-scale developments in the frame of this study. The experience gained in different European projects e.g. NURISP and NURES SAFE (NURESAFE, 2019) devoted to multi-physics and multi-scale code coupling and the unique and innovative features of the code coupling approach based on the SALOME-platform, ICOCO, and the MEDCoupling libraries (SALOME, 2019) have been the major reasons to rely on these methodologies. On the other hand, open-source codes are not comparable with commercial CFD-codes regarding numerical stability, user-friendliness, computational efficiency when solving very large problems.

# 3 Strategies for Multi-Scale Coupling of TRACE/SCF and TRACE/TrioCFD

## 3.1 Principle approaches

This study focus is on the multi-scale coupling of the following codes:

- TRACE and SubChanFlow (SCF).
- TRACE and TrioCFD.

For the coupling of TRACE with SCF, the Exterior Communication Interface (ECI) has been selected mainly due to the following reasons:

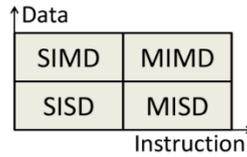
- a) ECI is a parallel interface, which directly handles the unit of network communication i.e. SOCKET (the base of MPI protocol). The data transfer is highly flexible and efficient;
- b) ECI is an in-built module of TRACE developed for code coupling but not yet exploited. Hence, no significant programming effort to TRACE is necessary for coupling with other codes;
- c) ECI requires that the codes to be coupled must be modularized and that each code is equipped with the ECI-module. Hence, SCF needs to be modularized first and then an ECI-interface must be developed for SCF despite the complexity of ECI.

It is worth to mention that ECI is deeply embedded and distributed within the TRACE-source code assuring high flexibility. The coupling of TRACE and TrioCFD is based on the Interface for Code Coupling (ICoCo) approach since TrioCFD includes already an ICoCo interface developed for the coupling with the neutronic code DYN3D (Grahn, et al., 2017) and with a system thermal-hydraulic code CATHARE (Tenchine, et al., 2012). Additional reasons for the selection of ICoCo are listed below:

- a) ICoCo is a highly standardized interface developed within the European NURISP project (Deville & Perdu, 2012), it is highly extendable and easy to maintain;
- b) ICoCo is the first well-developed and in-built module in TrioCFD. Hence, limited programming effort is necessary to wrap TrioCFD with ICoCo;
- c) ICoCo is a wrapper of the real solution engine meaning that no significant modifications of the source code of TRACE are necessary.

### 3.1.1 The Exterior Communication Interface (ECI)

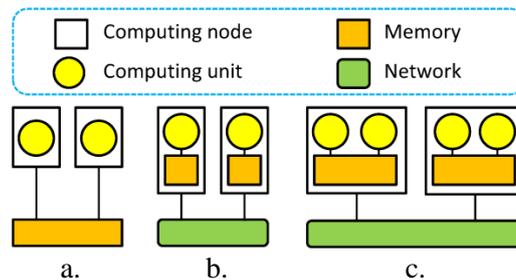
Before the details of ECI are explained, basic concepts about computer architecture are recalled. Computer architectures can be classified as follows, see Figure 3-1: Single Instruction Single Data (SISD), Single Instruction Multiple Data (SIMD), Multiple Instruction Single Data (MISD) and Multiple Instruction Multiple Data (MIMD). The SISD machine is a serial machine since it just deals with one instruction and one data at a time. The other three are Parallel machines. There are not too many MIMD instances in real life since it is unusual to process one data with more than one instruction. However, a typical MISD example could be the computer in space shuttle which is designated to guarantee the shuttle's stability and safety. As to SIMD and MIMD, they both process multi-data at a time. The difference is that the former processes data with a single instruction while later with instructions. MIMD is more popular than SIMD in life and can be further classified into three types based on the configuration of the computing unit and memory.



**Figure 3-1 – Computer architecture acronym classification.**

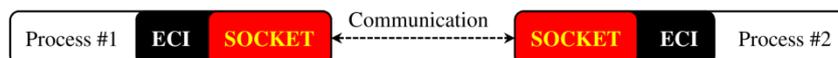
The MIMD is the most popular computer architecture, and it can be further grouped into three types based on the configuration of the computing unit and memory, as shown in Figure 3-2. For shared memory computers, Figure 3-2a, memory could be accessed by every computing node (or computing unit). Hence, the nodes could transfer data directly through shared memory. Advantage of this kind of architecture is speed. However, good performance can only be expected when there are not too many nodes. In such a case, access violation can happen when too many nodes try to access shared memory.

Utilizing a distributed memory architecture solves this problem, as depicted in Figure 3-2b. The nodes process data in their local memory and they exchange data through the network. However, the bandwidth and speed of the network become the new system bottleneck. Therefore, the distributed shared memory architecture, as illustrated in Figure 3-2c, was proposed. Compared with distributed memory, the distributed shared memory machines have more than one unit located in one node with shared memory, which could significantly enhance the computing capability of one single node and thus reduce the network demand.

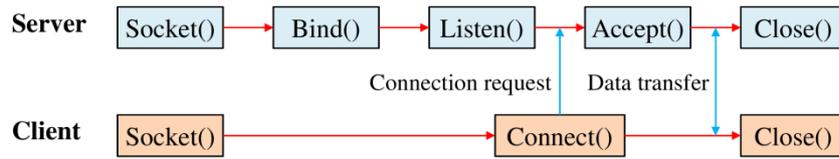


**Figure 3-2 – MIMD machine classification: a) shared memory, b) distributed memory, c) distributed shared memory.**

Recently, the new ECI-module was implemented in TRACE for multi-tasking capability paving the way for coupling of TRACE with other solvers. ECI offers two different message-passing mechanisms: 1) Shared memory communication (OpenMP is a typical shared memory model); 2) SOCKET-based communication. In Figure 3-3, the communication mechanisms based on SOCKET is presented. SOCKET is normally the bottom level of all Message-Passing mechanisms through the network. There, Process #1 could be TRACE and Process #2 could be another code e.g. SCF. For efficient communication of both codes, each code must have an ECI-module. The following Message Passing Interface (MPI) and Parallel Virtual Machine (PVM) are all SOCKET-based approaches. For a typical socket-based network, six functional routines need to be called in sequence to establish the communication channel and to transfer data: socket, bind, listen, connect, accept and close. A general overview of this process is shown in Figure 3-4.



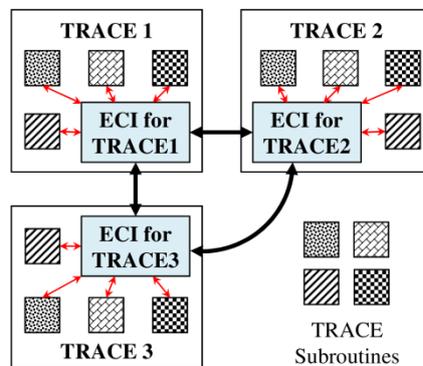
**Figure 3-3 – ECI SOCKET communication mechanism.**



**Figure 3-4 – Procedures of SOCKET-based communication.**

Socket endows the coupled system with a quite flexible and expandable capability, especially if it is within distributed memory computational systems. Consequently, the ECI-module has been selected for the coupling of TRACE with SCF. It is worth to mention that the ECI-module of TRACE is closely linked to data flow (synchronization points) and file structure of TRACE. For the coupling of SCF with TRACE, a new ECI-module for SCF needs to be developed. In addition, the new ECI-module must be embedded within the data flow of SCF.

The ECI-module binds codes together by implementing a multi-task system (Mahaffy & C., 2003). For this purpose, several tasks have to be identified. Each pair of the involved tasks can directly exchange data with each other through SOCKET; it behaves like a server-less system, sees Figure 3-5. This is quite different from normal network communications, which usually include one server and several clients. Moreover, the number of coupled codes is not limited and additional tasks may be conveniently merged into the system taking benefits from the socket model properties. For instance, in Figure 3-5 there are three TRACE simulations forming a multi-task system based on ECI. Furthermore, more TRACE simulations or other codes could be easily included in the system, if these codes are equipped with their own ECI-module.



**Figure 3-5 – Multi-task system based on ECI.**

Typical tasks of ECI are the generation of the multi-task system, coordination, and synchronization of the inter-task communications. In this sense, the multi-task system is a concept while ECI is the tool for the implementation of the concept. Normally, ECI uses three steps to establish a multi-task system, as illustrated in Figure 3-6:

- Firstly, all codes broadcast to all other codes through their ECIs and establish connections with the identified ECIs, where potential communications are expected.
- Then, the connected task-pairs perform further recognition to connect the actual components.
- Finally, the variables that store the real data for the components are connected together. Up to this point, the communication channels for data transfer are established and ready for use.

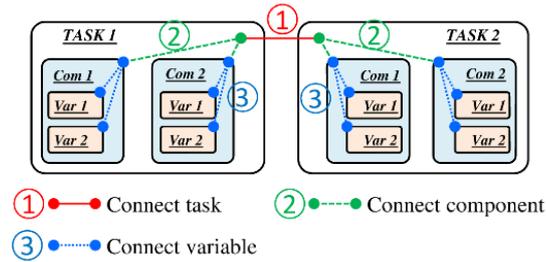


Figure 3-6 – Three steps to establish a multi-task system based on ECI.

### 3.1.2 The Interface for Code Coupling (ICoCo)

ICoCo is an object-oriented generic interface for code coupling developed by the European NURISP project for the multi-scale thermal-hydraulic coupling of CATHARE and TrioCFD. Each involved code in the coupled system is referred to as the “Problem” notion, which can be treated as an object computing a time-dependent simulation. ICoCo specifies several methods that the problem has to provide as well as the descriptions of what they are supposed to do. It does not contain any real functional codes but just poses a framework and standard. Developers have to fill in the frame and establish the connections between ICoCo and the target codes. For instance, ICoCo defines a function named “SetDatafile” under its standard and tells the developer that this function should take the responsibility to define the path of the input and output directories as well as manage all the related files. However, it does not contain any practical coding to implement that function. Developers have to do the work by themselves.

As the supplement of basic functions, ICoCo also supplies the methods to insert various input ports and output ports to the coupled-codes making the inter-code interaction quite flexible and convenient. With the properly developed functions and interactive ports, the codes now are able to form a coupled simulation scheme with the help of a supervisor, which completes the coupling system. This is a typical server-client system (the supervisor is the server and TRACE with TrioCFD are the clients) and its general form is shown in Figure 3-7.



Figure 3-7 – Scheme of the multi-scale coupling of TRACE and TrioCFD based on ICoCo.

As shown in Figure 3-7, ICoCo is a cross-language interface. Two aspects are involved: 1) ICoCo itself could be written in both Python and C++; 2) The codes to be coupled could be written in Python, C++, and Fortran. In this dissertation, TRACE is written in Fortran and TrioCFD in C++. According to ICoCo, three options are available for the supervisor: C++, Python, and SALOME. For the first two options, users have to develop the supervisor scripts in either Python or C++, while for SALOME, the built-in YACS (SALOME, 2019) module offers a Graphic User Interface (GUI) to organize the calculation chains and computational routes by just “drag” and “click” using the mouse. SALOME is the best choice for small-scale solvers because the calculation can be conveniently monitored. However, due to the poor support to MPI protocol, SALOME may not be the first choice for simulations where a parallel capability is necessary. Instead, either C++ or Python script is the best choice. Since MPI is important for TrioCFD, it plays a key role in the supervisor selection. In this work, C++ scripts are selected as the supervisor because it not only supports MPI but there is a community and accumulated experience using MPI. It is worth to note that though

Python supports MPI, this capability was not widely used as C++. Nevertheless, Python is still a promising alternative benefiting from its rapidly growing user community.

As discussed in Chapter 1, mesh and field mapping is one of the critical challenges of multi-scale coupling. A new and powerful mesh-field manipulating library – MEDCoupling has been inherently adopted by ICoCo to handle mesh and field mapping issues between coupled codes. This library is also an in-built module in TrioCFD and in the SALOME platform. Because of the peculiarities of the MEDCoupling library and the request of flexible coupling, finally, two essential prerequisites are necessary for a successful coupling based on ICoCo, Figure 3-7:

- Each code should have explicit meshes which could be used for post-processing, field mapping, and interpolation with other codes;
- Each code must be split into several functional components for flexible coupling.

These two requisites are mandatory before any code coupling based on ICoCo-can be implemented. This is the case of the TRACE code. Hence, in the frame of this dissertation, several source code modifications have been necessary.

## 3.2 Selection of the coupling methodologies

The following two sections present and discuss the criteria considered for the selection of the different coupling strategies and the selected coupling approaches.

### 3.2.1 Selection criteria for multi-scale coupling

Following Figure 1-8, five main strategies for multi-scale coupling exist. All sub-strategies are evaluated by a standardized scoring mechanism. The evaluation criteria along with their weighting factors (the degree of importance of the criteria to the coupling) are selected and listed in Table 3-1.

**Table 3-1 – The evaluation criteria with the weighting factors of the five main coupling strategies.**

Strategy Group	Evaluation criteria-weighting factor				
<b>S1: Architecture</b>	Efficiency-5	Flexibility-4	Scalability-3	Maintainability-2	Simplicity-1
<b>S2: Operation mode</b>	Efficiency-4	Scalability-3	Maintainability-2	Simplicity-1	
<b>S3: Domain coupling</b>	Robustness-3	Realizability-2	Simplicity-1		
<b>S4: Field mapping</b>	Accuracy-5	Capability-4	Universality-3	Efficiency-2	Simplicity-1
<b>S5: Temporal coupling</b>	Accuracy-4	Robustness-3	Efficiency-2	Simplicity-1	

From the general perspective of the coupling approaches, the more efficient the better, the more accurate the better, the more stable the better and the easier the better. Three scores are used to evaluate the approaches, where 0 represents poor, 1 for normal and 2 is good. Five criteria are proposed for coupling architecture and the weighting factors are assigned according to their importance. The final evaluating scores are calculated in Table 3-2.

- 1) **Efficiency:** first of all, the data within an internal coupled system is exchanged via memory, which is the fastest way of data transfer. Nevertheless, the inter-process-parallel capability is impossible under this architecture. As to the server-client and server-less coupling, though the popular data transfer is through networks and the speed is slower than that via memory, the shared memory could achieve rather high data transfer rate. Moreover, the potential inter-process-parallel capability of the two architectures makes it possible to handle big cases by dispersing the tasks to plenty of processors, which increase the calculation efficiency. These are the reasons why the server-client and server-less coupling architectures make two scores

while the internal coupling gets only one. As to the last option - the coupling through IO files, it is actually not a fully independent coupling architecture from the other three. Because the IO data communication must be realized by, at least one of the other three options. Nevertheless, the reason to include it in the lists is that the data transfer is not through neither memory nor networks but files, which is the most inefficient coupling method.

- 2) **Flexibility:** The codes involved in a server-less architecture could communicate with each other through their private channels upon demand. It is easy to assign and change specified data at specified channels. While for that in a server-client system, all of the information has to be in the control of one code or a standalone supervisor. Each client code is isolated from the other client codes. Changes or assignments of the transferred data usually go to two channels and the server. So, two scores are for the server-less and one is for sever-client. The internal coupling and the IO coupling are the least-flexible architectures because everything is frozen once the coupling is established for the former method. Any changes to the data lead to modifications to the entire coupling system. As to the latter method, the condition is even more complex since new logics have to be developed in order to change the writing or reading content in the files. Therefore, no score was given to these two architectures.
- 3) **Scalability:** For the server-client and server-less coupling, the new well-developed component is conveniently plugged into the coupling system without significant modifications to the existing codes. Thus, they both get two scores thanks to their good scalability. Due to the same drawback as explained in the previous paragraph, if new code integration is necessary, the entire existing system of the internal coupling have to be re-organized in total and the IO writing and reading logics of the existing IO coupling system have to be updated to include the new components. This is a repeatable and error-prone work. Thus no score is assigned to both methods.
- 4) **Maintainability:** This character has positive correlations with the flexibility and scalability of the coupled systems. For the server-client and server-less system, the codes update or upgrade could focus on specified components while keeping others untouched. Therefore, they both make two scores. For the internal and IO coupling, as it is mentioned above, no matter where the maintenance takes place, the entire system is affected. This is not the ideal way to maintain the codes while neither the worst way. Therefore, they both get one score here.
- 5) **Simplicity:** Normally, the IO coupling is the easiest method because there is no need to dig into the codes numerical source but just concerns the input and output logic. The internal coupling is a little bit complex because the codes numeric has to be carefully inspected in order to properly organize the computation sequence of the coupling system. The worst situation regarding simplicity is found in case of the server-client and server-less system because not only the codes numeric should be inspected but also the parallel mechanisms have to pay lots of effort in order to achieve successful synchronization of the codes.

**Table 3-2 – Evaluation of the coupling codes’ architecture.**

<b>S1: Architecture</b>	<b>Effic-5</b>	<b>Flexi-4</b>	<b>Scala-3</b>	<b>Maint-2</b>	<b>Simp-1</b>	<b>Final score</b>
<b>S1.1 Internal coupling</b>	1	0	0	1	1	$1 \times 5 + 0 \times 4 + 0 \times 3 + 1 \times 2 + 1 \times 1 = 8$
<b>S1.2 IO files coupling</b>	0	0	0	1	2	$0 \times 5 + 0 \times 4 + 0 \times 3 + 1 \times 2 + 2 \times 1 = 3$
<b>S1.3 Server-client</b>	2	1	2	2	0	$2 \times 5 + 1 \times 4 + 2 \times 3 + 2 \times 2 + 0 \times 1 = 24$
<b>S1.4 Server-less</b>	2	2	2	2	0	$2 \times 5 + 2 \times 4 + 2 \times 3 + 2 \times 2 + 0 \times 1 = 28$

The evaluation of the operation mode is closely relevant to that of the coupling architecture and hence it can be treated as its subset to some extent. The scores are calculated in Table 3-3. From the evaluation of the coupling architecture, the parallel operation has overwhelming superiority over the serial mode on efficiency and scalability. While conversely, that outstanding superiority results in the complexity of the maintenance and development of such coupling approaches. It is worth to note that the “parallel” mode here not only refers to the inter-process but also the in-process parallelization. So, for the internal coupling, though the inter-process parallel is not available, the process itself could be

parallelized e.g. the CFD codes. As a consequence, the maintainability here focuses on the entire system while not specified components or functions.

**Table 3-3 – Evaluation of the operation mode of the coupling system.**

<b>S2: Operation Mode</b>	<b>Effic-4</b>	<b>Scala-3</b>	<b>Maint-2</b>	<b>Simp-1</b>	<b>Final score</b>
<b>S2.1 Serial</b>	0	0	2	2	$0 \times 4 + 0 \times 3 + 2 \times 2 + 2 \times 1 = 5$
<b>S2.2 Parallel</b>	2	2	0	0	$2 \times 4 + 2 \times 3 + 0 \times 2 + 0 \times 1 = 14$

Regarding the domain coupling, Table 3-4 shows the scores assigned to the two options. As explained before, the domain-overlapping method is more robust and hence superior compared to the domain-decomposition method. The criteria realizability and simplicity is related to the evaluation of the workload for users and developers. Users have to split the integral system code model into several pieces in order to connect with the sub-channel or CFD-pieces if the domain-decomposition method is adopted. This is a repeatable, error-prone work and not user-friendly. On the contrary, in the case of the domain overlapping method, the model of the system code can be directly used. The only challenging task is the identification of the proper location in the source code where to get and put data. Moreover, the use of modern mesh-manipulation libraries e.g. the MEDCoupling library could simplify this identification work and high-level automation of the domain matching can be implemented. On the contrary, the easier for the user the more difficult for the developer, so the domain overlapping is rather tough to implement.

**Table 3-4 – Evaluation of the domain coupling approaches.**

<b>S3: Domain Coupling</b>	<b>Robust-3</b>	<b>Realize-2</b>	<b>Simp-1</b>	<b>Final score</b>
<b>S3.1 Decomposition</b>	1	0	2	$1 \times 3 + 0 \times 2 + 2 \times 1 = 5$
<b>S3.2 Overlapping</b>	2	2	0	$2 \times 3 + 2 \times 2 + 0 \times 1 = 10$

Table 3-5 shows the results of the evaluation of the three field-mapping approaches using the criteria described above. It is worth to note that the scores for the criteria are assumed to be well-developed.

- 1) **Accuracy:** If the mesh-position relationships are defined by the users, the mesh overlapping and interpolation can hardly be correctly calculated unless the meshes are all of the regular shapes and regularly arranged. Poor mesh mapping may lead to errors. So, the user-manual scores zero. Alternatively, the user can develop special routines to handle this problem in a more convenient and accurate manner. However, this is an open problem depending heavily on the selected method and level of the toolkit. Thus the user-subroutine gets one score. Finally, the third-party library is characterized by high accuracy and interpolation functions. Normally, this kind of library is powerful enough to calculate the mesh mapping more precisely. So, it scores 2 here.
- 2) **Capability:** Generally speaking, the third-party library is more powerful than other alternatives and it can handle more complicated cases than the private subroutines. Thus, the library makes 2 and the user-subroutine makes 1. The worst is user-manual.
- 3) **Universality:** the third party library is able to handle various mesh or field mapping types and it is accessible for various coupled codes. So, it is scored 2. The user-subroutines are normally more specific to special problems. Thus it gets 1. If users define the mesh relations by hand, there is no universality at all.
- 4) **Efficiency and simplicity:** The third party library has the superior capability and can be adapted to various situations and cases. Consequently, the inter logics must be complex and special effort has to be made in order to properly use it. For all other options, the logics can be quite simple and just focus on specified problems. Therefore, they could reach high efficiency compared to the third-party libraries.

**Table 3-5 – Evaluation of the field mapping approaches.**

<b>S4: Field Mapping</b>	<b>Accur-5</b>	<b>Capab-4</b>	<b>Univ-3</b>	<b>Effic-2</b>	<b>Simp-1</b>	<b>Final score</b>
<b>S4.1 User manual</b>	0	0	0	2	2	$0 \times 5 + 0 \times 4 + 0 \times 3 + 2 \times 2 + 2 \times 1 = 6$
<b>S4.2 User subroutine</b>	1	1	0	1	1	$1 \times 5 + 1 \times 4 + 0 \times 3 + 1 \times 2 + 1 \times 1 = 12$
<b>S4.3 3<sup>rd</sup> party library</b>	2	2	2	0	0	$2 \times 5 + 2 \times 4 + 2 \times 3 + 2 \times 2 + 0 \times 1 = 24$

The last evaluation is related to the temporal coupling. The scores are presented in Table 3-6. The difference between the three temporal coupling methods has already been discussed in chapter 2. The data convergence between the coupled codes is not guaranteed under explicit coupling while for semi-implicit and implicit, the solution consistency between codes within one time step can be both achieved through inner-iteration techniques and inherent matrix solutions, respectively. This is why both the accuracy and robustness of the last two methods get two scores while the explicit gets one. Moreover, for most cases, the system efficiency keeps growing from explicit, semi-implicit to implicit coupling, as a benefit from the consistent fields between codes within one time step. On the opposite direction, the coupling system becomes more complex due to the applied techniques accelerating the convergence rate.

**Table 3-6 – Evaluation of the temporal coupling approaches.**

<b>S5: Temporal Coupling</b>	<b>Accur-4</b>	<b>Robust-3</b>	<b>Effic-2</b>	<b>Simp-1</b>	<b>Final score</b>
<b>S5.1 Explicit</b>	1	1	0	2	$1 \times 4 + 1 \times 3 + 0 \times 2 + 2 \times 1 = 9$
<b>S5.2 Semi-implicit</b>	2	2	1	1	$2 \times 4 + 2 \times 3 + 1 \times 2 + 1 \times 1 = 17$
<b>S5.3 Implicit</b>	2	2	2	0	$2 \times 4 + 2 \times 3 + 2 \times 2 + 0 \times 1 = 18$

Considering the above tables for reference, the higher the score is, the better is the performance of the coupling system; or in other words, the more advanced is the coupling system. Moreover, the scores supply a quantitative standard for the selection of the multi-scale coupling approach.

### 3.2.2 Selected coupling methodologies for TRACE/SubChanFlow(SCF) and TRACE/TrioCFD

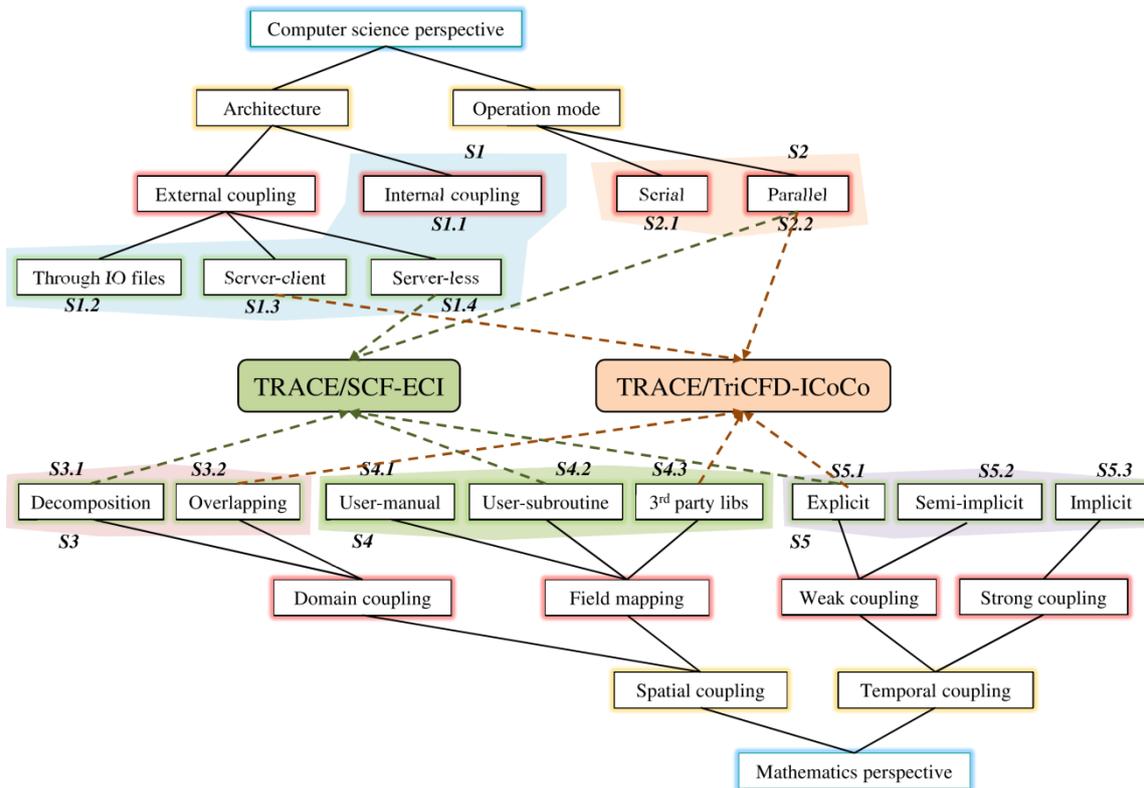
Since ECI and ICoCo have special features, some coupling approaches are inherently determined for the two-coupled pairs. The coupling of TRACE/SCF using ECI is a server-less and parallel coupling system that takes profit of the ECI-capabilities. The coupling of TRACE and TrioCFD using ICoCo works as a server-client coupling system, where the coupled codes can be implemented both in serial and parallel mode. Since the parallel capability is very important for the CFD code, the MPI-protocol is applied. The mesh-manipulation in ICoCo is inherently handled by the MEDCoupling library.

The domain-overlapping approach assures the physical fluxes and their gradients to be consistent. Thus the system's mathematical steadiness and the differentiability are guaranteed, while domain-decomposition is tricky to achieve this goal. Despite its superior, only a limited thermal-hydraulic coupled codes use the domain-overlapping method due to its complexity compared to the domain-decomposition method. For example, three-dimensional data e.g. volume fields are to be exchanged in the domain-overlapping approach while only two-dimensional boundary data is exchanged in the domain interfaces when domain-decomposition method is applied. Such a selection is the concession to avoid sophisticated operations and interactions between the codes' meshes, since neither the user manual definition nor the user-defined subroutine is practical to handle complicated coupling cases. Nevertheless, the use of MEDCoupling library makes the implementation of domain-overlapping approach feasible and attractive. Hence, the domain-overlapping approach has been selected for the coupling of TRACE and TrioCFD.

On the contrary, since ECI concerns more about the data synchronization rather than the meshing capability and there is no motivation for developing powerful meshing capabilities for the TRACE and SCF coupling, the domain-decomposition is the best choice. However, even though the manipulation of three-dimensional meshes is not needed, the two-dimensional mesh and field mapping are required for flexible and accurate inter-code data exchange. In order to improve the accuracy, capability, and efficiency of the coupled codes, a user-defined toolkit is selected for field mapping between TRACE and SCF. The details of this toolkit are available in Appendix A.

Up to this point, the S1 to S4 strategies are appropriate for the two envisaged code coupling problems and the last open selection is S5 for temporal coupling. The fully implicit coupling is rather complicated to be implemented in two solvers that differ from each other significantly e.g. TRACE, SCF, and TRACE, TrioCFD. Therefore, the preferable temporal coupling method is either an explicit or semi-implicit approach. Considering that the current two coupling systems mainly focus on function realization and do not go deep into numerical optimization, the explicit temporal coupling method is the one selected in this study. However, a phenomenon like shock wave, bubble collapse sometimes is in consideration which could not be properly handled by explicit coupled code without significant modification, verification and validation. The semi-implicit coupling approach is desired for those cases. Nevertheless, the upgrade from explicit to semi-implicit is straightforward.

Consequently, the TRACE/SCF coupling is based on a server-less-parallel-domain-decomposition-user-subroutine-explicit implementation and the TRACE/TrioCFD coupling relies on a server-client-domain-overlapping-third-party-library-explicit approach. Figure 3-8 exhibits the selected multi-scale coupling approaches, which will be implemented in the frame of this dissertation.



**Figure 3-8 – The selected approaches for the multi-scale coupled codes TRACE/SCF and TRACE/TrioCFD.**



## 4 The Multi-Scale Simulator TRACE/SCF based on ECI

This chapter first demonstrates the development of ECI for SCF. Then each aspect of the coupled code of TRACE/SCF is presented. The verification and validation are done first by the analysis of an academic case and then by the analysis of the VVER-1000 coolant-mixing benchmark (Kolev, et al., 2010). In both investigations, the extended capability of the coupled code to describe the coolant mixing inside the core is evaluated and the obtained results are compared to the ones of the stand-alone TRACE-simulations.

### 4.1 Development of ECI for SCF

The major advantage of ECI is the capability to parallelize the calculation of the coupling system randomly. Thereby, the coupled codes may be potentially extended to be run on powerful distributed computing systems. However, this natural merit also leads to a crucial challenge for the coupled system: synchronization. The precondition to implementing the coupling based on ECI is to assure that the “right data” is transferred at the “right place” and at the “right time”. This is a complex task, especially for the multi-task parallel system. Hence, the key challenges for the development of an ECI-module for SCF are the questions: how to determine which data to be transferred at which time and through which channel in order to implement tight and accurate data synchronization.

The ECI-module adopts dozens of synchronization points e.g. for a TRACE-TRACE coupling, which strictly reflect the solving procedures of TRACE set of equations (in an implicit coupling approach). However, since the logic of the SCF-solver engine differs from that of TRACE, only a subset of the synchronization points are used for the coupling of TRACE and SCF. In order to develop the ECI-module, the SCF-source code is first re-structured. Then, the ECI-module has been developed and implemented into SCF. Figure 4-1 shows the difference between the data transfer between two TRACE-processes and between TRACE and SCF.

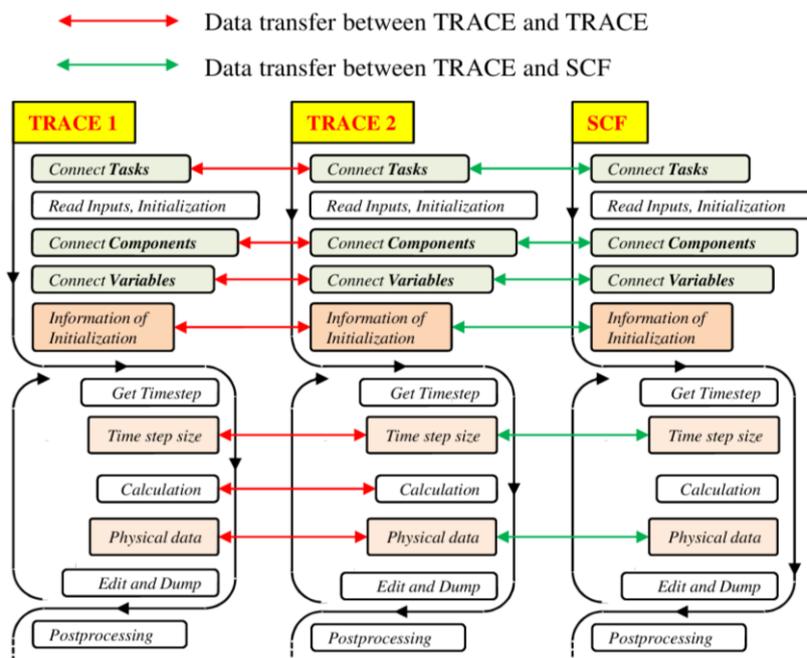
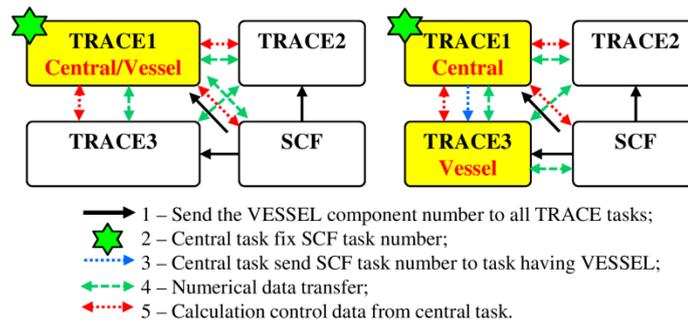


Figure 4-1 – The data transfer between TRACE/TRACE and TRACE/SCF based on ECI.

Before the execution of the codes, first the communication channels for the tasks are established. Then, each code reads the inputs and executes the initialization. After the identification of the components and data belonging to other tasks, the channels between components and variables are sequentially established. Now, the system is ready for data transfer. Some global flags and geometry parameters (the initialization information) are first transferred i.e. before the time-step iteration loop. When the calculation enters into the time-step loop, at each time step, the time-step size is calculated first. Then, the time-step size is transferred. Afterwards, the numerical solution is performed. TRACE and TRACE could communicate during the calculation because they are implicitly coupled. There is no data exchange between TRACE and SCF. Finally, the computational results of the current time step are exchanged.

With the implementation of ECI in SCF, the multi-tasking capability is also feasible. In this coupling approach, SCF simulates the core and TRACE simulates the rest of the Reactor Pressure Vessel (RPV) using the 3D VESSEL-component. In case that a TRACE-problem is solved using the multi-tasking capability, more than one TRACE-executables could be present. In such a situation, SCF must be able to identify the target TRACE containing the VESSEL-component to establish the communication channel between TRACE and SCF while all other channels with TRACE that do not contain the VESSEL-component are in silence mode. In general terms, it can be done also manually by specifying special parameters in the source code. However, this is not the most convenient option since it complicates the maintenance of the coupled code. Hence, an automatic mechanism that enables SCF to find the target TRACE among all the different tasks have been developed and implemented. A scheme of this automatic approach including the different function is illustrated in Figure 4-2.



**Figure 4-2 – Automatic mechanism of ECI in SCF to identify the “target TRACE” using the auto-location function.**

- 1) First, SCF broadcasts the “wish number” of the VESSEL-component to all TRACE tasks.
- 2) Then, each TRACE checks whether they have the VESSEL-component.
  - a) If the “central TRACE-simulation” contains the target VESSEL, it writes an SCF’s task number into memory and establishes the communication channel for the numerical data transfer.
  - b) If the “central TRACE-simulation” does not have the target VESSEL, it broadcasts the SCF’s task number to all other TRACE-simulations. The TRACE-simulation with the VESSEL-component writes the SCF’s task number into memory and establishes the communication channel with SCF for numerical data transfer.
- 3) Next, numerical data is exchanged through numerical channels.
- 4) At the same time, the variables controlling the calculation are broadcasted by the “central TRACE-simulation” to all other involved codes or tasks. They are also gathered from all other tasks to the central task.

## 4.2 Description of the coupled system TRACE/SCF-ECI

In this subchapter, the domain-decomposition, the user-defined subroutine, and the explicit temporal coupling approaches are described.

### 4.2.1 The domain decomposition coupling approach

In order to illustrate the challenges related to the domain decomposition method, Figure 4-3a represents a TRACE model of a PWR using the 3D VESSEL-component to represent the Reactor Pressure Vessel (RPV) a FILL- and BREAK-components to represent the inlet and outlet boundary conditions. In this model, the core is represented by nine axial elevations ( $z$ ), one ring ( $r$ ), and four azimuthal sectors ( $\theta$ ). The view A-A displays the downcomer and the core, where the downcomer is represented by four azimuthal sectors and the second radial ring. Figure 4-3b shows the SCF-model of the same core represented by nine channels ( $x, y$ ) with nine axial meshes ( $z$ ) for the purpose of illustration only. One could say that the core is represented by nine Cartesian meshes of SCF.

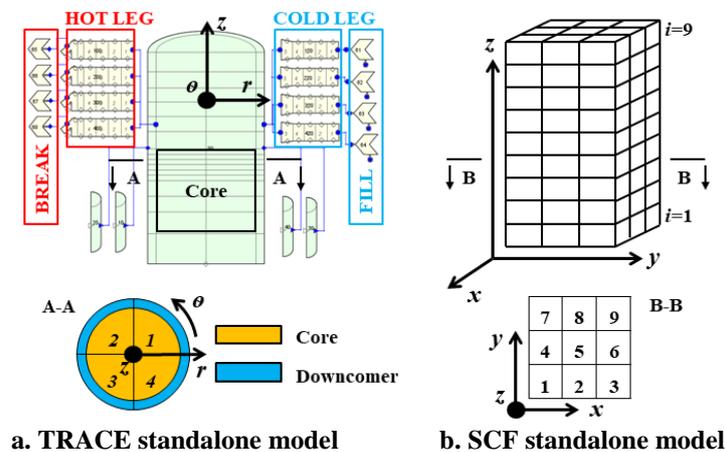


Figure 4-3 – Standalone models of TRACE and SCF representing a 4-loops PWR vessel and core.

In order to avoid the three-dimensional fields (physical variables) exchange and simplify the implementation process, the domain-decomposition method is used for the coupling of TRACE and SCF. In the frame of this method, SCF predicts the core behavior and TRACE solves the rest of the RPV, as shown in Figure 4-4. Within the system, two interfaces (2D planes) are defined for the data exchange (2D boundary conditions): one below and the other above the core.

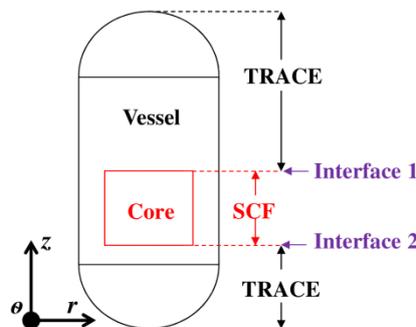


Figure 4-4 – Sketch of the domain decomposition approach for multi-scale coupling.

For the implementation of the domain-decomposition based on ECI, the original TRACE model of the RPV is modified as shown in Figure 4-5. There, no core at all is considered in the TRACE model. At the core inlet plane, four PIPE- and BREAK-components are connected to the four sectors of the first ring below the core. In a similar manner, at the core outlet, each of the four core sectors is connected with a PIPE and FILL-component. These new TRACE-components (FILL, BREAK, and PIPE) are indeed the communication interfaces on TRACE side for the data exchange with SCF. The data of SCF at the core entrance and at the core outlet is sent to TRACE by written them into the FILL and BREAK-components as boundary conditions. This data is used by TRACE to solve the RPV-domain without the core. Then, the TRACE data are passed to SCF by means of the PIPE-components connected to the FILL and BREAK-components.

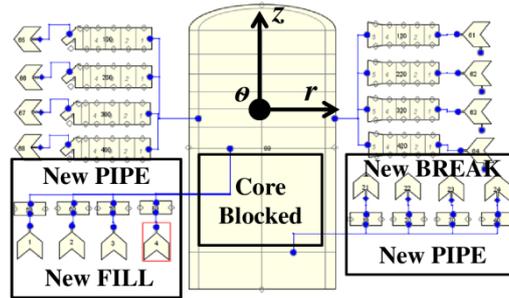


Figure 4-5 – Revised TRACE model for coupling.

The SCF model of the core remains unchanged. However, several modifications of the SCF-source code are necessary. Remember that SCF represents the core in quasi-three dimensions, where the core is subdivided into many axial cells, Figure 4-6. Physical data of a simulation is stored in the cells. In addition, SCF uses a so-called “ghost cell” at the bottom and top of the core layers in order to store the user-defined boundary conditions, Figure 4-6. But the “ghost cell” is not included in the real geometry. In this implementation, SCF writes data from TRACE at two ghost cells e.g. boundary conditions and send data from the two actual cells adjacent to ghost cells to TRACE.

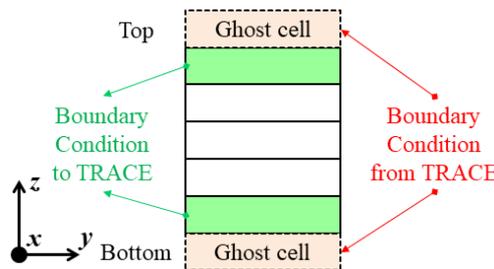


Figure 4-6 – SCF axial discretization of the core showing the ghost cells and data transfer.

In Figure 4-7, a more general and integrated diagram illustrates the data transfer mechanism between TRACE and SCF. Mass flow, coolant temperature, and solute concentration are extracted from SCF “top cells” and they are transferred to TRACE new FILL-components as the inlet boundary condition. Coolant pressure is extracted from SCF “bottom cells” and it is transferred to TRACE new BREAKs as the inlet boundary condition. In turn, mass flow, coolant temperature, and solute concentration are extracted from TRACE “new PIPES” connected to the “new BREAKs” and transferred to SCF “bottom ghost cells” as the inlet boundary condition. Coolant pressure is extracted from TRACE new PIPES connected to the new FILLs and it is transferred to SCF “top ghost cells” as the outlet boundary condition.

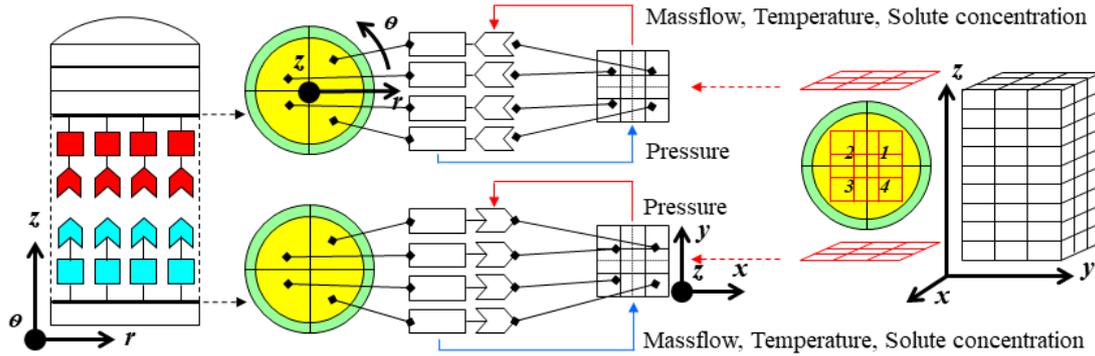


Figure 4-7 – Data transfer mechanism between TRACE and SCF within the domain decomposition approach.

Two algorithms for the data transfer at the interface-components are developed to enhance the flexibility and simplicity of the modeling process:

- 1) The new BREAK- and FILL-components can be attached to the VESSEL-component at different locations to simulate different flow scenarios,
- 2) TRACE is able to distinguish automatically between normal FILL- and BREAK-components and the ones used as interface-components for data exchange. This assures a highly automated coupling of the two codes.

In Figure 4-8, the flexibility of defining the “interface-components” in the TRACE-model is illustrated by three different cases. Case 1 and 2 show the normal conditions when all TRACE sectors or meshes in the core inlet and outlet are connected with the interface-components. Case 3 exhibits special conditions, where only some of the TRACE-sectors (or meshes) have interface-components. The arrangement of such interface-components is quite flexible. TRACE can handle various such configurations in an automatic way.

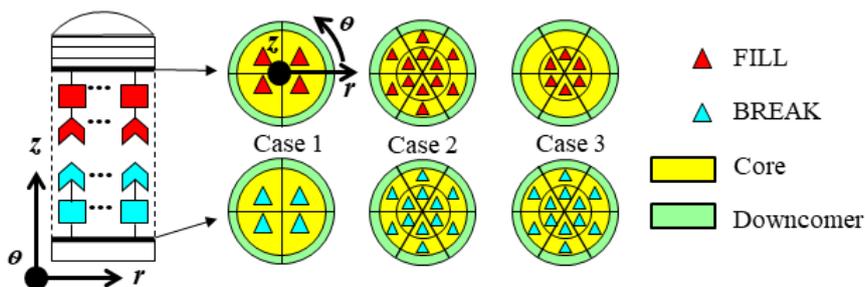


Figure 4-8 – Flexible configuration of the interface-components into original TRACE-model.

In addition, the interface-components can be mixed together with the normal components. If you take the TRACE-model shown in Figure 4-5 as an instance, there are in total eight FILLs and BREAKs and sixteen PIPES including four normal FILLs, four normal BREAKs, eight normal PIPES and also four interface FILLs, four interface BREAKs, and eight interface PIPES. The users could specify those interfaces in the TRACE source code. If this is the case, modifications of the source code have to be performed every time when a new model is developed. This approach is inadequate. Hence, a simple and direct method has been developed to enable TRACE to recognize the interface-components by just adding a new input parameter in the TRACE input file. In Figure 4-9, the mechanism of the auto-recognition of interface-components by TRACE is illustrated.

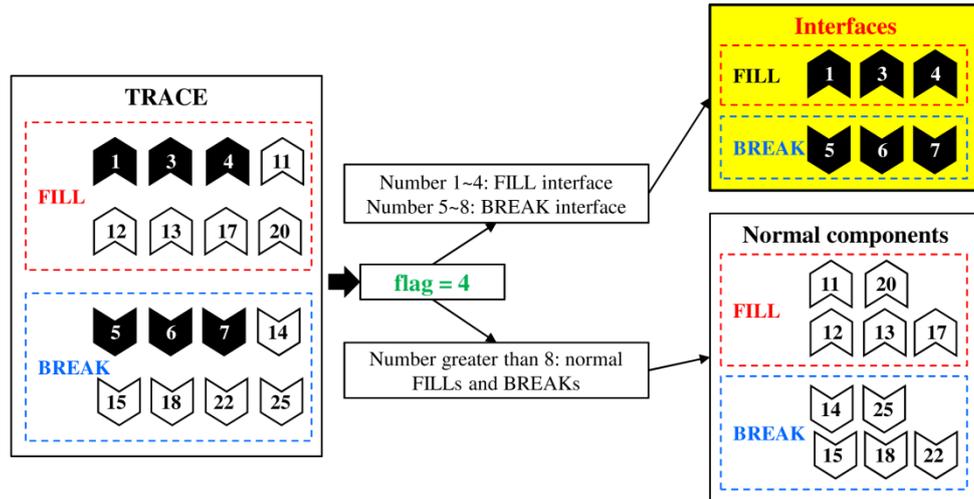


Figure 4-9 – Mechanism of the auto-recognition of interface-components implemented in TRACE.

Assuming that there are eight FILLs and BREAKs in the TRACE-model. If the new input parameter is set to 4, all the FILLs with the component number between 1 to 4 and all the BREAKs with the component numbers between 5 to 8 are recognized as interfaces while all others are treated as normal components. The PIPES connecting the interface FILLs or BREAKs can be also recognized automatically as data transfer interfaces.

#### 4.2.2 The method for field mapping between two thermal-hydraulic domains

Looking at Figure 4-7 more carefully, a mesh mismatch between the interfaces of TRACE and SCF is observed. Hence, in Figure 4-10, a detailed mapping of the TRACE-mesh with the SCF-mesh is shown. Both meshes differ from each other. In the TRACE-model, four sectors of a cylindrical 3D VESSEL represent the core plane while the SCF-model consists of nine regular square meshes. However, in real cases, the TRACE-model can have much more sectors and more rings to represent the core while the SCF-model may consist of hundreds of square meshes.

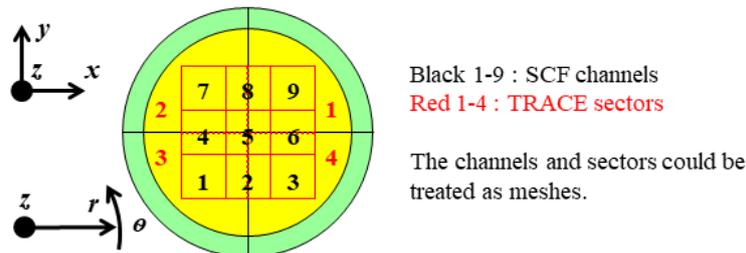


Figure 4-10 – Different TRACE and SCF-meshes valid for the models shown in Figure 4-7.

Figure 4-11 shows a typical SCF-model consisting of 157 meshes and the TRACE-model of 16 sectors. In order to develop a radial mapping between the two different meshes of the involved codes, care must be taken. For example, the red marked mesh of SCF will get data from four TRACE- sectors, namely from sector 2, 3, 10 and 11, according to Figure 4-11 right. The thermal-hydraulic parameters predicted by SCF for this mesh need to be passed also to four TRACE- sectors located at two different rings and at two different azimuthal sectors. In real applications, SCF-models may have irregular

meshes (areas, shapes, and arrangement) to be mapped to the TRACE-mesh. Consequently, an efficient mapping approach has to be developed for the implementation of the mesh or field mapping.

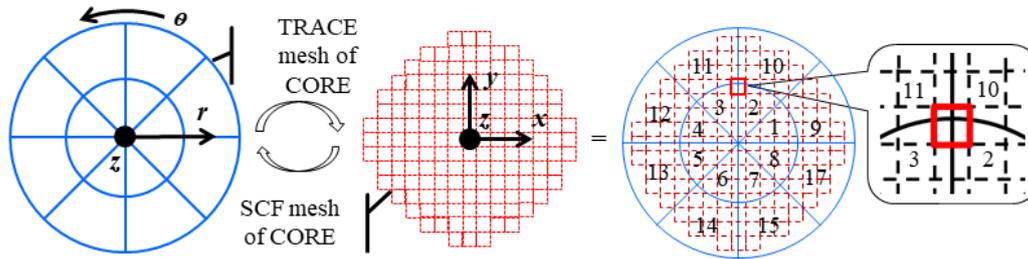


Figure 4-11 – Mapping of the TRACE-mesh with the SCF-mesh for a typical PWR.

In this dissertation, a new toolkit has been developed to automatically compute the mesh mapping relationship between TRACE and SCF. In this approach, all SCF-channels are simplified to circles. The relative relationship of the position of the TRACE-sectors and the SCF-mesh (circles) is inspected carefully and finally, 12 main categories are figured out. Moreover, 46 sub-categories are specified covering all possibilities for the radial mapping between TRACE and SCF. More details of the categories are given in Appendix A.

Hereafter, the major features of this toolkit are highlighted: the contribution ratios from each SCF-mesh to each TRACE-sector is automatically calculated based on their overlapping area and vice versa, regardless of the shape of the SCF-mesh, of the partition of the TRACE-sector and of the core mesh layout. The predicted values of the contribution ratio are stored in two specially designed arrays for further data mapping process. In Figure 4-12, the two arrays, which correspond to Figure 4-10, are given. The array on the left side represents the contribution ratio from each SCF-mesh to TRACE-sector. Note that the sum of each row equals 1. The array on the right side represents the contribution ratio from each TRACE-sector to SCF-mesh. Note that the sum of each column equals 1. The more complicated the meshes are, the larger is the size of the arrays. The mechanism of this toolkit is exactly an overlapping-area-weighted algorithm, which ensures an accurate and user-friendly mapping process. This approach is very similar to the approach used in TRACE to map the thermal-hydraulics cells to the neutronic nodes (Xu & Seker, 2010). But, it is a totally automatic way and no additional information is required.

		SCF channels								
		1	2	3	4	5	6	7	8	9
TRACE sectors	1	0	0	0	0	1/9	2/9	0	2/9	4/9
	2	0	0	0	2/9	1/9	0	4/9	2/9	0
	3	4/9	2/9	0	2/9	1/9	0	0	0	0
	4	0	2/9	4/9	0	1/9	2/9	0	0	0

		SCF channels								
		1	2	3	4	5	6	7	8	9
TRACE sectors	1	0	0	0	0	1/4	1/2	0	1/2	1
	2	0	0	0	1/2	1/4	0	1	1/2	0
	3	1	1/2	0	1/2	1/4	0	0	0	0
	4	0	1/2	1	0	1/4	1/2	0	0	0

Figure 4-12 – The arrays storing the contribution ratio data between TRACE and SCF corresponds to Figure 4-10.

### 4.2.3 The explicit temporal coupling

An explicit temporal coupling approach is developed for TRACE and SCF for both steady-state and transient simulations. In this approach, the TRACE steady-state option is activated, where TRACE checks for the rate of change of various thermal-hydraulic parameters to decide about the achievement of a converged solution. In the case of SCF, the steady-state mode is also activated. As mentioned, the

SCF transient simulation mode always starts with a steady-state calculation. Figure 4-13 shows the flowchart of each code and the points for data transfer between them as implemented.

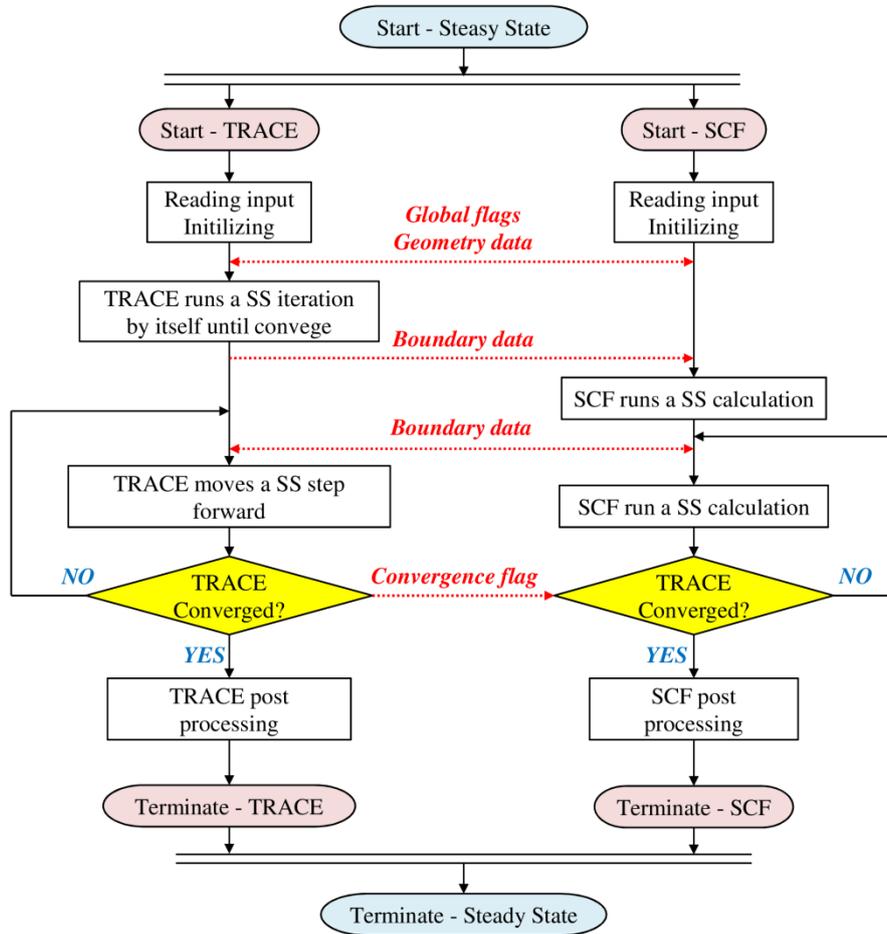
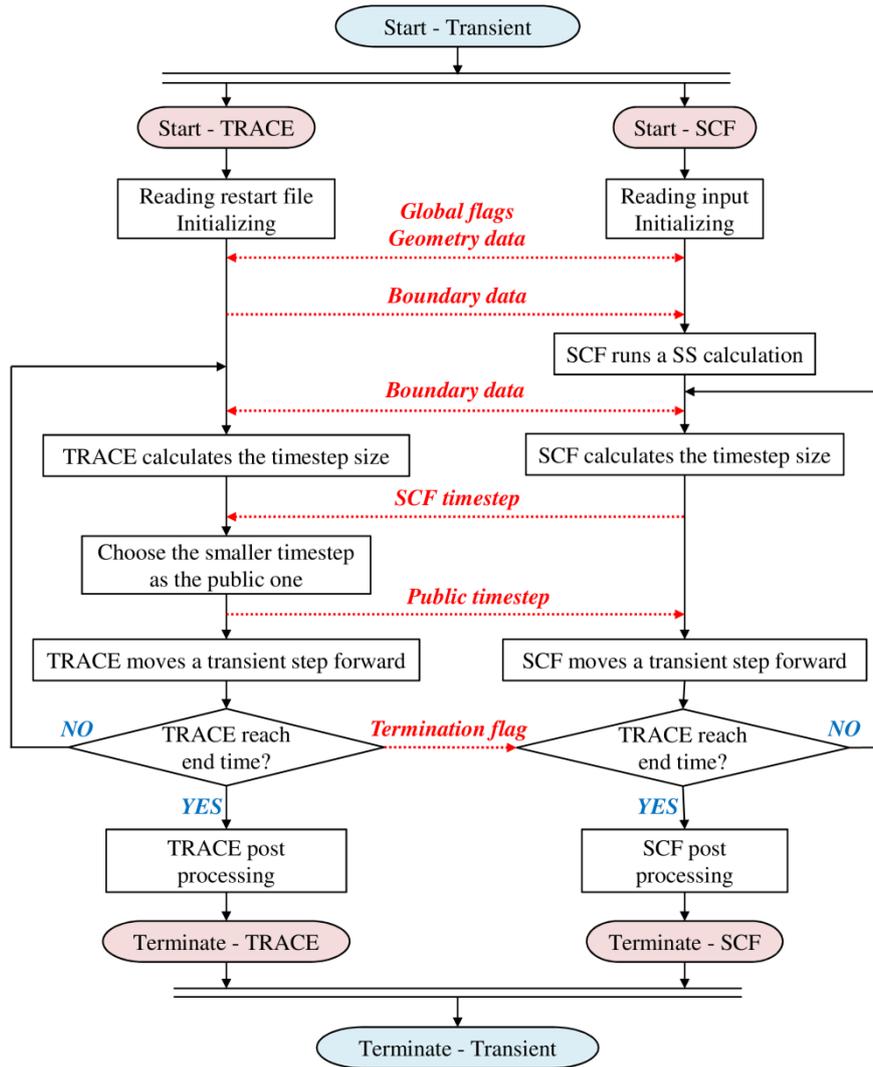


Figure 4-13 – Steady-State (SS) calculation scheme of TRACE/SCF.

Since the parallel coupling of both codes shown in Figure 4-13 is based on ECI, the communication channels are pre-established once the codes are launched. Then, the two executables read their inputs and do the initialization. After that, some global and geometry data is exchanged. Next, TRACE runs from its initial condition a steady-state standalone simulation. Since the calculation of this stage is meaningless, no data exchange is carried out with SCF. Once TRACE converges, the boundary condition data are extracted and submitted to SCF. Then, SCF runs a stand-alone steady-state simulation. Next, TRACE and SCF enter into the iteration loop. Within a step, the boundary condition data from the two codes is firstly exchanged in both directions. The codes then run their own steady-state simulations synchronously. After that, TRACE checks its convergence and passes the flag to SCF. If TRACE converged, both of the codes perform the post-processing and then the simulations terminate. If not, the codes will continue the iteration. In this scheme, TRACE is in control of SCF. The TRACE convergence flag determines whether SCF terminates or continues.

Figure 4-14 displays the flowchart of the transient procedure of the coupled TRACE/ SCF system based on ECI. TRACE uses the restart data obtained after a previous steady-state calculation as the initial condition to run a transient restart case. Furthermore, TRACE can directly start with a transient calculation as well. For a transient run, the criterion for the termination of the simulation is the problem time rather than the fulfillment of the convergence criteria of thermal-hydraulic parameters. In both codes, the transient simulation mode is activated.



**Figure 4-14 – Transient calculation scheme of TRACE/SCF.**

The communication channels between both codes are established once the codes are launched. Then, TRACE read the restart and input files and SCF read the input file, respectively. After the initialization, some global and geometry data are exchanged. Next, the boundary data is extracted from TRACE and submitted to SCF as boundary conditions. Then, SCF runs a standalone steady-state simulation. Next, TRACE together with SCF enters the time advancement loop. Within a step, the boundary data from the two codes is firstly exchanged bi-directionally. The codes then calculate their own time step sizes. The step size of SCF has to be transferred and to be compared with the one of TRACE. The smaller one is selected as the public time step size. Then it is transferred back to SCF. Next, the two codes will advance in time according to the time step in a synchronized manner. After that, TRACE will check whether it reaches the end of the problem time. For this case, the flag is passed back to SCF. Finally, both codes perform the post-processing and then the simulations terminate. If not, the codes continue within the time advancement loop. Here is this scheme, TRACE is in control of SCF. The TRACE termination flag determines whether SCF should stop or continue.

## 4.3 Verification of TRACE/SCF-ECI with an academic coolant mixing and boron dilution case

### 4.3.1 Description of the problems

The coolant mixing problem is to simulate the reactor pressure vessel and the core behavior of a PWR four-loop plant. This TRACE model and SCF model are shown in Figure 4-3. The diagram of the configuration of the 4 TRACE-loops (with hot-legs and cold-legs) and the 9 SCF-channels is depicted in Figure 4-15. SCF is supposed to be able to predict a more realistic and accurate thermal-hydraulic condition by considering the lateral transport between neighboring thermal-hydraulic channels compared to the system code TRACE. TRACE represents an RPV of a PWR with four inlet and outlet boundary conditions for the flow and temperature of the primary circuit, where the core is represented by four sectors (3D volumes). The mixing pattern predicted by TRACE for only four azimuthal sectors of the downcomer propagates to the core. In SCF, the core is radially represented by as many thermal-hydraulic meshes as fuel assemblies. Moreover, the lateral transport among all thermal-hydraulic meshes at each axial layer of the core is predicted by SCF using either the equal mass or equal volume model.

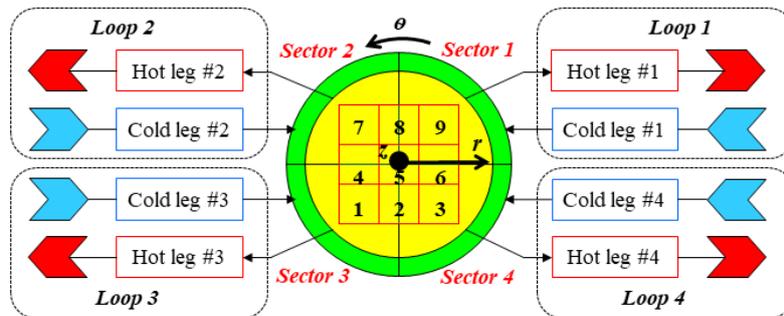


Figure 4-15 – The configuration of the four TRACE-loops and nine SCF-channels of TRACE/SCF for the academic testing case.

During the transient, the coolant temperature of cold leg 1 increases from 400 K to 500 K within 10 seconds and remains at that temperature for 20 seconds. The coolant temperature of all other cold legs is constant at 400 K, see Table 4-1.

The solute boron concentration is 0.01 expressed in a mass ratio (the solute mass to water mass ratio, around 9900 ppm). The transient lasts for fifty seconds and the coolant inlet velocity is fixed to 10m/s for all four loops. Under such conditions, a coolant mixing will take place in the downcomer and the resulting mixing pattern is propagated through the reactor core. Two simulations, one with TRACE and another one with TRACE/SCF, are performed to analyze this test problem.

Table 4-1 – Boundary conditions for the coolant mixing problem of the TRACE/SCF academic testing case.

	<i>Loop #1 Inlet</i>			<i>Loop #2 #3 #4 Inlet</i>
<i>Time (t)</i>	0s ~ 20s	20s ~ 30s	30s ~ 50s	0s ~ 50s
<i>Temperature</i>	400K	$400 + (t-20)*10K$	500K	400K
<i>Velocity</i>	10m/s			10m/s
<i>Solute mass ratio</i>	0.01			0.01

Regarding the boron dilution problem, the goal is to investigate the thermal-hydraulic behavior of the reactor pressure vessel and the core, when the boron concentration of the loop-1 is changed linearly from 0.01 to 0.05, as listed in Table 4-2. All other parameters correspond to those of the coolant mixing problem.

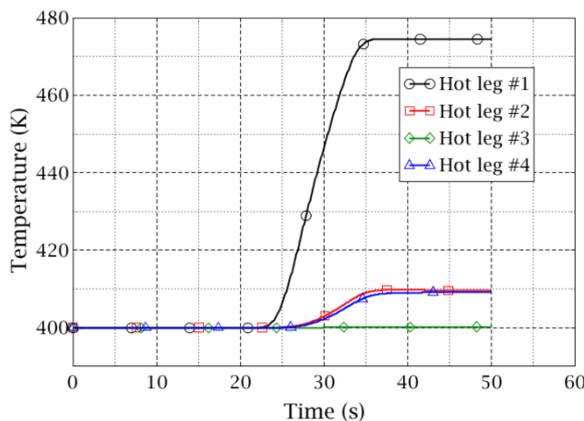
**Table 4-2 – Boundary conditions for the boron dilution problem of the TRACE/SCF academic testing case.**

<i>Time (t)</i>	<i>Loop #1 Inlet</i>			<i>Loop #2 #3 #4 Inlet</i>
	0s ~ 20s	20s ~ 30s	30s ~ 50s	0s ~ 50s
<i>Temperature</i>	400K			400K
<i>Velocity</i>	10m/s			10m/s
<i>Solute mass ratio</i>	0.01	$0.01 + (t-20)*0.004$	0.05	0.01

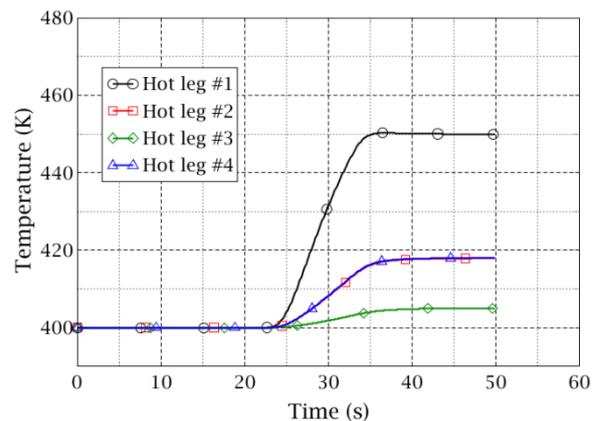
### 4.3.2 Performed simulations and discussion of results

#### a) Coolant mixing problem

The coolant temperatures of the hot legs predicted by TRACE standalone and by TRACE/SCF are shown for the duration of the transient in Figure 4-16 and Figure 4-17. There, it can be observed, that TRACE predicts a lower temperature increase of in the loops 2 and 4 of around 9 to 10 K than the one predicted by TRACE/SCF which amounts around 18 K. As a result of it, the heat up of hot leg #1 predicted by TRACE is also higher than the one predicted by the coupled code (74 K > 50 K). The main reason for it is the fact that TRACE/SCF predicts a stronger lateral coolant mixing along with the core height between the hot meshes and the neighbor meshes linked to the sector 2 and 4 than TRACE does. One of the reasons is that the lateral flow model in SCF behaves better than that of TRACE. The other reason is that the SCF-model is more detailed (12 mesh boundaries for cross-flow) compared to the very coarse four sector model of TRACE. The temperature increases at loop 2 and 4 are more or less the same because they are symmetrically located with respect to loop 1. Moreover, the temperature at loop 3 was not that much affected because it has no direct mass exchange or heat transfer interfaces with the mesh connected to loop 1. The temperature rise in loop 3 comes from loop 2 and 4 circumferentially because of the heat and mass transfer imposed by the lateral flow.

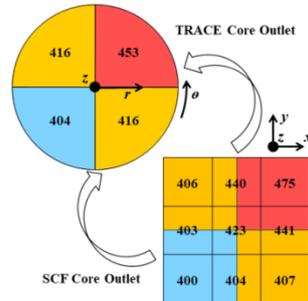


**Figure 4-16 – Computed temporal evolution of the different hot-leg temperatures in an academic coolant mixing problem by TRACE standalone. Transient starts  $t=0s$  and conditions are given in Table 4-1.**



**Figure 4-17 – Computed temporal evolution of the different hot-leg temperatures in an academic coolant mixing problem by TRACE/SCF. Transient starts  $t=0s$  and conditions are given in Table 4-1.**

Figure 4-18 displays the coolant temperature distributions at the core outlet of TRACE and SCF within the coupling system, from which, it can be stated that the TRACE data perfectly reflects the overlapping-area-weighted algorithm discussed in section 4.2.2 and presented in Figure 4-12. The slight temperature difference between the TRACE core outlet (Figure 4-18) and the hot legs (Figure 4-17) indicate some heat transfer in the upper plenum.

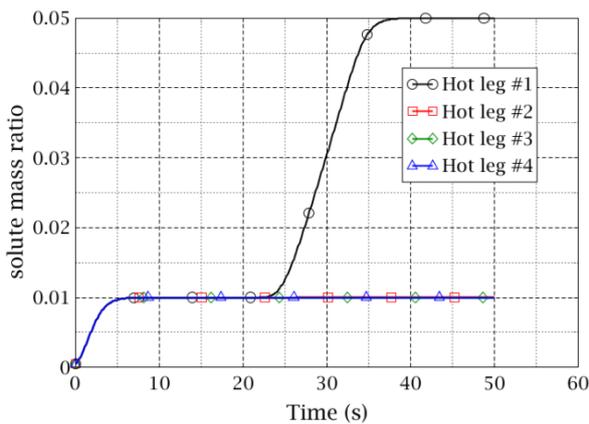


**Figure 4-18 – Temperature distributions at the core outlet by TRACE/SCF show correct data mapping between the two different meshes.**

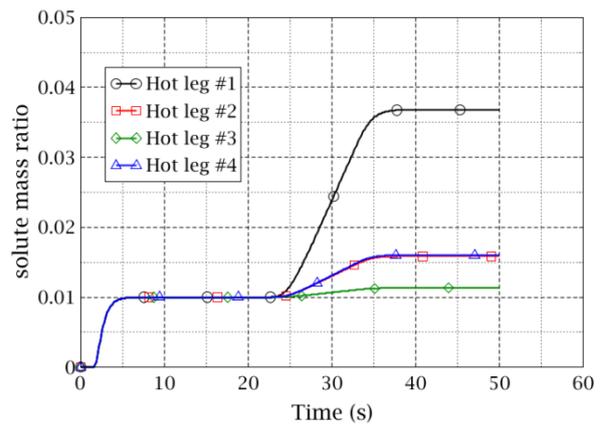
The promising results not only prove that the data exchange between the coupled codes works correctly i.e. it is transferred to the right place at the right time, but it also shows that the two algorithms presented in section 4.2 are consistently implemented. In this case, four additional FILLS and BREAKs are appended to the VESSEL component as interfaces and automatically distinguished from the normal components by the programmed algorithms.

**b) Boron dilution problem**

This problem is solved by TRACE and TRACE/SCF. In Figure 4-19, the temporal evolution of the boron concentration by TRACE standalone at the hot legs is plotted. Figure 4-20 shows the solute mass ratio at the four hot legs by the coupled code TRACE/SCF. Here TRACE predicts a higher value of the solute mass ratio, of leg-1 compared to the value predicted by the coupled code TRACE/SCF for the same hot leg, Figure 4-20. Due to the stronger lateral mixing along the core height predicted by SCF, an increase of the solute mass ratio in the hot legs 2 and 3 are predicted only by TRACE/SCF.



**Figure 4-19 – Computed temporal evolution of the different hot-leg solute mass ratio in an academic coolant mixing problem by TRACE standalone. Transient starts t=0s and conditions are given in Table 4-2.**



**Figure 4-20 – Computed temporal evolution of the different hot-leg solute mass ratio in an academic coolant mixing problem by TRACE/SCF. Transient starts t=0s and conditions are given in Table 4-2.**

## 4.4 Validation of TRACE/SCF-ECI using the VVER-1000 coolant mixing benchmark

The validation of the new TRACE/SCF capability is done by the analysis of the VVER-1000 coolant mixing benchmark problem. Selected parameters predicted by the coupled code are compared with TRACE standalone and measured data at the Kozloduy nuclear power plant.

### 4.4.1 Description of the VVER-1000 coolant mixing experiment

The coolant mixing experiment was performed during the commissioning phase of Kozloduy NPP to study the mixing of loop flows in the reactor vessel of VVER-1000 V320 (Kolev, et al., 2010). In the test, the steam isolation valve was closed and the steam generator was isolated resulting in the heat-up of primary coolant in the sector of the downcomer linked to the affected steam generator (first loop). Due to the coolant mixing in the downcomer, the temperature of the other primary loop is partly also affected. The mixing pattern established in the downcomer propagates to the core where also coolant mixing takes place. VVER-1000 is a four-loop pressurized water reactor with hexagonal core geometry and horizontal steam generators. The core is open-type and contains 163 hexagonal fuel assemblies. The location of the main inlet and outlet nozzles of the reactor vessel is non-uniform in the azimuthal direction and asymmetric with respect to the core symmetry axes. The cross-section sketch of the reactor vessel is described in Figure 4-21.

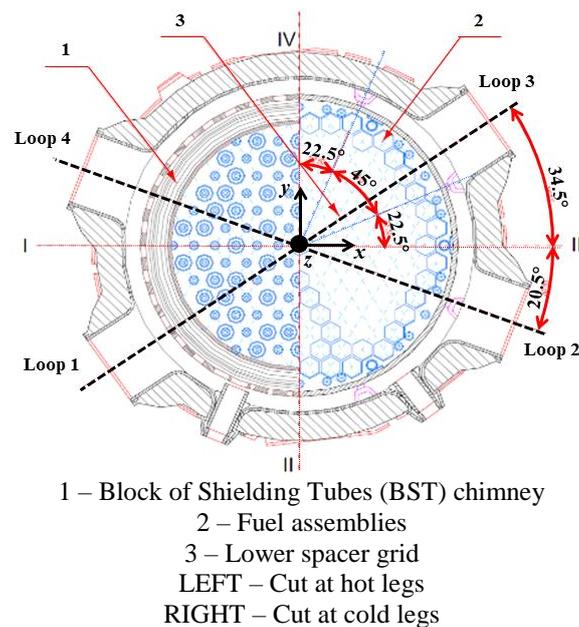


Figure 4-21 – Vessel cross-sectional sketch of VVER-1000 including the vessel inlet and outlet.

The main initial operating parameters before the test are summarized in Table 4-3.

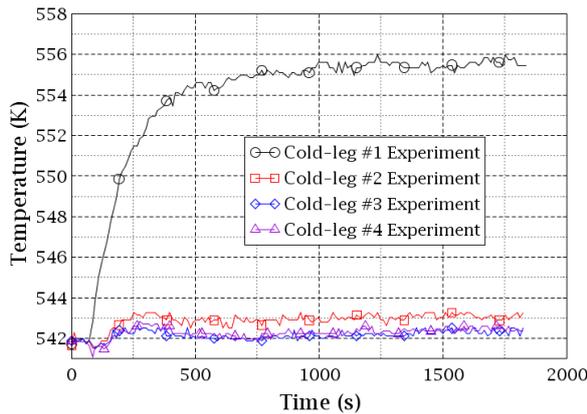
The transient test consists of the following events:

- 1) Closure of the steam isolation valve and isolation of the steam generator from feed water of loop 1;
- 2) Coolant temperature at the cold leg of loop 1 increases by about 14 degrees, Figure 4-22;
- 3) Coolant mixing occurs first in the downcomer;
- 4) Coolant mixing takes place in the lower plenum, core and upper plenum;

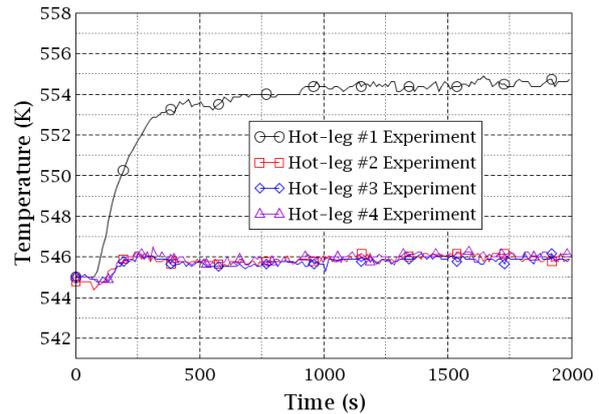
Coolant temperature at the other three loops all increase by different degrees, see Figure 4-22 and Figure 4-23

**Table 4-3 – Main Initial Operating Parameters before Test**

<i>Parameter</i>	<i>Initial State</i>	<i>Accuracy</i>
Thermal power, MW	281	± 60
Pressure above the core, MPa	15.593	± 0.3
Pressure drop over RPV, MPa	0.418	± 0.043
Coolant temperature at core inlet #1, K	541.75	± 1.5
Coolant temperature at core inlet #2, K	541.85	± 1.5
Coolant temperature at core inlet #3, K	541.75	± 1.5
Coolant temperature at core inlet #4, K	541.75	± 1.5
Coolant temperature at core outlet #1, K	545	± 2.0
Coolant temperature at core outlet #2, K	545	± 2.0
Coolant temperature at core outlet #3, K	544.9	± 2.0
Coolant temperature at core outlet #4, K	545	± 2.0
Mass flow rate of loop #1, kg/s	4737	± 110
Mass flow rate of loop #2, kg/s	4718	± 110
Mass flow rate of loop #3, kg/s	4682	± 110
Mass flow rate of loop #4, kg/s	4834	± 110



**Figure 4-22 – Measured evolution of the coolant temperature at cold legs during the transient of the VVER-1000 coolant mixing benchmark.**



**Figure 4-23 – Measured evolution of the coolant temperature at hot legs during the transient of the VVER-1000 coolant mixing benchmark.**

The test lasts for 1800 seconds and the final core power increased up to 286 MW. Due to the isolation of the steam generator 1, the heat transfer from the primary to the secondary side decreases leading to the increase of the coolant temperature on the cold-leg-1 and finally on the hot-leg-1. The coolant temperature of the other three loops also increases due to the coolant mixing in the downcomer and lower plenum. Reverse heat transfer occurs in the isolated steam generator #1. At about 300s, the temperature of cold-leg-1 exceeds that of hot-leg-1, see Figure 4-22 and Figure 4-23. The difference stabilize to 0.6~0.8 degrees in about 20 min. Cold-leg-2 has a slightly higher coolant temperature than cold-leg-3 and 4 during the transient, while the outflows on the four hot-legs almost have the same temperature, indicating the coolant well mix.

## 4.4.2 Description of the thermal-hydraulic models of the RPV and core

TRACE/SCF employs a domain decomposition approach in which SCF simulates the reactor core while TRACE simulates the rest of the vessel. For the simulation of TRACE/SCF, modifications e.g. blockage of the core and addition of FILL and BREAK components at the core inlet and outlet planes which represent the interface planes of the coupled code should be performed to the original TRACE model, Figure 4-24. The SCF model has 163 hexagonal cells demonstrating the fuel assemblies and a hexagonal core geometry, Figure 4-25. No further modification to the original model is required.

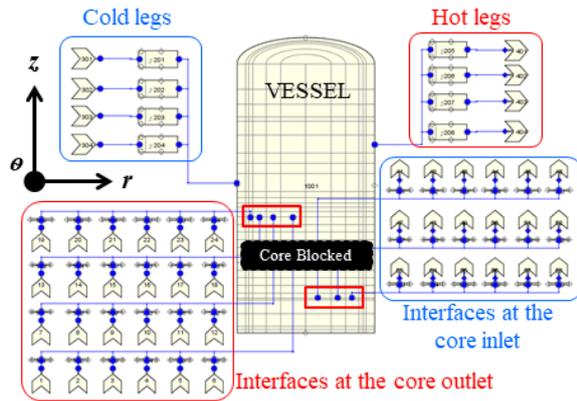


Figure 4-24 – TRACE model for the VVER-1000 coolant mixing benchmark.

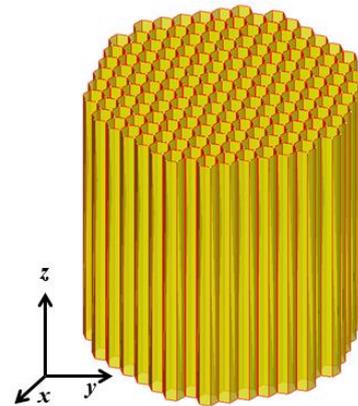


Figure 4-25 – SCF model for the VVER-1000 coolant mixing benchmark.

In order to map the mesh data of both codes in a consistent manner, an overlapping-area-weighted algorithm and a toolkit have been used (see Appendix A). The data mapping is illustrated in Figure 4-26, where TRACE extracts the pressure at its core outlet and transfer it to SCF after a data mapping process. Meanwhile, it extracts the mass flow rate and coolant temperature at the core inlet and transfers to SCF. The data from SCF to TRACE occurs in the opposite way.

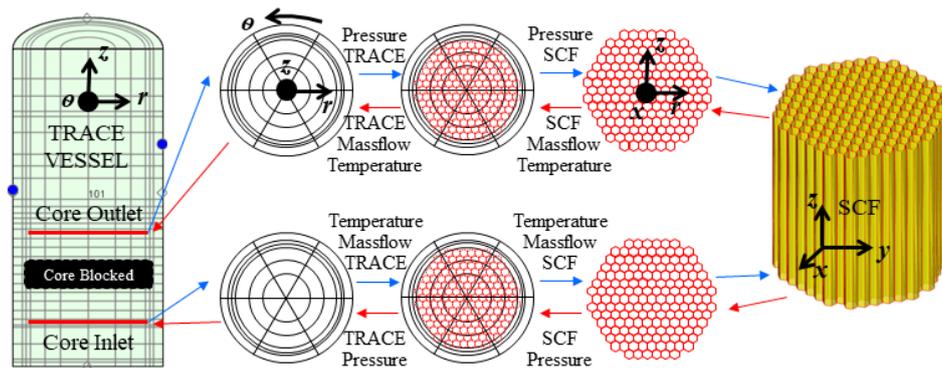


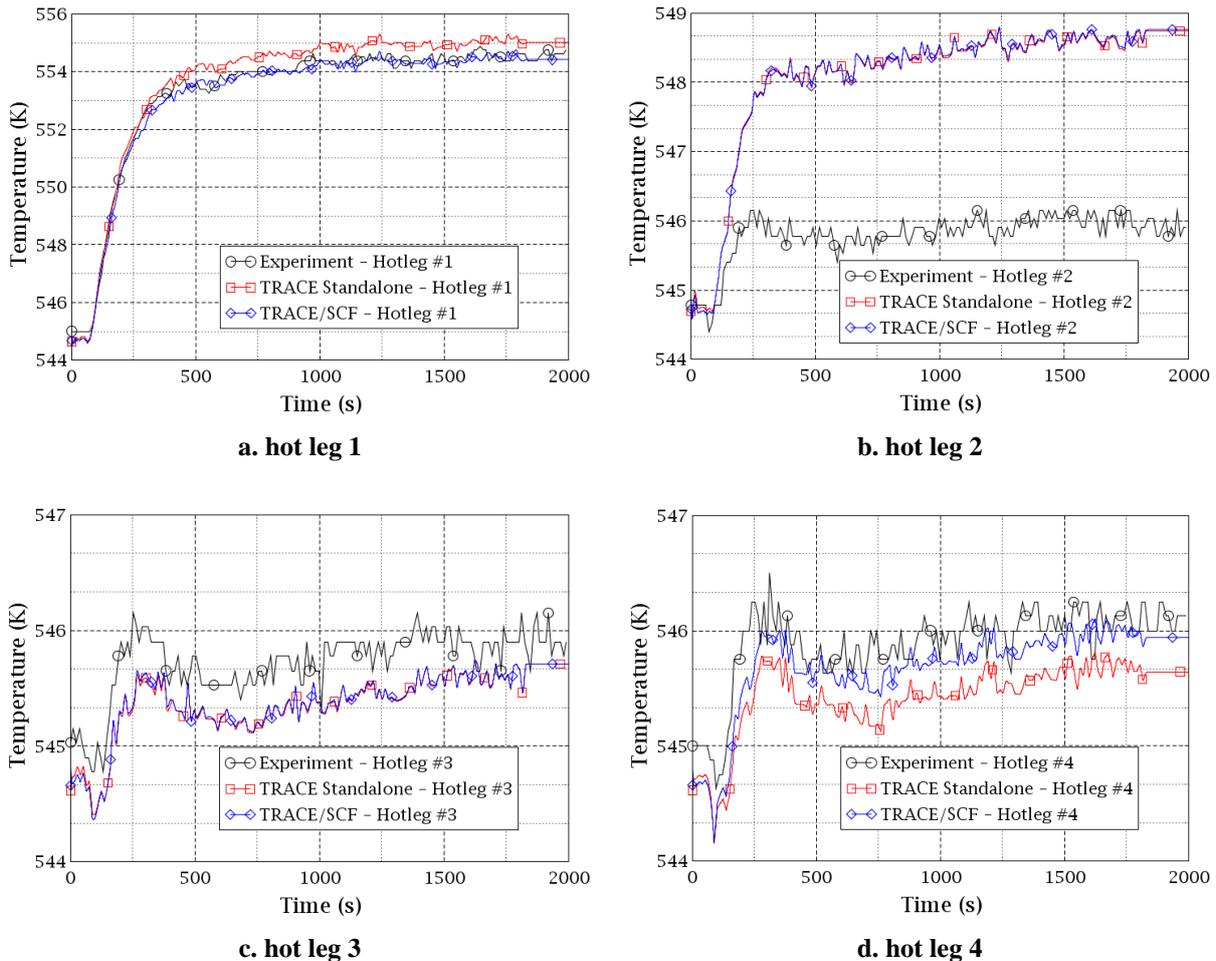
Figure 4-26 – Data Mapping between TRACE and SCF for the VVER-1000 Coolant Mixing Benchmark

## 4.4.3 Discussion of selected results

The coolant mixing test is analyzed with both TRACE standalone and the coupled TRACE/SCF. The codes predict the coolant mixing in the downcomer, the lower plenum, and the propagation of this mixing pattern through the core as well as the mixing in the upper plenum. The time-dependent

boundary conditions e.g. coolant temperature at the cold legs and outlet pressure at the hot legs are taken directly from the benchmark specifications.

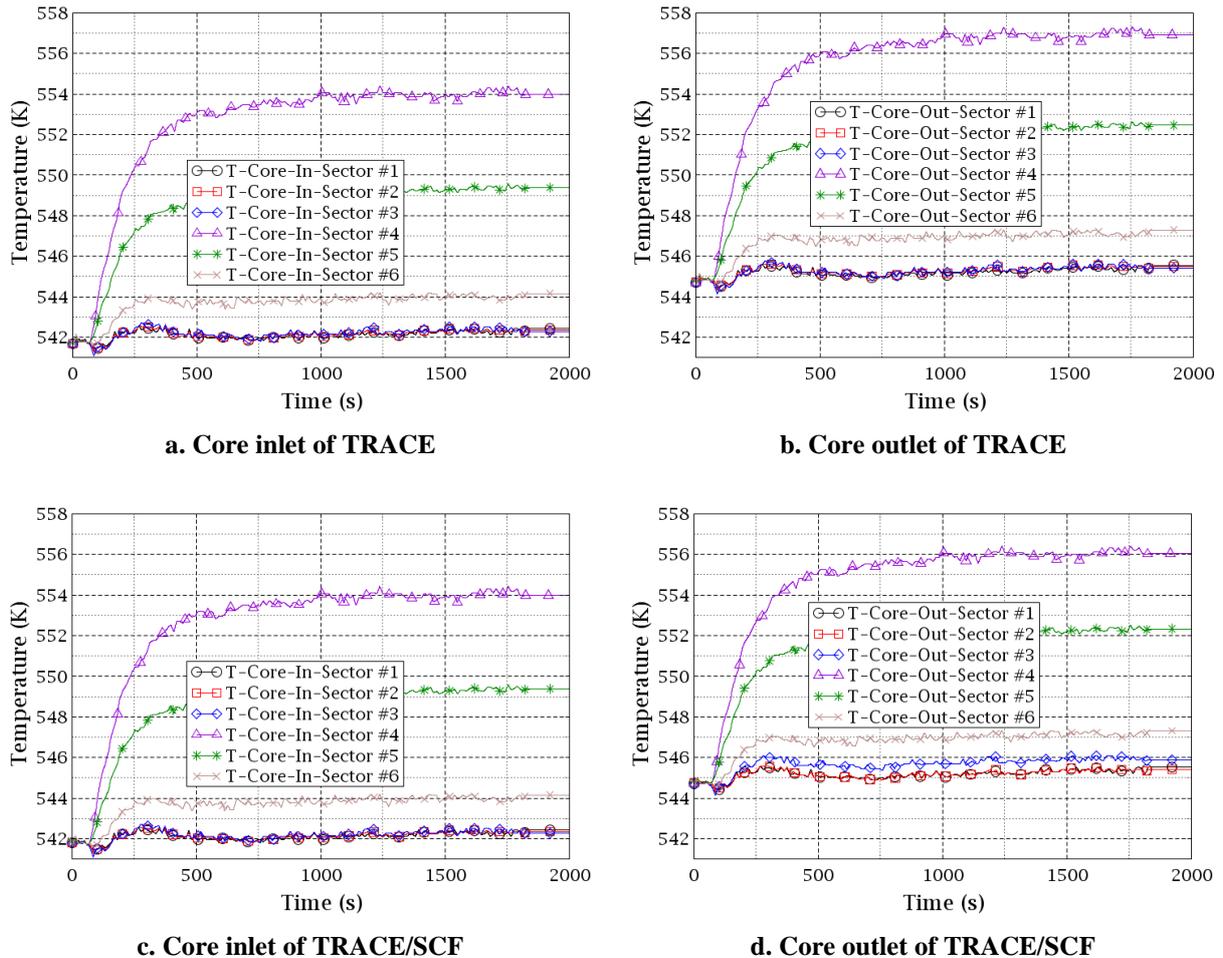
The coolant temperature at the four hot legs predicted by TRACE and TRACE/SCF is compared to the measured data in Figure 4-27. The temperature rise of hot-leg-1 can be observed in Figure 4-27a, where TRACE standalone tends to over-predict the heat-up while the coolant temperature calculated by TRACE/SCF is very close to the plant data. In Figure 4-27b and Figure 4-27c, the predictions of TRACE and TRACE/SCF are all not satisfactory. They over-predict the heat-up at hot-leg-2 while the coolant temperatures at hot-leg-3 are under-predicted. In Figure 4-27d, TRACE standalone under-predicts the heat-up on hot-leg-4 while TRACE/SCF gives closer result to the measured data.



**Figure 4-27 – Temporal evolution of the measured and computed (by TRACE standalone and TRACE/SCF) coolant temperature at the hot legs of the VVER-1000 coolant mixing benchmark, which show TRACE/SCF could produce a closer result to the measured data than that by TRACE standalone.**

Before digging any further into the simulated data, it is worth to note that the cold and hot legs in the RPV model of TRACE are located at sector #4, #6, #1, and #3, respectively, see Figure 4-28. In addition, the loops are not arranged symmetrically and there is an anti-clockwise angle shift with respect to the core symmetry axes or the loops in TRACE model, also see Figure 4-28. TRACE could hardly represent this kind of irregular geometry unless the RPV model is split into a larger number of sectors in the azimuthal direction. But this will lead to the poor numerical performance of any code with coarse 3D thermal-hydraulic cells. Here, CFD-codes are more suitable to treat such kind of non-symmetrical problems than system codes.

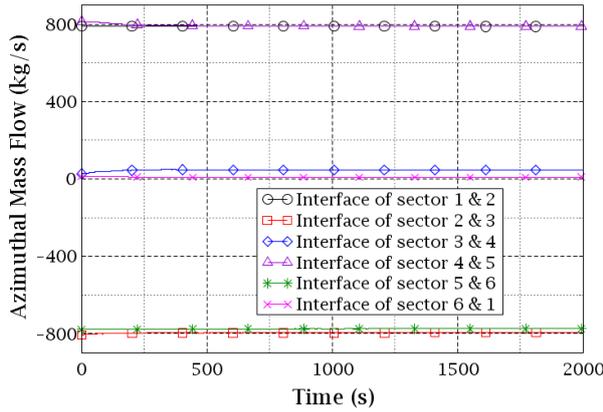




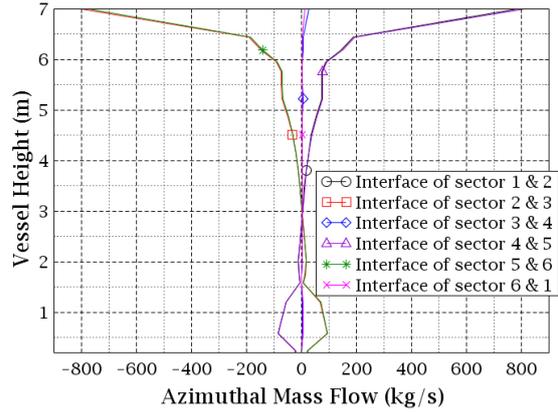
**Figure 4-29 – Temporal evolution of the computed (by TRACE and TRACE/SCF) coolant temperature at the core inlet and outlet (on TRACE mesh) of the VVER-1000 coolant mixing benchmark, which shows TRACE/SCF could predict stronger coolant mixing between sector 4 and 3 than that by TRACE.**

Figure 4-30 shows the azimuthal mass flowrate curves at the downcomer inlet, where the absolute mass flowrate at the interfaces between sector 1 and 2 (face 1&2), face 2&3, face 4&5, face 5&6 are as large as 800 kg/s while that at face 3&4 and face 6&1 are quite small. The positive direction of the flow at each face is right the same with the normal vector on the anti-clockwise direction (refer to the coordination shown in Figure 4-28). A positive value means an anti-clockwise flow while a negative value means a clockwise flow. The data indicate that the coolant in sector 2 comes mainly from sector 1 and 3 and the coolant in sector 5 comes mainly from sector 4 and 6. The curves also illustrate that the coolant mixing effect between sector 4 (first loop) and sector 3 (fourth loop), between sector 6 (loop 2) and sector (loop 3) is relatively weak.

Figure 4-31 plots the mass flowrate through the six azimuthal interfaces along the downcomer in the reactor pressure vessel. The mass flowrate from sector 4 (loop 1) and sector 6 (loop 2) to sector 5 decrease from 800 kg/s at the downcomer inlet to 0 kg/s at the RPV height of around 3 m. After that, the flow reverses from sector 5 to sector 4 and sector 6. The conditions are the same for sector 1, 2 and 3. Note the mass flow rate through face 3&4 and face 6&1 keeps close to 0 kg/s along the downcomer. Figure 4-31 tells the fact that there are two somehow isolated mixing regions within the downcomer where sector 1 to 3 form a mixing body while sector 4 to 6 form the other one. It also shed light on the remarkable coolant temperature rise at hot-leg-2 compared that at hot-leg-3 and 4.

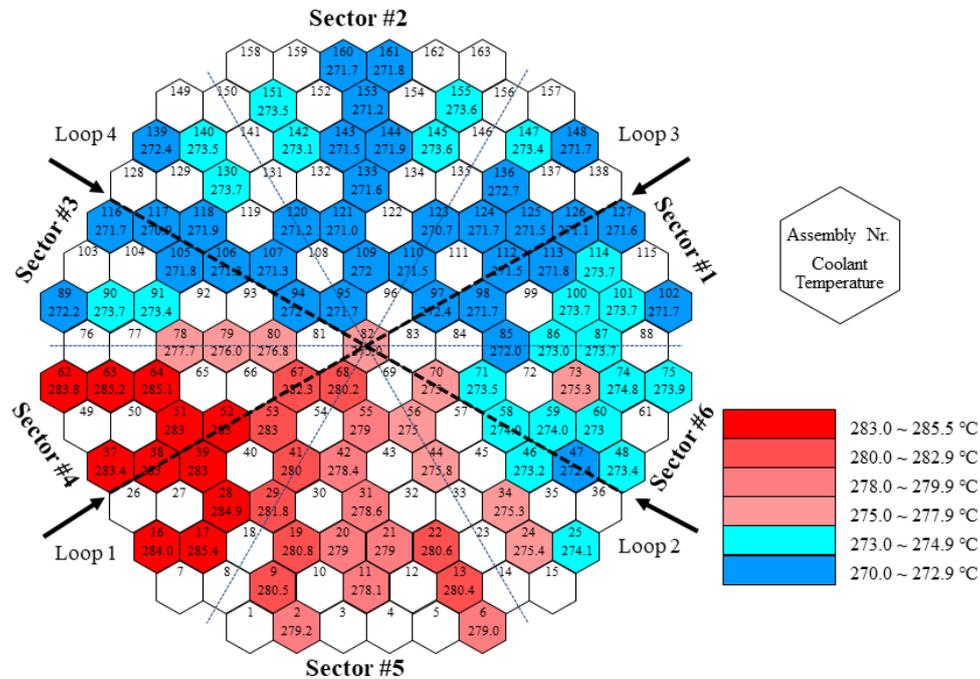


**Figure 4-30 – The computed azimuthal mass flowrate over the six azimuthal faces at the downcomer inlet for the VVER-1000 coolant mixing benchmark.**

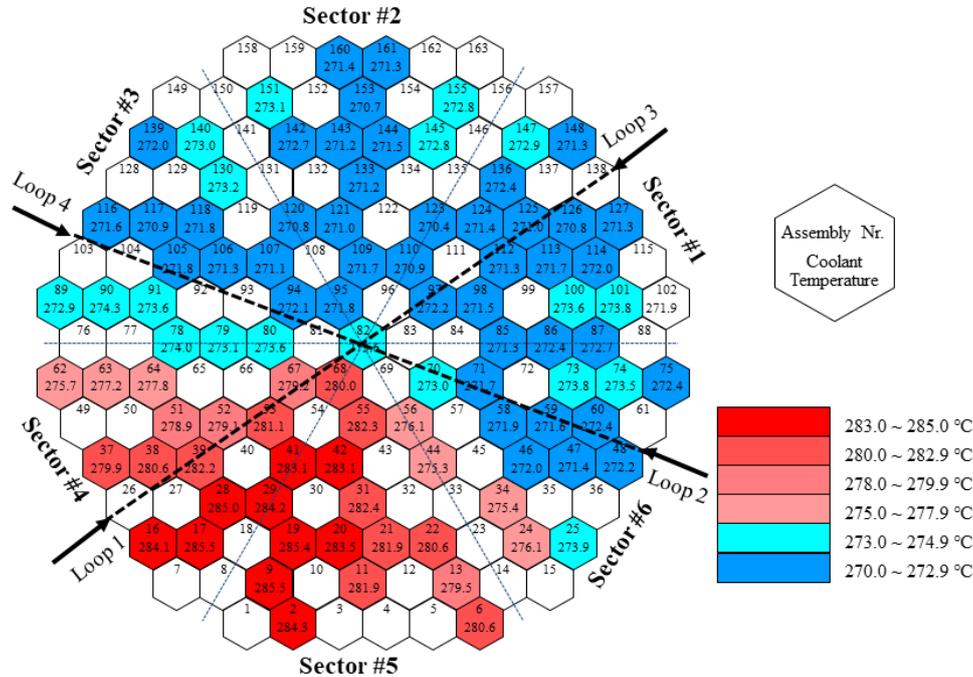


**Figure 4-31 – The computed azimuthal mass flowrate over the six azimuthal faces along the downcomer for the VVER-1000 coolant mixing benchmark.**

Figure 4-32 and Figure 4-33 present the coolant temperature distribution at the core outlet assembly-wise predicted TRACE/SCF i.e. by SCF and measured in the experiment respectively. There, one can observe that the location of the four loops is shifted anti-clockwise compared to their positions in the TRACE model. A significant temperature change can be seen between sector 3 and sector 4 (refer to the model configuration in Figure 4-28) in Figure 4-32 confirming a relatively weak coolant mixing between the two sectors. Moreover, the rapid temperature change could be extended to include the entire horizontal line above which the coolant is cooler (sector 1, 2, and 3) while below which the coolant is hotter (sector 4, 5, and 6). In the experimental data shown in Figure 4-33, a more smooth temperature transition area is seen between sector 3 and 4. Furthermore, the measured temperature is more symmetrically distributed than the ones of the simulations.



**Figure 4-32 – Coolant temperature distribution at the core outlet by TRACE/SCF for the VVER-1000 coolant mixing benchmark showing the hottest coolant in sector 4.**



**Figure 4-33 – Measured coolant temperature distribution at the core outlet for the VVER-1000 coolant mixing benchmark showing the hottest coolant between sector 4 and sector 5.**

It could be concluded from this section that SCF in the coupled codes improves the thermal-hydraulic prediction in the core region, while better simulations of the thermal-hydraulic behavior e.g. the coolant mixing in the reactor pressure vessel of this VVER-1000 RPV can only be expected if the non-symmetrical location is correctly described by a model. In chapter 5, TRACE is coupled to an open-source CFD code – TrioCFD through ICoCo.

There is another problem arising when the codes validation is undergoing, which is the significantly increasing computation time by TRACE/SCF than that of TRACE standalone. During the validation process, the simulation efficiency reduction is unacceptable when SCF is coupled to TRACE. So, the coupled codes optimization is necessary.

## 4.5 The optimization and options for steady-state and transient simulation of TRACE/SCF-ECI

In the previous sections, the coupling of TRACE with SCF using ECI has been described, verified and validated by an academic coolant-mixing problem and a VVER-1000 coolant-mixing benchmark. The results exhibit a correct implementation and better description of the thermal-hydraulic phenomena in the RPV. However, the efficiency evaluation of the coupled codes is found not satisfactory. In order to improve the code coupling efficiency, two optimized methods aiming to accelerate the simulations' speed are developed.

### 4.5.1 Shortcomings of the current temporal coupling

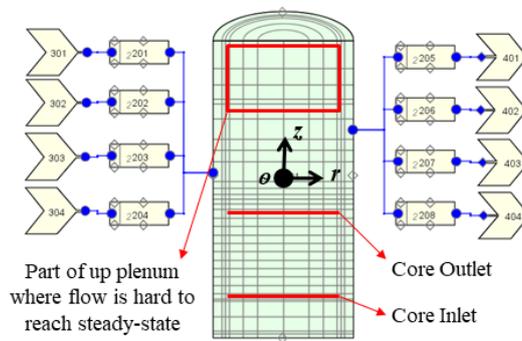
During the testing process, both steady-state and transient coupling schemes spend too much CPU-time compared to the stand-alone versions. For example, TRACE standalone takes one hour to run the previous VVER coolant mixing transient while TRACE/SCF spend 12 hours to the same problem.

This section describes the shortcomings of the coupling schemes followed by the improvements implemented so far.

**a) Steady-state temporal coupling**

During a steady-state calculation of TRACE/SCF, the boundary data of both codes are exchanged at the two interface planes located at the core inlet and outlet. Especially, SCF always passes the converged data to TRACE at each time step. The two codes run in parallel and they synchronize data transfer at the end of each time step. However, some pitfalls arise in a real simulation case.

A steady-state simulation performed with a model of a VVER-1000 reactor shown in Figure 4-34 has been performed to identify the reasons for the low efficiency of the stationary temporal coupling. The main part of the TRACE-model is the VESSEL-component, which has 30 axial levels, 6 radial rings and 6 azimuthal sectors, i.e. in total 1080 3D-cells. Four hot legs and cold legs are connected to the VESSEL. The boundary conditions of the test for the steady-state simulations are derived from Table 4-3 and shown in Table 4-4. The initial velocity in the vessel is set to 0.0 m/s.



**Figure 4-34 – Schematic composition of the TRACE model of the VVER-1000 vessel.**

For this steady-state case, TRACE standalone takes 14min and TRACE/SCF takes 2h15min. During the examination of the significant increase of CPU time consumption, a slow convergence rate of the liquid velocity in TRACE is observed during the simulation, which indicates that the velocity field distribution and its variation have to be carefully inspected. Since the core inlet and outlet planes correspond to the data transfer interfaces, it is necessary to check whether the low convergence rate of the coolant velocity in TRACE are caused by the data transferred from SCF or not. So, the coolant velocity converging behavior at those locations of TRACE should be examined.

**Table 4-4 – Boundary and initial conditions of the steady-state test case to identify the poor efficiency of the TRACE/SCF steady-state calculation.**

	<i>Loop #1</i>	<i>Loop #2</i>	<i>Loop #3</i>	<i>Loop #4</i>
<i>Inlet Mass Flow Rate</i>	4737 kg/s	4718 kg/s	4682 kg/s	4834 kg/s
<i>Outlet Pressure</i>	15.5 MPa			
<i>Inlet Temperature</i>	541.75 K			
<i>Initial Velocity in the Vessel</i>	0.0 m/s			

Figure 4-35 presents the coolant velocity curves of six selected cells at the core inlet and outlet (the positions are marked out in Figure 4-34), from which it can be inferred that the whole system needs around 60 seconds to reach a steady state. Moreover, it can also be noticed from the figure that the velocity at the core inlet and outlet converge quite fast in around 5 seconds and almost keeps constant for the remaining 55 seconds. It is to note that all of the cells at the core inlet and outlet follow similar

curves like that of the selected cells shown in Figure 4-35 (grow from zero to constant in 5 seconds and no change for the last 55 seconds). The curves in Figure 4-35 are just selected for demonstration. The positions of the six cells are in Table 4-5.

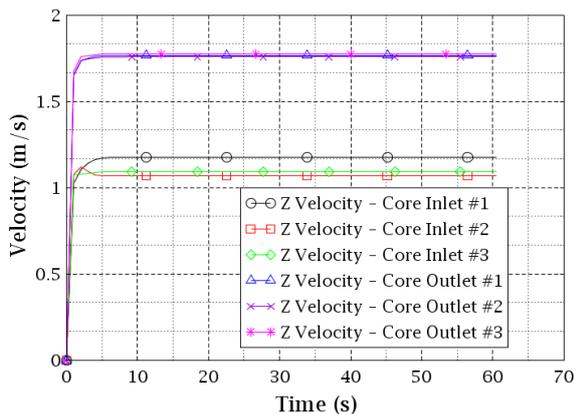
**Table 4-5 – The selected six cells demonstrating the convergence curves of the coolant velocity at the core inlet and outlet for TRACE/SCF.**

	Core inlet			Core outlet		
	#1	#2	#3	#1	#2	#3
$r$	0~0.7537 m	0.7537~1.182 m	1.18~1.582 m	0~0.7537 m	0.7537~1.182 m	1.182~1.582 m
$\theta$	0° ~ 60°	60° ~ 120°	120° ~ 180°	0° ~ 60°	60° ~ 120°	120° ~ 180°
$z$	1.853 m	1.853 m	1.853 m	5.403 m	5.403 m	5.403 m

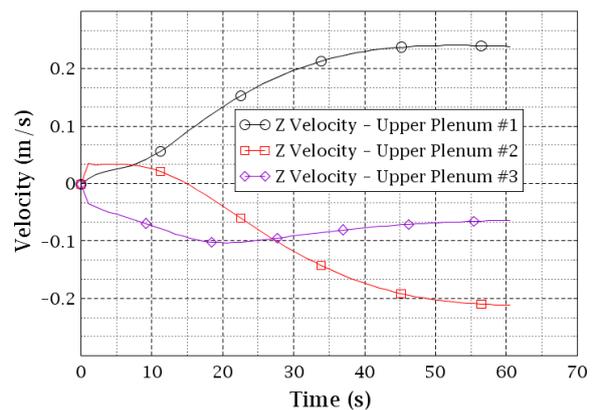
There should be some other parts of the system causing the low convergence rate. Upon closer examination, the coolant velocity instability in the upper plenum is identified to be the crucial factor resulting in the poor convergence performance. The velocity curves are plotted in Figure 4-36, where the velocity of three selected cells in the upper plenum is shown. It is to note that most of the cells in the upper plenum display slowly converging curves. The curves in Figure 4-36 are just selected for demonstration of the low-velocity convergence rate in the upper plenum. The positions of the six cells are in Table 4-6.

**Table 4-6 – The selected three cells demonstrating the poor convergence rate of the coolant velocity at the vessel upper plenum for TRACE/SCF.**

	Upper plenum		
	#1	#2	#3
$r$	0~0.7537 m	0.7537~1.182 m	1.18~1.582 m
$\theta$	0° ~ 60°	60° ~ 120°	120° ~ 180°
$z$	11.353 m	11.353 m	11.353 m



**Figure 4-35 – The temporal evolution of the coolant velocity of the six selected cells at the core inlet and outlet for a VVER-1000 TRACE/SCF model demonstrating the fast convergence of the velocity at those positions.**



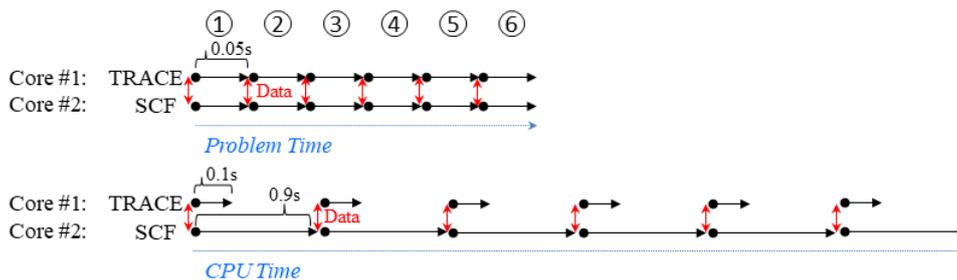
**Figure 4-36 – The temporal evolution of the coolant velocity of the three selected cells at the vessel upper plenum for a VVER-1000 TRACE/SCF model demonstrating the poor convergence rate of the velocity at those positions.**

When the steady-state simulation of TRACE/SCF runs under the temporal coupling scheme described in section 4.2, the data from TRACE to SCF at the core inlet and outlet became stable in 5 seconds.

The feedback from SCF to TRACE should also become stable in a subsequent time step, which means the calculation of SCF from 5 to 60 seconds is not relevant.

**b) Transient temporal coupling**

During the testing of TRACE/SCF transient simulation, a significant increase of the CPU time has been observed, is almost 10 times larger than the one of a TRACE standalone simulation. For a step-to-step transient coupling, each SCF-step costs around 9 times larger CPU-time than a TRACE-step. This transient coupling scheme is straightforward since TRACE and SCF can get the right feedback from each other properly within each time step. However, part of the SCF-calculation in the transient coupling scheme is unnecessary, similar to the steady-state coupling. A scheme of the time-advancement of TRACE/SCF as a function of the problem time and of the CPU-time for the current transient temporal coupling is exhibited in Figure 4-37. The problem time represents the real transient time while the CPU-time, or to be more precise, the elapsed time, stands for the computing time.



**Figure 4-37 – Example demonstrating the un-optimized transient temporal coupling of TRACE/SCF.**

In order to keep tight synchronization of data transfer during the transient simulation, TRACE and SCF have to use the same time step, which is supposed to be 0.05s in Figure 4-37. Here, TRACE spends 0.1s to complete the 0.05s transient step while SCF spends 0.9s. The transient-case in Figure 4-37 will cost in total  $0.9s \cdot 6steps \cdot 2cores = 10.8s$  (TRACE has to wait until SCF finishes the current step since the Exterior Communication Interface (ECI) for the coupling codes runs under blocking mode) to simulate a  $0.05s \cdot 6steps = 0.3s$  transient. It is worth to note that the 10.8s is actually the CPU time. The computing time on clock is  $0.9s \cdot 6steps = 5.4s$ . This transient coupling scheme is correct and straightforward from the programming point of view.

The numerical method applied by the TRACE solver is a kind of a semi-implicit approach, which sufficiently eliminates the Courant number limit on time step size but the material Courant number limit still exists. SCF uses a kind of a fully implicit approach, which successfully eliminates the almost all-numerical limitation on time step growing. The only limit comes from the physics meaning that the maximum time step size of SCF is determined in order to cover the details of the real transients without losing any physical information. For most cases, timescales of the real transients known as the proper SCF time step are larger than TRACE time step size. However, in order to ensure numerical stability and data synchronization, the coupling system has to select and use the smallest time step size of TRACE and SCF, normally the TRACE one. Therefore, restricting SCF to use a smaller time step leads to loss of computational efficiency.

It is to note that the TRACE time step size could significantly reduce if the incoming data from SCF fluctuates fiercely. This is also a possible reason causing the low efficiency of the coupled code. Though the previous case does not suffer such conditions, it is necessary to take this effect into consideration. The relaxation could be applied to the transferred fields between TRACE and SCF for each time step. This dissertation does not implement this method but it is a promising method.

## 4.5.2 The optimized temporal coupling approaches

### a) The optimized steady-state temporal coupling

The main task of the optimization is to remove the unnecessary calculation steps of SCF during the steady-state simulation and thus reduce the computational time. A new optimized steady-state temporal coupling scheme of TRACE/SCF has been developed, which automatically monitors the variation of the exchanged data at the two plane interfaces (core inlet and outlet). In the coupling system, TRACE acts as the master and runs over the whole simulation, while SCF acts as the slave. At the beginning of the simulation, both TRACE and SCF are launched in a parallel way. After a short period of time, when the exchanged data at the two interfaces turns stable, SCF stops and waits while TRACE runs standalone until an upper limit (inherited from TRACE steady-state criteria) for the data change is detected at the two interfaces. Then, SCF is reactivated and it continues the simulation until the data become stable again. This mechanism is illustrated in Figure 4-38.

- 1) Phase 1: TRACE and SCF run together until TRACE-data passed to SCF get stable;
- 2) Phase 2: TRACE run standalone while SCF hangs until a new upper limit for the data variation is detected at the interfaces;
- 3) Phase 3: TRACE and SCF run together until TRACE-data to SCF get stable again;
- 4) Phase 4: TRACE continues until the whole system converges.

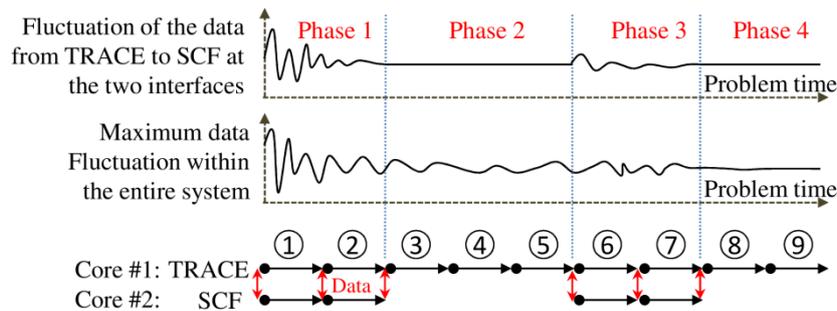
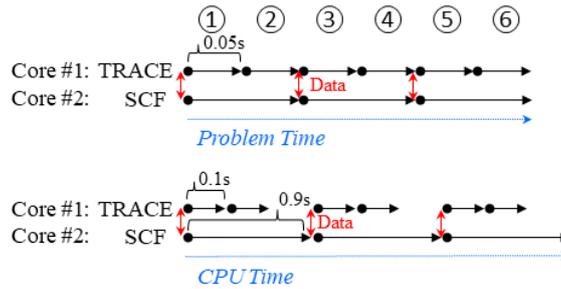


Figure 4-38 – Scheme of the optimized steady-state temporal coupling for TRACE/SCF.

The transient in Figure 4-38 includes nine TRACE steps. For the non-optimized steady-state coupling scheme, nine SCF-steps are calculated as well. If the consumed CPU-time for a single TRACE-step is 0.1s and 0.9s for an SCF-step, The total consumed CPU-time should be 16.2 s ( $0.9s \cdot 9steps \cdot 2cores$ ) (TRACE has to wait for SCF, meaning that even though the TRACE-step is 0.1s, it has to spend 0.8s waiting.). For the optimized steady-state coupling scheme, 5 SCF steps are needed and the consumed CPU-time is around 8.2 s ( $0.1s \cdot 5steps + 0.9s \cdot 4steps$ ) $\cdot 2core$ . As a result, the CPU-time is reduced from 16.2 s to 8 s ( $16.2s - 8.2s$ ). In general, when the exchanged data at the two interfaces fall below the threshold value, it will keep constant until the end without a re-activation of SCF. Nevertheless, the reactivated capability of SCF is still presented in Figure 4-38 for general purpose.

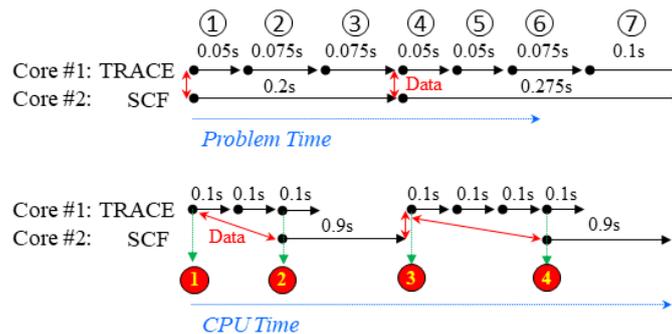
### b) The optimized transient temporal coupling

The goal is to elaborate an alternative method compared to the original step-to-step transient coupling scheme in order to enable TRACE and SCF use different time step in the coupling system while keeping a tight synchronization during a transient simulation. The main idea is to overlap multi TRACE-minor steps with one single large SCF-time step and to determine the data synchronization points in an elaborated manner. Hence, a so-called staggered-step transient coupling scheme was implemented, see Figure 4-39. The example is a 0.3s transient including six TRACE-steps and three SCF-steps. The TRACE time step size is 0.05s and one of SCF is 0.1s. Every two TRACE-steps correspond to one SCF-step. Data synchronization performs at the end of each SCF-step.



**Figure 4-39 – Scheme of the optimized transient temporal coupling for TRACE/SCF.**

Compared with the non-optimized step-to-step transient-coupling scheme in Figure 4-37, the total consumed CPU-time of the optimized staggered-step transient coupling scheme is 5.4 s ( $0.9s \times 3steps \times 2cores$ ), which is 5.4 s faster ( $10.8s - 5.4s$ ) than the original approach. The computing time on the clock is reduced from 5.4s to 2.7 s ( $0.9s \times 3steps$ ), meaning a 50% efficiency improvement. It can also be concluded that the larger the SCF-time step is, the more efficient the coupling system will be. Instead of the ideal case shown in Figure 4-39, time step sizes of TRACE and SCF during real transients are calculated by the codes themselves and cannot be pre-determined. For the sake of tight synchronization, a self-adapted algorithm has been developed, illustrated in Figure 4-40.



**Figure 4-40 – Self-adapted algorithm for the optimized transient temporal coupling of TRACE/SCF.**

The transient is simulated with nine TRACE-steps of the sizes of 0.05s, 0.075s, 0.075s, 0.05s, 0.05s, 0.075s and 0.1s, consecutively. SCF runs only two steps for this transient. Its first time step size is 0.2 s (i.e.  $0.05s + 0.075s + 0.075s$ ), which is exactly the sum of the first three TRACE-steps. The second time step size is 0.275 s (i.e.  $0.05s + 0.05s + 0.075s + 0.1s$ ), which is exactly the sum of the last four TRACE-steps. The following remarks explain the self-adapted-synchronization-point-fixing algorithm:

- 1) At point 1: The TRACE self-calculated time step size is 0.05s and the one of SCF is 0.2s. The codes exchange the boundary conditions.
- 2) From point 1 to point 2: TRACE runs standalone and SCF waits. The TRACE time step sizes are accumulated until their sum reaches 0.2s which is the SCF first self-calculated time step size.
- 3) At point 2: A 0.125s ( $0.05s + 0.075s$ ) of problem time has already been done by TRACE after two steps. The size of its third-time step is calculated to be 0.075s. The system judges that  $0.125s + 0.075s$  is larger or equal to 0.2s. Then SCF wakes up.
- 4) From point 2 to point 3: TRACE and SCF run in parallel and both move one step forward.
- 5) At point 3: The TRACE self-calculated step size is 0.05s and the one of SCF is 0.27s (pay attention, not 0.275s). The two codes exchange boundary conditions. SCF hangs.
- 6) From point 3 to point 4: TRACE runs standalone and SCF waits. The TRACE-time step sizes are accumulated until their sum reaches 0.27s, which is the SCF second time step size.

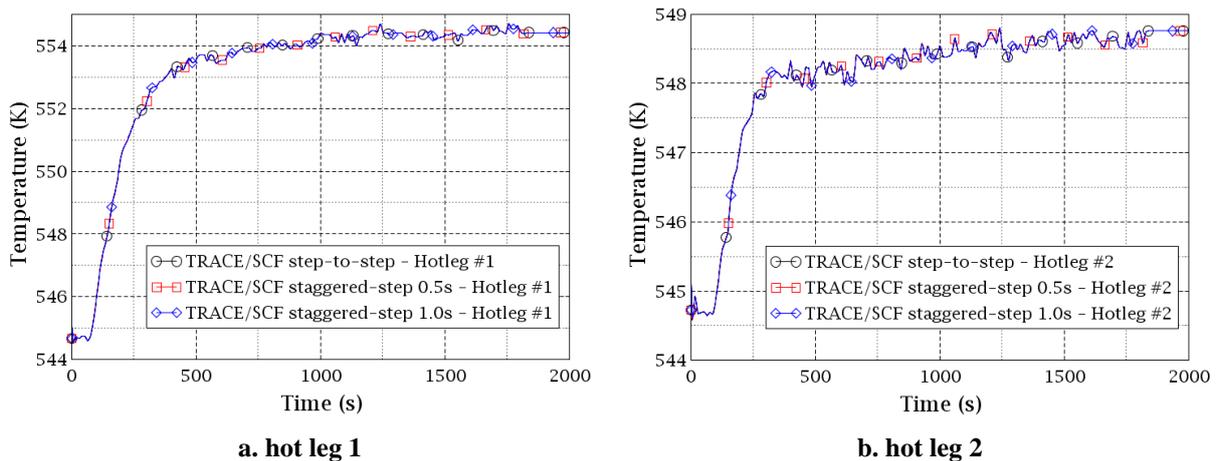
- 7) At point 4: A 0.175s (i.e. 0.05s+0.05s+0.075s) of problem time has already been done by TRACE after three steps. The size of its new time step is calculated to be 0.1s. The system judges that 0.175s+0.1s=0.275s is larger or equal to 0.27s. Then SCF wakes up. The second time step size of SCF is reset from 0.27s to 0.275s in order to guarantee the synchronization.

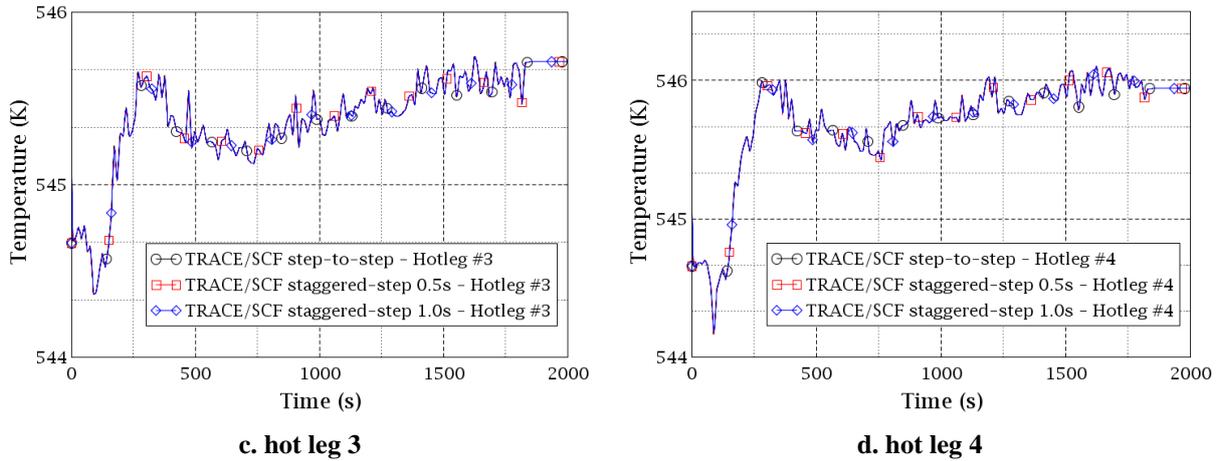
Steps 1 to 7 are exactly the principles of the self-adapted-synchronization-point-fixing algorithm developed for TRACE/SCF. Users do not have to do anything for the synchronization process but they should carefully set the limit of SCF-maximum time step size to ensure the simulation will not miss any information of the real transients. According to Figure 4-40, the CPU-time is reduced from  $0.9s \times 7 \text{steps} \times 2 \text{cores} = 12.6s$  (non-optimized step-to-step transient coupling) to  $4.6s$  ( $(0.1s+0.1s+0.9s+0.1s+0.1s+0.1s+0.9s) \times 2 \text{cores}$ ) (the optimized staggered-step transient coupling). The computing time on clock is reduced from 6.3s ( $0.9s \times 7 \text{steps}$ ) to 2.3s ( $0.1s \times 2 \text{steps} + 0.9s \times 1 \text{step} + 0.1s \times 3 \text{steps} + 0.9s \times 1 \text{step}$ ). With the ability of this algorithm, the staggered-step scheme assures great efficiency over the step-to-step scheme. Nevertheless, the latter method should not be completely abandoned because it may perform better by omitting the logics to match time steps between SCF and TRACE when simulations of very fast transients are performed, where data-transferred is necessary for each time-step.

### 4.5.3 Test of the efficiency of the optimized temporal coupling approach

In order to demonstrate the increase of the computational efficiency of TRACE/SCF coupled with ECI, the coolant mixing test described in section 4.5.1 is simulated here with the optimized TRACE/SCF considering the following options: 1) Step-to-step as the base; 2) Staggered-step while the maximum SCF time step size is set to 0.5s; 3) Staggered-step while the maximum SCF time step size is set to 1.0s.

The coolant temperature evolution at the four hot legs predicted with TRACE/SCF for the three options is compared to each other in Figure 4-41. It can be seen, that all three curves almost overlap each other and there are no obvious differences between the three data sets. The result indicates that the staggered-step temporal coupling approach behaves as good as the step-to-step coupling method.





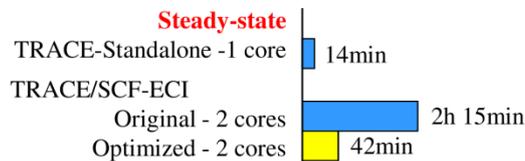
**Figure 4-41 – Comparison of the computed temporal evolution of the four hot leg temperatures in the VVER-1000 coolant mixing benchmark using different time coupling schemes.**

On the premise that the staggered-step method provides physically correct results, the efficiency of this approach is compared with the un-optimized scheme. The computer environment is given in Table 4-7.

**Table 4-7 – Information on the environment for the test of the optimized temporal coupling of TRACE/SCF.**

Operating system	Debian GNU/Linux 8.0
Thermal hydraulic codes	TRACE V5.1051 and SubChanFlow 3.3
Hardware information	Processor – 48 Intel(R) Xeon(R) CPU E5-2697 v2 @ 2.7GHz, Installed memory (RAM) – 378 GB, System type – 64 bit.

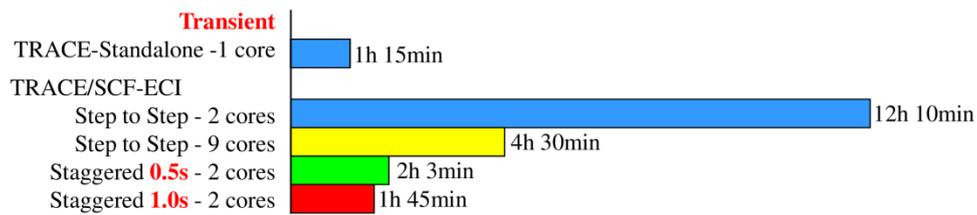
As to the steady-state calculation shown in Figure 4-42, TRACE standalone takes 14 minutes to complete the simulation with one core while the original TRACE/SCF take 2h15min on two cores (one core for TRACE and the other one for SCF). After codes optimization, the coupled steady-state calculation time was reduced to 42 minutes, which is a 69% efficiency improvement.



**Figure 4-42 – Computational effort for the coolant mixing benchmark using different steady-state coupling scheme (original coupled (un-optimized) code, and optimized coupled code) compared to TRACE standalone.**

For the transient calculation, a TRACE-standalone simulation is first performed using one core. The elapsed time is 1 hour 15 min. The coupled code TRACE/SCF in the step-to-step mode using two cores (TRACE for 1 and SCF for 1) requires 12 hours 10 min. Since part of SCF is parallelized using OpenMP, eight cores are assigned to SCF and one core to TRACE in the new testing phase, which results in execution time of 4 h 30 min. TRACE/SCF is then executed under the staggered-step mode

using two cores were used (one for SCF and one for TRACE). Two executions are performed, where the SCF-maximum time step size is set to 0.5s and 1.0s, respectively. The computing time on the clock of the first run turns out to be 2 h 3 min and that of the second run is 1 h 45 min, see Figure 4-43. This is a nearly six times faster speed compared to that by TRACE/SCF simulation using the step-to-step mode. This demonstrates the high efficiency of the staggered-step temporal coupling approach. It can be concluded that the larger the SCF-maximum time step size is the faster the simulation is. However, the SCF time step size should be carefully selected according to the simulated problems. It still has a limit.



**Figure 4-43 – Computational effort for the coolant mixing benchmark using different transient coupling scheme (original coupled (un-optimized) code, and optimized coupled code) compared to TRACE standalone.**

## 5 The Multi-Scale Simulator TRACE/TrioCFD based on ICoCo

### 5.1 Modification of TRACE to allow data transfer

For the coupling of TRACE with TrioCFD using ICoCo, an ICoCo-module has to be implemented in TRACE. In addition, a supervisor is needed to control the data exchange and to synchronize the time-advancement of the coupled codes. There are three options for supervisor implementation:

- 1) C++ script.
- 2) Python script.
- 3) The SALOME platform.

The C++ script is the best choice as a supervisor due to the support to MPI protocol. However, the use of the SALOME-platform (YACS module) is still a promising alternative that uses a built-in Graphical User Interface (GUI) for the realization of user-friendly and transparent coupling procedures. For example, if two solvers are equipped with ICoCo-module and implemented in the SALOME-platform, a C++ supervisor also works, but not vice versa. This is because more works are involved in developing the TRACE-ICoCo module for SALOME than a pure C++ supervisor outside SALOME.

The next sections describe the TRACE-ICoCo module associated with the C++ supervisor and the upgrade of the ICoCo-module for its implementation inside SALOME. Finally, additional programming work is needed to adapt the ICoCo-module of TrioCFD for the coupling with TRACE-ICoCo.

#### 5.1.1 Implement ICoCo for TRACE

ICoCo is a collection of functional standards. Hence, developing an ICoCo for a specific solver requires the definition of a program frame and the building of connection channels between ICoCo and the solvers. Selected main functions defined in ICoCo-frame are listed in Table 5-1 along with a brief explanation.

**Table 5-1 – Some of the main functions defined by ICoCo.**

<b>setDataFile</b>	Specify the input file, its location, output files' location, and argument.
<b>initialize</b>	Do the initialization and generate the MED-format mesh in either file or memory.
<b>Presenttime</b>	Present the current computer time.
<b>computeTimeStep</b>	Calculate the time-step value for the current step.
<b>initTimeStep</b>	Initialize the time-step size.
<b>solveTimeStep</b>	Run the current time-step.
<b>getOutputMEDField</b>	Extract the desired field in MED format and write the field to the MED-format mesh.
<b>getOutputFieldsNames</b>	Get the name of the extracted MED-format field.
<b>getInputFieldsNames</b>	Get the name of a MED-format field from another code.
<b>setInputMEDField</b>	Translate and write the MED-format field from another code to the code's memory.
<b>terminate</b>	Terminate the execution.

A critical review of these functions leads to the following conclusions about the solvers to be equipped with the ICoCo-module:

- 1) TRACE has to be equipped with meshing in a specific format (MED-format) since the spatial mapping of the two involved solution domains relies on the mesh superposition (MED-format) and on the functionalities of the MEDCoupling library. The data of calculation parameters of each solver stored in the fields of the meshing is used for both feedback-exchange among the involved solvers and for post-processing purposes.
- 2) Each solver, e.g. TRACE, must be modularized and split from one integral executable to several functional components (static or dynamic libraries) for the purpose of flexible inter-communication and synchronization with the other solvers.

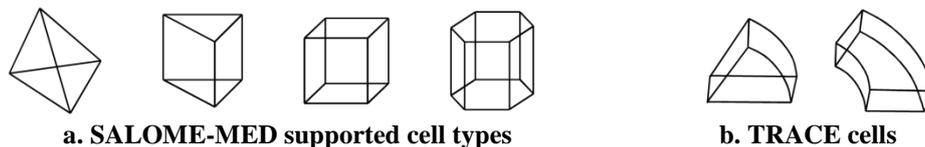
These two aspects are presented and discussed in the following sub-sections.

### **a) Development of the MED-format mesh for TRACE**

For a successful coupling, physical field mapping between different code meshes is one of the critical issues, which largely determines the efficiency and even validity of the coupling. But before trying to solve this challenge, the meshes have to be properly and explicitly defined as the essential prerequisites, no matter in a file or in the memory.

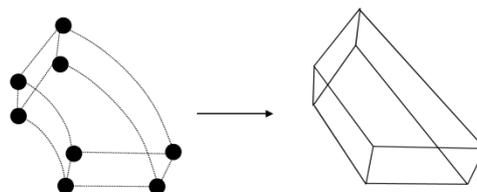
Almost all CFD codes have their own meshes, which are logically isolated from the physical fields and the numerical definition of a problem. But for TRACE, the mesh is implicitly defined in the input file which contains a mixture of numerical data and mesh description. Hence, a meshing based on MED format must be developed for TRACE in an explicit way. Since the coupling between TRACE and TrioCFD focus on the phenomena inside the RPV, a meshing for the 3D VESSEL component of TRACE is developed.

The MEDCoupling library of SALOME supports five cell types: tetrahedron, triangular prism, hexahedron, hexagonal prism, and polyhedron, which are displayed in Figure 5-1a. But they can't be applied to represent a TRACE cell. This is because the cells in TRACE are either fan-shaped or annular (Figure 5-1b). So, the only possible option is the polyhedron.



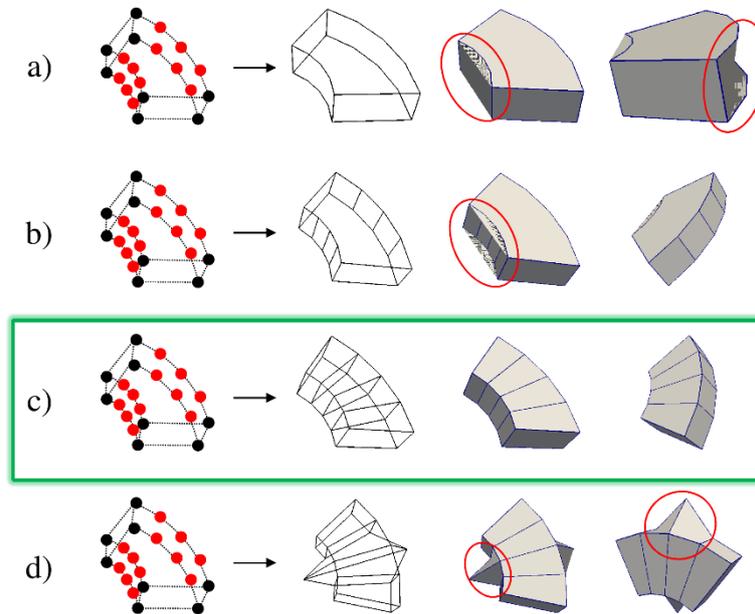
**Figure 5-1 – Cells supported by MEDCoupling library and the typical TRACE cells.**

Normally, only eight points are used to define a TRACE annular cell because TRACE uses the cylindrical-coordinate system to model the VESSEL component. However, since MEDCoupling library applies only Cartesian coordinates, the derived eight points from TRACE input file for one single cell are not sufficient to form cells which contain space curves under Cartesian coordinates. The direct result is a cubic which could not represent the real TRACE annular cell, Figure 5-2.



**Figure 5-2 – Building of a TRACE typical cell if only eight points are applied**

In order to complete the cell, some assistant points must be inserted (the red points in Figure 5-3). Those points could be treated as an interpolation between two original TRACE points. The interpolation is automatically done based on the user-defined resolution (the curves amount to approximate a circle). However, there arises another challenge known as the creation of the spatial faces with those points. The possible approaches are presented in Figure 5-3, following which are the explanations.



**Figure 5-3 – Simulation of a typical TRACE cell using different approximation methods a) direct point use, b) quadrangles approximation for spatial faces, c) quadrangles approximation for spatial and multi-points (upper and bottom) faces, d) use of non-coplanar points.**

- Directly use the points to form the spatial faces. Only six faces are involved in the annular volume. But this approach causes misshapened cells highlighted by the red circles.
- Use several sets of quadrangles to approximate the two spatial faces. Now the volume is properly defined. However, there are some display problems for the two planes containing more than four points highlighted by the red circles.
- Use several sets of quadrangles to approximate both the spatial faces and the multi-points planes. Now the annular volume is perfectly defined and displayed and considered to be adequate for further use.
- This is exactly not an option for the cell definition but disclaims that the co-planarity of the points is the crucial reason affecting the cells' building.

Up to this point, a TRACE mesh could be explicitly built by the third approach with the help of the capabilities of the MEDCoupling library. No additional definitions except the TRACE input file are required. The mesh generation logic is defined within the ICoCo framework out of TRACE source, which means no TRACE source is touched. Actually, this meshing capability for TRACE is developed within the “initialize” function (Table 5-1), which means that within a coupling simulation system, once the TRACE-“initialize” function is executed, the TRACE meshes are generated and the mesh files are located in the root directory. Main steps of this meshing developing process are:

- Link the MEDCoupling library to the TRACE “initialize” function. Make the MEDCoupling meshing manipulation functions available within this function;
- Compile TRACE to a dynamic library. Link the library to the TRACE “initialize” function. Make the TRACE functions and arrays e.g. the arrays storing the VESSEL geometry data visible within this function;

- 3) Develop a logic referring to the TRACE VESSEL geometry arrays, through which derive the basic structural mesh points from TRACE memory (the black points on the left plots of Figure 5-2 and Figure 5-3).
- 4) Develop the logic to meanly insert the auxiliary points (the red points on the left plots of Figure 5-3). Pay attention that the basic points in step 3 and the auxiliary points are generated for entire VESSEL component;
- 5) With the prepared points set, define their relationships e.g. which subset of points belong to which cell and how they should be connected in order to properly present the cell. Those point-relationships are stored in an array and will be further processed by MEDCoupling library;
- 6) With the connectivity array and the points set, use the MEDCoupling library to build the TRACE normal-cell-mesh (the volume mesh in a fan or annular shape). Store the mesh in the memory and write it to file as well;
- 7) Additionally, two additional TRACE meshes are generated except the normal-cell-mesh: the tetrahedron-cell-mesh and the edge-mesh. The motivations of such meshes development are explained in the following paragraphs. The edge-mesh (Figure 5-6) is directly derived from the normal-cell-mesh with the help of MEDCoupling library;
- 8) The generation of tetra-cell-mesh is a little bit complicated. The points set for this mesh is totally the same with the normal-cell-mesh, however, the points-relationships or in other words the connectivity array significantly differ that of the normal-cell-mesh. With the MEDCoupling library, a TRACE VESSEL mesh which only contains tetrahedron cells while do not reflect the real TRACE cells could be built. Nevertheless, the relations of the tetrahedron cells and the normal TRACE cells are implicitly defined with some tricks.

A VVER-RPV-model has been utilized to verify the newly-developed mesh generation function of TRACE. Three kinds of meshes are generated:

- 1) Figure 5-4 presents the normal-cell TRACE mesh, which is explained by previous paragraphs;
- 2) Figure 5-5 displays the tetra-cell TRACE mesh. The TRACE vessel model is composed of lots of tetrahedron cells which is totally different from the polyhedron cell as the normal-cell TRACE mesh. Thanks to the good support of the tetrahedron cell by MEDCoupling, the development of the tetra-cell TRACE mesh is not that tricky than the normal-cell TRACE mesh. The motivation of this work is to simplify the mesh interpolation between different meshes;
- 3) Figure 5-6 displays the edge TRACE mesh. This kind of mesh is derived from the normal-cell TRACE mesh and is used to store and interpolate edge-based physical fields.

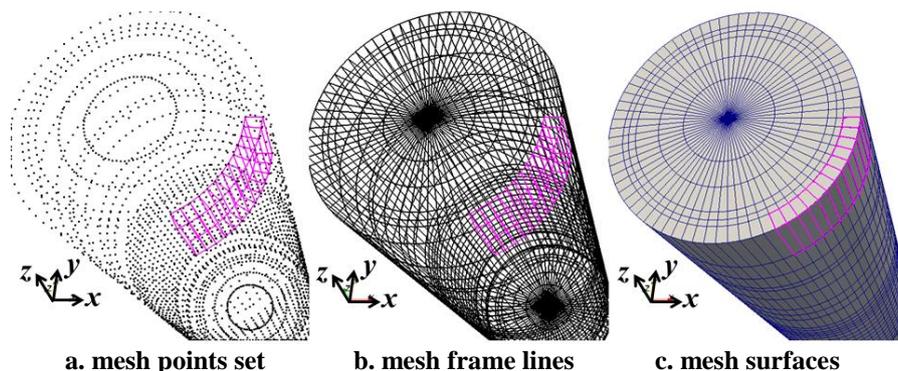


Figure 5-4 – MED normal-cell-mesh of TRACE for VVER-1000 reactor.

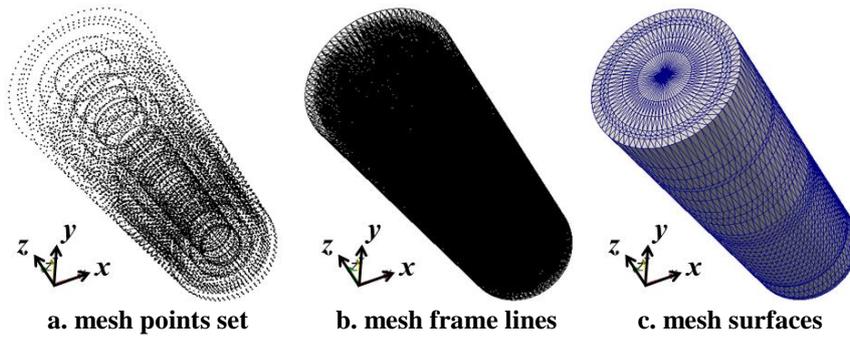


Figure 5-5 – MED tetra-cell-mesh of TRACE for VVER-1000 reactor.

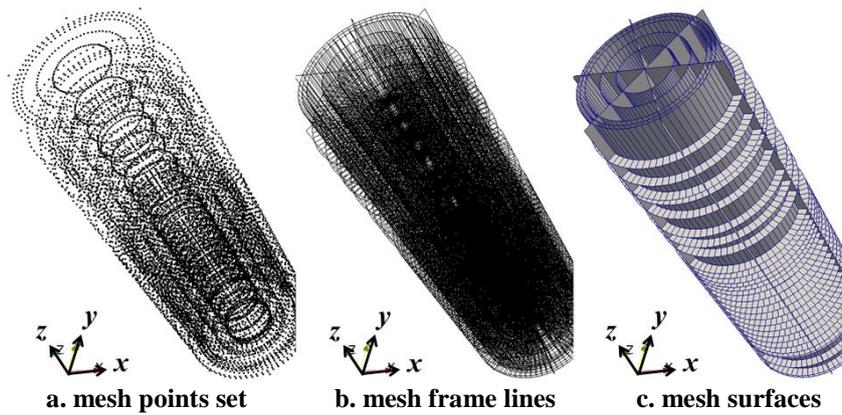


Figure 5-6 – MED edge-mesh of TRACE for VVER-1000 reactor.

The normal-cell mesh is used to store the cell-based fields e.g. density, temperature for post-processing purpose. Pay attention that the purple-line-covered volume in Figure 5-4 stands for a single cell unit while not the simple cubic as shown in the figure. The tetra-cell mesh is developed for mesh-interpolation and coupling purpose, since the interpolation tools of MEDCoupling can't recognize normal TRACE cells (fan-shaped and annular cells) and the best-recognized interpolation type is tetrahedron as shown in Figure 5-7. The edge mesh has to be developed for both post-processing and interpolation of edge-data e.g. velocity and mass flow. Completed with the development of some other key functions, various kinds of data are written to the three meshes for both post-processing and field-interpolation purpose. At the last of this section, it is to note that the typical TRACE meshes, no matter in annular form or fan form, are actually approximated with groups of simple quadrangles or hexahedrons (Figure 5-8). Since this is an approximation, the resolution of the lines, edges, faces, and volumes could be manually defined by the users, which is quite a flexible feature for various purposes.

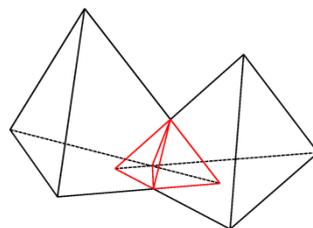


Figure 5-7 – The tetrahedron-mesh intersection could be properly handled by MEDCoupling library.

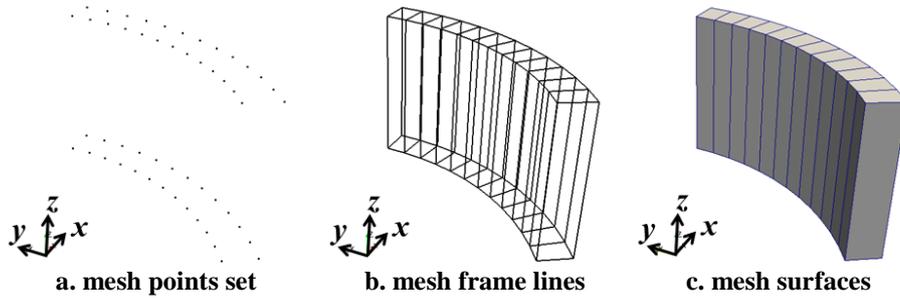


Figure 5-8 – A typical TRACE cell is approximated with a group of quadrangles or hexahedrons.

**b) Development of the ICoCo functional components for TRACE**

There are two steps to develop the ICoCo functional components for TRACE:

- 1) Re-organize the TRACE source to reflect the ICoCo functions.
- 2) Develop the C++ envelope to wrap the TRACE Fortran computing engine.

Finally, the TRACE-ICoCo module is developed and their descriptions are listed in Table 5-2.

Table 5-2 – The ICoCo functions for TRACE

<b>setDataFile</b>	inputPath	Declare the path of the input files.
	outputPath	Declare the directory where to put the output files.
	fileName	Specify the input file's name.
	argument	Input the necessary argument to control the calculation.
<b>initialize</b>	origin_X_m	The offset on x coordination of the generated mesh.
	origin_Y_m	The offset on y coordination of the generated mesh.
	origin_Z_m	The offset on z coordination of the generated mesh.
	theta_grad	The rotation degree of the generated mesh.
	return	Return the total transient problem time defined in the input file.
<b>Presenttime</b>	No input or output ports.	
<b>computeTimeStep</b>	run	-1 – this step will be skipped. 1 – normal condition, go ahead.
	return	Positive value – the calculated time-step size for the current step. -1.0 – the current step will be skipped.
<b>initTimeStep</b>	dt	Positive value – The time-step size for the current step. -1.0 – the current step will be skipped.
	return	The current problem time.
<b>solveTimeStep</b>	Run	-1 – this step will be skipped. 1 – normal condition, go ahead.
	return	0 – keep the iteration. 1 – stop the code. 2 – repeat the current time step. 3 – for steady-state, it means TRACE convergences; for transient, begin the iteration right now.
<b>getOutputMEDField</b>	field_name	The desired fields for output.
	internal_time	0.0 – write the field into the mesh for each time-step. Non 0.0 – only write the field into the mesh for the internal-time.
	height	0.0 – a three-dimensional field is desired. Non 0.0 – a two-dimensional field is desired and the field is derived from the given height of the vessel.
	return	The desired field.

<b>getOutputFieldsNames</b>	field	The entire field in MED format.
	return	The name of the input field.
<b>getInputFieldsNames</b>	field_name	The name of the input field.
	return	The name of the field.
<b>setInputMEDField</b>	field_name	The name of the input field.
	src_field	The entire field in MED format.
<b>IsStationary</b>	field	The field to be compared, for instance at a current time step, the previous field will be automatically saved.
	error	The user input convergence error criteria.
	step	The user-defined of how many steps should be skipped to compare the two fields.
	stop	Tell this routine if it is time to delete the keep-in-memory array so that to clear its occupied memory.
	return	A bool indicating whether the field converges or not.
<b>terminate</b>	No input or output ports.	

For TRACE standalone (Figure 5-9), all of the sources including subroutines and modules (mainly in Fortran) are compiled to a single executable which can directly run a simulation. Compared to the original TRACE (Figure 5-9), the structures of TRACE with ICoCo now involves more levels (Figure 5-10). In order to develop the ICoCo module for TRACE, the following work has to be implemented:

- 1) Re-organize the TRACE source to reflect the ICoCo functionalities. The new Fortran source of TRACE is represented by the first inner pie of Figure 5-10;
- 2) Since the ICoCo functions can not directly access TRACE data with TRACE original subroutines, new Fortran subroutines have to be developed acting as the data-transfer interface between TRACE and ICoCo. Those new subroutines are represented by the inter-ring of Figure 5-10. The TRACE original source is wrapped by the newly developed Fortran subroutines and flexible communications (bidirectional arrows) are possible between them;
- 3) The TRACE reorganized source and the new subroutines are compiled to a static library. The library now has suitable input and output ports with the help of the new Fortran subroutines;
- 4) According to the ICoCo framework, each of the functions listed in Table 5-2 has to be developed. They are in C++. These ICoCo functions are represented by the outer ring of Figure 5-10;
- 5) The ICoCo wrapper together with the TRACE computation engine (the static library involving TRACE reorganized source and the new developed Fortran subroutines) are compiled to a dynamic library. This TRACE-ICoCo library now is ready to be called by the supervisor.

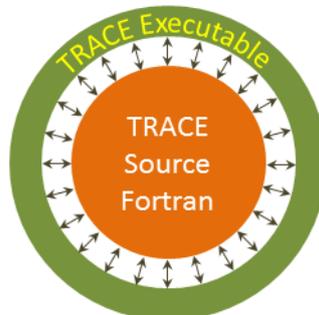
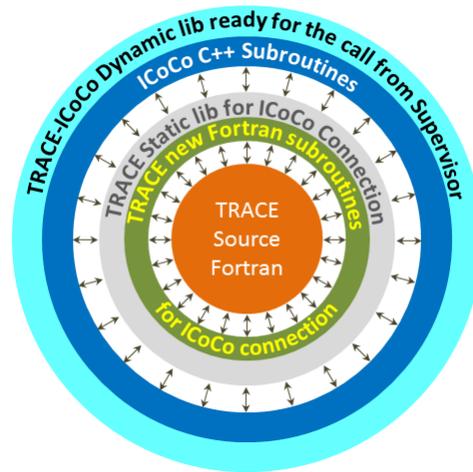


Figure 5-9 – The original structure of TRACE with one executable.

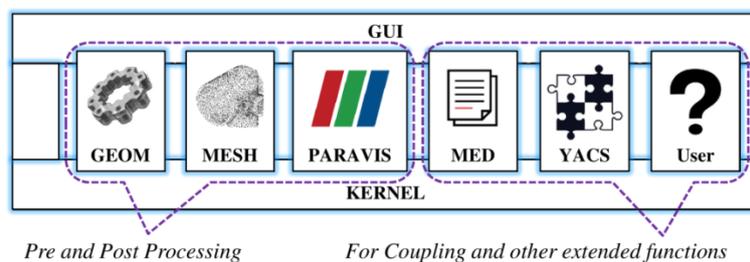


**Figure 5-10 – The structure of TRACE with new developed Fortran subroutines and the C++ ICoCo wrapper.**

It is worth to note that now within the multi-scale thermal-hydraulic project in KIT-INR, an ICoCo module has been developed for SCF and they are successfully coupled to TRACE-ICoCo. This is an approach also based on the domain-overlapping method, which differs from the coupled code TRACE/SCF based ECI with the domain decomposition method.

### 5.1.2 Implementation of TRACE-ICoCo into SALOME

The SALOME platform is an open-source software framework for numerical pre-, post-processing, and integration of numerical solvers in various scientific domains. It is supported by CEA, EDF, and OPEN CASCADE. SALOME has already been employed to perform a wide range of simulations, which are typically related to industrial equipment in power plants (nuclear power plants, wind turbines, dams...). SALOME is mainly composed of eight modules (Figure 5-11) among which, KERNEL and GUI provide the core functionalities of SALOME, GEOM is for CAD tasks, MESH is in charge of generating computational grids, PARAVIS is nothing but ParaView devoted to post-processing, MED contains lots of mesh interpolation tools and also supplies a universal data format standard for all other modules, YACS is used to organize and monitor the calculation chains, the last stands for the user-developed modules. In sum total, KERNEL and GUI are the base, GEOM, MESH, and PARAVIS are for pre- and post-processing, MED, YACS, and user-module are closely related to coupling issues.

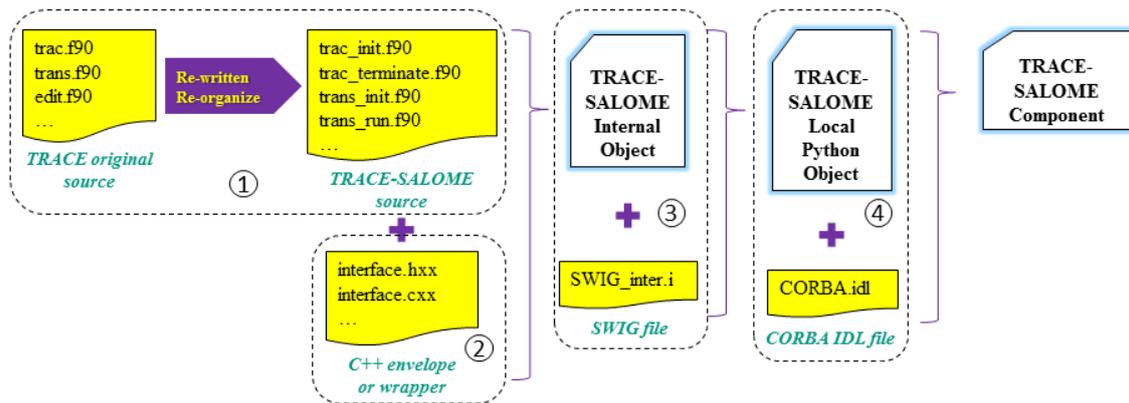


**Figure 5-11 – Diagram of SALOME architecture.**

SALOME-YACS provides a user-friendly GUI to organize the calculation by dragging and connecting some basic functional-components. The implementation of TRACE to SALOME-YACS is complicated. Nevertheless, with the help of SALOME-tools (a toolkit to standardize the

developmental work within SALOME), the entire process can be generalized into four main steps (Figure 5-12).

- 1) Re-written and re-organize the original TRACE source code to meet the functional modularized request of SALOME-YACS.
- 2) Develop a C++ envelope to wrap the lower TRACE-Fortran computing engine forming a so-called TRACE-SALOME internal object.
- 3) Develop a SWIG file to stick the internal object to SALOME-YACS python layer forming a so-called TRACE-SALOME local python object.
- 4) Develop a CORBA file to establish the communication channels for TRACE module forming the final so-called TRACE-SALOME Component.



**Figure 5-12 – Schematics of the different development steps 1-4 to implement TRACE into SALOME.**

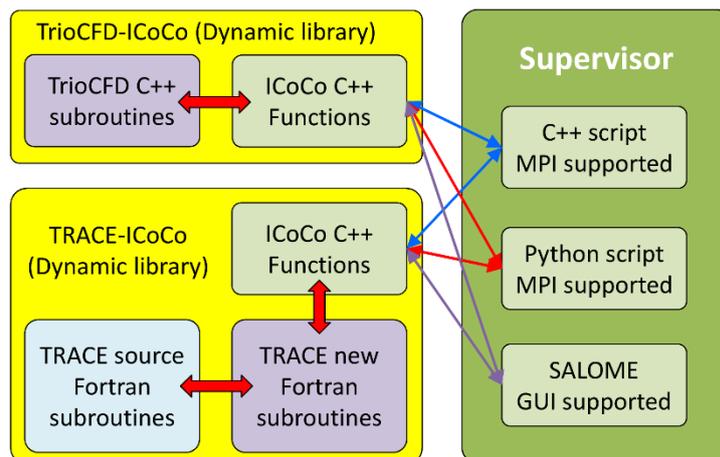
The last two steps shown in Figure 5-12 are automatically done by the SALOME-tools. Hence, no need for the users to spend time and effort on neither the SWIG nor the CORBA layer, which are very complex features of SALOME. The main challenges for the users are step 1 and 2 that requires in-depth knowledge of the TRACE source code and involve substantial programming effort.

After TRACE-ICoCo is properly implemented into the SALOME platform, the functional components are visible in the SALOME-YACS panel. Then, they can be dragged and connected to each other to form a calculation diagram. The results can be easily processed by SALOME-PARAVIS.

It is worth to note that within the multi-scale thermal-hydraulic project in KIT-INR, SCF with its ICoCo module has been also implemented into the SALOME platform, whose procedure is quite similar to that of TRACE. Now, TRACE and SCF were coupled successfully within the SALOME platform and they could run properly under the supervision of the YACS GUI module.

## 5.2 Description of the coupled system of TRACE/TrioCFD-ICoCo

The fundamental principles of the code coupling based on ICoCo are shown in Figure 5-13. Based on the discussion of the different coupling approaches done in section 4.5, the TRACE/TioCFD coupling is implemented based on the explicit temporal coupling with a C++ script as a supervisor and it relies on the domain-overlapping approach for the spatial mapping between the involved domains.



**Figure 5-13 – Schematics coupling logics of the coupled codes TRACE/TrioCFD with ICoCo coordinated by an external supervisor.**

Due to the complexity of the domain-overlapping approach, the key elements of these methods and the basic principles are presented and discussed hereafter.

## 5.2.1 The domain-overlapping method

The advantage of the domain-overlapping approach compared to the domain-decomposition approach is its robustness and efficiency. Moreover, another benefit of the domain-overlapping approach is that the models of the involved codes are the same for both standalone and coupled simulations. Thus, no additional modifications are required to transfer models from a system code standalone simulation to a coupled simulation. This is compulsory in case of the domain-decomposition approach, where some additional interfaces have to be defined and added to the original standalone models. However, the domain-overlapping approach may face the following drawbacks:

- 1) Intensive alterations of the source of the system code may go down to the governing equation set in order to properly use the data from CFD-codes since there are no predefined interfaces in system code receiving the incoming data as updated boundary conditions.
- 2) In-depth understanding of the source code is needed and heavy programming work may be necessary.

These are the main reasons for the few implementations of the domain-overlapping approach. The first domain-overlapping application is the multi-scale coupling of the thermal-hydraulic system code – CATHARE and the CFD code – TRIO\_U for steady-state and transient simulations of e.g. Gas Fast Reactor (Perdu & Vandroux, 2008) and for the analysis of the Phenix Sodium-cooled Fast Reactor (SFR) (Tenchine, et al., 2012) described in detail in (Bavière, et al., 2014).

There are also several other attempts to use domain-overlapping method e.g. SAS4A/SASSYS-1/STAR-CD (Fanning & Thomas, 2010) and RELAP5/STAR-CCM+ (Jeltsov, et al., 2013) (Papukchiev, et al., 2015). However, they are not the ideal domain-overlapping approach since the feedback from CFD codes to system codes just occurs at the boundaries and is either 1D or 2D, while the fields in the overlapped domain of the system codes are never refined by the CFD results, which might lead to a sudden physical fields change over the coupling interfaces in the system code. This feedback is not in a natural way and might introduce non-physical effects into the system simulation. Moreover, the new boundary conditions from system code to CFD code might not be ideally computed due to the limited correlation from CFD to system code.

The first time that 3D CFD results are used to correlate the solution of a system thermal-hydraulic code in the entire overlapped domains was done by the coupling of RELAP5 and FLUENT. The “Closure-on-Demand” (Cadinu, et al., 2008) or “Coupling-by-Closure” (Cadinu & Kudinov, 2009) method was invented to properly implement the domain-overlapping approach. It drops the convective pressure drop as well as the form-friction pressure drop from the system code’s motion equation and uses the total pressure drop from FLUENT to represent the two pressure drop contributions by superimposing the two effects all into the form-friction loss coefficient.

Following this basic idea, TRACE and STAR-CCM+ were coupled. The CFD fine fields are used to calculate the new closure coefficients for TRACE, which produce CFD-like hydraulic results in the entire overlapped domain. This coupled code works well for both 1D (Grunloh & Manera, 2016) and 3D (Grunloh, 2016) system models. And as a supplement to the correlation for convective and form-friction pressure drops in RELAP5/FLUENT, the inertial effect was also well examined and eliminated from the total pressure drop from STAR-CCM+ to counteract the non-erasable inertial pressure drop in TRACE motion equation. Furthermore, the velocity fields in the system code overlapped domain are correlated by applying a newly developed “Velocity Matching Faces (VMF)” method to reconstruct the disrupted inner-relationship between the pressure and velocity fields. Another highlight of this work is the in-depth exploration and optimization of the performance of this domain-overlapping coupling approach. As a result, the Inertial Domain Overlapping (IDO) and Stabilized Inertial Domain Overlapping (SIDO) methods were established. This is the first well-developed coupled code using the closure-updated method for domain-overlapping coupling where the feedback is bidirectional and the data synchronization is strictly ensured by adopting an Operator Splitting (OP) temporal coupling method (Zerkak, et al., 2015). The highlight of this method is that the CFD results in the overlapped domain can be used to correlate corresponding fields of the system codes in the entire overlapped domain. For the test phase, various academic cases were applied and the results were encouraging. Despite it, the following native drawbacks still remain:

- 1) The coupling relations of pressure and velocity are unnecessarily interrupted;
- 2) The correlations are only for the system hydraulic parameters e.g. coolant velocity and pressure, while other key parameters e.g. coolant temperature and boron concentration are not included.

In order to overcome these drawbacks and extend the correlations to cover the coolant temperature and boron concentration, a so-called Dynamic-Implicit-Additional-Source (DIAS) approach has been developed in this dissertation. This method is able to correlate four TRACE thermal-hydraulic parameters in the entire TRACE overlapped domain using the high-resolution data predicted by TrioCFD. These parameters are discussed hereafter:

- 1) The coolant velocity and pressure. The fine pressure and coolant velocity fields predicted by TrioCFD are used to update the friction and form loss coefficient  $K$  for each edge located in the overlapped domains of TRACE. Hence, TRACE predicts TrioCFD-like hydraulic results. The two parameters are correlated together because they are tightly coupled in the momentum equation.
- 2) The coolant temperature. The correlation goes deep into the TRACE-numeric i.e. the linearization process and appends an implicit additional heat source to the energy equation. This method assures that the coolant temperature in the TRACE overlapped domains coincide with ones of TrioCFD.
- 3) The boron concentration. The correlation is implemented at the end of each time step and appends an implicit additional boron source to the boron function. This method assures that the boron concentration in TRACE overlapped domains correspond to the ones of TrioCFD.

Details of the DIAS method are explained in section 5.3 alongside with its verification. The following sections partly depend on the DIAS method. It is to note that to implement the DIAS method for the domain-overlapped correlation from TrioCFD to TRACE doesn’t mean the refined TRACE fields in the overlapped domain is taken for final analysis. Because the fine TrioCFD result is already available

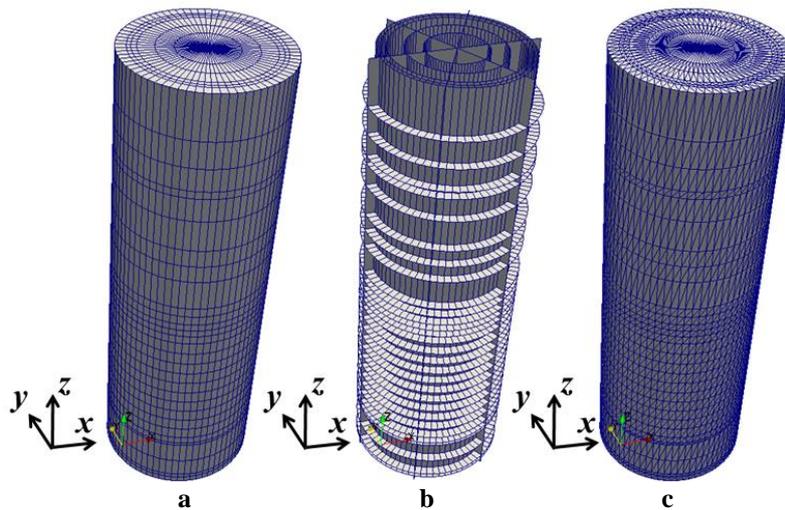
in the overlapped domain. It is meaningless to analysis course field instead of the fine field. Nevertheless, the overlapped-domain correlations to TRACE is to ensure the fine information from TrioCFD is passed to TRACE as much as possible thus let TRACE be able to give better-refined boundary conditions to TrioCFD for the following time steps.

## 5.2.2 The data transfer between TRACE and TrioCFD thermal-hydraulic domains

In section 5.1, the different kinds of meshes developed for TRACE have been discussed, namely:

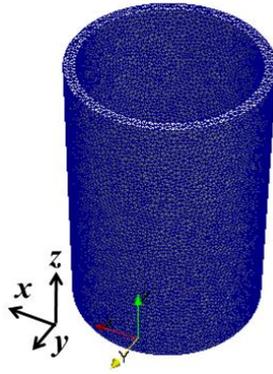
- 1) The normal-cell mesh (Figure 5-14a) represents the real TRACE's cells in fan-shape or annular-shape. It is used for storing the normal cell-based data e.g. the coolant temperature, the pressure, etc. for post-processing;
- 2) The tetrahedron-mesh (Figure 5-14c) is used for the mesh or field interpolation and mapping process for cell-based data and it is especially used for the coupling purposes;
- 3) The edge-mesh (Figure 5-14b) describes all the inner faces within the TRACE vessel component. It is first used for the edge-based field interpolation and mapping e.g. the coolant velocity, the pressure drop. It is also used for the edge-based data post-processing.

All TRACE-meshes in this chapter are based on the VVER-1000 nuclear power plant model.



**Figure 5-14 – Geometric structure of the normal-cell mesh (a), edge mesh (b), and tetrahedron-cell mesh (c) for TRACE.**

For the purpose of demonstration and explanation, TrioCFD only simulates the downcomer while TRACE represents the whole reactor vessel. So the overlapped region only covers the downcomer region. In addition, the TrioCFD model is simplified to a cylindrical shell as shown in Figure 5-15. This simplified TrioCFD model is only used for general illustration purpose. So no geometrical or mesh details are given.



**Figure 5-15 – Downcomer mesh of TrioCFD for demonstration of data flow within the coupled code TRACE/TrioCFD.**

Since the DIAS method includes correlations to four system parameters in the overlapped domain, e.g. the coolant velocity, the pressure, the coolant temperature, and the boron concentration, the data transfers between TRACE and TrioCFD are correspondingly specified into two categories:

- 1) The first one includes both the edge mesh and tetrahedron-cell mesh because the correlations to the two hydraulic parameters are performed simultaneously;
- 2) The second one is for the correlations of the coolant temperature and boron concentration where the tetrahedron-cell mesh is involved.

Figure 5-16 presents the data flow for hydraulic correlations of the coupled code. The data flow follows the procedures listed below for the hydraulic parameter correlations.

- 1) TRACE runs a time step forward and saves the three-dimensional pressure fields in the normal-cell mesh for post-processing and in the tetrahedron-cell mesh for the field mapping in the next step. At the meantime, the three-dimensional coolant velocity fields are saved in the edge mesh for both post-processing and field mapping purpose.
- 2) With the help of the MEDCoupling library, insert the upper boundary 2D edge mesh of TrioCFD into the TRACE's edge mesh and derive the desired inlet velocity boundary conditions for TrioCFD by performing the edge-to-edge mesh interpolation. Then, insert the lower boundary 2D edge mesh of TrioCFD into the TRACE's tetrahedron-cell mesh and derive the desired outlet pressure boundary conditions for TrioCFD by performing the cell-edge mesh interpolation. The boundary condition transfer from TRACE to TrioCFD is shown in Figure 5-17a.
- 3) With the newly updated inlet and outlet boundary conditions, TrioCFD runs a step forward and write all of the calculated fields in its mesh.
- 4) TRACE assembles the tetrahedron-cell and edge meshes to the TrioCFD 3D mesh, derive the three-dimensional pressure and velocity fields and write them to the tetrahedron-cell and edge meshes. The transfer from TrioCFD to TRACE is exhibited in Figure 5-17b and Figure 5-17c.
- 5) Since the tetrahedron-cell mesh does not represent the real TRACE cells, the pressure field in it is first integrated and then written to the normal-cell mesh.
- 6) The pressure field in the normal-cell mesh is first used to calculate the pressure drop across each edge and then the pressure drop field together with the velocity field is used by the DIAS method to correct the friction and form losses coefficients at the edges. In Chapter 12, the details of the correlation in the frame of the DIAS method are described.
- 7) With the updated parameters, TRACE moves a step forward and generate the fresh new fields.

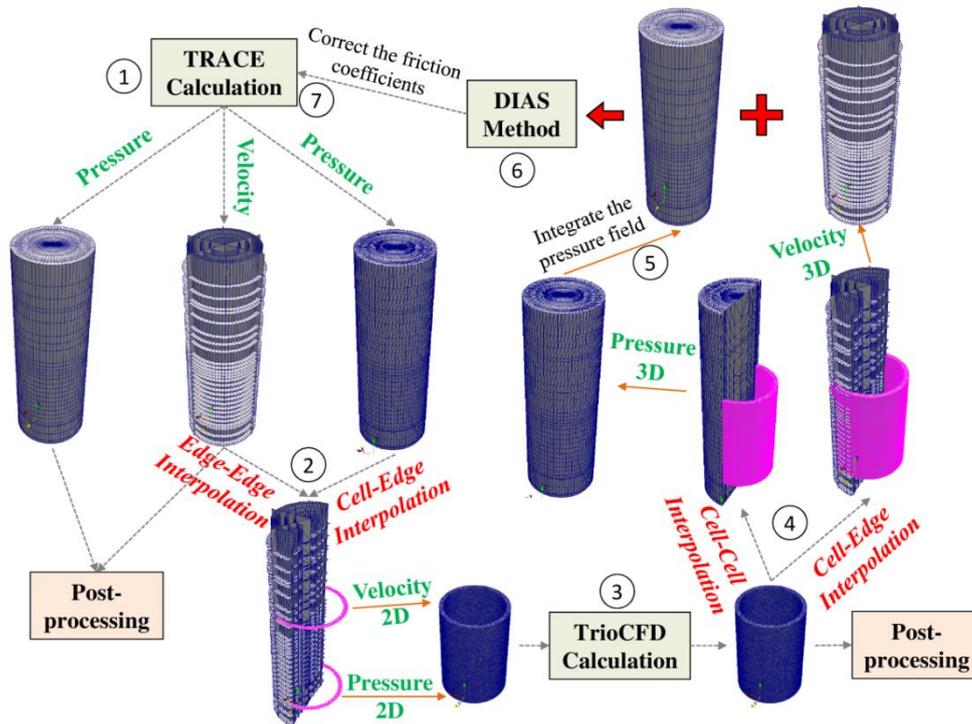


Figure 5-16 – The data flow for hydraulic correlations between TRACE and TrioCFD within the coupling system.

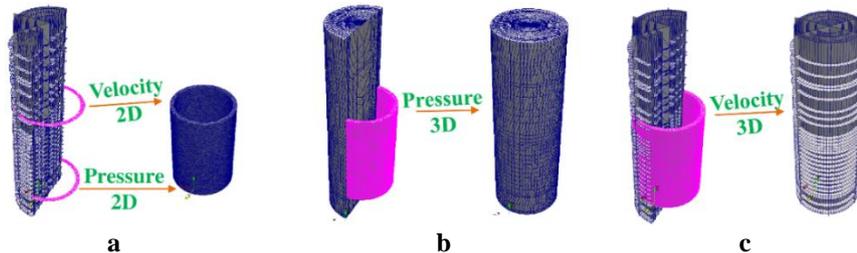


Figure 5-17 – The transferred data for hydraulic correlations between TRACE and TrioCFD within the coupling system.

Figure 5-18 presents the data flow for coolant temperature and boron concentration correlations. The data flow follows the procedures listed below for the coolant temperature and born concentration correlations.

- 1) TRACE runs a time step forward and saves the three-dimensional coolant temperature or born concentration fields in the normal-cell mesh for post-processing and in the tetrahedron-cell mesh for the field mapping in the next step.
- 2) With the help of the MEDCoupling library, insert the upper boundary 2D edge mesh of TrioCFD into the TRACE's edge mesh and derive the desired inlet coolant temperature or born concentration boundary conditions for TrioCFD by performing the cell-edge mesh interpolation. This boundary condition transfer from TRACE to TrioCFD is emphasized in Figure 5-19a.
- 3) With the new updated inlet boundary condition, TrioCFD runs a step forward and write all of the calculated fields in its mesh. Those fields are for both post-processing and field mapping.
- 4) TRACE assembles its tetrahedron-cell mesh to the TrioCFD 3D mesh, derives the desired three-dimensional coolant temperature or born concentration fields, and write them back to the

tetrahedron-cell mesh. This boundary condition transfer from TrioCFD to TRACE is presented in Figure 5-19b.

- 5) Since the tetrahedron-cell mesh does not represent the real TRACE cells, the coolant temperature or born concentration fields in it is first integrated and then written to the normal-cell mesh.
- 6) The coolant temperature or born concentration field is used by the DIAS method to correct the corresponding fields in TRACE at each cell within the entire overlapped domain. Please refer to Chapter 12 for the working principles of the DIAS-method.
- 7) With the updated parameters, TRACE moves a step forward and generates the new fields.

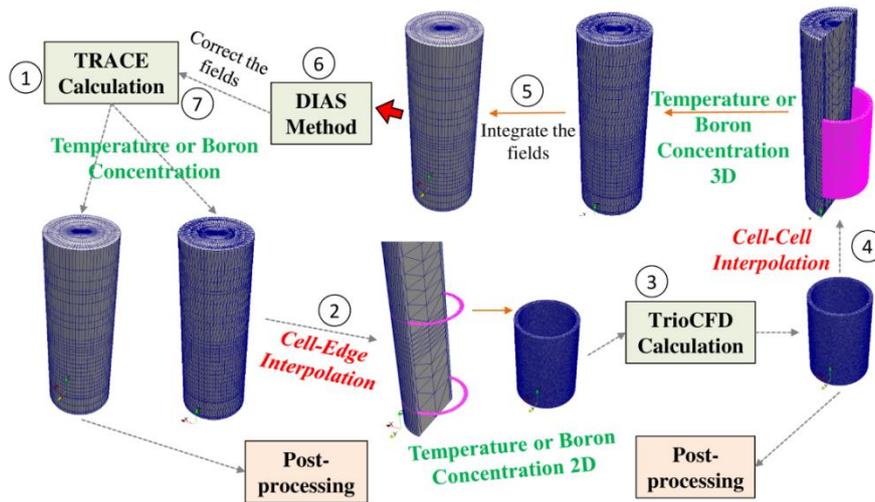


Figure 5-18 – The data flow for coolant temperature and boron concentration correlations between TRACE and TrioCFD within the coupling system.

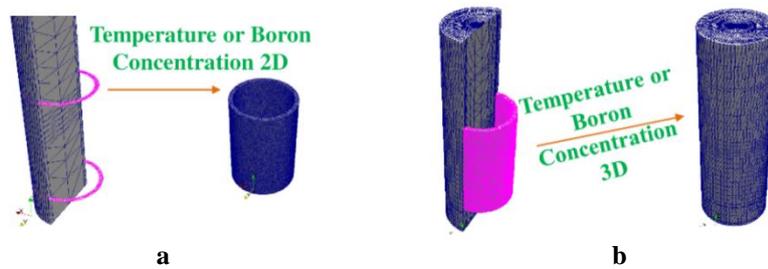


Figure 5-19 – The transferred data for coolant temperature and boron concentration correlations between TRACE and TrioCFD within the coupling system.

At the end of this chapter, it is worth to note that the TRACE/SCF based on ICoCo in SALOME also applies the domain overlapping method with 3D correlations, and the DIAS method as well.

### 5.2.3 The explicit temporal coupling of TRACE/TrioCFD

According to the discussion in the previous chapters and sections, the ICoCo-based multi-scale coupled code TRACE/TrioCFD is a server-client system coordinated by a C++ supervisor in a parallel manner. The mesh and fields mapping and interpolation are handled by the third-party mesh-field – MEDCoupling library, which is closely related to SALOME and it is an in-built module of TrioCFD. The data synchronization during the time advancement between the involved solvers is based on an explicit approach. The overview of the explicit temporal coupling scheme for TRACE/TrioCFD based

on ICoCo is displayed in Figure 5-20. The calculation procedure coordinated by the C++ supervisor is described hereafter:

- 1) The supervisor launches both TRACE and TrioCFD.
- 2) The two codes read in their own input file and do the initialization at the same time. The MPI setting up is finalized if necessary.
- 3) The two codes run into their own time step loops and they calculate first their own time step size. Then, the supervisor gathers the two-time step size, selects the smaller one, and sends the public time step back to the two codes. With that, the two codes reset their current time step.
- 4) The codes inform the supervisor which field they expect from the other code and prepare specified templates for the mesh-interpolation or field-mapping processes. The supervisor sends the desired fields' names to the involved solvers.
- 5) The codes recognize and check the fields' name from the other code and depending on the names, the corresponding fields are extracted from the current codes' memories. The supervisor receives the physical fields and interpolates them with the already-prepared templates thus generating the final fields. Then they are sent to the destinations.
- 6) The codes receive their desired fields from the supervisor and use them to update the conditions for the current time step.
- 7) The codes calculate their current time step and iterate unless the problem time is achieved.
- 8) Once the problem is done, the supervisor terminates both TRACE and TrioCFD runs and then it finalizes the whole execution.

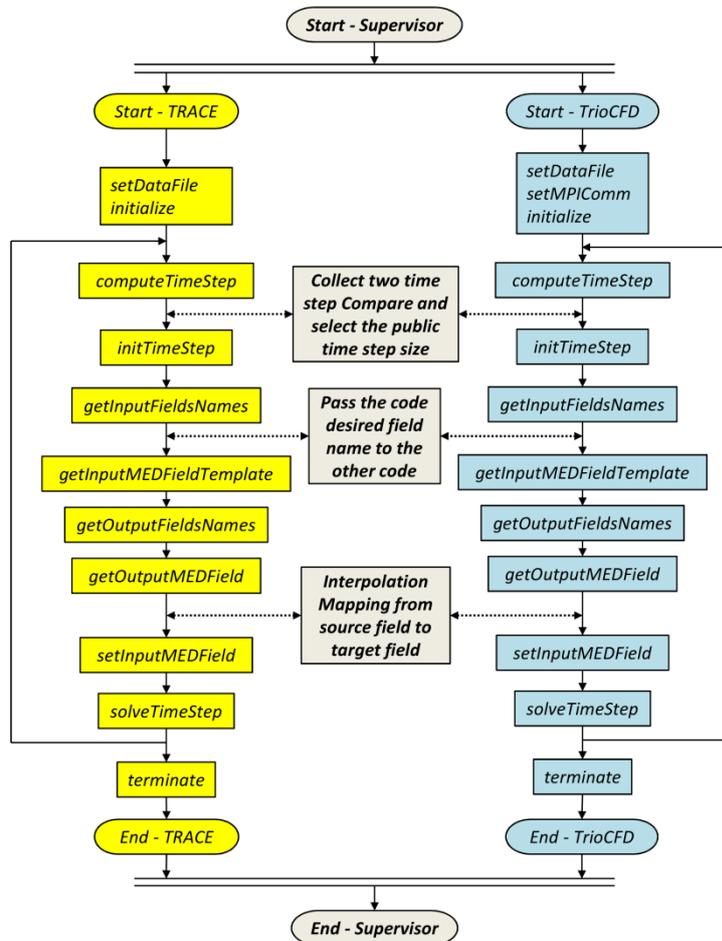


Figure 5-20 – Schematics of the execution of the explicit temporal coupling scheme of TRACE/TrioCFD based on ICoCo.

In this explicit temporal coupling approach, the data transfer is only performed within each time step. The implementation of a semi-implicit temporal coupling is not a significant task because only the supervisor script needs to be re-organized while the ICoCo functions shown in Figure 5-20 remain untouched.

## 5.3 The Dynamic-Implicit-Additional-Source (DIAS) method for domain-overlapping coupling

### 5.3.1 Principles of the DIAS approach

As stated in Chapter 1, there are two spatial coupling approaches available for multi-scale coupling:

- 1) Domain-decomposition.
- 2) Domain-overlapping.

In the case of the domain-overlapping approach, system codes are in charge of the whole NPP model no matter which parts of the NPP are simulated by sub-channel or CFD codes resulting in several overlapped domains. Most of the multi-scale coupled pairs adopt the domain-decomposition method (Appendix A) since the feedbacks as boundary conditions can be introduced from the source code to the target code in a straightforward manner and a limited programming effort is desired. However, this method requires the decomposition of the original integrated system model into several sub-bodies. In order to keep the integrity of the system model as well as to improve the robustness of the coupled codes, the domain-overlapping approach was put forward and successfully implemented in recent years. In most recent implementations, the data transfer is only two dimensional and the three-dimensional data transfer in the whole spatial overlapped domains is not considered. In the most advanced implementations, the three-dimensional spatial fields of the CFD-code are used to correlate the system codes' solution in the entire overlapped domains.

This 3D correlation could introduce more CFD fine data to the system code than the 2D correlation occurring only at the domain interfaces. Thus more reliable results are expected from the system code for the following time step. In turn, more reliable boundary conditions are provided from the system code to the CFD code for the next time step, and with the refined boundary conditions, the CFD code is capable to predict more accurate results. Another advantage of the 3D correlation for domain overlapping coupling is that it avoids the sudden fields change over interfaces between the codes' thermal-hydraulic domains since the correlations take place at the entire 3D overlapped domain of the system code in a smooth manner rather than on the 2D interface introducing a discontinuity. This characteristic is potentially beneficial for the numerical solution, which was briefly inspected in the doctoral thesis of Grunloh (Grunloh, 2016).

Now, in case of the coupled code TRACE/TrioCFD, the three-dimensional field manipulating process is handled by the MEDCoupling library. Hence, the essential challenge is how to use three-dimensional data to correlate TRACE-solution in the entire overlapped domain. For this purpose, the Dynamic-Implicit-Additional-Source (DIAS) method has been developed. The DIAS approach is used to correlate four system thermal-hydraulic parameters:

- 1) Pressure drop.
- 2) Coolant velocity.
- 3) Coolant temperature.
- 4) Boron concentration.

With the correlations, the system code is able to predict thermal-hydraulic parameters with TrioCFD in the entire overlapped domain. It is to note that the high resolved fields of TRACE benefiting from the correlation from TrioCFD in the 3 dimensional overlapped domain are not taken for the final

analysis because the fine CFD results are already available. The key motivation is to let TRACE produce more optimized boundary conditions for TrioCFD. This is also one of the motivations to develop the bi-directional coupling (two-way coupling or on-line coupling in some work) instead of the unidirectional coupling (one-way coupling or off-line coupling in some work).

### **a) The correlations to TRACE pressure drop and coolant velocity**

The basic idea for using CFD data to improve the hydraulic dynamic prediction capability of the system code is that the latter one applies abundant empirical correlations and related models in the simulation leading to limited application scenarios and sometimes even poor results. As an example of a 1D components in TRACE, the governing motion equation is Equation 2-2 whose contributions is a sum of different pressure drops arising from different sources as shown in Equation 5-1. Similar kinds of pressure drops can also be derived for sub-channel and CFD codes, Equation 5-2. The “sub” means sub-channel code. From left to right, they are:

- a)  $\Delta P_{acc}$  – The acceleration pressure drop or inertial pressure drop originating from the variation of coolant velocity and density along the problem time.
- b)  $\Delta P_{con}$  – The convective pressure drop which comes from the changes of the momentum in the cell volume.
- c)  $-\Delta P_{tot}$  – The static pressure difference which is the driving force of the flow.
- d)  $\Delta P_{fric}$  – The friction pressure drop which is caused by the friction between the fluid and the solid structures.

$$\frac{V_{j+\frac{1}{2}}^{n+1} - V_{j+\frac{1}{2}}^n}{\Delta t} + V_{j+\frac{1}{2}}^n \cdot \frac{\partial \tilde{V}^{n+1}}{\partial x} \Big|_{j+\frac{1}{2}} + \frac{1}{\langle \rho \rangle_{j+\frac{1}{2}}^n} \cdot \frac{\bar{P}_{j+1}^{n+1} - \bar{P}_j^{n+1}}{\Delta x} + K_{j+\frac{1}{2}}^n \cdot \left( 2V_{j+\frac{1}{2}}^{n+1} - V_{j+\frac{1}{2}}^n \right) \cdot \left| V_{j+\frac{1}{2}}^n \right| = 0;$$

$$\Delta P_{acc}^{TRACE} + \Delta P_{con}^{TRACE} + (-\Delta P_{tot}^{TRACE}) + \Delta P_{fric}^{TRACE} = 0. \quad \text{Equation 5-1}$$

$$\Delta P_{acc}^{Sub/CFD} + \Delta P_{con}^{Sub/CFD} + (-\Delta P_{tot}^{Sub/CFD}) + \Delta P_{fric}^{Sub/CFD} = 0. \quad \text{Equation 5-2}$$

Normally, all of the four pressure drop terms from sub-channel or CFD codes were assumed to be more precise than that from the system codes as e.g. TRACE. Though the superposition principle is not valid for the Navier-Stokes-Equation (a non-linear system), it works for the linearized discretized momentum equation. This is the initial motivation of suppressing the calculations of  $\Delta P_{con}$  and  $\Delta P_{fric}$  in the system codes while let CFD do the work, just as the RELAP5/FLUENT and TRACE/STAR-CCM+ coupled code. In this approach, the sub-channel code and CFD code are used to let TRACE produce more realistic flow distributions in the entire overlapped domains.

The convective and friction pressure drops in Equation 5-1 are replaced with that from the fine results, as Equation 5-3 shows (Though the superposition principle is not valid for the Navier-Stokes-Equation as a non-linear system, it works for the linearized discretized momentum equation). The reason why the acceleration pressure drop can't be erased is because of the ineradicable time term which explicitly binds the current and next time steps. In order to counteract  $\Delta P_{acc}^{TRACE}$ , the corresponding acceleration pressure drop  $\Delta P_{acc}^{Sub/CFD}$  is eliminated from the total static pressure drop  $\Delta P_{tot}^{Sub/CFD}$  of sub-channel or CFD codes. The new added two pressure difference are further integrated into one artificial friction pressure drop  $\Delta P_{fric, fake}^{TRACE} = \Delta P_{con}^{Sub/CFD} + \Delta P_{fric}^{Sub/CFD} = \Delta P_{tot}^{Sub/CFD} - \Delta P_{acc}^{Sub/CFD}$  by imposing a fresh friction coefficient  $C_{j+\frac{1}{2}, fake}^n$  which equals  $\Delta x \cdot \rho \cdot K_{j+\frac{1}{2}, fake}^n$ . This is exactly the fundamental of the Inertial Domain Overlapping (IDO) and Stabilizer Inertial Domain Overlapping (SIDO) methods

applied in TRACE/STAR-CCM+. This modifies the TRACE- equation describing the momentum transfers to Equation 5-4.

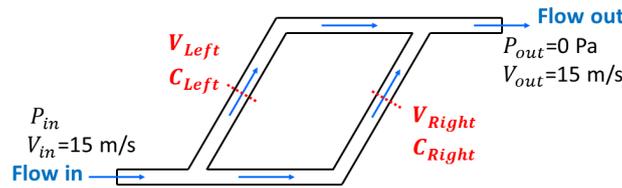
$$\begin{aligned}\Delta P_{tot}^{TRACE} &= \Delta P_{acc}^{TRACE} + \Delta P_{con}^{Sub/CFD} + \Delta P_{fric}^{Sub/CFD} \\ &= \Delta P_{acc}^{TRACE} + (\Delta P_{tot}^{Sub/CFD} - \Delta P_{acc}^{Sub/CFD}).\end{aligned}\quad \text{Equation 5-3}$$

$$\frac{V_{j+\frac{1}{2}}^{n+1} - V_{j+\frac{1}{2}}^n}{\Delta t} + \frac{1}{\langle \rho \rangle_{j+\frac{1}{2}}^n} \cdot \frac{\bar{p}_{j+1}^{n+1} - \bar{p}_j^{n+1}}{\Delta x} + \frac{C_{j+\frac{1}{2},fake}^n}{\Delta x \cdot \langle \rho \rangle_{j+\frac{1}{2}}^n} \cdot (2V_{j+\frac{1}{2}}^{n+1} - V_{j+\frac{1}{2}}^n) \cdot |V_{j+\frac{1}{2}}^n| = 0.\quad \text{Equation 5-4}$$

The artificial friction pressure drop  $\Delta P_{fric,fake}^{TRACE}$  in Equation 5-4 is proportional to the product of  $C_{j+\frac{1}{2},fake}^n$  and the square of velocity. Thus the fresh artificial coefficient is calculated by Equation 5-5.

$$C_{j+\frac{1}{2},fake}^n = \frac{\Delta P_{fric,fake}^{TRACE}}{V^2} = \frac{\Delta P_{tot}^{Sub/CFD} - \Delta P_{acc}^{Sub/CFD}}{V^2}.\quad \text{Equation 5-5}$$

With the determined  $\Delta P_{fric,fake}^{TRACE} = \Delta P_{con}^{Sub/CFD} + \Delta P_{fric}^{Sub/CFD} = \Delta P_{tot}^{Sub/CFD} - \Delta P_{acc}^{Sub/CFD}$ , the remaining problem is to evaluate a velocity corresponding to the artificial friction coefficient which reflects the flow's physical resistance parameters. The TRACE velocity -  $V^{TRACE}$  is used in the calculation of  $C_{j+\frac{1}{2},fake}^n$  by the previous coupling work of TRACE/STAR-CCM+. By applying this correlation, the TRACE calculation produces a consistent overall pressure drop with STAR-CCM+, whereas the velocity field is not influenced at all. This is believed to be caused due to the destruction of the pressure – velocity coupling. A flow splitter model which has two flow paths, designated as the left and right legs has been designed as the demonstration of this effect, see Figure 5-21. This is a fully mathematical proof procedure with only principles illustrated thus no further geometry details of the model is presented.



**Figure 5-21 – Sketch of a flow splitter model for demonstration of the coupling relationship of the pressure and velocity.**

Assume the values in the first line of Table 5-3 are calculated by TRACE while the values in the second line come from CFD code (this is a 1D general flow case which doesn't fit sub-channel codes). With another assumption that this is a steady-state simulation, the artificial friction pressure drop -  $\Delta P_{fric,fake}^{TRACE}$  turns out to be 147 Pa. Now, the governing motion equation of TRACE becomes Equation 5-4. According to Equation 5-5, the artificial friction coefficient in TRACE/STAR-CCM+ on the left

leg –  $C_{left}^{corr}$  is calculated to be  $\Delta P_{fric,fake}^{TRACE}/V_{left}^{TRACE^2} = 147/5^2 = 5.88$  and that on the right leg –  $C_{right}^{corr}$  is calculated to be  $\Delta P_{fric,fake}^{TRACE}/V_{right}^{TRACE^2} = 147/10^2 = 1.47$ .

**Table 5-3 – Hydraulic results of TRACE and Sub-channel or CFD for the flow splitter model.**

	$V_{left}$	$V_{right}$	$P_{in}$
TRACE	5 m/s	10 m/s	100 Pa
CFD	7 m/s	8 m/s	147 Pa

With the new correlated friction coefficients on the two legs, there comes up with the equation -  $\Delta P_{total}^{corr,TRACE} = (V_{left}^{corr,TRACE})^2 \cdot 5.88 = (15 - V_{left}^{corr,TRACE})^2 \cdot 1.47$ , where  $\Delta P_{total}^{corr,TRACE}$  is the overall pressure drop of the flow splitter which exactly equals  $\Delta P_{fric,fake}^{TRACE}$ . The  $V_{left}^{corr,TRACE}$  is the to-be-calculated correlated velocity on the left leg. This is a simple quadratic equation with one unknown, in which the  $V_{left}^{corr,TRACE}$  is easily solved to be 5 m/s. Consequently  $V_{right}^{corr,TRACE}$  is 10 m/s and the overall pressure drop is 147 Pa. It is found that if the TRACE velocity is applied in Equation 5-5, although the correlated overall pressure drop of TRACE could perfectly match the CFD data, the correlated velocities on the two legs show no significant difference with that of TRACE standalone, confirming the destruction of the pressure and velocity coupling relationship. However, to be more precise, the coupling relationship of the pressure and velocity is never disrupted since this kind of velocity distribution as well as the overall pressure drop fit the total inlet velocity and the correlated artificial friction coefficients uniquely, which is reflected in the solution of the quadratic equation with one unknown. Instead of that, the real trap is the TRACE velocity used in Equation 5-5.

For a real flow in a determined geometry (all physical parameters are fixed e.g. the structure shape and the wall roughness, etc.) with determined outlet pressure and inlet flow velocity, the flow conditions e.g. the velocity and pressure fields are unique for most cases. In other words, for a determined friction coefficient pairs in the flow splitter model, the velocity distribution in the two legs and the overall pressure drop are unique. Suppose the CFD solution strictly follows the real physical process and evaluates the right velocity distribution and pressure drop, the friction coefficients in the CFD models must also be identical with the real geometry and flow. In order to ensure that the correlated TRACE can produce consistent velocity distribution and overall pressure drop with the CFD code, the artificial friction coefficients should also be the same with the CFD model. Thus, the velocity in Equation 5-5 should come from the CFD data.

With the CFD velocities, the artificial friction coefficient on the left leg –  $C_{left}^{corr}$  is calculated to be  $\Delta P_{fric,fake}^{TRACE}/V_{left}^{CFD^2} = 147/7^2 = 3$  and that on the right leg –  $C_{right}^{corr}$  is calculated to be  $\Delta P_{fric,fake}^{TRACE}/V_{right}^{CFD^2} = 147/8^2 = 2.3$ . Then the quadratic equation with one unknown -  $\Delta P_{total}^{corr,TRACE} = (V_{left}^{corr,TRACE})^2 \cdot 3 = (15 - V_{left}^{corr,TRACE})^2 \cdot 2.3$  establish. By solving this equation,  $V_{left}^{corr,TRACE}$  is 7 m/s,  $V_{right}^{corr,TRACE}$  is 8 m/s and the overall pressure drop 147 Pa. This result set perfectly matches the CFD-data by producing the right overall pressure drop and additionally the right velocity distribution. As a consequence, the VMF developed in TRACE/STAR-CCM+ which artificially force the correlated velocity distribution consistent with the CFD is unnecessary since the flow automatically adjust its distribution between the two legs with the right friction coefficients naturally.

The velocity for the current time step (the  $n^{\text{th}}$  temporal layer) from CFD –  $V_{left,right}^{CFD,n}$  is used to calculate the artificial friction coefficient on the current temporal layer –  $C_{left,right}^{corr,n}$  for velocity and pressure on the next temporal layer –  $V_{left,right}^{corr,n+1}$  and  $P_{total}^{corr,n+1}$ . Normally, this is the reason causing one system simulation step lagged compared to the CFD solution. However, such kind of treatment is common in TRACE e.g. the heat transfer coefficients for the next time step are calculated based on the

data at the current time step to simplify the implementation of the Stability-Enhancing Two-Step (SETS) method, though bounded instabilities would occur during calculations. This kind of treatment is also beneficial for the linearization of the motion equation, known as  $C_{j+\frac{1}{2},fake}^n$  in Equation 5-4.

Moreover, the correlation covers the entire SETS solution process since the new friction coefficient is updated by CFD data before the calculation of the stabilizer velocity in Equation 2-1.

The hydraulic correlation present in this paper can be extended to cases with multi-parallel flow paths e.g. a flow splitter having three or more legs since the to-be-solved equation set is always closed. Furthermore, the same algorithm is also available for the 3D VESSEL component in TRACE which can be treated as a multi-flow-path geometry. This is extremely important for the KIT project of coupling TRACE and TrioCFD, where the overlapped domain is located exactly in the VESSEL component. Another highlight of this method is that the hydraulic correlations to TRACE are not only for the overall pressure drop and the macro velocity distribution, but also to correlate the detailed pressure and velocity distributed field in the whole overlapped domains. This feature will be demonstrated later on.

The hydraulic correlation approach presented in this chapter is similar to the traditional Additional Source Term Method (ASTM). In ASTM, the difference between the right and to-be-correlated values are directly appended to the motion equation and thus forcing the correlated values to be equal than the expected data. But the DIAS method does not explicitly introduce the difference to the motion equation. It implicitly “guide” the system code to approximate the CFD-results dynamical along with the problem timeline.

Due to the fact that the faces are separated into three groups storing the velocities in three directions, the DIAS correlations performed on those overlapped edges should also reflect the corresponding three directions. Moreover, for each of the three correlation groups, the most crucial problem is to ensure the correct calculation for both the forward and backflow. This problem is also carefully managed by TRACE. Take the momentum equation for single-phase one-dimensional flow - Equation 2-2 for instance, the friction loss coefficient –  $K$  is calculated from the Churchill empirical correlation and it is always positive, with which the pressure drop caused by the friction loss is calculated to be  $KV/V$ . For forward flow, the velocity is positive and the friction pressure drop is also positive. While for counter-flow, the velocity is negative and thus the calculated friction pressure drop is also negative. Now, the problem becomes how to define the positive direction for an edge. Normally, for 1D component, the normal vector of edge points from left to right and from down to up. While for the 3D component: 1) radial direction – outward; 2) azimuthal direction - counterclockwise; 3) axial direction – upward. The definition of positive directions comes exactly from the numbering rules of the TRACE cells.

Assume the convective pressure drops in Equation 5-1 and Equation 5-2 are zero, the total pressure drop in Equation 5-5 contains the friction pressure drop. The calculated correlated friction coefficients on all of the three directions should reflect the real physical realities and must be positive, no matter for forward or counter flow. Here, for all of the three directions, both of the friction pressure drop and the coolant velocity are neither all positive or all negative, for forwarding flow and counter flow, respectively. So according to the formulation –  $K=P/(V/V)$ , the calculated  $K$  must be positive. The pressure drop in the formula now includes both the friction and convective pressure drop. Consequently, the correlated  $K$  here is no longer a pure friction coefficient reflecting the real geometry, but is a hybrid parameter and could be either positive or negative now. Details of the mechanism are illustrated by Figure 5-22 along with the following descriptions.

The base model in the figure is the straight pipe (black lines wrapped) and the other four boundary groups (green, red, blue, and orange lines) are used to represent the converging and diverging pipes. For both flow conditions, the correlated pressure drop across the  $j+1/2$  edge is calculated by  $P_{j+1/2,corre} = P_{j+1/2,conv} + K_{j+1/2,trace}V_{j+1/2}/|V_{j+1/2}|$ , and the correlated friction coefficient calculated by  $K_{j+1/2,corre} = P_{j+1/2,corre} / (V_{j+1/2}/|V_{j+1/2}|)$ . The convective pressure drops  $P_{j+1/2, conv}$  are both zero and the acceleration pressure drop is neglected because this is assumed a steady flow.

As to the forward flow, the correlated pressure  $P_{j+1/2,corre}$  comes only from the friction pressure drop  $K_{j+1/2,trace}V_{j+1/2}/|V_{j+1/2}|$  which is positive and the calculated correlated friction coefficient  $K_{j+1/2,corre}$  must be positive because the coolant velocity is also positive. For the orange lines wrapped geometry which is actually a converging pipe, the convective pressure drop is always positive thus the calculated correlated friction coefficient  $K_{j+1/2,corre}$  must be greater than zero. Assume the red lines represent a special diverging pipe whose convective pressure drop is right the opposite number of the friction pressure drop, the calculated correlated friction coefficient  $K_{j+1/2,corre}$  is thus right zero. For the blue lines wrapped geometry whose diverging rate is smaller than the red, the absolute value of the convective pressure drop is smaller than the friction pressure drop thus the calculated correlated friction coefficient  $K_{j+1/2,corre}$  is greater than zero. For the green lines wrapped geometry whose diverging rate is larger than the red, the absolute value of the convective pressure drop is larger than the friction pressure drop thus the calculated correlated friction coefficient  $K_{j+1/2,corre}$  is less than zero. Apparently, the so-called correlated friction coefficient for forwarding flow is beyond the real physical significance of the friction coefficient which should always be positive.

As to the backward flow in the straight pipe, the correlated pressure  $P_{j+1/2,corre}$  comes only from the friction pressure drop  $K_{j+1/2,trace}V_{j+1/2}/|V_{j+1/2}|$  which is always negative and the calculated correlated friction coefficient  $K_{j+1/2,corre}$  must be positive because the coolant velocity is also negative. For the orange lines wrapped geometry which is actually a converging pipe under the counter flow condition, the convective pressure drop is always negative thus the calculated correlated friction coefficient  $K_{j+1/2,corre}$  must be positive. Assume the red lines represent a special diverging pipe whose convective pressure drop is right the opposite number of the negative friction pressure drop, the calculated correlated friction coefficient  $K_{j+1/2,corre}$  is thus right zero. For the blue lines wrapped geometry whose diverging rate is smaller than the red, the positive convective pressure drop is smaller than the absolute value of the negative friction pressure drop thus the calculated correlated friction coefficient  $K_{j+1/2,corre}$  is greater than zero. For the green lines wrapped geometry whose diverging rate is larger than the red, the positive convective pressure drop is larger than the absolute value of the negative friction pressure drop thus the calculated correlated friction coefficient  $K_{j+1/2,corre}$  is less than zero. Apparently, the so-called correlated friction coefficient for counter flow is beyond the real physical significance of the friction coefficient which should always be positive.

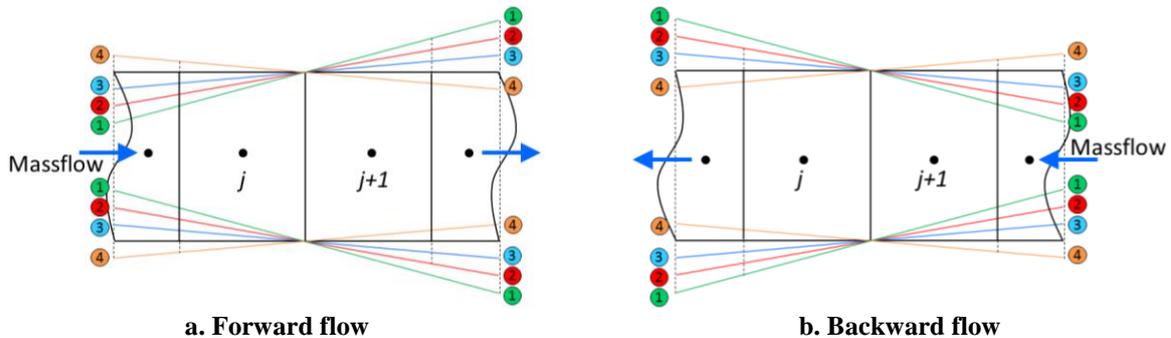
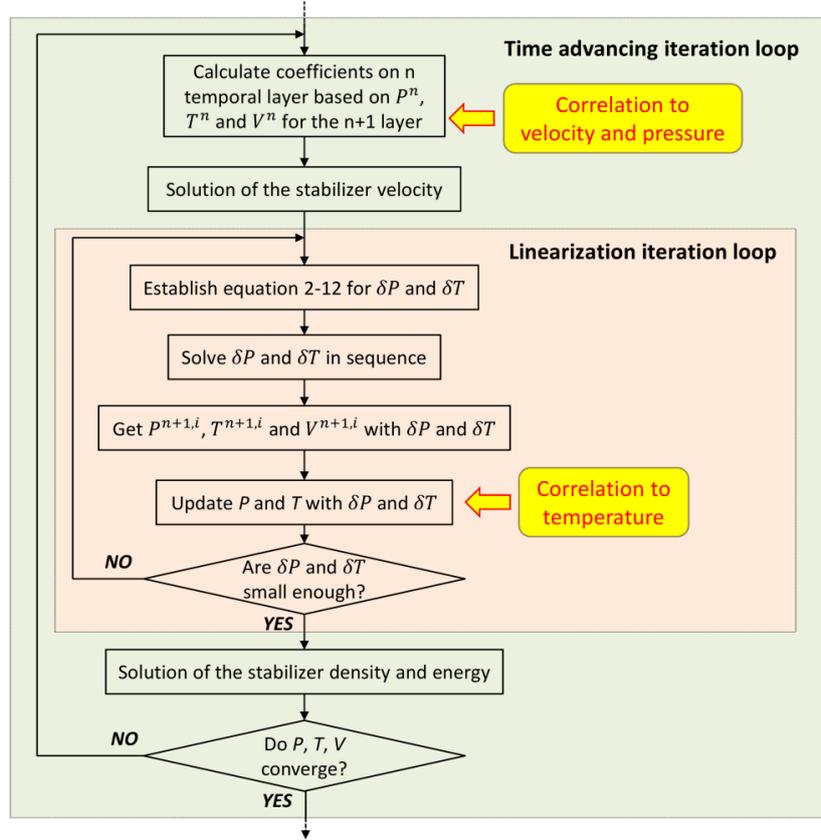


Figure 5-22 – The typical 1D single-phase flow model demonstrating the calculation of the correlated friction coefficient –  $K$  under forwarding and backward flow conditions in diverging and reducing pipes.

**b) The correlations to TRACE coolant temperature**

Different from the velocity and pressure correlations which modify the friction coefficient within the time advancing iteration loop, the correlation to coolant temperature takes place deep in the linearization iteration loop which is the inner loop inside a single time step calculation. The schematic of the TRACE numeric containing the two iteration loops is shown in Figure 5-23.



**Figure 5-23 – Schematics of the solution flow in TRACE within each time step. The temperature correlation happens in the linearization iteration loop.**

There in the steps, where the temperature and the hydraulic correlations take place are indicated. It could be figured out that the artificial friction coefficient is correlated as the prior step before the launching of the SETS process and the temperature correlation occurs within the linearization iteration loop which strictly represents the semi-implicit step in the solution procedures. From the overall perspective, the new pressure and temperature got from the inner loop are used to solve the final stabilizer continuity and energy equations for the final density and energy at the current completed time step. The final pressure and temperature is the result of a further solution of the state equations based on the final density and energy. However, the computation has been never performed in order to reduce the simulation resource burden. Normally, the pressure and temperature got from the semi-implicit step are good enough for updating related coefficients for the next time step.

It can be inferred from the second line of Equation 2-12 that the temperature variation ( $\delta T$ ) for a linearization step is the function of the sole pressure variation ( $\delta P$ ), indicating an simple algebraic equation set in the form of Equation 5-6 which is directly solved. Here the “ $a$ ,  $b$ ” represent the known coefficients in the equations. The superscript  $i$  represents the step counter for the inner loop and the subscript  $j$  stands for the cell number in the model mesh. The temperature for the current linearization step is calculated by Equation 5-7. The tilde above the temperature indicates this is a variable in the semi-implicit step which is the same with that in Equation 2-10 and Equation 2-11.

$$\begin{bmatrix} \delta T_1^i \\ \vdots \\ \delta T_j^i \end{bmatrix} = \begin{bmatrix} a_{1,1}^i & \cdots & a_{1,j}^i \\ \vdots & \ddots & \vdots \\ a_{j,1}^i & \cdots & a_{j,j}^i \end{bmatrix} \cdot \begin{bmatrix} \delta P_1^i \\ \vdots \\ \delta P_j^i \end{bmatrix} + \begin{bmatrix} b_1^i \\ \vdots \\ b_j^i \end{bmatrix}. \quad \text{Equation 5-6}$$

$$\begin{bmatrix} \tilde{T}_1^{n+1,i+1} \\ \vdots \\ \tilde{T}_j^{n+1,i+1} \end{bmatrix} = \begin{bmatrix} \tilde{T}_1^{n+1,i} \\ \vdots \\ \tilde{T}_j^{n+1,i} \end{bmatrix} + \begin{bmatrix} \delta T_1^i \\ \vdots \\ \delta T_j^i \end{bmatrix}. \quad \text{Equation 5-7}$$

The goal of the correlation is letting the final coolant temperature at the current time step equal the sub-channel or CFD results. As the basic idea, the operation can be performed once the temperature is fixed after the inner iteration to force the fresh temperature equal that from other codes. However, this is an inconditionate manipulation and the information of the right temperature fields is forbidden to propagate during the inner iteration where the coefficients e.g.  $a$  in the matrix are updated for each inner step based on the calculated temperature for the current inner step. Taking this effect into consideration, the temperature correlation is moved forward in order to produce consistent temperature fields with sub-channel or CFD codes for each inner step calculation.

The correlation works in a rather simple way by replacing the calculated temperature  $\tilde{T}_{TRACE}^{n+1,i+1}$  on each inner iteration step with the sub-channel or CFD results ( $T_{Sub,CFD}^n$ ) straightforward. Nevertheless, the hidden rationale could be the addition of a virtual source ( $S^i$ ) as shown in Equation 5-8. The source can be perceived as an auto-self-adapted and dynamically calculated source set according to the difference between  $T_{Sub,CFD}^n$  and  $\tilde{T}_{TRACE}^{n+1,i+1}$ , though nothing is calculated here. The direct replacement of  $\tilde{T}_{TRACE}^{n+1,i+1}$  by  $T_{Sub,CFD}^n$  can be regarded as the results of a dynamic and implicit regulation by the virtual source, which reflects the core concept of the Dynamic-Implicit-Additional-Source (DIAS) method. It is worth to note here that the coolant temperature field in the entire overlapped domain identical with that from sub-channel or CFD codes, which is similar to the hydraulic correlations.

$$\begin{aligned} \begin{bmatrix} \tilde{T}_{1,TRACE}^{corr,n+1,i+1} \\ \vdots \\ \tilde{T}_{j,TRACE}^{corr,n+1,i+1} \end{bmatrix} &= \begin{bmatrix} T_{1,Sub,CFD}^n \\ \vdots \\ T_{j,Sub,CFD}^n \end{bmatrix} = \begin{bmatrix} \tilde{T}_{1,TRACE}^{n+1,i+1} \\ \vdots \\ \tilde{T}_{j,TRACE}^{n+1,i+1} \end{bmatrix} + \begin{bmatrix} S_1^i \\ \vdots \\ S_j^i \end{bmatrix} \\ &= \begin{bmatrix} \tilde{T}_{1,TRACE}^{n+1,i} \\ \vdots \\ \tilde{T}_{j,TRACE}^{n+1,i} \end{bmatrix} + \begin{bmatrix} a_{1,1}^i & \cdots & a_{1,j}^i \\ \vdots & \ddots & \vdots \\ a_{j,1}^i & \cdots & a_{j,j}^i \end{bmatrix} \cdot \begin{bmatrix} \delta P_1^i \\ \vdots \\ \delta P_j^i \end{bmatrix} + \begin{bmatrix} b_1^i \\ \vdots \\ b_j^i \end{bmatrix} + \begin{bmatrix} S_1^i \\ \vdots \\ S_j^i \end{bmatrix}. \end{aligned} \quad \text{Equation 5-8}$$

### **c) The correlations to TRACE boron dissolution**

The solubility equation is not included in the main governing equation set. It is solved at the end of each time step. A similar approach has been developed for the boron concentration correlation as that to the coolant temperature correlation, which directly replaces the TRACE data with fine results in the overlapped domains. This manipulation is treated as adding a dynamic and implicit additional source term at the right hand of the solubility equation. As a linear system, the solution matrix can be solved directly.

The boron correlation is carried out as the last step of the DIAS approach which is located directly after the solution of final energy and density in the time advancing loop, see **Figure 5-24**. The correlation can force TRACE to reproduce identical boron distributions in the entire overlapped region as the fine data from the other two codes. Moreover, it has to be stated that though the DIAS approach correlates the hydraulic parameters (coolant velocity and pressure), the coolant temperature, and the solute concentration all by appending some kind of dynamic and implicit additional sources to the corresponding equations, the additional source for hydraulic equation is particular since it correlates the coolant velocity and pressure synchronously and “guide” the system code to produce better result

while the additional source to thermal and solubility equations only correlate one parameter and just “force” the system code reproduce consistent fields with that from the sub-channel or CFD codes.

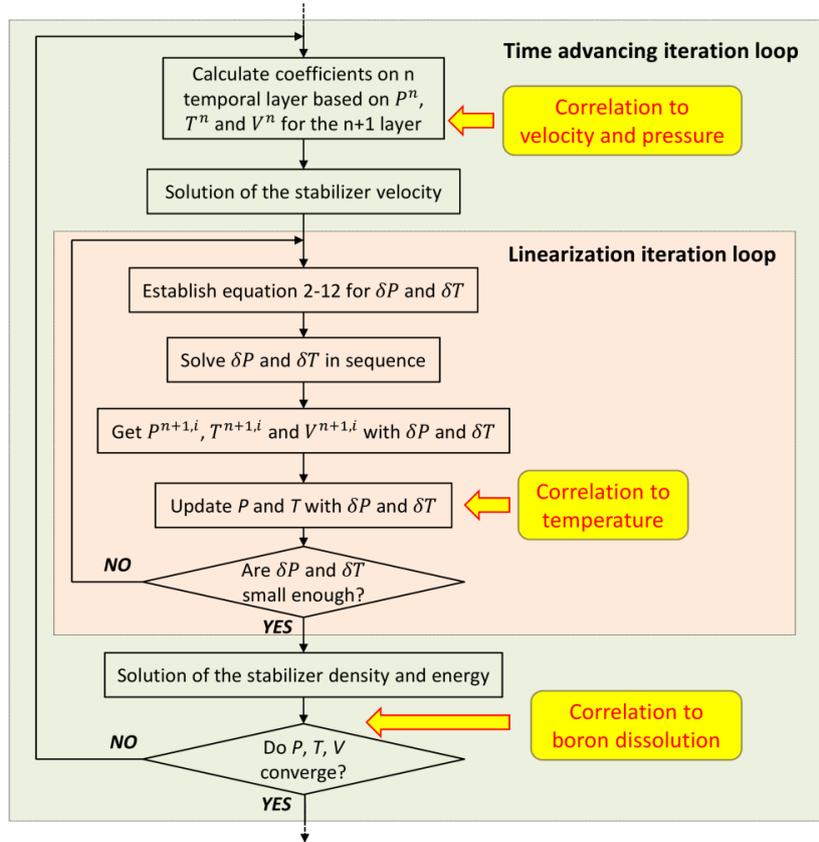


Figure 5-24 – Schematics of the solution flow in TRACE within each time step. The boron concentration correlation happens at the end of time step calculation.

### 5.3.2 Verification of the DIAS Method with a flow splitter case

The DIAS-based correlated physical parameters in TRACE include the pressure field, the coolant velocity field, the coolant temperature field, and the boron concentration field in the whole overlapped regions. Due to the fact that the momentum equation deals with the complex coupling relationship between the pressure and velocity, the DIAS correlation to hydraulic parameters is first verified with a flow splitter case simulated by TRACE and TrioCFD.

It has been revealed in section 2.1.2 that the mathematical models applied by TRACE are not that accurate in calculating the pressure drop and velocity distributions for a flow system, which is demonstrated with a 1D flow splitter model having two parallel legs and two side junction geometries. Here in this section, the same model is adopted to verify the corresponding hydraulic correlations of the DIAS method from a general viewpoint whose conclusion could be flexibly extended to the TRACE 3D VESSEL component. The geometric specification of the model and assumed boundary conditions are illustrated in Figure 5-25.

The model has a flow-in side junction and a flow-out side junction which are among the most critical structures challenging the simulation accuracy. So, in addition to the inspection of overall pressure drop in the flow splitter, the pressure distribution at the side junctions is to be analyzed. Figure 5-26

and Figure 5-27 display the TRACE model and pressure field simulated with TrioCFD as well as the data points on which the physical fields are to be compared.

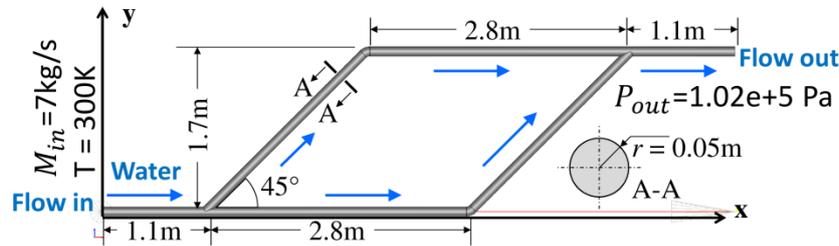


Figure 5-25 – Specification of the flow splitter problem.

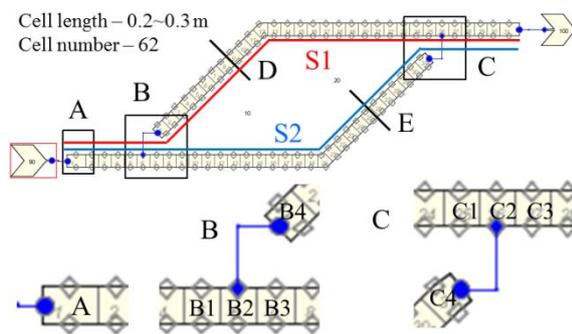


Figure 5-26 – Flow splitter model in TRACE and the positions where the data are extracted.

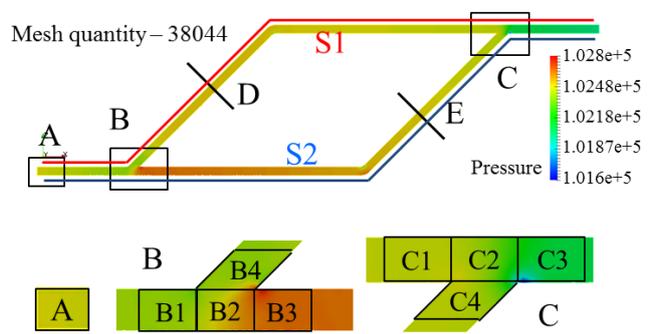


Figure 5-27 – Pressure field in the flow splitter by TrioCFD and the positions where the data are extracted corresponds to that of TRACE.

Table 5-4 gives the simulation results from the original TRACE, the DIAS-correlated TRACE, and TrioCFD, including the pressure difference from the outlet on positions A, B1, B2, B3, B4, C1, C2, C3, C4 and the velocities on D and E. As it is shown in Figure 5-26 and Figure 5-27, the data derivation position from TrioCFD fully corresponds to that in TRACE model. The CFD data in the volumes (A, B1, B2, B3, B4, C1, C2, C3, C4) are averaged for the sake of comparison with TRACE. Table 5-4 gives relative pressures to the outlet pressure. Additionally, the differences of the data from the original TRACE and DIAS-correlated TRACE compared to TrioCFD are also listed in Table 5-4. The TrioCFD pressures are the averaged values corresponded to TRACE cells in the same position. The differences represent the absolute values of the ratio of the difference between TRACE and DIAS-correlated TRACE to TrioCFD results.

Obvious advantages of the DIAS-correlated TRACE compared to the original TRACE are observed by a dramatic reduction in Diff 2 than Diff 1. Despite the consistent overall pressure drop with TrioCFD, the without-DIAS pressure differences on position B2, B3, B4, C3, and C4 are quite large confirming they are indeed the critical structures where the simplified physical model used by TRACE is not adequate. The consequence of the simplified pressure distribution translated into an erroneous poor velocity distribution was produced by TRACE which shows even larger deviation from the CFD results. Significant pressure and velocity refinement are observed when the DIAS correlations are included, where the differences are dramatically reduced by orders of magnitude. The improvements on TRACE prediction capabilities introduced by the DIAS correlations are displayed more vividly in Figure 5-28 and Figure 5-29 which illustrate that the DIAS-correlations are effective in the entire overlapped domains and correlate the full pressure and velocity fields. The flow path – S1 and S2 in Figure 5-28 are marked in Figure 5-26 and Figure 5-27.

**Table 5-4 – Calculated relative pressure to the outlet and the velocities of the original TRACE, the DIAS-correlated TRACE, and TrioCFD on the predefined positions in Figure 5-26 and Figure 5-27.**

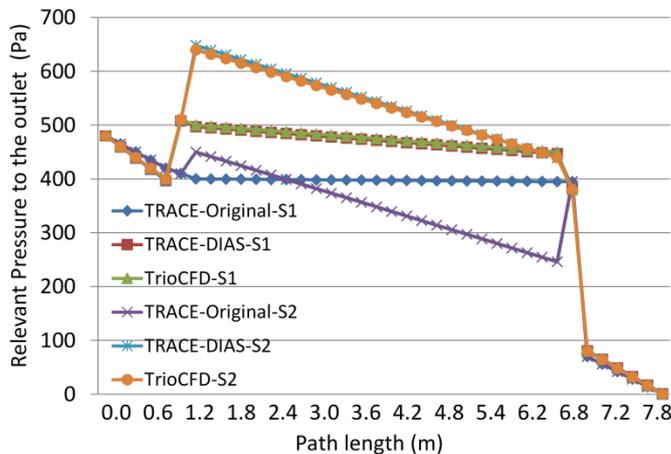
Position	TRACE original	TRACE DIAS-correlated	TrioCFD	Diff 1	Diff 2
A	480 Pa	480 Pa	480 Pa	0%	0%
B1	420 Pa	397 Pa	400 Pa	5%	0.75%
B2	410 Pa	508 Pa	510 Pa	19.6%	0.392%
B3	450 Pa	648 Pa	640 Pa	29.7%	1.25%
B4	400 Pa	497 Pa	500 Pa	20%	0.6%
C1	395 Pa	447 Pa	450 Pa	12.2%	0.667%
C2	395 Pa	384 Pa	380 Pa	3.95%	1.05%
C3	70 Pa	81 Pa	80 Pa	12.5%	1.25%
C4	246 Pa	439 Pa	440 Pa	44.1%	0.227%
D	0.06 m/s	0.33 m/s	0.32 m/s	81.3%	3.13%
E	0.83 m/s	0.56 m/s	0.57 m/s	45.6%	1.75%

$Diff\ 1 = ABS(TRACE\_original - TrioCFD)/TrioCFD \times 100\%$

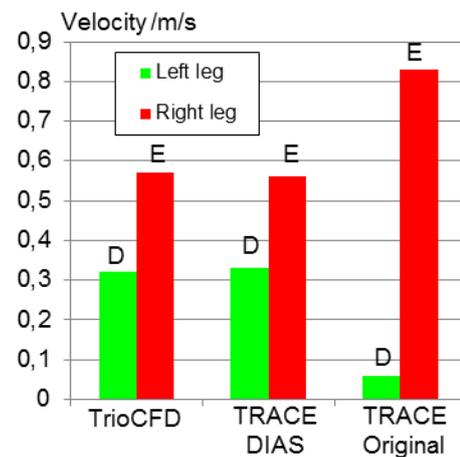
$Diff\ 2 = ABS(TRACE\_DIAS\_correlated - TrioCFD)/TrioCFD \times 100\%$

Data on D and E are coolant velocities while on other positions are relative pressures to outlet pressure.

It is worth mentioning that the edges between two adjacent cells in the TRACE model shown as Figure 5-26 are all equipped with a artificial friction coefficient which is calculated based on the averaged pressures and velocities from TrioCFD. Apparently, in order to derive the averaged pressures and velocities, the fine fields of TrioCFD have to be mapped to TRACE coarse mesh getting the low-resolution fields. Here in this paper, a powerful library – MEDCoupling which is open source and able to manage various mesh and field interpolations has been used for the data translation between TRACE and TrioCFD. This library is normally published together with the European open source platform – SALOME.



**Figure 5-28 – The pressure distribution along the two flow paths simulated by TrioCFD, TRACE with DIAS correlation, and TRACE original.**



**Figure 5-29 – The velocity distribution on the two legs simulated by TrioCFD, TRACE with DIAS correlation, and TRACE original.**

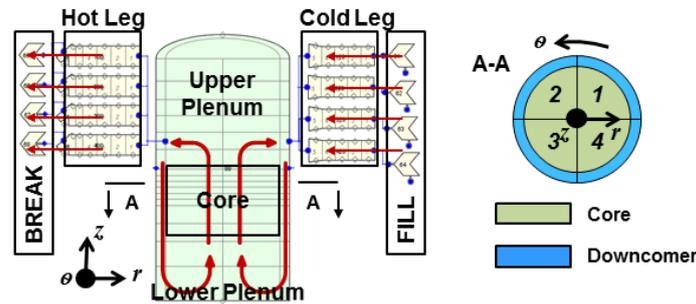
### 5.3.3 Verification of the DIAS method with an academic reactor problem

For the DIAS verification of reactor cases, the coupled code TRACE/TrioCFD-ICoCo causes significant computation intensity due to the wide range of testing cases. In order to perform an

efficient, comprehensive, and pure verification of the DIAS method in the reactor systems, the TRACE/SCF-ICoCo is applied considering the three plain ideas:

- 1) The DIAS method is a universal approach for domain-overlapping coupling, which means it is not only available for coupling of system code with CFD codes but also valid for the coupling of system code with sub-channel codes (TRACE/SCF-ICoCo is based on DIAS method).
- 2) The structure of SCF is simpler, so it is easier to verify and demonstrate the capability of the DIAS method.
- 3) The mesh density of SCF is relatively low and reduces the computation cost during the testing phase.

For the verification of the DIAS-method implemented in the domain-overlapping coupling of TRACE with SCF, an academic problem has been selected. This is the same problem as used in the verification of TRACE/SCF-ECI in section 4.3, see Figure 4-3. Now the TRACE model is shown in Figure 5-30. The 3D RPV model consists of a 3D coarse mesh representation of the RPV including the core. The inlet (cold legs) and outlets (hot legs) boundary conditions of the four-loop plant are represented by a FILL and BREAK-component connected to a PIPE. The mass flow rate and the temperature of the cold legs are defined in the FILL component while the pressure boundary condition is defined in the BREAK-components. In the VESSEL, the coolant flows downward along the downcomer and reverses the direction in the lower plenum to upward flow going through the core and finally reaches the BREAK component through the hot legs. The VESSEL component consists of sixteen axial levels, four azimuthal sectors, and two radial rings, resulting in a total of 128 3D cells. The SCF model has nine rectangular channels and nine levels, totally 81 cells, see Figure 4-3.



**Figure 5-30 – Configuration of the academic reactor model in TRACE for DIAS verification**

All of the four elements of the DIAS method including the correlations to pressure, coolant velocity, coolant temperature, and boron concentration are tested in this section with the reactor model shown in Figure 5-30.

#### **a) Testing of the pressure and velocity correlations**

The DIAS-method is tested for the TRACE/SCF coupling based on the domain-overlapping approach. The adopted sub-channel code SCF leads to an overlapped domain in the core area. For demonstration purposes, the SCF model represented the core by nine parallel channels as shown in Figure 5-31 together with the TRACE core model. The mesh interpolation and data-mapping are managed by the open-source library MEDCoupling embedded in the European open source platform SALOME.

Six cases are simulated in the vessel for overlapped domain coupling. The only variable is the coolant inlet velocity at loop 1 (Figure 5-31b) which ranges from 10 m/s to 20 m/s with 2 m/s intervals, whereas the inlet velocities of the other three loops are fixed to be 10 m/s. Meanwhile, the inlet coolant temperature and the outlet pressures at the four loops are kept identical during the testing phase. The boundary conditions for the six cases are listed in Table 5-5.

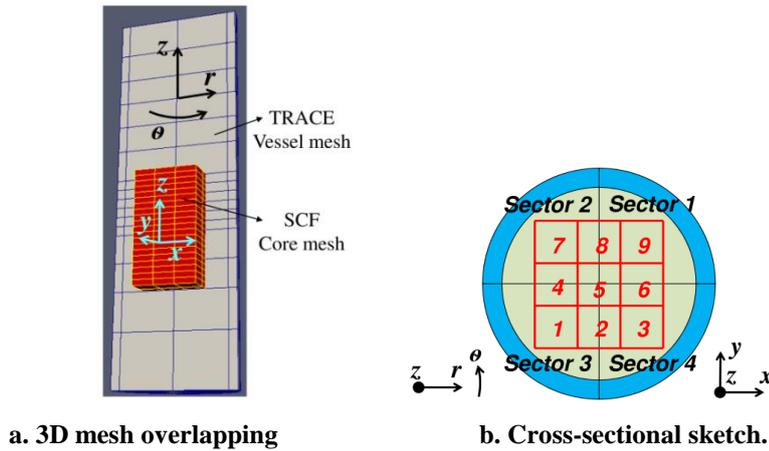


Figure 5-31 – Position relation of TRACE VESSEL and SCF channels.

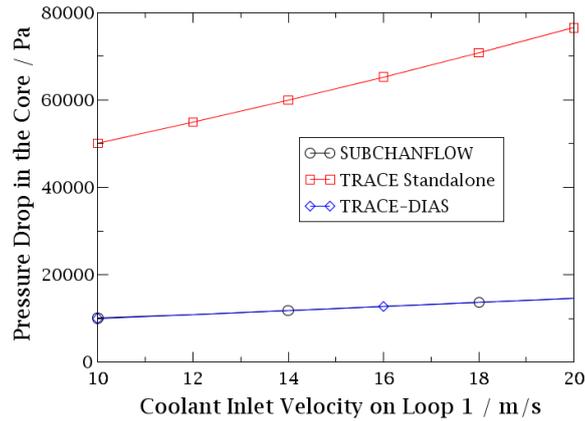
Table 5-5 – Boundary conditions of the test cases for hydraulic correlations.

	Loop 1	10 m/s	12 m/s	14 m/s	16 m/s	18 m/s	20 m/s
<i>Inlet Velocity</i>	Loop 2~4	10 m/s					
<i>Temperature</i>	Loop 1~4	400 K					
<i>Outlet Pressure</i>	Loop 1~4	15.55 MPa					

The increase of the mass flow rate of the loop-1 lead to a non-symmetrical thermal-hydraulic condition for the core i.e. non-symmetrical coolant velocities and pressure distributions within the core is depicted in Table 5-6. There, the total pressure drops predicted by the stand-alone versions of SCF and TRACE are compared to the ones of the coupled solution TRACE/SCF with the DIAS correction. Figure 5-32 displays the three pressure drop curves over the core, from which, it can be concluded that TRACE with DIAS-correlation in TRACE/SCF is able to predict values very close to the ones of SCF. However, it has to be noted that the large pressure difference between SCF and TRACE standalone mainly comes from the artificially reduced resistance coefficients applied in SCF compared to the ones used in TRACE. TRACE assigned artificial friction loss factors on selected levels of the core. It is not caused by the difference between the codes' numerical solutions. The goal of this verification is to show the ability of the DIAS approach applied to the domain-overlapping coupling of TRACE/SCF.

Table 5-6 – Calculated pressure drop (Pa) in the four sectors along the entire core predicted by SCF, TRACE standalone, and TRACE with DIAS correlations.

	Sector	10 m/s	12 m/s	14 m/s	16 m/s	18 m/s	20 m/s
<i>SCF</i>	#1	10046	10917	11811	12734	13685	14665
	#2	10046	10904	11789	12706	13653	14630
	#3	10046	10897	11776	12688	13632	14607
	#4	10046	10904	11789	12705	13652	14629
<i>TRACE Standalone</i>	#1	50041	54999	60149	65494	71039	76786
	#2	50041	54893	59966	65257	70764	76487
	#3	50041	54836	59861	65117	70597	76300
	#4	50041	54892	59963	65253	70759	76481
<i>TRACE DIAS</i>	#1	10028	10938	11835	12759	13712	14692
	#2	10028	10918	11801	12716	13663	14639
	#3	10028	10914	11794	12705	13649	14624
	#4	10028	10918	11801	12716	13663	14639



**Figure 5-32 – Averaged pressure drop over the core predicted by SCF, TRACE standalone, and TRACE with DIAS correlations.**

The quasi-flat pressure drops in the core’s four sectors (Figure 5-31b) are actually the result of the quasi-flat velocity distribution in the core area, which indicates that the coolant is already well mixed in the downcomer and lower plenum before entering the core. Table 5-7 gives the derived velocities predicted by SCF, TRACE standalone, and TRACE/SCF with DIAS correlations. Though the flow velocities at the four core outlets don’t show an obvious difference with each other, the DIAS correlations still try to regulate the flow distribution closer to SCF than TRACE standalone.

**Table 5-7 – Velocity (m/s) in the four sectors of the core predicted by SCF, TRACE standalone, and TRACE with DIAS correlations.**

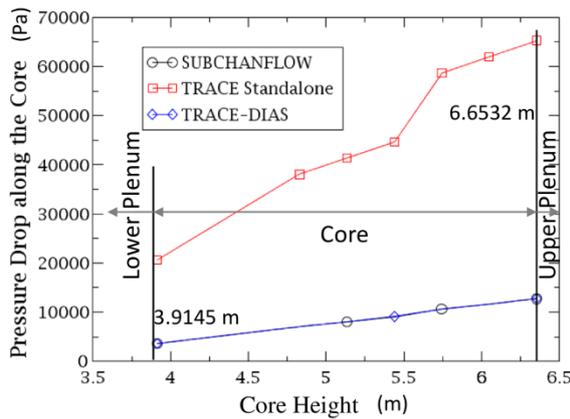
	<i>Sector</i>	<i>10 m/s</i>	<i>12 m/s</i>	<i>14 m/s</i>	<i>16 m/s</i>	<i>18 m/s</i>	<i>20 m/s</i>
<i>SCF</i>	<i>#1</i>	3.9607	4.1679	4.3722	4.5743	4.7782	4.9739
	<i>#2</i>	3.9607	4.1563	4.3534	4.551	4.7488	4.9468
	<i>#3</i>	3.9607	4.1517	4.3452	4.5401	4.7362	4.933
	<i>#4</i>	3.9607	4.156	4.353	4.5504	4.7481	4.9459
<i>TRACE Standalone</i>	<i>#1</i>	3.9595	4.1571	4.3549	4.5528	4.7507	4.9487
	<i>#2</i>	3.9595	4.1576	4.3556	4.5536	4.7516	4.9495
	<i>#3</i>	3.9595	4.1574	4.3553	4.5532	4.7512	4.9491
	<i>#4</i>	3.9595	4.1577	4.3557	4.5537	4.7517	4.9497
<i>TRACE DIAS</i>	<i>#1</i>	3.9595	4.1624	4.3637	4.5639	4.7632	4.9618
	<i>#2</i>	3.9595	4.1571	4.3549	4.5527	4.7506	4.9486
	<i>#3</i>	3.9595	4.1532	4.3483	4.5443	4.741	4.9383
	<i>#4</i>	3.9595	4.157	4.3547	4.5525	4.7504	4.9483

In addition to the ability of the DIAS-approach in correlating TRACE to produce SCF-like total pressure drops and core outlet velocity profiles, the DIAS approach can further regulate the pressure drops and velocity distributions in the whole overlapped domain i.e. in the core. The case with the loop-1 inlet velocity of 16 m/s has been selected to demonstrate this capability and the pressure drops along the core height are given in Table 5-8. Similar to the discussion for the core total pressure drops, the results at the four core sectors are examined individually due to the unsymmetrical inlet boundary conditions at the inlets. However, it can be observed in Table 5-8 that the differences among the four loops are rather small and thus it can be neglected. The comparison of the loops averaged values is sufficient to evaluate the correlations which are shown in Figure 5-33 and Figure 5-34. The former plot shows the accumulated pressure drop from the core bottom to top and the latter displays the pressure difference between two adjacent axial cells. Both of the two plots prove the good recurrence ability of DIAS correlations for SCF data, in the entire overlapped area. It should be noted that the

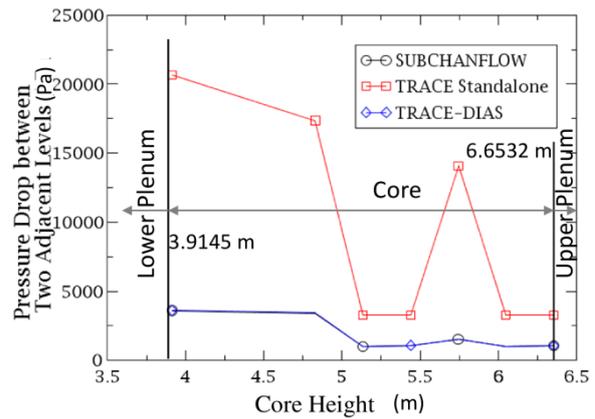
core height represents the relative height from the vessel bottom, and the gravitational components are already removed from the total pressure drops. The large pressure drop predicted by TRACE originates from the artificial friction factor introduced to TRACE model simulating the spacer in the core. SCF also models this configuration but in a more accurate manner.

**Table 5-8 – Calculated pressure drop (Pa) in the four sectors of the core predicted by SCF, TRACE standalone, and TRACE with DIAS correlations when inlet velocity of loop 1 is 16 m/s.**

	Height	3.91 m	4.83 m	5.13 m	5.44 m	5.74 m	6.05 m	6.35 m
<b>SCF</b>	#1	3624	7054	8039	9127	10672	11666	12734
	#2	3597	7022	8007	9096	10642	11637	12706
	#3	3582	7003	7987	9076	10622	11618	12688
	#4	3597	7021	8006	9095	10641	11636	12705
<b>TRACE Standalone</b>	#1	20878	38279	41561	44844	58925	62209	65494
	#2	20628	37996	41294	44592	58665	61962	65257
	#3	20521	37855	41153	44453	58516	61817	65117
	#4	20622	37989	41288	44586	58660	61957	65253
<b>TRACE/SCF DIAS</b>	#1	3652	7086	8071	9160	10702	11694	12759
	#2	3594	7021	8007	9099	10647	11645	12716
	#3	3582	7006	7992	9083	10632	11631	12705
	#4	3593	7020	8005	9097	10646	11644	12716



**Figure 5-33 – Pressure drop along with the core predicted by SCF, TRACE standalone, and TRACE with DIAS correlations when inlet velocity of loop 1 is 16 m/s.**

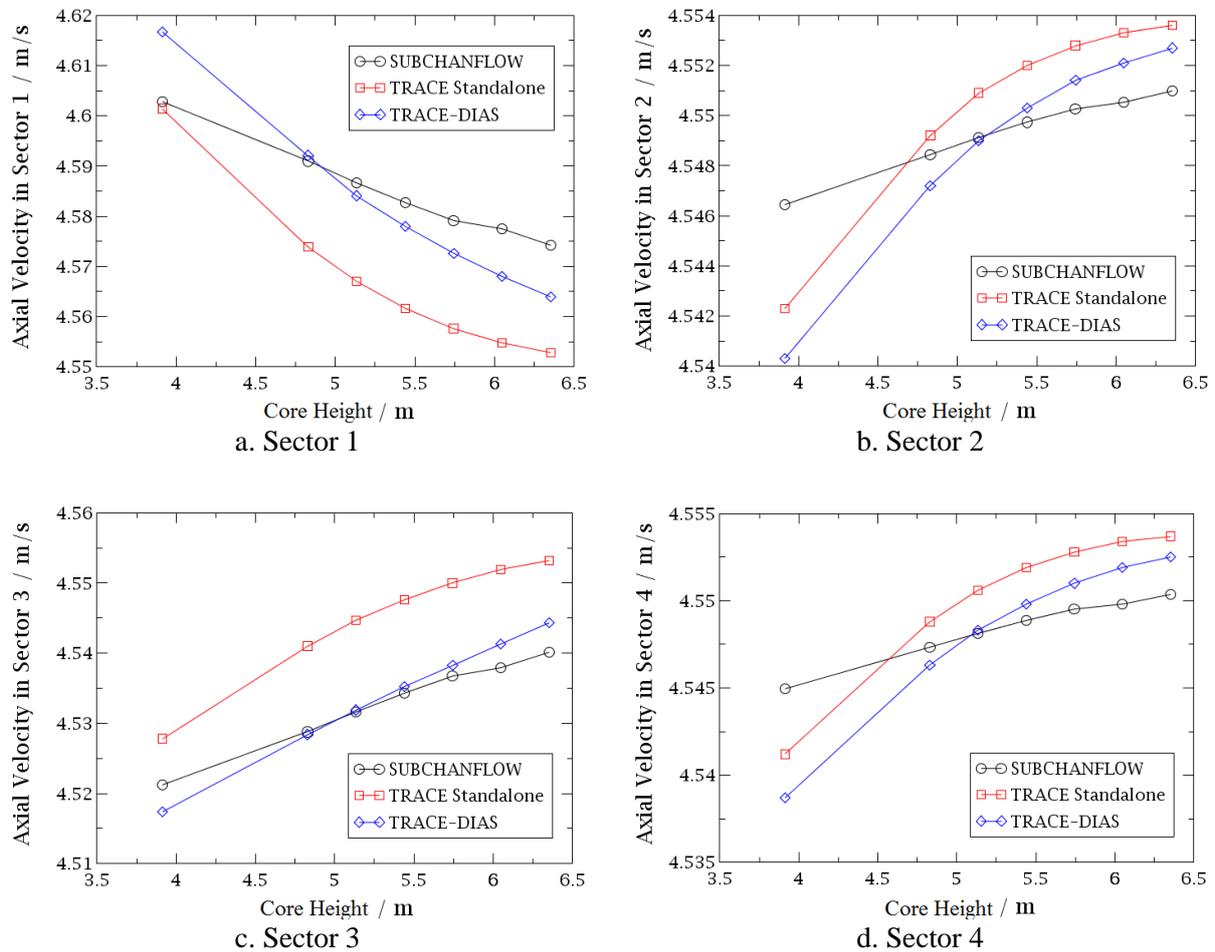


**Figure 5-34 – Pressure drop between two adjacent levels in the core predicted by SCF, TRACE standalone, and TRACE with DIAS correlations when inlet velocity of loop 1 is 16 m/s.**

In order to complete the verification of DIAS correlation on hydraulic parameters, the velocity distribution in the whole overlapped domain is analyzed. The same case selected to demonstrate this capability and the velocity distributions in the core are given in Table 5-9. The values do not show significant difference with each other due to the sufficient mixing in the downcomer and lower plenum. Nevertheless, the DIAS correlations can still regulate the coolant velocity distribution to approximate the SCF results, see Figure 5-35.

**Table 5-9 – Velocity (m/s) distribution in the core predicted by SCF, TRACE standalone, and TRACE with DIAS correlations when inlet velocity of loop 1 is 16 m/s.**

	<i>Height</i>	<i>3.91 m</i>	<i>4.83 m</i>	<i>5.13 m</i>	<i>5.44 m</i>	<i>5.74 m</i>	<i>6.05 m</i>	<i>6.35 m</i>
<i>SCF</i>	<i>#1</i>	4.6028	4.591	4.5867	4.5827	4.5791	4.5775	4.5743
	<i>#2</i>	4.5465	4.5485	4.5491	4.5497	4.5503	4.5505	4.551
	<i>#3</i>	4.5212	4.5288	4.5316	4.5343	4.5368	4.5379	4.5401
	<i>#4</i>	4.545	4.5473	4.5481	4.5489	4.5495	4.5498	4.5504
<i>TRACE Standalone</i>	<i>#1</i>	4.6014	4.5739	4.567	4.5616	4.5576	4.5548	4.5528
	<i>#2</i>	4.5423	4.5492	4.5509	4.552	4.5528	4.5533	4.5536
	<i>#3</i>	4.5278	4.541	4.5447	4.5476	4.55	4.5519	4.5532
	<i>#4</i>	4.5412	4.5488	4.5506	4.5519	4.5528	4.5534	4.5537
<i>TRACE/SCF DIAS</i>	<i>#1</i>	4.6167	4.5921	4.5841	4.578	4.5726	4.568	4.5639
	<i>#2</i>	4.5403	4.5472	4.549	4.5503	4.5514	4.5521	4.5527
	<i>#3</i>	4.5174	4.5284	4.5319	4.5352	4.5383	4.5413	4.5443
	<i>#4</i>	4.5387	4.5463	4.5483	4.5498	4.551	4.5519	4.5525



**Figure 5-35 – Computed mean axial velocity (m/s) distribution in the core four core sectors along the core height as predicted by SCF, TRACE standalone, and TRACE with DIAS correlations when inlet velocity of loop 1 is 16 m/s.**

In the figures, the shapes of TRACE standalone and the DIAS-correlated TRACE in TRACE/SCF are almost the same meaning that the DIAS correlations on hydraulic parameters still embody the

behavior of TRACE, though the source data comes from SCF. The correlations can be treated as the pushing of TRACE standalone curves to SCF curves as integrals thus narrowing the gaps between them. It can be inferred that the DIAS correlations to the pressure and velocity fields in the overlapped core do not strictly follow the results of SCF, which is distinctly different from the perfect correlations on the flow splitter described in section 5.3.2. The most essential reason is that the correlations in the core within the vessel are performed in an “open” system and the correlated part “suspends” in the non-correlated parts, while the correlations to the flow splitter is a “closed” system with fixed boundary conditions in the overlapped domain. Due to this effect, the correlations not only regulate the hydraulic conditions in the overlapped core but also influence the upstream and downstream of the core, as shown in Figure 5-36. This is exactly the reason why the correlated TRACE can’t fully reproduce SCF hydraulic result, and also the reason why the correlated data reflects TRACE behavior.

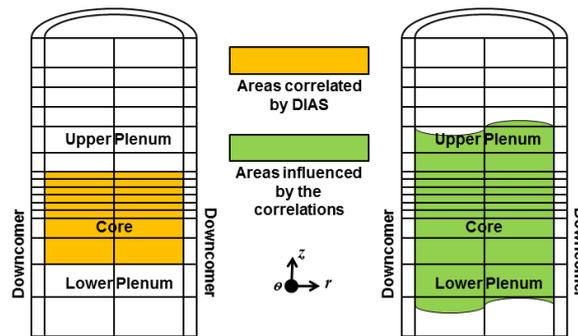


Figure 5-36 – The areas correlated by DIAS and influenced by the correlations.

From what has been discussed above, the capability of the DIAS approach in correlating TRACE to produce SCF-like pressure drops and velocity distributions in the entire overlapped regions is considered to be verified for forced convective flows. Despite the specific simulation tools, this approach supplies a universal idea to use sub-channel or CFD data to correlate system code’s hydraulic performance.

**b) Testing of the temperature correlation**

The 3D temperature field calculated by SCF in the core is used to correlate TRACE prediction when performing a TRACE/SCF simulation. In this section, a total of six cases are investigated. The boundary conditions used therefore are listed in Table 5-10. The single variable is the coolant inlet temperature at loop 1 (Figure 5-31) which ranges from 400K to 500K with 20K intervals, whereas the temperatures at the other three loops are fixed to be 400K. Meanwhile, the inlet coolant velocities and the outlet pressures at the four loops are considered to be identical during the testing phase.

Table 5-10 – Boundary conditions of the test cases for temperature correlation.

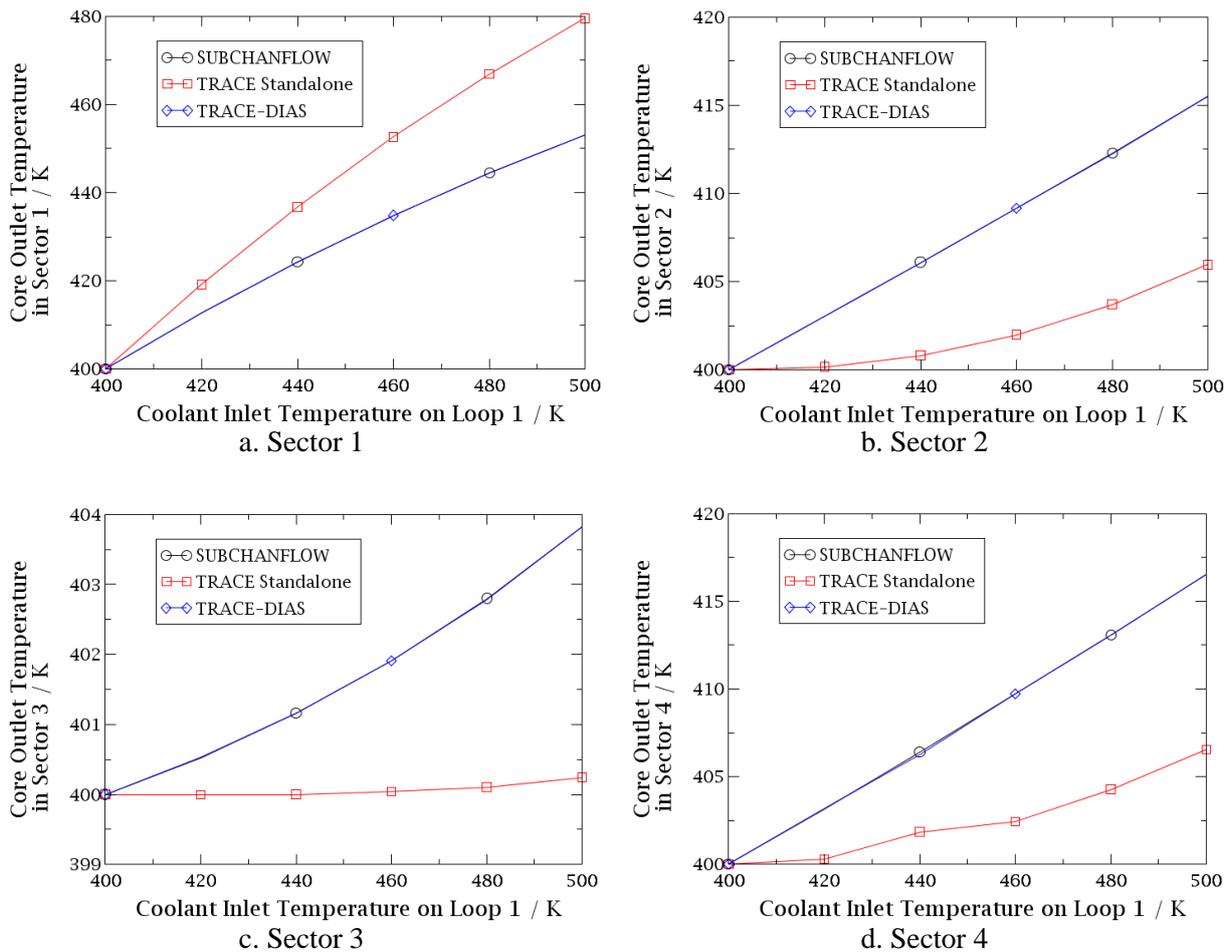
	Loop 1	400K	420K	440K	460K	480K	500K
<i>Inlet Temperature</i>	Loop 2~4	400 K					
<i>Velocity</i>	Loop 1~4	10 m/s					
<i>Outlet Pressure</i>	Loop 1~4	15.55 MPa					

From Table 5-11, it is found that the result of the DIAS-correlated TRACE in TRACE/SCF agrees well with that of SCF, while TRACE standalone exhibits a significant deviation. Furthermore, the more smooth temperature distribution is described by SCF and TRACE/SCF with DIAS correlations than TRACE standalone, indicating that more heat transfer is predicted by the former two codes than the latter one. This phenomenon can be derived from Figure 5-37, where the coolant outlet

temperatures of SCF and TRACE/SCF with DIAS correlations decrease at sector 1 and increase at the other three sectors than TRACE standalone. The higher temperature increase in sector 2 and 4 than sector 3 is because they are directly adjacent to sector 1 while sector 3 locates farthest from sector 1.

**Table 5-11 – Temperature (K) at the core outlets predicted by SCF, TRACE standalone, and TRACE with DIAS correlations.**

	<i>Sector</i>	<i>400K</i>	<i>420K</i>	<i>440K</i>	<i>460K</i>	<i>480K</i>	<i>500K</i>
<i>SCF</i>	<i>#1</i>	400	412.781	424.315	434.862	444.426	453.069
	<i>#2</i>	400	403.054	406.101	409.159	412.284	415.493
	<i>#3</i>	400	400.516	401.159	401.91	402.8	403.822
	<i>#4</i>	400	403.155	406.389	409.716	413.094	416.547
<i>TRACE Standalone</i>	<i>#1</i>	400	419.19	436.744	452.634	466.891	479.526
	<i>#2</i>	400	400.17	400.818	401.989	403.717	405.966
	<i>#3</i>	400	399.993	400	400.04	400.103	400.245
	<i>#4</i>	400	400.277	401.85	402.437	404.277	406.533
<i>TRACE/SCF DIAS</i>	<i>#1</i>	400	412.718	424.31	434.853	444.401	453.066
	<i>#2</i>	400	403.067	406.098	409.155	412.264	415.491
	<i>#3</i>	400	400.529	401.156	401.907	402.789	403.821
	<i>#4</i>	400	403.161	406.285	409.709	413.075	416.545



**Figure 5-37 – Comparison of simulated core outlets temperature (K) with variational inlet temperature predicted by SCF, TRACE standalone, and TRACE with DIAS correlations at loop 1.**

The 500K inlet case has been selected to evaluate the performance of DIAS correlations on correlating the temperature field in the whole overlapped core area. Table 5-12 gives the coolant temperature distributions along with the core height predicted by SCF, TRACE with DIAS correlations, and TRACE standalone. The small temperature changes in the four sectors along the core observed in the TRACE standalone results indicate that the heat transfer in the core region is relatively weak. The DIAS correlated temperature becomes almost the same with that of SCF in the core entrance region which is represented by the first and second cells in the TRACE model and the temperature fields were flatted in a rather small range in the upper half core.

**Table 5-12 – Coolant temperature (K) distribution in the core predicted by SCF, TRACE standalone, and TRACE with DIAS correlations when the inlet temperature of loop 1 is 500K.**

	<i>Height</i>	<i>3.91 m</i>	<i>4.83 m</i>	<i>5.13 m</i>	<i>5.44 m</i>	<i>5.74 m</i>	<i>6.05 m</i>	<i>6.35 m</i>	<i>6.66 m</i>
<i>SCF</i>	<i>#1</i>	455.57	454.74	454.14	453.91	453.69	453.46	453.27	453.07
	<i>#2</i>	415.61	415.55	415.51	415.5	415.49	415.49	415.49	415.49
	<i>#3</i>	403.6	403.66	403.71	403.73	403.75	403.77	403.8	403.82
	<i>#4</i>	416.69	416.63	416.58	416.57	416.56	416.55	416.55	416.55
<i>TRACE Standalone</i>	<i>#1</i>	479.54	479.54	479.53	479.53	479.53	479.53	479.53	479.53
	<i>#2</i>	405.3	405.61	405.68	405.74	405.82	405.87	405.92	405.97
	<i>#3</i>	400.23	400.23	400.23	400.24	400.24	400.24	400.24	400.25
	<i>#4</i>	406.55	406.54	406.54	406.54	406.54	406.53	406.53	406.53
<i>TRACE DIAS</i>	<i>#1</i>	455.57	454.73	454.14	453.91	453.69	453.46	453.27	453.07
	<i>#2</i>	415.60	415.55	415.51	415.5	415.49	415.49	415.49	415.49
	<i>#3</i>	403.59	403.66	403.71	403.73	403.75	403.77	403.79	403.82
	<i>#4</i>	416.69	416.63	416.58	416.57	416.55	416.55	416.54	416.55

Up to this point, the capability of the DIAS approach in correlating TRACE to produce SCF-consistent temperature distributions in the entire overlapped regions is demonstrated. Despite the specific simulation tools, this approach supplies a universal idea to use sub-channel or CFD data to correlate system code's hydraulic performance.

### **c) Testing of the solute boron correlation**

The boron concentration has been selected as the typical solute in NPP system to evaluate the DIAS correlations on solute fields in the vessel. The 3D boron field calculated by SCF in the core is used to correlate TRACE prediction. Six cases are analyzed; the corresponding boundary conditions are listed in Table 5-13. The single variable is the inlet boron mass ratio at loop 1 (Figure 5-31) which ranges from 0.01 to 0.06 with 0.01 intervals, whereas the inlet boron mass ratio at the other three loops are fixed to be 0.01. Meanwhile, the inlet coolant velocities, temperatures, and the outlet pressures at the four loops keep identical during the testing phase.

**Table 5-13 – Boundary conditions of the test cases for solute concentration correlation.**

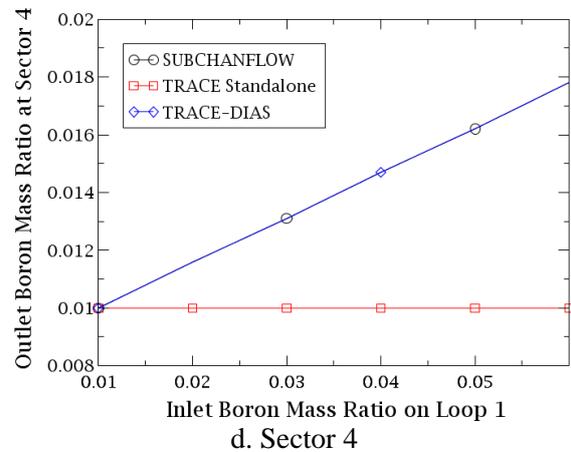
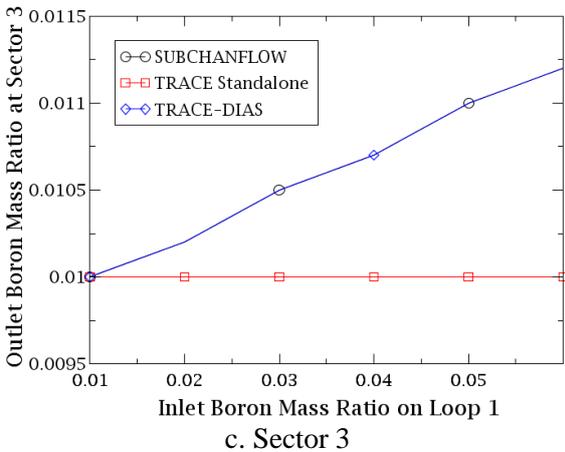
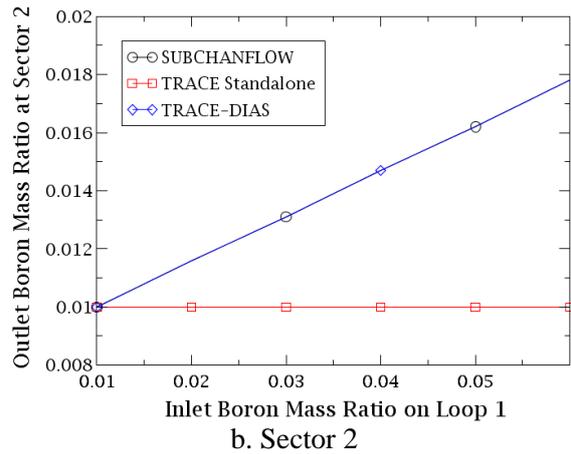
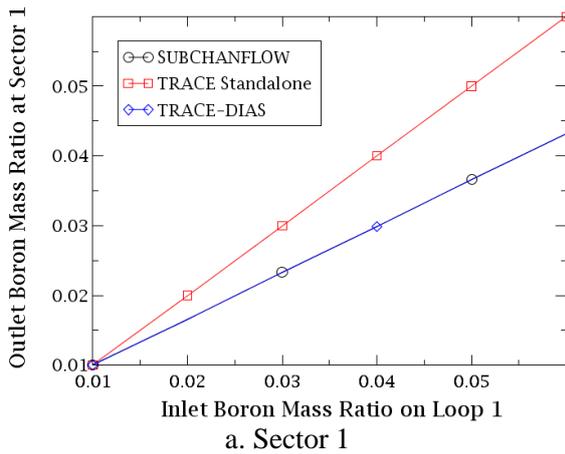
<i>Inlet Boron Mass Ratio</i>	<i>Loop 1</i>	<i>0.01</i>	<i>0.02</i>	<i>0.03</i>	<i>0.04</i>	<i>0.05</i>	<i>0.06</i>
		<i>Loop 2~4</i>	0.01				
<i>Temperature</i>	<i>Loop 2~4</i>	400 K					
<i>Velocity</i>	<i>Loop 1~4</i>	10 m/s					
<i>Outlet Pressure</i>	<i>Loop 1~4</i>	15.55 MPa					

Table 5-14 provides the calculated boron mass ratio at the core outlets for the six test cases by SCF, TRACE standalone and TRACE/SCF with DIAS correlations. A full consistency has been observed for the results of SCF and TRACE/SCF with DIAS correlations, which demonstrates the ability of the

DIAS-approach in correlating TRACE solute concentration field to SCF result. As to TRACE standalone, the identical simulated data with the inlet boundary conditions indicate that there is almost no mass transfer occurring between the azimuthal sectors in TRACE. The curves are in Figure 5-38.

**Table 5-14 – Boron mass ratio at the core outlets predicted by SCF, TRACE standalone, and TRACE with DIAS correlations.**

	<i>Sector</i>	<i>0.01</i>	<i>0.02</i>	<i>0.03</i>	<i>0.04</i>	<i>0.05</i>	<i>0.06</i>
<i>SCF</i>	<i>#1</i>	0.01	0.0166	0.0233	0.0299	0.0366	0.0432
	<i>#2</i>	0.01	0.0116	0.0131	0.0147	0.0162	0.0178
	<i>#3</i>	0.01	0.0102	0.0105	0.0107	0.011	0.0112
	<i>#4</i>	0.01	0.0116	0.0131	0.0147	0.0162	0.0178
<i>TRACE Standalone</i>	<i>#1</i>	0.01	0.02	0.03	0.04	0.05	0.06
	<i>#2</i>	0.01	0.01	0.01	0.01	0.01	0.01
	<i>#3</i>	0.01	0.01	0.01	0.01	0.01	0.01
	<i>#4</i>	0.01	0.01	0.01	0.01	0.01	0.01
<i>TRACE DIAS</i>	<i>#1</i>	0.01	0.0166	0.0233	0.0299	0.0366	0.0432
	<i>#2</i>	0.01	0.0116	0.0131	0.0147	0.0162	0.0178
	<i>#3</i>	0.01	0.0102	0.0105	0.0107	0.011	0.0112
	<i>#4</i>	0.01	0.0116	0.0131	0.0147	0.0162	0.0178



**Figure 5-38 – Calculated boron mass ratio distribution at the core outlets predicted by SCF, TRACE standalone, and TRACE with DIAS correlations with variational inlet boron mass ratio at Loop 1.**

The correlations exactly smooth the boron mass ratio distribution in the core which is derived from Figure 5-38 where the outlet boron mass ratio of SCF and TRACE/SCF with DIAS correlations decrease at sector 1 and increase at the other three sectors than TRACE standalone. The higher increment in sector 2 and 4 than sector 3 is because they are directly adjacent to sector 1 which owns the highest boron mass ratio while sector 3 locates farthest from sector 1.

The case with 0.06 boron mass ratio inlet was selected to evaluate the DIAS correlations in the whole overlapped domain i.e. the core. As the results in Table 5-15 show, the data by TRACE/SCF with DIAS correlations is fully consistent with the SCF description while TRACE standalone produces significant deviation. Hence, it can be concluded that the DIAS approach makes possible that TRACE strictly follows the prediction of boron mass ratio in the entire overlapped domains.

**Table 5-15 – Boron mass ratio distribution in the core predicted by SCF, TRACE standalone, and TRACE with DIAS correlations when inlet boron mass ratio of loop 1 is 0.06.**

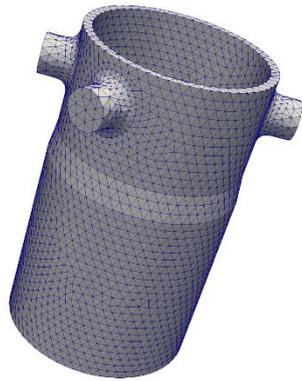
	<i>Height</i>	<i>3.91 m</i>	<i>4.83 m</i>	<i>5.13 m</i>	<i>5.44 m</i>	<i>5.74 m</i>	<i>6.05 m</i>	<i>6.35 m</i>	<i>6.66 m</i>
<i>SCF</i>	<i>#1</i>	0.0439	0.0437	0.0436	0.0435	0.0434	0.0434	0.0433	0.0432
	<i>#2</i>	0.0176	0.0176	0.0177	0.0177	0.0177	0.0177	0.0178	0.0178
	<i>#3</i>	0.011	0.0111	0.0111	0.0111	0.0111	0.0112	0.0112	0.0112
	<i>#4</i>	0.0176	0.0176	0.0177	0.0177	0.0177	0.0177	0.0177	0.0178
<i>TRACE Standalone</i>	<i>#1</i>	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
	<i>#2</i>	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	<i>#3</i>	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	<i>#4</i>	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
<i>TRACE DIAS</i>	<i>#1</i>	0.0439	0.0437	0.0436	0.0435	0.0434	0.0434	0.0433	0.0432
	<i>#2</i>	0.0176	0.0176	0.0177	0.0177	0.0177	0.0177	0.0178	0.0178
	<i>#3</i>	0.011	0.0111	0.0111	0.0111	0.0111	0.0112	0.0112	0.0112
	<i>#4</i>	0.0176	0.0176	0.0177	0.0177	0.0177	0.0177	0.0177	0.0178

## 5.4 Validation of TRACE/TrioCFD-ICoCo using the VVER-1000 coolant mixing benchmark

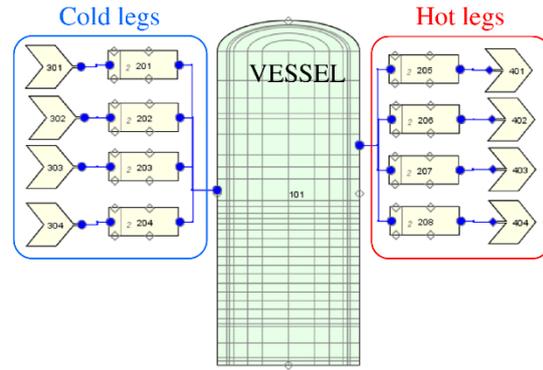
The coupled code TRACE/TrioCFD based on ICoCo is validated by the analysis of the VVER-1000 coolant mixing benchmark. This is the same case used for validation of TRACE/SCF based on ECI in section 4.4. The benchmark verification has been sufficiently described and it will not be repeated in this chapter. In this investigation, the capability of TRACE/TrioCFD based on ICoCo is evaluated and the obtained results are compared to the standalone TRACE-simulations as well as the measured data.

### 5.4.1 Description of the thermal-hydraulic models of TRACE and TrioCFD

TRACE/TrioCFD employs a domain overlapping approach in which TrioCFD simulates the downcomer (Figure 5-39) in the vessel while TRACE simulates the whole vessel region (Figure 5-40).

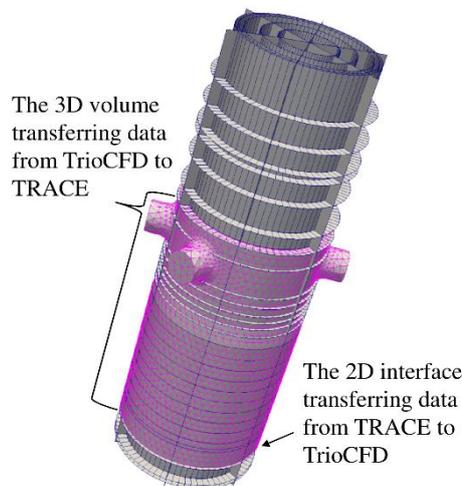


**Figure 5-39 – TrioCFD downcomer-mesh of the VVER-1000 coolant mixing benchmark for the validation of TRACE/TrioCFD-ICoCo.**



**Figure 5-40 – TRACE model of the VVER-1000 coolant mixing benchmark for the validation of TRACE/TrioCFD-ICoCo.**

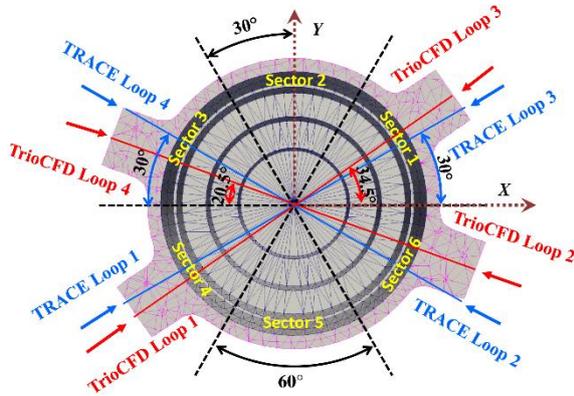
Thanks to the implemented ICoCo functionalities, TRACE now has meshes for the VESSEL component. The assembled meshes of TRACE (here is the edge mesh) and TrioCFD are displayed in Figure 5-41 where an obvious overlapped thermal-hydraulic domain in the downcomer can be observed. In this case, the inlet boundary condition of TrioCFD is pre-defined in the input file. The outlet boundary condition of TrioCFD comes from TRACE dynamically through the 2D interface where mesh interpolation and field mapping are handled by the MEDCoupling library. The 3D volumetric fields of TrioCFD are translated by MEDCoupling and passed to TRACE to correlate its corresponding fields through the DIAS method.



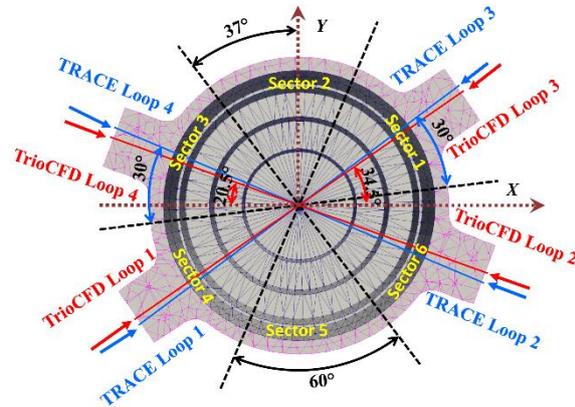
**Figure 5-41 – The assembled meshes of TRACE and TrioCFD of the VVER-1000 coolant mixing benchmark for the validation of TRACE/TrioCFD-ICoCo.**

According to the benchmark specification, the four loops are not symmetrically arranged. The real unsymmetrically arranged meshes are shown in Figure 5-42 where the TRACE inlets (blue arrows and lines) are positioned symmetrically according to TRACE azimuthal sectors (yellow texts) while TrioCFD inlets (red arrows and lines) shift an anticlockwise degree. This mesh arrangement precisely reflects reality. It is expected to predict the temperature distribution of the coolant flowing into the core in a more precise manner, by taking the unsymmetrical vessel geometry into consideration. However, this mesh arrangement leads to a mismatch between the vessel inlet and outlet nozzles, since the positions of the problem inlets are now defined by TrioCFD while the positions of the outlets are still by TRACE (TrioCFD simulates only the downcomer while the upper part of the vessel is still

handled by TRACE). Due to the unsymmetrically arranged inlets and outlets, the assembled mesh of Figure 5-42 can't be used for the analysis of the coolant temperature distribution at the hot legs. Thus, the case where the TRACE mesh is also shifted an anticlockwise degree and symmetrically assembled to the TrioCFD mesh (see Figure 5-43) have to be simulated as well.



**Figure 5-42 – Real unsymmetrically assembled meshes for the validation of TRACE/TrioCFD-ICoCo.**

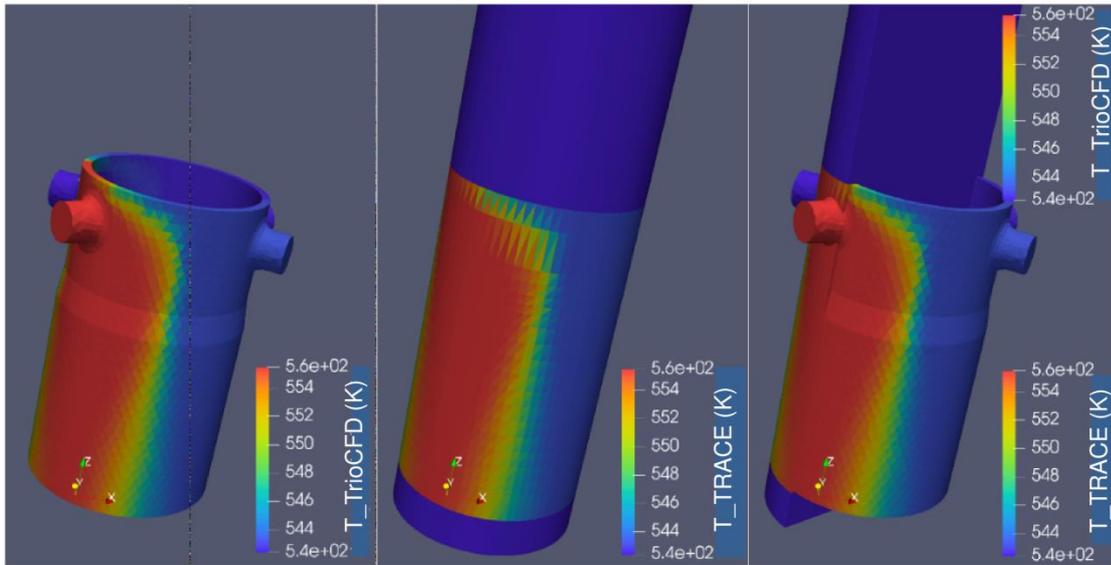


**Figure 5-43 – Idealized symmetrically assembled meshes for the validation of TRACE/TrioCFD-ICoCo.**

The symmetrically assembled meshes are used to investigate the coolant temperature distribution at the hot legs while the unsymmetrically assembled meshes are used to investigate the coolant temperature distribution at the core outlets. It is worth to note that the symmetrical mesh-pair has been developed to ensure the position-matching of inlet and outlet nozzles. It is obsolete if TrioCFD also takes the upper plenum into account. However, since the upper plenum involves complex geometry which leads to difficult mesh, it is reasonable to use simple models to demonstrate the capabilities of the coupled code.

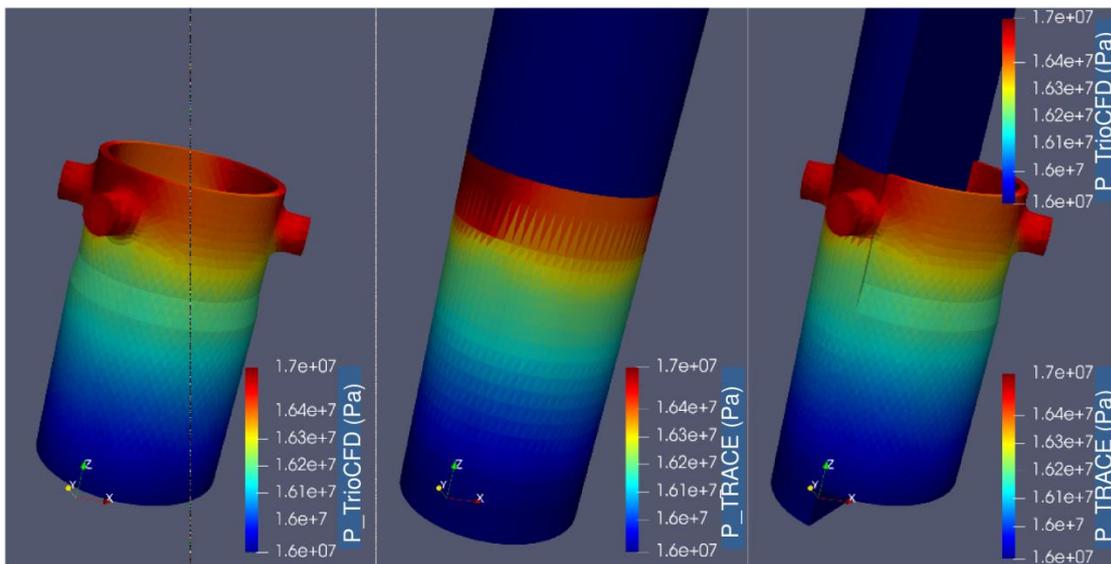
## 5.4.2 Discussion of the selected results

The correlated fields of TRACE within the coupled code during the simulation include the coolant temperature, the pressure, and the coolant velocity. The correlations are performed in the entire overlapped domain. Figure 5-44 illustrates the coolant temperature distributions on the fine downcomer mesh, the TRACE mesh, and the overlapped meshes in sequence. The coolant temperature matching between TRACE and TrioCFD is evident, indicating the temperature translation is executed correctly. It can be observed that the hot coolant enters the downcomer from one loop and gradually diffuses over a larger area along the main flow direction.



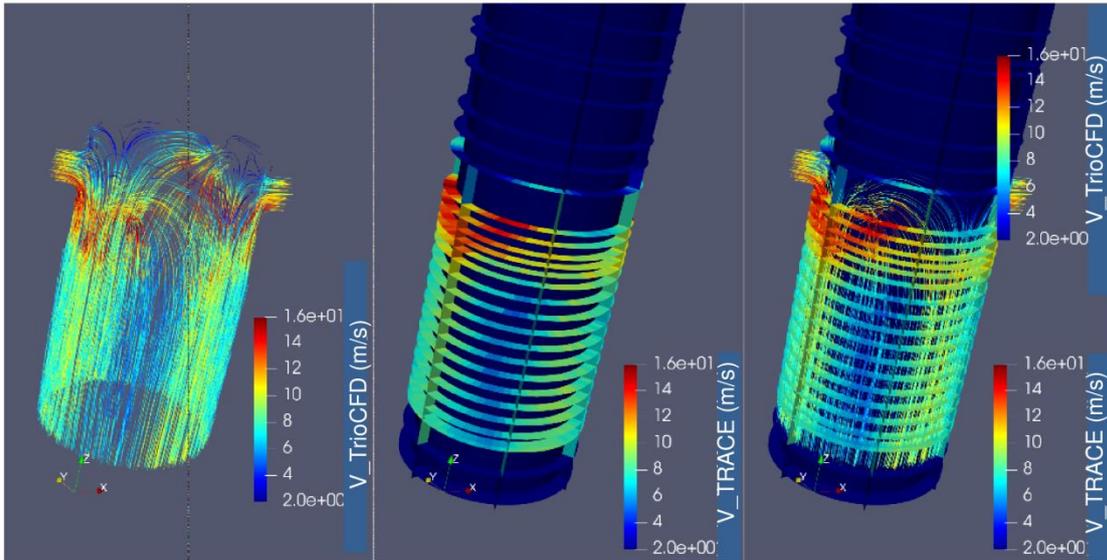
**Figure 5-44 – The coolant temperature distribution in the downcomer on meshes of TRACE and TrioCFD.**

Figure 5-45 depicts the pressure distributions on the fine downcomer mesh, the TRACE mesh, and the overlapped meshes in sequence. The pressure matching between TRACE and TrioCFD is evident, indicating the pressure translation from TrioCFD to TRACE is executed correctly. It is obvious that the pressure decreases along the main flow direction.



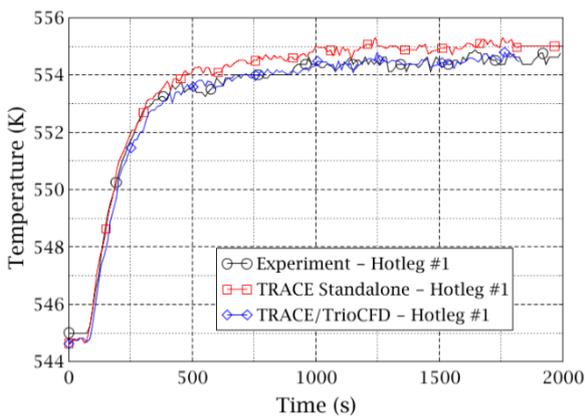
**Figure 5-45 – The pressure distribution in the downcomer on meshes of TRACE and TrioCFD.**

Figure 5-46 displays the coolant velocity distributions on the fine downcomer mesh, the TRACE mesh, and the overlapped meshes in sequence. The velocity matching between TRACE and TrioCFD is also evident, indicating the coolant velocity translation from TrioCFD to TRACE is executed correctly. The flow streamlines tell that the coolant enters the vessel through the four inlet nozzles and spreads to the downcomer immediately. The main flow is downward while limited amount of flow is upward and then reverse to contribute to the downward flow.

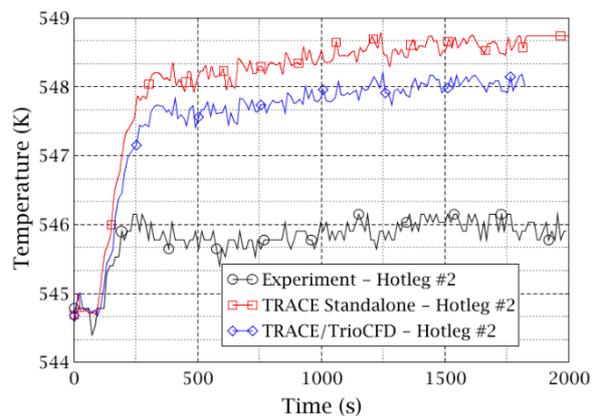


**Figure 5-46 – The coolant velocity distribution in the downcomer on meshes of TRACE and TrioCFD.**

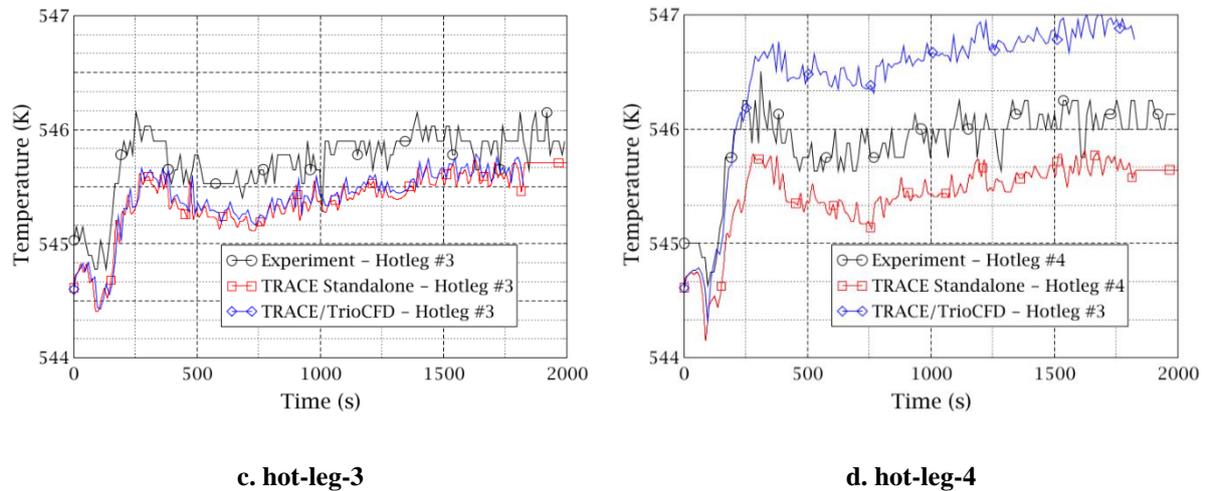
The coolant temperature at the four hot legs predicted by TRACE and TRACE/TrioCFD is compared to the measured data in Figure 5-47. The temperature rise of hot-leg-1 can be observed in Figure 5-47a, where TRACE standalone tends to over-predict the heat-up while the coolant temperature calculated by TRACE/TrioCFD is very close to the plant data. In Figure 5-47b, both of TRACE standalone and TRACE/TrioCFD over-predict the coolant temperature at hot-leg-2. Nevertheless, the coupled code exhibits an improvement and the curve is located closer to the measured data than that of TRACE standalone. At hot-leg-3 (Figure 5-47c), the predictions of TRACE and TRACE/TrioCFD are all not satisfactory, where the coolant temperature are both under-predicted compared to the measured data. In Figure 5-47d, TRACE standalone under-predicts the heat-up at hot-leg-4 while TRACE/TrioCFD over-predicts the heat up. It could be concluded that TRACE/TrioCFD performs better than TRACE standalone on the coolant temperature prediction at the hot-legs, though there is still room for improvement.



**a. hot-leg-1**



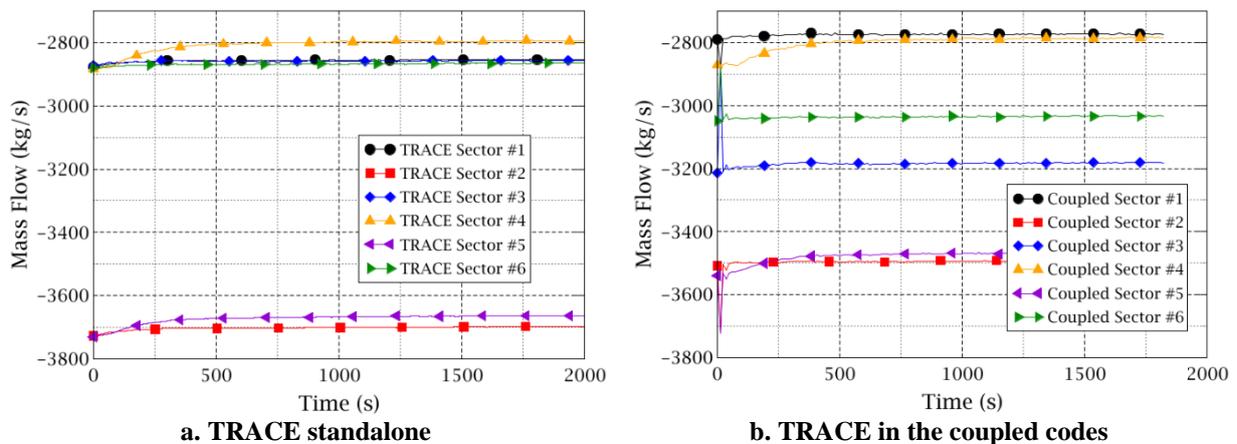
**b. hot-leg-2**



**Figure 5-47 – Comparison of calculated coolant temperatures at the hot-legs of TRACE, TRACE/TrioCFD with the measured data.**

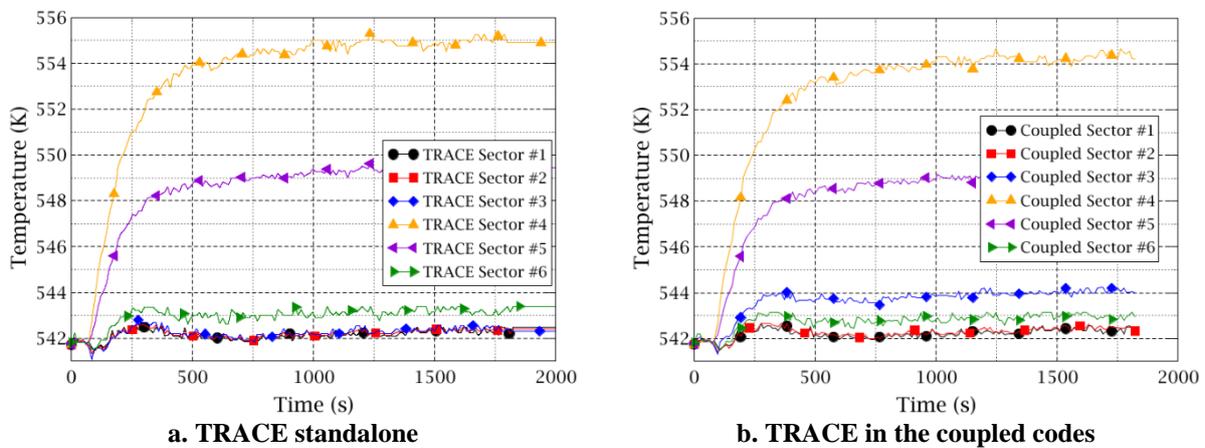
The encouraging result indicates TrioCFD indeed introduce positive influence to TRACE in the overlapped domain within the coupled code. In order to more clearly reveal this effect, the coolant mass flow rate and temperature distribution at the downcomer outlet of TRACE within the coupled code are carefully inspected and compared with that of TRACE standalone. Figure 5-48 depicts the simulated coolant mass flow distribution. The curves on the left figure represent data from TRACE standalone. The curves on the right figure display the data from TRACE within the coupled code. The “sector” formulation in the graph legends corresponds to the six azimuthal sectors of the TRACE model (see Figure 5-42 and Figure 5-43).

The downward mass flow across sector 1, 3, 4, and 6 of TRACE standalone are roughly the same and are between 2800 and 2900 kg/s, while the mass flow across sector 2 and 5 are as large as 3700 kg/s. Compared with that, the mass flow predicted by TRACE within the coupled code is much flatter. Moreover, the inlet differences are reflected more clearly by the coupled code. This kind of hydraulic correlation is handled by the DIAS method which uses the coolant velocity and pressure fields in the entire overlapped domain to correlate the friction coefficient of TRACE, thus “guide” TRACE produce the quasi-CFD hydraulic fields.



**Figure 5-48 – Computed coolant mass flow rate distribution at the downcomer outlet of TRACE standalone and TRACE within the coupled code.**

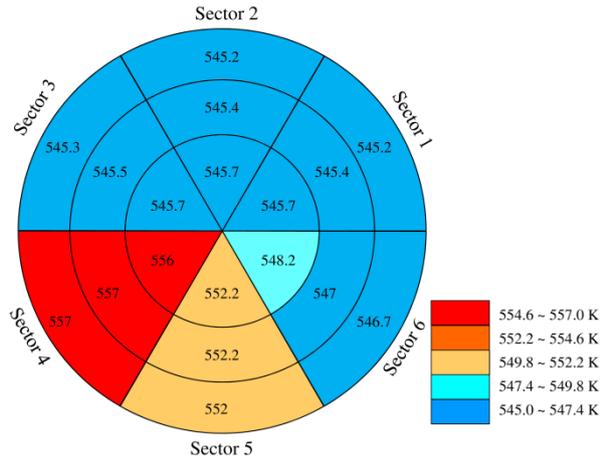
Figure 5-49 illustrates the simulated coolant temperature distribution at the downcomer outlet as a function of time. The curves on the left figure represent data from TRACE standalone. The curves on the right figure display the data from TRACE within the coupled code. The “sector” formulation in the graph legends also corresponds to the six azimuthal sectors of the TRACE. Similar curve trends are observed as that at the four hot-legs shown in Figure 5-47. The coolant temperature on sector 4 (corresponds to loop 1) of TRACE within the coupled code is lower than that of TRACE standalone (Figure 5-47a depicts this phenomenon as well). The coolant temperature on sector 6 (corresponds to loop 2) of TRACE within the coupled code is slightly lower than that of TRACE standalone (Figure 5-47b depicts this phenomenon as well). The coolant temperature on sector 1 (corresponds to loop 3) of the two calculations shown no significant difference (Figure 5-47c depicts this phenomenon as well). The coolant temperature on sector 3 (corresponds to loop 4) of TRACE within the coupled code is higher than that of TRACE standalone (Figure 5-47d depicts this phenomenon as well). Additionally, the coolant temperature on sector 5 (the sector between loop 1 and loop 2) locates right in the middle of the graph scale telling that the heat-up of sector 6 (loop 2) comes from sector 4 (loop 1) through sector 5.



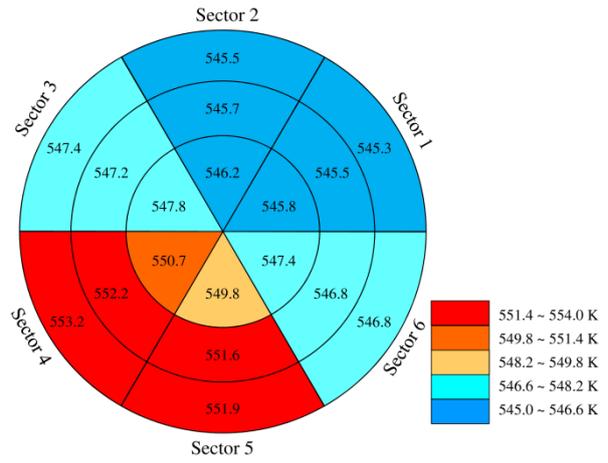
**Figure 5-49 – The coolant temperature distribution at the downcomer outlet of TRACE standalone and TRACE within the coupled code.**

The data analyzed by the previous part originate all from the symmetrically assembled meshes of TRACE and TrioCFD (see Figure 5-43). They are appropriate for the analysis of the coolant distribution at the hot-legs by omitting the mismatch of inlet and outlet nozzles. However, the inherent asymmetry between the vessel and the core is missing. This leads to a distortion of the flow conditions entering the core. Thus the actual case where the unsymmetrical assembled meshes are applied has to be simulated when the flow condition in the core area is concerned.

Figure 5-50 provides the calculated coolant temperature distribution predicted by TRACE standalone at the core outlet. A sharp temperature change over sector 3 and 4 is observed indicating a weak coolant mixing between the two sectors. The high coolant temperature in sole sector 4 not only signify the limited mass and heat transfer from this sector to others but is also partly due to the wrong location of the inlet nozzle defined by TRACE. By applying the correlation from TrioCFD to TRACE whose meshes precisely reflect the right VVER-vessel geometry, the coolant temperature at the core outlet now distributes much flatter (see Figure 5-51), which basically follows the profile of the measured data shown in Figure 4-33. There, the hottest coolant occupies both sector 4 and sector 5, and the centerline of the hottest part locates closer to the boundary of the two sectors. This more realistic coolant temperature distribution at the core outlet is believed to come from the right unsymmetrical geometry modeled by TrioCFD as well as the enhanced coolant mixing introduced from TrioCFD to TRACE in the overlapped downcomer region.



**Figure 5-50 – The coolant temperature distribution at the core outlet predicted by TRACE standalone at the end of the problem.**



**Figure 5-51 – The coolant temperature distribution at the core outlet predicted by TRACE/TrioCFD at the end of the problem.**

## 6 Summary and Perspectives

This dissertation was devoted to the development of multi-scale coupling systems by combining single thermal-hydraulic solvers (CFD, sub-channel and system codes) together using different methodologies (Exterior Communication Interface – ECI, Interface for Code Coupling - ICoCo) in order to improve the simulation of the multi-dimensional thermal-hydraulic phenomena taking place inside the core and the reactor pressure vessel of a Pressurized Water Reactor (PWR). Moreover, the new coupled codes were tested, verified and validated using appropriate both academic problems and experimental data.

Hereafter, the main achievements will be summarized. Based on the performed investigations, the ideas for future research on promising fields are given.

### 6.1 Summary

Three main achievements are accomplished in this thesis corresponding to the objectives proposed at the thesis beginning. They are:

- 1) The various multi-scale thermal-hydraulic coupling approaches are reviewed, classified, and evaluated.
  - a. The diverse approaches are reviewed and they are classified into five main strategies which consist of different sub-strategies. Based on that, a universal standard for multi-scale thermal-hydraulic coupling approaches is put forward.
  - b. A quantized evaluation criterion is put forward. It is applied to evaluate the classified approaches and the standard quantitatively. The standard along with its evaluation supplies a general guideline in selecting appropriate approaches for new coupling tasks.
- 2) The system code TRACE is coupled with the sub-channel code SCF based on the Exterior Communication Interface (ECI). The multi-scale thermal-hydraulic coupling codes – TRACE/SCF-ECI is developed, verified, and validated.
  - a. A fresh ECI module is developed for SCF. Its socket-property enables SCF couple with any other ECI-equipped codes on distributed computation systems.
  - b. TRACE and SCF are coupled through the ECI interface. This is a server-less system in which the two codes are separated from each other and the calculation don't need an additional supervisor. TRACE and SCF run in parallel and their simulating thermal-hydraulic domains separate. The data exchange takes place at the domain boundaries and performs only one time during a time-step (explicit coupling).
  - c. A new Fortran-toolkit is developed to manage the field translation between different meshes or grids of TRACE and SCF. It is able to translate the field by a build-in overlapping-area-weighted algorithm automatically and precisely. Users are freed from the repeatable and error-prone work in defining the relationship between the different grids.
  - d. TRACE/SCF-ECI is verified with an academic coolant mixing case and is further validated with a VVER-1000 coolant mixing benchmark. For both cases, enhancement of the coolant mixing is observed thanks to the SCF capability embedded in the coupling system. The encouraging results indicate that the coupling code is correctly implemented and works in a proper manner.
  - e. New optimized computing schemes are proposed. They can significantly improve the coupled codes' efficiency. The efficiency is tested with the VVER-1000 coolant mixing benchmark.

- 3) The system code TRACE is coupled with the open-source CFD code TrioCFD based on the Interface for Code Coupling (ICoCo). The multi-scale thermal-hydraulic coupling codes – TRACE/TrioCFD-ICoCo is developed, verified, and validated.
  - a. A fresh ICoCo module is developed for TRACE. Here, TRACE is split into several functional components according to ICoCo standard. A new mesh-generation function is developed with which the vessel component in TRACE can be represented by explicit meshes. The data in the vessel component can be post-processed in a 3D manner. Additionally, TRACE together with its ICoCo module is implemented into SALOME platform.
  - b. TRACE and TrioCFD are coupled through the ICoCo interface. This is a server-client system in which the two codes are separated from each other and are coordinated by a newly-developed C++ supervisor. TRACE and TrioCFD run in parallel and their simulating thermal-hydraulic domains overlap with each other. The data exchange takes place at the domain boundaries as well as the entire overlapped domain. The 3<sup>rd</sup> mesh-processing library – MEDCoupling is used to handle the field translation between the different meshes of the two codes. The explicit coupling approach is adopted.
  - c. A new Dynamic-Implicit-Additional-Coupling (DIAS) method is put forward and it define the algorithm using the 3D data from SCF to refine TRACE solutions. This method now can handle four fields: the coolant velocity, the pressure, the coolant temperature, and the boron concentration. It is verified with a flow splitter problem an academic reactor case. The results are encouraging.
  - d. TRACE/TrioCFD-ICoCo is validated against a VVER-1000 coolant mixing benchmark. The result tells that the coolant mixing can be predicted in a more realistic manner by the coupled codes, which indicates that the coupling code is correctly implemented and works in a proper manner.

All of the objectives declared in this thesis have been achieved. Additionally, several miscellaneous functions are also developed in order to make the coupling systems more flexible, more extensible, more automatic, and more powerful.

## 6.2 Perspectives

As it can be inferred from the dissertation, this is a more method-oriented work with limited application and validation cases. In order to sufficiently, comprehensively evaluate the newly-developed multi-scale thermal-hydraulic codes as well as the newly-developed methods, additional validation work is necessary. For example, use sufficient data from real plant or experimental facilities to validate the coupled codes under different safety-relevant scenarios such as boron dilution, coolant mixing during a steam line break accident, etc. The following lines of research and development may be subject to further research:

- a) For TRACE/SCF-ECI: The cylindrical VESSEL component can be assembled with a Cartesian VESSEL component to represent the core and part of the SCF model may go deep down to the pin level. New mapping logics have to be developed in order to properly map data between the Cartesian VESSEL and the SCF meshes;
- b) For TRACE/TrioCFD-ICoCo: In the current coupling codes, the inlet of TrioCFD still comes from the pre-defined boundary conditions, which is not that accurate as the dynamically derived outlet pressure boundary condition from TRACE. The reason is that the inlet of TrioCFD in this dissertation comes from the TRACE PIPE component which is 1D and does not have an explicit mesh. So, the data mapping at the PIPE component is not realized yet. Nevertheless, the MED mesh of the PIPE components and other components can be developed in the following study to achieve different and flexible coupled models e.g. the overlapped domain in the steam line, in the pressurizer, in the steam generator, etc. This

will make the local CFD analysis with system dynamic feedback in other parts out of the vessel possible.

Especially for ICoCo-based coupling, since TRACE, TrioCFD and SCF are equipped now with ICoCo (pre-defined MED mesh and pre-split functional component), a complete multi-scale coupling system can be implemented based on ICoCo. Here, TRACE simulates the whole nuclear power plant, SCF is in charge of the core, and TrioCFD is in responsible for the downcomer and lower plenum (the upper plenum could also be included). This system should be based on the domain-overlapping and DIAS method. In this coupling system, TRACE gives vessel inlet Boundary Condition (BC) to TrioCFD and gives core outlet BC to SCF. Then, it uses the 3D-feedback from TrioCFD and SCF to correlate the fields in its overlapped domains via DIAS-method. TrioCFD (simulates the downcomer and lower plenum) get its inlet BC from TRACE and its outlet BC from SCF, run a step forward, and then pass the 3D fields back to TRACE and gives the BC to SCF. SCF gets its inlet BC from TrioCFD and outlet BC from TRACE, run a step forward, and give the 3D fields to TRACE and the corresponding BC to TrioCFD. This is a complete multi-scale thermal-hydraulic coupling system based on ICoCo.

Furthermore, since the real nuclear power systems not only involve multi-scale thermal-hydraulic phenomena but also involve multi-physical aspects, works in the future also include other physical solvers e.g. a neutronic code. In this way, a multi-scale multi-physical simulation system is built. Currently, TRACE itself already has an in-built PARCS neutronics module. However, it is internally coupled to TRACE and has poor expansibility. The development of an ICoCo-module for PARCS will pave the way for such promising developments. Such a code system is very much appropriate to simulate complex scenarios such as the Main Steam Line Break accident scenario, where multi-scale thermal-hydraulic phenomena take place and strong feedback between the core neutronics and the plant thermal-hydraulics exists.



## References

- Abu-Khader, M. M., 2009. Recent advances in nuclear power: A review. *Progress in Nuclear Energy*, 51(2), pp. 225-235. <https://doi.org/10.1016/j.pnucene.2008.05.001>.
- Al-Othman, A. et al., 2019. Nuclear desalination: A state-of-the-art review. *Desalination*, Volume 457, pp. 39-61. <https://doi.org/10.1016/j.desal.2019.01.002>.
- Angeli, P. E., Bieder, U. & Fauchet, G., 2015. *Overview of the TrioCFD code: main features. V&V procedures and typical applications to nuclear engineering*. Chicago, U.S., NURETH-16: The 16th International Topical Meeting on Nuclear Reactor Thermal Hydraulics, August 30-September 4, 2015, ISBN: 978-0-89448-722-4, Paper ID: 13006, pp. 252-265.
- Antal, S., Ettore, S., Kunz, R. & Podowski, M., 2000. *Development of a next generation computer code for the prediction of multicomponent multiphase flows*. Cargese, France: International Meeting on Trends in Numerical and Physical Modeling for Industrial Multiphase Flow, 27 September, 2000.
- Aragonés, J. M. et al., 2004. Methods and results for the MSLB NEA benchmark using SIMTRAN and RELAP-5. *Nuclear Technology*, Volume 146, pp. 29-40. <https://doi.org/10.13182/NT04-A3484>.
- Basualdo, J. & Sanchez, V., 2017. *PARCS-SUBCHANFLOW-TRANSURANUS multiphysics coupling for improved PWR's simulations*. Fukui and Kyoto, Japan, ICAPP 2017: International Congress on Advances in Nuclear Power Plants, April 24-28, 2017, ISBN: 978-4-89047-167-6, Paper ID: 17096.
- Bavière, R. et al., 2014. A first system/CFD coupled simulation of a complete nuclear reactor transient using CATHARE2 and TRIO U. preliminary validation on the Phénix reactor natural circulation test. *Nuclear Engineering and Design*, Volume 277, pp. 124-137. <https://doi.org/10.1016/j.nucengdes.2014.05.031>.
- Berkhan, A., Sanchez, V. & Imke, U., 2011. *Validation of PWR-relevant models of SUBCHANFLOW using the NUPEC PSBT data*. Toronto, Ontario, Canada, NURETH-14: The 14th International Topical Meeting on Nuclear Reactor Thermal hydraulics, September 25-30, 2011, ISBN 978-1-926773-05-6, Paper ID: 191.
- Bestion, D., 2010. From the direct numerical simulation to system codes - perspective for the multi-scale analysis of LWR thermal hydraulics. *Nuclear Engineering and Technology*, Volume 42, pp. 608-619, <https://doi.org/10.5516/NET.2010.42.6.608>.
- Bestion, D., 2014. The difficult challenge of a two-phase CFD modelling for all flow regimes. *Nuclear Engineering and Design*, Volume 279, pp. 116–125, <https://doi.org/10.1016/j.nucengdes.2014.04.006>.
- Bestion, D., 2018. *System code models and capabilities*. Vitanza, Italy, THICKT-2008, 5-9 May, 2008, Paper ID: Session III paper 06, pp. 81-106.
- Bestion, D. et al., 2012. *Multi-scale Thermal hydraulic analyses performed in NURESIM and NURISP projects*. Anaheim, California, U.S., 20th International Conference on Nuclear Engineering and the ASME 2012 Power Conference, July 30 - August 3, 2012, DOI: 10.1115/ICONE20-POWER2012-54891, Paper ID: 54891.
- Bieder, U. & Graffard, E., 2007. Qualification of the CFD code Trio\_U for full scale reactor applications. *Nuclear Engineering and Design*, 238(3), pp. 671-679, <https://doi.org/10.1016/j.nucengdes.2007.02.040>.

- Böttcher, M., 2008. Detailed CFX-5 study of the coolant mixing within the reactor pressure vessel of a VVER-1000 reactor during a non-symmetrical heat-up test. *Nuclear Engineering and Design*, Volume 238, pp. 445–452, <https://doi.org/10.1016/j.nucengdes.2007.02.054>.
- Böttcher, M. & Krüßmann, R., 2010. Primary loop study of a VVER-1000 reactor with special focus on coolant mixing. *Nuclear Engineering and Design*, Volume 240, pp. 2244–2253, <https://doi.org/10.1016/j.nucengdes.2010.02.044>.
- Bousbia-Salah, A. & D’Auria, F., 2007. Use of coupled code technique for Best Estimate safety analysis of nuclear power plants. *Progress in Nuclear Energy*, 49(1), pp. 1-13, <https://doi.org/10.1016/j.pnucene.2006.10.002>.
- Cadinu, F., Kozłowski, T. & Kudinov, P., 2008. *A closure-on-demand approach to the coupling of CFD and system thermal-hydraulic codes*. Seoul, Korea, NUTHOS-7: The 7th International Topical Meeting on Nuclear Reactor Thermal Hydraulics, Operation and Safety, October 5-9, 2008, Paper ID: 234.
- Cadinu, F. & Kudinov, P., 2009. *Development of a “coupling-by-closure” approach between CFD and system thermal-hydraulics codes*. Kanazawa City, Ishikawa Prefecture, Japan, NURETH-13: The 13th International Topical Meeting on Nuclear Reactor Thermal Hydraulics, September 27-October 2, 2009, Paper ID: 1238.
- Calleja, M. et al., 2014. Coupling of COBAYA3/SUBCHANFLOW inside the NURESIM platform and validation using selected benchmarks. *Annals of Nuclear Energy*, Volume 71, pp. 145-158, <https://doi.org/10.1016/j.anucene.2014.03.036>.
- Calvin, C. & Nowak, D., 2010. *Handbook of nuclear engineering: 12 high-performance computing in nuclear engineering*. Boston, MA, U.S.: Springer, DOI: 10.1007/978-0-387-98149-9, ISBN: 978-0-387-98130-7.
- Chavez, V. J., Imke, U. & Sanchez-Espinoza, V., 2018. TWOPORFLOW: A two-phase flow porous media code, main features and validation with BWR-relevant bundle experiments. *Nuclear Engineering and Design*, Volume 338, pp. 181-188, <https://doi.org/10.1016/j.nucengdes.2018.08.009>.
- Cheng, J. P., Yan, L. M. & Li, F. C., 2016. CFD simulation of a four-loop PWR at asymmetric operation conditions. *Nuclear Engineering and Design*, Volume 300, pp. 591-600, <https://doi.org/10.1016/j.nucengdes.2016.02.018>.
- Cheng, W., Ferng, Y., Chen, S. & Chieng, C., 2015. Development of CFD methodology for investigating thermal-hydraulic characteristics in a PWR dome. *Nuclear Engineering and Design*, Volume 284, pp. 284-292, <https://doi.org/10.1016/j.nucengdes.2014.11.042>.
- Cheng, Z. & Rao, Y., 2015. Strategies for developing subchannel capability in an advanced system thermal hydraulic code: a literature review. *AECL Nuclear Review*, 4(1), pp. 23-41, <https://doi.org/10.12943/ANR.2015.00039>.
- Chorin, A., 1968. Numerical solution of the Navier-Stokes equations. *Mathematics of Computation*, 22(104), pp. 745-762.
- COBRA-FLX-Team, 2010. *COBRA-FLX: a core thermal-hydraulic analysis code: technical report*. Report. Washington D.C. U.S.: AREVA NP inc, ANP-100311NP.
- D’Auria, F. & Galassi, G., 2010. Scaling in nuclear reactor system thermal-hydraulics. *Nuclear Engineering and Design*, 240(10), pp. 3267–3293, <https://doi.org/10.1016/j.nucengdes.2010.06.010>.

- D'Auria, F. et al., 2004. *Neutronics/thermal-hydraulics coupling in LWR technology, state-of-the-art report. Report*. Paris: Nuclear Energy Agency, Organisation for Economic Co-Operation and Development, ISBN: 92-64-02084-5, NEA No. 5436, CRISSUE-S - WP2.
- Daeubler, M. et al., 2015. High-fidelity coupled Monte Carlo neutron transport and thermal-hydraulic simulations using Serpent 2/SUBCHANFLOW. *Annals of Nuclear Energy*, Volume 83, pp. 352-375, <https://doi.org/10.1016/j.anucene.2015.03.040>.
- Deville, E. & Perdu, F., 2012. *Documentation of the Interface for Code Coupling: ICoCo. Report*. Saclay, France: NURISP-NUclear Reactor Integrated Simulation Project, DEN/DANS/DM2S/STMF/LMES/RT/12-029/A.
- El-Emam, R. S. & Özcan, H., 2019. Comprehensive review on the techno-economics of sustainable large-scale clean hydrogen production. *Journal of Cleaner Production*, Volume 220, pp. 593-609, <https://doi.org/10.1016/j.jclepro.2019.01.309>.
- Emonot, P., Souyri, A., Gandrille, J. & Barre, F., 2011. CATHARE-3: a new system code for thermal-hydraulics in the context of the NEPTUNE project. *Nuclear Engineering and Design*, 241(11), pp. 4476-4481, <https://doi.org/10.1016/j.nucengdes.2011.04.049>.
- Fanning, T. H. & Thomas, J. W., 2010. *Advances in coupled safety modeling using systems analysis and high-fidelity methods. Report*. Oak Ridge, U.S.: Argonne National Laboratory, DOI: 10.2172/982349, ANL-GENIV-134.
- Gonzalez-Vargas, J. A., Sanchez-Espinoza, V., R.Stieglitz & Macian-Juan, R., 2018. Development and validation of the new coupled code system TRADYN. *Annals of Nuclear Energy*, Volume 112, pp. 685-692, <https://doi.org/10.1016/j.anucene.2017.07.027>.
- Grahn, A., Gommlich, A., Kliem, S. & Bilodid, Y. K. Y., 2017. Simulation of an MSLB scenario using the 3D neutron kinetic core model DYN3D coupled with the CFD software Trio\_U. *Nuclear Engineering and Design*, Volume 315, pp. 117-127, <https://doi.org/10.1016/j.nucengdes.2017.02.002>.
- Grunloh, T. & Manera, A., 2016. A novel domain overlapping strategy for the multiscale coupling of CFD with 1D system codes with applications to transient flows. *Annals of Nuclear Energy*, Volume 90, pp. 422-432, <https://doi.org/10.1016/j.anucene.2015.12.027>.
- Grunloh, T. P., 2016. *A novel multi-scale domain overlapping CFD/STH coupling methodology for multi-dimensional flows relevant to nuclear applications. Dissertation*. Michigan: University of Michigan, <http://hdl.handle.net/2027.42/133243>.
- Guelfi, A. et al., 2007. NEPTUNE: a new software platform for advanced nuclear thermal hydraulics. *Nuclear Science and Engineering*, 156(3), pp. 281-324, <https://doi.org/10.13182/NSE05-98>.
- Hamilton, S. et al., 2016. An assessment of coupling algorithms for nuclear reactor core physics simulations. *Journal of Computational Physics*, Volume 311, pp. 241-257, <https://doi.org/10.1016/j.jcp.2016.02.012>.
- Höhne, T., Krepper, E. & U.Rohde, 2009. Application of CFD codes in nuclear reactor safety analysis. *Science and Technology of Nuclear Installations*, 2010(3), pp. 1-8, <http://dx.doi.org/10.1155/2010/198758>.
- IAEA, 2007. *Use and development of coupled computer codes for the analysis of accidents at nuclear power plants. Report*. Vienna, Austria: IAEA-TECDOC-1539, ISBN: 92-0-101207-1.
- Ilvonen, M., Hovi, V. & Inkinen, P., 2010. *PORFLO development, applications and plans in 2008-2009. Report*. Finland: VTT Technical Research Centre of Finland, VTT-R-01414-10.

- Imke, U. & Sanchez-Espinoza, V., 2012. Validation of the subchannel code SUBCHANFLOW using the NUPEC PWR tests (PSBT). *Science and Technology of Nuclear Installations*, Volume 2012, pp. 1-12, <http://dx.doi.org/10.1155/2012/465059>.
- Imke, U., Sanchez, V. & Gomez, R., 2010. *SUBCHANFLOW: A new empirical Knowledge based Subchannel Code*. Berlin, Germany, German Annual Meeting on Nuclear Technology, INFORUM GmbH, May 4-6, 2010, Paper ID: 202.
- Ivanov, A., Sanchez, V. & Imke, U., 2011. *Development of a coupling scheme between MCNP5 and SUBCHANFLOW for the pin- and fuel assembly-wise simulation of LWR and innovative reactors*. Rio de Janeiro, RJ, Brazil, M&C 2011: International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering, May 8-12, 2011, ISBN: 978-85-63688-00-2, Paper ID: mc30\_14.
- Jeltsov, M., K.Kööp, Kudinov, P. & Villanueva, W., 2013. *Development of a domain overlapping coupling methodology for STHCFD analysis of heavy liquid metal thermal-hydraulics*. Pisa, Italy, NURETH-15: The 15th International Topical Meeting on Nuclear Reactor Thermal - Hydraulics, May 12-17, 2013, ISBN: 978-88-902391-2-0, Paper ID: 466.
- Jeong, J., Yoon, H., Park, I. & Cho, H., 2010. The CUPID code development and assessment strategy. *Nuclear Engineering and Technology*, 42(6), pp. 636–655, <https://doi.org/10.5516/NET.2010.42.6.636>.
- Kessides, I. N., 2010. Nuclear power: understanding the economic risks and uncertainties. *Energy Policy*, Volume 38, pp. 3849-3864, <https://doi.org/10.1016/j.enpol.2010.03.005>.
- Knoll, D. & Keyes, D., 2004. Jacobian-free Newton–Krylov methods: a survey of approaches and applications. *Journal of Computational Physics*, Volume 193, pp. 357–397, <https://doi.org/10.1016/j.jcp.2003.08.010>.
- Kolev, N. et al., 2010. *VVER-1000 Coolant Transient Benchmark Phase 2 (V1000CT-2): Final Specifications of the VVER-1000 Vessel Mixing Problem. Report*. Paris: OECD Nuclear Energy Agency, NEA/NSC/DOC(2010)10, ISBN: 978-92-64-99152-1.
- Krepper, E. et al., 2008. The inhomogeneous MUSIG model for the simulation of polydispersed flows. *Nuclear Engineering and Design*, Volume 238, pp. 1690-1702, <https://doi.org/10.1016/j.nucengdes.2008.01.004>.
- Kucukboyaci, V. & Sung, Y., 2015. *COBRA-TF parallelization and application to PWR reactor core subchannel DNB analysis*. Nashville, ANS MC2015 – Joint International Conference on Mathematics and Computation(M&C), Supercomputing Nuclear Applications (SNA) and the Monte Carlo (MC) Method, April 19-23, 2015, ISBN: 978-0-89448-720-0, CASL-U-2015-0167-000, Paper ID: 176, pp. 1-18.
- Lerchl, G. et al., 2012. *ATHLET mod 3.0 cycle A. Report*. Cologne: Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) mbH, GRS-P-1/Vol.3 Rev.3.
- Mahaffy, J. & C., M., 2003. *An exterior communications interface for the USNRC consolidated code. Report*. State College, USA: Penn State University Applied Research Laboratory, NEA-CSNI-R--2001-2.
- Mahaffy, J. et al., 2007. *Best practice guidelines for the use of CFD in nuclear reactor safety applications*, Paris: Nuclear Energy Agency, Committee on Safety of Nuclear Installations, NEA/CSNI/R(2014)11.

Mañes, J. P. et al., 2014. Validation of NEPTUNE-CFD two-phase flow models using experimental data. *Science and Technology of Nuclear Installations*, Volume 2014, pp. 1-19, <http://dx.doi.org/10.1155/2014/185950>.

MED, 2019. *MEDCoupling developer's guide*. [Online]  
Available at: <http://docs.salome-platform.org/latest/dev/MEDCoupling/developer/index.html>  
[Accessed 04 02 2019].

Moorthi, A., Sharma, A. K. & Velusamy, K., 2018. A review of sub-channel thermal hydraulic codes for nuclear reactor core and future directions. *Nuclear Engineering and Design*, Volume 332, pp. 329–344, <https://doi.org/10.1016/j.nucengdes.2018.03.012>.

Niceno, B., Sato, Y., Badillo, A. & Andreani, M., 2010. Multi-scale modeling and analysis of convective boiling towards the prediction of CHF in rod bundles. *Nuclear Engineering and Technology*, 42(6), pp. 620-635, <https://doi.org/10.5516/NET.2010.42.6.620>.

Nuclear, 2019. *World nuclear association*. [Online]  
Available at: <http://www.world-nuclear.org/information-library/current-and-future-generation/nuclear-power-in-the-world-today.aspx>  
[Accessed 19 03 2019].

NURESAFE, 2019. *Nuclear reactor safety simulation platform*. [Online]  
Available at: <http://www.nuresafe.eu/index.php?art=24>  
[Accessed 16 04 2019].

Papukchiev, A. et al., 2015. Comparison of different coupling CFD–STH approaches for pre-test analysis of a TALL-3D experiment. *Nuclear Engineering and Design*, Volume 290, pp. 135-143, <https://doi.org/10.1016/j.nucengdes.2014.11.008>.

Perdu, F. & Vandroux, S., 2008. *System / CFD coupling for reactor transient analysis. An application to the gas fast reactor with CATHARE and TRIO\_U*. Anaheim, CA USA, ICAPP 2008 International Congress in Advances in Nuclear Power Plants, June 8-12, 2008, ISBN: 0-89448-061-8, Paper ID: 8094, pp. 1762-1769.

Petruzzi, A. & D'Auria, F., 2007. Thermal-hydraulic system codes in nuclear reactor safety and qualification procedures. *Science and Technology of Nuclear Installations*, 2008, [http://dx.doi.org/10.1155/2008/460795\(3\)](http://dx.doi.org/10.1155/2008/460795(3)), pp. 1-16.

RELAP5-3D, 2019. *RELAP-3D home*. [Online]  
Available at: <https://relap53d.inl.gov/SitePages/Home.aspx>  
[Accessed 25 09 2019].

Runchal, A. & Sagar, B., 1993. *PORFLOW: A multifluid multiphase model for simulating flow, heat transfer, and mass transport in fractured porous media. Report*. Washington, DC, U.S.: Nuclear Regulatory Commission, NUREG/CR--5991; CNWRA--92-003 ON: TI93008514.

Saghafi, M. & Ghofrani, M. B., 2016. Accident management support tools in nuclear power plants: A post-Fukushima review. *Progress in Nuclear Energy*, Volume 92, pp. 1-14, <https://doi.org/10.1016/j.pnucene.2016.06.006>.

Salko, R. K. & Avramova, M. N., 2015. *COBRA-TF subchannel thermal-hydraulic code (CTF) theory manual. Manual*. State College: Pennsylvania State University, CASL-U-2015-0054-000.

SALOME, 2019. *SALOME - YACS module*. [Online]  
Available at: <https://www.salome-platform.org/user-section/about/yacs>  
[Accessed 16 04 2019].

SALOME, 2019. *SALOME-the open source integration platform for numerical simulation*. [Online] Available at: <https://www.salome-platform.org/> [Accessed 07 08 2019].

Sanchez, V., Imke, U. & Ivanov, A., 2010. *SUBCHANFLOW: a thermal-hydraulic sub-channel program to analyse fuel rod bundles and reactor cores*. Cancun, Mexico, 17th Pacific Basin Nuclear Conference, October 24-30, 2010, ISBN: 978-607-95174-1-0, Paper ID: 17021.

Song, K.-N., Lee, S.-B. & Lee, S.-H., 2007. Performance evaluation of new spacer grid shapes for PWRs. *Nuclear Engineering and Technology*, 39(6), pp. 737-746, ISSN 1738-5733.

Sung, Y., Schueren, P. & Meliksetian, A., 1999. *VIPRE-01 modeling and qualification for pressurized water reactor non-LOCA thermal-hydraulic safety analysis*, Pittsburgh, U.S.: Westinghouse Electric Company, WCAP-15306-NP-A.

Tenchine, D. et al., 2012. Status of CATHARE code for sodium cooled fast reactors. *Nuclear Engineering and Design*, Volume 245, pp. 140-152, <https://doi.org/10.1016/j.nucengdes.2012.01.019>.

Toumi, I. et al., 2000. FLICA-4: a three-dimensional two-phase flow computer code with advanced numerical methods for nuclear applications. *Nuclear Engineering and Design*, 200(1-2), pp. 139–155, [https://doi.org/10.1016/S0029-5493\(99\)00332-5](https://doi.org/10.1016/S0029-5493(99)00332-5).

TRACE, 2016. *TRACE V5.1051 theory manual*. Washington, DC: Division of Safety Analysis Office of Nuclear Regulatory Research, U. S. Nuclear Regulatory Commission.

Walker, H. & Ni, P., 2011. Anderson acceleration for fixed-point iterations. *SIAM Journal on Numerical Analysis*, Volume 49, pp. 1715–1735, <https://doi.org/10.1137/10078356X>.

Wang, Q., Li, R. & He, G., 2018. Research status of nuclear power: A review. *Renewable and Sustainable Energy Reviews*, Volume 90, pp. 90-96, <https://doi.org/10.1016/j.rser.2018.03.044>.

Xu, Y. & Seker, V., 2010. *PARCS v3.0 U.S. NRC core neutronics simulator user manual. Manual*. Michigan: Department of Nuclear Engineering and Radiological Sciences, University of Michigan, UM-NERS-09-0001.

Yadigaroglu, G., 2005. Computational Fluid Dynamics for nuclear applications: from CFD to multi-scale CMFD. *Nuclear Engineering and Design*, 235(2-4), pp. 153–164, <https://doi.org/10.1016/j.nucengdes.2004.08.044>.

Yeckel, A., Lun, L. & Derby, J. J., 2009. An approximate block Newton method for coupled iterations of nonlinear solvers: Theory and conjugate heat transfer applications. *Journal of Computational Physics*, Volume 228, pp. 8566-8588, <https://doi.org/10.1016/j.jcp.2009.08.003>.

Ye, Q., 2016. Safety and effective developing nuclear power to realize green and low-carbon development. *Advances in Climate Change Research*, Volume 7, pp. 10-16, <https://doi.org/10.1016/j.accre.2016.06.005>.

Yetisir, M. et al., 2018. Fuel assembly concept of the canadian supercritical water-cooled reactor. *Journal of Nuclear Engineering and Radiation Science*, 4(1), pp. 1-7, <https://doi.org/10.1115/1.4037818>.

Zeng, M. et al., 2016. Nuclear energy in the Post-Fukushima Era : Research on the developments of the Chinese and worldwide nuclear power industries. *Renewable and Sustainable Energy Reviews*, Volume 58, pp. 147-156, <https://doi.org/10.1016/j.rser.2015.12.165>.

Zerkak, O., Kozlowski, T. & Gajev, I., 2015. Review of multi-physics temporal coupling methods for analysis of nuclear reactors. *Annals of Nuclear Energy*, Volume 84, pp. 225-233, <https://doi.org/10.1016/j.anucene.2015.01.019>.

Zhang, H. et al., 2018. Efficient simultaneous solution of multi-physics multi-scale nonlinear coupled system in HTR reactor based on nonlinear elimination method. *Annals of Nuclear Energy*, Volume 114, pp. 301-310, <https://doi.org/10.1016/j.anucene.2017.12.014>.

Zhang, H. et al., 2018. The improvement of coupling method in TINTE by fully implicit scheme. *Nuclear Science and Engineering*, Volume 190, pp. 156-175, <https://doi.org/10.1080/00295639.2018.1426299>.

Zhang, H. et al., 2018. An assessment of coupling algorithms in HTR simulator TINTE. *Nuclear Science and Engineering*, Volume 190, pp. 287–309, <https://doi.org/10.1080/00295639.2018.1442061>.



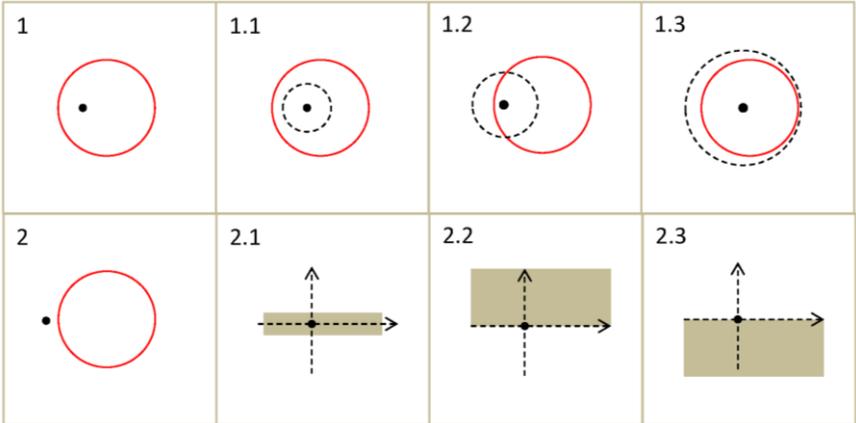
# Appendix A. Principles of the Mapping Toolkit

In this work of coupling TRACE and SCF, a user subroutine for field and mesh mapping was developed with the purpose of generating the two arrays for cases whose sections and channels might have arbitrary nodal, area, shape and location, automatically. The subroutine has the ability to handle complicated cases by covering all possible configurations. Since the calculation of the overlapping area for a channel with arbitrary shape is meaningless (SCF only need the area of the channels and don't care about their shapes), one key assumption assuming that all the channels are circles is proposed. The subroutine will first classify the channels into two basic classes according to their relative positions to the TRACE center (Appendix Figure 1-1 and Appendix Figure 1-2). The channels which cover the center point belong to class one and are further classified to three sub-classes according to their relative positions to the first TRACE radial section (Appendix Figure 1-1.1, 1.2, 1.3).

- 1) Sub-class 1.1: the channel fully covers the first TRACE ring;
- 2) Sub-class 1.2: the channel intersects the first TRACE ring;
- 3) Sub-class 1.3: the channel locates fully in the first TRACE ring.

The channels which separate from the center point belong to class two and are further classified to three sub-classes according to the relative positions of their centers and the x-axis.

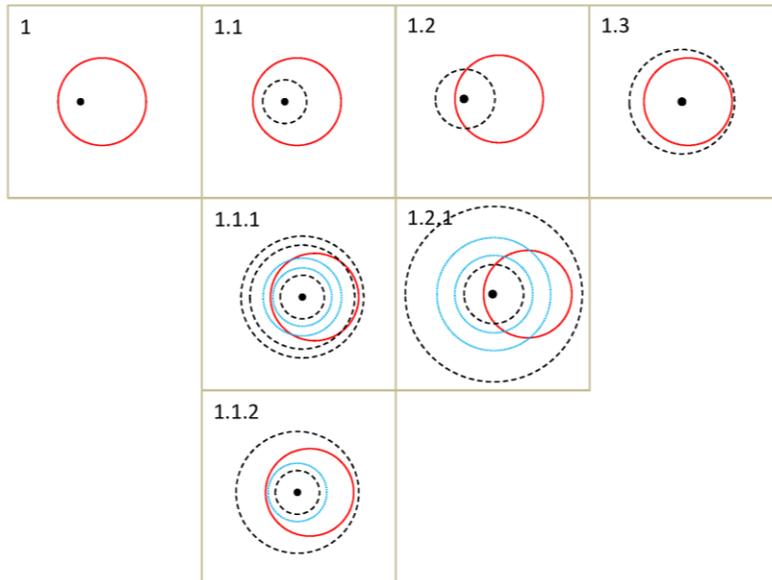
- 1) Sub-class 2.1: the channel center locates on the x-axis;
- 2) Sub-class 2.2: the channel center locates up upon x-axis;
- 3) Sub-class 2.3: the channel center locates under the x-axis.



Appendix Figure 1 – Two basic classes of the position of the SCF channels.

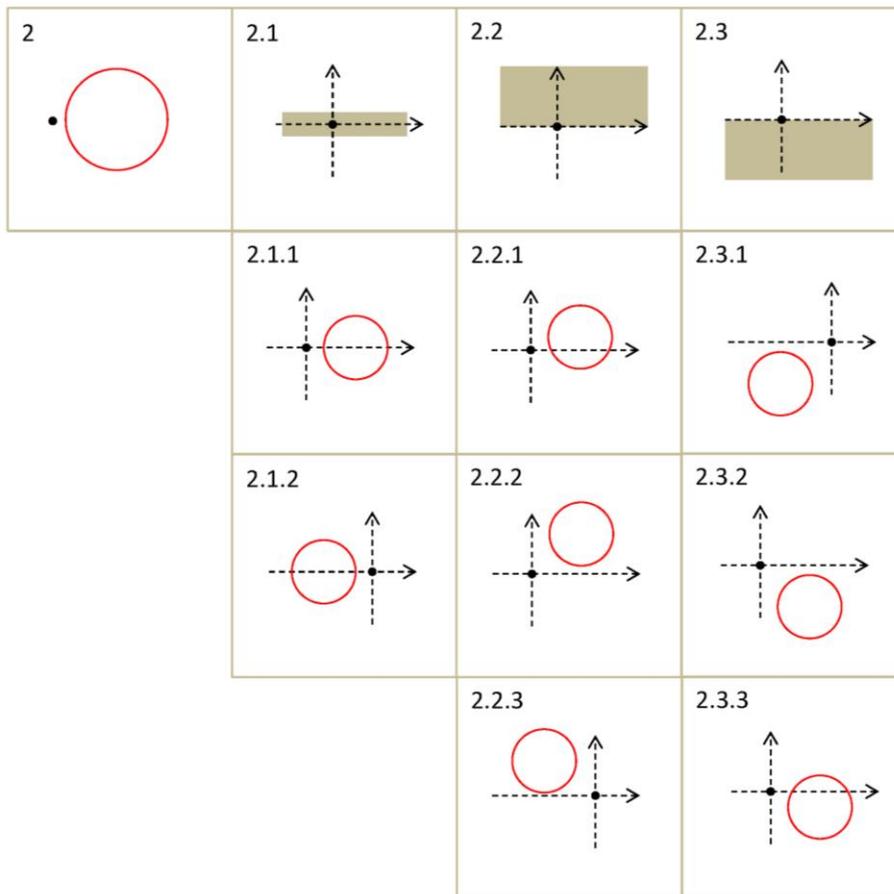
The sub-classes 1.1 to 1.3 can be further classified as Appendix Figure 2 presents. As before, the red circle represents the SCF channel. The black dash circles represent TRACE radial sections. The new blue circles represent some possible indeterminate TRACE radial sections.

- 1) Sub-class 1.1.1: the channel is intersected by more than one ring;
- 2) Sub-class 1.1.2: the channel is not intersected by any rings;
- 3) Sub-class 1.2.1: the channel is intersected by more than one ring other than the first ring.



**Appendix Figure 2 – Specific classification of class one.**

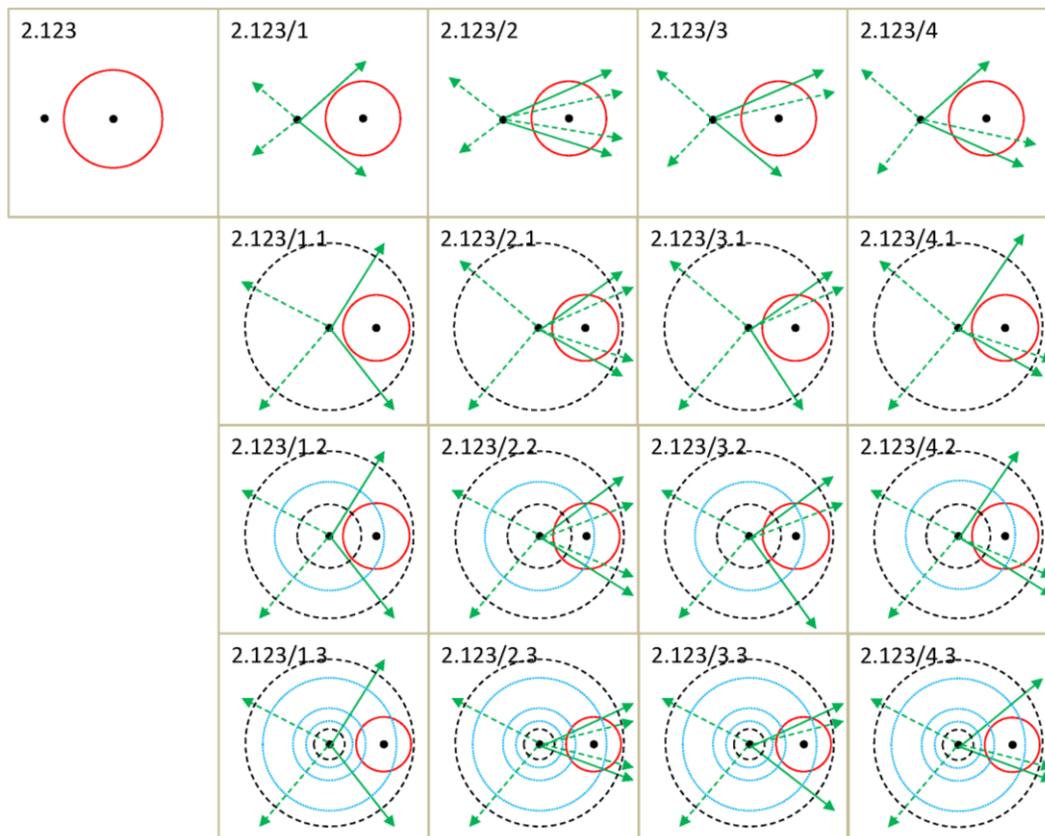
Sub-classes 2.1 to 2.3 can be further classified as Appendix Figure 3 presents.



**Appendix Figure 3 – Specific classification of class two.**

- 1) Sub-class 2.1.1: the channel center locates on the positive x-axis;
- 2) Sub-class 2.1.2: the channel center locates on the negative x-axis;
- 3) Sub-class 2.2.1: the channel intersects the positive x-axis;
- 4) Sub-class 2.2.2: the channel locates at the first quadrant and doesn't intersect the positive x-axis;
- 5) Sub-class 2.2.3: the channel center locates at the second quadrant;
- 6) Sub-class 2.3.1: the channel center locates at the third quadrant;
- 7) Sub-class 2.3.2: the channel locates at the fourth quadrant and doesn't intersect the positive x-axis;
- 8) Sub-class 2.3.3: the channel intersects the positive x-axis.

Gather sub-classes 2.1.1 to 2.3.3 together and label it as class 2.123. Now, take the azimuthal sectors into consideration. The total 2.123 can be clarified into four types: 2.123/1, 2.123/2, 2.123/3 and 2.123/4 (Appendix Figure 4). Then take the radial sectors into consideration, each of the four types can be further classified into three types (Appendix Figure 4). The green arrows represent the determinate TRACE azimuthal sectors. The dash green arrows represent the indeterminate TRACE azimuthal sectors. A full explanation of the cases is summarized after the figure. Compared with class two, the classifications of class one (Appendix Figure 2) don't include the azimuthal sectors. This is because the channel must intersect all of the azimuthal sectors since it covers the center point.



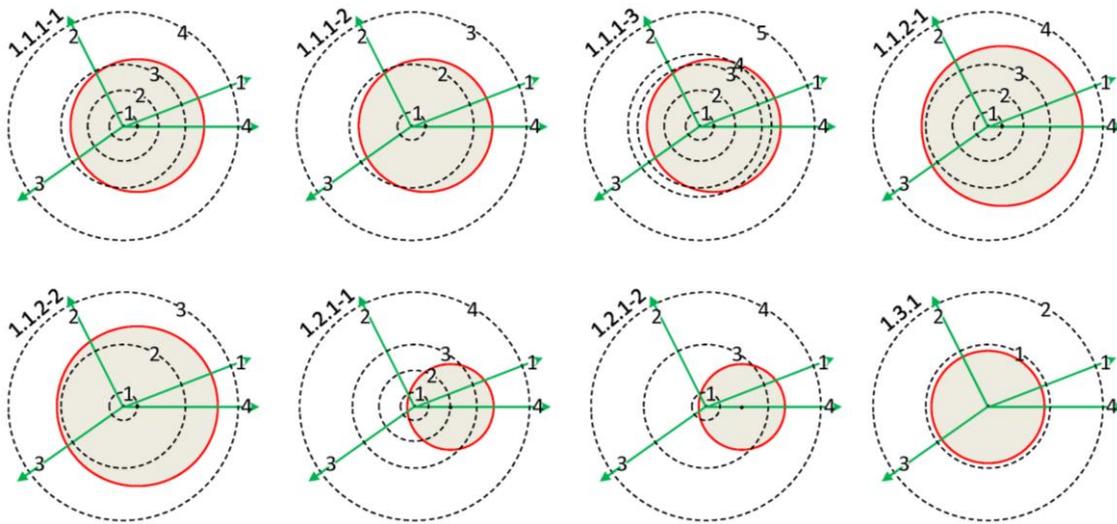
Appendix Figure 4 – Full classifications of class two.

- 1) 2.123/1: the channel is not intersected by any azimuthal sectors;
- 2) 2.123/2: both the upper and lower parts of the channel are intersected by the sectors;
- 3) 2.123/3: only the upper part of the channel is intersected by the sectors;
- 4) 2.123/4: only the lower part of the channel is intersected by the sectors.

The three further classifications of all the four types of 2.123/1, 2.123/2, 2.123/3 and 2.123/4 are more or less the same. Thus their explanation can be put together.

- 1) 2.123/1.1, 2.1, 3.1, 4.1: the channels locate fully in the first ring;
- 2) 2.123/1.2, 2.2, 3.2, 4.2: the channel intersects with more than one rings including the first ring;
- 3) 2.123/1.3, 2.3, 3.3, 4.3: the channel separates from the first ring and may intersect or don't intersect rings.

Based on the recognized channel type, the subroutine will then calculate the overlapping areas between SCF channels and TRACE sections and finally generate two contribution arrays storing the mapping proportions. In order to fully test the subroutine, cases should be designed to cover all possibilities. Appendix Figure 5 lists the test cases for class one channels. The instruction follows the figure.

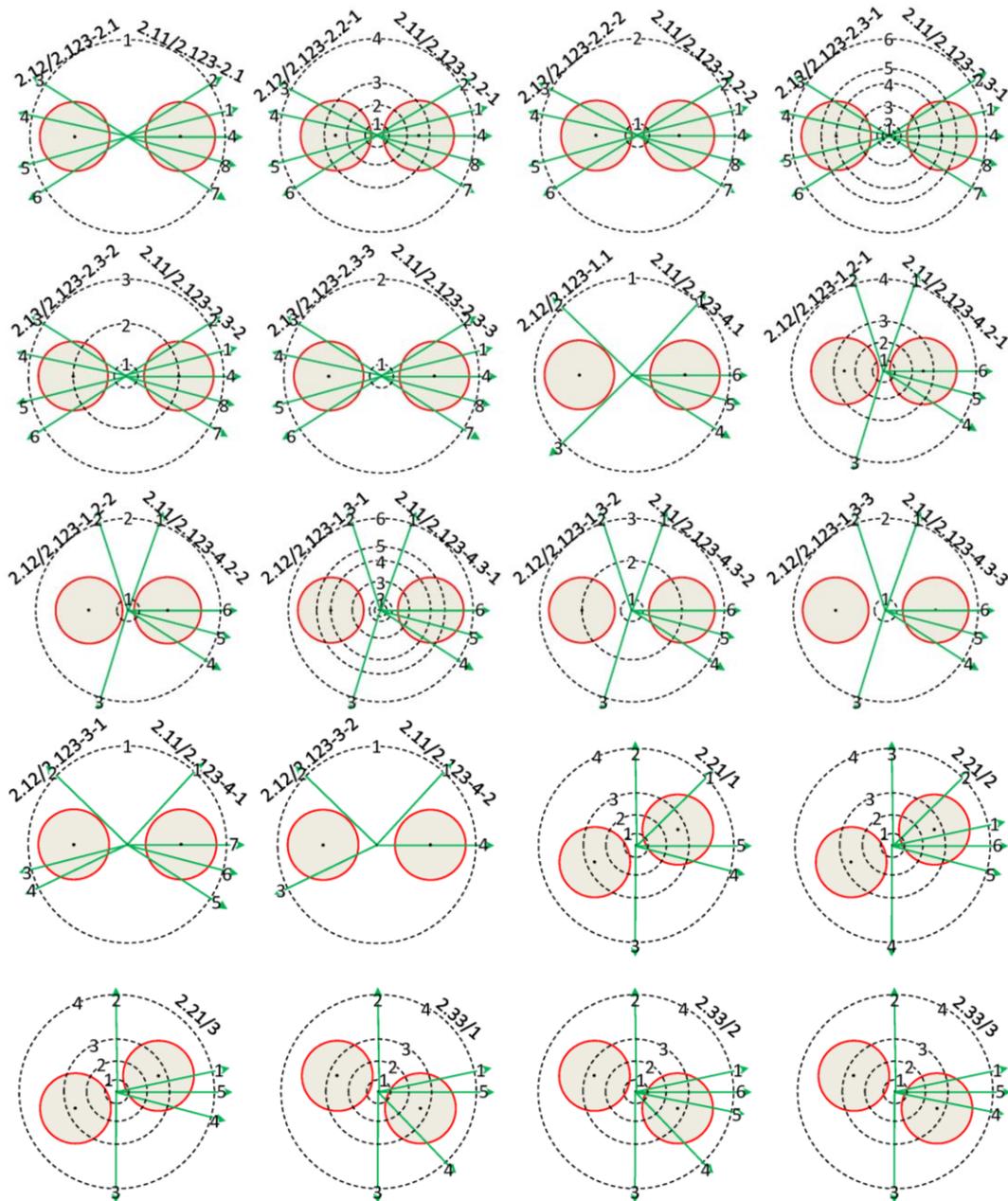


Appendix Figure 5 – Test cases for channels belonging to class one.

The introduction of test cases should be combined with the classifications in Appendix Figure 2.

- 1) 1.1: The channel area totally covers the first vessel radial section.
  - a. 1.1.1: The channel boundary intersects one of the vessel radial circles, either the innermost one of the outermost one.
    - i. 1.1.1-1: More than one vessel radial rings are covered by the channel.
    - ii. 1.1.1-2: Only one vessel radial ring is covered by the channel and only one ring intersects the channel.
    - iii. 1.1.1-3: More than one vessel rings are covered by the channel and intersect the channel.
  - b. 1.1.2: There is no intersection between the channel and vessel radial circles.
    - i. 1.1.2-1: More than three vessel rings are covered by the channel.
    - ii. 1.1.2-2: Only two vessel rings are covered by the channel.
- 2) 1.2: The channel area intersects the first vessel radial ring.
  - a. 1.2.1: The channel area intersects the first vessel radial ring.
    - i. 1.2.1-1: More than two vessel radial rings intersect the channel.
    - ii. 1.2.1-2: Only two vessel rings intersect the channel.
- 3) 1.3: The vessel first radial circle totally covers the channel area.
  - a. 1.3.1: There is only one condition.

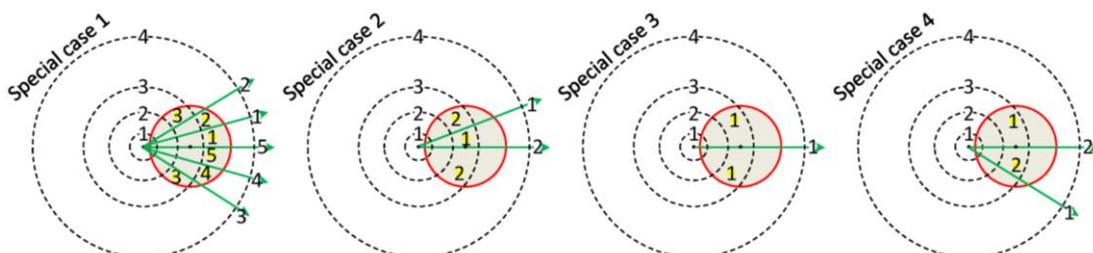
Test cases for class two include more items (Appendix Figure 6). Take reference to Appendix Figure 4, according to the position of the channels' center, they can be divided into eight types. Since each test case has two channels which are central symmetrical to each other. Four types of cases are enough to cover all the possibilities. Since the 360-degree line is always a boundary, cases which have centers in quadrant 1 and 4 must be divided into two sub-cases respectively according to whether they intersect the axis x. If taking the azimuthal sectors into consideration, most of the eight cases above will be re-divided into nine types. Since the radial sections processing module is almost isolated from the module which processes the azimuthal sections (this is the fact for almost all the channel types except 2.123/1.1, 2.123/1.2 and 2.123/1.3), it is not necessary to test all the nine conditions. That is to say, an additional model to test the azimuthal section processing module will not be needed if the previous model has already tested such kind functions.



Appendix Figure 6 – Test cases for channels belonging to class two.

- 1) 2.1.1 – 2.1.2: These cases include two channels - channel 2.1.1 and 2.1.2.
  - a. 2.123 – 2 – Domi: These cases focus on condition 2.123/2 of channel 2.1.1 and channel 2.2.2. The majority functions of the radial and azimuthal processing module can be sufficiently examined.
    - i. 2.1: Corresponds to 2.123/2.1, the channel locates fully in first vessel circle and is intersected by azimuthal boundaries both on its upper and lower parts.
    - ii. 2.2: Corresponds to 2.123/2.2, the channel intersects the first vessel circle and is intersected by azimuthal boundaries both on its upper and lower parts.
      - 1: The channel intersects more than one vessel circles.
      - 2: The channel intersects only the first circle.
    - iii. 2.3: Corresponds to 2.123/2.3, the channel is separated from the first vessel circle and is intersected by azimuthal boundaries both on its upper and lower parts.
      - 1: More than one vessel inner circles are separated from the channel and more than one circle intersects the channel.
      - 2: Only the first vessel inner circle is separated from the channel and only the second circle intersects the channel.
      - 3: Only the first vessel inner circle is separated from the channel and no vessel circles intersect the channel.
  - b. 2.123 – 4\_1\_3: With the majority, functions were tested by the models in 2.123-2-Domi, several remaining functions should be tested using channel conditions 2,123/1, 2,123/3 and 2,123/4.
- 2) 2.2.1 – 2.3.1: These cases include two channels - channel 2.2.1 and 2.3.1. There are three conditions to be tested as a supplement to the testing models above. Channel 2.2.1 is intersected by azimuthal boundaries on lower part while channel 2.3.1 is not intersected by any azimuthal boundaries.
  - a. 1: There is no first quadrant azimuthal boundaries in the lower part of the channel area.
  - b. 2: There are first quadrant azimuthal boundaries in the lower part of the channel area.
  - c. 3: There are only two fourth quadrant azimuthal boundaries in lower part of channel.
- 3) 2.2.2 – 2.3.1: The two types of the channel have already been tested above.
- 4) 2.3.3 – 2.2.3: These cases include two channels - channel 2.3.3 and 2.2.3. There are three conditions to be tested as a supplement to the testing models above. Channel 2.3.3 is intersected by azimuthal boundaries on the upper part while channel 2.3.1 is not intersected by any azimuthal boundaries.
  - a. 1: There is no fourth quadrant azimuthal boundaries in upper part of the channel area.
  - b. 2: There are fourth quadrant azimuthal boundaries in the upper part of the channel.
  - c. 3: There is one first quadrant azimuthal boundaries in upper part of the channel area.

A hiding trap in the testing process is that the uppermost and the lowermost parts of the channel may belong to the same TRACE section, which must be tested by some specially designed cases. Four additional cases were applied to test the special conditions which have a special relationship with 360 degrees azimuthal boundary (Appendix Figure 7).



Appendix Figure 7 – Testing cases for special conditions.

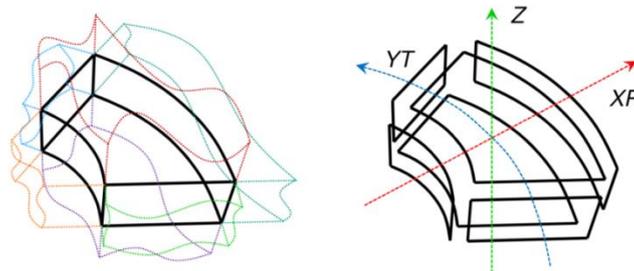
- 1) 1: Both the lower and upper parts of the channel area are intersected by more than one vessel azimuthal boundaries.
- 2) 2: Both the lower and upper parts of the channel area are only intersected by one vessel azimuthal boundary.
- 3) 3: The vessel core only has one section and the channel is intersected by the boundary.
- 4) 4: The lower part of the channel is crossed by two azimuthal boundaries while the upper part has no such boundaries crossed.

The subroutine was sufficiently tested by the testing cases (Appendix Figure 5, Appendix Figure 6, and Appendix Figure 7) and was proofed good robustness. As a matter of fact, the mapping between sections of TRACE model and channels of SCF model for real Nuclear Power Plant (NPP) may not make full use of all the subroutine's potential capabilities, which is determined by the normally regular uniform sections and channels. Nevertheless, if some unusual and strange configurations are to be implemented, the subroutine is still available. It is a tool processing a quite flexible geometry coupling between TRACE and SCF anyway and may also be applied to other codes' coupling.



## Appendix B. The peculiarity of the hydraulic coupling in TRACE/TrioCFD

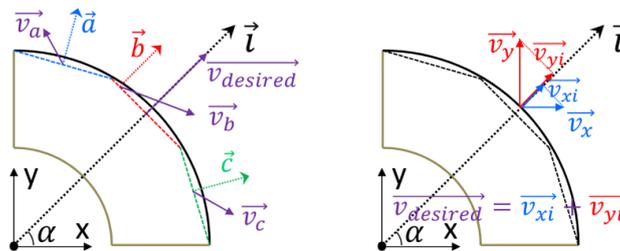
The different numeric of the involved solvers (TRACE and TrioCFD) e.g. the fact that the velocity fields in TRACE are all scalars while in TrioCFD they are vectors, requires a discussion of the peculiarities of the implemented coupling approach. Generally speaking, TRACE has its own logic to overcome the directionality problem of the velocity fields. Take the typical TRACE annular cell as shown in Appendix Figure 8. Normally it has six neighboring cells and six corresponding edges. The edges are in three directions under cylindrical coordinate: x or radial direction - XR, y or azimuthal direction - YT, and z or axial direction - Z. The direction of the velocity is based on the edge normal vector of each edge.



a. six cells (colorful) surrounding a cell (black)    b. six faces compose a cell

Appendix Figure 8 – A typical TRACE annular cell has six neighboring cells and is composed of six faces.

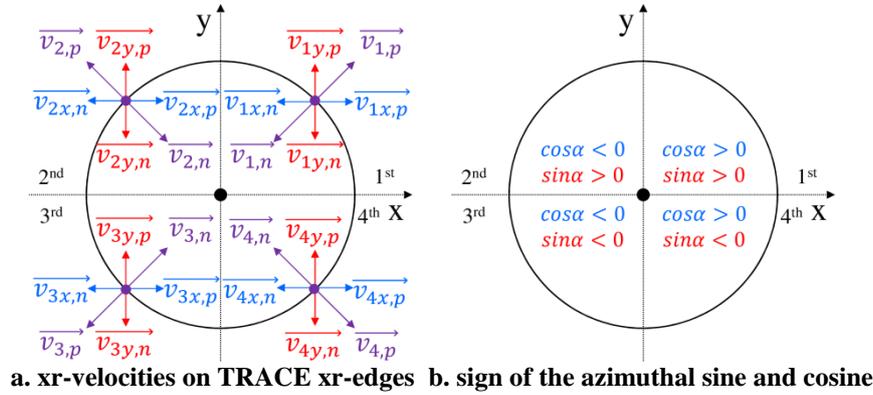
If the TRACE edge mesh in Appendix Figure 8b is examined, each real TRACE annular edge in XR direction is represented by a group of quadrangles. Suppose that there is a quadrant cell whose outer edge is approximated by three quadrangles (blue, red, and green), as shown in Appendix Figure 9. The normal vector of the edge is  $\vec{l}$ . While the normal vectors of the three quadrangles ( $\vec{a}$ ,  $\vec{b}$ ,  $\vec{c}$ ) varies in large degree, Appendix Figure 9a. Moreover, the velocities on the quadrangles could be in any direction ( $\vec{v}_a$ ,  $\vec{v}_b$ ,  $\vec{v}_c$ ). Since the radial velocity  $\vec{v}_{desired}$  on the edge is defined on the direction of  $\vec{l}$ , its calculation could become rather complex if the absolute velocity of those quadrangles is used. Alternatively, the x-velocity and y-velocity are derived separately for those quadrangles and they are integrated and averaged along x and y direction to two velocities on that edge:  $\vec{v}_x$  and  $\vec{v}_y$ . Then, their projection on  $\vec{l}$  is calculated and added together forming the  $\vec{v}_{desired}$ , see Appendix Figure 9b.



a. the velocities on xr faces of TRACE mesh    b. the desired velocity

Appendix Figure 9 – The velocities on the faces (the TRACE mesh) approximating the cylindrical (actual TRACE vessel geometry) are integrated and calculated to represent the velocity on the normal direction of a TRACE edge on xr-direction.

Suppose the angle between  $\vec{l}$  and the x-axis is  $\alpha$ . Now the final formula become  $\vec{v}_{desired} = \vec{v}_{xi} + \vec{v}_{yi} = \vec{v}_x \cdot \cos\alpha + \vec{v}_y \cdot \sin\alpha$ , Appendix Figure 9b. Pay attention that the signs of the velocities in TRACE only depends on the direction of the edges' normal vector, while the signs of the velocities derived from TrioCFD fully follow the Cartesian coordinates. The correct translation from  $\vec{v}_x, \vec{v}_y$  to  $\vec{v}_{desired}$  is necessary. The case shown in Appendix Figure 10 is proposed to demonstrate the conditions and based on which, the velocity translation is discussed.

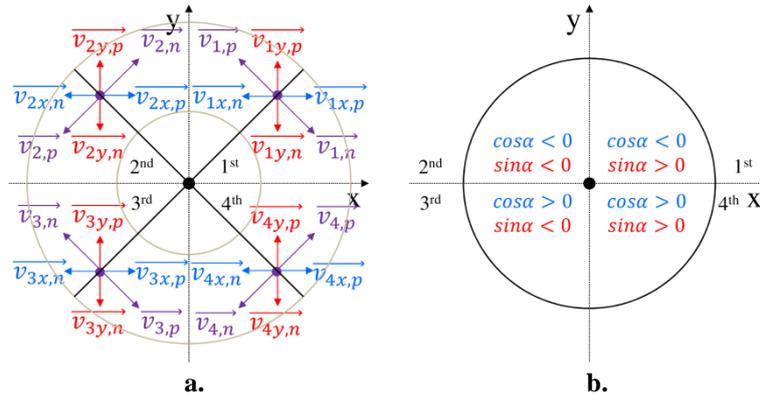


**Appendix Figure 10 – The possible xr-velocities on TRACE xr-edges can be the same or opposite to the face normal vector and decomposed to velocities on x and y-direction whose sign corresponds to the azimuthal sine and cosine.**

Take the velocities in the second quadrant for instance (Appendix Figure 10a), the purple velocities represent the desired velocities at the TRACE XR edge, the blue ones represent the x-velocities derived from TrioCFD, and the red ones represent the y-velocities derived from TrioCFD. The subscript – 1, 2, 3, 4 tells the velocity to locate in which quadrant. The subscripts – x and y tell whether this is an x- or y-velocity field from TrioCFD. The subscript p indicates this is a positive value while n indicates a negative one. It is apparent that the signs of the velocities of TRACE only depend on its normal vectors while that of TrioCFD depend on the Cartesian coordinates. According to the formulation  $\vec{v}_{2,p} = \vec{v}_{xi} + \vec{v}_{yi} = \vec{v}_{2x,n} \cdot \cos\alpha + \vec{v}_{2y,p} \cdot \sin\alpha$ ,  $\vec{v}_{2,n} = \vec{v}_{xi} + \vec{v}_{yi} = \vec{v}_{2x,p} \cdot \cos\alpha + \vec{v}_{2y,n} \cdot \sin\alpha$  and the trigonometric function's signs presented in Appendix Figure 10b, the two contributions –  $\vec{v}_{xi}$  and  $\vec{v}_{yi}$  always contribute to the desired velocities –  $\vec{v}_{2,p}$  and  $\vec{v}_{2,n}$ , only if, they are located in the same side of the edge. For cases where the three velocities are located on different sides, both contribution and subtraction would be introduced onto the desired velocities. Nevertheless, they reflect real physical significance.

After the discussion of the calculation of the XR velocities, the velocities on the YT edges also call for examination. Suppose there are four symmetrically arranged YT edges, Appendix Figure 11. All of the symbols and variables displayed in Appendix Figure 11a represent exactly the same concepts with Appendix Figure 10a. The essential difference is that the trigonometric function's signs in Appendix Figure 11b turn a clockwise quadrant compared to that in Appendix Figure 10b. This is because the normal vectors of the YT edges turn a clockwise angle compared to that of the XR edges.

Take the velocities in the second quadrant for instance, according to the formulation  $\vec{v}_{2,p} = \vec{v}_{xi} + \vec{v}_{yi} = \vec{v}_{2x,n} \cdot \cos\alpha + \vec{v}_{2y,n} \cdot \sin\alpha$ ,  $\vec{v}_{2,n} = \vec{v}_{xi} + \vec{v}_{yi} = \vec{v}_{2x,p} \cdot \cos\alpha + \vec{v}_{2y,p} \cdot \sin\alpha$  and the trigonometric function's signs presented in Appendix Figure 11b, the two contributions –  $\vec{v}_{xi}$  and  $\vec{v}_{yi}$  always contribute to the desired velocities –  $\vec{v}_{2,p}$  and  $\vec{v}_{2,n}$ , only if, they are located on the same side of the edge. For cases where the three velocities are located on different sides, both contribution and subtraction would be introduced onto the desired velocities. Nevertheless, they reflect real physical significance.



**Appendix Figure 11 – The possible yt-velocities on TRACE yt-edges can be the same or opposite to the face normal vector and decomposed to velocities on x- and y-direction whose sign corresponds to the azimuthal sine and cosine.**

Up to this point, the only remaining edges are those on axial or Z direction. Look back to Figure 5-14b, it can be stated that the TRACE Z edges are also approximated by groups of small faces. The derived velocities from TrioCFD are exactly located on those small faces, so they have to be integrated to get the Z velocities of TRACE edges. Nevertheless, thanks to the consistency of the Z direction of the TRACE coordination system and the Cartesian coordinates, no further translations are needed for the velocities on Z direction.



# List of Acronyms

<b>Betti Lab</b>	Bettis Atomic Power Laboratory
<b>BWR</b>	Boiled Water Reactor
<b>CEA</b>	French Alternative Energies and Atomic Energy Commission
<b>CFD</b>	Computational Fluid Dynamic
<b>CHF</b>	Critical Heat Flux
<b>CMFD</b>	Computational Multi-Fluid Dynamics
<b>CPR</b>	Critical Power Ratio
<b>DES</b>	Detached Eddy Simulation
<b>DIAS</b>	Dynamic-Implicit-Additional-Source
<b>DNS</b>	Direct Numerical Simulation
<b>ECI</b>	Exterior Communication Interface
<b>GFR</b>	Gas Fast Reactor
<b>GT-MHR</b>	Gas-Turbine – Modular Helium Reactor
<b>GUI</b>	Graphical User Interface
<b>ICoCo</b>	Interface for Code Coupling
<b>IDO</b>	Inertial Domain Overlapping
<b>INL</b>	Idaho National Laboratory
<b>INR</b>	Institute for Neutron Physics and Reactor Technology
<b>IO</b>	Input and Output
<b>KAERI</b>	Korea Atomic Energy Research Institute
<b>KIT</b>	Karlsruhe Institute of Technology
<b>KNS</b>	Karlsruhe compact sodium boiling loop
<b>KTH</b>	KTH Royal Institute of Technology
<b>LES</b>	Large Eddy Simulation
<b>LWR</b>	Light Water Reactor
<b>MD</b>	Molecular Dynamics

<b>MIMD</b>	Multiple Instruction Multiple Data
<b>MISD</b>	Multiple Instruction Single Data
<b>MPI</b>	Message Passing Interface
<b>MSLB</b>	Main StreamLine Break
<b>NEA</b>	Nuclear Energy Agency
<b>NPP</b>	Nuclear Power Plant
<b>NRC</b>	Nuclear Regulatory Commission
<b>OS</b>	Operator Splitter
<b>OSSI</b>	Operator Splitter Semi Implicit
<b>PNNL</b>	Pacific Northwest National Laboratory
<b>PSI</b>	Paul Scherrer Institute
<b>PVM</b>	Parallel Virtual Machines
<b>PWR</b>	Pressurized Water Reactor
<b>RANS</b>	Reynolds-Averaged Navier-Stokes
<b>RPV</b>	Reactor Pressure Vessel
<b>SCF</b>	SubChanFlow
<b>SCWR</b>	Super Critical Water Reactor
<b>SIDO</b>	Stabilized Inertial Domain Overlapping
<b>SFR</b>	Sodium cooled Fast Reactor
<b>SG</b>	Steam Generator
<b>SGTR</b>	Steam Generator Tube Ruptures
<b>SIMD</b>	Single Instruction Multiple Data
<b>SISD</b>	Single Instruction Single Data
<b>SJTU</b>	Shanghai Jiaotong University
<b>SNU</b>	Seoul National University
<b>TUM</b>	Technical University of Munich
<b>UDF</b>	User-Defined Functions
<b>UIUC</b>	The University of Illinois at Urbana-Champaign

<b>Uni Fukui</b>	University of Fukui
<b>UniPi</b>	University of Pisa
<b>V.D.F.</b>	Finite Volume Differences
<b>V.E.F.</b>	Finite Volume Elements
<b>VHTR</b>	Very High-Temperature Reactor
<b>VMF</b>	Velocity Matching Faces
<b>XJTU</b>	Xian Jiaotong University



# Nomenclature

$a, b, c$	quantities calculated from known parameters or variables
$A$	a known matrix or flow area of edge
$C$	friction factor, equals $\Delta x \cdot \rho \cdot K$
$e$	internal energy
$g$	gravity force
$K$	form-loss coefficient or wall friction coefficient
$P$	fluid pressure or total pressure
$q$	heat source
$t$	time
$T$	temperature
$V$	magnitude of the velocity
$x$	distance

## *Greek*

$\rho$	density
$\Delta H$	height difference
$\Delta P$	pressure difference
$\Delta t$	time step size
$\Delta x$	cell length for 1D components
$\delta P$	pressure difference between two linear iteration step
$\delta T$	Temperature difference between two linear iteration step

## *Subscripts*

$j+1$	downstream cell-center index
$j+1/2$	downstream cell-edge index
$j$	the local cell-center index

---

---

$j-1/2$	upstream cell-edge index
$acc$	variables associated with acceleration
$con$	variables associated with convection
$fric$	variables associated with friction
$tot$	total

### ***Superscripts***

$i$	the current linear iteration step
$i+1$	the next linear iteration step
$n$	the current time quantity
$n+1$	the next time quantity
$\sim$	the stabilizer quantity

### ***Operations***

$\langle \rangle$	averaged value of adjacent cells or edges
$\ $	absolute value